

precisely

MapBasic

Справочник

Версия 2019



Содержание

1 - Введение в MapBasic

Принятые обозначения.....	5
Основные сведения о языке программирования.....	6
Основы MapBasic.....	6
Хактеристика среды разработки MapBasic IDE.....	9
Функции.....	11
Работа с таблицами.....	17
Работа с файлами (не таблицами)	20
Работа с картами и графическими объектами.....	21
Работа с окнами отчётов.....	27
Создание интерфейса пользователя.....	27
Обмен данными с другими программами	31
Специальные операторы и функции.....	32
Техническая поддержка.....	32
Авторские права.....	35
Об открытом программном обеспечении.....	36

2 - Новые и улучшенные операторы и функции MapBasic

Новые функции и операторы MapBasic.....	38
Добавления к существующим функциям и операторам.....	38

3 - Алфавитный справочник языка MapBasic

Принятые обозначения функций и операторов.....	42
Описание функций и операторов.....	42
Приложение А: Библиотеки HTTP и FTP	

О библиотеках функций для работы с HTTP и FTP.....	1156
Процедура MICloseContent().....	1156
Процедура MICloseFtpConnection().....	1157
Процедура MICloseFtpFileFind().....	1157
Процедура MICloseHttpConnection().....	1158
Процедура MICloseHttpFile().....	1159
Процедура MICloseSession().....	1159
Функция MICreateSession().....	1160
Функция MICreateSessionFull().....	1161
Функция MIErrorDlg().....	1163
Функция MIFindFtpFile().....	1165
Функция MIFindNextFtpFile().....	1166
Функция MIGetContent().....	1166
Функция MIGetContentBuffer().....	1167
Функция MIGetContentLen().....	1168
Функция MIGetContentString().....	1169
Функция MIGetContentToFile().....	1169
Функция MIGetContentType().....	1170
Функция MIGetCurrentFtpDirectory().....	1171
Функция MIGetErrorCode().....	1172
Функция MIGetErrorMessage().....	1173
Функция MIGetFileURL().....	1173
Функция MIGetFtpConnection().....	1174
Функция MIGetFtpFile().....	1175
Функция MIGetFtpFileFind().....	1177
Процедура MIGetFtpFileName().....	1178

Функция MIGetHttpConnection().....	1179
Функция MIIsFtpDirectory().....	1180
Функция MIIsFtpDots().....	1180
Функция MIOpenRequest().....	1181
Функция MIOpenRequestFull().....	1182
Функция MIParseURL().....	1184
Функция MIPutFtpFile().....	1186
Функция MIQueryInfo().....	1187
Функция MIQueryInfoStatusCode().....	1188
Функция MISaveContent().....	1190
Функция MIsendRequest().....	1190
Функция MIsendSimpleRequest().....	1191
Функция MISetCurrentFtpDirectory().....	1192
Функция MISetSessionTimeout().....	1193
Приложение В: Библиотека XML	

О библиотеке XML.....	1197
Процедура MIXmlAttributeListDestroy().....	1197
Функция MIXmlDocumentCreate().....	1198
Процедура MIXmlDocumentDestroy().....	1198
Функция MIXmlDocumentGetNamespaces().....	1199
Функция MIXmlDocumentGetRootNode().....	1200
Функция MIXmlDocumentLoad().....	1201
Функция MIXmlDocumentLoadXML().....	1202
Функция MIXmlDocumentLoadXMLString().....	1203
Функция MIXmlDocumentSetProperty().....	1204
Функция MIXmlGetAttributeList().....	1205
Функция MIXmlGetChildList().....	1206
Функция MIXmlGetNextAttribute().....	1207
Функция MIXmlGetNextNode().....	1208
Процедура MIXmlNodeDestroy().....	1209
Функция MIXmlNodeGetAttributeValue().....	1209
Функция MIXmlNodeGetFirstChild().....	1210
Функция MIXmlNodeGetName().....	1211
Функция MIXmlNodeGetParent().....	1212
Функция MIXmlNodeGetText().....	1213
Функция MIXmlNodeGetValue().....	1214
Процедура MIXmlNodeListDestroy().....	1214
Процедура MIXmlSCDestroy().....	1215
Функция MIXmlSCGetLength().....	1216
Функция MIXmlSCGetNamespace().....	1216
Функция MIXmlSelectNodes().....	1217
Функция MIXmlSelectSingleNode().....	1218
Приложение С: Таблица символов	

Таблица символов.....	1221
Приложение D: Сведения об операторах	

Числовые операторы.....	1223
Операторы сравнения.....	1224
Логические операторы.....	1226
Географические операторы.....	1226
Автоматическое преобразование типов.....	1229
Замещающие символы.....	1230
Приложение E: Файл определителей MapBasic	

Файлы MAPBASIC.DEF.....	1232
Приложение F: Введение в справку MapBasic	

Поиск раздела в справочной системе.....	1280
Справочная система.....	1280
О программе просмотра справки.....	1281
Просмотр справки на другом компьютере.....	1281
Команды меню.....	1282
Описание диалогов MapBasic.....	1301

1 - Введение в MapBasic

Добро пожаловать в среду разработчика MapBasic! MapBasic – мощный и одновременно простой в использовании язык программирования, который позволит вам создавать собственные приложения в среде MapInfo® Pro.

В этом справочнике описаны все операторы и функции языка программирования MapBasic. Основы программирования на MapBasic и сведения об использовании среды разработки программ MapBasic приведены в *Руководстве пользователя MapBasic*.

В этом разделе:

В этом разделе

Принятые обозначения.....	5
Основные сведения о языке программирования.....	6
Основы MapBasic.....	6
Характеристика среды разработки MapBasic IDE.....	9
Функции.....	11
Работа с таблицами.....	17
Работа с файлами (не таблицами)	20
Работа с картами и графическими объектами.....	21
Работа с окнами отчётов.....	27
Создание интерфейса пользователя.....	27
Обмен данными с другими программами	31
Специальные операторы и функции.....	32
Техническая поддержка.....	32
Авторские права.....	35
Об открытом программном обеспечении.....	36

Принятые обозначения

В этом Справочнике приняты следующие условные обозначения в тексте:

Условное обозначение	Значение
If, Call, Map, Browse, Area	Слова и термины, начинающиеся с заглавной буквы, выделенные жирным шрифтом – это зарезервированные ключевые слова MapBasic. В этом справочнике у всех зарезервированных ключевых слов первая буква набрана с заглавной буквы; однако, при написании MapBasic-программ эти зарезервированные ключевые слова можно вводить любым способом: в верхнем регистре, в нижнем или смешанном.
Main, Pen, Object	Слова, начинающиеся с заглавной буквы, но набранные нежирным шрифтом, обычно являются именами специальных процедур или типами переменных.
<i>table, handler, window_id</i>	Слова, набранные курсивом – параметры операторов MapBasic. При оформлении оператора MapBasic, требуется задать каждому параметру необходимое выражение.
[<i>window_id</i>], [Interactive]	Зарезервированные ключевые слова или параметры в квадратных скобках являются необязательными.
{ On Off }	Синтаксическое выражение внутри фигурных скобок содержит список ключевых зарезервированных слов или параметров, разделенных вертикальной чертой (). Требуется выбрать один из перечисленных вариантов. Например, в примере, показанном слева ({ On Off }), требуется выбрать либо On , либо Off .
<code>"Note "Hello,world!"</code>	Примеры программ набраны шрифтом Courier.

Основные сведения о языке программирования

Далее приведены основные сведения о языке программирования MapBasic. Задача ставится слева; соответствующие ей операторы и функции перечислены справа и выделены **жирным** шрифтом. После имен функций стоят скобки ().

Основы MapBasic

Переменные

Объявление локальных и глобальных переменных:	Dim, Global
Изменение размера массива переменных:	ReDim, UBound(), UnDim
Объявление заданной структуры данных:	Тип

Большие целочисленные переменные

MapBasic использует целочисленные типы SmallInt, Integer, and LargeInt. LargeInt – тип хранения целых 64-разрядных (8 байт) чисел; используйте **оператор Dim**, чтобы создавать переменные LargeInt.

Короткое целое – SmallInt	Короткое целое число от -32 767 до 32 767 включительно; используется два байта.
Целое – Integer	Целое число от -2 147 483 648 до +2 147 483 647 включительно; используется четыре байта.

Длинное целое число – LargeInt

Целые числа от -9 223 372 036 854 775 808 до +9 223 372 036 854 775 807 включительно; хранятся в 8 байтах.

Переменные LargeInt поддерживаются MapBasic, но они не являются типом, полностью поддерживаемым в таблицах, потому что они могут работать некорректно с большими значениями (например, в тематических картах). Некоторые математические вычисления или функции создают значения типа LargeInt. Если значение выходит за пределы диапазона типа Integer, а приложение попытается сохранить результат в переменной Integer, произойдёт ошибка переполнения.

Следующие функции возвращают значения LargeInt:

- **Оператор Seek**
- **Функция LOF()**
- **Функция Server_GetODBCConn()**
- **Функция Server_GetODBCStmt()**

Следующие операторы допускают использование значений LargeInt позиций в файле:

- **Оператор Seek**
- **Оператор Get**

Следующие функции возвращают значения IntPtr, когда запрашиваются определённые атрибуты.

- **функция SystemInfo()** с атрибутами SYS_INFO_APPLICATIONWND, SYS_INFO_MAPINFOWND, SYS_INFO_MDICLIENTWND, and SYS_INFO_APPIDISPATCH.
- **функция WindowInfo()** с атрибутом WIN_INFO_WND.

Примечание: Зависящий от платформы тип, используемый для указателя или обработчика. Это помогает писать приложения, которые работают в обеих версиях MapInfo Pro – и 32-х и 64-разрядной. 64-разрядная версия MapInfo Pro рассматривает такие значения, как имеющие тип LargeInt, а 32-разрядная версия MapInfo Pro рассматривает как целое – Integer.

Циклы и другие управляющие операторы

Циклы:

For...Next, Exit For, Do...Loop, Exit Do, While...Wend

Ветвление:

If...Then, Do Case, GoTo

Другие управляющие операторы:

End Program, Terminate Application, End MapInfo

Вывод на печать и экспорт

Печать содержимого окна:	PrintWin
Печать в окне Сообщений:	Print
Настройка отчета:	Layout, Create Frame, Set Window
Экспорт окна в файл:	Экспорт окна
Управление принтером:	Set Window, Window Info()

Процедуры (Main и подпрограммы Sub)

Определение процедур:	Declare Sub, Sub...End Sub
Вызов процедуры:	Call
Выход из процедуры:	Exit Sub
Главная процедура Main:	Операции

Обработка ошибок

Настройка процедуры обработки ошибок:	OnError

Возвратить сведения об очередной ошибке:	Err(), Error\$()
Выход из процедуры обработки ошибок:	Resume
Моделирование ошибки:	Error

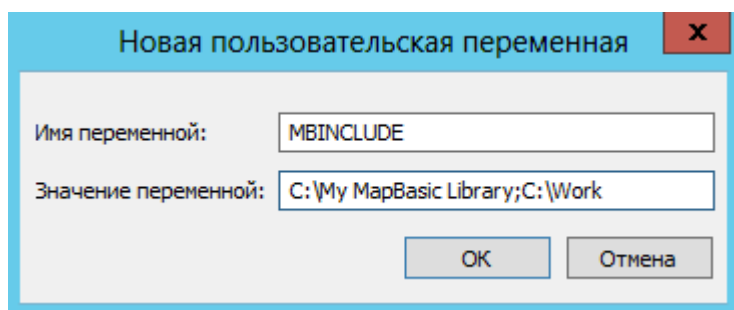
Хактеристика среды разработки MapBasic IDE

Задание маршрутов поиска для операторов MapBasic Include и модульных файлов

Для того чтобы упростить разработку MapBasic-библиотек определений и модулей, можно настроить переменные среды для поиска Include-файлов (DEF), необходимых для компиляции, и модульных файлов (MBO), используемых при линковке. По умолчанию MapBasic выполняет поиск этих файлов по адресу, заданному в коде MapBasic-программы, а затем в папке, где установлена программа MapBasic. Задав эти переменные среды, можно указать дополнительные папки, в которых будет выполняться поиск сразу после поиска по адресу, указанному в коде программы, но прежде поиска в папке MapBasic. Переменные среды называются **MBINCLUDE** и **MBMODULE** для Include-файлов и файлов модулей соответственно, а их значения перечисляются в списке, разделённом точками с запятой, и представляют собой адреса папок, в которых будет выполняться поиск. MapBasic будет выполнять поиск в этих папках и во всех подпапках, хранящихся в них.

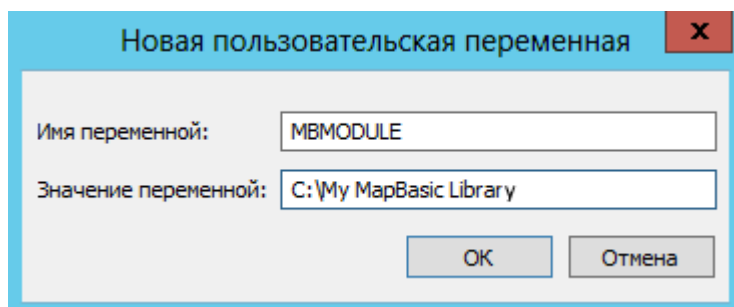
Задать переменные среды можно в диалоге, вызываемом командой **Дополнительные параметры системы**, операционной системы. Чтобы переменные среды вступили в силу, они должны быть заданы до запуска MapBasic.

Для того чтобы задать дополнительные папки для поиска Include-файлов, задайте переменную среду **MBINCLUDE**. Например, если имеются библиотеки DEF-файлов в папках или во вложенных в них папках: C:\My MapBasic Library и C:\Work, задайте переменную среды **MBINCLUDE**, как показано ниже:



Не используйте кавычек вокруг имен имен папок, даже если в адресе присутствуют пробелы.

Для того чтобы задать дополнительные папки для поиска файлов заранее скомпилированных модулей, задайте переменную среду **MBINCLUDE**. Например, если имеются библиотеки MBO-файлов в папках “C:\My MapBasic Library” и “C:\Work”, задайте переменную среды MBMODULE, как указано на рисунке:



Далее приведён пример MapBasic-кода:

```
Include "mapbasic.def"
Include "utilities.def"
Include "myapplication.def"
```

Предположим, что переменные среды заданы, как описано выше, тогда при компиляции MapBasic будет искать DEF-файлы сначала в папке, где находится компилируемый файл, затем в папке “C:\My MapBasic Library” и ее подпапках, далее в “C:\Work” и ее подпапках и, наконец, в папке, где установлен MapBasic. Если, например, файл “utilities.def” используется в нескольких написанных вами приложениях, то его можно сохранить где-нибудь в “C:\My MapBasic Library” или “C:\Work” и MapBasic сможет найти его без указания его адреса в коде.

Аналогично файл проекта (.mbp) может выглядеть следующим образом:

```
[Link]Application=myapplication.mbx
Module=myapplication.mbo
Module=library.mbo
```

mbo”, например, используется в нескольких приложениях, создаваемых вами, то его можно сохранить где-нибудь в “C:\My MapBasic Library”, и MapBasic сможет найти его без явного указания его адреса в файле проекта.

Функции

Пользовательские функции

Определение новых, пользовательских функций:	Declare Function, Function...End Function
Выход из функции:	Exit Function

Функции преобразования

Функции преобразования строки в код:	Asc()
Преобразование кода в строку:	Chr\$()
Преобразование строки в число:	Val()
Преобразование числа в строку:	Str\$(), Format\$()
Преобразование строки или числа в дату:	NumberToDate(), StringToDate()
Преобразование в номер года из двух цифр:	Set Date Window, DateWindow()
Преобразование типа объекта:	ConvertToRegion(), ConvertToPline()
Преобразование подписей в текст:	LabelInfo()
Преобразование точечного объекта в координаты MGRS:	PointToMGRS\$()
Преобразование координат MGRS в точечный объект:	MGRSToPoint()

Преобразование точечного объекта в координаты USNG:	PointToUSNG\$(obj, datumid)
Преобразование координат USNG в точечный объект:	USNGToPoint(string)

Функции даты и времени

Получить текущую дату	CurDate()
Выделить часть даты:	Day(), Month(), Weekday(), Year()
Получить текущие время и дату:	Time()
Преобразование строки или числа в дату:	NumberToDate(), StringToDate()
Получить текущие время и дату:	CurTime(), CurDateTime()
Получить дату и время из значения типа DateTime:	GetDate(), GetTime()
Создать величины типа DateTime или Time из числа:	NumberToDateTime(), NumberToTime()
Создать величину типа DateTime из чисел типа Date и Time:	MakeDateTime()
Создать величины типа DateTime или Time из строки:	StringToDateTime(), StringToTime()
Создать строку из величин типа Date или Time:	FormatDate\$(), FormatTime\$()
Извлечь компоненты величины Time:	Hour(), Minute(), Second()
Управление форматом года из двух цифр:	Set Date Window(), DateWindow()

DateTime – специальная функция

Значение DateTime представляет собой дату события. Когда в MapBasic вводится число или строка, она автоматически преобразует их в значения типа DateTime, если они присваиваются полям типа DateTime таблицы или MapBasic-переменной типа DateTime.

Для того чтобы представить дату можно пользоваться последовательностью 17 цифр:

- 4 для года
- по 2 для месяца, дня, часа, минут и секунд
- 3 для миллисекунд

Последовательность цифр является числом, если отсутствуют кавычки, и строкой, если кавычки присутствуют. Если введено менее 17 цифр, то строка и числительное интерпретируются по-разному. К строке нули добавляются справа, а к числительному – слева, поэтому "20150909" будет рассматриваться как строка "201509090000000000", а 20150909 – как 000000000020150909.

MapBasic интерпретирует строку с датой в соответствии с форматами, которые приняты на компьютере, где выполняется программа. Международный стандарт записи даты:

ГГГГ-ММ-ДД

Где ГГГГ обозначает год, ММ – месяц от 01 (январь) до 12 (декабрь), а ДД – день месяца от 01 до 31. Символы-разделители могут быть разными, например, точка (.), дефис (-) или обратная косая черта (/).

Компьютеры в США настроены на формат дат Месяц/День/Год. В США даты записываются следующим образом:

ММ/ДД/ГГГГ

Для того чтобы преобразовать дату в формат, принятый в США, используйте **оператор Set Format**. Для того чтобы избежать проблем, связанных с настройками компьютера, вызывайте функцию **NumberToDate()**, которая не зависит от настроек.

Когда вводятся данные в поле типа ДатаВремя в окне MapInfo Pro **Список** или с помощью инструмента **Информация!**, значение обрабатывается как символьная строка. Кавычки не обязательны вокруг значения, а символы-разделители, такие как обратная косая черта (/) или двоеточие (:), допускаются. Исключением является диалог **Обновить колонку**, в котором требуется ставить кавычки вокруг строковых значений.

Функции, которые возвращают значения даты переменной DateTime:

- **CurDateTime()**
- **MakeDateTime()**
- **NumberToDateTime()**
- **StringToDateTime()**

Date

Значение даты Date – целое число, представляющее дату в формате ГГГГММДД, где ГГГГ – год, ММ – месяц в диапазоне от 01 (январь) до 12 (декабрь), ДД – день месяца в диапазоне от 01 до 31.

Функции, которые возвращают значения даты DateTime: **функция CurDate()**, **функция GetDate()** и **функция NumberToDate()**.

Time

Значение времени Time целое число из девяти цифр в форме hhmmssfff, где hh – час, mm – минуты, ss – секунды, а fff – десятичная доля миллисекунд. Например, 214237582 представляет время 21:42:37.582 (или 9:42:37.582 P.M.).

Функции, которые возвращают значения времени Time:

- **CurTime()**
- **GetTime()**
- **NumberToTime()**
- **StringToTime()**

Функция **FormatTime\$()** позволяет задавать формат времени, который будет использоваться для получения результатов. Требуется следить за тем, чтобы удовлетворялись стандарты Microsoft на настройки местного времени. Буквы m, s и t обязательно должны вводиться в нижнем регистре; буква h в нижнем регистре предназначена для 12-часового отсчета, в то время, как в верхнем регистре указывает на 24-часовой. Кроме того, допускается использование f, ff или fff для десятых, сотых долей секунды или миллисекунд.

Часы	Значение
h	Часы без нулей перед значениями часов (12-часовой-отсчет).
hh	Часы с нулями перед значениями часов (12-часовой-отсчет).
H	Часы без нулей перед значениями часов (24-часовой-отсчет).
HH	Часы с нулями перед значениями часов (24-часовой-отсчет).
Минуты	Значение
m	Минуты без нулей перед значениями минут.
mm	Минуты с нулями перед значениями минут.
Секунды	Значение
s	Секунды без нулей перед значениями секунд.
ss	Секунды с нулями перед значениями секунд.
Маркер времени	Значение

t	Строка отметки времени, состоящая из единственного символа. Примечание: В некоторых языках не применяется, например, японском (Япония). В случае использования такого формата отметка времени, заданная системными параметрами LOCALE_S1159 (AM) и LOCALE_S2359 (PM), сокращается до единственного символа. Поэтому приложение может задать неправильное форматирование, когда одна и та же строка будет использоваться и для значений AM, и для PM.
tt	Строка отметки времени из нескольких символов.

Источник: <http://msdn2.microsoft.com/en-us/library/ms776320.aspx>

Математические функции

Тригонометрические функции:	Cos() , Sin() , Tan() , Acos() , Asin() , Atn()
Географические функции:	Area() , Perimeter() , Distance() , ObjectLen() , CartesianArea() , CartesianPerimeter() , CartesianDistance() , CartesianObjectLen() , SphericalArea() , SphericalPerimeter() , SphericalDistance() , SphericalObjectLen()
Случайные числа:	Randomize , Rnd()
Функции работы со знаком:	Abs() , Sgn()
Округление:	Fix() , Int() , Round()
Другие математические функции:	Exp() , Log() , Minimum() , Maximum() , Sqr()

Строковые функции

Преобразование регистра:	UCase\$() , LCase\$() , Proper\$()

Поиск подстроки:	<code>InStr()</code>
Выделение части строки:	<code>Left\$()</code> , <code>Right\$()</code> , <code>Mid\$()</code> , <code>MidByte\$()</code>
Удаление пробелов:	<code>LTrim\$()</code> , <code>RTrim\$()</code>
Форматирование:	<code>Format\$()</code> , <code>Str\$()</code> , <code>Set Format</code> , <code>FormatNumber\$()</code> , <code>DeformatNumber\$()</code>
Определение длины строки:	<code>Len()</code>
Преобразование кодов:	<code>Chr\$()</code> , <code>Asc()</code>
Сравнение строк	<code>Like()</code> , <code>StringCompare()</code> , <code>StringCompareIntl()</code>
Повторение символов:	<code>Space\$()</code> , <code>String\$()</code>
Работа с единицами измерения:	<code>UnitAbbr\$()</code> , <code>UnitName\$()</code>
Преобразование точечного объекта в координаты MGRS:	<code>PointToMGRS\$()</code>
Преобразование координат MGRS в точечный объект:	<code>MGRSToPoint()</code>
Преобразование строки EPSG в выражение CoordSys:	<code>EPSGToCoordSysString\$()</code>
Преобразование точечного объекта в координаты USNG:	<code>PointToUSNG\$(obj, datumid)</code>
Преобразование координат USNG в точечный объект:	<code>USNGToPoint(string)</code>

Работа с таблицами

Создание и изменение таблицы

Открытие таблицы:	Open Table
Закрытие одной или более таблиц:	Close Table , Close All
Создание пустой таблицы:	Create Table
Создание таблицы из другого файла:	Register Table
Импорт/экспорт таблиц/файлов:	Import , Export
Изменение структуры таблицы:	Alter Table , Add Column , Create Index , Drop Index , Create Map , Drop Map
Создание отчета Crystal Reports:	Create Report From Table
Загрузка отчета Crystal Reports:	Open Report
Создание, изменение и удаление строк:	Insert , Update , Delete
Упаковка таблицы:	Pack Table
Управление режимами работы с таблицей:	Set Table
Сохранение изменений в таблице:	Commit Table
Восстановление после изменений:	Rollback
Переименование таблицы:	Rename Table
Удаление таблицы:	Drop Table

Запросы к таблицам

Позиционировать курсор:	Fetch , EOT()
Выбор данных, работа с выборками:	Select , SelectionInfo()
Поиск по адресу:	Find , Find Using , CommandInfo()
Поиск по координатам:	SearchPoint() , SearchRect() , SearchInfo()
Информация о таблице:	NumTables() , TableInfo()
Получить информацию о колонке:	NumCols() , ColumnInfo()
Создать таблицу запроса из окна списка:	Create Query
Информация из метаданных таблицы:	GetMetadata\$() , Metadata
Сшитые таблицы:	TableInfo() , GetSeamlessSheet()
Запрос растровой таблицы или таблицы поверхности:	RasterTableInfo() function , GridTableInfo() , GetGridCellValue() function , IsGridCellNull() function , getcurrentpathfunction() function

Работа с удаленными базами данных

Создание новой таблицы:	Server Create Table
Связь с сервером:	Server_Connect() , Server_ConnectInfo()
Начало работы с сервером:	Server Begin Transaction
Локальная область хранения:	Server Bind Column

Получить информацию о колонке:	Server_ColumnInfo() , Server_NumCols()
Передача SQL-оператора:	Server_Execute()
Позиционировать курсор:	Server Fetch , Server_EOT()
Сохранить изменения:	Server Commit
Отменить изменения:	Server Rollback
Освободить ресурсы:	Server Close
Присоединение геоинформации к удаленной таблице:	Server Create Map
Изменение стилей оформления объектов:	Server Set Map
Синхронизация связанной таблицы:	Server Refresh
Создание связанной таблицы:	Server Link Table
Разорвать связь с таблицей:	Unlink
Отключение от сервера:	Server Disconnect
Информация о драйверах:	Server_DriverInfo() , Server_NumDrivers()
Получение идентификатора ODBC-соединения:	Server_GetODBCConn()
Получение идентификатора ODBC-оператора:	Server_GetODBCStmt()
Установить стиль объекта:	Server Create Style

Работа с файлами (не таблицами)

Ввод/вывод в файлы

Открыть или создать файл:	Open File
Закрыть файл:	Close File
Удалить файл:	Kill
Переименовать файл:	Rename File
Копировать файл:	Save File
Восстановить данные из файла:	Get, Seek, Input #, Line Input
Запись в файл:	Put, Print #, Write #
Определить статус файла:	EOF(), LOF(), Seek(), FileAttr(), FileExists()
Создание таблицы из другого файла:	Register Table
Повторить после ошибки совместного доступа:	Set File Timeout

Имена файлов и каталогов

Имена системных каталогов:	ProgramDirectory\$(), HomeDirectory\$(), ApplicationDirectory\$()

Выделение части имени файла:	<code>PathToTableName\$()</code> , <code>PathToDirectory\$()</code> , <code>PathToFileName\$()</code>
Полное имя файла:	<code>TrueFileName\$()</code>
Выбор файла пользователем:	<code>FileOpenDlg()</code> , <code>FileSaveAsDlg()</code>
Временный файл:	<code>TempFileName\$()</code>
Поиск файлов:	<code>LocateFile\$()</code> , <code>GetFolderPath\$()</code>

Работа с картами и графическими объектами

Создание объектов карты

Операторы создания:	<code>Create Arc</code> , <code>Create Ellipse</code> , <code>Create Frame</code> , <code>Create Line</code> , <code>Create Object</code> , <code>Create PLine</code> , <code>Create Point</code> , <code>Create Rect</code> , <code>Create Region</code> , <code>Create RoundRect</code> , <code>Create Text</code> , <code>AutoLabel</code> , <code>Create Multipoint</code> , <code>Create Collection</code>
Функции создания:	<code>CreateCircle()</code> , <code>CreateLine()</code> , <code>CreatePoint()</code> , <code>CreateText()</code>
Сложные операции:	<code>Create Object</code> , <code>Buffer()</code> , <code>CartesianBuffer()</code> , <code>CartesianOffset()</code> , <code>CartesianOffsetXY()</code> , <code>ConvexHull()</code> , <code>Offset()</code> , <code>OffsetXY()</code> , <code>SphericalOffset()</code> , <code>SphericalOffsetXY()</code>
Хранение объектов в таблице:	<code>Insert</code> , <code>Update</code>
Создание областей:	<code>Objects Enclose</code>

Изменение объектов на карте

Изменение атрибутов объекта:	Alter Object
Изменение типа объекта:	ConvertToRegion(), ConvertToPLine()
Сдвиг объектов:	Objects Offset, Objects Move
Изменяемый объект:	Выбрать изменяемый объект
Удаление частей объекта:	CreateCutter, Objects Erase, Erase(), Objects Intersect, Overlap()
Слияние объектов:	Objects Combine, Combine(), Create Object
Поворот объектов:	Rotate(), RotateAtPoint()
Разрезание объектов:	Objects Pline, Objects Split
Добавить узлы на пересечениях:	Objects Overlay, OverlayNodes()
Управление разрешением объекта:	Set Resolution
Помещение объектов в таблицу:	Insert, Update
Проверка объектов:	Objects Check
Работа с объектами:	Objects Disaggregate, Objects Snap, Objects Clean

Информация об объектах карты

Чтение числовых параметров:	Area(), Perimeter(), Distance(), ObjectLen(), Overlap(), AreaOverlap(), ProportionOverlap()

Чтение координат:	<code>ObjectGeography()</code> , <code>MBR()</code> , <code>ObjectNodeX()</code> , <code>ObjectNodeY()</code> , <code>ObjectNodeZ()</code> , <code>Centroid()</code> , <code>CentroidX()</code> , <code>CentroidY()</code> , <code>ExtractNodes()</code> , <code>IntersectNodes()</code>
Чтение параметров координат и единиц измерения расстояний, площадей и листа:	<code>SessionInfo()</code>
Единицы измерения:	<code>Set Area Units</code> , <code>Set Distance Units</code> , <code>Set Paper Units</code> , <code>UnitAbbr\$()</code> , <code>UnitName\$()</code>
Координатные системы:	<code>Set CoordSys</code>
Стили оформления объектов:	<code>ObjectInfo()</code>
Опрос подписей на слоях карты:	<code>LabelFindByID()</code> , <code>LabelFindFirst()</code> , <code>LabelFindNext()</code> , <code>Labelinfo()</code>

Работа с объектами

Использовать многопоточность, распараллеливание при работе с объектами:	<code>Create Object as Buffer</code> , <code>Objects Erase</code> , <code>Objects Intersect</code> , <code>Objects Overlay</code>

Работа со стилями объектов

Текущие стили объектов:	<code>CurrentPen()</code> , <code>CurrentBorderPen()</code> , <code>CurrentBrush()</code> , <code>CurrentFont()</code> , <code>CurrentLinePen()</code> , <code>CurrentSymbol()</code> , <code>Set Style</code> , <code>TextSize()</code>
Атрибуты стилей:	функция <code>LayerStyleInfo()</code> , <code>StyleAttr()</code>
Создание стилей:	<code>MakePen()</code> , <code>MakeBrush()</code> , <code>MakeFont()</code> , <code>MakeSymbol()</code> , <code>MakeCustomSymbol()</code> , <code>MakeFontSymbol()</code> , <code>Set Style</code> , <code>RGB()</code>

Выбор объектов по стилям оформления:	<code>ObjectInfo()</code>
Изменение стиля объекта:	<code>Alter Object</code>
Восстановление стиля символов:	<code>Reload Symbols</code>
Выражения для стилей:	<code>оператор Pen</code> , <code>оператор Brush</code> , <code>оператор Symbol</code> , <code>оператор Font</code>

Работа с окнами

Определить количество окон:	<code>NumWindows()</code> , <code>NumAllWindows()</code>
Переместить окно наверх:	<code>FrontWindow()</code>
Закрыть или спрятать окно:	<code>Close Window</code>
Закрыть все:	<code>Close All</code>
Процедуры:	<code>WinClosedHandler</code> , <code>WinFocusChangedHandler</code>

Работа с окнами карт

Открытие окна карты:	<code>Map</code>
Создание/изменение 3D-карт:	<code>Create Map3D</code> , <code>Set Map3D</code> , <code>Map3DInfo()</code> , <code>Create PrismMap</code> , <code>Set PrismMap</code> , <code>PrismMapInfo()</code>
Добавление слоя на карту:	<code>Add Map</code>
Удаление слоя с карты:	<code>Remove Map</code>

Подписывание объектов слоя:	AutoLabel
Показать все выбранные объекты слоя:	Изменение изображения в окне карты
Информация о карте и слоях:	MapperInfo() , LabelOverrideInfo() , LayerInfo() , StyleOverrideInfo()
Управление настройкой карты:	Set Map
Создание и изменение тематического слоя:	Shade , Set Shade , Create Ranges , Create Styles , Create Grid , Relief Shade
Опрос подписей на слоях карты:	LabelFindByID() , LabelFindFirst() , LabelFindNext() , LabelInfo() , LabelOverrideInfo()

Работа с классическим окном отчёта

В настоящее время окно **Отчёт**, которое было добавлено в MapInfo Pro версии 12.5 (2014), чтобы заменить старое окно **Отчёт**, не содержит многих характерных свойств. Карты, созданные в MapInfo Pro более ранних версий чем 12.5, можно просматривать и править в современном окне **Отчёт**. Однако, окно Отчёт может не содержать всех элементов исходного классического отчёта. Мы знаем, как много времени занимает создание безупречных карт, подготовленных для печати и распространения, поэтому мы сохранили **классическое** окно **Отчёт**, чтобы использовать его для карт, созданных в более ранних версиях чем 12.5.

Дополнительную информацию о классических окнах **Отчёт** смотрите в разделе *Об окнах отчёта* в *Руководстве пользователя MapBasic*.

Создать окно отчёта:	Layout
Создать рамку в отчёте:	Create Frame
Изменить отчёт:	Set Layout
Запрос к окну отчёта:	WindowInfo
Печать отчёта:	PrintWin

Установить фокус на окне отчёта или вывести его вперед:

Set Window, Run Menu Command

Работа с окном конструктора легенды

Создать легенду карты:

Create Designer Legend

Обновить окно и изменить его ориентацию:

Set Designer Legend

Создать, изменить или удалить раздел в легенде:

Add Designer Frame, Alter Designer Frame, Remove Designer Frame

Работа с окнами картографических легенд

Вместо окна картографической легенды можно пользоваться окном **Легенда**, которое появилось в версии 11.5. Окно картографической легенды работает в программах и прикладных системах, созданных в версия MapInfo Pro, предшествующих 11.5. В новых проектах мы рекомендуем вам пользоваться специальным окном **Легенда**, что облегчит совместимость легенд с будущими версиями MapInfo Pro.

Создать картографическую и тематическую легенды:

Create Cartographic Legend, Create Legend

Обновить и задать свойства окна:

Set Cartographic Legend

Создать, изменить или удалить раздел в легенде:

Add Cartographic Frame, Alter Cartographic Frame, Remove Cartographic Frame

Работа с окнами отчётов

Позволяет создавать окно Отчёт :	Layout
Создать рамку в отчёте:	Create Frame, Alter Designer Frame
Добавить в отчёт карту, таблицу (список), легенду карты, изображение или текст:	Map, Browse, Create Designer Legend, Add Image Frame, Create Text, Alter Designer Text
Добавить в отчёт геометрическую фигуру:	Create Line, Create Ellipse, Create Rect, Create RoundRect
Изменить отчёт или переместить рамку вперёд:	Set Layout, Run Menu Command
Запросить в отчёте рамку с картой или списком:	WindowInfo
Получить информацию об отчёте или рамке отчёта:	LayoutInfo(), LayoutItemInfo()
Печать отчёта:	PrintWin
Установить фокус на окне Отчёт или вывести его вперёд:	Set Window, Run Menu Command

Создание интерфейса пользователя

Для 64-разрядной версии MapInfo Pro, поскольку в ней используется новый ленточный интерфейс вместо меню, команды MapBasic, работающие с панелями инструментов и меню, создают новую вкладку **ИНСТРУМЕНТЫ** на ленте MapInfo Pro и все изменения, сделанные при помощи этих команд, отображаются на этой вкладке.

Панели инструментов

В 64-разрядной версии MapInfo Pro поддерживаются все перечисленные MapBasic-команды, работающие с панелями инструментов – ButtonPads, кроме **ButtonPadInfo()**.

Создание новой панели:	Create ButtonPad
Изменение панели:	Alter ButtonPad
Изменение кнопки:	Alter Button
Состояние панели:	ButtonPadInfo()
Определение нажатой кнопки:	CommandInfo()
Восстановление стандартных панелей:	Create ButtonPad As Default, Create ButtonPads As Default

Меню

В 64-разрядной версии MapInfo Pro команды **MenuItemInfoByHandler()** и **MenuItemInfoByID()**, перечисленные ниже, используются только для элементов управления инструментов-дополнений, а не стандартных элементов управления MapInfo Pro.

Создание нового меню:	Create Menu
Переопределение строки меню:	Create Menu Bar
Изменение меню:	Alter Menu, Alter Menu Item
Изменение строки меню:	Alter Menu Bar, Menu Bar
Вызов команды из меню:	Run Menu Command
Состояние элемента меню:	MenuItemInfoByHandler(), MenuItemInfoByID()

Диалоги

Стандартные диалоги:	Ask() , Note , ProgressBar , FileOpenDlg() , FileSaveAsDlg() , GetSeamlessSheet()
Показать новый диалог:	Dialog
Обработка событий в диалоге:	Alter Control , TriggerControl() , ReadControlValue() , Dialog Preserve , Dialog Remove
Определение нажатия ОК:	CommandInfo(CMD_INFO_DLG_OK)
Подавление вывода индикатора выполнения:	Set ProgressBars
Изменение диалога MapInfo Pro:	Alter MapInfoDialog

Окна

Управление показом окон:	Open Window , Close Window , Set Window
Открытие новых окон:	Map , Browse , Graph , Layout , Create Redistricter , Create Legend , Create Cartographic Legend , LegendFrameInfo
Определение идентификатора окна:	FrontWindow() , WindowID()
Изменение открытых окон:	Set Map , Shade , Add Map , Remove Map , Set Browse , Set Graph , Set Layout , Create Frame , Set Legend , Set Cartographic Legend , Set Redistricter , StatusBar , Alter Cartographic Frame , Add Cartographic Frame , Remove Cartographic Frame
Чтение информации об окне:	WindowInfo() , MapperInfo() , LayerInfo()
Печать окна:	PrintWin

Управление обновлением окна:	Set Event Processing , Update Window , Control DocumentWindow clause
Счётчик количества окон:	NumWindows() , NumAllWindows()

Обработчики системных событий

Обработка изменения выбора в таблице:	SelChangedHandler
Обработка закрытия окна:	WinClosedHandler
Обработка изменения изображения в окне карты:	WinChangedHandler
Обработка изменения фокуса окна:	WinFocusChangedHandler
Реакция на получение DDE-сообщения из другой программы:	RemoteMsgHandler , RemoteQueryHandler()
Управление OLE Automation:	RemoteMapGenHandler
Проверка использования программ MapBasic:	ToolHandler
Обработка завершения выполнения программы:	EndHandler
Обработка потери/приобретения фокуса окна MapInfo Pro:	ForegroundTaskSwitchHandler
Отмена обработки событий:	Set Handler

Обмен данными с другими программами

DDE (динамический обмен данными; только в Windows)

Открытие канала DDE-связи:	<code>DDEInitiate()</code>
Посылка команды по каналу DDE-связи:	<code>DDEExecute</code>
Передача значений по каналу DDE-связи:	<code>DDEPoke</code>
Получение значений по каналу DDE-связи:	<code>DDERequest\$()</code>
Закрытие DDE-связи:	<code>DDETerminate</code> , <code>DDETerminateAll</code>
Обработка приема выполняемого сообщения:	<code>RemoteMsgHandler</code> , <code>RemoteQueryHandler()</code> , <code>CommandInfo(CMD_INFO_MSG)</code>

Интегрированная картография

Вызов окна MapInfo Pro:	<code>Set Application Window</code>
Выбор порождающего окна карты:	<code>Set Next Document</code>
Создание окна легенды:	<code>Create Legend</code>
Создать окно отчёта:	<code>Layout</code>

Специальные операторы и функции

Задание имени и аргументов метода/функции в файле сборки .Net	Declare Method()
Запуск другой программы:	Run Program
Информация о системе:	SystemInfo()
Выполнение строки интерпретатором:	Run Command
Сохранение файла рабочего набора:	Save Workspace
Загрузка файла рабочего набора или .MBX-файла:	Run Application
Настройка дигитайзера:	Set Digitizer
Передача звукового сигнала:	Beep
Настройка данных, считываемых CommandInfo:	Set Command Info
Настройка задержки при перетаскивании объекта:	Set Drag Threshold
Переключиться между 1-разрядным и 32-разрядным цветом пикселей курсора:	Set Cursor

Техническая поддержка

Корпорация Precisely обеспечивает начальный бесплатный период технической поддержки при покупке нового программного продукта или обновлённой версии, таким образом, у вас есть все возможности для успешного начального освоения продукта. После завершения срока бесплатного использования, Precisely предлагает вам выбрать один из вариантов расширенной

технической поддержки: для частного использования, для бизнеса, для корпоративного использования.

Служба технической поддержки всегда готова помочь вам. В этом разделе описывается информация, которую вам нужно представить при звонке в службу технической поддержки. Здесь также объясняются некоторые технические процедуры, которые помогут вам в разрешении проблем.

Пожалуйста, не забывайте указывать серийный номер, партнёрский номер или номер контракта, когда связываетесь со службой технической поддержки.

Как связаться со службой технической поддержки

Для получения технической поддержки необходимо зарегистрировать продукт. Это можно легко сделать во время установки или в любое другое время, связавшись со службой поддержки в рабочее время.

Полная техническая поддержка представляется для текущей версии MapBasic и двух предыдущих.

О службе технической поддержки

Получить техническую консультацию можно в службе поддержки в вашем регионе. Связаться с ближайшим офисом вы можете, выбрав раздел **Contact Support** на нашем сайте:

<http://support.precisely.com>

Он-лайн служба технической поддержки

Он-лайн служба технической поддержки — это еще один способ сообщить о возникшей проблеме и решить её с помощью центра технической поддержки. Если у вас еще нет ID пользователя, то вы должны зарегистрироваться при первом посещении этого сайта.

<http://support.precisely.com>

Прежде чем позвонить

Пожалуйста, соберите и держите под рукой эту информацию, когда связываетесь со службой технической поддержки.

1. Серийный номер. Для получения технической поддержки вы должны иметь зарегистрированный серийный номер.
2. Наименование вашей организации и имя пользователя. Человек, обращающийся в службу поддержки должен являться контактным лицом, указанным в соглашении о поддержке.
3. Номер версии продукта.

4. Наименование и версия операционной системы.
5. Краткое описание сути проблемы. Здесь может быть полезна следующая информация:
 - Сообщения об ошибках
 - Ситуация, при которой возникают проблемы
 - Постоянство — возникает ли проблема эпизодически или постоянно?

Предполагаемое время ответа

Большая часть вопросов может быть решена во время телефонного разговора. Если решение не будет найдено сразу, служба технической поддержки пришлёт ответ до наступления конца рабочего дня. Работник службы технической поддержки будет сообщать вам, о том как решается проблема.

Все обращения в службу технической поддержки по электронной почте или через оперативную службу обрабатываются также, как и обращения по телефону; однако, случаются задержки до нескольких часов, которых невозможно избежать, они вызваны скоростью передачи и обработки.

Ошибки в программном обеспечении

Если вы обнаружите ошибку в программном обеспечении, свяжитесь с представителями Precisely, ошибка будет занесена в базу данных, ей будет присвоен номер и вы сможете контролировать ситуацию с исправлением ошибки. В новых версиях программы большинство выявленных ошибок текущей версии будет исправлено.

Другие информационные ресурсы

Архив MapInfo-L

Корпорация Precisely совместно с Билом Тоеном (Bill Thoen), предоставляет Web-архив почтовой переписки пользователей MapInfo-L. В системе имеется поиск, по теме и дате публикации.

Корпорация предоставляет базу данных для сообщества пользователей, но администрирование рассылочного почтового сервера MapInfo-L осуществляется Билом Тоеном (Bill Thoen). Подробная информация о MapInfo-L размещена на веб-странице MapInfo-L по адресу: <http://groups.google.com/group/mapinfo-l?hl=en>.

Авторские права

Информация, содержащаяся в этом документе, может быть изменена без уведомления и не представляет собой обязательств со стороны продавца или его представителей. Никакая часть этого документа не может быть воспроизведена или передана в какой-либо форме любыми средствами, механическими или электронными, включая фотокопирование, без письменного разрешения Precisely, 2 Blue Hill Plaza, #1563, Pearl River, NY 10965.

© 1985, 2020 Precisely. Все права защищены.

Контактная информация и информация о поддержке находится по адресу:

<http://support.precisely.com>.

© 2020 участники лицензии OpenStreetMap, CC-BY-SA; смотрите сайт OpenStreetMap

<http://www.openstreetmap.org> (лицензия доступна по адресу:

<http://www.opendatacommons.org/licenses/odbl/index.html>) или CC-BY-SA

<http://creativecommons.org/licenses/by-sa/2.0>.

libtiff © 1988-2020 Сэм Леффтер, © 2020 Silicon Graphics International, ранее Silicon Graphics Inc. Все права защищены.

libgeotiff © 2020 Нильс Д. Риттер (Niles D. Ritter).

Amigo, Portions © 2020 Three D Graphics, Inc. Все права защищены.

Halo Image Library © 2020 Media Cybernetics Inc. Все права защищены.

Часть авторских прав LEAD Technologies, Inc. © 1991–2020. Все права защищены..

Часть авторских прав © 1993-2020 Кен Мартин (Ken Martin), Вилл Шрёдер (Will Schroeder), Билл Лоренсен (Bill Lorensen). Все права защищены.

ECW выпущен ERDAS © 1993-2020 Intergraph Corporation, подразделением Hexagon Group. и/или поставщиков этой компании. Все права защищены.

Часть прав принадлежит © 2020 Intergraph Corporation, подразделению Hexagon Geospatial AB. Все права защищены.

MrSID, MrSID Decompressor и логотип MrSID являются торговыми марками, принадлежащими LizardTech, подразделению компании Celartem. Используется по лицензии. Часть авторских прав на эту программу принадлежит © 1995-2020 LizardTech компании Celartem и/или университету штата Калифорния или охраняется патентами США: 5,710,835 и используется по лицензии. Все права защищены. MrSID защищён патентами и авторскими правами США и международными патентами и авторскими правами, заявки на иностранные патенты поданы. Воспроизведение или несанкционированное использование запрещено.

Содержит FME® Objects © 2005--2020 Safe Software Inc. Все права защищены.

Amyuni PDF Converter © 2000-2020 AMYUNI Consultants — AMYUNI Technologies Inc. Все права защищены.

Civic England – Public Sector Symbols (условные знаки) охраняются авторским правом © 2020 West London Alliance. Эти условные знаки можно использовать бесплатно. Для того чтобы получить подробную информацию об этих условных знаках, включая сведения о том, как получить их для использования в других приложениях, посетите веб-сайт альянса западных пригородов Лондона по адресу: <http://www.wla.london>

© 2006-2020 TomTom International BV. Все права защищены. Эти материалы являются собственностью и охраняются законами об авторском праве и другими законами об охране интеллектуальной собственности, которыми владеет компания TomTom, или на которые имеются права, принадлежащие компании TomTom. Использование этих материалов ограничено условиями лицензионного соглашения. Вы несёте ответственность за несанкционированное копирование или незаконное распространение этих материалов.

Microsoft Bing: Права на все материалы службы Bing © 2020 принадлежат Microsoft Corporation и/или другим авторам, One Microsoft Way, Redmond, WA 98052, USA. Все права защищены. Название, авторские права и прочая интеллектуальная собственность службы Bing принадлежит Microsoft или партнёрам. Microsoft, Windows, Windows Live, логотип Windows logo, MSN, логотип MSN (бабочка), Bing и другие продукты и службы Microsoft могут быть либо зарегистрированными торговыми знаками, либо торговыми знаками Microsoft в США и/или других странах.

В этом продукте содержится 7-Zip, который используется на условиях лицензии GNU Lesser General Public (стандартной общественной лицензии ограниченного применения GNU) версии 3 29 июня 2007 с дополнительными ограничениями unRAR. Текст лицензии можно получить по адресу: <http://www.7-zip.org/license.txt>. Текст лицензии GNU можно получить по адресу: <http://www.gnu.org/licenses/lgpl.html>. Исходный код доступен по адресу: <http://www.7-zip.org>.

Программы и другие продукты, упомянутые здесь, могут являться торговыми марками компаний их производящих, что признаётся этим заявлением. Названия торговых марок используются в тексте только на пользу владельца торговой марки, без намерения нарушить права торговой марки.

Об открытом программном обеспечении

QT Assistant 5.2.0

В этом продукте используется QT Assistant версии 5.2.0, который используется на условиях лицензии Lesser General Public License (GNU), версии 2.1, февраль 1999. Текст лицензии можно получить по адресу: <http://www.gnu.org/licenses/lgpl-2.1.txt>. Исходные коды этого программного обеспечения доступны по адресу: <http://qt-project.org/downloads>.

2 - Новые и улучшенные операторы и функции MapBasic

В этом разделе:

В этом разделе

Новые функции и операторы MapBasic.....	38
Добавления к существующим функциям и операторам.....	38



Новые функции и операторы MapBasic

Следующие функции добавлены в этой версии:

Имя функции/оператора	Описание
Функция MenuItemEnabled()	Эта функция возвращает идентификаторы включённых команд меню.
Функция Atan()	Эта функция возвращает значение арктангенса числа.
Функция Atan2()	Эта функция возвращает арктангенс y/x в диапазоне $-\pi$ до π радиан.

Добавления к существующим функциям и операторам

Общие функции и операторы

Имя функции/оператора	Описание
Оператор Select	Добавлен новый пункт Limit. Добавлена поддержка пространственных агрегаторов и перекрестных объединений.
Оператор Update	Добавлена поддержка выражений в предложении Where.
Оператор Delete	Добавлена поддержка выражений в предложении Where.

Имя функции/оператора	Описание
Функция WindowInfo()	Расширенные параметры рендеринга с визуальными эффектами, с помощью которых можно управлять сглаживанием стилей оформления, теперь также используются для окон конструктора отчёта. Добавлено два новых типа окон: • WIN_NOTIFICATIONS • WIN_QUERY_BUILDER
Функция GetPreference()	Содержит пятнадцать дополнительных разделов настроек и более 230-ти уникальных определителей настроек, которые можно раскрывать в запросе о текущих параметрах используемых настроек.
Оператор Run Menu Command	Добавлена новая команда.
Функция SystemInfo()	Появились новые параметры attribute: • SYS_INFO_LICENSE_MODE • SYS_INFO_SUBSCRIPTION_PLAN
Функция MapperInfo()	Появились новые параметры attribute: • MAPPER_INFO_CARTO_SCALE • MAPPER_INFO_CARTO_SCALE_EXPR
Оператор Set Window	Расширенные параметры рендеринга с визуальными эффектами, с помощью которых можно управлять сглаживанием стилей оформления, теперь также используются для окон конструктора отчёта. Добавлено два новых типа окон: • WIN_NOTIFICATIONS • WIN_QUERY_BUILDER
Функция TableInfo()	Добавлен новый атрибут: • TAB_INFO_DOMINANT_OBJECT_TYPE

Имя функции/оператора	Описание
Функция WindowID()	Расширенные параметры рендеринга с визуальными эффектами, с помощью которых можно управлять сглаживанием стилей оформления, теперь также используются для окон конструктора отчёта. Добавлено два новых типа окон: • WIN_NOTIFICATIONS • WIN_QUERY_BUILDER
Функция SelectionInfo()	Добавлен новый атрибут: • SEL_INFO_ISNEW

3 - Алфавитный справочник языка MapBasic

В этом разделе детально описываются операторы и функции языка MapBasic. Операторы и функции перечислены по алфавиту.

В этом разделе

Принятые обозначения функций и операторов.....	42
Описание функций и операторов.....	42



Принятые обозначения функций и операторов

Каждая функция и оператор описаны в следующем формате:

Назначение

Краткое описание функции или оператора.

Предупреждение!

Информация об ограничениях (например, "Функция DDEInitiate доступна только в среде Microsoft Windows", "Оператор **For...Next** нельзя выполнить в окне **MapBasic**").

Синтаксис

Формат, в котором применяется функция или оператор и объяснение аргументов.

Возвращаемая величина

Тип возвращаемого функцией значения.

Описание

Подробное описание функции или оператора и прочая нужная информация.

Пример:

Короткий пример.

Описание заканчивается списком ссылок на соответствующую функцию или оператор.

Большинство операторов MapBasic можно запускать прямо в среде MapInfo Pro, вводя их в окне **MapBasic**. Если оператор нельзя запустить в окне **MapBasic**, об этом будет сказано в разделе "Предупреждение". Как правило, через окно **MapBasic** нельзя вводить операторы управления программой, такие как циклы и переходы.

Описание функций и операторов

В следующем разделе дано описание функций операторов и предложений. Некоторые разделы содержат подробную информацию о применении функции или оператора.

Функция Abs

Назначение

Возвращает абсолютное значение числа (модуль). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Abs ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Abs()** возвращает абсолютное значение числа, полученного в результате вычисления выражения *num_expr*.

Если результат вычисления *num_expr* больше или равен нулю, **Abs()** возвращает результат *num_expr* без изменений. Если результат вычисления *num_expr* меньше нуля, **Abs()** возвращает результат *num_expr*, умноженный на минус единицу (-1).

Пример

```
Dim f_x, f_y As Float  
f_x = -2.5  
f_y = Abs(f_x)  
  
' f_y now equals 2.5
```

См. также:

[функция Sgn\(\)](#)

Функция Acos()

Назначение

Возвращает арккосинус числа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Acos ( num_expr )
```

num_expr – численное выражение, результат которого должен находиться в диапазоне от единицы до минус единицы включительно.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Acos()** вычисляет арккосинус числа, полученного в результате вычисления выражения *num_expr*. Другими словами, **Acos()** возвращает величину угла (в радианах), косинус которого равен *num_expr*.

Результат, возвращаемый **Acos()**, представляет собой угол в радианах. Диапазон значения угла – между 0 и Pi радиан (число Pi равно приблизительно 3.141593, и Pi/2 радиан равно 90 градусам).

Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG. Ваша программа должна включать файл MAPBASIC.DEF для того, чтобы ссылаться на DEG_2_RAD или RAD_2_DEG.

Так как значения косинуса могут находиться в диапазоне от минус единицы до единицы, то и результат вычисления выражения *num_expr* должен быть не менее минус единицы и не более единицы.

Пример

```
Include "MAPBASIC.DEF"
Dim x, y As Float
x = 0.5
y = Acos(x) * RAD_2_DEG
```

```
' y will now be equal to 60,
' since the cosine of 60 degrees is 0.5
```

См. также:

[функция Asin\(\)](#), [функция Atn\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Функция ActiveWindow()

Назначение

Возвращает текущее активное окно. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

в 64-разрядной версии MapInfo Pro возвращает текущее активное окно менеджера привлечения.

Синтаксис

```
ActiveWindow ( )
```

Возвращаемая величина

Целое число типа Integer.

Оператор Add Cartographic Frame

Назначение

Оператор **Add Cartographic Frame** изменяет положение, заголовок, подзаголовок, рамку и стиль существующего раздела легенды, который создает [оператор Create Cartographic Legend](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Cartographic Frame
[ Window legend_window_id ]
[ Custom ]
[ Default Frame Title { def_frame_title } [ Font... ] ]
```

```
[ Default Frame Subtitle { def_frame_subtitle } [ Font... ] ]
[ Default Frame Style { def_frame_style } [ Font... ] ]
[ Default Frame Border Pen... pen_expr ]
Frame From Layer { map_layer_id | map_layer_name }
[ Position ( x , y ) [ Units paper_units ] ]
[ Using
[ Column { column | object [ FromMapCatalog { On | Off }]] ]
[ Label { expression | default } ]
[ Title [ frame_title ] [ Font... ] ]
[ SubTitle [ frame_subtitle ] [ Font... ] ]
[ Border Pen... ]
[ Style [Font...] [ NoRefresh ]
[ Text { style_name } { Line Pen...
| Region Pen... Brush...
| Symbol Symbol... } ]
[ , ... ]
]
[ , ... ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#).

def_frame_title – это строковая величина, которая по умолчанию определяет заголовок раздела легенды. Эта величина может включать специальный символ "#", который будет замещаться именем текущего слоя.

def_frame_subtitle – это строковая величина, которая по умолчанию определяет заголовок подраздела легенды. Эта величина может включать специальный символ "#", который будет замещаться именем текущего слоя.

def_frame_style – то строковая величина, которая определяет каждый символ в каждом разделе легенды. Символ “#” будет замещаться именем текущего слоя. Символ % будет замещаться словами “Line”, “Point”, “Region”, которые будут соответствовать типу географических объектов (эти названия могут быть другими в русской версии). Например, "% of #" будет соответствовать тексту “Region of States” для слоя STATES.TAB.

pen_expr – это выражение, возвращающее объект типа Pen, например, *MakePen(width, pattern, color)*. Если по умолчанию линия рамки определена, то она останется по умолчанию такой же для раздела легенды. Если предложение Pen для рамки задано для раздела легенды, то оно будет определять тип линии для рамки вместо заданного по умолчанию.

map_layer_id или *map_layer_name* определяют слой карты; задаются либо типом SmallInt (коротким целым), например, используйте 1 для определения самого верхнего слоя, не считая косметического; либо это строковая переменная, соответствующая имени таблицы, отображенной на карте. Для тематического слоя необходимо определять параметр *map_layer_id*.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цистеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки

- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

frame_title – строковая переменная, определяющая заголовок раздела легенды. Если будет определено предложение заголовка раздела **Title**, то будет использоваться эта величина вместо величины *def_frame_title*.

frame_subtitle – строковая переменная, определяющая подзаголовок раздела легенды. Если определено предложение **SubTitle**, то оно будет использоваться вместо величины *def_frame_subtitle*.

column – атрибутивное имя колонки из раздела слоя таблицы или колонка "object" (означает, что стили легенды базируются на уникальном стиле в файле карты). По умолчанию это "object".

style_name – строковая переменная, которая указывает к какому типу: символу, линии или области относится объект в разделе легенды.

Описание

Если ключевое слово **Custom** включено в описание, то каждый раздел легенды должен включать в себя и предложение **Position**. Если слово **Custom** пропущено и легенда отображается в виде книжной или альбомной ориентации, то разделы легенды будут добавлены в самый конец.

Предложение **Position** контролирует положение раздела в окне легенды. Верхний левый угол окна легенды имеет позицию 0, 0. **Position** использует "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры), см. [оператор Set Paper Units](#). MapBasic поддерживает свои единицы измерения, по умолчанию это дюймы; программа MapBasic может поменять эти единицы, используя [оператор Set Paper Units](#). Можно изменить эти настройки единиц измерения, включив предложение **Units** в предложение **Position**.

По умолчанию в этом операторе настройки применяются только к разделам, созданным в этом операторе. Они не действуют на существующие разделы. Стандартные режимы разделов, которые использует [оператор Create Cartographic Legend](#), не действуют на разделы, созданные этим оператором ранее.

Когда сохраняете рабочий набор, то в рабочий набор записывается новое предложение **FromMapCatalog OFF**. Номер версии рабочего набора увеличивается до 800. Если определено предложение **FromMapCatalog ON**, мы не записываем его в рабочий набор. Это стандартный режим. Номер версии рабочего набора в этом случае не увеличивается.

FromMapCatalog ON отыскивает стили из MapCatalog для таблицы прямого доступа. Если таблица не является таблицей прямого доступа, MapBasic возвращается к заданному по умолчанию поведению для таблицы с непрямым доступом вместо того, чтобы допустить ошибку. Стандартное поведение для таблицы с отключённым доступом это **FromMapCatalog Off** (то есть, используются стили карты).

FromMapCatalog OFF отыскивает уникальные стили Карты для таблицы прямого доступа сервера. Эта таблица должна быть таблицей прямого доступа, которая поддерживает стилизацию каждой записи. Если таблица прямого доступа не поддерживает стили записей, то применяется заданное по умолчанию поведение для таблиц прямого доступа – используются заданные по умолчанию стили из MapCatalog (**FromMapCatalog ON**).

Label – это либо выражение для подписи, либо слово "default" (последнее означает, что при создании текста используется стиль оформления рамки по умолчанию, если только предложение Style не содержит текст). Стандартным значением является default.

Предложение **Style** и ключевое слово **NoRefresh** позволяют создавать собственные разделы легенды, которые не будут изменяться при обновлении легенды. Если ключевое слово **NoRefresh** используется в предложении **Style**, то стили не извлекаются из таблицы. В этом случае предложение **Style** должно содержать ваш собственный список стилей, используемых в разделе легенды. Это достигается явным заданием значения **Text** и предложений **Line**, **Region** или **Symbol**.

См. также:

[оператор Create Cartographic Legend](#), [оператор Set Cartographic Legend](#), [оператор Alter Cartographic Frame](#), [оператор Remove Cartographic Frame](#)

Оператор Add Column

Назначение

Добавляет новую временную колонку в открытую таблицу или обновляет уже созданную колонку данными из другой таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Column table ( column [ datatype ] )
{ Values const [ , const ... ] |
  From source_table
  Set To expression
  [ Where { dest_column = source_column |
    Within | Contains | Intersects } ]
  [ Dynamic ] }
```

table – имя открытой таблицы, к которой будет добавляться колонка;

column – имя добавляемой колонки;

datatype – тип данных колонки, который может быть одним из следующих: Char(*width*); Float; Integer; SmallInt; Decimal(*width*, *decimal_places*); Date, **DateTime** или Logical; по умолчанию используется Float.

source_table – имя другой открытой таблицы, таблицы-источника;

expression – выражение, извлекающее данные из таблицы-источника *source_table* для возможного обобщения;

dest_column – имя колонки изменяемой таблицы (*table*);

source_column – имя колонки-образца из таблицы-источника *source_table*;

Dynamic – делает колонку динамически вычисляемой: если оператор имеет это ключевое слово, то последующие изменения данных в таблице-источнике (*source_table*) повлекут немедленное изменение в таблице с колонкой, созданной оператором.

Описание

Оператор **Add Column** используется для создания новой временной колонки в открытой таблице MapInfo Pro. Созданная колонка не становится постоянной даже после сохранения таблицы на диске. Однако если временная колонка создана на базе данных из постоянных колонок другой таблицы, и вы сохранили рабочий набор, то последний будет включать информацию о временной колонке, которая позволит вам вызвать вновь таблицу с этой колонкой, загрузив рабочий набор. Для создания постоянной колонки в таблице используйте **оператор Alter Table** и **оператор Update**.

См. также:

оператор Alter Table, **оператор Update**

Явное задание данных для новой колонки

Для этого используется предложение **Values**. После него задайте список постоянных значений, разделенных запятыми.

В следующем примере временная колонка таблицы WARDS заполняется числами, заданными в предложении **Value**.

```
Open Table "wards"
Add Column wards(percent_dem)
Values 31,17,22,24,47,41,66,35,32,88
```

Заполнение новой колонки данными из другой таблицы

Если используется предложение **From** вместо **Values**, которое задает имя другой таблицы (*source_table*), данные из которой будут использоваться в новой колонке. Обе таблицы должны быть открыты.

Если используется предложение **From**, MapInfo Pro проводит объединение двух таблиц. Условия объединения задаются в предложении **Where**. Если предложение **Where** опустить, MapInfo Pro автоматически примет решение о наиболее удобном способе объединения таблиц.

Простейшее предложение **Where** в форме *Where колонка = "имя_колонка"* задает объединение двух таблиц по совпадающим значениям в их колонках. Этот метод приемлем, если колонка Вашей таблицы имеет величины, сходные с величинами в колонке из другой таблицы (например, Вы добавляете колонку в таблицу STATES и другая таблица также имеет колонку с именами штатов).

Если обе таблицы имеют графические объекты, то предложением **Where** можно задать географическое объединение. Так, если оператор включает предложение **Where Contains**, MapInfo Pro производит объединение, проверяя, могут ли объекты из таблицы *source_table* содержать объекты из изменяемой таблицы.

Следующий оператор добавляет колонку в таблицу STORES. Новая колонка будет содержать имена округов, которые выделяются из таблицы COUNTY:

```
Add Column
stores(county char(20)   'add "county" column
From counties           'derive data from counties table...
Set to cname            'using the counties table's "cname" column
Where Contains          'join: where a county contains a store site
```

Метод создания колонки с ключом **Where Contains** возможен в том случае, если таблица имеет точечные объекты и вторая таблица имеет объекты, которые могут содержать эти точки.

В следующем фрагменте таблице STATES добавляется временная колонка. Значения для нее извлекаются из таблицы CITY_1K, содержащей данные о крупных городах США. После выполнения оператора **Add Column**, каждая строка таблицы STATES будет содержать количество крупных городов в каждом штате.

```
Open Table "states" Interactive
Open Table "city_1k" Interactive

Add Column states(num_cities)
From city_1k      'derive values from other table
Set To Count(*)   'count cities in each state
Where Within      'join: where cities fall within state
```

Предложение **Set To** задает правило обобщения данных функцией **Count(*)**. Описание функций обобщения приводится ниже.

Заполнение уже существующей колонки данными из другой таблицы

Для того, чтобы обновить содержимое уже существующей колонки вместо того, чтобы создать новую временную колонку, надо опустить параметр *datatype* и использовать предложение **From** вместо предложения **Values**. При обновлении MapBasic будет игнорировать ключевое слово **Dynamic**.

Обобщение данных в новой колонке

Если используется предложение **From**, то можно вычислить значения для новой колонки, обобщая данные из другой исходной таблицы. Для определения правила обобщения данных используется предложение **Set To**, включающее в себя функции обобщения.

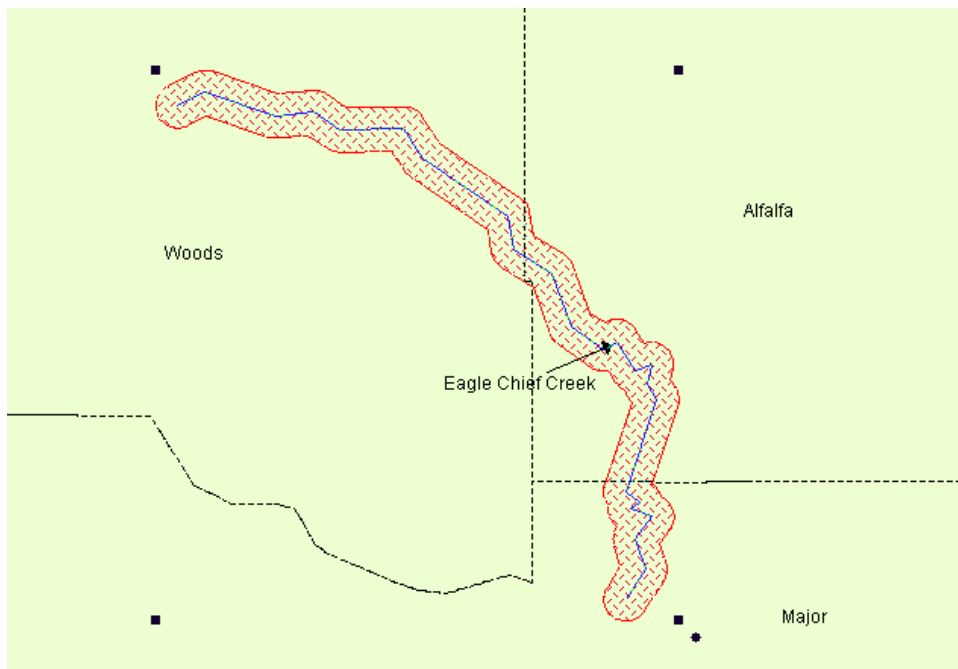
Следующая таблица приводит список функций обобщения:

Функция	Синтаксис	Что будет в новой колонке
Avg()	Avg(col)	средняя величина всех значений поля col из строк таблицы-источника, которые участвуют в объединении
Count()	Count(*)	число строк в таблице-источнике, которые используются в объединении
Max()	Max(col)	наибольшая величина в колонке col из строк таблицы-источника, которые участвуют в объединении
Min()	Min(col)	наименьшая величина в колонке col из строк таблицы-источника, которые участвуют в объединении
Sum()	Sum(col)	сумма значений из колонки по всем строкам таблицы-источника, которые используются в объединении
WtAvg()	WtAvg(col, weight_col)	взвешенная средняя величина значений поля col из строк таблицы-источника; колонки с большим значением <i>weight_col</i> вносят больший вклад в результат, чем колонки с меньшим значением <i>weight_col</i>
Proportion Avg()	Proportion Avg(col)	среднее число, вычисленное с учетом перекрытия одних областей другими
Proportion Sum()	Proportion Sum(col)	сумма, вычисленная с учетом взаимного перекрытия областей
Proportion WtAvg()	Proportion WtAvg(col, weight_col)	взвешенная средняя величина с учетом перекрытия областей.

Примечание: Функция Count возвращает целое значение. Остальные функции возвращают числа с плавающей точкой. (Ни одна из функций обобщения MapBasic, не возвращает десятичные числа. Десятичные поля являются только формой представления данных. Все вычисления производятся с числами с плавающей запятой.)

В основном функции обобщения работают только с неграфическими данными таблицы. Но последние три функции (Proportion Sum, Proportion Avg, Proportion WtAvg) выполняют вычисления с учетом взаимного расположения объектов. В следующем примере показан смысл этого предложения.

Допустим, имеется таблица COUNTIES, содержащую границы административных районов и демографическую информацию (например, численность населения) каждого района области. Существует еще таблица RISK, которая содержит области. Объекты таблицы RISK представляют зоны различной степени риска, например, риска возможного затопления при разливе близлежащей реки.



1 Границы районов. 2 Буферная зона риска.

Пользуясь данными этих двух таблиц, Вы хотите рассчитать численность населения, живущего в зонах риска. Если половина района покрыта зоной риска, то будем считать, что половина населения попадает в зону риска; если треть района перекрыта зоной риска, то мы полагаем, что при наводнении опасность угрожает трети населения и т.д.

Следующий пример показывает, как с помощью функции обобщения **Proportion Sum** вычисляется новая колонка "population_at_risk", представляющая количество населения, подвергаемое риску:

```
Add Column Risk(population_at_risk Integer)
From counties
```

```
Set To Proportion Sum(county_pop)
Where Intersects
```

Для каждого района, часть которого попадает в зону возможного затопления, MapInfo Pro вычисляет долю пострадавшего населения и все это суммирует.

Функция **Proportion Sum** вычисляет результат на основании предположения, что суммируемые числа прямо пропорциональны ассоциированным с ними площадям. Если используется **Proportion Sum** для подсчета статистики населения в регионе, половина которого попадает в другой регион, MapInfo Pro прибавит к сумме половину населения региона. На практике часто оказывается, что данные распределяются по территории неравномерно. Например, большая часть населения штата Нью-Йорк живет в городе Нью-Йорк.

Если используется функция **Proportion Sum** в случаях неравномерного распределения данных, то результаты могут быть некорректными. Чтобы добиться достоверности, разбивайте большие области на меньшие, в которых данные распределены равномерно.

Функция **Proportion Avg** вычисляет средний результат, учитывая взаимное перекрытие объектов. Продолжая пример с риском наводнения, представим, что в таблице COUNTIES содержится колонка (median_age) "средний_возраст", которая содержит значения среднего возраста жителей каждого города.

Следующий оператор вычисляет средний возраст живущих в зоне риска:

```
Add Column Risk(age Float)
From Counties
Set To Proportion Avg(median_age)
Where Intersects
```

Для каждой записи в таблице COUNTIES MapInfo Pro вычисляет отношение зоны риска к площади района. В результате получается число от нуля до единицы. MapInfo Pro умножает это число на средний возраст жителей города (средний_возраст) и затем суммирует результаты. Так, если средний возраст равен 50 и в зоне риска находится 10% площади района, то MapInfo Pro добавит в общую сумму число 5, то есть 10% от числа 50.

Следующий пример подробно показывает, как функции **Proportion Sum** и **Proportion Avg** рассчитывают промежуточные суммы и результаты. Например:

Если риску затопления подвержена половина территории, то к промежуточной сумме прибавляется половина значения. Если в рискованной зоне находится 10% территории, то к промежуточной сумме прибавляется 10% от значения. Функция Proportion Avg рассчитывает среднюю величину, поэтому если 4 округа попадают в зону риска, то после вычисления промежуточной суммы, мы делим ее на 4.

Если 50% округа "county1" попадает в зону риска и население этого округа равно 66 тысячам, то вклад этого округа в промежуточную сумму равен 33 тысячам.

Если 30% округа "county2" попадает в зону риска и население этого округа равно 100 тысячам, то вклад этого округа в промежуточную сумму равен 30 тысячам.

Если 20% округа "county3" попадает в зону риска и население этого округа равно 50 тысячам, то вклад этого округа в промежуточную сумму равен 10 тысячам.

Если 10% округа "county4" попадает в зону риска и население этого округа равно 60 тысячам, то вклад этого округа в промежуточную сумму равен 6 тысячам.

Тогда значение Proportion Sum равно $33+30+10+6 = 82$ тысяч,

а значение Proportion Avg равно $(33+30+10+6)/4 = 20.5$ тысяч.

Proportion WtAvg работает так же, как и **Proportion Avg**, но позволяет еще и извлекать весовые коэффициенты из отдельной колонки данных. Например:

Возьмем за базу предыдущий пример и добавим в рассмотрение колонку "RuralPercent", отражающую долю сельской местности. Этот коэффициент показывает, насколько меньше населения живет в сельской местности и, следовательно, насколько меньше народу попадает в зону риска.

Если 50% округа "county1" попадает в зону риска, и население его равно 66 тысячам, а в сельская местность (RuralPercent) занимает 0.8 от территории округа, то $(0.5 * 66 * 0.8) = 26$.

Если округа "county3", "county4" и "county5" являются на 50% сельскими, то:

для "county3" $0.3 * 100 * 0.5 = 15$

для "county4" $0.2 * 50 * 0.5 = 5$

для "county5" $0.1 * 60 * 0.5 = 3$

Тогда пропорциональное средневзвешенное будет равно: $(26 + 15 + 5 + 3)/2.3 = 21.3043$

Использование функций *Proportion...* с объектами иного типа, чем "область"

Если используется одна из трех функций **Proportion** и таблица source_column содержит объекты типа "область", MapInfo Pro вычисляет процент перекрытия одной области другой областью. Однако, если к записям таблицы-источника присоединены объекты другого графического типа, MapInfo Pro считает каждый объект лежащим либо внутри, либо снаружи обрабатываемой области (объект считается лежащим внутри области, если его центроид лежит внутри области).

Динамически вычисляемая колонка

Если при создании новой колонки Вы использовали ключевое слово **Dynamic**, то новая колонка будет динамически вычисляемой, то есть изменения данных в таблице-источнике (source_table) повлекут немедленное изменение в таблице с колонкой, созданной этим оператором.

Если создать динамически вычисляемую колонку и потом закрыть таблицу-источник, то значения в колонке будут зафиксированы (то есть колонка больше уже не будет динамически вычисляемой).

Аналогичный эффект срабатывает при географическом объединении: если была создана динамически вычисляемая колонка и закрыта какая-либо карта, используемая в географическом объединении, то значения в колонке зафиксируются.

Оператор Add Custom Frame

Назначение

Добавляет рамку в окно **Отчёт**.

Синтаксис

```
Add Custom Frame [ Window window_id ]
[ Position ( x, y ) [ Units paper_units ] ]
[ Width frame_width [ Units paper_units ] ]
[ Height frame_height [ Units paper_units ] ]
[ Pen ... ] [ Brush ... ] [ Priority n ]
[ Name frame_name ]
From { File addin_mbx_file_name.mbx }
Using serialized_custom_frame_file_name.mlcf
View xps_custom_frame_file_name.xps
```

window_id – идентификатор окна **Отчёт**, целое число;

x, *y* – задаёт в "бумажных" единицах (*paper_units*) положение левого верхнего угла рамки с изображением в окне **Отчёт**.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цигеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 цигеро, 72 пункта
- 1 пункт (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 цигеро
- 1 цигеро = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 пунктов
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 цигеро, 28.34646 пунктов
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 пунктов

frame_width и *frame_height* – задают ширину и высоту рамки в окне **Отчёт**.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

Frame_name – строка, представляющая имя рамки.

Addin_mbx_file_name.mbx – строка, представляющая имя дополнительной mbx-программы, которая создаёт контент рамки и управляет её поведением. Имя файла это либо короткое имя, либо относительный адрес, если MBX-файл хранится в папке `Tools` MapInfo Pro.

Serialized_custom_frame_file_name.mlcf – строка, представляющая имя серийной рамки. Этот файл создаётся дополнительной программой и может быть любого формата, который выбрал разработчик дополнительной программы, если этот файл может быть использован для повторного создания заказной рамки с помощью MapBasic-команды. Расширение этого файла – `.mlcf`.

Xps_custom_frame_file_name.xps – строка, представляющая имя xps-файла, который создаётся, когда заказная рамка сохраняется в рабочем наборе. Расширение этого файла – `.xps`. Этот файл требуется в случае, либо когда при повторном выполнении mapbasic-команды происходит ошибка загрузки дополнительной MBX-программы, либо если имя файла серийной рамки вызывается для отмены серии заказной рамки. В этом случае xps-файл будет использоваться в качестве альтернативы при отображении заказной рамки в состоянии, в котором она была при сохранении.

Описание

Оператор **Add Custom Frame** предназначен для совместимости wor-файлов и загрузки wor-файлов. Выполнение оператора когда-нибудь помимо загрузки рабочего набора (`.wor`), например, в окне **MapBasic** или в MapBasic-программе приведёт к появлению сообщения об ошибке, в котором говорится, что выполнение команды не поддерживается в окне **MapBasic** или программы.

Заказные рамки нельзя клонировать. Клонирование окна **Отчёт**, в котором имеется заказная рамка, приводит к тому, что MapInfo Pro клонирует все рамки окна **Отчёт** кроме заказной рамки.

Примечание: Масштабная линейка в окне **Отчёт** является примером заказной рамки.

Оператор Add Designer Frame

Оператор **Add Designer Frame** добавляет разделы легенды в окно **Легенда**, который создает оператор **Create Designer Legend**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Designer Frame
[ Window legend_window_id ]
[ Custom ]
```



```

Frame From Layer { map_layer_id | map_layer_name }
[ Position ( x, y ) [ Units paper_units ] ]
[ Title { frame_title [ Font... ] } ]
[ SubTitle { frame_subtitle [ Font... ] } ]
[ Columns number_of_columns ] |
[ Height frame_height [ Units paper_units ] ]
[ Border Pen .... ] [ Brush ... ]
[ Priority priority_n ]
[ Name frame_name ]
[ Region [ Height region_height [ Units paper_units ] ] ]
[ Region [ Width region_width [ Units paper_units ] ] ]
[ Line [ Width line_width [ Units paper_units ] ] ]
[ Auto Font Size { On | Off } ]
[ Order { Default | Ascending | Descending |
{ Custom id | id : id [ , id | id : id ... ] ... } } ]
[ Using
[ Column { column | Object } [ FromMapCatalog { On | Off } ] ]
[ Label { expression | Default } ] ]
[ Display { Gap | Palette | Bar
[ [ Thickness t_value ] [ Length l_value ]
[ Units paper_units ] [ Horizontal | Vertical ] ] ] ]
[ Filter Max max_value ]
[ Sort Alpha ]
[ Into ID frame_id ]
[ Style [ Font... ] [ Norefresh ] [ Count { On | Off } ]
[ Text { style_name }
{ [ Line Pen... ] | [ Region Pen... ] [ Brush... ] | [ Symbol... ] |

Collection [ Symbol... ] [ Line Pen... ] [ Region Pen... ] [
Brush... ] }
[ Count count_value ]
[ Display { On | Off } ] ]
[ Text . . . ] ]

```

map_window_id — целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#). Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

map_layer_id или *map_layer_name* определяют слой карты; это может быть целая величина Smallint (например, используйте 1 для определения самого верхнего слоя, не считая косметического) или строковая величина, соответствующая имени таблицы, отображенной на карте. Для тематического слоя необходимо определять параметр *map_layer_id*.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "см" для сантиметров). Список "бумажных" единиц измерения приведен в разделе: [Оператора Set Paper Units](#).

frame_title — строковая переменная, определяющая заголовок раздела легенды. Если для рамки задано предложение **Title**, то именно оно будет использоваться для этой рамки вместо стандартного заголовка рамки, значение которого хранится вместе с другими параметрами окна, использованными при создании окна.

frame_subtitle — строковая переменная, определяющая подзаголовок раздела легенды. Если для рамки задано предложение **Subtitle**, то именно оно будет использоваться для этой рамки вместо стандартного подзаголовка рамки, значение которого хранится вместе с другими параметрами окна, использованными при создании окна.

number_of_columns — количество колонок, которые будут показаны в разделе.

frame_height — используется для запоминания высоты рамки легенды, изначально задаваемой предложением **Column**. Представляет высоту рамки в "бумажных" единицах. Значение записывается в WOR-файл, если размер рамки был изменен вручную.

priority_n — целочисленное значение, обозначающее порядок сортировки разделов легенды, которая встраивается в окно **Отчёт**.

framename — строка, определяющая название раздела легенды в окне **Отчёт**.

Значения *region_height* и *region_width* представляют новые высоту и ширину условных знаков площадных объектов. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров.

Значение *line_width* представляет ширину линейного условного знака. Можно задать от 12 до 144 пунктов, от 1 до 12 пик, от 0.666667 до 2 дюймов, от 4.23333 до 50.8 миллиметров или от 4.23333 до 50.8 сантиметров.

column — имя атрибутивной колонки из раздела слоя таблицы.

Expression — корректное MapBasic-выражение.

t_value — толщина цветовой полосы для растровой легенды. Единицами измерения являются дюймы, если другие единицы измерения не заданы с помощью оператора **Units**.

l_value — длина цветовой полосы для растровой легенды. Единицами измерения являются дюймы, если другие единицы измерения не заданы с помощью оператора **Units**.

max_value — максимальное количество цветов, которое отображается в легенде раstra.

Frame_id — идентификатор рамки (ID от 1 до общего количества рамок) для пустой рамки отчёта в текущем окне отчёта.

style_name — строковая переменная, которая указывает к какому типу: символу, линии или области относится объект в разделе легенды.

count_value — значение счётчика определённой строки растровой легенды.

Описание

Изначально окно конструктора легенды создает **оператор Create Designer Legend**. Далее можно добавить перестраивать легенду, добавлять рамки и изменять их свойства. В отличие от оператора **Add Cartographic Frame** оператор **Add Designer Frame** позволяет задавать количество колонок в рамке легенды предложением **Columns**, чего не делает оператор **Add Cartographic Frame**.

Если ключевое слово **Custom** включено в описание, то каждый раздел легенды должен включать в себя и предложение **Position**. Если слово **Custom** пропущено и легенда отображается в виде книжной или альбомной ориентации, то разделы легенды будут добавлены в самый конец.

Параметр **Frame From Layer** представляет одну легенду, создаваемую в окне **Легенда**. Используйте короткое целочисленное (SmallInt) значение, например, либо единицу (1), чтобы задать самый верхний слой карты, отличный от косметического слоя, либо строку, в которой представлено имя слоя существующей карты. Для тематического слоя необходимо определять параметр *map_layer_id*.

Предложение **Position** позволяет задавать положение раздела, если определён параметр **Custom** в MapBasic-операторе, который определяет окно. X-координата измеряется от левой стороны окна **Легенда**, а Y-координата от верхней стороны окна **Легенда**. Начало координат (0,0) находится в левом верхнем углу окна **Легенда**.

Если у раздела имеется заголовок, задаваемый параметром **Title**, то этот параметр переопределяет стандартный заголовок, который, обычно, задаётся параметром **Default Frame Title**. Если для оформления шрифта заголовка используется параметр **Font**, то стандартный шрифт заголовка раздела переопределяется, если такой параметр существует.

Если у раздела имеется подзаголовок, задаваемый параметром **Subtitle**, то этот параметр переопределяет стандартный заголовок, который, обычно, задаётся параметром **Default Frame Subtitle**. Если для оформления шрифта подзаголовка используется параметр **Font**, то стандартный шрифт подзаголовка раздела переопределяется, если такой параметр существует.

Предложение **Height** используется вместо предложения **Columns**, если пользователь изменял размер раздела. Если наличествуют обе, то используется предложение **Columns**.

Параметры **Columns** и **Height** можно применять и к тематическим, и к картографическим легендам. Однако, все свойства тематических легенд должны быть заданы с помощью **оператора Set Legend**.

Предложение **Border** используется только для того, чтобы задавать рамочку и фон легенды, когда она встраивается в окно **Отчёт**.

Параметр **Pen** всегда завершается корректным **предложением Pen**. Это предложение позволяет включать (плотное заполнение цветом) или выключать (штриховка) и задавать цвет рамочки вокруг раздела, который встраивается в окно **Отчёт**.

Brush — корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение используется только для того, чтобы задавать цвет фона легенды, когда она встраивается в окно **Отчёт**.

Предложение **Priority** позволяет задавать обратный порядок сортировки разделов легенды, которые встраиваются в окно **Отчёт**.

Предложение **Name** позволяет задавать название легенды, которая встраивается в окно **Отчёт**.

Предложение **Region Height** позволяет задавать высоту условных знаков легенды и переопределяет стандартное значение, заданное для раздела.

Предложение **Region Width** позволяет задавать ширину условных знаков легенды и переопределяет стандартное значение, заданное для раздела.

Предложение **Line Width** позволяет задавать толщину условных знаков линий легенды и переопределяет стандартное значение, заданное для раздела.

Auto Font Size — позволяет включать и выключать режим подстройки размера шрифтов условных знаков на основе размеров шрифта, заданных параметром **Style** стиля оформления.

С помощью параметра **Order** добавлена возможность либо включать сортировку, либо изменять способ сортировки рядов разделов легенд. Легенды карты можно сортировать либо по стилям, либо задавая индивидуальный порядок сортировки. Сортировать можно по следующим значениям параметра сортировки: **Default**, **Ascending**, **Descending** или **Custom**. Параметр **Order** записывается в рабочем наборе, только если легенда отсортирована в возрастающем, убывающем или индивидуальном порядке.

Порядок сортировки по умолчанию **Default** — это порядок сортировки, который устанавливает **Конструктор легенды**. Если легенда создаётся по уникальным стилям оформления объектов карты, то ряды условных знаков раздела будут перечислены в порядке, в котором объекты используются на карте. Если задан индивидуальный порядок сортировки условных знаков раздела легенды карты — **Custom**, то значение идентификатора *id* начинается с 1, увеличивается сверху вниз. Если проверить порядок перечисления условных знаков раздела в диалоге **Свойства раздела легенды**, то первый условный знак имеет идентификатор равный 1, номера следующих увеличиваются по мере перечисления в списке. Более подробные сведения содержит раздел [Настройка порядка сортировки](#) и раздел об [операторе Alter Designer Frame](#).

Into ID позволяет вставлять легенду в пустую рамку отчёта. Если рамка не пустая, то появится сообщение об ошибке. Если *frame_id* равняется 1 и рамка не пустая, то появится сообщение об ошибке "Рамка отчёта 1 должна быть пустой". Параметр **Into ID** является дополнительным, необязательным и любые легенды созданные или добавленные без него автоматически располагаются вдоль стороны карты, что является обычной стратегией. Окно **Карта** должно располагаться на той же странице отчёта, на которой размещается легенда: добавление легенды на другую страницу отчёта или, когда карта отсутствует в окне **Отчёт**, приводит к ошибке: "Родительская карта должна быть внедрена на текущей странице окна отчёта..". Идентификатор этой ошибки 1067.

Когда используется предложение **Into ID**, другие предложения пропускаются и не приводят к появлению сообщений об ошибках. Такими предложениями могут быть: **Position**, **Units**, **Height**, **Border**, **Brush**, **Priority** и **Name**. Размер рамки задаётся таким, чтобы в ней оптимально поместилась легенда.

Параметр **Using** позволяет определять, как создаётся легенда векторного слоя. Он ссылается на элементы таблицы при помощи предложений **Column** и **Label**.

Если задано предложение **Column**, то значение этого параметра должно представлять имя атрибутивной колонки в таблице, соответствующей слою, представленному в разделе, либо

должно стоять слово **Object**, обозначающее объектную колонку (это значит, что стили легенды основываются на уникальных стилях MAP-файла). По умолчанию используется **Object**.

FromMapCatalog ON отыскивает стили из MapCatalog для таблицы прямого доступа. Если таблица не является таблицей прямого доступа, MapBasic возвращается к заданному по умолчанию поведению для таблицы с непрямым доступом вместо того, чтобы допустить ошибку. Стандартное поведение для таблицы с отключённым доступом это **FromMapCatalog Off** (то есть, используются стили карты).

FromMapCatalog OFF отыскивает уникальные стили Карты для таблицы прямого доступа сервера. Эта таблица должна быть таблицей прямого доступа, которая поддерживает стилизацию каждой записи. Если таблица прямого доступа не поддерживает стили записей, то применяется заданное по умолчанию поведение для таблиц прямого доступа – используются заданные по умолчанию стили из MapCatalog (**FromMapCatalog ON**).

Предложение **Label** — это либо MapBasic-выражение *expression* для подписи, либо слово **Default** (последнее означает, что при создании текста используется стиль оформления рамки по умолчанию, если только предложение стилизации не описывает текст). Стандартным значением является **Default**.

Примечание: Сначала, для каждого векторного слоя, порождающего раздел легенды, исследуется соответствующий TAB-файл на наличие значений метаданных для заголовка легенды, подзаголовка, структуры колонок и т.п. Если метаданные присутствуют, то ими переопределяются значения, заданные предложением **Using**. Смотрите разделы: [функция GetMetadata\\$\(\)](#) и [функция Metadata](#).

Примечание: Когда MapInfo Pro создает легенду на основе типов объектов, то сначала рисуются точечные символы, затем линии, а затем области. Объекты типа коллекция изображаются последними. В пределах объекта типа коллекция порядок рисования таков: точки, линии, объекты.

Предложение **Display { Gap | Palette | Bar }** применяется только к растровым легендам.

Display Bar — позволяет отображать легенду непрерывного растра с помощью цветовой полосы. Для такого предложения существуют параметры, опции, позволяющие задавать размер полосы и ориентацию: толщина — **Thickness**, длина — **Length**, единицы измерения — **Units** и для ориентации — (**Horizontal** или **Vertical**). Параметр **Units**, задающий единицы измерения, является дополнительным, необязательным. Если от не задаётся, то MapInfo Pro использует дюймы. Если пропущен параметр, отвечающий либо за толщину, либо длину, либо ориентацию, MapInfo Pro использует значения, заданные предложением **Default Frame Bar**.

Display Palette — позволяет отображать легенду непрерывного растра в виде условных знаков, содержащих цвета заливок, с подписями. Размер условных знаков зависит от параметров, которыми задаются ширина и высота разделов, либо, которые используются в качестве стандартных, принимаемых по умолчанию.

Display Gap — также позволяет отображать условные знаки цветов заливки, но для тематических, классифицированных растровых слоёв. Если предложение **Display** пропущено, а исходный растровый слой является:

- непрерывным, MapInfo Pro использует предложение **Display Bar**, когда имеется более 20 цветов, в остальных случаях используется **Display Palette**;
- тематическим, классифицированным — MapInfo Pro использует предложение **Display Gap**.

Предложение **Filter Max** позволяет задавать максимальное количество строк или цветов, создаваемых для легенды растра. Это предложение является дополнительным, необязательным и является корректным только для разделов растровых легенд. Стандартным значением является 100. Если это предложение не используется, то MapInfo Pro отобразит в растровой легенде 100 оттенков цвета. Возможные значения — от 1 до 1600.

Предложение **Sort Alpha** используется с тематическими, классифицированными разделами растровых легенд. Оно позволяет сортировать подписи в алфавитном порядке.

Предложение **Style** и ключевое слово **NoRefresh** позволяют создавать собственные легенды, которые не будут изменяться при обновлении легенды. Однако, предложение **Style** должно содержать собственный, индивидуальный список стилей, используемых в легенды. Это выполняется с помощью параметров операторов в предложении **Text** и, соответственно, в предложениях **Line**, **Region**, **Symbol** или **Collection**.

Предложение **Font** позволяет задавать стиль оформления текста подписей строк и переопределяет параметры шрифта, заданные оператором **Default Frame Style**.

Параметр счётчика **Count { On | Off }** применяется только к растровым легендам и позволяет управлять показом значения количества записей счётчика строк. По умолчанию этот режим **выключен**. Этот параметр игнорируется, когда растровая легенда отображается в иде полосы цветов.

Каждая строка легенды начинается с предложения **Text**, за которым следуют предложения **Line**, **Region**, **Symbol** или **Collection**.

В конце предложения **Text** можно применять дополнительное выражение **Count**, которое применяется только к растровым легендам. Если растровый слой содержит счётчик записей, то они будут сохранены и использоваться в каждой строке легенды. Это не следует менять, поэтому **оператор Alter Designer Frame** не поддерживает изменения такого значения.

В конце каждого предложения **Style** может применяться дополнительный оператор **Display On**. **Display On** — делает ряд видимым, а **Display Off** — невидимым. Если параметр **Display** отсутствует, то ряд с условным знаком видим. Параметр **Display** записывается в рабочем наборе только для невидимых в разделе легенды условных знаков.

Обратите внимание на следующее:

- При добавлении разделов легенды из MapInfo Pro, новые разделы будут выбраны.
- Когда разделы легенды добавляются из MapBasic-программы, новые разделы не будут выбраны и предыдущая выборка останется неизменной.
- **Функция LayoutInfo()** включает тематические легенды в счётчик разделов.

Примеры

Сначала **функция NumAllWindows()** возвращает нам количество открытых окон документов, затем мы организуем по ним цикл, используя **функцию WindowID()** или **функцию WindowInfo()**, чтобы найти окно легенды по типу и ID-номеру окна, после чего применяем оператор Add Legend Designer к найденному окну.

```
Dim i, wndLegend as integer
for i = 1 to NumAllWindows()
  If WindowInfo(WindowID(i), WIN_INFO_TYPE) = WIN_LEGEND_DESIGNER then
    wndLegend = WindowInfo(WindowID(i), WIN_INFO_WINDOWID)
  end if
next
```

Следующий пример добавляет рамки.

```
Add Designer Frame Window wndLegend Frame From Layer 2 Columns 2
```

Для того чтобы получить идентификатор пользуйтесь следующим приёмом, где `window_id` – целое значение идентификатора окна **Отчёт**, а `frame_id` – число, с помощью которого задаётся рамка в окне **Отчёт**, где требуется выполнить запрос.

```
LayoutItemInfo(window_id, frame_id, 10)
```

Например, если в отчёте имеется единственная рамка с картой и одну или несколько легенд, то параметр `frame_id` рамки с картой будет 1, а рамки с легендой — 2. Следующий оператор возвратит идентификатор окна для окна **Легенда**, где 2 относится к рамке с легендой:

```
Dim LD_window as Integer
LD_window = LayoutItemInfo(window_id, 2, 10)
```

Результат вышеприведенного оператора можно использовать, для того чтобы добавлять легенды:

```
Add Designer Frame Window window_id Frame From Layer . . .
```

См. также:

оператор Create Designer Legend, **оператор Set Designer Legend**, **оператор Add Designer Frame**, **оператор Add Designer Text**, **оператор Add Image Frame**, **оператор Add Image Page**, **оператор Alter Designer Frame**, **оператор Remove Designer Frame**

Оператор Add Designer Page

Этот оператор добавляет одну страницу в окно **Отчёт**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Designer Page
[ Window layout_win ]
```

window_id – идентификатор окна **Отчёт**, целое число;

Описание

Оператор **Add Designer Page** добавляет оду страницу в окно **Отчёт**. Страница добавляется в конец коллекции страниц (в конце списка уменьшенных изображений). Нет ограничений на количество страниц, которое можно добавлять, это зависит от доступной памяти.

См. также:

[оператор Create Designer Legend](#), [оператор Add Designer Frame](#), [оператор Add Designer Text](#), [оператор Add Image Frame](#), [оператор Remove Designer Page](#)

Оператор Add Designer Text

Этот оператор добавляет текстовые рамки в окно **Легенда**, которое создаётся оператором [Create Designer Legend](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Designer Text
[ Window legend_window_id ]
Legend Text Frame
Text { frame_text [ Font... ] }
[ Position ( x, y ) [ Units paper_units ] ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#).

frame_text – текст для заголовка, подзаголовка легенды, либо обычный текст (например, сообщение об авторских правах).

x – определяет требуемое расстояние от верхнего края рабочего стола MapInfo до верхнего края окна.

y – определяет требуемое расстояние от левого края рабочего стола MapInfo до левого края окна.

Примечание: Здесь под рабочим столом подразумевается область с данными на экране (без строк заголовка и состояния программы и панелей инструментов).

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цитеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Описание

Legend Text Frame создает текстовый раздел в окне **Легенда**.

Если в предложении **Text** не задан параметр шрифта **Font**, то используется стандартный шрифт.

Предложение **Position** контролирует положение раздела в окне **Легенда**. Верхний левый угол окна **Легенда** имеет позицию 0, 0. **Position** использует "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры), см. [оператор Set Paper Units](#). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя [оператор Set Paper Units](#). Вы можете задать свои единицы измерения, явно задав предложение **Units** в предложение **Position**.

Пример

```
Add Designer Text Window frontwindow()
  Legend Text Frame
    Text "This is My title" Font("Batang", 3, 12, 16711680)
```

См. также:

[оператор Create Designer Legend](#), [лнепатоп Add Designer Frame](#), [оператор Add Designer Text](#), [оператор Add Image Frame](#), [оператор Add Image Page](#), [оператор Alter Designer Text](#), [оператор Remove Designer Text](#)

Оператор Add Image Frame

Назначение

Добавляет рамку с изображением в окно **Отчёт**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Add Image Frame [ Window window_id ]
[ Into ID frame_id ]
[ Position ( x, y ) [ Units paper_units ] ]
[ Width frame_width [ Units paper_units ] ]
[ Height frame_height [ Units paper_units ] ]
[ Angle angle ]
[ Pen .... ] [ Brush ... ] [ Priority n ] [ Name frame_name ]
From { File image_file_name }
```

window_id — идентификатор окна **Отчёт**, целое число;

Frame_id — идентификатор рамки (ID от 1 до общего количества рамок) для пустой рамки отчёта в текущем окне отчёта.

x, y — задаёт в "бумажных" единицах (*paper_units*) положение левого верхнего угла рамки с изображением в окне **Конструктор отчёта**.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "см" для сантиметров).

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

angle Дополнительное, необязательное предложение **Angle** позволяет задавать в градусах угол поворота рамки с изображением; угол задаётся вещественным числом. По умолчанию это нуль (0).

frame_width и *frame_height* — задают ширину и высоту рамки в окне **Отчёт**. Окно **Отчёт** сохраняет отношение сторон изображения (отношение между высотой и шириной) поэтому, когда задаются оба параметра: и высота, и рамка, используется только последнее измерение, по которому вычисляется неизвестное измерение.

image_file_name — строка с именем добавляемого файла изображения. Поддерживаются форматы: jpg, png, bmp, gif, tif и ico.

n — целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

frame_name — строка, определяющая имя рамки с изображением в окне конструктора отчёта. Если рамке с изображением в окне **Отчёт** присваивается имя, то оно будет сохранено в рабочем наборе. Когда параметр Name записывается в рабочем наборе, то номер версии WOR-файла меняется на 1500.

Описание

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Дополнительное, необязательный параметр **Position** позволяет разместить изображение внутри окна **Отчёт**. Если оно опущено, то изображение размещается в центре видимой области отчёта.

Предложения **Width** и **Height** определяют ширину и высоту рамки с изображением. Если предложений **Width** и **Height** нет, то используются значения измерений изображений. Если задаётся только одно значение, то по нему вычисляется неизвестное значение, используя известное отношение сторон изображения.

Brush — корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение создает фон для элемента оформления.

Pen — корректное **предложение Pen**. Это предложение задает стиль рамки элемента украшения.

Into ID позволяет вставлять изображение в пустую рамку отчёта. Если рамка не пустая, то появится сообщение об ошибке. Если *frame_id* равняется 1 и рамка не пустая, то появится сообщение об ошибке "Рамка отчёта 1 должна быть пустой".

Когда используется предложение **Into ID**, другие предложения пропускаются и не приводят к появлению сообщений об ошибках. Такими предложениями являются: **Position**, **Width** и **Height**. Размер рамки задаётся таким, чтобы в ней оптимально поместилось изображение.

См. также:

оператор Add Designer Frame, **оператор Add Designer Text**, **оператор Add Image Frame**, **оператор Add Image Page**, **функция FrontWindow()**, **оператор Set Paper Units**, **функция WindowID()**

Оператор Add Map

Назначение

Добавляет на карту один или несколько слоев с графическими объектами, группу слоев. Можно выбрать группу слоев, в которую необходимо добавить новые слои, и/или положение.

Синтаксис 1

```
Add Map
[ Window window_id ] [ Auto ]
Layer table [ , table [ Animate ] ... ]
[ [ DestGroupLayer group_id ] Position position ]
```

window_id – идентификатор окна карты, целое число;

table – имя открытой таблицы, объекты которой будут добавлены в окно карты. Эта таблица должна иметь разрешение на присоединение графических объектов (mappable).

group_id – идентификатор группы слоев; либо целое число (номер группы), либо строка (имя группы).

Параметр *position* – целочисленный указатель на номер слоя в списке группы, после которого требуется добавить новые слои.

Синтаксис 2

```
Add Map
[ Window window_id ] [ Auto ]
GroupLayer ( "friendly_name" [ , item ... ] )
[ [ DestGroupLayer group_id ] Position position ]
```

где:

```
item = table | [ GroupLayer ( "friendly_name" [ , item ... ] )
```

Описание

Оператор **Add Map** добавляет на карту либо одну или несколько открытых таблиц, либо группу слоев, но не одновременно в одном и том же операторе. Группа слоев может содержать любое количество вложенных слоев. После выполнения оператора MapInfo Pro перерисовывает окно карты, если только это не было запрещено **оператором Set Event Processing Off**, либо **оператором Set Map Redraw Off**.

Для получения идентификатора открытого окна для параметра *window_id* используются **функция FrontWindow()** и **функция WindowID()**. Если в операторе **Add Map** параметр *window_id* опущен, то слой будет добавлен в самое верхнее открытое окно.

Ключевое слово **Auto** позволяет MapInfo Pro автоматически упорядочить на карте слои или группы слоев. Растровые таблицы и таблицы с объектами типа "область" помещаются в самый конец списка слоев, а таблицы с точечными объектами – в начало списка.

Если слово **Auto** было опущено, то слой с объектами заданной таблицы будет самым верхним слоем на карте, другими словами, при перерисовке изображения в окне объекты этого слоя

или группы слоев выводятся на экран последними. Для изменения порядка слоев используется оператор **Set Map**.

Если используется параметр **DestGroupLayer**, ключевое слово Auto не применяется, а слои или группа слоев будут добавлены в список слоев в заданную группу, в заданном положении. Идентификатор группы 0 принадлежит самому верхнему слою списка. Если параметр **DestGroupLayer** пропущен номер идентификатора группы принимается равным 0.

Параметр *position* – целочисленный указатель на номер слоя в списке группы, после которого требуется добавить новые слои. Если параметр position пропущен, то используется первый номер группы (position = 1). Если заданный номер position превышает количество слоев в группе, то новые слои, и/или группа слоев будет вставлены в конец группы назначения.

Идентификатор слоя или группы слоев может быть либо число или имя. Диапазон номеров идентификатора групп может быть от 0 до количества групп в списке слоев.

Примечание: Нельзя добавлять слои между тематическими слоями. Тематические слои добавляются на карту в определенном, зависящем от исходных таблиц порядке, который требуется поддерживать. Если заданное положение добавленного слоя попадает между существующими в списке тематическими слоями, то его положение в списке будет изменено, чтобы избежать изменения порядка перечисления тематических слоев.

Добавление слоев с разными проекциями

Если добавляется слой с растровой таблицей и карта еще не содержит растровых слоев, то карта воспринимает координатную систему и проекцию растрового изображения. Если карта уже имеет два или более растровых слоев, то проекция окна заменяется на проекцию того растрового изображения, которое занимает большую часть окна.

Если включен режим перепроецирования растра, MapInfo Pro не меняет координатную систему карты даже если вы добавляете к ней растровую таблицу.

Если добавляемый слой не является растровым, то MapInfo Pro продолжает показывать окно карты с использованием той проекции и координатной системы, которые действовали до применения оператора **Add Map**, даже если таблица имела свою собственную и координатную систему. Если собственные проекции таблицы не совпадают с проекциями карты, MapInfo Pro динамически преобразует координаты таблицы для отображения ее данных на слое карты.

Примечание: При обновлении окна карты, содержащей такие слои, перерисовка изображения будет замедлена, так как MapInfo производит математические вычисления для перепроецирования данных.

Использование слоя анимации для ускорения перерисовки карты

Если в операторе **Add Map** присутствует слово Animate, то добавляемый слой становится анимационным. Когда объект на анимационном слое перемещается, окно карты перерисовывается очень быстро, потому что MapInfo Pro перерисовывает только слой с анимацией.

Пример работы анимационного слоя можно найти в программе `ANIMATOR.MB` из комплекта поставки.

Эффект анимации полезен в приложениях, отображающих процессы реального времени, в которых объекты карты должны часто и быстро перерисовываться. Например, пусть требуется разработать прикладную систему управления группой грузовых автомобилей, в которой каждый грузовик представлен точечным объектом. Информацию о положении грузовика получается с помощью устройства спутникового позиционирования GPS (Global Positioning Satellite), и эта информация должна незамедлительно отражаться в окне карты. В задачах подобного типа, когда объекты на карте постоянно перемещаются, их лучше размещать на анимационном слое, а не на обычном.

Следующие операторы открывают таблицу `Vehicles` и делают соответствующий слой анимационным:

```
Open Table "vehicles" Interactive
Add Map Layer vehicles Animate
```

Обычно, последняя таблица, за которой следует ключевое слово **Animate**, будет анимационным слоем. Единоновременно анимационным может быть только один слой.

Чтобы прекратить анимацию, используйте **оператор Remove Map Layer Animate**.

Анимационные слои подчиняются специальным ограничениям. Например, пользователь не может применить инструмент Информация к объекту анимационного слоя. Каждое окно карты может иметь только один анимационный слой. Более подробные сведения об анимационных слоях можно найти в *Руководстве пользователя MapBasic*.

Пример

```
Open Table "world"
Map From world
Open Table "cust1992" As customers
Open Table "lead1992" As leads
Add Map Auto Layer customers, leads
```

Пример добавления группы слоев:

```
Open Table world
Open Table worldcap
Add Map Auto GroupLayer("new group", worldcap, world)
Open Table ocean
Add Map Layer ocean DestGrouplayer "new group" position 3
```

См. также:

оператор Map, оператор Remove Map, оператор Set Map

Функция AdornmentInfo()

Назначение

Возвращает информацию об элементах оформления, таких как масштабная линейка. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
AdornmentInfo( window_id, attribute )
```

window_id – целое число, идентификатор окна; *Окно должно быть окном элемента управления.*

attribute – целочисленный код типа информации, которая будет возвращена. Значения приведены в таблице ниже.

Возвращаемая величина

Float, smallint, integer, logical, pen, brush, font или string, в зависимости от параметра атрибута.

Описание

Функция **AdornmentInfo()** возвращает значения в зависимости от кода атрибута. Коды определены в MAPBASIC.DEF.

Код атрибута	ID	Возвращаемое значение AdornmentInfo()
ADORNMENT_INFO_TYPE	1	Целое – Integer. Тип элемента оформления: 0 (ноль) для масштабной линейки.
ADORNMENT_INFO_MAP_WINDOWID	2	Целое – Integer. WindowID для порождающего окна карты.
ADORNMENT_INFO_IS_FIXED_POS	3	Логическое – Logical. TRUE фиксированной позиции, FALSE для прикрепленной.
ADORNMENT_INFO_FIXED_POS_X	4	Вещественное – Float. Значение X для фиксированной позиции.
ADORNMENT_INFO_FIXED_POS_Y	5	Вещественное – Float. Значение Y для фиксированной позиции.

Код атрибута	ID	Возвращаемое значение AdornmentInfo()
ADORNMENT_INFO_FIXED_POS_UNITS	6	Символьное – Char. Единицы измерения для значений X и Y для прикрепленной позиции.
ADORNMENT_INFO_DOCKED_POS	7	Короткое целое число – SmallInt. Возвращает показатель прикрепленной позиции.
ADORNMENT_INFO_DOCKED_OFFSET_X	8	Вещественное – Float. Значение X смещения по отношению к прикрепленной позиции.
ADORNMENT_INFO_DOCKED_OFFSET_Y	9	Вещественное – Float. Значение Y смещения по отношению к прикрепленной позиции.
ADORNMENT_INFO_DOCKED_UNITS	10	Символьное – Char. Единицы измерения для значений X и Y для прикрепленной позиции.
ADORNMENT_INFO_BACKGROUND_PEN	11	Pen. Стиль контура для фона элемента оформления.
ADORNMENT_INFO_BACKGROUND_BRUSH	12	Brush. Стиль штриховки для фона элемента оформления.
ADORNMENT_INFO_SB_TYPE	20	Короткое целое число – SmallInt. Тип масштабной линейки.
ADORNMENT_INFO_SB_MAP_UNITS	21	Символьное – Char. Единицы измерения расстояний в масштабной линейке.
ADORNMENT_INFO_SB_PAPER_UNITS	22	Символьное – Char. Единицы измерения макета в масштабной линейке. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
ADORNMENT_INFO_SB_BAR_LENGTH	23	Вещественное – Float. Исходный размер масштабной линейки.
ADORNMENT_INFO_SB_BAR_DRAW_LEN	24	Вещественное – Float. Размер нарисованной масштабной линейки.
ADORNMENT_INFO_SB_BAR_HEIGHT	25	Вещественное – Float. Высота масштабной линейки.
ADORNMENT_INFO_SB_AUTO_SCALING	26	Логическое – Logical. Индикатор режима автомасштабируемости.
ADORNMENT_INFO_SB_CARTO_SCALE	27	Логическое – Logical. Индикатор показа картографического масштаба в инструментальной панели.
ADORNMENT_INFO_SB_BAR_PEN	28	Pen. Стиль линии для масштабной линейки.

Код атрибута	ID	Возвращаемое значение AdornmentInfo()
ADORNMENT_INFO_SB_BAR_BRUSH	29	Brush. Стиль штриховки для масштабной линейки.
ADORNMENT_INFO_SB_BAR_FONT	30	Шрифт – Font. Шрифт текста для масштабной линейки.
ADORNMENT_INFO_SB_DISPLAY_SCALE	31	Вещественное – Float. Числовое значение масштаба, показанное в масштабной линейке.
ADORNMENT_INFO_SB_AUTOOFF_SCALE	32	Вещественное – Float. Значение масштаба, используемое, если отключён режим автоматического масштабирования.
ADORNMENT_INFO_SB_AUTOON_SCALE	33	Вещественное – Float. Значение масштаба, используемое, если включён режим автоматического масштабирования.
ADORNMENT_INFO_SB_SCALE_STRING	34	Символьное – Char. Масштаб в виде текстовой строки (с разделителями тысяч и десятичными разделителями).
ADORNMENT_INFO_SB_CARTO_VALUE	35	Вещественное – Float. Картографический масштаб в виде числа, без округления.
ADORNMENT_INFO_SB_CARTO_STRING	36	Символьное – Char. Картографический масштаб в виде текстовой строки.

Пример

```
print AdornmentInfo(MapperInfo(FrontWindow(), 201), 1)
```

Оператор Alter Button

Назначение

Управляет выбором и доступностью кнопки в панели инструментов или группе ленты.

Синтаксис

```
Alter Button { handler | ID button_id }
[ { Enable | Disable } ]
[ { Check | Uncheck } ]
```

handler – обработчик, который уже назначен существующей кнопке. Обработчиком *handler* может быть имя процедуры MapBasic и стандартный командный код из файла `MENU.DEF`, например, `M_TOOLS_RULER` или `M_WINDOW_LEGEND`);

button_id – идентификатор кнопки.

Описание

Если оператор **Alter Button** использует обработчик, например, имя процедуры, то изменения затронут все кнопки, вызывающие этот обработчик. Если используется предложение ID, то MapInfo Pro изменяет только кнопку с номером *button_id*.

Ключевое слово **Disable** делает кнопку недоступной. Кнопка закрашивается серым цветом и не реагирует на указание мышкой.

Ключевое слово **Enable** возвращает кнопку в активное состояние.

Слова **Check** и **Uncheck** нажимают или отжимают кнопку типа `ToggleButton` (кнопку, для которой состояние выбора фиксируется). Слово **Check** делает “нажатой” кнопку `ToggleButton`, а слово **Uncheck** возвращает кнопку в исходное положение. Например, для кнопки “Показать окно Статистики”:

```
Alter Button M_WINDOW_STATISTICS Check
```

Примечание: Нажатие кнопки не означает автоматическое выполнение действий, закрепленных за кнопкой; так, например, выбор кнопки Показать/Скрыть окно Статистики оператором `Alter Button` не открывает окно Статистика, а только показывает эту кнопку нажатой. Чтобы открыть окно Статистики надо использовать соответствующий **оператор Open Window Statistics**.

Можно использовать слово **Check** для изменения состояния кнопки `ToolButton`. Однако, изменение состояния нажатия на кнопку типа `ToolButton` не то же самое, что и выбор инструмента этой кнопкой. Для выбора инструмента после изменения состояния кнопки можно использовать **оператор Run Menu Command**, например:

```
Run Menu Command M_TOOLS_RULER
```

Для того чтобы активизировать созданный инструмент, используется оператор *Run Menu Command ID IDnum*.

Полезно учитывать при использовании этой команды в 64-разрядной версии MapInfo Pro:

1. Эта команда позволяет включать\выключать или выбирать\отменять выбор элемента управления в группе инструментов.
2. Если команда отмечена\снята с использованием оператора `run menu command`, то этим затрагиваются связанные элементы управления, которые имеют свойство *isChecked*.

См. также:

[оператор Alter ButtonPad](#), [оператор Create ButtonPad](#), [оператор Run Menu Command](#)

Оператор Alter ButtonPad

Назначение

Показывает/скрывает панель инструментов или добавляет/переопределяет в ней кнопку. В 64-разрядной версии выполняет тоже для группы команд в ленте. Кроме того, позволяет изменять уже созданную панель инструментов, если панель инструментов не существует, то MIPго будет автоматически создавать такую панель инструментов. В 64-разрядной версии выполняет тоже для группы команд в ленте.

Синтаксис

```
Alter ButtonPad { current_title | ID pad_num }
[ Add button_definition [ button_definition ... ] ]
[ Remove { handler_num | ID button_id } [ , ... ] ]
[ Title new_title ]
[ Tab tab_name | tab_caption ]
[ Width width ]
[ Position ( x, y ) [ Units unit_name ] ]
[ ToolbarPosition ( row, column ) ]
[ { Show | Hide } ]
[ { Fixed | Float | Top | Left | Right | Bottom } ]
[ Destroy ]
```

current_title – имя панели (например, "Операции");

pad_num – ID-номер инструментальной панели:

- 1 – для панели Операции
- 2 – для панели Пенал
- 3 – для панели Программы
- 4 – для панели Команды
- 5 – для панели СУБД
- 6 – для панели Web-службы
- 7 – зарезервировано

handler_num — целочисленный код обработчика, например, M_TOOLS_RULER (1710), из файла MENU.DEF.

button_id — уникальный идентификатор для новой кнопки;

new_title — заголовок инструментальной панели; пользователь его видит, когда панель находится в плавающем режиме. В 64-разрядной версии MapInfo Pro является заголовком группы команды ленты.

tab_name, *tab_caption*. Когда приходится менять существующую вкладку ленты, параметр *tab_name* является именем вкладки, например, "TabMap", а не текстом названия вкладки, который выводится на ленте.

Примечание: Имена существующих элементов управления ленты, NAMES, используются для построения и работы модуля ProSampleAddin, который включён в MapBasic и в примере программы-инструмента Dump Object Model.

```
(\MapBasic\Samples\RIBBONINTERFACE\DotNet\ProSampleAddIn)
```

width — ширина панели, измеряется количеством кнопок. Не поддерживается в 64-разрядной версии MapInfo Pro.

x, *y* — координаты верхнего левого угла панели в "бумажных" единицах. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#). Не поддерживается в 64-разрядной версии MapInfo Pro.

unit_name — имя "бумажной" единицы (например, "in" — дюйм, "cm" — сантиметр). Не поддерживается в 64-разрядной версии MapInfo Pro.

row, *column* — координаты инструментальной панели, когда она находится в закреплённом состоянии (например, 0, 0 задает расположение строки инструментов, прижатой к левому краю, и самой верхней строкой, а 0, 1 — расположение второй панели инструментов в верхнем ряду). Не поддерживается в 64-разрядной версии MapInfo Pro.

- Значение *row* отсчитывается сверху вниз. Это значение высчитывается относительно других строк с инструментами, находящемся выше или ниже. Если в той же позиции расположена строка меню, то отсчёт ведётся относительно строки меню. Когда панель инструментов находится прямо под строкой меню, номером её строки является значение равное 0. Если панель находится выше строки меню, то её значение равняется -1.
- Значение *column* отсчитывается слева направо. Это значение высчитывается относительно других инструментальных строк, находящемся правее или левее. Например, если панель инструментов закреплена слева и строка меню закреплена в левом положении, то номером столбца левее меню будет -1. Номером колонки, которая располагается правее меню, будет 0.

Каждый параметр *button_definition* является либо ключевым словом **Separator**, либо группой предложений следующего синтаксиса:

```
{ PushButton | ToggleButton | ToolButton }
  Calling { procedure | menu_code | OLE methodname | DDE server, topic }
```

```
[ ID button_id ]
[ Small | ExtraSmall | Large Icon icon_code [ File file_spec ] ]
[ Cursor cursor_code [ File file_spec ] ]
[ DrawMode dm_code ]
[ HelpMsg msg ]
[ ModifierKeys { On | Off } ]
[ { Enable | Disable } ]
[ { Check | Uncheck } ]
```

procedure — имя процедуры-обработчика, вызываемой при нажатии на кнопку;

menu_code — стандартный в MapInfo Pro код меню из файла MENU.DEF, например M_FILE_OPEN (102); при нажатии кнопки MapInfo Pro выполняет эту команду меню;

methodname — строковая величина, задающая имя метода OLE. Синтаксис предложения **Calling OLE** смотрите в разделе: [Оператор Create ButtonPad](#).

server и *topic* — строковые величины, задающие сервер DDE-связи и имя раздела (topic). Более подробно о синтаксисе предложения **Calling DDE** смотрите в разделе: [Оператор Create ButtonPad](#).

button_id — назначает уникальный номер для кнопки. Этот номер можно использовать либо как число, используемое при вызове Справки (Tag in Help); либо как идентификатор кнопки в ее процедуре-обработчике, когда один и тот же обработчик вызывают несколько кнопок; либо вызывая [оператор Alter Button](#).

Icon *icon_code* — задает пиктограмму, которая будет на кнопке. Здесь *icon_code* может быть одним из специальных кодов файла ICONS.DEF, например, MI_ICON_RULER (11). Предложение File задает файл ресурсов изображений; в этом случае параметр *icon_code* должен быть целочисленным идентификатором одного из ресурсов заданного файла. Размер кнопки можно задать с помощью идентификатора файла ресурсов *icon_code* для кнопок малого размера и *icon_code*+1 для кнопок большого размера; то есть одна кнопка представлена в двух вариантах с id-номерами *icon_code* и *icon_code*+1.

Примечание: Для параметра *Icon Code* теперь можно использовать Uri-синтаксис в токене File.

Small | ExtraSmall | Large — позволяют задавать размер иконок-пиктограмм.

Cursor *cursor_code* — задает форму указателя мыши, если пользователь выберет кнопку инструмента (тип ToolButton); коды *cursor_code*, такие как MI_CURSOR_ARROW (0), задаются в файле ICONS.DEF. Это предложение используется только для кнопок инструментов. Предложение **File** задает имя файла ресурсов изображений; в этом случае параметр *cursor_code* должен быть целочисленным идентификатором одного из ресурсов заданного файла.

В 64-разрядной версии MapInfo Pro вид курсора поддерживается тремя способами:

1. В виде строки, описанной в файле Icons.def: `SetRbnToolBtnCtrlCursor(MyToolButton, "138")`

2. В виде строки с ключевым словом FILE и именем DLL, где хранятся курсоры:
`SetRbnToolBtnCtrlCursor(MyToolButton, "136 FILE gcsres32.dll")`
3. Как целое число с идентификатором курсора MapInfo в Icons.def:
`SetRbnToolBtnCtrlCursorId(MyToolButton, MI_CURSOR_FINGER_LEFT)`

`dm_code` — определяет, сможет ли пользователь перемещать объекты мышкой или только выбирать их; коды `dm_code`, такие как DM_CUSTOM_LINE (33), задаются в файле `ICONS.DEF`.

`msg` — строковая величина, задает текст подсказки, появляющейся в строке сообщений при указании на кнопку, а также может задавать текст для плавающей подсказки ToolTip. Первая часть строки `msg` используется строкой сообщений. Если величина `msg` включает в себя литеры `\n`, то текст, следующий за `\n`, используется подсказкой ToolTip.

Предложение **ModifierKeys** управляет использованием клавиш **SHIFT** и **CTRL** в режиме рисования, сопровождающемся прорисовкой образа объекта ("rubber-band"), инструментом кнопки типа ToolButton. По умолчанию применяется режим Off, не использующий клавиши SHIFT и CTRL.

Описание

Оператор **Alter ButtonPad** используется для отображения, скрытия, изменения или уничтожения существующих инструментальных панелей. Более подробная информация об инструментальных панелях содержится в *Руководстве пользователя MapBasic*.

Для того, чтобы показать панель, в операторе используется ключевое слово **Show**; для того, чтобы её скрыть — ключевое слово **Hide**; см. пример ниже.

Ключевые слова **Fixed** и **Float** задают состояние панели, в котором она отображается на экране: закреплённой в верхнем краю окна или плавающей в виде вспомогательного окна. Пользователь также может управлять тем, закреплена ли панель или нет, перетаскивая её в верхнюю часть экрана. Дополнительные возможности управления положением панели дают ключевые слова **Top** (то же самое, что и **Fixed**), **Left**, **Right** и **Bottom**.

Предложение **Position** задает расположение панели на экране, если она находится в плавающем состоянии. Предложение **ToolbarPosition** задает расположение панели, прикреплённой к верхнему краю рабочего окна.

С помощью ключевого слова **Destroy** можно убрать панель насовсем, то есть имя этой панели инструментов больше не будет отображаться в списке диалога команды **Инструменты**.

Оператор **Alter ButtonPad** может добавлять кнопки в панели инструментов Операции и Пенал. Существуют три типа кнопок, которые можно добавить оператором: тип PushButton — кнопка, нажатие которой приводит к определенному действию, например, к открытию диалога; тип ToggleButton — кнопка с фиксированным выбором, то есть после нажатия она остается в состоянии выбора (на панели она "утоплена"), для отмены выбора надо нажать ее снова; тип ToolButton — кнопка инструмента, который пользователь может использовать только в окне карты или отчёта.

Если создаете кнопку с ключом **Disable**, новая кнопка будет недоступна для выбора (недоступная кнопка закрашивается серым). Доступной кнопку делает **оператор Alter Button**.

Однако, если обработчиком (handler) кнопки определена стандартная в MapInfo Pro команда, MapInfo Pro автоматически изменит доступность кнопки в зависимости от возможности выполнения этой команды.

Если добавить слово **Check** в описание элементов ToggleButton или ToolButton, то соответствующая кнопка при появлении будет автоматически выбрана.

Если пользователь нажал на кнопку типа ToolButton, MapInfo Pro автоматически вызывает соответствующий обработчик инструмента, который будет действовать до тех пор, пока пользователь не отменит его (например, клавишей ESC). В процедуре обработчика может быть использована функция **CommandInfo()** для определения места, на которое пользователь указал мышкой. Если две или более инструментальные кнопки вызывают один обработчик, то функция **CommandInfo()** поможет определить идентификатор кнопки, вызвавший его.

Добавлять кнопки в существующие на вкладке группы можно с помощью оператора **Alter ButtonPad**. Например,

```
Alter ButtonPad "PushButtons" Add PushButton small
```

Когда изменяется панель инструментов **ButtonPad**, для того чтобы удалить кнопку, требуется задавать имя вкладки, на которой располагается панель инструментов **ButtonPad**. Например,

```
alter buttonpad "MapFile" add pushbutton calling HelloRibbon Small icon  
46 Title "Hello" helpmsg "help on hello" Tab "TabMap"
```

Встроенные картинки для кнопок и указателя мышки

Предложение **Icon** определяет пиктограмму, которую пользователь увидит на кнопке. Если опустить предложение **File**, то параметр *n* должен задавать код из файла `ICONS.DEF`, такой как `MI_ICON_RULER` (11).

Примечание: MapInfo Pro имеет много встроенных иконок для кнопок, не являющихся частью нормального пользовательского интерфейса. Для того чтобы проверить вид иконок, выполните пример программы `ICONDEMO.MBX`. Оттуда можно скопировать код определения любой иконки в буфер обмена (чтобы затем вставить код в вашу программу).

В Windows предложение **File file_spec** ссылается на DLL-файл, который содержит ресурсы растровых изображений, а параметр *n* в предложении **Icon** является идентификатором ID одного из них. Более подробно создание новых кнопок в Windows описано в *Руководстве пользователя MapBasic*.

Для кнопок панелей инструментов типа ToolButton также можно определить в предложении **Cursor** картинку, форму которой примет указатель мыши в то время, пока активен инструмент, вызванный этой кнопкой. Коды указателей мыши описаны в `ICONS.DEF`, например `MI_CURSOR_CROSSHAIR` (138) или `MI_CURSOR_ARROW` (0). Правила использования собственных ресурсов для указателя мыши такие же, как при определении пиктограммы кнопки.

Информацию о требованиях к пиктограммам MapInfo Pro содержат разделы: **Оператор Create ButtonPad** и **О размерах пиктограмм**.

Режимы рисования

Определение ToolButton может включать предложение **DrawMode**, которое определяет, может ли пользователь перемещать инструмент (например, чтобы нарисовать линию) или только щелкнуть (например, чтобы нарисовать точку). В следующей таблице перечислены возможные режимы рисования. Коды в левом столбце определены в файле `ICONS.DEF`.

Параметр DrawMode	ID	Описание
DM_CUSTOM_POINT	34	Инструмент может только указывать, а не рисовать.
DM_CUSTOM_LINE	33	При рисовании инструментом от указателя тянется прямая линия, закрепленная другим концом в точке, в которой была нажата клавиша мышки.
DM_CUSTOM_RECT	32	При рисовании инструментом указатель растягивает прямоугольник, закрепленный противоположным по диагонали углом в точке, в которой была нажата клавиша мышки.
DM_CUSTOM_CIRCLE	30	При рисовании указатель растягивает окружность.
DM_CUSTOM_ELLIPSE	31	При рисовании инструментом указатель растягивает эллипс. Если добавлено предложение <code>ModifierKeys</code> , то при нажатой клавише <code>SHIFT</code> будет растягиваться окружность.
DM_CUSTOM_POLYGON	35	Рисует многоугольник (полигон). Чтобы узнать, какой объект был нарисован пользователем, используется функция <code>CommandInfo()</code> (<code>CMD_INFO_CUSTOM_OBJ</code>).
DM_CUSTOM_POLYLINE	36	Рисует ломаную (полилинию). Чтобы узнать, какой объект был нарисован пользователем, используется функция <code>CommandInfo()</code> (<code>CMD_INFO_CUSTOM_OBJ</code>).

Все режимы рисования, исключая режим `DM_CUSTOM_POINT(34)`, поддерживают режим автопрокрутки, который позволяет автоматически перемещать окно карты или отчёта при операциях выбора и рисования с участием мыши. Отключает автопрокрутку **оператор `Set Window`**.

Примечание: MapBasic поддерживает дополнительный режим рисования, недоступный из среды MapInfo Pro. Если **Calling** задано в виде **Calling `M_TOOLS_SEARCH_POLYGON(1733)`**, то инструмент может рисовать многоугольник.

Когда пользователь дважды щёлкнет мышкой для закрытия полигона, MapInfo Pro выберет все объекты (с доступных слоев карты), попавшие в многоугольник. Многоугольник не сохраняется.

Традиционное поведение больше не поддерживается в панелях инструментов

1. [{ Fixed | Float | Top | Left | Right | Bottom }]
2. [Width width]
3. [Position (x, y) [Units unit_name]]
4. [ToolbarPosition (row, column)]

```
Alter Button { handler | ID button_id }
[ { Enable | Disable } ][ { Check | Uncheck } ]
```

Изменения в команде ButtonPad позволяют включать/выключать или отмечать/снимать флажок с элемента управления, добавленного в инструментальную панель или в группу инструментов.

Если какая-либо команда в меню MapInfo Pro отмечена/снята, то отмечаться/сниматься флажок будет только у тех элементов управления, которые имеют параметр IsChecked.

Пример:

```
Alter ButtonPad ID 1 Add PushButton ID 23002 HelpMsg "Open Table" calling
102
Alter Button ID 23002 Enable
Alter Button ID 23002 Disable
Alter Button ID 23002 Check
Alter Button ID 23002 Uncheck
```

Примеры

Следующие операторы показывают панель инструментов Операции и скрывают панель Пенал:

```
Alter ButtonPad "Main" Show
Alter ButtonPad "Drawing" Hide
```

Следующий пример прикрепляет панель Операции к верхнему краю рабочего окна:

```
Alter ButtonPad "Main" Fixed ToolbarPosition(0,0)
```

Теперь переведем панель Операции в плавающее состояние и поместим ее на расстоянии полдюйма от верхнего и левого краев экрана.

```
Alter ButtonPad "Main" Float Position(0.5,0.5) Units "in"
```

Прикладная программа из стандартной поставки SCALEBAR содержит оператор **Alter ButtonPad**, который добавляет кнопку типа ToolButton на панель инструментов Программы. ("ID 3" обозначает панель Программы.)

```
Alter ButtonPad ID 3
Add
  Separator
  ToolButton
    Icon MI_ICON_CROSSHAIR
    HelpMsg "Draw a distance scale on a map\nScale Bar"
    Cursor MI_CURSOR_CROSSHAIR
    DrawMode DM_CUSTOM_POINT
    Calling custom_tool_routine
Show
```

Примечание: Ключевое слово **Separator** вставляет пустое пространство между последней кнопкой панели Программы и новой кнопкой "+".

См. также:

[оператор Alter Button](#), [функция ButtonPadInfo\(\)](#), [оператор Create ButtonPad](#), [оператор Set Window](#)

Оператор Alter Cartographic Frame

Назначение

Оператор **Alter Cartographic Frame** изменяет положение, заголовок, подзаголовок, рамку и стиль существующего раздела легенды, который создает [оператор Create Cartographic Legend](#). Для изменения размера, позиции или заголовка окна легенды, используйте [оператор Set Window](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Этот оператор не позволяет изменить параметры Title, Subtitle и Fonts для тематического раздела в окне **Легенда**. Эти изменения можно внести, используя [оператор Set Legend](#).

Синтаксис

```
Alter Cartographic Frame
[ Window legend_window_id ]
Id { frame_id }
[ Position ( x, y ) [ Units paper_units ] ]
[ Title [ frame_title ] [ Font... ] ]
[ SubTitle [ frame_subtitle ] [ Font... ] ]
[ Border Pen... ]
[ Style [ Font... ] ]
```

```
[ ID { id } Text { style_name } ]
[Line Pen... | Region Pen... Brush... | Symbol Symbol... ] ]
[ , ... ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#).

frame_id – ID-номер раздела легенды. Нельзя использовать имя слоя. Например, три раздела легенды могут иметь ID-номера 1, 2 и 3.

x, y – задаёт в "бумажных" единицах (*paper_units*) положение левого верхнего угла рамки с изображением в окне **Конструктор отчёта**.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цисцерио.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

frame_title – строковая переменная, определяющая заголовок раздела легенды.

frame_subtitle – строковая переменная, определяющая подзаголовок раздела легенды.

id – положение внутри списка стилей для данного раздела. Количество стилей в рамке может вам вернуть [функция LegendFrameInfo\(\)](#) с атрибутом FRAME_NUM_STYLES (13).

style_name – это строковая величина, которая отображает следующий за каждым символ для раздела с указанным ID-номером. Символ “#” будет замещаться именем текущего слоя. Символ % будет замещаться словами “Line”, “Point”, “Region”, которые будут соответствовать типу географических объектов (эти названия могут быть другими в русской версии). Например, “% of #” будет заменено на “Region of States” для раздела легенды, соответствующей слою таблицы STATES.TAB (текст может быть другим в русской версии).

Описание

Если предложение **Window** не определено, MapInfo Pro будет использовать самое верхнее окно легенды.

Предложение **Position** контролирует положение раздела в окне легенды. Верхний левый угол окна легенды имеет позицию 0, 0. Position использует "бумажные" единицы измерения, такие, как “in” (дюймы) или “cm” (сантиметры), см. [оператор Set Paper Units](#)). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя [оператор Set Paper Units](#). Оператор **Alter Cartographic Frame** может изменить единицы измерения с помощью предложения **Units** в предложении **Position**.

В предложения **Title** и **SubTitle** можно вводить новый текст, новый шрифт или и то, и другое.

Предложение **Style** должно содержать список определений для стилей, отображающихся в разделе. Можно обновлять только тип Style для собственного стиля. Можно обновлять текст для любого стиля. Нет возможности добавлять или удалять стили для любых типов разделов легенды.

См. также:

[оператор Create Cartographic Legend](#), [оператор Set Cartographic Legend](#), [оператор Add Cartographic Frame](#), [оператор Remove Cartographic Frame](#)

Оператор Alter Control

Назначение

Изменяет состояние элемента диалога, созданного приложением.

Синтаксис

```
Alter Control id_num
[ Title { title | From Variable array_name } ]
[ Value value ]
[ { Enable | Disable } ]
[ { Show | Hide } ]
[ Active ]
```

id_num – целочисленный идентификатор одного из элементов активного диалога;

title – новый заголовок для элемента диалога, строковая величина;

array_name – имя или массив величин, используемый для элементов типа ListBox, MultiListBox, RadioGroup и PopupMenu;

value – новое значение для элемента диалога.

Предупреждение!

Этот оператор нельзя запускать из окна **MapBasic**.

Описание

Оператор **Alter Control** изменяет атрибуты элемента активного диалога, окно которого было открыто оператором Dialog; применение оператора **Alter Control** возможно только пока диалоговое окно открыто, т. е. в специальной подпрограмме, называемой процедурой-обработчиком элемента диалога, вызов которой учитывается при создании диалога.

Если на экране находятся два или более диалоговых окон, то оператор **Alter Control** воздействует на активное окно, которое лежит поверх остальных.

Параметр *id_num* определяет элемент диалога, который будет изменяться. Значение параметра соответствует значению параметра *id_num*, заданного в предложении **ID** в операторе **Dialog**.

Изменение состояния и атрибутов элемента диалога производится при помощи предложений **Title**, **Value**, **Enable/Disable**, **Hide/Show**, **Active**. Оператор может использовать либо одно из этих предложений, либо одновременно несколько, либо все. То есть одновременно оператор **Alter Control** может изменить имя, значение и режим доступности элемента диалога.

Однако, не все атрибуты могут меняться для каждого типа элемента диалога. Например, кнопка типа **Button** может быть включена и выключена, но не иметь атрибута.

Предложение **Title** назначает текст для большинства элементов (исключение составляют элементы типа **Picker** и **EditText**; текст элемента **EditText** определяется значением через предложение **Value**). Если меняется текстовый атрибут для элементов типа **ListBox**, **MultiListBox** или **PopupMenu**, то предложение **Title** может читать новое содержимое элемента из строкового массива переменных, если оно дополнено ключевым словом **From Variable**.

Ключевое слово **Active** используется только для элемента **EditText**. Оператор **Alter Control...Active** помещает курсор в текстовое окошко элемента.

Ключевое слово **Hide** прячет элемент диалога, оставляя пустое место. Показать вновь элемент можно, используя ключевое слово **Show**.

Для полной отмены выбора в списке элемента **MultiListBox** определите значение элемента (параметр *value*) равным нулю. Для того, чтобы добавить к текущему выбору в списке элемента **MultiListBox** еще одну строку, выполните оператор **Alter Control** с положительным значением, соответствующим номеру строки в списке.

Примечание: В этом случае не нужно выполнять оператор **Alter Control** внутри обработчика элемента **MultiListBox**.

Можно использовать оператор **Alter Control** для изменения текста, появляющегося в элементе **StaticText**. Однако, MapInfo Pro не может увеличить величину элемента **StaticText** после того как он был создан. Поэтому, если планируете поменять текст элемента **StaticText**, добавьте к нему при определении пробелы. Например оператор **Dialog** может включать следующее предложение:

```
Control StaticText ID 1 Title "Message goes here" + Space$(30)
```

Пример

Следующая программа создает диалог с двумя флажками и кнопками **ОК** и **Отмена**. Когда диалог открывается, кнопка **ОК** не активна (окрашена серым). Кнопка **ОК** становится доступной пользователю, когда он установит один или оба флажка.

```

Include "mapbasic.def"
Declare Sub Main
Declare Sub checker
Sub Main
  Dim browse_it, map_it As Logical
  Dialog
    Title "Display a file"
    Control CheckBox
      Title "Display in a Browse window"
      Value 0
      Calling checker
      ID 1
      Into browse_it
    Control CheckBox
      Title "Display in a Map window"
      Value 0
      Calling checker
      ID 2
      Into map_it
    Control CancelButton
    Control OKButton
      ID 3
      Disable
  If CommandInfo(CMD_INFO_DLG_OK) Then
    ' ... then the user clicked OK...
  End If
End Sub
Sub checker
  ' If either check box is checked,
  ' enable the OK button; otherwise, Disable it.
  If ReadControlValue(1) Or ReadControlValue(2) Then
    Alter Control 3 Enable
  Else
    Alter Control 3 Disable
  End If
End Sub
End Sub

```

Оператор Alter Designer Frame

Назначение

Оператор **Alter Designer Frame** изменяет позицию, заголовки и стили легенды, которую создает **оператор Create Designer Legend**. (Для изменения размера, позиции или заголовка окна **Легенда**, используйте **оператор Set Window**.) Оператор не работает с рамками тематических карт. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Примечание: Этот оператор не позволяет изменить параметры Title, Subtitle и Fonts для тематического раздела в окне **Легенда**. Эти изменения можно внести, используя **оператор Set Legend**.

Примечание: Ширина рамок разделов окна **Легенда** всегда задаётся автоматически по содержимому, поэтому нельзя использовать, чтобы задавать параметр ширины **Width** раздела окна **Легенда**.

Синтаксис

```
Alter Designer Frame
[ Window legend_or_layout_window_id ]
Id { frame_id }
[ Position ( x, y ) [ Units paper_units ] ]
[ Title [ frame_title ] [ Font... ] ]
[ SubTitle [ frame_subtitle ] [ Font... ] ]
[ Columns number_of_columns ] |
[ Height frame_height [ Units paper_units ] ]
[ Width frame_width [ Units paper_units ] ]
[ Name frame_name ]
[ Region [ Height region_height [ Units paper_units ] ] ]
[ Region [ Width region_width [ Units paper_units ] ] ]
[ Line [ Width line_width [ Units paper_units ] ] ]
[ Auto Font Size { On | Off } ]
[ Bar [ Thickness t_value ] [ Length l_value ]
[ Units paper_units ] [ Horizontal | Vertical ] ]
[ Order { Default | Ascending | Descending |
{ Custom id | id : id [ , id | id : id ... ] ... } } ]
[ Style [ Font... ] [ Count { On | Off } ]
[ ID { id } Text { style_name } ]
[ Line Pen... | Region Pen... Brush... | Symbol Symbol... ] ]
[ , ... ]
[ Angle frame_angle ]
[ Metadata SetKey keyname To keyvalue ]
```

legend_or_layout_window_id — целочисленный идентификатор окна **Легенда** или **Отчёт**, который можно получить с помощью **функции FrontWindow()** и **функции WindowID()**.

frame_id — идентификатор рамки или раздела. Нельзя использовать имя слоя. Например, три раздела легенды могут иметь ID-номера 1, 2 и 3.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "cm" для сантиметров). "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

frame_title — строковая переменная, определяющая заголовок раздела легенды.

frame_subtitle — строковая переменная, определяющая подзаголовок раздела легенды.

number_of_columns — количество колонок, которые будут показаны в разделе.

frame_height — используется для запоминания высоты рамки легенды, изначально задаваемой предложением Column. Представляет высоту рамки в "бумажных" единицах. Значение записывается в WOR-файл, если размер рамки был изменен вручную.

frame_width — ширина рамки или раздела в "бумажных" единицах. Значение записывается в WOR-файл, если размер рамки был изменен вручную. Параметр *width* нельзя задавать для раздела **легенды**.

frame_name — строка, определяющая название рамки в окне **Отчёт**. Параметр *frame_name* нельзя задавать для раздела **легенды**. Если рамке окна **Отчёт** присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

region_height — новая высота образца для картографического раздела окна **Легенда** или рамки окна **Отчёт**. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров. Если значение не задано, то используется значение 32 пункта (может быть задано как стандартное).

region_width — новая ширина образца для картографического раздела окна **Легенда** или рамки окна **Отчёт**. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров. Если значение не задано, то используется значение 32 пункта (может быть задано как стандартное).

line_width — новая ширина линий для картографического раздела окна **Легенда** или рамки окна **Отчёт**. Можно задавать от 12 до 144 пунктов, от 1 до 12 пик, от 0.666667 до 2 дюймов, от 4.23333 до 50.8 миллиметров или от 4.23333 до 50.8 сантиметров. Если значение не задано, то используется значение 36 пункта (может быть задано как стандартное).

t_value — толщина рамки вокруг растровой легенды.

l_value — длина рамки вокруг растровой легенды.

id — положение внутри списка стилей для данного раздела. Количество стилей в рамке может вам вернуть **функция LegendFrameInfo()** с атрибутом FRAME_NUM_STYLES (13).

style_name — это строковая величина, которая отображает следующий за каждым символ для раздела с указанным ID-номером. Символ “#” будет замещаться именем текущего слоя. Символ % будет замещаться словами “Line”, “Point”, “Region”, которые будут соответствовать типу географических объектов (эти названия могут быть другими в русской версии). Например, “% of #” будет заменено на “Region of States” для раздела легенды, соответствующей слою таблицы STATES.TAB (текст может быть другим в русской версии).

frame_angle — вещественное значение типа Float: возвращает угол поворота рамки с текстом или изображением (в градусах) Для рамок с символами оно может применяться для изменения поворота символов TrueType (в градусах).

key_name — строковая величина, представляющая собой имя ключа метаданных.

Keyvalue — строка, представляющая значение, которое требуется присвоить ключу.

Описание

Если опустить параметр **Window**, то оператор будет обрабатывать либо самое верхнее окно **Легенда**, либо окно **Отчёт**, в зависимости от ситуации.

Предложение **Position** контролирует положение раздела в окне **Легенда** или **Отчёт**. Верхний левый угол окна имеет положение 0, 0. Значения положения используют параметры в "бумажных" единицах, например, "in" (дюймы) или "cm" (сантиметры). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя **оператор Set Paper Units**. Оператор **Alter Designer Frame** может изменить единицы измерения с помощью предложения **Units** в предложении **Position**.

В предложения **Title** и **SubTitle** можно вводить новый текст, новый шрифт или и то, и другое.

Предложение **Columns** определяет количество колонок, которые будут показаны в рамке.

Рамки и разделы всех типов имеют минимальные и максимальные значения параметров высоты и ширины — **Height** и **Width**. Если задать значение параметра *frame_width* или *frame_height* вне доступного диапазона, появится сообщение об ошибке, в котором будет указан разрешённый диапазон.

- В случае рамки с изображением всегда используется соотношение сторон. Например, для изображения, имеющего отношение сторон равное 0.5

```
Alter Designer Frame FrontWindow() ID 1 Width 100 'Height will be set to 50
```

```
Alter Designer Frame FrontWindow() ID 1 Height 100 'Width will be set to 200
```

Если параметры высоты и широты — **Width** and **Height** задаются в одном и том же операторе, то последний используемый параметр имеет приоритет. В следующем примере высота задаётся равной 100, и ширина – 200, чтобы удовлетворить соотношению сторон.

```
Alter Designer Frame FrontWindow() ID 1 Width 50 Height 100
```

- В случае рамки, содержащей линейные фигуры, если **Width** задан нулём, то линия будет вертикальной. Задайте параметр **Height** равным нулю для горизонтальной линии. Если оба параметра заданы нулевыми, появится сообщение об ошибке.

Примечание: Параметр **Width** нельзя использовать, чтобы задавать для ширину раздела **легенды**.

Предложение **Region Height** определяет высоту раздела с образцами в окне **Легенда**.

Предложение **Region Width** определяет ширину раздела с образцами в окне **Легенда**.

Предложение **Line Width** определяет ширину образца линии в разделе образцов в окне **Легенда**.

Auto Font Size контролирует режим подстройки размера шрифтов в окне образцов при изменении размера этого окна.

Предложение **Bar** позволяет менять ширину или длину полосы после её создания. Оно должно включать хотя бы одно из следующего: **Thickness**, **Length** или ориентацию — (**Horizontal** | **Vertical**). Предложения **Units**, задающее единицы измерения, является дополнительным, необязательным — по умолчанию MapInfo Pro автоматически применяет дюймы. Если добавить только выражение **Units**, опустив другие параметры, то это приведёт к ошибке при выполнении (ошибка некорректного выражения). Значения параметров **Thickness** и **Length** могут быть в диапазоне от 0.1 до 72.0 дюймов. MapInfo Pro пропускает значения, выходящие за пределы диапазона других размеров условных знаков (и усекает полосу, чтобы она отображалась в пределах диапазона).

С помощью параметра **Order** добавлена возможность либо включать сортировку, либо изменять способ сортировки рядов разделов легенд. Легенды карты можно сортировать либо по стилям, либо задавая индивидуальный порядок сортировки. Сортировать можно по следующим значениям параметра сортировки: **Default**, **Ascending**, **Descending** или **Custom**. Параметр **Order** записывается в рабочем наборе, только если легенда карты отсортирована в возрастающем, убывающем или индивидуальном порядке.

Порядок сортировки по умолчанию **Default** — это порядок сортировки, который устанавливает **Конструктор легенды**. Если легенда создаётся по уникальным стилям оформления объектов карты, то ряды условных знаков раздела будут перечислены в порядке, в котором объекты используются на карте.

Если задан индивидуальный порядок сортировки условных знаков раздела легенды карты — **Custom**, то значение идентификатора *id* начинается с 1, увеличивается сверху вниз. Если проверить порядок перечисления условных знаков раздела в диалоге **Свойства раздела легенды**, то первый условный знак имеет идентификатор равный 1, номера следующих

увеличиваются по мере перечисления в списке. Дополнительная информация содержится в разделе: [Настройка порядка сортировки](#).

Параметр **Display** управляет способом показа каждого условного знака. Каждый условный знак в MapBasic-команде начинается со специального выражения **Style**. В конце каждого выражения **Style** может применяться дополнительный оператор **Display**. **Display On** делает ряд видимым, а **Display Off** — невидимым. Если параметр **Display** отсутствует, то ряд с условным знаком видим. Параметр **Display** записывается в рабочем наборе только для невидимых в разделе легенды условных знаков.

Примечание: Параметры **Columns** и **Height** можно применять и к тематическим, и к картографическим легендам. Однако, все свойства тематических легенд должны быть заданы с помощью [оператора Set Legend](#).

В качестве единиц **Units** используются cm (сантиметры), mm (миллиметры), in (дюймы), pt (точки), а также пикс. Единицы преобразуются друг в друга по следующим правилам:

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Предложение **Style** должно содержать список определений для стилей, отображающихся в разделе. Можно обновить стиль для раздела легенды, созданной с использованием ключевого слова **NoRefresh**. Можно обновлять текст для любого стиля. Нет возможности добавлять или удалять стили для любых типов разделов легенды.

Параметр счётчика **Count { On | Off }** применяется позволяет управлять показом значения количества записей счётчика строк растровой легенды. По умолчанию этот режим выключен — **Off**. Это применяется только для разделов тематических, классифицированных растровых легенд. В легендах непрерывных растров счётчики не отображаются, а этот параметр будет пропускаться. MapInfo Pro сохраняет параметр счётчика **Count** в WOR-файле. При работе с WOR-файлом MapInfo Pro отображает счётчик, только если его параметры заданы.

В предложении **Count** сохраняется значение определённой строки раздела растровой легенды. Его не следует изменять. Значения являются 64-разрядными целыми в диапазоне от нуля (0) до 9 223 372 036 854 775 807. Если задано некорректное значение, то никакое значение не передаётся, поэтому запрос с помощью LEGEND_STYLE_INFO_COUNT возвращает -1. Легенды непрерывных растров не отображают счётчики, смотрите раздел **Count** выше.

Предложение **Metadata SetKey** помогает управлять метаданными, которые хранятся в рамке отчёта.

Некоторые предложения могут быть применены к тематическим легендам. Это:

```
[ Position ( x, y ) [ Units paper_units ] ]
```

```
[ Columns number_of_columns ] |
[ Height frame_height [ Units paper_units ] ]
```

Пример

Изменяется стиль рамки 1 для легенды 1 слоя городов. В примере изменяются заголовок, подзаголовок, и текст, показанный рядом с символами.

```
Alter Designer Frame
Window FrontWindow()
ID 1
Position (1, .5) Units "in"
Title "Big Cities"
Subtitle "125 Biggest Cities"
Style Font ("Arial",2,12,255)
ID 1 Text "City Points"
```

См. также:

[оператор Add Designer Frame](#), [оператор Create Designer Legend](#), [оператор Remove Designer Frame](#), [оператор Set Designer Legend](#)

Настройка порядка сортировки

Оба [оператора Create Designer](#) и [Alter Designer Frame](#) теперь содержат параметр Order, с помощью которого можно задавать индивидуальный порядок сортировки – Custom.

```
Custom id | id : id [ , id | id : id ... ]
```

Где диапазон идентификаторов рядов условных знаков задаётся в возрастающем порядке (Id2 > Id1):

```
Id1 : Id2
```

Изменения порядка перечисления списка

Синтаксис настройка порядка сортировки такой же, как и в предложении **Order**, которое описано в разделе об [операторе Set Map](#). Этот параметр довольно легко использовать, когда требуется изменить порядок перечисления в начале списка, но возникают сложности, если требуется изменить порядок перечисления ближе к концу списка. Например, если требуется изменить порядок перечисления трех используемых в легенде рядов условных знаков на обратный:

```
Order Custom 3, 2, 1
```

Остальные ряды условных знаков можно не трогать. Если имеется 10 рядов условных знаков, а требуется изменить порядок перечисления двух последних, то придется перечислять идентификаторы в таком порядке:

```
Order Custom 1, 2, 3, 4, 5, 6, 7, 8, 10, 9
```

Для того чтобы сократить запись, можно использовать следующий синтаксис:

```
Order Custom 1:8, 10, 9
```

Разметка диапазонов значений

С помощью двоеточия (:) вы можете размечать диапазоны значений:

```
Long form:
Order Custom 2, 5, 6, 7, 8, 9, 10, 1, 3, 4

Short form:
Order Custom 2, 5:10, 1, 3, 4
Order Custom 2, 5:10, 1, 3:4 (same as above but also valid)
Order Custom 2, 5:10, 1 (same as above but also valid)
```

```
Long form:
Order Custom 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18,
19, 20, 11

Short form:
Order Custom 1:10, 12:20, 11
```

В списке не должно быть повторений – они порождают ошибки. Следующая запись ошибочна:

```
Order Custom 1:5, 8, 4:7
```

потому что в ней повторяются значения 4 и 5. Чтобы это увидеть, перепишем список полностью:

```
Order Custom 1, 2, 3, 4, 5, 8, 4, 5, 6, 7
```

Альтернативный синтаксис можно использовать при создании и изменении легенды в окне **Легенда**. В рабочих наборах короткий вариант синтаксиса используется, когда разделы в окне **Легенда** содержат более 50 рядов условных знаков, перечисленных в индивидуальном порядке.

Оператор Alter Designer Text

Оператор **Alter Designer Text** позволяет изменять строку текста, шрифт и положение рамки с текстом в окне **Легенда** или **Отчёт**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

Alter Designer Text

```
[ Window legend_or_layout_window_id ]
[ ID textframe_id [ Text { frame_text [ Font... ] }
  [ Position ( x, y ) [ Units paper_units ] ] ] ]
```

Legend_or_layout_window_id – целочисленный идентификатор окна **Легенда** или окна **Отчёт**, который можно получить с помощью **функции FrontWindow()** и **функции WindowID()**.

textframe_id – уникальный номер текстового раздела (т.е. не раздела легенды) в окне **Легенда**.

frame_text – текст для заголовка, подзаголовка легенды, либо обычный текст (например, сообщение об авторских правах).

x – определяет требуемое расстояние от верхнего края рабочего стола MapInfo до верхнего края окна.

y – определяет требуемое расстояние от левого края рабочего стола MapInfo до левого края окна.

Примечание: Здесь под рабочим столом подразумевается область с данными на экране (без строк заголовка и состояния программы и панелей инструментов).

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также циперо.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Описание

Если в предложении **Text** не задан параметр шрифта **Font**, то используется стандартный шрифт.

Предложение **Position** контролирует положение раздела в окне **Легенда** или **Отчёт**. Верхний левый угол окна легенды имеет позицию 0, 0. **Position** использует "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры), см. **оператор Set Paper Units**. MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя **оператор Set Paper Units**. Вы можете задать свои единицы измерения, явно задав предложение **Units** в предложение **Position**.

Если опустить параметр **Window**, то оператор будет обрабатывать либо самое верхнее открытое окно, либо самое верхнее окно **Легенда**, либо самое верхнее окно **Отчёт**, в зависимости от ситуации.

Пример

```
Alter Designer Text Window frontwindow()
  ID 1
    Text "Title Changed" Font("Arial Greek", 0, 14, 14680064)
```

См. также:

[оператор Create Designer Legend](#), [оператор Add Designer Text](#), [оператор Remove Designer Text](#)

Оператор Alter MapInfoDialog

Назначение

Делает недоступными, прячет или присваивает значение элементу стандартного диалогового окна в MapInfo Pro.

Ограничения!

Осторожно: Оператор **Alter MapInfoDialog** может не поддерживаться в будущих версиях MapInfo Pro. В результате, программы MapBasic, которые используют этот оператор, могут не работать корректно в следующих версиях MapInfo Pro. Используйте этот оператор с осторожностью.

Синтаксис 1 (присвоение нестандартных установок)

```
Alter MapInfoDialog dialog_ID
  Control control_ID
    { Disable | Hide | Value new_value } [ , { Disable... } ]
  [ Control... ]
```

Синтаксис 2 (восстановление стандартных установок)

```
Alter MapInfoDialog dialog_ID Default
```

dialog_ID – целое число, идентификатор изменяемого диалогового окна MapInfo Pro;

control_ID – целое число от 1 и более, идентификатор изменяемого элемента диалога;

new_value – новое значение элемента диалога.

Описание

Оператор позволяет делать недоступными, скрытыми и присваивать значения элементам стандартных диалогов MapInfo Pro – кнопкам, флажкам и т. п.

Примечание: Используйте этот оператор только для изменения стандартных диалогов MapInfo Pro. Пользуйтесь оператором **Dialog** для работы с элементами диалогов, которые создает **оператор Alter Control**.

Определение идентификатора ID

Для определения идентификатора диалога необходимо загрузить MapInfo Pro командой:

```
mapinfopro.exe -helpdiag
```

После того, как программа MapInfo Pro будет запущена с аргументом *-helpdiag*, выведите нужное диалоговое окно на экран и нажмите на кнопку Справки в нем. Обычно, эта кнопка вызывает окно Справочной системы, но в случае использования аргумента *-helpdiag*, MapInfo Pro показывает идентификатор данного диалога.

Примечание: Существуют разные системные диалоги (такие как **Открыть** и **Сохранить**) в разных версиях Windows. Если требуется внести изменения в системный диалог, и если приложение используется в разных версиях Windows, то необходимо приготовить две версии оператора **Alter MapInfoDialog** по одному для каждой версии Windows.

Каждый элемент диалога также имеет идентификатор. Например, большинство кнопок ОК имеют идентификатор 1, а кнопка Отмена (Cancel) - идентификатор 2. Определить идентификатор элемента диалога можно только с помощью программы, не входящей в состав пакета MapInfo, такой, как программа Microsoft Spy++ из пакета компилятора С. В пакет MapBasic программа Spy++ не входит.

Измененное состояние элементов диалога и значения, присвоенные оператором **Alter MapInfoDialog**, сохраняются, только если пользователь закрыл диалог с подтверждением **ОК**. Например оператор **Alter MapInfoDialog** можно использовать для помещения адреса в диалог **Найти**, но MapInfo Pro не выполнит поиск, пока не откроется диалог и пользователь не нажмет на кнопку **ОК**.

Возможные виды изменений

Ключевое слово **Disable** используется для того, чтобы сделать элемент недоступным для пользователя (в диалоге элемент закрашивается серым цветом).

Ключевое слово **Hide** прячет элемент, делает его невидимым.

Ключевое слово **Value** изменяет значение элемента.

Если вносите изменения в системный диалог (например, **Открыть** таблицу), то можете изменить выбор в поле со списком (combo box), или изменить текст подписи (static text), кнопки или в текстового окошка.

Изменять положение элементов можно в диалоге **Настройка печати**. Кнопкам Книжная и Альбомная соответствуют числа 1056 и 1057 соответственно.

Если вы изменяете диалог MapInfo Pro, то следующий список приводит виды изменений, которые можно в нём сделать:

- **Кнопка, подпись (static text), текстовое окошко, окошко со списком (editable combo box)**: можно менять текст, задавая новый текст параметром *new_value*.
- **Список (list box), окошко со списком (combo box)**: можно менять выбор в списке, задавая номер элемента списка в параметре *new_value*.
- **Флажок**: можно устанавливать (значение равно 1) или сбрасывать (значение равно 0).
- **Переключатель**: можно менять выбор кнопки, задавая значение 0 (не выбрана) или 1 (выбрана).
- **Кнопка стиля символа**: можно менять установку стиля символа, например, для этого используется [функция MakeSymbol\(\)](#).
- **Кнопка стиля линии**: можно менять установку стиля линии.
- **Кнопка стиля штриха**: можно менять установку стиля заливки.
- **Кнопка стиля шрифта**: можно менять установку стиля заливки.
- **Кнопка стиля оформления области (линия/штрих)**: можно задать новое значение стиля линии или стиля штриха. (Пример использования этого элемента можно видеть в диалоге **Стиль области** MapInfo Pro, который открывается если дважды щелкнуть мышкой по изменяемому региону.)

Пример

Следующий фрагмент вносит в диалог **Найти** текстовую строку с адресом ("23 Main St.") в окошко первого окна и прячет кнопку Снова.

```
If SystemInfo(SYS_INFO_MIVERSION) = 400 Then
  Alter MapInfoDialog 2202
    Control 5 Value "23 Main St."
    Control 12 Hide
End If
Run Menu Command M_ANALYZE_FIND
```

Идентификатор окна **Найти** – 2202. Control 5 ссылается на текстовое поле, в которое вводится адрес для поиска. Control 12 ссылается на кнопку Снова. Все номера идентификаторов в будущих версиях MapInfo Pro могут быть изменены. Поэтому используется [функция SystemInfo\(\)](#) для определения версии программы, в которой выполняется это приложение.

См. также:

[оператор Alter Control](#), [функция SystemInfo\(\)](#)

Оператор Alter Menu

Назначение

Добавляет элемент в список одного из существующих меню в окне MapInfo или убирает существующий элемент из списка меню.

Синтаксис 1

```
Alter Menu { menuname | ID menu_id }
Add menundef [ , menundef... ]
```

Где *menundef* – идентификатор элемента списка меню, который имеет следующий синтаксис:

```
newmenuitem
[ ID menu_item_id ]
[ HelpMsg help ]
[ { Calling handler | As menuname } ]
```

menuname – имя существующего меню (например, "File").

menu_id – целочисленный идентификатор меню от 1 до 64, где единица представляет меню File.

newmenuitem – строка: имя, которое будет добавлено к указанному меню;

menu_item_id – целочисленный идентификатор элемента меню, который в дальнейшем может использовать [оператор Alter Menu Item](#).

help – строковая величина, текст которой будет показываться в строке сообщений, когда элемент меню будет выбран (подсвечен);

handler – имя sub-процедуры обработчика или командный код, имена для которых определены в файле MENU.DEF (например, M_FILE_NEW) или же специальный код обработки события выбора меню механизмами OLE или DDE. Если задать код команды для стандартной команды MapInfo Pro типа "Показать/Скрыть" (например, M_WINDOW_STATISTICS), строка *newmenuitem* должна начинаться с восклицательного знака и включать в себя символ (^), чтобы отделять режимы показа и скрытия. Особенности разных способов обработки см. в разделе: [Оператор Create Menu](#).

Синтаксис 2

```
Alter Menu { menuname | ID menu_id }
Remove { handler | submenuname | ID menu_item_id }
[ , { handler | submenuname | ID menu_item_id } ... ]
```

menuname – заголовок уже определенного меню;

menu_id – целочисленный идентификатор меню от 1 до 64, где единица представляет меню File;

handler – имя sub-процедуры или код для стандартной команды MapInfo Pro;

submenuname – заголовок удаляемого подменю, иерархически подчиненного заданному меню;

menu_item_id – целочисленный идентификатор элемента меню.

Полезно учитывать при использовании этой команды в 64-разрядной версии MapInfo Pro:

1. Если одно и тоже меню добавлено в двух разных местах, то с помощью этой команды изменять можно только последнее добавленное.
2. Если любое заранее заданное меню MapInfo Pro добавлено в раскрывающийся список, то с помощью этой команды можно изменить только тот элемент меню, который находится в группе меню.

Описание

Оператор **Alter Menu** добавляет или убирает элемент в списке меню (или подменю).

Элемент меню *menuname* в операторе может быть идентифицирован его именем (например, "File" или "Файл") .

Если требуется заменить одно из стандартных меню MapInfo Pro, задавайте в операторе **Alter Menu** его номер **ID**. Параметр **ID** принимает значения от 1 до 64. Ниже приводится таблица с именами стандартной системы меню MapInfo Pro и соответствующими им значениями идентификатора ID.

Таблица 1. ID-номер меню

Имя меню	Код	ID	Описание
Файл	M_FILE	1	Стандартное меню Файл.
Правка	M_EDIT	2	Стандартное меню Правка.
Найти	M_SEARCH	3	Стандартное меню Поиск
Запрос	M_QUERY	3	Стандартное меню Запрос.
Программы	M_PGM	4	Меню программ.
Инструменты	M_TOOLS	4	Стандартное меню Программы.

Имя меню	Код	ID	Описание
Настройки	M_OPTIONS	5	Стандартное меню Настройки.
Окно	M_WINDOW	6	Стандартное меню Окно.
Справка	M_HELP	7	Стандартное меню Справка.
Список	M_BROWSE	8	Меню Список Используется, когда активно окно списка.
Карта	M_MAP	9	Меню Карта. Используется, когда активно окно карты.
Отчёт	M_LAYOUT	10	Меню Отчёт. Используется, когда активно окно Отчёта.
График	M_GRAPH	11	Меню График. Используется, когда активно окно Графика.
MapBasic	M_MAPBASIC	12	Меню MapBasic. Используется, когда активно окно MapBasic .
Районирование	M_REDISTRIBUTE	13	Меню Районирование. Используется, когда активно окно Районирование.
Объекты	M_OBJECTS	14	Стандартное меню Объекты.
Таблица	M_TABLE	15	Стандартное меню Таблица.
Открыть Web-службу	M_OPEN_WEBSERVICE	39	Откройте пункт меню Web-службы в меню Файл.
Найти выборку	M_FIND_SELECTION	40	Пункт меню Выбрать в меню Запрос.
Рабочая область Oracle	M_ORACLE_TOOLS	41	Пункт меню Рабочая область Oracle в меню Таблица.
Обслуживание	M_TAB_MAINTENANCE	42	Пункт меню Изменить в меню Таблица.
Растр	M_TAB_RASTER	43	Пункт меню Растр в меню Таблица.
Лицензирование	M_LICENSING	44	Пункт меню Лицензирование в меню Справка.
Карты сервера тайлов	M_TILESERVER	45	Раздел меню Сервер в меню Файл.

Имя меню	Код	ID	Описание
Конструктор легенды	M_LEGEND_DESIGNER	49	Меню Легенда .
Конструктор отчёта	M_LAYOUT_DESIGNER	51	Меню Отчёт .

Далее перечислены всплывающие, быстрые меню, которые появляются при нажатии на правую кнопку мыши.

Таблица 2. ID-номера для быстрых меню

Имя меню	Код	ID	Описание
DefaultShortcut	M_SHORTCUT_DFLT	16	Стандартное всплывающее, быстрое меню. Это меню открывается при нажатии на правую кнопку мыши в окнах, не имеющих собственных всплывающих, быстрых меню.
MapperShortcut	M_SHORTCUT_MAPPER	17	Краткое, раскрывающееся меню окна карты.
BrowserShortcut	M_SHORTCUT_BROWSER	18	Краткое, раскрывающееся меню окна списка.
LayoutShortcut	M_SHORTCUT_LAYOUT	19	Краткое, раскрывающееся меню окна отчёта.
GrapherShortcut	M_SHORTCUT_GRAPHER	20	Краткое, раскрывающееся меню окна графика. Меню содержит команды создания графиков.
CmdShortcut	M_SHORTCUT_CMD	21	Краткое, раскрывающееся меню окна MapBasic .
RedistrictShortcut	M_SHORTCUT_REDISTRICTER	22	Краткое, раскрывающееся меню окна районирования; открывается для окна списка районов.
LegendShortcut	M_SHORTCUT_LEGEND	23	Краткое, раскрывающееся меню окна легенды.

Имя меню	Код	ID	Описание
GrapherShortcut	M_SHORTCUT_GRAPHTDG	24	Краткое, раскрывающееся меню окна графика. Меню содержит команды форматирования и изменения уже имеющегося графика.
3DMapShortcut – меню 3D-карты	M_SHORTCUT_3DMAP	25	Краткое, раскрывающееся меню окна трехмерной карты.
MessageWinShortcut	M_SHORTCUT_MSG_WIN	26	Краткое, раскрывающееся меню для окна сообщений.
StatisticsWinShortcut	M_SHORTCUT_STAT_WIN	27	Краткое, раскрывающееся меню для окна статистики.
AdornmentShortcut	M_SHORTCUT_ADORNMENT	32	Краткое, раскрывающееся меню для окна элементов оформления.
LcLayersShortcut	M_SHORTCUT_LC_LAYERS	33	Краткое, раскрывающееся меню окна управления слоями, которое появляется, если нажать правую клавишу мыши на обыкновенном слое в списке слоёв.
LcMapsShortcut	M_SHORTCUT_LC_MAPS	34	Краткое, всплывающее меню окна управления слоями, которое появляется, если нажать правую клавишу мыши на иконке карты в списке слоёв.
LcGroupsShortcut	M_SHORTCUT_LC_GROUPS	35	Краткое, всплывающее меню окна управления слоями, которое появляется, если нажать правую клавишу мыши на группе слоёв в списке слоёв.
TableAdornmentShortcut	M_SHORTCUT_TABLEADORNMENT	36	Зарезервировано для будущего использования.
TableListWindowShortcut	M_SHORTCUT_TLV_TABLES	37	Краткое, всплывающее меню окна списка таблиц, которое появляется, если нажать правую клавишу мыши на имени таблицы.

Имя меню	Код	ID	Описание
OverridesShortcut	M_SHORTCUT_LC_OVERRIDES	38	Всплывающее меню окна управления слоями, которое появляется, если нажать правую клавишу мыши на в области элементов управления Масштабный диапазон для подписей или Масштабный диапазон для стиля объектов.
BrowserFilterShortcut	M_SHORTCUT_BROWSER_SORTFILTER	47	Краткое, всплывающее меню окна списка, которое появляется, если нажать правую клавишу мыши на кнопке Сортировать/Фильтр.
BrowserSelectShortcut	M_SHORTCUT_BROWSER_SELECT	48	Краткое, всплывающее меню окна списка, которое появляется, если нажать правую клавишу мыши на кнопке Выбрать.
LegendDesignerShortcut	M_SHORTCUT_LEGEND_DESIGNER	50	Окно конструктора легенд

Таблица 3. Прочие ID-номера для меню

Имя меню	Код	ID	Описание
3DWindow – меню 3D-карты	M_3DMAP	28	Окно 3D-карты.
График	M_GRAPHTDG	29	Меню График.
Легенда	M_LEGEND	31	Меню Легенда.

При изменении пользовательского меню (даже если вы создаете его, задавая номер Custom ID, такой как 999), вам придется обращаться к нему через имя Custom *menuname*, а не номер Custom *ID*.

Примеры

Добавим в список меню Файл новую команду:

```
Alter Menu "File" Add
"Special" Calling sub_procedure_name
```

Следующий фрагмент делает то же самое, что и предыдущий пример, только используется идентификатор меню.

```
Alter Menu ID 1 Add
"Special" Calling sub_procedure_name
```

Теперь добавим значение идентификатора для команды **ID**, равный 300, который позже сможет использовать **оператор Alter Menu Item**.

```
Alter Menu ID 1 Add
"Special" ID 300 Calling sub_procedure_name
```

В следующем примере удаляется ранее созданный элемент из меню Файл.

```
Alter Menu ID 1 Remove sub_procedure_name
```

Программа, TextBox, использует **оператор Create Menu**, чтобы создать меню под названием "TextBox", а затем с помощью оператора **Alter Menu** добавляет меню TextBox в виде иерархического меню, в меню Программы:

```
Alter Menu "Tools" Add
"(-",
"TextBox" As "TextBox"
```

В следующем примере добавляется новая команда в быстрое меню для окна карты (это меню появляется при нажатии на правую кнопку мыши в окне карты).

```
Alter Menu ID 17 Add
"Find Nearest Site" Calling sub_procedure_name
```

См. также:

оператор Alter Menu Bar, оператор Alter Menu Item, оператор Create Menu, оператор Create Menu Bar

Оператор Alter Menu Bar

Назначение

Добавляет или удаляет заголовки меню в строку меню окна MapInfo.

Синтаксис

```
Alter Menu Bar { Add | Remove }
  { menuname | ID menu_id }
  [ , { menuname | ID menu_id } ... ]
```

menuname – имя меню (например, "Файл");

menu_id – целочисленный идентификатор меню от 1 до 22, где единица представляет меню Файл.

Полезно учитывать при использовании этой команды в 64-разрядной версии MapInfo Pro:

1. В 64-разрядной версии MapInfo Pro эта команда добавляет или удаляет пункт меню из группы меню на вкладке **ИНСТРУМЕНТЫ**.
2. Подпункты меню будут рассматриваться как раскрывающиеся списки в группе меню:
 - a. Открыть Web-службу
 - b. Найти выборку
 - c. Рабочая область Oracle
 - d. Обслуживание
 - e. Растр
 - f. Лицензирование
 - g. Карты серверов тайлов

Описание

Оператор **Alter Menu Bar** добавляет или убирает один или более заголовков меню в строку меню.

Параметр *menuname* может быть строкой или выражением строкового типа, результатом которого является имя одного из стандартных меню в MapInfo (например, "Файл" или "Правка").

Параметр *menuname* может быть также именем меню, который создает **оператор Create Menu** (см. примеры).

Примечание: Поэтому, если планируете использовать прикладную программу в MapInfo другой языковой версии, используйте в операторе **Alter Menu** вместо имени номер, задаваемый предложением ID. Каждое из стандартных меню MapInfo Pro (Файл, Правка и т.п.) имеет также уникальный ID-номер, который можно использовать независимо от того, как переведен заголовок меню. Например, используя ID 2 мы всегда обращаемся к меню Edit или Правка.

Список имен стандартной системы меню MapInfo Pro и соответствующих им значений идентификаторов приведен в разделе: **Оператор Alter Menu**.

Как добавить меню

Оператор **Alter Menu Bar Add** добавляет заголовок меню в строку меню с правого края. Если требуется вставить меню в определенное место в строке меню, то для переопределения строки меню выполните **оператор Create Menu Bar**.

В Windows, если добавить много заголовков меню в строку меню, то в ней появляется вторая строчка.

Как убрать меню

Оператор **Alter Menu Bar Remove...** удаляет заголовок из строки меню. При этом меню не пропадает и может быть в любой момент восстановлено так, как показано ниже. Следующая пара операторов сначала удаляет меню Запрос из строки меню, а затем помещает его снова туда крайним справа:

```
Alter Menu Bar Remove "Query"
Alter Menu Bar Add "Query"
```

После того, как оператор **Alter Menu Bar Remove...** удалит меню, MapInfo Pro отменяет все клавишные сокращения, ранее назначенные командам, которые находились в удаленном списке меню. Например, пользователь MapInfo Pro может нажать **Ctrl+O** и открыть из меню **Файл** диалог **Открыть**; однако, если оператором **Alter Menu Bar Remove** меню **Файл** было удалено, то MapInfo Pro проигнорирует нажатие **Ctrl+O**.

Пример

В следующем примере создается меню Данные, затем выполняется оператор **Alter Menu Bar Add** для добавления этого меню в строку меню MapInfo Pro.

```
Declare Sub addsub
Declare Sub editsub
Declare Sub delsub
Create Menu "DataEntry" As
  "Add" Calling addsub,
  "Edit" Calling editsub,
  "Delete" Calling delsub
'Remove the Window menu and Help menu
Alter Menu Bar Remove ID 6, ID 7
'Add the custom menu, then the Window & Help menus
Alter Menu Bar Add "DataEntry", ID 6, ID 7
```

Перед тем как поместить заголовок созданного меню в строку меню, эта программа сначала удаляет меню **Справка** (идентификатор 7) и меню **Окно** (идентификатор 6). Затем в правый конец строки меню добавляется сначала заголовок меню **Данные**, а затем меню **Окно** и **Справка**. Так мы добиваемся того, что заголовки **Окно** и **Справка** всегда будут последними в строке меню.

См. также:

оператор **Alter Menu**, оператор **Alter Menu Item**, оператор **Create Menu**, оператор **Create Menu Bar**, оператор **Menu Bar**

Оператор Alter Menu Item

Назначение

Изменяет состояние элемента списка меню.

Синтаксис

```
Alter Menu Item { handler | ID menu_item_id }
{ [ Check | Uncheck ] |
  [ Enable | Disable ] |
  [ Text itemname ] |
  [ Calling handler | As menuname ] }
```

handler – имя sub-процедуры обработчика или код для стандартной команды MapInfo Pro;

menu_item_id – целочисленный идентификатор элемента *menu_item_id*, который задаётся при создании списка меню (**оператор Create Menu** или **оператор Alter Menu**).

itemname – новый текст для элемента меню (может содержать управляющие коды);

menuname – заголовок уже определенного меню;

Полезно учитывать при использовании этой команды в 64-разрядной версии MapInfo Pro:

1. В 64-разрядной версии MapInfo Pro эта команда используется только для элементов управления, создаваемых пользователем.
2. Эта команда используется только для элементов управления дополнительных программ.
3. Эта команда не оказывает влияния на существующие команды меню.
4. Изменение текста, которое предписывает новое сокращение, вступит в силу только в том случае, если оно не используется каким-либо другим элементом управления приложения.
5. Вызов параметра оператора As Clause не поддерживается.

Описание

Оператор **Alter Menu Item** изменяет значения атрибутов одного или более элементов списка меню. Например, оператор **Alter Menu Item** может сделать команду недоступной для выбора (на экране она закрашивается серым).

Элемент меню может задаваться либо именем обработчика *handler*, который запускается при выборе элемента в списке меню, либо идентификатором в предложении **ID**. Заметим, что

один и тот же обработчик может вызывать разные элементы меню. Поэтому, если оператор **Alter Menu Item** использует имя процедуры-обработчика, то MapInfo Pro будет менять все элементы, вызывающие этот обработчик из всех меню. Если используете предложение **ID**, то MapInfo Pro изменит атрибуты только одного элемента меню.

Оператор **Alter Menu Item** может использовать идентификатор **ID** только для тех элементов, для которых при создании списка меню он был определен. Приложение MapBasic не может использовать идентификатор, который был задан другим приложением MapBasic.

Режимы **Check** и **Uncheck** регулируют наличие или отсутствие флажка – галочки у команды в меню. Заметим, что пункт меню может быть отмеченным галочкой, только если при его определении была задана эта возможность (например, если в **операторе Create Menu** перед именем меню поставлен символ "!").

Режимы **Disable** и **Enable** определяют доступность выбора элемента меню. Недоступные элементы закрашиваются серым цветом. Заметим, что MapInfo Pro автоматически делает некоторые элементы меню доступными и недоступными в соответствии с текущим состоянием среды программы MapInfo Pro. Например, команда **Заккрыть** становится серой, если не открыто ни одной таблицы. Поэтому приложение MapBasic не может изменять доступность стандартного элемента меню MapInfo Pro. Можно обращаться к инструментальным средствам, как к элементам меню (например, `M_TOOLS_RULER` в `MENU.DEF`), но с помощью оператора **Alter Menu Item** нельзя сделать их недоступными.

В предложении **Text** можно изменить имя элемента.

Предложение **Calling** задает имя процедуры-обработчика, вызываемой элементом меню. Если пользователь выберет этот элемент в меню, то MapInfo Pro запустит на выполнение эту процедуру.

Примеры

Создается меню Данные, содержащее четыре команды, и затем добавляется в строку заголовков меню MapInfo.

```
Declare Sub addsub
Declare Sub editsub
Declare Sub delsub
Create Menu "DataEntry" As
  "Add" Calling addsub,
  "Edit" Calling editsub,
  "Delete" ID 100 Calling delsub,
  "Delete All" ID 101 Calling delsub
'Remove the Help menu
Alter Menu Bar Remove ID 7
'Add both the new menu and the Help menu
Alter Menu Bar Add "DataEntry" , ID 7
```

Следующий **оператор Alter Menu Item** переименовывает команду "Правка" в команду "Правка..."

```
Alter Menu Item editsub Text "Edit..."
```

Следующий оператор делает команду "Удалить всё" недоступной.

```
Alter Menu Item ID 101 Disable
```

Следующий оператор делает недоступными две команды: "Удалить все" и "Удалить", так как они используют один и тот же обработчик *delsub*.

```
Alter Menu Item delsub Disable
```

См. также:

[оператор Alter Menu](#), [оператор Alter Menu Bar](#), [оператор Create Menu](#)

Оператор Alter Object

Назначение

Изменяет форму, положение или графический тип существующего объекта. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Alter Object obj
{ Info object_info_code, new_info_value |
  Geography object_geo_code , new_geo_value |
  Node {
    Add [ Position polygon_num, node_num ] ( x, y ) |
    Set Position polygon_num, node_num ( x , y ) |
    Remove Position polygon_num, node_num } }
```

obj – переменная типа Object;

object_info_code – целочисленный код, который возвращает [функция ObjectInfo\(\)](#) (например, OBJ_INFO_PEN).

new_info_value – новое значение для кода *object_info_code* (например, новая величина типа Pen);

object_geo_code – целочисленный код, который возвращает [функция ObjectGeography\(\)](#) (например, OBJ_GEO_POINTX).

new_geo_value – новое значение для кода *object_geo_code* (например, новая X-координата);

polygon_num – короткое целое число, идентификатор для одного полигона в объекте регион (область);

node_num – короткое целое число, идентификатор для одного узла в полилинии или полигоне;

x, y – координаты X и Y узла.

Описание

Оператор **Alter Object** изменяет форму, местоположение и/или графический стиль существующего объекта.

Эффект действия оператора **Alter Object** зависит от того, какое предложение используется в конструкции оператора **Info**, **Node** или **Geography**. Если оператор использует предложение **Info**, то MapBasic изменяет графический стиль оформления объекта (например, стиль линии и штриховки). Если оператор использует предложение **Node**, то MapBasic добавляет, удаляет или передвигает узлы объекта типа "Полилиния" или "Область". Если оператор включает в себя предложение **Geography**, то MapBasic изменяет географические атрибуты всех объектов, не являющихся полилиниями и областями (например, координаты X или Y точечного объекта).

Предложение Info

Оператор **Alter Object** с предложением **Info** изменяет стиль оформления объекта, например, стиль линии и штриховки. В предложении **Info** изменяются те атрибуты, значения которых можно запросить с помощью [функции ObjectInfo\(\)](#).

Например, [функция ObjectInfo\(\)](#) позволяет определить текущий стиль штриховки (величину типа Brush).

```
Dim b_fillstyle As Brush
b_fillstyle = ObjectInfo(Selection.obj, OBJ_INFO_BRUSH)
```

И, наоборот, следующий оператор **Alter Object** восстанавливает прежнее значение стиля штриховки:

```
Alter Object obj_variable_name
Info OBJ_INFO_BRUSH, b_fillstyle
```

Заметьте, что один и тот же код (OBJ_INFO_BRUSH) используют [функция ObjectInfo\(\)](#), и оператор **Alter Object**.

В следующей таблице в первой колонке приводятся имена кодов для использования в предложении **Info** для выполнения различных видов изменений стиля. Значения *obj_info_code* находятся в файле MAPBASIC.DEF. Если требуется использовать коды в операторе **Alter Object...Info**, включите в начало вашей программы файл "MAPBASIC.DEF".

Значения <i>obj_info_code</i>	ID	Результат выполнения Alter Object
OBJ_INFO_PEN	2	Изменяется стиль линии или контура. Параметр <i>new_info_value</i> должен иметь значение типа Pen.

Значения <i>obj_info_code</i>	ID	Результат выполнения Alter Object
OBJ_INFO_SYMBOL	2	Изменяется стиль символа. Параметр <i>new_info_value</i> должен иметь значение типа Symbol.
OBJ_INFO_BRUSH	3	Изменяется стиль штриховки объекта. Параметр <i>new_info_value</i> должен иметь значение типа Brush.
OBJ_INFO_SMOOTH	4	Изменяется режим сглаживания углов для полилиний. Параметр <i>new_info_value</i> должен иметь значение логического типа (TRUE или FALSE).
OBJ_INFO_FRAMEWIN	4	Меняет содержимое рамки на изображение другого окна. Параметр <i>new_info_value</i> должен быть целочисленным идентификатором окна.
OBJ_INFO_FRAME_TITLE	6	Заменяет заголовок в рамке. Параметр <i>new_info_value</i> должен иметь значение типа String.
OBJ_INFO_TEXTFONT	2	Изменяется стиль шрифта. Параметр; <i>new_info_value</i> должен иметь значение типа Font.
OBJ_INFO_TEXTSTRING	3	Меняет текст в текстовом объекте. Параметр <i>new_info_value</i> должен иметь значение типа String.
OBJ_INFO_TEXTSPACING	4	Изменяет расстояние между строками в текстовом объекте. Параметр <i>new_info_value</i> должен иметь значение типа Float равным 1, 1.5, или 2.
OBJ_INFO_TEXTJUSTIFY	5	Изменяет значение выравнивания для текстовых объектов. Параметр <i>new_info_value</i> должен иметь одно из следующих целочисленных значений: /n0 – выравнивание влево /n1 – центрирование /n2 – выравнивание вправо.
OBJ_INFO_TEXTARROW	6	Изменяет вид указки для текстового объекта. Параметр <i>new_info_value</i> должен иметь одно из следующих целочисленных значений: /n0 – нет указки /n1 – только линия /n2 – линия со стрелкой.

Предложение Geography

Оператор **Alter Object** с предложением **Geography** изменяет расположение объекта.

Предложение **Geography** действительно для всех типов объектов за исключением полилиний

и областей. Для изменения расположения объектов последних двух типов используйте предложение **Node** (которое описано ниже) вместо предложения **Geography**.

Предложение **Geography** позволяет изменять атрибуты, которые возвращает **функция ObjectGeography()**. Например, следующая **функция ObjectGeography()** конечные координаты линейного объекта.

```
Dim o_cable As Object
Dim x, y As Float
x = ObjectGeography(o_cable, OBJ_GEO_LINEENDX)
y = ObjectGeography(o_cable, OBJ_GEO_LINEENDY)
```

Следующий оператор **Alter Object** изменяет координаты концов линейного объекта:

```
Alter Object o_cable
  Geography OBJ_GEO_LINEENDX, x
Alter Object o_cable
  Geography OBJ_GEO_LINEENDY, y
```

Примечание: Заметим, что и **функция ObjectGeography()**, и оператор **Alter Object** используют один и тот же код (OBJ_GEO_LINEENDX).

В следующей таблице в первой колонке приводятся имена кодов для использования в предложении **Geography** для осуществления различных видов географических изменений. Именам *obj_geo_code* присвоены целочисленные коды для удобства использования их в операторе. Они находятся в файле MAPBASIC.DEF. Если требуется использовать эти имена в операторе **Alter Object...Geography**, включите в начало вашей программы файл "MAPBASIC.DEF".

Значения <i>obj_geo_code</i>	ID	Результат выполнения Alter Object
OBJ_GEO_MINX	1	Изменяет X-координату верхнего левого угла минимального прямоугольного покрытия (МПП).
OBJ_GEO_MINY	2	Изменяет Y-координату верхнего левого угла МПП.
OBJ_GEO_MAXX	3	Изменяет X-координату нижнего правого угла МПП.
OBJ_GEO_MAXY	4	Изменяет Y-координату нижнего правого угла МПП.
OBJ_GEO_ARCBEGANGLE	5	Изменяет начальный угол дуги.
OBJ_GEO_ARCENDANGLE	6	Изменяет конечный угол дуги.
OBJ_GEO_LINEBEGX	1	Изменяет координату начальной точки линии.

Значения <i>obj_geo_code</i>	ID	Результат выполнения Alter Object
OBJ_GEO_LINEBEGY	2	Изменяет координату начальной точки линии.
OBJ_GEO_LINEENDX	3	Изменяет координату конечной точки линии.
OBJ_GEO_LINEENDY	4	Изменяет координату конечной точки линии.
OBJ_GEO_POINTX	1	Изменяет координату точечного объекта.
OBJ_GEO_POINTY	2	Изменяет координату точечного объекта.
OBJ_GEO_ROUNDRAIUS	5	Изменяет радиус закругления для объекта типа "скругленный прямоугольник".
OBJ_GEO_TEXTLINEX	5	Изменяет координату по оси X конца текстовой строки объекта.
OBJ_GEO_TEXTLINEY	6	Изменяет координату по оси Y конца текстовой строки объекта.
OBJ_GEO_TEXTANGLE	7	Изменяет угол поворота текстового объекта.

Предложение Node

Оператор **Alter Object** с предложением **Node** добавляет, перемещает или убирает узлы полилиний и многоугольников (полигонов), составляющих области.

Если предложение **Node** включает предложение **Add** оператор **Alter Object** добавляет узел к объекту. Включив в **Node** предложение **Remove**, можно удалить узел. Предложение **Set Position** меняет местоположение узла.

Обычно вместе с оператором **Alter Object** и предложением **Node** используются **оператор Create Pline** и **оператор Create Region**. Эти операторы позволяют создавать полилинии или полигоны. Операторы Create требуют явного задания количества узлов объекта на этапе компиляции. Однако, в ряде случаев вы не можете знать, сколько узлов будет содержать объект при работе программы.

Если нельзя задать количество узлов, то можно, используя **оператор Create Pline** или **оператор Create Region** создать "пустые" объекты (т. е. с нулевым количеством узлов). С помощью оператора **Alter Object...Node Add** можно добавить к созданному объекту любое количество узлов.

В предложении **Node** предложение **Position** задает два параметра: *polygon_num* и *node_num*, указывающие на конкретный узел, который будет изменен или удален. Предложение **Position**

при определении узла задавать необязательно. Параметры *polygon_num* и *node_num* могут принимать значения от 1 и более.

Параметр *polygon_num* задает, какой полигон из мультиполигонального объекта будет обработан.

Центроиды регионов

Центроиды регионов могут быть заданы командой **Alter Object** со следующим синтаксисом:

```
Alter Object Obj Geography OBJ_GEO_CENTROID, PointObj
```

Обратите внимание, что *PointObj* это точечный объект. Это отличает его от других значений, вводимых оператором **Alter Object Geography**, которые являются скалярными. Точечный объект для своего определения на карте нуждается в двух координатах. Вводимый объект Point проверяется на предмет того, может ли он быть нормальным центроидом, то есть, попадает ли он внутрь данного региона. Если имеющийся объект *Obj* не является регионом, или если *PointObj* не является точечным объектом, или если точка не является нормальным центроидом, то будет возвращено сообщение об ошибке.

Вот один из вариантов использования значений X и Y для центроида:

```
Alter Object Obj Geography OBJ_GEO_CENTROID, CreatePoint(X, Y)
```

Также опросить центроид может **функция ObjectGeography()**:

```
PointObj = ObjectGeography(Obj, OBJ_GEO_CENTROID)
```

Кроме того, информацию о центроиде возвращают **функция Centroid()**, **функция CentroidX()** и **функция CentroidY()**.

OBJ_GEO_CENTROID определяется в файле MAPBASIC.DEF.

Объекты группа точек и коллекция

Оператор **Alter Object** поддерживает также следующие типы объектов.

- **Группа точек:** устанавливает символ группы точек , как показано ниже:

```
Alter Object obj_variable_mpoint
Info OBJ_INFO_SYMBOL, NewSymbol
```

- **Коллекция:** используя оператор **Alter Object** с предложением **Info**, можно переустановить части коллекции (регион, полилиния или группа точек) внутри объекта "коллекция". Предложение **Info** позволяет видоизменять атрибуты, которые возвращает **функция**

ObjectInfo(). Например, **функция ObjectInfo()** определяет часть региона в объекте коллекции:

```
Dim ObjRegion As Object
ObjRegion = ObjectInfo(Selection.obj, OBJ_INFO_REGION)
```

Следующий оператор **Alter Object** позволяет переустановить часть регионов, входящих в коллекцию:

```
Alter Object obj_variable_name
Info OBJ_INFO_REGION, ObjRegion
```

Примечание: Код (OBJ_INFO_REGION) использует и **функция ObjectInfo()**, и оператор **Alter Object**.

Оператор **Alter Object** позволяет добавлять или удалять точки в объект типа "группа точек".

```
Alter Object obj Node statement
```

Чтобы вставить узлы в группу точек:

```
Dim mpoint_obj as object
Create Multipoint Into Variable mpoint_obj 0
Alter Object mpoint_obj Node Add (0,1)
Alter Object mpoint_obj Node Add (2,1)
```

Примечание: Узлы для группы точек всегда добавляются в конец таблицы.

Чтобы удалить узлы из группы точек:

```
Alter Object mpoint_obj Node Remove Position polygon_num, node_num
```

mpoint_obj – объект типа "группа точек";

polygon_num – игнорируется для группы точек, рекомендуется установить значение 1;

node_num – число удаляемых узлов.

Чтобы установить точки внутри группы точек:

```
Alter Object mpoint_obj Node Set Position polygon_num, node_num (x,y)
```

mpoint_obj – объект типа "группа точек";

polygon_num – игнорируется для группы точек, рекомендуется установить значение 1;

node_num – число узлов, которые будут изменены;

x and *y* – новые координаты узла *node_num*.

Пример

```
Dim myobj As Object, i As Integer
Create Region Into Variable myobj 0
For i = 1 to 10
  Alter Object myobj
  Node Add (Rnd(1) * 100, Rnd(1) * 100)
Next
```

Примечание: После использования оператора **Alter Object** для изменения объекта, используйте **оператор Insert** или **оператор Update** для сохранения объекта в таблице.

См. также:

оператор Create Pline, оператор Create Region, оператор Insert, функция ObjectGeography(), функция ObjectInfo(), оператор Update

Оператор Alter Table

Назначение

Изменяет структуру открытой таблицы. Не может быть применен к связанным таблицам. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Alter Table table (
  [ Add columnname columntype [ , ... ] ]
  [ Modify columnname columntype [ , ... ] ]
  [ Drop columnname [ , ... ] ]
  [ Rename oldcolumnname newcolumnname [ , ... ] ]
  [ Order columnname, columnname [ , ... ] ] )
[ Interactive ]
```

table – имя открытой таблицы;

columnname – имя колонки (поля) в открытой таблице, длина которой не должна превышать 31 символ и состоит из букв, цифр и символа подчеркивания и не может начинаться с цифры;

columntype – тип данных колонки (поля) в таблице (включая ширину поля, если необходимо);

oldcolumnname – старое имя колонки (поля) для переименования;

newcolumnname – новое имя колонки (поля) для переименования.

Описание

Оператор **Alter Table** используется для изменения структуры открытой таблицы. С его помощью можно добавлять и удалять колонки, изменять у существующих колонок ширину или тип данных, переименовывать колонки и менять порядок их расположения.

Примечание: Если в таблице были изменения, то перед тем как выполнить оператор **Alter Table** вы должны сохранить их на диске или восстановить таблицу с диска, или отменить изменения.

Параметр *columntype* может иметь следующие значения: integer – целое число; SmallInt – короткое целое число; float – десятичное число с плавающей точкой; decimal(size, decplaces) – десятичное число фиксированной длины (size) и числом знаков после запятой (decplaces); char(size) – строка символов, где size – максимальное количество символов в поле; date – дата; logical – логическая величина; **DateTime**.

Предложение **Add** в операторе **Alter Table** используется для добавления новой колонки в таблицу. Предложение **Modify** используется для изменения типа данных колонки. Предложение **Drop** используется для удаления колонки. Предложение **Rename** изменяет имя колонки. Предложение **Order** позволяет вам задать свой порядок расположения колонок. Один оператор **Alter Table** одновременно может включать в себя все пять предложений. Каждое предложение может обращаться к списку колонок; поэтому одним оператором **Alter Table** вы можете сделать все структурные изменения в таблице (см. пример).

Предложение **Order** работает только с колонками и не влияет на порядок записей в таблице. Перечислите через запятую имена полей в нужном порядке. Первое имя в списке соответствует самой левой колонке в окне списка. В таком же порядке будут расположены поля в окне **Информация** – в верхней строке будет значение из первой колонки таблицы и так далее по порядку.

Если программа применяет оператор **Alter Table** к таблице, которая имеет мемо-поля, то последние будут утеряны. Предупреждения об этом на экран выводиться не будет.

Оператор **Alter Table** может послужить причиной удаления слоев из окна карты, а также привести к потере объектов тематических слоев или объектов с косметического слоя. Если используется ключевое слово **Interactive**, то MapInfo Pro предложит пользователю сохранить тематические или/и косметические объекты.

Пример

Таблица GCPOP.TAB состоит из следующих колонок: "pop_88", "metsize", "fipscod" и "utmcode". Оператор **Alter Table** делает следующее: В предложении **Rename** меняет имя колонки "pop_88" на "population". В предложении **Drop** удаляет колонки "metsize", "fipscod", "utmcode". В предложении **Add** создает колонку "schoolcode" типа SmallInt и колонку "federalaid" вещественного типа. В предложении **Order** заново определяет порядок колонок в таблице.

```
Open Table "gcpop"
Alter Table gcpop
```

```
(Rename pop_88 population
Drop metsize, fipscode, utmcode
Add schoolcode SmallInt, federalaid Float
Order schoolcode, population, federalaid)
```

См. также:

[оператор Add Column](#), [оператор Create Index](#), [оператор Create Map](#), [оператор Create Table](#)

Функция ApplicationDirectory()

Назначение

Возвращает имя каталога, в котором находится файл выполняющегося приложения MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ApplicationDirectory$ ( )
```

Возвращаемая величина

Строковое выражение, путь к каталогу.

Описание

Вызов функции **ApplicationDirectory\$ ()** из приложения поможет вам определить каталог (или папку), из которого была запущена прикладная программа. Если в данный момент никаких приложений не загружено, и вы в окне **MapBasic** вызовете функцию **ApplicationDirectory\$ ()**, то она возвратит вам пустую строку.

Для определения каталога, в котором установлена программа MapInfo Pro, используется [функция ProgramDirectory\\$ \(\)](#).

Пример

```
Dim sAppPath As String
sAppPath = ApplicationDirectory$ ( )
' At this point, sAppPath might look like this:
'
' "C:\MAPBASIC\CODE\"
```

См. также:

функция [ProgramDirectory\(\)](#), функция [ApplicationName\(\)](#)

Функция `ApplicationName$()`

Назначение

Возвращает строку, содержащую имя работающей в данный момент программы MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ApplicationName$()
```

Возвращаемая величина

Строка, содержащая имя работающей в данный момент программы MapBasic.

Описание

Вызвав `ApplicationName$()` из программы MapBasic, можно определить имя этой программы. Если ни одна программа в данный момент не работает, а функция вызвана из окна **MapBasic**, то `ApplicationName$()` возвращает пустую строку.

Чтобы определить адрес на диске работающей в данный момент программы, используется [функция `ApplicationDirectory\$\(\)`](#).

Пример

```
Dim sAppName As String
sAppName = ApplicationName$()
' At this point, sAppName might look like this:
'
' "Test.MBX"
```

См. также:

[Функция `ApplicationDirectory\(\)`](#)

Функция Area

Назначение

Возвращает географическую площадь объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Area( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – единицы измерения площади (например, "sq km" – квадратные километры).

Возвращаемая величина

Вещественное – Float

Описание

Функция **Area()** возвращает площадь географического объекта, определённого параметром *obj_expr*.

Функция возвращает площадь области в единицах измерения, указанных в параметре *unit_name*; к примеру, чтобы получить площадь в акрах, укажите “акры” в качестве значения параметра *unit_name*. Список доступных единиц измерения смотрите в разделе: [Оператор Set Area Units](#).

Только полигоны, эллипсы, прямоугольники и прямоугольники со скругленными углами имеют площадь. Результат применения функции к точечным, текстовым объектам, а также к прямой линии, дуге и полилинии будет равен нулю. Для скругленного прямоугольника функция **Area()** возвращает приблизительное значение. MapBasic вычисляет площадь прямоугольников со скругленными углами, как если бы они были обычными прямоугольниками.

В большинстве случаев MapInfo Pro проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система - план, то выполняются декартовы вычисления.

Примеры

В этом примере демонстрируется, как используется функция **Area()** для вычисления площади географического объекта. Обратите внимание, что выражение *tablename.obj* (аналогично *states.obj*) представляет географический объект из текущей строки указанной таблицы.

```
Dim f_sq_miles As Float
Open Table "states"
Fetch First From states
f_sq_miles = Area(states.obj, "sq mi")
```

Можно также использовать функцию **Area()** в составе оператора SQL Select:

```
Select state, Area(obj, "sq km")
From states Into results
```

См. также:

[функция ObjectLen\(\)](#), [функция Perimeter\(\)](#), [функция CartesianArea\(\)](#), [функция SphericalArea\(\)](#), [оператор Set Area Units](#)

Функция AreaOverlap()

Назначение

Вычисляет площадь пересечения двух замкнутых объектов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
AreaOverlap( object1, object2 )
```

object1 и *object2* – два замкнутых (имеющих площадь) объекта.

Возвращаемая величина

Значение с плавающей точкой (в текущих территориальных единицах измерения MapBasic).

См. также:

[функция IntersectNodes\(\)](#), [функция Overlap\(\)](#), [функция ProportionOverlap\(\)](#), [оператор Set Area Units](#)

Функция Asc()

Назначение

Возвращает код первого символа строки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Asc ( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Целое – Integer

Описание

Функция **Asc()** возвращает код первого символа строки *string_expr*.

Если строка *string_expr* пустая, то функция **Asc()** возвращает ноль.

Примечание: Вычислительные платформы, на которых работает MapInfo Pro, имеют общие коды от 32 (пробел) до 126 (тильда). В этот диапазон входят все цифры и буквы латинского алфавита. Буквы русского алфавита в разных системах имеют разные коды.

В системах, поддерживающих двухбайтовые коды символов (например, Windows Japanese),: если первый символ в строке *string_expr* однобайтовый, то функция **Asc()** возвращает код от 0 до 255; если первый символ в строке *string_expr* двухбайтовый, то функция, **Asc()** возвращает код от 256 до 65535.

В системах, не поддерживающих двухбайтовые коды функция **Asc()** возвращает значения в диапазоне от 0 до 255.

В Unicode версиях MapInfo Pro эта функция работает с системным набором символов таким же образом как и не-Unicode версии MapInfo Pro.

Пример

```
Dim code As SmallInt  
code = Asc("Afghanistan")
```

```
' code will now be equal to 65,
' since 65 is the code for the letter A
```

См. также:

[функция CharVal\(\)](#), [функция Chr\\$\(\)](#), [таблицы кодов символов](#), [определение](#)

Функция Asin()

Назначение

Возвращает арксинус. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Asin( num_expr )
```

num_expr – численное выражение, результат которого должен находиться в диапазоне от единицы до минус единицы включительно.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Asin()** вычисляет арксинус числа, полученного в результате вычисления выражения *num_expr*. Другими словами, **Asin()** возвращает величину угла в радианах, синус которого равен параметру *num_expr*.

Результатом вычисления **Asin()** является угол, значение которого возвращается в радианах. Диапазон значения угла находится между $-\pi/2$ и $\pi/2$ радиан (число π равно приблизительно 3.141593, и $\pi/2$ радиан равно 90 градусам).

Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG. Чтобы ваша программа могла использовать эти коэффициенты конвертирования, она должна содержать файл "MAPBASIC.DEF".

Так как синус меняет свои значения от единицы до минус единицы, выражение *num_expr* должно возвращать числа от 1 до -1.

Пример

```

Include "MAPBASIC.DEF"
Dim x, y As Float
x = 0.5
y = Asin(x) * RAD_2_DEG

' y will now be equal to 30,
' since the sine of 30 degrees is 0.5

```

См. также:

[функция Acos\(\)](#), [функция Atn\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Функция Ask()

Назначение

Показывает диалоговое окно с сообщением или вопросом и предлагающее подтвердить или отменить предложение (**ОК** или **Cancel**). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Ask( prompt, ok_text, cancel_text )
```

prompt – текст сообщения, заданный в кавычках;

ok_text – текст на кнопке подтверждения (например, "Да" или "Продолжить");

cancel_text – текст на кнопке отмены (например, "Отменить" или "Нет").

Возвращаемая величина

Логическое

Описание

Функция **Ask()** формирует и выводит на экран диалоговое окно, в котором пользователю предлагается ответить "да" или "нет". Параметр *prompt* задает текст в окне вроде "Такой файл уже существует; продолжить?". Длина текста, который показывает функция **Ask()**, может достигать 2000 символов, но показаны будут только первые 299.

Диалоговое окно имеет две кнопки: одна для положительного ответа, другая для отрицательного. Параметры *ok_text* и *cancel_text* задают текст для этих кнопок (например, "ОК" или "Да" – для положительного ответа, "Нет" или "Стоп" – для отрицательного).

Если пользователь нажмет на кнопку с надписью *ok_text* функция **Ask()** вернет TRUE. При нажатии на кнопку с надписью *cancel_text* или клавишу Esc в диалоге функция **Ask()** возвращает FALSE. Текст для кнопок *ok_text* и *cancel_text* должен быть лаконичен и сжат. Если ваша фраза настолько велика, что не помещается в пределах стандартной кнопки, то лучше воспользуйтесь оператором **Dialog** вместо функции **Ask()**. Кнопка *ok_text* выбирается по умолчанию, то есть вместо нажатия мышкой эту кнопку можно выбрать клавишей **Enter**.

Пример

```
Dim more As Logical
more = Ask("Do you want to continue?", "OK", "Stop")
```

См. также:

оператор Dialog, [оператор Note](#), [оператор Print](#)

Функция Atn()

Назначение

Возвращает арктангенс. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Atn ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Atn()** возвращает арктангенс от числа, полученного в результате вычисления выражения *num_expr*. Другими словами, **Atn()** возвращает угловое значение в радианах, при котором тангенс равен числу *num_expr*. Число *num_expr* может быть любым.

Результатом вычисления **Atn()** является угол, значение которого возвращается в радианах. Диапазон значения угла находится между $-\pi/2$ и $\pi/2$ радиан (число π равно приблизительно 3.141593, и $\pi/2$ радиан равно 90 градусам).

Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG. Чтобы ваша программа могла использовать эти коэффициенты конвертирования, она должна содержать файл "MAPBASIC.DEF".

Пример

```
Include "MAPBASIC.DEF"
Dim val As Float

val = Atn(1) * RAD_2_DEG
'val is now 45, since the
'Arc tangent of 1 is 45 degrees
```

См. также:

[функция Acos\(\)](#), [функция Asin\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Функция Atan()

Назначение

Возвращает значение арктангенса числа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Atan ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Atan()** возвращает арктангенс числа, полученного в результате вычисления выражения *num_expr*. Другими словами, **Atan()** возвращает угол, при котором тангенс равен числу *num_expr*. Число *num_expr* может быть любым.

Результатом вычисления **Atan()** является угол, значение которого возвращается в радианах. Диапазон значения угла находится между $-\pi/2$ и $\pi/2$ радиан.

Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG. (Обратите внимание: для того чтобы программа могла использовать коэффициенты конвертирования DEG_2_RAD и RAD_2_DEG, она должна содержать файл "MAPBASIC.DEF".

Пример

```
Include "MAPBASIC.DEF"
Dim val As Float

val = Atan(1) * RAD_2_DEG
'val is now 45, since the
'Arc tangent of 1 is 45 degrees
```

См. также:

[функция Acos\(\)](#), [функция Asin\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Функция Atan2()

Назначение

Эта функция возвращает арктангенс y/x в диапазоне $-\pi$ до π радиан. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Float Atan2( y, x ) ;
```

num_expr – численное выражение угла в радианах.

Возвращаемая величина

Вещественное (Float) – в радианах.

Описание

Функция **Atan2** возвращает арктангенс y/x в диапазоне от $-\pi$ до π радиан. Если оба параметра **Atan2** равны 0, функция возвращает 0. Результат в радианах.

Функция **Atan2** использует знаки обоих параметров, для того чтобы определять квадрант возвращаемого значения.

Пример

```
Include "mapbasic.def
"Dim x, y, result As Float
x = -10.0 y = 10.0
result = ATan2(y,x)
'result will now be equal to 2.35619
```

См. также:

[функция Acos\(\)](#), [функция Atan\(\)](#), [функция Asin\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Оператор AutoLabel

Назначение

Размещает подписи объектов в окне карты на косметическом слое. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
AutoLabel
[ Window window_id ]
[ { Selection | Layer layer_id } ]
[ Overlap [ { On | Off } ] ]
[ Duplicates [ { On | Off } ] ]
```

window_id – идентификатор окна, целое число;

layer_id – имя таблицы или идентификатор слоя, целое число.

Описание

Оператор **AutoLabel** создает подписи для объектов (объекты типа “Текст”) в окне карты. Подписываются только те объекты, которые в данный момент видны в окне карты. Предложение

Window определяет окно карты. Если предложения **Window** нет, MapBasic обращается к самому верхнему окну карты. Предложение **Selection** определяет действие оператора на выборку. Если **Selection** и **Layer** не заданы, то подписываются все слои.

Предложение **Overlap** задает режимы взаимного перекрытия подписей. По умолчанию режим перекрытия установлен на **Off** (подписи не могут накладываться друг на друга). Чтобы MapInfo Pro подписывала объекты, не взирая на перекрытия, задайте режим **Overlap On**. Предложение **Duplicates** управляет количеством подписей для одного объекта. По умолчанию установлено значение **Off** (множественность подписей запрещена). Оператор **AutoLabel** использует текущие установки шрифта и расположение подписей. Позволяет задавать параметры подписей, выполняя команду **Слои** (на вкладке **ГЛАВНАЯ** в группе **Окна** документов). Изменить установки шрифта и расположение подписей из приложения MapBasic можно, используя **оператор Set Map..**

Пример

```
Open Table "world" Interactive
Open Table "worldcap" Interactive
Map From world, worldcap
AutoLabel
Window FrontWindow( )
Layer world
```

См. также:

оператор Set Map

Функция BarChartThemeInfo()

Назначение

Возвращает информацию об определённом тематическом слое столбцов карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
BarChartThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id map_window_id – идентификатор окна Карты;

thematic_layer_id Короткое целое SmallInt, которое идентифицирует тематический слой столбцов карты.

attribute Код – короткое целое (SmallInt), см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	BarChartThemeInfo() возвращает
BARCHARTTHEME_INFO_GRADUATION	1	Короткий целочисленный (<i>SmallInt</i>) результат, представляющий, как разность численных значений соответствуют разнице высот столбцов. Поддерживается три метода. BARCHARTTHEME_INFO_GRAD_ROOT(0) "SQRT" (квадратного корня) присваивает возрастающие высоты столбцов по мере увеличения квадратных корней значений. BARCHARTTHEME_INFO_GRAD_LOG(1) – постоянный размер символов. BARCHARTTHEME_INFO_GRAD_LOG(2) – изменяет высоты по логарифмам значений.
BARCHARTTHEME_INFO_ORIENTATION	2	Короткий целочисленный (<i>SmallInt</i>) результат, представляющий положение нижней границы столбчатой диаграммы относительно центроидов каждого из объектов. Возможны следующие 9 значений. BARCHARTTHEME_INFO_ORI_CC(0) – центр, центр . BARCHARTTHEME_INFO_ORI_TL(1) – верхняя граница, левая граница . BARCHARTTHEME_INFO_ORI_TC(2) – левая граница, центр . BARCHARTTHEME_INFO_ORI_TR(3) – верхняя граница, правая граница . BARCHARTTHEME_INFO_ORI_CL (4) – центр, левая граница. BARCHARTTHEME_INFO_ORI_CR(5) – центр, правая граница . BARCHARTTHEME_INFO_ORI_BL(6) – нижняя граница, левая граница. BARCHARTTHEME_INFO_ORI_BC(7) – нижняя граница, центр . BARCHARTTHEME_INFO_ORI_BR(8) – нижняя граница, правая граница.

код атрибута	ID	BarChartThemeInfo() возвращает
BARCHARTTHEME_INFO_PEN	3	Предложение Pen, описывающее линию границы столбцов диаграммы
BARCHARTTHEME_INFO_CHART_TYPE	4	Короткий целочисленный (<code>SmallInt</code>) результат, представляющий тип столбчатой диаграммы: стопкой или независимый. BARCHARTTHEME_INFO_CHART_STA(0) – столбцы стопкой. BARCHARTTHEME_INFO_CHART_MUL(1) – несколько столбцов.
BARCHARTTHEME_INFO_GRAD_STACKS	5	Логический (<code>Logical</code>) результат (Т или F), которым определяется, является ли диаграмма столбцов размерной, градуированной или нет.
BARCHARTTHEME_INFO_INDEP_SCALES	6	Логический (<code>Logical</code>) результат (Т или F), которым определяется, является ли диаграмма столбцов независимой или нет.
BARCHARTTHEME_INFO_HEIGHT	7	Вещественный (<code>Float</code>) результат, которым определяется численное значение, соответствующее высоте столбчатой диаграммы в "бумажных" единицах.
BARCHARTTHEME_INFO_HEIGHT_UNIT	8	Строковый (<code>String</code>) результат, которым определяется единица измерения "бумажных" единиц высоты столбчатой диаграммы.
BARCHARTTHEME_INFO_HEIGHT_VALUE	9	Вещественный (<code>Float</code>) результат, которым определяется численное значение, соответствующее "бумажным" единицам столбчатой диаграммы, используемых в BARCHARTTHEME_INFO_HEIGHT
BARCHARTTHEME_INFO_WIDTH	10	Вещественный (<code>Float</code>) результат, которым определяется численное значение, соответствующее ширине столбчатой диаграммы в "бумажных" единицах.
BARCHARTTHEME_INFO_WIDTH_UNIT	11	Строковый (<code>String</code>) результат, которым определяется единица измерения "бумажных" единиц ширины столбчатой диаграммы.
BARCHARTTHEME_INFO_BRUSH	12	Предложение Brush, описывающее штриховку столбцов диаграммы

См. также:

Функция [BarChartThemeItemInfo\(\)](#) на стр. 132

Функция BarChartThemeItemInfo()

Назначение

Возвращает информацию об определённой тематической характеристике на тематическом слое столбцов карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
BarChartThemeItemInfo ( map_window_id, thematic_layer_id, item_id,  
attribute )
```

map_window_id *map_window_id* – идентификатор окна карты;

thematic_layer_id Короткое целое *SmallInt*, которое идентифицирует тематический слой столбцов карты.

range_item_id Целое (*Integer*), которое идентифицирует тематическое выражение. Счётчик тематических выражений, использованных на этом слое, можно запрашивать с помощью команды **ThemeInfo()**.

attribute Код – короткое целое *SmallInt*, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

Код атрибута	ID	BarChartThemeItemInfo() возвращает
BARCHARTTHEME_ITEM_INFO_BRUSH	1	Предложение со стилем штриховки, использованной при отображении заданной характеристики с идентификатором ID на тематическом слое столбцов.

См. также:

Функция BarChartThemeInfo() на стр. 129

Оператор Beep

Назначение

Подает звуковой сигнал. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Beep
```

Описание

Оператор **Beep** посылает команду подачи звука на динамик компьютера.

Оператор Browse

Назначение

Открывает окно списка. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Browse expression_list From table
[ Position ( x, y ) [ Units paper_units ] ]
[ Width window_width [ Units paper_units ] ]
[ Height window_height [ Units paper_units ] ]
[ Row n ]
[ Column n ]
[ Min | Max | Floating | Docked | Tabbed | AutoHidden | Hide ]
[ Pen .... ] [ Priority n ] [ Name framename ]
[ Into { Window layout_win_id } { ID empty_frame_id } ]
```

expression_list — выражения, задающие через запятую колонки, или звездочка (*);

table — имя открытой таблицы;

x , y — координаты верхнего левого угла окна Списка в "бумажных" единицах *paper_units*; С помощью предложения **Into Window** задаётся положение относительно верхнего левого угла окна **Отчёт**.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "cm" для сантиметров).

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Floating — соответствует плавающему окну.

Docked — соответствует прикреплённому в стандартном положении окну.

Tabbed — соответствует окнам, оформленным в виде вкладок, которые также, иногда, вызываться как документы.

AutoHidden — соответствует автоматически скрытому окну.

Hide — дополнительное, необязательное ключевое слово, с помощью которого создаётся окно в скрытом состоянии, и работает подобно оператору `Set Window id hide`. Вывести на экран скрытое окно можно, используя оператор `Set Window...Show statement`

Примечание: *AutoHidden* — соответствует автоматически скрытому окну.

Примечание: Все четыре выше перечисленных состояния окна относятся только к 64-разрядной версии MapInfo Pro.

window_width и *window_height* — определяют размер окна списка в *paper_units*; С помощью предложения **Into Window** задаются ширина и высота рамки в окне **Отчёт**. Если действительные значения ширины или высоты не заданы, то эти значения параметров рамки будут присвоены автоматически.

n — положительное целое число от 1.

framename — строка, представляющая имя списка в рамке отчёта. Если рамке с изображением в окне **Отчёт** присваивается имя, то оно будет сохранено в рабочем наборе. Когда параметр **Name** записывается в рабочем наборе, то номер версии WOR-файла меняется на 1500.

layout_win_id — идентификатор окна **Отчёт**, целое число;

layout_win_id — идентификатор окна **Отчёт**, целое число;

Описание

Оператор **Browse** открывает новое окно списка для открытой таблицы. Можно также использовать при подготовке макета для того, чтобы вставить окно списка в окно **Отчёт**. В

этом описании будет использоваться имя окна списка либо для окна списка MapInfo Pro, либо для рамки со списком в окне **Отчёт**.

Список может показать все поля таблицы, если параметр *expression_list* равен звездочке (*), или же вид окна Списка может быть определен списком выражений. Предложение *expression_list* может, наоборот, состоять из списка выражений, каждое из которых задает одну колонку в окне Списка. Выражением может быть имя колонки, оператор, функция или число, которое определяет одну колонку окна списка. Имена колонок создаются из выражений, определяющих значения в колонке. Так, если значения в колонке задаются выражением *population / area(obj, "acre")*, то оно и будет именем колонки. Чтобы создать псевдоним выражения, добавьте это имя после выражения; см. пример ниже.

Предложение **Position** задает расположение окна на экране. X-координата задаёт расстояние (в "бумажных" единицах) либо от левого края окна MapInfo Pro до левого края окна списка, либо от левого края окна **Отчёт** до левого края рамки со списком в окне конструктора отчёта. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. Координата Y определяет расстояние от верхнего края окна списка до верхнего края окна MapInfo в "бумажных единицах".

Предложения **Width** и **Height** определяют ширину и высоту окна Списка. Если предложений **Width** и **Height** нет в операторе, MapInfo Pro самостоятельно определит размер следующим образом: высота окна списка приблизительно равна четверти рабочего окна MapInfo, так, чтобы было показано хотя бы несколько строк, а ширина зависит от длины названий колонок.

Предложения **Width** и **Height** определяют ширину и высоту окна Списка. Если предложений **Width** и **Height** нет в операторе, MapInfo Pro самостоятельно определит размер следующим образом: высота окна списка приблизительно равна четверти рабочего окна MapInfo, так, чтобы было показано хотя бы несколько строк, а ширина зависит от длины названий колонок.

Если оператор **Browse** включает в себя ключевое слово **Max**, то окно будет открыто полностью на всю рабочую область окна MapInfo Pro или заполнит весь макет в окне конструктора отчёта. Аналогично, если оператор **Browse** включает в себя ключевое слово **Min**, то окно будет свернуто в икону.

Предложение **Row** используется для определения, какая строка будет самой верхней в окне Списка. Если в операторе **Browse** не задано предложение **Row**, то верхней будет первая строка.

Предложение **Column** используется для определения, какая колонка будет самой левой в окне Списка. Если в операторе **Browse** не задано предложение **Column**, то самой левой будет первая колонка.

Pen — корректное **предложение Pen**. Это предложение задает стиль рамки элемента украшения.

n — целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

Результатом действия оператора **Into Window** является новая рамка в окне **Отчёт**.

Используйте параметр **ID**, чтобы добавить открытую карту в пустую рамку отчёта. Будут использованы свойства рамки, заданные ещё для пустой рамки (стили линии и штриховки, положение, размер, имя и параметр z-order). Сообщение об ошибке появится, если:

- `From Window` с идентификатором ID не является картой или списком: 1696 "Ошибка: требуется окно карты или списка."
- Рамка с идентификатором ID не является пустым: 1698 "Рамка ^0 отчёта должна быть пустой."
- `From Window` не закрывает: 1697 "Не удалось поместить окно в пустую рамку."
- Легенды карты не клонируются: 1708 "Не удалось поместить легенды в рамку карты."

В следующем примере создаётся список на основе открытой таблицы `World`, которая добавляется в пустую рамку окна отчёта.

```
Browser World Into Window layout_win_id ID empty_frame_id
```

Прежде повторно использовать уже заполненную рамку, её требуется очистить от содержимого, смотрите описание [оператора Set Designer Frame](#). После выполнения такого оператора остаётся пустая рамка того же размера и в том же месте, где была заполненная рамка, присвоенное которой имя сохраняется.

Пример

Этот пример демонстрирует, как открытую таблицу `WORLD` показать в окне списка.

```
Open Table "world"
Browse * From world
```

Данные из той же таблицы можно показать в окне списка по-другому, явно задавая колонки списка и используя выражения для них.

```
Open Table "world"
Browse
  country,
  population,
  population/area(obj, "sq km") "Density"
From world
```

Результатом будет окно списка из трех колонок. Первые две колонки содержат данные так, как они хранятся в файле таблицы `WORLD`. Третья колонка является вычисляемой, и MapBasic помещает в нее результат деления населения на площадь страны для каждой строки таблицы. Ей присваивается псевдоним "Density" (Плотность), который пользователь увидит в окне в строке заголовков колонок

См. также:

[функция FrontWindow\(\)](#), [оператор Set Browse](#), [оператор Set Designer Frame](#), [оператор Set Paper Units](#), [оператор Set Window](#), [функция WindowID\(\)](#)

Функция BrowserInfo()

Назначение

Возвращает информацию об окне Списка, в частности, общее число строк и колонок в окне, номеру строки, номеру колонки или значение в выбранной ячейке. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
BrowserInfo( window_id, attribute )
```

window_id – целое число, идентификатор окна;

attribute – целочисленный код типа информации, которая будет возвращена. Значения его представлено в таблице ниже.

Возвращаемая величина

Float, Logical или String; зависит от значения параметра attribute.

Описание

Функция BrowserInfo() возвращает информацию об окне карты-призмы. Эта функция не применима к окну Районирования.

Параметр *window_id* определяет, к какому из окон Списка обращается функция. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Функция BrowserInfo() может возвращать различные значения. Они зависят от параметра attribute. Параметр attribute должен принимать одно из следующих значений в таблице; коды определены в MAPBASIC.DEF.

параметр attribute	ID	Возвращаемая величина
BROWSER_INFO_NROWS	1	Общее число строк в окне списка.
BROWSER_INFO_NCOLS	2	Общее число колонок в окне списка.

параметр attribute	ID	Возвращаемая величина
BROWSER_INFO_CURRENT_ROW	3	Номер строки в окне списка, содержащей текущую ячейку. Нумерация строк начинается с единицы (1).
BROWSER_INFO_CURRENT_COLUMN	4	Номер колонки в окне списка, содержащей текущую ячейку. Нумерация колонок начинается с нуля (0).
BROWSER_INFO_CURRENT_CELL_VALUE	5	Значение, содержащееся в текущей ячейке окна списка.

Ошибки

ERR_BAD_WINDOW (590), если нет такого окна;

ERR_FCN_ARG_RANGE (644), если значение аргумента выходит за пределы, заданные при его определении.

ERR_WANT_BROWSER_WIN (312), если ID-номер не соответствует окну списка.

См. также:

[функция FrontWindow\(\)](#), [функция WindowID\(\)](#)

Предложение Brush

Назначение

Задаёт стиль штриховки графических объектов. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Brush brush_expr
```

brush_expr – выражение, результат которого есть величина типа Brush. Например, результат выполнения функции **MakeBrush**(*pattern*, *fgcolor*, *bgcolor*). (См. более подробную информацию о переменной Brush в разделе: [Функция MakeBrush\(\)](#)).

Описание

Предложение **Brush** входит в состав операторов, в которых необходимо задавать стиль штриха для следующих объектов: многоугольник (полигон), область (регион), прямоугольник и эллипс. **Brush** не является отдельным оператором. Различные операторы, такие как [оператор Create](#)

Ellipse, могут возвращать переменную этого типа. Слово **Brush** может сопровождаться выражением, возвращающим значение типа Brush. Например, такого вида:

```
Brush br_var
```

или же значение может задаваться вызовом функции:

```
Brush MakeBrush(64, CYAN, BLUE)
```

В некоторых операторах (таких как **оператор Set Map**), переменная **Brush** задается набором из трех целочисленных параметров (pattern, foreground_color, background_color), заключенных в скобки, например:

```
Brush(64, CYAN, BLUE)
```

В некоторых операторах MapBasic предложение **Brush** используется как параметр (т. е. переменная с именем **Brush**), а не в развернутом виде (когда после зарезервированного слова Brush ставится имя переменной со штриховкой или заливкой). Одним из примеров является **оператор Alter Object**.

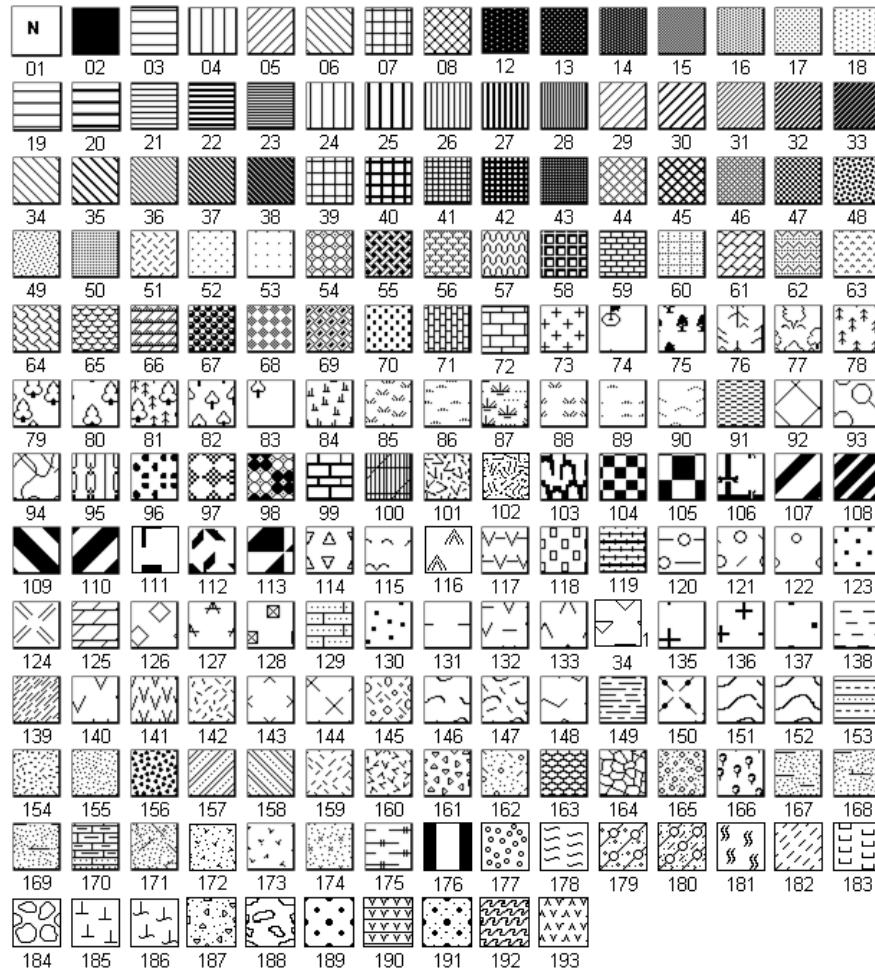
В таблице приводится определение компонент стиля штриховки замкнутых объектов:

Компонента стиля	Описание
штриховка	Целое число от 1 до 8 и от 12 до 186. Задаёт рисунок штриха. Смотрите таблицу ниже.
цвет	Целочисленный код цвета для рисунка штриха. Может быть заменен вызовом функции; см. раздел: Функция RGB() . Также можно использовать имена для кодов цвета (как в примере выше), если в тексте программы есть ссылка на файл стандартных определений MAPBASIC.DEF. В нем определены имена для следующих цветов: BLACK – черный; WHITE – белый; RED – красный; GREEN – зеленый; BLUE – синий; CYAN – голубой; MAGENTA – фиолетовый; YELLOW – желтый.
цвет фона	Целочисленный код цвета фона штриха.

Чтобы задать прозрачный фон, выберите штриховку под номером 3 или выше и не задавайте фон в предложении **Brush..** Например, задав *Brush(5, BLUE)*, получите штриховку тонкими синими линиями без фона. Не задавая параметр фона, вы поступаете аналогично сбросу флажка **Фон** в диалоге MapInfo Pro **Стиль области**.

Прозрачный фон задает **функция MakeBrush()** со значением фона -1.

Ниже приводится таблица кодов и соответствующих им рисунков штриховок. Заметим, что для штриха номер 1 (прозрачный) или 2 (ровная заливка), параметр background_color не влияет на заливку.



Подробный список штриховок приведен в *Справочной системе* – запустите MapInfo Pro и в меню *Справка* MapInfo Pro и запустите поиск слова *Штриховки*.

См. также:

функция **CurrentBrush()**, функция **MakeBrush()**, предложение **Pen**, предложение **Font**, предложение **Symbol**

Функция Buffer()

Назначение

Возвращает объект типа "Область", представляющий собой буферную зону вокруг выбранного объекта (область, граница которой отстоит от границы объекта на заданное расстояние). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Buffer( inputobject, resolution, width, unit_name )
```

inputobject – объектное выражение (выражение, результат которого есть величина типа Object);

resolution – число узлов многоугольника, принимаемого как окружность (число типа SmallInt);

width – радиус буфера, число типа Float; если *width* отрицательно и *inputobject* – закрытый объект, то возвращается объект меньший, чем исходный. Если задано отрицательное значение *width*, а объект является линейным (линией, полилинией или дугой) или же точкой, то используется абсолютные значение *width* и создается буфер больший чем объект.

unit_name – имя единицы измерения расстояний *width* (например, "mi" – миля, "km" – километр).

Возвращаемая величина

Область. Величина типа Object.

Описание

Функция **Buffer()** возвращает буферную зону (объект типа "полигон") вокруг объекта.

Функция **Buffer()** может создавать буферные зоны только вокруг одного объекта. Для создания буфера вокруг нескольких объектов, используйте **оператор Create Object As Buffer**. Объект должен быть создан с использованием координатной системы MapBasic. Метод расчета буфера зависит от координатной системы. Для планов используется декартов метод вычисления. В остальных случаях будет использован сферический метод вычисления

Пример

Следующий фрагмент программы создает объект "прямая линия". Создается буферная зона вокруг линии в 10 миль.

```
Dim o_line, o_region As Object
o_line = CreateLine(-73.5, 42.5, -73.6, 42.8)
o_region = Buffer( o_line, 20, 10, "mi")
```

См. также:

оператор Create Object, оператор Set Buffer Version

Функция ButtonPadInfo() – устаревшее

Назначение

Возвращает информацию о состоянии Инструментальной панели. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ButtonPadInfo ( pad_name, attribute )
```

pad_name – строковая величина, представляющая имя панели инструментов; например, “Операции”, “Пенал”, “Программы” или “Команды” для стандартных панелей или то имя, которое было определено при создании новых панелей.

attribute – целочисленный код, управляющий типом результата функции.

Возвращаемая величина

Зависит от значения параметра *attribute*.

Описание

В зависимости от значения *attribute*, функция вернет следующую информацию об Инструментальной панели. Коды определены в MAPBASIC.DEF.

<i>attribute-код</i>	ID	ButtonPadInfo() возвращает:
BTNPAD_INFO_FLOATING	1	Логическое — возвращается значение TRUE, если панель плавающая, и значение FALSE, если панель находится в прикрепленном состоянии.
BTNPAD_INFO_WIDTH	2	Короткое целое число – SmallInt: ширина панели, выраженная в виде количества кнопок (не включая разделители).
BTNPAD_INFO_NBTNS	3	Короткое целое число – SmallInt. Количество кнопок на панели.
BTNPAD_INFO_X	4	X-координата верхнего левого угла Инструментальной панели. Если панель закреплена, то это целое число. Значение отрицательно, если панель закреплена слева от строки меню. Если панель не закреплена, то значение есть вещественное число, в "бумажных" единицах измерения. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
BTNPAD_INFO_Y	5	Y-координата верхнего левого угла Инструментальной панели. Значение отрицательно, если панель закреплена выше строки меню.
BTNPAD_INFO_WINID	6	Целое – Integer. Идентификатор панели.
BTNPAD_INFO_DOCK_POSITION	7	Возвращает позицию панели, соответствующую положениям Floating , Left , Top , Right или Bottom . Для плавающих панелей может использоваться вместо BTNPAD_INFO_FLOATING. Возвращаемые значения: • BTNPAD_INFO_DOCK_NONE (0) • BTNPAD_INFO_DOCK_LEFT (1) • BTNPAD_INFO_DOCK_TOP (2) • BTNPAD_INFO_DOCK_RIGHT (3) • BTNPAD_INFO_DOCK_BOTTOM (4)

Пример

```

Include "mapbasic.def"
If ButtonPadInfo("Main", BTNPAD_INFO_FLOATING) Then
  '...then the Main pad is floating; now let's dock it.
  Alter ButtonPad "Main" ToolbarPosition(0,0) Fixed
End If

```

См. также:

оператор **Alter ButtonPad**

Оператор Call

Назначение

Вызывает sub-процедуру или внешнюю процедуру (DLL, XCMD).

Предупреждение!

Нельзя использовать оператор **Call** в окне **MapBasic** в Mapinfo Pro.

Синтаксис

```
Call subproc [ ( [ parameter ] [ , ... ] ) ]
```

subproc – имя подпрограммы;

parameter – параметр, передающий подпрограмме значение из основного модуля.

Описание

Оператор **Call** используется для вызова подпрограммы. Обычно подпрограмма является стандартной процедурой MapBasic (заданной операторами Sub... End Sub). Так же с помощью оператора **Call** Windows может быть вызвана процедура из динамической библиотеки (DLL).

Когда оператор **Call** вызывает условную процедуру MapBasic, MapBasic выполняет операторы, заданные в этой процедуре, до тех пор пока не будет выполнено условие, приводящее к оператору End Sub или Exit Sub. Затем MapBasic возвращается в программу, откуда был произведен вызов, к следующему оператору после **Call**. Оператор **Call** может обращаться только к тем процедурам, которые являются частью текста программы.

Программа MapBasic должна в главной процедуре содержать **оператор Declare Sub**. Это требование должно выполняться как по отношению к процедурам, организованным оператором Sub, так и к внешним процедурам (DLL и XCMD).

Передача значений параметров

Подпрограмма может не иметь параметров. Тогда вызов такой процедуры может выглядеть так:

```
Call subroutine
```


или

```
Call subroutine()
```

Если sub-процедура имеет параметры, то они могут быть объявлены двумя способами: ссылкой ("by reference") или значением ("by value"). По умолчанию параметр объявляется как "ссылка". Если параметр подпрограммы был определен таким образом, то при вызове sub-процедуры соответствующим параметром вызова должна быть переменная.

Такой параметр не только передает значение переменной в процедуру, но и возвращает значение обратно. Таким образом, параметр, передаваемый ссылкой, позволяет не только доставлять значение из вызывающей процедуры, но и возвращать значение обратно в этом же параметре.

Объявление величины параметра как значения осуществляется при помощи ключевого слова **ByVal** перед именем параметра в списке операторов **Sub** и **Declare Sub**. Если один из параметров подпрограммы был определен таким образом, то при вызове sub-процедуры соответствующим параметром вызова может быть как постоянная величина, так и переменная или выражение. Однако в этом случае новое значение параметра не может быть возвращено обратно той же переменной.

Вызов внешних процедур Sub-процедуры могут пересылать как простые параметры, так и массивы переменных. При этом в списке вызова Вы пишете только имя массива без скобок.

Пример

Процедура "Cube" возводит число в куб (в третью степень) и возвращает результат. Процедура имеет два параметра: первый для возводимого степень вещественного числа, второй для результата.

```
Declare Sub Cube(ByVal original As Float, cubed As Float)
  Dim x, result As Float
  Call Cube( 2, result)
  ' result now contains the value: 8 (2 x 2 x 2)
  x = 1
  Call Cube( x + 2, result)
  ' result now contains the value: 27 (3 x 3 x 3)
End Program
Sub Cube (ByVal original As Float, cubed As Float)
  ' Cube the "original" parameter, and store
  ' the result in the "cubed" parameter.
  cubed = original ^ 3
End Sub
```

См. также:

[оператор Declare Sub](#), [оператор Exit Sub](#), [оператор Global](#), [оператор Sub...End Sub](#)

Функция CartesianArea()

Назначение

Возвращает площадь, используя вычисления в декартовой системе координат. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianArea( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – единицы измерения площади (например, "sq km" – квадратные километры).

Возвращаемая величина

Вещественное – Float

Описание

Функция **CartesianArea()** возвращает площадь географического объекта, указанного выражением *obj_expr*.

Функция возвращает площадь области в единицах измерения, указанных в параметре *unit_name*; к примеру, чтобы получить площадь в акрах, укажите "акры" в качестве значения параметра *unit_name*. Смотрите раздел: [Оператор Set Area Units](#), где описаны возможные единицы измерения.

Функция **CartesianArea()** всегда будет возвращать значение площади, рассчитанное по декартовым алгоритмам. Величина -1 будет возвращаться для данных в координатах "Широта/Долгота", поскольку такие данные не могут быть конвертированы в декартовы.

Только полигоны, эллипсы, прямоугольники и прямоугольники со скругленными углами имеют площадь. По определению, значение функции **CartesianArea()** для точки, дуги, текста, линии или полилинии это ноль. Функция **CartesianArea()** возвращает приблизительные результаты, когда применяется к скругленным прямоугольникам. MapBasic вычисляет площадь прямоугольников со скругленными углами, как если бы они были обычными прямоугольниками.

Примеры

Следующие примеры показывают, как функция **CartesianArea()** может вычислять площадь одиночного картографического объекта. Обратите внимание, что выражение *tablename.obj*

(аналогично `states.obj`) представляет географический объект из текущей строки указанной таблицы.

```
Dim f_sq_miles As Float
Open Table "counties"
Fetch First From counties
f_sq_miles = CartesianArea(counties.obj, "sq mi")
```

Можно также совмещать функцию **CartesianArea()** и **оператор Select**, как показано в следующем примере.

```
Select lakes, CartesianArea(obj, "sq km")
From lakes Into results
```

См. также:

функция Area(), функция SphericalArea()

Функция CartesianBuffer()

Назначение

Возвращает объект типа "Область", представляющий собой буферную зону вокруг выбранного объекта (область, граница которой отстоит от границы объекта на заданное расстояние). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianBuffer( inputobject, resolution, width, unit_name )
```

inputobject – объектное выражение (выражение, результат которого есть величина типа Object);

resolution – число узлов многоугольника, принимаемого как окружность (число типа SmallInt);

width – это величина с плавающей запятой, представляющая радиус буфера; если ширина отрицательна, и если объект *inputobject* является замкнутым, то возвращаемый объект будет по размерам меньше исходного.

unit_name – имя единицы измерения расстояний *width* (например, "mi" – миля, "km" – километр).

Возвращаемая величина

Область

Описание

Функция **CartesianBuffer()** возвращает полигон, представляющий буферную зону. Она обрабатывает только один объект за раз.

Для создания буфера вокруг нескольких объектов, используйте **оператор Create Object As Buffer**. Если задано отрицательное значение *width*, а объект является линейным (линией, полилинией или дугой) или же точкой, то используется положительное значение *width* и создается буфер больший чем объект.

Функция **CartesianBuffer()** будет рассчитывать буферную зону, в предположении, что объект спроецирован на плоскость и используя ширину *width* для расчета декартового расстояния буферной зоны вокруг объекта.

Если *inputobject* задан в проекции "Широта/Долгота", то будут использоваться сферические вычисления независимо от того, какая будет применяться функция, связанная с буфером.

Пример

Следующая программа создает линейный объект, затем создает буфер вокруг него. Буферная зона занимает 10 миль во всех направлениях вокруг линии.

```
Dim o_line, o_region As Object
o_line = CreateLine(-73.5, 42.5, -73.6, 42.8)
o_region = CartesianBuffer( o_line, 20, 10, "mi")
```

См. также:

функция Buffer(), **создание объектов карты**, **оператор Set Buffer Version**

Функция CartesianConnectObjects()

Назначение

Возвращает объект, представляющий кратчайшее или самое большое расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianConnectObjects ( object1, object2, min )
```

object1 и *object2* — выражения, указывающие на объекты.

min — логическое выражение. Если это выражение принимает значение "Да" (TRUE), функция вычисляет минимальное расстояние между объектами, а если выражение принимает значение "Нет" (FALSE), вычисляется максимальное расстояние.

Возвращаемая величина:

Оператор возвращает объект типа "Полилиния", состоящий из одного сегмента и двух точек, и представляющий собой кратчайшее расстояние (*min* == TRUE), или самое большое расстояние (*min* == FALSE) между объектами *object1* и *object2*.

Описание

Одна точка из полученного в результате объекта-полилинии принадлежит объекту — *object1*, а другая точка принадлежит объекту — *object2*. Обратите внимание, что расстояние между двумя исходными объектами можно вычислить с помощью функции **ObjectLen()**. Если существует несколько вариантов кратчайших или наибольших расстояний (например, полученный сегмент представляет одно расстояние, а существуют еще и другие сегменты равной длины), эти функции возвращают лишь один из вариантов. Не существует способа определить, является ли полученный объект полилинии единственным кратчайшим расстоянием.

CartesianConnectObjects() возвращает объект-полилинию, соединяющую *object1* и *object2* по кратчайшему (*min* == TRUE) или самому длинному (*min* == FALSE) пути, вычисленному в декартовых координатах. Если вычисления не могут быть сделаны, используя метод расчётов на плоскости (то есть координатная система MapBasic — "Широта/Долгота"), то эта функция выдаст сообщение об ошибке.

См. также:

функция **ConnectObjects()**, функция **SphericalConnectObjects()**, оператор **Nearest**, оператор **Farthest**, функция **CartesianObjectDistance()**, функция **ObjectDistance()**, функция **SphericalObjectDistance()**

Функция CartesianDistance()

Назначение

Возвращает расстояние между двумя точками, заданными координатами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianDistance( x1, y1, x2, y2, unit_name )
```

$x1$ и $x2$ – х-координаты (долгота);

$y1$ и $y2$ – у-координаты (широта);

unit_name – это строковая величина, соответствующая имени единиц измерения расстояния (например, "km").

Возвращаемая величина

Вещественное – Float

Описание

Функция **CartesianDistance()** вычисляет расстояние между двумя точками. Функция возвращает измеренное расстояние в единицах, указанных параметром *unit_name*; например, что бы получить расстояние в милях, укажите "mi" как *unit_name*. Список доступных единиц измерения смотрите в разделе **оператор Set Distance Units**.

Функция **CartesianDistance()** всегда возвращает значение, используя для расчетов декартовый алгоритм. Будет возвращено значение -1 для данных в системе координат "Широта/Долгота", поскольку система "Широта/Долгота" не проективная и не декартовая.

Координаты по осям X и Y должны быть заданы в текущей системе координат MapBasic. По умолчанию MapInfo Pro предполагает, что координаты заданы в мировой системе координат ("Широта/Долгота"). Можно установить систему координат по умолчанию, выполнив **оператор Set CoordSys**.

Пример

```
Dim dist, start_x, start_y, end_x, end_y As Float
Open Table "cities"
Fetch First From cities
start_x = CentroidX(cities.obj)
start_y = CentroidY(cities.obj)
Fetch Next From cities
end_x = CentroidX(cities.obj)
end_y = CentroidY(cities.obj)
dist = CartesianDistance(start_x, start_y, end_x, end_y, "mi")
```

См. также:

математические функции, функция CartesianDistance(), функция Distance()

Функция CartesianObjectDistance()

Назначение

Возвращает расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianObjectDistance( object1, object2, unit_name )
```

object1 и *object2* — выражения, указывающие на объекты.

unit_name — строка, определяющая единицы измерения расстояния.

Возвращаемая величина

Вещественное – Float

Описание

CartesianObjectDistance() возвращает минимальное расстояние между объектами *object1* и *object2*, вычисленное в декартовых координатах. Возвращаемое значение измеряется в единицах *unit_name*. Если вычисления не могут быть сделаны, используя метод расчётов на плоскости (то есть координатная система MapBasic — "Широта/Долгота"), то эта функция выдаст сообщение об ошибке.

См. также:

[функция ObjectDistance\(\)](#), [функция SphericalObjectDistance\(\)](#), [оператор Nearest](#), [оператор Farthest](#), [функция SphericalConnectObjects\(\)](#), [функция ConnectObjects\(\)](#), [функция CartesianConnectObjects\(\)](#)

Функция CartesianObjectLen()

Назначение

Возвращает географическую протяженность объекта линии или полилинии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianObjectLen( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – это строковая величина, соответствующая имени единиц измерения расстояния (например, "km").

Возвращаемая величина

Вещественное – Float

Описание

Функция **CartesianObjectLen()** возвращает длину объекта. Помните, что ненулевые длины имеют только объекты линий и полилиний, для измерения периметра прямоугольника, эллипса или полигона, используется [функция Perimeter\(\)](#).

Функция **CartesianObjectLen()** всегда будет возвращать значение, используя декартовый алгоритм. Величина -1 будет возвращаться для данных в системе "Широта/Долгота", поскольку "Широта/Долгота" не проективна и не декартова.

Функция **CartesianObjectLen()** возвращает длину, измеренную в единицах длины, определенных параметром *unit_name* parameter; например, для получения длины в милях, укажите "mi" как *unit_name*. См. описание единиц измерения в разделе: [Оператор Set Distance Units](#).

Пример

```
Dim geogr_length As Float
Open Table "streets"
Fetch First From streets
geogr_length = CartesianObjectLen(streets.obj, "mi")
' geogr_length now represents the length of the
' street segment, in miles
```

См. также:

[функция SphericalObjectLen\(\)](#), [функция CartesianObjectLen\(\)](#), [функция ObjectLen\(\)](#)

Функция CartesianOffset()

Назначение

Возвращает копию исходного объекта, смещённого на определенную дистанцию и угол, определенный функцией Cartesian DistanceType. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianOffset( object, angle, distance, units )
```

object – объект, подвергаемый сдвигу.

angle – угол перемещения объекта.

distance – расстояние перемещения объекта.

units – строка, представляющая единицы измерения расстояния *distance*.

Возвращаемая величина

Объект

Описание

Эта функция создает новый объект, являющийся копией исходного и сдвинутый на расстояние *distance* вдоль угла *angle* (в градусах, где 0 градусов соответствует положительному направлению по оси X, а угол увеличивается против часовой стрелки). Строковое значение *units*, похожее на то, которые используются функцией **ObjectLen()** или функцией **Perimeter()**, задает единицы измерения расстояния. Используется декартовый тип DistanceType. Если система координат для исходного объекта это "Широта/Долгота", то будет выдано сообщение об ошибке, поскольку декартовые расстояния не годятся для "Широты/Долготы". В случае ошибки возвращается объект NULL. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций семейства Offset фактическое смещение рассчитано в некоторой установленной точке на объекте (например, центр описывающего прямоугольника), и затем это значение преобразовано из исходных единиц в единицы текущей системы координаты. Если система

координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
CartesianOffset(Rect, 45, 100, "mi")
```

См. также:

функция [CartesianOffsetXY\(\)](#)

Функция CartesianOffsetXY()

Назначение

Возвращает копию исходного объекта, смещённую на заданное расстояние по X и Y, используя декартовы координаты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianOffsetXY( object, xoffset, yoffset, units )
```

object – объект, подвергаемый сдвигу.

xoffset и *yoffset* – расстояние вдоль осей X и Y, на которые будет сделано смещение;

units – строка, представляющая единицы измерения расстояния distance.

Возвращаемая величина

Объект

Описание

Эта функция создает новый объект, который является копией исходного объекта, смещенного на расстояние *xoffset* вдоль оси X и на расстояние *yoffset* вдоль оси Y. Строковое значение *units*, похожее на то, которые используются функцией [ObjectLen\(\)](#) или функцией [Perimeter\(\)](#), задает единицы измерения расстояний. Используется декартовый тип DistanceType. Если система координат для исходного объекта это "Широта/Долгота", то будет выдано сообщение об ошибке, поскольку декартовые расстояния не годятся для "Широты/Долготы". В случае ошибки возвращается объект NULL. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций семейства Offset фактическое смещение рассчитано в некоторой установленной точке на объекте (например, центр описывающего прямоугольника), и затем это значение преобразовано из исходных единиц в единицы текущей системы координаты. Если система координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
CartesianOffset(Rect, 45, 100, "mi")
```

См. также:

функция [CartesianOffset\(\)](#)

Функция CartesianPerimeter()

Назначение

Возвращает периметр графического объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CartesianPerimeter( obj_expr , unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – это строковая величина, соответствующая имени единиц измерения расстояния (например, "km").

Возвращаемая величина

Вещественное – Float

Описание

Функция **CartesianPerimeter()** вычисляет периметр объекта *obj_expr*. Функция **Perimeter()** определена для следующих типов объектов: эллипсов, прямоугольников, скругленных прямоугольников и полигонов. Другие типы объектов имеют периметр нулевой длины.

Функция **CartesianPerimeter()** всегда будет возвращать значение, вычисленное по декартовому алгоритму. Величина -1 будет возвращаться для денных в системе Широта/Долгота, поскольку Широта/Долгота не проективна и не декартова.

Возвращаемое функцией **CartesianPerimeter()** значение длины периметра измеряется в единицах длины, определенных параметром *unit_name*; например, для получения длины в милях, укажите "mi" в качестве *unit_name*. См. описание единиц измерения в разделе: **Оператор Set Distance Units**. **CartesianPerimeter()** возвращает приближенный результат при измерении периметра скругленного прямоугольника. MapBasic вычисляет периметр прямоугольников со скругленными углами как если бы они были обычными прямоугольниками.

Пример

Следующий пример показывает как можно использовать функцию **CartesianPerimeter()** для определения периметра географического объекта.

```
Dim perim As Float
Open Table "world"
Fetch First From world
perim = CartesianPerimeter(world.obj, "km")
' The variable perim now contains
' the perimeter of the polygon that's attached to
' the first record in the World table.
```

Можно также использовать функцию **CartesianPerimeter()** внутри **оператора Select**. Следующий **оператор Select** извлекает информацию из таблицы States и сохраняет результаты во временной таблице Results. Поскольку **оператор Select** включает функцию **CartesianPerimeter()**, таблица Results будет включать колонку, показывающую периметр каждого штата.

```
Open Table "states"
Select state, CartesianPerimeter(obj, "mi")
  From states
  Into results
```

См. также:

функция CartesianPerimeter(), функция SphericalPerimeter(), функция Perimeter()

Функция Centroid()

Назначение

Возвращает центральную точку объекта (центроид). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Centroid( obj_expr )
```

obj_expr – выражение, определяющее объект.

Возвращаемая величина

точка

Описание

Функция **Centroid()** возвращает точечный объект, расположенный в центре объекта, представленного параметром *obj_expr*. Центроид области не совпадает с центром масс объекта. От расположения центроида области зависит расположение подписей, полученных в результате автоподписывания, точек геокодирования и расположения графиков и диаграмм на тематическом слое карты. Всегда можно передвинуть центроид области в нужное место на изменяемом слое карты в режиме изменения формы.

Если объект *obj_expr* является точечным, то функция **Centroid()** его и вернет. Если объект *obj_expr* – линия, то в результате применения **Centroid()** получится точка между ее концами.

Если *obj_expr* представляет объект типа "полилиния", то **Centroid()** возвращает середину среднего сегмента полилинии.

Для других типов объектов *obj_expr* функция **Centroid()** возвращает точку с координатами действительного центроида. Для прямоугольников, дуг, текстов или эллипсов точка будет равноудалена между верхней и нижней границами, и между правой и левой границами. Для объекта типа "Область" центроид всегда лежит внутри объекта, но не обязательно находится в геометрическом центре объекта.

Пример

```
Dim pos As Object  
Open Table "world"
```

```
Fetch First From world
pos = Centroid(world.obj)
```

См. также:

[оператор Alter Object](#), [функция CentroidX\(\)](#), [функция CentroidY\(\)](#)

Функция CentroidX()

Назначение

Возвращает координату центральной точки (центроида) по оси X. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CentroidX( obj_expr )
```

obj_expr – выражение, результат которого есть величина типа Object.

Возвращаемая величина

Вещественное – Float

Описание

Функция **CentroidX()** возвращает X-координату (или долготу) центроида точки объекта. См. раздел о [функции Centroid\(\)](#) где приводится концепция центроида для разных географических объектов (линий, областей, и т.д.).

Информация о координатах возвращается в текущей системе координат MapBasic; по умолчанию MapBasic использует координатную систему Долгота/Широта. Координатную систему можно изменить, используя [оператор Set CoordSys](#).

Примеры

В следующем примере функция **CentroidX()** вычисляет долготу центроида одного географического объекта.

```
Dim x As Float
Open Table "world"
Fetch First From world
x = CentroidX(world.obj)
```

Можно совместить функцию **CentroidX()** и **оператор Select**. Следующий **оператор Select** извлекает информацию из таблицы "World" и помещает ее во временную таблицу "Results". Поскольку **оператор Select** включает в себя функцию **CentroidX()** и **функция CentroidY()**, таблица "Results" будет включать колонки, показывающие широту и долготу для центроидов всех штатов.

```
Open Table "world"
Select country, CentroidX(obj), CentroidY(obj)
From world Into results
```

См. также:

функция Centroid(), **функция CentroidY()**, **оператор Set CoordSys**

Функция CentroidY()

Назначение

Возвращает координату центральной точки (центроида) по оси Y. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CentroidY( obj_expr )
```

obj_expr – выражение, определяющее объект.

Возвращаемая величина

Вещественное – Float

Описание

Функция **CentroidY()** возвращает X-координату (или долготу) центроида точки объекта. См. раздел о **функции Centroid()**, где приводится концепция центроида для разных географических объектов (линий, областей, и т.д.).

Информация о координатах возвращается в текущей системе координат MapBasic; по умолчанию MapBasic использует координатную систему Долгота/Широта. Координатную систему можно изменить, используя **оператор Set CoordSys**.

Пример

```
Dim y As Float
Open Table "world"
Fetch First From world
y = CentroidY(world.obj)
```

См. также:

функция [Centroid\(\)](#), функция [CentroidX\(\)](#), оператор [Set CoordSys](#)

CharSet clause

Purpose

Specifies which character set MapBasic uses for interpreting character codes.

Примечание: See the *MapInfo Pro User Guide* for changes affecting this clause.

Syntax

```
CharSet char_set
```

char_set is a string that identifies the name of a character set; see table below.

Description

The **CharSet** clause specifies which character set MapBasic should use when reading or writing files or tables. Note that **CharSet** is a clause, not a complete statement. Various file-related statements, such as the [Open File statement](#), can incorporate optional **CharSet** clauses.

What Is A Character Set?

Every character on a computer keyboard corresponds to a numeric code. For example, the letter "A" corresponds to the character code 65. A character set is a set of characters that appear on a computer, and a set of numeric codes that correspond to those characters.

Different character sets are used in different countries. For example, in the version of Windows for North America and Western Europe, character code 176 corresponds to a degrees symbol; however, if Windows is configured to use a different character set, character code 176 may represent a different character.

Call *SystemInfo(SYS_INFO_CHARSET)* to determine the character set in use at run-time.

How Do Character Sets Affect MapBasic Programs?

If your files use only standard ASCII characters in the range of 32 (space) to 126 (tilde), you do not need to worry about character set conflicts, and you do not need to use the **CharSet** clause.

Even if your files include "special" characters (for example, characters outside the range 32 to 126), if you do all of your work within one environment (e.g., Windows) using only one character set, you do not need to use the **CharSet** clause.

If your program needs to read an existing file that contains "special" characters, and if the file was created in a character set that does not match the character set in use when you run your program, your program should use the **CharSet** clause. The **CharSet** clause should indicate what character set was in use when the file was created.

The **CharSet** clause takes one parameter: a string expression which identifies the name of the character set to use. The following table lists all character sets available.

Character Set	Comments
"Neutral"	No character conversions performed.
"ISO8859_1"	ISO 8859-1 (UNIX)
"ISO8859_2"	ISO 8859-2 (UNIX)
"ISO8859_3"	ISO 8859-3 (UNIX)
"ISO8859_4"	ISO 8859-4 (UNIX)
"ISO8859_5"	ISO 8859-5 (UNIX)
"ISO8859_6"	ISO 8859-6 (UNIX)
"ISO8859_7"	ISO 8859-7 (UNIX)
"ISO8859_8"	ISO 8859-8 (UNIX)
"ISO8859_9"	ISO 8859-9 (UNIX)
"PackedEUCJapanese"	UNIX, standard Japanese implementation.

Character Set	Comments
"WindowsLatin2"	Windows Eastern Europe
"WindowsArabic"	
"WindowsCyrillic"	
"WindowsGreek"	
"WindowsHebrew"	
"WindowsTurkish"	
"WindowsTradChinese"	Windows Traditional Chinese
"WindowsSimpChinese"	Windows Simplified Chinese
"WindowsJapanese"	
"WindowsKorean"	
"CodePage437"	DOS Code Page 437 = IBM Extended ASCII
"CodePage850"	DOS Code Page 850 = Multilingual
"CodePage852"	DOS Code Page 852 = Eastern Europe
"CodePage855"	DOS Code Page 855 = Cyrillic
"CodePage857"	
"CodePage860"	DOS Code Page 860 = Portuguese
"CodePage861"	DOS Code Page 861 = Icelandic
"CodePage863"	DOS Code Page 863 = French Canadian
"CodePage864"	DOS Code Page 864 = Arabic
"CodePage865"	DOS Code Page 865 = Nordic
"CodePage869"	DOS Code Page 869 = Modern Greek

Character Set	Comments
"LICS"	Lotus worksheet release 1,2 character set
"LMBCS"	Lotus worksheet release 3,4 character set
"UTF-8"	Unicode character set (can be used in a Workspace, MapInfo Extended TAB file)
"UTF-16"	Unicode character set (can be used in a Native Extended TAB file)

Примечание: You never need to specify a CharSet clause in an **Open Table statement**. Each table's .TAB file contains information about the character set used by the table. When opening a table, MapInfo Pro reads the character set information directly from the .TAB file, then automatically performs any necessary character translations.

Specify "UTF-8" for a workspace character set or Native Extended (NativeX) format table character set to allow characters from any language to be represented. It can also be used when importing, opening or registering a non-native table.

To set MapBasic to compile files and create executable application files (.mbx) in UTF-8 charset, start the MapBasic development environment from command line with the parameter **-utf8** included.

To force MapInfo Pro to save a table in a specific character set, include a **CharSet** clause in the **Commit Table statement**.

See Also:

Commit Table statement, **Create Table statement**, **Export statement**, **Open File statement**, **Register Table statement**, **Character Code Table Definitions**

Функция ChooseProjection()

Назначение

Показывает диалог **выбора проекции** и возвращает координатную систему, выбранную пользователем. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ChooseProjection$( initial_coordsys, get_bounds )
```

initial_coordsys – строковая величина, которую задает **предложение CoordSys**. Она используется для установки той координатной системы, которая первый раз выбирается в диалоге. Если *initial_coordsys* является пустой или предложение CoordSys соответствует неправильной координатной системе, то по умолчанию в первом указании координатной системы используется система Широта/Долгота.

get_bounds – логическая величина, которая определяет, какие границы ввести пользователю при использовании плановых координат. Если *get_bounds* истинно, то появляется диалог, в котором надо определить границы Карты. Если это выражение ложно, то диалог не появляется и используются границы, заданные по умолчанию.

Описание

Эта функция отображает диалог **выбора проекции** и возвращает выбранную систему координат в виде строковой величины. Возвращаемая строковая величина имеет тот же формат, что и предложение CoordSys. Используйте эту функцию, если Вы хотите позволить пользователю установить проекцию внутри Вашего приложения.

Пример

```
Dim strNewCoordSys As String
strNewCoordSys = ChooseProjection$( "", True)
strNewCoordSys = "Set " + strNewCoordSys
Run Command strNewCoordSys
```

См. также:

функция MapperInfo()

Функция CharVal()

Назначение

Возвращает код первого символа строки. Эта функция является Unicode-аналогом функции **Asc()**, в которой возвращаемое значение строкового выражения (*string_expr*) связывает коды символов Unicode вместо системных кодов символов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CharVal( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Целое – Integer

Описание

Функция **CharVal()** возвращает код первого символа строки *string_expr*. Если *string_expr* является нулевой строкой, то **CharVal()** возвращает значение равной нулю.

Примечание: Помимо присваивания кодов символов, использование и поведение **CharVal()** являются такими же как и случае функции **Asc()**.

ChrU\$() – может применяться только в Unicode-версиях MapInfo Pro и возвращает нулевое значение в не-Unicode версиях MapInfo Pro.

Пример

В следующем примере отображаются соответствующие Unicode-кодировки символов: 8364, 1587 и 24330:

```
Dim code As SmallInt
code = CharVal("€")
code = CharVal("ﷺ")
code = CharVal("𐄎𐄌𐄍")
```

См. также:

[функция Asc\(\)](#), [функция ChrU\\$\(\)](#), [таблицы кодов символов](#), [определение](#)

Функция Chr\$

Назначение

Возвращает символ, соответствующий заданному коду символа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Chr$ ( num_expr )
```

num_expr – целочисленное выражение 0 от 255 (для двухбайтных кодировок от 0 до 65,535), включительно.

Возвращаемая величина

Строка – String

Описание

Функция **Chr\$()** возвращает строковое значение длиной в один символ, которое соответствует коду, полученному в результате вычисления выражения `num_expr`. Для большинства систем параметр `num_expr` должен быть положительным целым числом от 0 до 255. В операционных платформах, поддерживающих систему двухбайтовых кодов (например, Windows Japanese), параметр `num_expr` может быть числом от 0 до 65535.

Примечание: Вычислительные платформы, на которых работает MapInfo Pro, имеют общие коды от 32 (пробел) до 126 (тильда). В этот диапазон входят все цифры и буквы латинского алфавита. Буквы русского алфавита в разных системах имеют разные коды.

Если в результате вычисления `num_expr` получается дробное число, MapBasic округляет его до целого.

12-й символ в Windows соответствует переводу на новую страницу (form-feed). Оператор *Print Chr\$(12)* удобно использовать для очистки окна "Сообщения". Символ под номером 10 создает новую строку; см. пример ниже.

34-й символ в Windows соответствует двойной кавычке ("). Когда строка включает *Chr\$(34)*, MapBasic вставит в текст кавычки.

В Unicode версиях MapInfo Pro эта функция работает с системным набором символов таким же образом как и не-Unicode версии MapInfo Pro.

Ошибки

ERR_FCN_ARG_RANGE (644), если неправильно задано значение аргумента.

Пример

```
Dim s_letter As String * 1
s_letter = Chr$(65)
Note s_letter ' This displays the letter "A"
Note "This message spans" + Chr$(10) + "two lines."
```

См. также:

[функция Asc\(\)](#), [функция ChrU\\$\(\)](#), [таблицы кодов символов](#), [определение](#)

Функция ChrU\$()

Назначение

Возвращает символ, соответствующий заданному коду символа. Эта функция является Unicode-аналогом функции **Chr\$()**, в которой параметр кодировки `num_expr` связывает коды символов Unicode вместо системных кодов символов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ChrU$( num_expr )
```

Num_expr – целое значение в диапазоне от 0 до 65,535, включительно.

Возвращаемая величина

Строка – String

Описание

Функция **Chr\$()** возвращает строковое значение длиной в один символ, которое соответствует коду, полученному в результате вычисления выражения `num_expr`.

Примечание: Помимо присваивания кодов символов, использование и поведение **ChrU\$()** являются такими же как и случае **Chr\$()**. **ChrU\$()** – может применяться только в Unicode-версиях MapInfo Pro и возвращает символ пробела [space] в не-Unicode версиях MapInfo Pro.

Пример

В следующем примере отображается Unicode-символ "€", рабский Unicode-символ "س", традиционный китайский Unicode-символ "弊":

```
Note ChrU$(8364)
Note ChrU$(1587)
Note ChrU$(24330)
```

См. также:

[функция CharVal\(\)](#), [функция Chr\\$\(\)](#), [таблицы кодов символов](#), [определение](#)

Оператор Close All

Назначение

Закрывает все открытые таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Close All [ Interactive ]
```

Описание

Выполняя оператор **Close All**, MapBasic закрывает все открытые таблицы, включая те, которые изменялись. При этом все изменения после закрытия теряются. Предупреждения об этом на экран выводиться не будет. Если не хотите потерять текущие изменения в таблицах, используйте директиву **Interactive** для вывода на экран диалога, предлагающего пользователю сохранить или игнорировать изменения.

См. также:

[оператор Close Table](#)

Оператор Close Connection

Назначение

Закрывает соединение, которое открыл [оператор Open Connection](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Close Connection connection_handle
```

connection_handle – целочисленное выражение, представляющее из себя значение, которое возвращает [оператор Open Connection](#).

Описание

Оператор Close Connection закрывает соединение, номер которого было возвращает **оператор Open Connection**. Все режимы сервиса, поддерживавшего соединение, теряются.

См. также:

оператор Open Connection

Оператор Close File

Назначение

Закрывает открытый файл.

Синтаксис

```
Close File [ # ] filenum
```

filenum – целое число, идентификатор открытого файла.

Описание

Оператор **Close File** используется для закрытия файлов, для открытия которых использовался **оператор Open File**.

Примечание: Оператор **Open File** и **оператор Close File** работают с файлами, но не с таблицами MapInfo Pro. Для обращения к таблицам MapBasic имеет другие операторы; например, **оператор Open Table** используется для открытия таблицы.

Пример

```
Open File "cxdata.txt" For INPUT As #1
,
' read from the file... then, when done:
,
Close File #1
```

См. также:

оператор Open File

Оператор Close Table

Назначение

Закрывает открытую таблицу. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Close Table table [ Interactive ]
```

table – имя открытой таблицы.

Описание

Оператор **Close Table** используется для закрытия таблицы. Для закрытия сразу всех открытых таблиц используется **оператор Close All**.

Когда таблица показывается в одном или нескольких окнах графиков или списков, то при закрытии таблицы эти окна также закрываются. Если оператор **Close Table** закрывает единственную таблицу в окне карты, то закрывается и окно. Если применить оператор **Close Table** к связанной таблице, которая имеет несохраненные изменения, то MapInfo Pro запомнит изменения до следующего сеанса работы.

Сохранение изменений в таблице

Если не задать слово **Interactive**, изменения, которые не были сохранены на диске, при закрытии будут утеряны. Сообщений об этом не выводится. Если не хотите потерять текущие изменения в таблицах, используйте слово **Interactive** для вывода на экран диалога, предлагающего пользователю сохранить или игнорировать изменения.

Если требуется гарантировать несохранение изменений в таблице, опустите слово **Interactive** или используйте **оператор Rollback** перед **Close Table**. Если же, наоборот, требуется гарантировать сохранение изменений в таблице, выполните **оператор Commit Table** перед **Close Table**. Для определения, есть ли в таблице изменения, не сохраненные на диске, используется **функция TableInfo()** (*table*, **TAB_INFO_EDITED**).

Сохранение тематических и косметических объектов

Если закрываете таблицу, графические объекты которой показываются на единственном некосметическом слое окна карты, то будет закрыто и это окно. Чтобы сохранить объекты косметического и/или тематического слоев из этого окна, используйте ключевое слово **Interactive**.

Если **Interactive** опущено, выполнение оператора **Close Table** не сопровождается диалогом с предложением сохранить тематические и косметические объекты. Если включить слово **Interactive**, появится диалог с запросом о сохранении тематических и косметических объектов, если это имеет смысл. (Сообщение не будет выводиться, если на карте нет тематических слоев и объектов на косметическом слое.)

Примеры

```
Open Table "world"
' ... when done using the WORLD table,
' close it by saying:
Close Table world
```

Чтобы отменить выбор строк, закройте таблицу Selection.

```
Close Table Selection
```

См. также:

[оператор Close All](#), [оператор Commit Table](#), [оператор Open Table](#), [оператор Rollback](#), [функция TableInfo\(\)](#)

Оператор Close Window

Назначение

Закрывает или скрывает окна. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Close Window window_spec [ Interactive ]
```

window_spec – имя окна (такое как Ruler), код окна (такой как WIN_RULER) или целое число, определяющее окно.

Описание

Оператор **Close Window** закрывает или прячет окна MapInfo Pro.

Для закрытия документального окна (окна **Карта**, **Список**, **График** и **Отчёт**) необходимо задать его целым числом в параметре *window_spec*. Для закрытия документального окна (окна карты, списка, графика и отчёта) необходимо задать его целым числом в параметре *window_spec*.

Это число можно определить с помощью либо **функции `FrontWindow( )`**, либо **функции `WindowID( )`**.

Чтобы закрыть специальное окно или Инструментальную панель, надо в *window_spec* задать специальное имя окна или его код. Например, окно "Линейка" можно задавать либо именем Ruler, либо кодом WIN_RULER.

Чтобы закрыть окно элементов оформления, задайте ID-номер элемента, который возвращает **функция `MapperInfo( )`**.

В следующей таблице приводятся имена и коды для параметра *window_spec*:

Значение	Описание окна
Help	Окно Справочной системы (WIN_HELP).
Info	Окно Информация (WIN_INFO).
LayerControl	Окно Управление слоями (WIN_LAYER_CONTROL). В приложениях интегрированной картографии это относится к модальному диалогу.
Legend	Окно Легенда (WIN_LEGEND).
MapBasic	Окно MapBasic (WIN_MAPBASIC).
Message	Окно Сообщение , которое использует оператор <code>Print</code> (WIN_MESSAGE).
MoveMapTo	Окно Переместиться по Карте (WIN_MOVE_MAP_TO).
Ruler	Окно инструмента Линейка (WIN_RULER).
Statistics	Окно Статистика (WIN_STATISTICS).
TableList	Окно Список таблиц (WIN_TABLE_LIST). В приложениях интегрированной картографии это относится к модальному диалогу.

Примечание: ID-номера окон **Список таблиц**, **Управление слоями** и **Переместиться по карте** к игнорируются оператором **`Set Window`**, функцией **`WindowInfo( )`** и функцией **`WindowID( )`**.

Сохранение тематических и косметических объектов

Если требуется закрыть карту, то может возникнуть необходимость сохранить объекты с косметического или тематических слоев. Чтобы сохранить объекты косметического и/или тематического слоев из этого окна, используйте ключевое слово **Interactive**.

Если **Interactive** опущено, выполнение оператора **Close Window** не сопровождается диалогом с предложением сохранить тематические и косметические объекты. Если включить слово **Interactive**, появится диалог с запросом о сохранении тематических и косметических объектов, если это имеет смысл. (Сообщение не будет выводиться, если на карте нет тематических слоев и объектов в косметическом слое.)

Пример

```
Close Window Legend
```

См. также:

[оператор Open Window](#), [оператор Print](#), [оператор Set Window](#)

Функция ColumnInfo()

Назначение

Возвращает информацию о колонке в открытой таблице. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ColumnInfo( { tablename | tablenum } ,  
           { columnname | "COLn" } , attribute )
```

tablename – имя открытой таблицы;

tablenum – целое число, идентифицирующее таблицу;

columnname – имя колонки в этой таблице;

n – целое число, номер колонки в таблице;

attribute – код, управляющий типом результата функции.

Возвращаемая величина

Зависит от значения параметра *attribute*.

Описание

Функция **ColumnInfo()** возвращает информацию об одной колонке в открытой таблице.

Первый параметр функции задает таблицу (именем или идентификатором). Вторым параметром определяет колонку. Параметр *attribute* должен принимать значения целочисленного кода, задающего тип возвращаемой функцией информации. В следующей таблице в первой колонке приводятся имена кодов для использования в качестве параметра *attribute*.

Параметры	ID	ColumnInfo() возвращает:
COL_INFO_NAME	1	Имя колонки (строка).
COL_INFO_NUM	2	Номер колонки (короткое целое число).
COL_INFO_TYPE	3	Тип колонки (короткое целое число). Смотрите вторую таблицу.
COL_INFO_WIDTH	4	Ширина символьного или десятичного поля (короткое целое число). Используется только для символьных и десятичных колонок.
COL_INFO_DECPLACES	5	Число знаков после десятичной точки в поле десятичного типа (короткое целое число).
COL_INFO_INDEXED	6	Признак индексирования колонки (логическая величина).
COL_INFO_EDITABLE	7	Признак наличия изменений значений колонки (логическая величина)

Если функция **ColumnInfo()** имеет значение параметра *attribute* равным COL_INFO_TYPE, MapBasic возвращает одно из следующих:

ColumnInfo() возвращает:	ID	Тип колонки:
COL_TYPE_CHAR	1	Символьный
COL_TYPE_DECIMAL	2	Десятичный с фиксированной запятой
COL_TYPE_INTEGER	3	Целочисленный (4 байта)
COL_TYPE_SMALLINT	4	Короткое целое число (2 байта)

ColumnInfo() возвращает:	ID	Тип колонки:
COL_TYPE_DATE	5	Дата.
COL_TYPE_LOGICAL	6	Логический (TRUE или FALSE).
COL_TYPE_FLOAT	8	Вещественный.
COL_TYPE_TIME	37	Время.
COL_TYPE_DATETIME	38	ДатаВремя
COL_TYPE_LARGEINT	39	Длинное целое

Все определения из этих таблиц находятся в файле `MAPBASIC.DEF`. Если Вы будете использовать имена, включите в начало Вашей программы оператор `Include "MAPBASIC.DEF"`.

Ошибки

ERR_TABLE_NOT_FOUND (405), если не найдена данная таблица;

ERR_FCN_ARG_RANGE (644), если значение аргумента выходит за пределы, заданные при его определении.

Пример

```
Include "MAPBASIC.DEF"
Dim s_col_name As String, i_col_type As SmallInt
Open Table "world"
s_col_name = ColumnInfo("world", "col1", COL_INFO_NAME)
i_col_type = ColumnInfo("world", "col1", COL_INFO_TYPE)
```

См. также:

функция NumCols(), функция TableInfo()

Функция Combine()

Назначение

Возвращает либо объект типа "Область", либо "Полилиния", представляющий объединение двух объектов. Не применяется к объектам типа "Текст". Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Combine( object1, object2 )
```

object1, *object2* – два объектных выражения, результатом вычисления которых должны быть либо два объекта, имеющих площадь (например, область и окружность), либо два линейных объекта (например, прямая линия и полилиния).

Возвращаемая величина

Объект, являющийся объединением *object1* и *object2*.

Описание

Функция **Combine()** возвращает объект, являющийся результатом географического объединения двух объектов. Объединение двух смежных объектов удаляет границу между ними.

Функция **Combine()** обновлена и теперь поддерживает комбинирование разнородных объектов, а также может обрабатывать точки, группы точек и коллекции. Ранее оба исходных объекта должны были быть либо линейными (Линии, Полилинии или Дуги), порождая в результате полилинию; либо замкнутыми (Области, Прямоугольники, Скругленные прямоугольники или Эллипсы), порождая области. Не была разрешена комбинация разнородных объектов с участием Точек, Групп точек и Коллекций. К текстовым объектам функция **Combine()** неприменима.

Точки, Группы точек и Коллекции, введенные в MapInfo Pro 6.5, могут участвовать в отдельных операциях комбинации. В следующей таблице приведены все возможные варианты комбинирования объектов и типы результатов:

Тип исходного объекта	Тип исходного объекта	Тип результата
Точка или Группа точек	Точка или Группа точек	Группа точек

Тип исходного объекта	Тип исходного объекта	Тип результата
Линейный (Линия, Полилиния, Дуга)	Линейный	Полилиния
Замкнутый (Область, Прямоугольник, Скругленный Прямоугольник, Эллипс)	Замкнутый	Полигон
Точка, Группа точек, Линейный, Замкнутый полигон, Коллекция	Точка, Группа точек, Линейный, Замкнутый полигон, Коллекция	Коллекция

Результаты, которые возвращает **Combine()**, похожи на результаты, полученные командой MapInfo Pro **Объединить** (на вкладке **ОБЪЕКТЫ**), исключая тот факт, что команда **Объединить** изменяет исходные объекты, а функция **Combine()** не меняет выражения *object1* или *object2*. Кроме того функция **Combine()** не проводит операцию обобщения данных.

Объект, полученный в результате выполнения функции **Combine()**, оформляется так же как объект *object1*. Коллекция объектов, полученная в результате, унаследует от объекта *object1* все возможные компоненты стиля, а остальные компоненты останутся такими же, как у *objects2*. Например, если *object1* – это область, а *object2* – полилиния, то к коллекции, полученной в результате, будет применена штриховка и окаймление от объекта *object1* и стиль линии от объекта *object2*.

См. также:

[предложение Brush](#), [оператор Objects Combine](#), [предложение Pen](#), [оператор Set Combine Version](#)

Функция CommandInfo()

Назначение

Возвращает информацию о последних внутрисистемных событиях. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CommandInfo( attribute )
```

attribute – целочисленный код типа информации, которая будет возвращена.

Возвращаемая величина

Logical, Float, Integer или String.

Описание

Функция **CommandInfo()** возвращает информацию о последних событиях в среде MapInfo Pro – например, какие изменения произошли во временной таблице "Selection", в какое место окна указал пользователь, использовал ли он только мышь или нажал на кнопку мышки в комбинации с клавишей **SHIFT** и так далее.

Вызов после закрытия диалога

Вызывая функцию **CommandInfo()** сразу после закрытия окна диалога, можно использовать в качестве параметра *attribute* один из следующих кодов:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_DLG_OK	1	Величина типа Logical: TRUE – если в диалоговом окне была выбрана кнопка OK ; FALSE – если пользователь отменил диалог, нажав на кнопку Cancel или на клавишу ESC . (Этот код в функции предназначен только для информации о диалоговых окнах, которые вызывает Оператор Dialog.)
CMD_INFO_STATUS	1	Величина типа Logical: TRUE, если пользователь позволил процессу, снабженному диалогом со шкалой (см. оператор ProgressBar), завершиться самостоятельно; FALSE – если пользователь в этом диалоге нажал на кнопку Cancel , прекращающую выполнение.

Вызов из обработчиков новых меню и диалогов

Если приложение строит свои меню с элементами, вызывающими обработчик, или строит диалог, элементы которого вызывают процедуры-обработчики, то в этих процедурах можно использовать функцию **CommandInfo()** со следующими кодами *attribute*:

код атрибута	ID	Результат CommandInfo (attribute):
CMD_INFO_DLG_DBL	1	Величина типа Logical: TRUE, если пользователь использовал при указании на элемент списка элемент ListBox или MultiListBox двойное нажатие на клавишу мышки. Этот код используется только внутри процедуры-обработчика элемента диалога, построенного приложением MapBasic.
CMD_INFO_MENUITEM	8	Целое число (Integer), представляющее идентификатор элемента меню, который запустил этот обработчик. Этот код используется только внутри процедуры-обработчика элемента меню.

Вызов из обработчиков системных событий

Если ваше приложение имеет процедуры-обработчики системных событий (например, SelChangedHandler), то в этих процедурах можно использовать функцию **CommandInfo()** со следующими кодами. Более подробную информацию можно найти в описании процедур SelChangedHandler, RemoteMsgHandler, WinChangedHandler и WinClosedHandler.) Если использовать процедуру SelChangedHandler:

attribute-код	ID	Результат CommandInfo (attribute):
CMD_INFO_SELTYPE	1	1 если единственная запись добавлена в выборку; 2 если единственная запись удалена из выборки; 3 если несколько записей добавлены в выборку; 4 если отменен выбор нескольких записей.
CMD_INFO_ROWID	2	Целое число (Integer) – номер строки, которая была добавлена в выбор или исключена из выбора (работает только, если одна строка была выбрана или исключена из выбора).
CMD_INFO_INTERRUPT	3	Логическая величина (Logical). TRUE, если пользователь прервал выбор клавишей ESC , FALSE – иначе.

Из **процедуры RemoteMsgHandler**, функции **RemoteQueryHandler()** или **процедуры RemoteMapGenHandler**:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_MSG	1000	Строка, представляющая собой “выполняемое” сообщение или имя элемента послания MapInfo Pro от программы-клиента. Более подробное описание см. в разделах: Процедура RemoteMsgHandler , Функция RemoteQueryHandler() или Процедура RemoteMapGenHandler .

Процедура WinChangedHandler или Процедура WinClosedHandler:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_WIN	1	Целое число (Integer), идентификатор окна, в котором было изменение или которое было закрыто. Более подробное описание см. в разделах: Процедура WinChangedHandler или Процедура WinClosedHandler .

Процедура ForegroundTaskSwitchHandler:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_TASK_SWITCH	1	Целое число (Integer), указывающее, что MapInfo Pro либо только что стало активным приложением, либо только что перестало быть активным. Результатом может быть один из кодов: SWITCHING_INT0_MI Pro (если MapInfo Pro получает фокус) SWITCHING_OUT_OF_MapInfo Pro (если MapInfo Pro теряет фокус)

Вызов после операции поиска

После того, как был выполнен **Оператор Find**, параметр функции *attribute* может принимать одно из следующих значений:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_X or CMD_INFO_Y	1, 2	Действительное число, координата (по оси X или Y) найденного места.
CMD_INFO_FIND_RC	3	Целое число (Integer), код, который возвращает Оператор Find в случае успеха.

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_FIND_ROWID	4	Целое число (Integer), номер строки (Row ID), которой соответствует найденный объект.

Вызов из процедуры-обработчика инструмента

Следующие коды можно задавать, если используется **Процедура ToolHandler**.

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_CUSTOM_OBJ	1	Объектная величина: полилиния или многоугольник, нарисованный пользователем. Применяется с режимами рисования DM_CUSTOM_POLYLINE или DM_CUSTOM_POLYGON.
CMD_INFO_X	1	X-координата точки, где была нажата кнопка мыши: • на карте, долгота в установленных единицах и в соответствии с текущей в MapBasic системой в списке, номер колонки в окне (единица соответствует самой левой колонке в окне); • в списке, номер колонки в окне (единица соответствует самой левой колонке в окне, ноль соответствует колонке с флажками выбора).* • в отчёте, расстояние между левым краем отчёта и указанной точкой в текущих "бумажных" единицах (ноль соответствует точке на левом крае листа отчёта). "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units.
CMD_INFO_Y	2	Y-координата точки, где была нажата кнопка мыши: • на карте, широта в установленных единицах и в соответствии с текущей в MapBasic системой координат; • в списке, номер строки в окне (единица соответствует самой верхней строке в окне, ноль соответствует строке с заголовками).* • в отчёте, расстояние между верхним краем отчёта и указанной точкой в текущих "бумажных" единицах (ноль соответствует точке на верхнем крае листа отчёта);
CMD_INFO_SHIFT	3	Логическая величина: TRUE (логическое "Да"), если пользователь при указании нажал на клавишу SHIFT .
CMD_INFO_CTRL	4	Логическая величина: TRUE (логическое "Да"), если пользователь при указании нажал на клавишу CTRL .

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_X2	5	X-координата точки, в которой пользователь отпустил кнопку мыши. Использование этого кода возможно, если инструменту был назначен режим рисования (например, использовался код DM_CUSTOM_LINE).
CMD_INFO_Y2	6	Y-координата точки, в которой пользователь отпустил кнопку мыши.
CMD_INFO_TOOLBTN	7	Целое число (Integer), идентификатор кнопки инструмента, который вызвал этот обработчик.

* Функция CommandInfo() игнорирует указания мыши на самую верхнюю левую ячейку окна списка, то есть над флажками для выбора и слева от строки заголовков. Игнорируются также указания мыши за пределами последних колонок и строк. Игнорируются также указания мыши за пределами последних колонок и строк

Поддержка геолинка (Hotlink)

Приложения MapBasic, запущенные инструментом Геолинк (Hotlink), могут получать информацию об активном объекте через функцию **CommandInfo()**. В таблице приведен список атрибутов которые могут быть запрошены:

<i>attribute</i> -код	ID	Результат CommandInfo (<i>attribute</i>):
CMD_INFO_HL_WINDOW_ID	17	ID-номер окна карты или списка.
CMD_INFO_HL_TABLE_NAME	18	Имя таблицы, ассоциированной со слоем карты или окном списка.
CMD_INFO_HL_ROWID	19	ID-номер строки таблицы соответствующей объекту карты или записи в списке.
CMD_INFO_HL_LAYER_ID	20	ID-номер слоя, если программа была запущена из окна карты.
CMD_INFO_HL_FILE_NAME	21	Имя запущенного файла.

См. также:

функция **FrontWindow()**, функция **SelectionInfo()**, оператор **Set Command Info**, функция **WindowInfo()**

Оператор Commit Table

Назначение

Сохраняет последнюю редакцию таблицы на диске или сохраняет ее копию. Ранее не было разрешено сохранять запросы, которые содержали неопределенные типы, что часто случалось при запросах ObjectInfo. Мы добавили параметр Interactive, позволяющий задавать неопределенные типы, такие как запросы. Если параметр Interactive не задан, подставляются типы по умолчанию. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Commit Table table
[ As filespec
  [ Type { NATIVE [ Charset char_set ] |
          NATIVEX [ Charset char_set ] [ Locale locale ] |
    DBF [ Charset char_set ] |
    Access Database database_filespec Version version Table tablename
    [ Password pwd ] [ Charset char_set ] |
    QUERY |
    ODBC Connection ConnectionNumber Table tablename
    [ Type SQLServerSpatial { Geometry | Geography } ]
    [ ConvertDateTime { ON | OFF | INTERACTIVE } ] |
    GPKG [ Database database_filespec ] [ Table tablename ] [ Charset
char_set ] } ]
  [ CoordSys... ]
  [ Version version_pro ] ]
[ Interactive ]
[ BlockSize block_size ]
[ { Interactive | Automatic commit_keyword } ]
[ ConvertObjects { ON | OFF | INTERACTIVE } ]
```

tableName – строковая переменная, определяющая имя таблицы, которая появится в Access; имя таблицы может включать в себя имя схемы, которой принадлежит таблица. Если имени схемы не задано, то таблица принадлежит стандартной схеме. Пользователь должен задать правильное имя схемы, должен знать имя учетной записи и обладать правами доступа к указанной схеме. Это касается только SQL Server 2005. Когда используется предложение **GPKG**, оно является именем таблицы, которая должна появиться в GeoPackage.

filespec – параметры файла (включая адрес каталога); Это то место, где сохраняется TAB-файл MapInfo;

char_set – имя набора символов; см. раздел [Предложение CharSet](#).

Locale – строка, которая определяет локаль создаваемого индекса, например, "en-US". Если этот параметр отсутствует, то используется системная локаль.

version – выражение, определяющее версию базы данных Microsoft Jet, используемую для новой базы данных. Допускаются значения: 4.0 (для Access 2000), 3.0 (для Access '95/'97), 12.0 (для Access 2007). Если выражение опущено, то используется номер версии 12.0. Если база данных, в которой создается таблица уже существует, то определение версии базы данных игнорируется;

database_filespec – строка, которая определяет имя и DOS-маршрут к доступной базе данных.

- База данных Access, когда используется **Access**. Если такая база не существует, MapInfo Pro создаст новый Access-файл MDB или ACCDB;
- Файл GeoPackage, когда используется **GPKG**.

pwd – пароль на уровне базы данных, определяемый при включении защиты базы данных.

ConnectionNumber – целое значение, номер соединения с базой данных;

version_pro – 100, для таблиц, которые могут читаться ранними версиями MapInfo Pro, или 300 (MapInfo 3.0 формат) для не-Access таблиц. Для таблиц Access версия должна быть равна 410;

Block_size – значение в диапазоне от 0 до 32767. Если этот параметр не задан, то умолчанию используется значение равное 16384. Если значение меньше 512, то оно будет округлено до 512.

commit_keyword – одно из следующих ключевых слов: **NoCollision**, **ApplyUpdates**, **DiscardUpdates**.

Описание

Если предложение **As** не определено оператор **Commit Table** сохраняет любые изменения в таблице. Это аналогично выполнению команды **Сохранить** MapInfo Pro.

Оператор **Commit Table**, который включает предложение **As**, делает то же, что команда **Создать копию**. Предложение **As** может быть использовано для сохранения таблицы под другим именем, в другом месте, или в виде другого типа файла, проекции.

Для сохранения таблицы под новым именем, укажите новое имя в строковой переменной *filespec*. Для сохранения таблицы в другом месте, укажите в начале строковой переменной *filespec* маршрут к файлу.

Дополнительное, необязательное предложение **Type** обозначает формат локальной таблицы: либо родной, нативный формат MapInfo (**NATIVE**), либо расширенный вариант формата MapInfo Extended (**NATIVEX**). Используйте предложение **Type** внутри предложения **As**. Формат **NATIVEX** поддерживает таблицы, размер которых превышает 2GB, и содержащих наборы символов UTF-16 и UTF-16. Когда предложение **Type** не задано, по умолчанию выбирается тип таблицы **NATIVE**.

Дополнительное, необязательное предложение **Charset** задаётся вместе с предложением **Type** и сохраняет определённый набор символов. Параметр *char_set* должен быть строковой константой, такой как "MacRoman" или "WindowsLatin1". Подробнее см. [Предложение CharSet](#).

- Если используется отдельно от предложения **Type**, подразумевается набор символов исходной таблицы. Использование несовместимого набора символов может привести к тому, что символьные данные целевой таблицы будут преобразованы в символы подчёркивания (например, WindowsCyrillic в WindowsLatin1). Предпочтительнее преобразовывать в наборы символов UTF-8 или UTF-16.
- Если предложения **DBF** и **Table** не используются, MapBasic применяет стандартный для аппаратной платформы набор символов, который используется в сеансе работы. Обратите внимание, что оператор **Commit Table** не поддерживает UTF-8 и UTF-16 для DBF.

Предложение **Locale**, используемое совместно с предложением **NATIVEX**, позволяет специфицировать локаль создаваемого индекса, что влияет на сортировку таблицы. Для этого используется строковое значение, например, "en-US". Если этот параметр не задан, то используется стандартная системная локаль. Это предложение применяется только к таблицам NativeX с наборами символов (charset) UTF-8 или UTF-16.

Для сохранения таблицы с использованием других систем координат или проекций, включите предложение **CoordSys** в предложение **As**. Обратите внимание, что только геокодируемые таблицы могут иметь систему координат и проекцию.

Для сохранения запроса используйте тип таблицы QUERY. Могут быть сохранены только запросы, сделанные пользователем через интерфейс, и запросы, которые создает [оператор Run Command](#) в MapBasic. Оператор **Commit Table** создает файлы TAB и QRY.

Предложение **Version** контролирует формат таблицы. Если указать Version 100, MapInfo Pro сохранит таблицу в формате, читаемом ранними версиями MapInfo. Если указать Version 300, MapInfo Pro сохранит таблицу в формате, используемом MapInfo 3.0. Обратите внимание, что объекты типа "Полилиния" и "Область", имеющие более 8,000 узлов и полилинии, состоящие из множества сегментов требуют версию 300. Если пропустить предложение **Version**, то таблица сохранится в формате версии 300.

Примечание: Если приложение MapBasic использует оператор **Commit Table...As**, действующий на таблицу, у которой есть мемо-поля, то эти мемо-поля не сохранятся в новой таблице. Предупреждения об этом на экран выводиться не будет. Если таблица сохраняется в новой таблице с помощью команды MapInfo Pro (**Создать копию**), то MapInfo Pro предупреждает пользователя о потере мемо-полей. Однако, при сохранении новой таблицы из программы MapBasic, предупреждения не будет. Однако, при сохранении новой таблицы из программы MapBasic, предупреждения не будет.

ODBC – определяет копию таблицы, которая сохраняется в базе данных, определенной номером удаленного соединения *ConnectionNumber*;

Предложение **SQLServerSpatial** обеспечивает пространственные данные с типами данных GEOGRAPHY и GEOMETRY.

ConvertDateTime – если в исходной таблице существуют колонки любого из типов Time или Date, то, в зависимости от того, какой тип поддерживает сервер, они будут преобразованы в

DATETIME или **TIMESTAMP**. Однако, этим преобразованием можно управлять, используя специальный параметр *ConvertDateTime*. Если в исходной таблице нет колонок ни типа **Time**, ни **Date**, то этот параметр не используется. Если параметру *ConvertDateTime* присвоено значение **ON** (режим по умолчанию), то колонки типов **Time** или **Date** будут преобразованы в **DATETIME** или **TIMESTAMP**. Если параметру *ConvertDateTime* присвоено значение **OFF**, то преобразований не будет, а операция будет при необходимости отменена. Если параметру *ConvertDateTime* присвоено значение **INTERACTIVE**, то появится диалог, в котором пользователю будет предложено выбрать необходимое действие. Пользователь может либо выбрать конвертирование и тогда преобразование типов будет осуществлено, либо отменить операцию преобразования.

Для всех поддерживаемых серверов необходимо преобразовывать тип времени **Time** (Oracle, PostGIS, SQL Server Spatial, MS SQL Server и Access), а тип даты **Date** необходимо преобразовывать для серверов на основе MS SQL Server и Access.

Примечание: Для серверов MS SQL Server и Access такое ограничение может быть продиктовано соображениями совместимости с предыдущими версиями. Мы советуем использовать тип данных **DATETIME** вместо типа данных **TIMESTAMP**. Если всё ещё используется тип данных **DATETIME**, операция преобразования не будет выполнена.

CoordSys – система координат; см. подробнее раздел **Предложение CoordSys**.

В предложении **BlockSize** можно использовать значение от нуля (0) до 32767. Если этот параметр не задан, то умолчанию используется значение равное 16384. Если значение меньше 512, то оно будет округлено до 512. Как правило, чем больше размер блока, тем выше скорость передачи. Когда выполняется преобразование из формата **NativeX** (MapInfo Extended) в формат **Native** (MapInfo), сохраняется размер блока файла **NativeX**. Пользователи могут задавать новый размер блока при фиксации (выполнении операции commit) из **NativeX** в формат **Native**.

Если в предложении **ConvertObjects** задать значение параметра **ON**, то будут конвертированы все встретившиеся неподдерживаемые объекты в поддерживаемые объекты. Если задать значение **OFF**, то неподдерживаемые объекты конвертироваться не будут. Если такие объекты встретятся, то будет выдано сообщение об ошибке – о том, что таблица не может быть сохранена. Если задать значение параметра **Interactive**, то появится предложение выбрать вариант действий при появлении в таблице неподдерживаемых объектов.

Пример

Следующий пример открывает таблицу **STATES**, затем использует оператор **Commit Table**, чтобы сделать копию этой таблицы под новым именем (**ALBERS**). Необязательное **предложение CoordSys** указывает на то, что таблица **ALBERS** сохранится с равноплощадной проекцией Альберса.

```
Open Table "STATES"
Commit Table STATES
As "ALBERS"
```

```
CoordSys Earth
Projection 9,7, "m", -96.0, 23.0, 20.0, 60.0, 0.0, 0.0
```

Следующий пример иллюстрирует соединение с удаленной базой данных:

```
dim hodb as integer
hodb = server_connect("ODBC", "dlg=1")
Open table "C:\MapInfo\USA"
Commit Table USA
as "c:\temp\as\USA"
Type ODBC Connection hodb Table "USA"
```

См. также:

оператор Rollback

Примеры использования ConvertDateTime

```
Commit Table DATETIME90 As "D:\MapInfo\Data\Remote\DATETIME90CPY.TAB"
Type ODBC Connection 1 Table ""EASYLOADER""."DATETIME90CPY""
ConvertDateTime Interactive
```

```
Server 1 Create Table ""EASYLOADER""."CITY_125AA"" (Field1
Char(10),Field2 Char(10),Field3 Char(10),MI_STYLE Char(254)) KeyColumn
SW_MEMBER ObjectColumn SW_GEOMETRY
or
Server 1 Create Table "EASYLOADER.CITY_125AA" (Field1 Char(10),Field2
Char(10),Field3 Char(10),MI_STYLE Char(254)) KeyColumn SW_MEMBER
ObjectColumn SW_GEOMETRY
Commit Table City_125aa As
"C:\Projects\Data\TestScripts\English\remote\City_125aacpy.tab" Type
ODBC
Connection 1 Table ""EASYLOADER""."CITY_125AACPY""
or
Commit Table City_125aa As
"C:\Projects\Data\TestScripts\English\remote\City_125aacpy.tab" Type
ODBC
Connection 1 Table "EASYLOADER.CITY_125AACPY"
```

Сохранение связанных таблиц

Сохранение связанной таблицы может породить конфликт, поскольку другие пользователи могут редактировать записи в удаленной базе данных. Следующие предложения позволяют контролировать то, что может произойти при конфликте. (Эти предложения не действуют при сохранении обычной таблицы MapInfo.)

Интерактивный

В случае конфликта MapInfo Pro показывает диалог **Разрешение конфликтов**. После успешного выполнения оператора **Commit Table Interactive**, MapInfo Pro показывает диалог обновления.

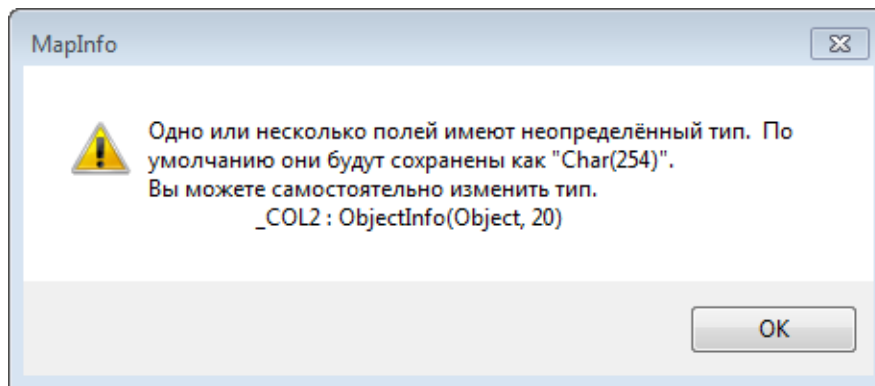
Interactive – при выполнении оператора Commit Table As этот параметр управляет случаями, когда пользователь сохраняет одну или несколько колонок неопределенного типа. Если параметр Interactive задан, то пользователю предьявляется сообщение о наличии колонок неопределенного типа и далее пользователь может выбрать типы для этих колонок. Если Interactive не задан, колонкам неопределенного типа назначается тип Char(254) по умолчанию.

Пример

Создайте запрос в диалоге **SQL-запрос** и нажмите **OK** или введите текст в окне **MapBasic**:

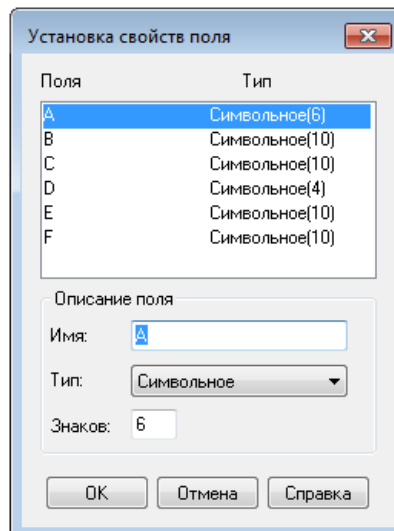
```
Select Highway, objectinfo(obj, 20) from US_HIWAY into Selection
```

Если выбрать текущий запрос и нажать кнопку **Создать копию**, появится сообщение:



Обычно этот диалог содержит список всех колонок, в которых есть неопределенные типы. В этом примере найдена только одна такая колонка.

Нажмите **OK**, для показа диалога **Свойства поля импорта**.



Используя этот диалог, Вы можете задавать тип колонкам. Если есть более одной колонки неопределенного типа, то тип им можно задать за один раз. Тип для колонок, который уже задан, изменять нельзя.

Нажмите **OK** и сохраните настройки.

Automatic NoCollision

В случае конфликта MapInfo Pro не выполняет сохранение. (Этот режим используется по умолчанию, то есть в случае, если не используются предложения Interactive или Automatic.)

Automatic ApplyUpdates

В случае конфликта MapInfo Pro сохраняет значения локальной копии связанной таблицы. (Это аналогично полному игнорированию конфликта.)

Automatic DiscardUpdates

В случае конфликта MapInfo Pro сохраняет значения из РСУБД (локальные изменения отменяются). Можно сохранить копию связанной таблицы, используя предложение **As**, но полученная в результате таблица не будет связанной и не может быть обновлена с сервера.

Соединения ODBC

Длина имени таблицы *tablename* варьируется в зависимости от базы данных. Мы рекомендуем применять 14 или менее символов для имени таблицы, в этом случае гарантирована корректная работа с любой базой данных. В операторе установлена максимально возможная длина имени таблицы, равная 31 символу.

Если используется предложение **As** и тип соединения **ODBC**, то копия таблицы будет сохранена в базе данных, указанной номером соединения *ConnectionNumber* под именем *tablename*. Если исходная таблица имеет географические объекты, то к таблице *tablename*, создаваемой в удаленной базе, могут быть добавлены три колонки: Key, Object и Style, независимо от того, есть такие колонки в исходной таблице или нет. Если исходная таблица не географическая, то только одна колонка Key может быть добавлена к таблице базы данных *tablename*, независимо от того, была ли такая колонка в исходных данных. Ключевая колонка Key будет использоваться для создания уникального индекса.

Пространственный индекс будет создаваться в колонке Object, если она есть. Поддерживаются следующие базы данных: Oracle, SQL Server, IIS (SQL Server Spatial, Universal Server) и Microsoft Access. Однако, для сохранения таблицы с пространственной геометрией/объектами (включая сохранение таблицы только с точечными объектами), потребуется наличие SpatialWare для SQL Server в дополнение к пространственным дополнениям для Oracle. Схема XY не поддерживается в этом операторе.

Сохранение таблицы план-схемы в пакете геоданных GeoPackage

Можно использовать MapBasic-оператор **Commit Table**, чтобы сохранять таблицы план-схем в пакете геоданных GeoPackage. Для этого требуется задать предложение **CoordSys** с обязательно с границами, параметры которых присваиваются во время выполнения оператора.

Примечание: Прежде чем сохранять таблицу, следует получить информацию о предложении **CoordSys** для таблицы план-схемы, например, с помощью

MapBasic-функции **TableInfo()**. Для этого напечатайте `Print TableInfo (tablename, 29)` в окне MapBasic.

Сохранение традиционного MapInfo TAB (Native) или расширенного MapInfo TAB (NativeX) файла

При сохранении традиционной MapInfo *.tab (Native) или расширенной MapInfo *.tab (NativeX) таблицы в DBF, SQLite или в пакете геоданных GeoPackage из MapInfo Pro, нельзя менять набор символов, используемый в целевом файле. Когда сохраняется в пакете геоданных GeoPackage, он записывается в UTF-16, а для других форматов он сохраняется в текущем системном наборе символов, используемом по умолчанию.

Если используется оператор **Commit Table** в MapBasic и требуется сохранить традиционную таблицу MapInfo *.tab (Native) или расширенную таблицу MapInfo *.tab (NativeX) в DBF, Access или пакет геоданных Geopackage, можно предусмотреть необязательное предложение **Charset**. Предложение **Charset** задаётся вместе с предложением **Type** и сохраняет определённый набор символов. Параметр **Charset** требует наличия строковой константы, например, "WindowsLatin1". Подробнее смотрите описание параметра **CharSet**.

Если предложение **Type** отсутствует, MapBasic предполагает набор символов из исходной таблицы. Использование несовместимого набора символов может привести к тому, что символьные данные целевой таблицы будут преобразованы в символы подчёркивания (например, WindowsCyrillic в WindowsLatin1). Предпочтительнее преобразовывать в наборы символов UTF-8 или UTF-16.

Если предложения **DBF** и **Table** не используются, MapBasic применяет стандартный для аппаратной платформы набор символов, который используется в сеансе работы. Обратите внимание, что оператор **Commit Table** не поддерживает UTF-8 и UTF-16 для DBF.

Функция Cond()

Назначение

Эта функция эквивалентна программному оператору `case`: Она сравнивает выражение с каждым значением результатов поиска в определённом порядке. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Cond( expression, search, result
[, search, result ] ... [, Default ] )
```

expression — выражение, которое требуется выполнить.

search — значение, с которым будет сравниваться выражение.

result — результат для значения поиска.

Возвращаемая величина

Если выражение совпадает с аргументом поиска, то возвращается значение результата.

Если совпадения с выражением не найдено, а стандартное условие **Default** не задано, то функция возвращает сообщение об ошибке.

Описание

Функция может выполнять несколько экземпляров поиска относительно соответствующих значений. Все значения параметров *expression*, *search* и *result* необходимы.

Параметр **Default** является дополнительным, необязательным. Он возвращается, когда совпадение не найдено. Если для параметра **Default** не задано значение, то возвращается сообщение об ошибке.

Пример

```
Cond (hire_date, sysdate, "NEW JOINEE", "EMPLOYEE")
```

```
Select State_Name "State", cond(Pop_Density, Pop_Density < 10, "Low  
Density", Pop_Density >= 10 AND Pop_Density < 100, "Medium Density",  
"High Density") "Population Density" from states
```

```
shade 1 with str$(Cond(Pop_Density, Pop_Density < 10, "Low  
Density", Pop_Density > 10 And Pop_Density < 100, "Medium Density", "High  
Density")) values "Low Density" brush(255,255,2), "Medium Density"  
brush(255,16711680,2), "High Density" brush(255,65280,2)
```

См. также:

[функция If\(\)](#)

Функция ConnectObjects()

Назначение

Возвращает объект, представляющий кратчайшее или самое большое расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ConnectObjects( object1, object2, min )
```

object1 и *object2* — выражения, указывающие на объекты.

min — логическое выражение. Если это выражение принимает значение "Да" (TRUE), функция вычисляет минимальное расстояние между объектами, а если выражение принимает значение "Нет" (FALSE), вычисляется максимальное расстояние.

Возвращаемая величина:

Оператор возвращает объект типа "Полилиния", состоящий из одного сегмента и двух точек, и представляющий собой кратчайшее расстояние (*min* == TRUE), или самое большое расстояние (*min* == FALSE) между объектами *object1* и *object2*.

Описание

Одна точка из полученного в результате объекта-полилинии принадлежит объекту — *object1*, а другая точка принадлежит объекту — *object2*. Обратите внимание, что расстояние между двумя исходными объектами можно вычислить с помощью функции **ObjectLen()**. Если существует несколько вариантов кратчайших или наибольших расстояний (например, полученный сегмент представляет одно расстояние, а существуют еще и другие сегменты равной длины), эти функции возвращают лишь один из вариантов. Не существует способа определить, является ли полученный объект полилинии единственным кратчайшим расстоянием.

ConnectObjects() возвращает объект типа полилиния, соединяющая *object1* и *object2* по кратчайшему (*min* == TRUE) или самому дальнему (*min* == FALSE) пути, полученному с использованием сферического метода вычислений. Если применить сферический методы вычислений не удастся, (например, если координатная система MapBasic - план), то используются декартовы вычисления.

См. также:

функция **SphericalConnectObjects()**, функция **CartesianConnectObjects()**, оператор **Nearest**, оператор **Farthest**, функция **CartesianObjectDistance()**, функция **ObjectDistance()**, функция **SphericalObjectDistance()**

Оператор Continue

Назначение

Возобновляет выполнение программы MapBasic (парный **оператор Stop**). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Continue
```

Предупреждение!

Оператор **Continue** используется только в окне **MapBasic** и не может быть частью программы.

Описание

Оператор **Continue** используется для возобновления выполнения программы MapBasic, которую остановил **оператор Stop**.

Оператор **Stop** используется для отладочных целей. Когда программа выполняет **оператор Stop**, она приостанавливается, и в меню **Файл** в окне MapInfo команда **Запустить программу MapBasic** меняется на **Продолжить программу MapBasic**. Продолжить выполнение программы можно, выбрав команду **Файл > Продолжить программу MapBasic**. Можно также ввести в окно **MapBasic** оператор **Continue** – это даст тот же эффект, что и команда **Продолжить программу MapBasic**.

Предложения Control Button / OKButton / CancelButton

Назначение

Отвечают за создание кнопки с текстом в диалоге пользователя (см. **оператор Dialog**).

Синтаксис

```
Control { Button | OKButton | CancelButton }
  [ Position x, y ] [ Width w ] [ Height h ]
  [ ID control_ID ]
```

```
[ Calling handler ]
[ Title title_string ]
[ Disable ] [ Hide ]
```

x, *y* – координаты левого верхнего угла кнопки в окне диалога в диалоговых единицах;

w – ширина кнопки в диалоговых единицах, по умолчанию 40;

h – высота кнопки в диалоговых единицах, по умолчанию 18;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры-обработчика, которая запускается при нажатии на кнопку;

title_string – текст на кнопке.

Описание

Если **оператор Dialog** включает предложение **Control Button**, то в диалоге создаются кнопки с текстом, нажатие на которые приводит к выполнению определенных действий. Если задано **OKButton** в качестве **Button** то создается кнопка подтверждения; нажатие на **OKButton** приводит к закрытию диалогового окна с сохранением всех установленных значений в диалоге. Кнопка типа **CancelButton** создает кнопку, позволяющую закрывать диалог без сохранения значений. Каждый диалог должен содержать не более одной кнопки подтверждения **OKButton** и не более одной кнопки отмены **CancelButton**. **Disable** делает флажок недоступным для выбора (закрашивается серым). **Hide** делает флажок в диалоговом окне изначально скрытым.

Для изменения состояния элемента диалога используйте **оператор Alter Control** (например, для показа скрытой кнопки).

Пример

```
Control Button
Title "&Reset"
Calling reset_sub
Position 10, 190
```

См. также:

оператор Alter Control, **оператор Dialog**

Предложение Control CheckBox

Назначение

Добавляет управление окном документа к диалогу пользователя (см. **оператор Dialog**), который может быть порожденным при использовании интегрированной картографии.

Синтаксис

```
Control CheckBox
[ Position x, y ] [ Width w ]
[ ID control_ID ]
[ Calling handler ]
[ Title title_string ]
[ Value log_value ]
[ Into log_variable ]
[ Disable ] [ Hide ]
```

x, y – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

w – ширина в диалоговых единицах;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры-обработчика, которая запускается при изменении режима флажка;

title_string – текст справа от флажка;

log_value – логическая величина, задающая начальное значение: FALSE – флажок сброшен;

log_variable – имя логической переменной, которой будет присвоено значение элемента после закрытия диалога.

Описание

Если **оператор Dialog** включает предложение **Control CheckBox**, в диалоге создается флажок – элемент, который может иметь только два значения.

Предложение **Value** позволяет присваивать элементу диалога начальное значение. Если предложение **Value** опущено или, наоборот, присутствует и задает начальное значение TRUE, то флажок при открытии диалогового окна будет установлен. Если предложение **Value** задает значение FALSE, то флажок будет сброшен. **Disable** делает флажок недоступным для выбора (закрашивается серым). **Hide** делает флажок в диалоговом окне изначально скрытым.

Пример

```
Control CheckBox
  Title "Include &Legend"
  Into showlegend
  ID 6
  Position 115, 155
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [функция ReadControlValue\(\)](#)

Предложение Control DocumentWindow

Назначение

Часть [оператора Dialog](#)); добавляет управление окном документа диалогу, который может передаваться приложению интегрированной картографии.

Синтаксис

```
Control DocumentWindow
[ Position x, y ]
[ Width w ] [ Height h ]
[ ID control_ID ]
[ Disable ] [ Hide ]
```

x, y – определяют позицию элемента управления в единицах диалога;

w – определяет ширину элемента управления в единицах диалога; стандартная ширина 100;

h – определяет высоту элемента управления в единицах диалога; стандартная высота 100;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

Disable – делает элемент управления изначально неактивным;

Hide – изначально скрывает элемент управления.

Примечание: Это предложение работает только в 32-разрядных версиях MapInfo Pro.

Описание

Если **оператор Dialog** содержит предложение **Control DocumentWindow**, то диалог включает в себя элемент управления окна документа, который может быть восстановлен при помощи **оператора Set Next Document**.

Предложение Control EditText

Назначение

Отвечает за создание текстового окошка ввода в диалоге пользователя (см. **оператор Dialog**).

Синтаксис

```
Control EditText
[ Position x, y ] [ Width w ] [ Height h ]
[ ID control_ID ]
[ Value initial_value ]
[ Into variable ]
[ Disable ] [ Hide ] [ Password ]
```

x, y – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

w – ширина в диалоговых единицах;

h – высота в единицах измерения диалога; если высота превышает 20 единиц, то текст разбивается на несколько строк;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

initial_value – строковая величина, задающая начальное значение;

variable – имя строковой переменной, которой будет присвоен текст из окошка после закрытия диалога кнопкой **OK**.

Ключевое слово **Disable** делает элемент недоступным (закрашивается серым).

Ключевое слово **Hide** прячет элемент из диалогового окна.

Ключевое слово **Password** включает "слепой" режим ввода текста (показывается звездочка вместо каждого введенного пользователем символа), используется для создания окошка ввода пароля.

Описание

Если пользователь вводит текст длиной больше, чем может поместиться в окошко, MapInfo Pro автоматически прокручивает экран и создает место для продолжения текста. Элемент EditText может иметь строковое значение длиной до 32767 символов.

Если высота больше 20 единиц, то окошко будет иметь две и более строк. В этом случае текст, который пользователь введет, будет автоматически разбиваться на строки. Если пользователь желает ввести символ перевода строки (обычно в Windows для этого надо нажать клавиши **Ctrl+Enter**), то в строку текста для элемента EditText нужно добавить `Chr$(10)`. Если в выражение *initial_value* уже вложены вызовы `Chr$(10)`, то в диалоге показывается уже разбитый на строки текст.

Для перемещения фокуса в элемент EditText используйте [оператор Alter Control...Active](#).

Пример

```
Control EditText
Value "Franchise Locations"
Position 65, 8 Width 90
ID 1
Into s_map_title
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [функция ReadControlValue\(\)](#)

Предложение Control GroupBox

Назначение

Отвечает за создание прямоугольной рамки с текстом в диалоге пользователя (см. [оператор Dialog](#)).

Синтаксис

```
Control GroupBox
[ Position x, y ] [ Width w ] [ Height h ]
[ ID control_ID ]
[ Title title_string ]
[ Hide ]
```

x, y – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

w – ширина в диалоговых единицах;

h – высота рамки в единицах измерения диалога;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

title_string – текст заголовка, который начинается от левого верхнего угла рамки.

Ключевое слово **Hide** прячет элемент из диалогового окна.

Пример

```
Control GroupBox
  Title "Level of Detail"
  Position 5, 30
  Height 40 Width 70
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#)

Предложения Control ListBox / MultiListBox

Назначение

Отвечают за создание списков в диалоге пользователя (см. [Оператор Dialog](#)).

Синтаксис

```
Control { ListBox | MultiListBox }
  [ Position x, y ] [ Width w ] [ Height h ]
  [ ID control_ID ]
  [ Calling handler ]
  [ Title { str_expr | From Variable str_array_var } ]
  [ Value i_selected ]
  [ Into i_variable ]
  [ Disable ] [ Hide ]
```

x, *y* – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

w – ширина окошка в единицах измерения диалога, по умолчанию – 80;

h – высота окошка в единицах измерения диалога, по умолчанию – 70;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры-обработчика, которая запускается при изменении пользователем выбора в списке или двойным указанием в списке;

str_expr – строковое выражение, которое задает тексты надписей при переключателях; элементы списка разделены точкой с запятой (;);

str_array_var – имя массива строковых переменных;

i_selected – короткое целое число (SmallInt), задающее номер элемента списка, который будет выбран при открытии диалога, по умолчанию в списке не будет выбрано ни одной строки;

i_variable – имя переменной типа SmallInt (короткое целое), которая будет использоваться для сохранения значения выбора в списке после закрытия диалога.

Ключевое слово **Disable** делает элемент недоступным (закрашивается серым).

Ключевое слово **Hide** прячет элемент из диалогового окна.

Описание

Если **оператор Dialog** включает предложение **Control ListBox**, то в диалог добавляется окошко списка. Если в окошко не вмещаются все строки списка, то окошко снабжается справа полосой прокрутки.

Элемент MultiListBox идентичен элементу ListBox, за исключением того, что в первом пользователь может выбрать несколько элементов при нажатой клавише **SHIFT**.

Предложение **Title** задает строки списка. Если за словом **Title** задан список текстов в кавычках, через точку с запятой, то каждый текст будет определять одну строчку списка. В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title "1st Quarter;2nd Quarter;3rd Quarter;4th Quarter"
```

Также список в предложении **Title** можно задавать массивом. В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title From Variable s_optionlist
```

Чтение значений элемента MultiListBox

Так как в окошке списка этого элемента может быть не выбрано ни одной строки, выбрана одна строка, несколько строк или весь список, то для элемента MultiListBox нужно спроектировать специальную процедуру чтения значений (присваиваемую, например, элементу OKButton). Изнутри процедуры-обработчика для чтения всех значений несколько раз вызывается **функция ReadControlValue()**.

При первом вызове **функция ReadControlValue()** дает номер первой строки, выбранной в списке. При следующем вызове **функция ReadControlValue()** дает номер второй строки и т. д. Когда все значения будут прочитаны, **функция ReadControlValue()** вернет ноль. Если при первом вызове **функция ReadControlValue()** возвратит ноль, то в списке ничего не было выбрано.

Двойной щелчок мышки в списке

Для списка может быть объявлена процедура-обработчик *handler*, которая вызывается всякий раз, когда пользователь указывает на список. Причем это может быть простое указание, а может быть двойное, то есть, если нажать два раза на клавишу мышки. Обычно, двойное указание в списке аналогично простому указанию и последующему нажатию на кнопку **ОК**, которая закрывает диалог с подтверждением всех изменений значений элементов в диалоге.

Пример использования в тексте программы NVIEWS.MB в директории <Your MapBasic Installation Directory>\SAMPLES\MAPBASIC\SNIPPETS.

Для определения, применялось ли пользователем двойное указание или нет, в процедуре обработчика спискового элемента диалога используется **функция CommandInfo()**. Вот пример такой процедуры обработчика:

```
Sub lb_handler
  Dim i As SmallInt
  If CommandInfo(CMD_INFO_DLG_DBL) Then
    ' ... then the user double-clicked.
    i = ReadControlValue( TriggerControl() )
    Dialog Remove
    ' at this point, the variable i represents
    ' the selected list item...
  End If
End Sub
```

Пример

```
Control ListBox
  Title "1st Quarter;2nd Quarter;3rd Quarter;4th Quarter"
  ID 3
  Value 1
  Into i_quarter
  Position 10, 92 Height 40
```

В тексте программы NVIEWS.MB можно видеть, как создается список и как можно применять двойное указание мышкой. Программа NVIEWS показывает диалог с элементом ListBox. Этот диалог закрывается, если пользователь либо выбрал элемент списка и нажал **ОК**, либо если указал на элемент списка дважды.

В этом примере предложение **Control ListBox** добавляет список в диалог **Named Views** ("Именованные виды"). Заметим, что элементу ListBox сопоставлен обработчик "listbox_handler."

```
Control ListBox
  Title desc_list
  ID 1
  Position 10, 20 Width 245 Height 64
  Calling listbox_handler
```

Если пользователь выбирает элемент списка, MapBasic вызывает процедуру-обработчик "listbox_handler". Из процедуры вызывается **функция CommandInfo()** для определения, какой щелчок мышкой был сделан - одиночный или двойной. Если был сделан двойной щелчок, процедура закрывает диалог, используя **оператор Dialog Remove**. Если **оператор Dialog Remove** не сработал, диалог останется на экране, пока пользователь не выберет **ОК** или **Отмена**.

```
Sub listbox_handler
  Dim i As SmallInt
  ' First, since user clicked on the name of a view,
  ' we can enable the OK button and the Delete button.
  Alter Control 2 Enable
  Alter Control 3 Enable
  If CommandInfo(CMD_INFO_DLG_DBL) = TRUE Then
    ' ...then the user DOUBLE-clicked.
    ' see which list item the user clicked on.
    i = ReadControlValue(1) ' read user's choice.
    Dialog Remove
    Call go_to_view(i) ' act on user's choice.
  End If
End Sub
```

MapBasic вызывает процедуру-обработчик и в случае одиночного, и в случае двойного щелчка мышкой. Процедура-обработчик должна различать одиночные и двойные щелчки.

См. также:

оператор Alter Control, **оператор Dialog**, **функция ReadControlValue()**, **функция CommandInfo()**

Предложения Control PenPicker/BrushPicker/SymbolPicker/FontPicker

Назначение

Отвечают за создание кнопок выбора стиля: линии, штриха, символа или шрифта в диалоге пользователя (см. **оператор Dialog**).

Синтаксис

```
Control { PenPicker | BrushPicker | SymbolPicker | FontPicker }
[ Position x, y ] [ Width w ] [ Height h ]
[ ID control_ID ]
[ Calling handler ]
[ Value style_expr ]
[ Into style_var ]
[ Disable ] [ Hide ]
```

x, *y* – координаты левого верхнего угла текста в окне диалога в специальных единицах измерения диалога;

w – ширина кнопки в единицах измерения диалога, по умолчанию 20;

h – высота кнопки в единицах измерения диалога, по умолчанию 20;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры-обработчика; если пользователь нажмет на кнопку определения стиля, а в появившемся диалоге нажмет **OK**, то MapBasic вызовет процедуру *handler*.

style_expr – выражение типа Pen, Brush, Symbol или Font, которое задает начальное значение элемента;

style_var – имя переменной типа Pen, Brush, Symbol или Font (тип переменной должен соответствовать типу элемента диалога).

Ключевое слово **Disable** делает элемент недоступным (закрашивается серым).

Ключевое слово **Hide** прячет элемент из диалогового окна.

Описание

Ключевые слова (PenPicker, BrushPicker, SymbolPicker или FontPicker) позволяют создавать в диалоге кнопку, нажатие на которую открывает стандартный диалог выбора стиля оформления объектов.

Пример

```
Control SymbolPicker
  Position 140,42
  Into sym_storemarker
```

См. также:

оператор Alter Control, оператор Dialog, функция ReadControlValue()

Функция ControlPointInfo()

Назначение

Возвращает растровые и географические координаты контрольных точек из растровой таблицы. Географические координаты возвращаются в текущей в программе MapBasic координатой системе.

Синтаксис

```
ControlPointInfo( table_id, attribute, controlpoint_num )
```

table_id – либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0. Таблица должна быть растровой, поверхностью (grid) или типа WMS.

attribute – целочисленный код, уточняющий тему запроса.

controlpoint_num – целое число, номер опрашиваемой контрольной точки. Нумерация начинаются с 1. Количество контрольных точек можно узнать, выполнив оператор

```
RasterTableInfo(table_id, RASTER_TAB_INFO_NUM_CONTROL_POINTS)
```

Возвращаемая величина

Координаты X или Y на растре возвращаются в виде целого числа (Integer). Географические координаты X или Y возвращаются в виде вещественного (Float) числа. Тип возвращаемого значения задается атрибутом attribute, а номер точки – числом controlpoint_num.

Параметр attribute определяет вид возвращаемой функцией информации. Коды в левом столбце (например, RASTER_CONTROL_POINT_X) определены в файле стандартных определений MAPBASIC.DEF.

attribute-код	ID	ControlPointInfo() возвращает:
RASTER_CONTROL_POINT_X	1	Целое число (Integer), представляющее X-координату контрольной точки номер controlpoint_num.
RASTER_CONTROL_POINT_Y	2	Целое число (Integer), представляющее Y-координату контрольной точки номер controlpoint_num.
GEO_CONTROL_POINT_X	3	Вещественное (Float) число, представляющее X-координату контрольной точки номер controlpoint_num.

attribute-код	ID	ControlPointInfo() возвращает:
GEO_CONTROL_POINT_Y	4	Вещественное (Float) число, представляющее Y-координату контрольной точки номер controlpoint_num.
TAB_GEO_CONTROL_POINT_X	5	Вещественное (Float) число, представляющее X-координату контрольной точки с номером заданным controlpoint_num, которое хранится в TAB-файле изображения
TAB_GEO_CONTROL_POINT_Y	6	Вещественное (Float) число, представляющее Y-координату контрольной точки с номером заданным controlpoint_num, которое хранится в TAB-файле изображения

Предложение Control PopupMenu

Назначение

Отвечает за создание раскрывающегося меню в диалоге пользователя (см. [оператор Dialog](#)).

Синтаксис

```
Control PopupMenu
[ Position x, y ]
[ Width w ]
[ ID control_ID ]
[ Calling handler ]
[ Title { str_expr | From Variable str_array_var } ]
[ Value i_selected ]
[ Into i_variable ]
[ Disable ]
```

x, y – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

w – ширина элемента в единицах измерения диалога, по умолчанию 20;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры, вызываемой, если пользователь выбрал пункт меню.

str_expr – строковое выражение, которое задает тексты надписей при переключателях; элементы списка разделены точкой с запятой (;);

str_array_var – имя массива строковых переменных;

i_selected – короткое целое число (SmallInt), задающее номер переключателя, который будет выбран при открытии диалога, при задании единицы (1) или по умолчанию будет выбран первый переключатель;

i_variable – имя переменной типа SmallInt (короткое целое), которая будет использоваться для сохранения номера выбранного элемента после закрытия диалога.

Ключевое слово **Disable** делает элемент недоступным (закрашивается серым).

Описание

Если **оператор Dialog** включает предложение **Control PopupMenu**, в диалоговом окне появится элемент в виде раскрывающегося меню. Такое меню можно открыть и выбрать из списка одно значение (пункт). При открытии диалогового окна показывается только тот пункт меню, который активен.

Если пользователь указывает на него мышкой, меню открывается полностью и можно выбрать другой пункт.

Предложение **Title** задает список элементов меню. Если за словом **Title** задан список текстов в кавычках, через точку с запятой, то каждый текст будет определять одну строку списка. В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title "Town;County;Territory;Region;Entire state"
```

Список в предложении **Title** можно также задавать массивом.

В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title From Variable s_optionlist
```

Пример

```
Control PopupMenu
  Title "Town;County;Territory;Region;Entire state"
  Value 2
  ID 5
  Into i_map_scope
  Position 10, 150
```

См. также:

оператор Alter Control, оператор Dialog, функция ReadControlValue()

Предложение Control RadioGroup

Назначение

Отвечает за создание кнопок переключателя в диалоге пользователя (см. [оператор Dialog](#)).

Синтаксис

```
Control RadioGroup
[ Position x, y ]
[ ID control_ID ]
[ Calling handler ]
[ Title { str_expr | From Variable str_array_var } ]
[ Value i_selected ]
[ Into i_variable ]
[ Disable ] [ Hide ]
```

x, y – координаты левого верхнего угла элемента в окне диалога в специальных единицах измерения диалога;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

handler – имя процедуры-обработчика, которая запускается при выборе пользователем другого переключателя списке или двойным указанием на другой переключатель;

str_expr – строковое выражение, которое задает тексты надписей при переключателях; элементы списка разделены точкой с запятой (;);

str_array_var – имя массива строковых переменных;

i_selected – короткое целое число (SmallInt), задающее номер переключателя, который будет выбран при открытии диалога, при задании единицы (1) или по умолчанию будет выбран первый переключатель;

i_variable – имя переменной типа SmallInt (короткое целое), которая будет использоваться для сохранения номера выбранного элемента после закрытия диалога.

Ключевое слово **Disable** делает элемент недоступным (закрашивается серым).

Ключевое слово **Hide** прячет элемент из диалогового окна.

Описание

Если [оператор Dialog](#) включает предложение **Control RadioGroup**, в диалоге появляется группа переключателей. Каждый переключатель состоит из надписи и кружочка. Выбранный

переключатель сопровождается заполненным кружочком. Выбранным может быть только один переключатель.

Предложение **Title** задает надписи при переключателях. Если за словом **Title** задан список текстов в кавычках, через точку с запятой, то каждый текст будет определять одну строку списка.

В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title "&Full Details;&Partial Details"
```

Список в предложении **Title** можно также задавать массивом. В следующем фрагменте за предложением **Title** следует строковый массив:

```
Title From Variable s_optionlist
```

Пример

```
Control RadioGroup
  Title "&Full Details;&Partial Details"
  Value 2
  ID 2
  Into i_details
  Calling rg_handler
  Position 15, 42
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [функция ReadControlValue\(\)](#)

Предложение Control StaticText

Назначение

Отвечает за создание текстового элемента в окне диалога, (неизменяемого текста) в диалоге пользователя (см. [оператор Dialog](#)).

Синтаксис

```
Control StaticText
[ Position x, y ]
[ Width w ] [ Height h ]
[ ID control_ID ]
```



```
[ Title title_string ]
[ Hide ]
```

x , y – координаты левого верхнего угла текста в окне диалога в специальных единицах измерения диалога;

w – ширина подписи в единицах измерения диалога;

h – высота подписи в единицах измерения диалога;

control_ID – целое; не может совпадать с идентификаторами других элементов управления в диалоге;

title_string – текст.

Ключевое слово **Hide** прячет элемент из диалогового окна.

Описание

Если требуется, чтобы текст в диалоге занимал несколько строк, то задайте для него прямоугольник, используя предложения **Width** и **Height**. Если предложения **Width** и **Height** не использовать, то текст будет состоять только из одной строки.

Пример

```
Control StaticText
  Title "Enter map title:"
  Position 5, 10
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#)

Функция ConvertToPline()

Назначение

Возвращает полилинию, которая аппроксимирует заданный объект. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ConvertToPline( object )
```

object – объект любого типа, кроме текстового и точечного.

Возвращаемая величина

Полилиния. Величина типа Object.

Описание

Функция **ConvertToPline()** возвращает объект типа "Полилиния", которая аппроксимирует объект object. Так, если object задает область, то функция **ConvertToPline()** вернет ломаную линию, представляющую границу области с тем же количеством узлов.

Результат функции **ConvertToPline()** будет таким же как при выполнении команды MapInfo Pro **ОбъектыПреобразовать в полилинии**. Отличие в том, что функция **ConvertToPline()** создает новый объект, не меняя исходный.

См. также:

оператор Objects Enclose

Функция ConvertToRegion()

Назначение

Возвращает область по форме заданного объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ConvertToRegion( object )
```

object – объект любого типа, кроме текстового и точечного.

Возвращаемая величина

Область – величина типа Object.

Описание

При выполнении этой операции сохраняются все значения стилей оформления объекта. Недостающие атрибуты используются по текущим значениям стилей. Если первый узел совпадает с последним, то при преобразовании полилинии они сливаются. Если первый и последний узлы полилинии не совпадают, то при преобразовании в область добавляется последний узел, совпадающий с первым.

Функция **ConvertToRegion()** возвращает область, имеющую такую же форму, какую имеет исходный объект. Так, если исходный объект – прямоугольник, то **ConvertToRegion()** вернет область прямоугольной формы.

Результат функции **ConvertToPline()** будет таким же, как при выполнении команды MapInfo Pro **Превратить в области** на вкладке **ОБЪЕКТЫ**. Отличие в том, что **ConvertToRegion()** создает новый объект, не меняя исходный.

См. также:

[оператор Objects Enclose](#)

Функция ConvexHull()

Назначение

Возвращает объект регион, который представляет собой полигон-контур, основанный на узлах входных объектов. Такой полигон можно получить, если натянуть резинку, охватывающую все узлы. Он содержит минимальное количество точек (таким образом, что все точки исходных объектов лежат на границах или внутри полигона). Полигон получается выпуклым – то есть все внешние углы созданного полигона-контура больше, чем 180 градусов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ConvexHull( inputobject )
```

inputobject – объектное выражение (выражение, результат которого есть величина типа Object);

Возвращаемая величина

Область. Величина типа Object.

Описание

Функция **ConvexHull()** возвращает регион, представляющий контур, охватывающий точки входного объекта. Функция **ConvexHull()** создает один контур за один раз. Чтобы создать контуры вокруг нескольких объектов, используйте оператор Create Object As ConvexHull.

Пример

Следующий пример выбирает штат New York из таблицы штатов, затем создает полигон-контур вокруг выборки.

```
Dim Resulting_object as object
select * from States
where State_Name = "New York"
Resulting_object = ConvexHull(selection.obj)
Insert Into States(obj) Values (Resulting_object)
```

См. также:

[оператор Create Object](#)

предложение CoordSys

Назначение

Задаёт систему координат. Если предложение **CoordSys** отсутствует, то предполагается, что данные приведены в проекции Широта/Долгота. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Syntax1
CoordSys Earth
[ Projection type,
  datum,
  unitname
  [ , origin_longitude ]
  [ , origin_latitude ]
  [ , standard_parallel_1 [ , standard_parallel_2 ] ]
  [ , azimuth ]
  [ , scale_factor ]
  [ , false_easting ]
  [ , false_northing ]
  [ , range ] ]
[ Affine Units unitname, A, B, C, D, E, F ]
[ Bounds ( minx, miny) ( maxx, maxy ) ]
```

Синтаксис2

```
CoordSys Nonearth
[ Affine Units unitname, A, B, C, D, E, F ]
```

```
Units unitname
Bounds ( minx, miny ) ( maxx, maxy )
```

Синтаксис3

```
CoordSys Layout Units paperunitname
```

Синтаксис4

```
CoordSys Table tablename
```

Синтаксис5

```
CoordSys Window window_id
```

type — положительное целое число, показывающее, какая координатная система используется;

datum — положительное целое число, указывающее какой регион используется;

unitname — строковая величина, показывающая, какие единицы измерения используются (например, “m” для метров); полный список названий единиц измерения смотрите в разделе Настройка единиц измерения;

origin_longitude — вещественное число в градусах, показывающее долготу;

origin_latitude — вещественное число в градусах, показывающее широту;

standard_parallel_1 и *standard_parallel_2* — вещественные значения широты, в градусах;

azimuth — вещественная величина, угол измерения в градусах;

scale_factor — вещественная величина, масштабный множитель;

range — вещественная величина, от 1 до 180, предписывающая, какая часть земной поверхности будет показана;

minX — вещественная величина, определяющая минимальное значение X;

miny — вещественная величина, определяющая минимальное значение y;

maXX — вещественная величина, определяющая максимальное значение X;

maXy — вещественная величина, определяющая максимальное значение y;

paperunitname — строковая величина, представляющая “бумажные” единицы измерения (например, “in” для дюймов);

tablename — имя открытой таблицы;

window_id — целая величина, идентификатор окна, соотнесенного с окном карты или отчёта;

A представляет масштабирование или растяжение вдоль оси X.

B представляет поворот или отклонение вдоль оси X.

C представляет сдвиг вдоль оси X.

D представляет масштабирование или растяжение вдоль оси Y.

E представляет поворот или отклонение вдоль оси Y.

F представляет сдвиг вдоль оси Y.

Описание

Все координаты приводятся относительно северо-восточного квадранта. Координаты объектов на территории США имеют отрицательную координату X, а координаты объектов в России и Европе (к востоку от Гринвича) имеют положительные координаты по оси X.

См. также:

[проекции CoordSys](#) и [проекции план-схемы](#), [единицы измерения отчёта CoordSys](#), [CoordSys параметр таблицы](#), [CoordSys параметр окна](#)

Мировые проекции и проекции плана в предложении CoordSys

Синтаксис 1 (географическая проекция)

```
CoordSys Earth
[ Projection type, datum, unitname
  [ , origin_longitude ] [ , origin_latitude ]
  [ , standard_parallel_1 [ , standard_parallel_2 ] ]
  [ , azimuth ] [ , scale_factor ]
  [ , false_easting ] [ , false_northing ]
  [ , range ] ]
[ Affine Units unitname, A, B, C, D, E, F ]
[ Bounds ( minx, miny ) ( maxx, maxy ) ]
```

Синтаксис 2 (проекция плана)

```
CoordSys Nonearth
[ Affine Units unitname, A, B, C, D, E, F ]
Units unitname
[ Bounds ( minx, miny ) ( maxx, maxy ) ]
```

type – положительное целое число, представляющее тип координатной системы;

datum – положительное целое число, определяющее референс-эллипсоид;

unitname – единица измерения расстояний, строковая величина (например, "m" – метры, список единиц приведен в разделе: [Оператор Set Distance Units](#)).

origin_longitude – долгота точки отсчета (нулевой точки) в градусах, вещественное число (тип Float);

origin_latitude – широта точки отсчета (нулевой точки) в градусах, вещественное число (тип Float);

standard_parallel_1 и *standard_parallel_2* – стандартные широты в градусах, вещественное число (тип Float);

azimuth – азимутальный угол в градусах, вещественное число (тип Float);

scale_factor – коэффициент искажения (масштабный коэффициент), вещественное число (тип Float);

range – вещественное число от 1 до 180 (тип Float), определяющее, какая часть Земли видна;

minx – минимальная X-координата, вещественное число (тип Float);

miny – минимальная Y-координата, вещественное число (тип Float);

maxx – максимальная X-координата, вещественное число (тип Float);

maxy – максимальная Y-координата, вещественное число (тип Float);

A представляет масштабирование или растяжение вдоль оси X.

B представляет поворот или отклонение вдоль оси X.

C представляет сдвиг вдоль оси X.

D представляет масштабирование или растяжение вдоль оси Y.

E представляет поворот или отклонение вдоль оси Y.

F представляет сдвиг вдоль оси Y.

Описание

Предложение **CoordSys** задает координатную систему, а также может дополнительно задавать проекцию для этой системы. Помните, что предложение **CoordSys** не является отдельным оператором. Предложение **CoordSys** может входить в состав разных операторов, например **оператор Set Map** может включать **CoordSys**, и в этом случае **оператор Set Map** переопределяет проекцию, используемую в соответствующем окне карты.

Используйте синтаксис **CoordSys Earth**, чтобы явно задавать координатные системы для карты Мира (то есть, карты, на которой координаты задаются в зависимости их положения на Земном шаре). Параметры предложения **Projection** задают проекцию карты (если она есть), и должны использоваться в соответствии с данной координатной системой. Если предложение **Projection** опущено, MapBasic использует референц-эллипсоид (датум) под номером 0. Предложение **Affine** описывает аффинное преобразование, используемое для трансформации координатной системы. Если предложение **Projection** опущено, то базовой назначается координатная система Долгота/Широта. Так как порожденные координаты могут быть представлены в других единицах, чем базовые, в предложении **Affine** требуется указать единицы для порожденной системы.

Используйте синтаксис **CoordSys Nonearth**, чтобы явно задавать координатные системы для планов и схем, таких как планы для помещений или конструкторских чертежей. В случае, когда

предложение **CoordSys** описывает не мировую систему, базовой координатной системой можно назначить любую декартову сеть. Предложение **Units** задает единицы измерения базовой координатной системы, а предложение **Affine** задает единицы измерения производной координатной системы.

Если предложение **CoordSys** использует **оператор Set Map** или **оператор Set Digitizer**, то предложение **Bounds** игнорируется. Если задано предложение **CoordSys**, предложение **Bounds** нужно обязательно использовать для карт в проекции План-Схема, а в остальных случаях предложение **Bounds** используется для повышения точности нанесения объектов на карту.

Предложение **Bounds** задает границы, в которых показывается карта. Объект не может быть создан за пределами заданных этим предложением границ. Если задается координатная система карты (Earth), то необходимо опустить предложение **Bounds**, так как MapInfo Pro пытается по умолчанию охватить всю Землю.

Примечание: Используя **оператор Create Map**, можно увеличить точность координат на карте, задав более узкий охват предложением **Bounds**.

Каждая картографическая проекция задается уравнением, имеющим свой индивидуальный набор параметров. Поэтому предложение **CoordSys** может иметь разный набор параметров в предложении **Projection**. Например, уравнение, задающее проекции Робинсона, использует параметры: *datum* (референс-эллипсоид), *unitname* (единицы измерения) и *origin_latitude* (начальную широту), а уравнение для модифицированной проекции Меркатора - *datum* (референс-эллипсоид), *unitname* (начальную широту), *origin_longitude* (начальную широту), *origin_latitude* (начальную долготу), *scale_factor* (коэффициент искажения), *false_easting* (восточное смещение) и *false_northing* (северное смещение).

Для более подробной информации о проекциях и координатных системах смотрите документацию MapInfo Pro.

Каждая программа MapBasic может иметь свои режимы предложения **CoordSys**, задающие индивидуальные координатные системы. Если программа MapBasic выполняет **оператор Set CoordSys**, на другие действующие программы MapBasic это не оказывает никакого эффекта.

Примеры

Допустим, режим представления существующей карты задает **оператор Set Map**. В приведенной ниже форме **оператор Set Map** задает режим показа карты в проекции Робинсона:

```
Set Map CoordSys Earth Projection 12, 12, "m", 0.
```

Первое число 12 определяет проекцию Робинсона; второе число 12 определяет применение этой проекции ко всему Земному шару; параметр "m" задает метры, как единицы измерения; ноль задает нулевую долготу как начальную.

Следующий оператор задает показ карты без проекций:

```
Set Map CoordSys Earth
```


В следующем примере открывается таблица WORLD, затем выполняется **оператор Commit Table**, сохраняющий ее под именем RWORLD. Новая таблица RWORLD сохраняется в проекции Робинсона.

```
Open Table "world" As World
Table world As "RWORLD.TAB"
CoordSys Earth Projection 12, 12, "m", 0.
```

Следующий пример определяет систему координат DCS, которая получена из системы координат UTM Zone 10 путем аффинного преобразования:

```
x1 = 1.57x - 0.21y + 84120.5
y1 = 0.19x + 2.81y - 20318.0
```

Здесь координаты (x1 , y1) представляют полученные координаты для DCS, а (x, y) исходные координаты UTM Zone 10. Если координаты DCS исчислялись в футах, то предложение **CoordSys** для DCS выглядело бы следующим образом:

```
CoordSys Earth
Projection 8, 74, "m", -123, 0, 0.9996, 500000, 0
Affine Units "ft", 1.57, -0.21, 84120.5, 0.19, 2.81, -20318.0
```

Единицы измерения CoordSys для отчетов

Синтаксис

```
CoordSys Layout Units paperunitname
```

paperunitname – строковая величина, представляющая имя "бумажной" единицы, например, "in" – дюймы, список единиц приведен в разделе: Оператор **Set Paper Units**.

Описание

Используйте синтаксис **CoordSys Layout** для задания координатной системы в окне отчета MapInfo Pro. Программа MapBasic должна выполнить оператор **Set CoordSys Layout** перед тем, как создавать объекты отчета и работать с ними. Параметр *unitname* задает "бумажные" единицы измерения, например, "in" для дюймов или "cm" для сантиметров.

Примеры

Например, следующий **оператор Set CoordSys** определяет дюймы, как единицы измерений в окне отчета:

```
Set CoordSys Layout Units "in"
```

Инструкция CoordSys для таблиц

Синтаксис

```
CoordSys Table tablename
```

tablename – имя открытой таблицы.

Описание

Используйте синтаксис **CoordSys Table** для определения координатной системы, в которой карта была сохранена.

Форма CoordSys для окон

Синтаксис

```
CoordSys Window window_id
```

window_id – целочисленный идентификатор окна карты или отчета.

Описание

Используйте синтаксис **CoordSys Window** для определения координатной системы, используемой в данном Окне.

Примеры

В следующем примере проекция в одном окне карты устанавливается такой же как в другом. В этом примере подразумевается, что целочисленные переменные *first_map_id* и *second_map_id* уже содержат идентификаторы двух окон карт.

```
Set Map
Window second_map_winid
CoordSys Window first_map_winid
```

См. также:

[оператор Commit Table](#), [оператор Set CoordSys](#), [оператор Set Map](#)

Функция CoordSysName\$()

Назначение

Возвращает строку с названием системы координат из описания системы координат MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CoordSysName$ ( string )
```

Возвращаемая величина

Строка

Пример

```
Note CoordSysName$("Coordsys Earth Projection 1, 62")
```

Показывает в диалоге MapInfo следующую строку:

```
Longitude / Latitude (NAD 27 for Continental US)
```

Примечание: Если системы координат с таким названием не существует в файле MAPINFO.PRJ, например, когда план-схема дана в геодезических футах, то эта функция возвратит пустую строку.

```
Note CoordSysName$("CoordSys NonEarth Units " + ""survey ft"" +  
"Bounds (0, 0) (10, 10)")
```

Если параметр CoordSys передан некорректно (заданы неправильные единицы измерения):

```
Note CoordSysName$("CoordSys Earth Projection 3, 74, " + ""foo"" +  
"-90, 42, 42.7333333333, 44.0666666667, 1968500, 0")
```

то будет возвращена ошибка о неправильной системе координат (Ошибка #727).

```
Invalid Coordinate System: CoordSys Earth Projection <content>
```

Функция CoordSysStringToEPSG()

Назначение

Преобразует предложение CoordSys MapBasic в целое значение EPSG для дальнейшего использования в функциях и операторах MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CoordSysStringToEPSG ( coordsys_string )
```

prj_string – это MapBasic предложение CoordSys. Значение EPSG (European Petroleum Survey Group) является целым числом; например, предложение CoordSys "Earth Projection 1, 104" будет преобразовано в код EPSG 4326. Полный список кодов EPSG, используемых MapInfo Pro, можно найти в файле MAPINFOW.PRJ, в каталоге в который была установлена программа MapInfo Pro. Коды EPSG можно определить по префиксу "p", за которым следует число.

Возвращаемая величина

Целое – Integer. Если значение EPSG не найдено, возвращает -1.

Описание

Функция CoordSysStringToEPSG() преобразует MapBasic-предложение CoordSys в целое значение EPSG.

Пример

Следующий пример вычисляет код EPSG, соответствующий координатной системе Earth Projection 1, 104.

```
print CoordSysStringToEPSG("Earth Projection 1, 104")
```

См. также:

предложение CoordSys

Функция CoordSysStringToPRJ()

Назначение

Преобразует предложение CoordSys MapBasic в строку PRJ-файла. Строки в формате PRJ используются для описания координатных систем MapInfo в файле MAPINFOW.PRJ. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CoordSysStringToPRJ$ ( coordsys_string )
```

prj_string – это MapBasic предложение CoordSys. Строка PRJ представляет собой описание координатной системы в файле MAPINFOW.PRJ; например, предложение CoordSys "Earth Projection 1, 104" соответствует PRJ-строке "1,104".

Возвращаемая величина

строка (string)

Описание

Функция CoordSysStringToPRJ() преобразует MapBasic-**предложение CoordSys** в целое значение EPSG.

Пример

Следующий пример вычисляет строку для PRJ-файла, соответствующую координатной системе "Earth Projection 1, 104".

```
print CoordSysStringToPRJ$("Earth Projection 1, 104")
```

См. также:

предложение CoordSys

Функция CoordSysStringToWKT()

Назначение

Преобразует **предложение CoordSys** MapBasic в строку WKT(Well-Known Text). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CoordSysStringToWKT$( coordsys_string )
```

wkt_string – это предложение CoordSys MapBasic.

Возвращаемая величина

WKT-строка Если WKT-строка не найдена, возвращается пустая строка.

Пример

После выполнения следующего оператора:

```
Print coordsysstringtowkt$("CoordSys Earth Projection 8, 74, " + ""m""
+
", -123, 0, 0.9996, 500000, 0 Affine Units " + ""ft"" + ", 1.57, -0.21,
84120.5, 0.19, 2.81, -20318.0")
```

возвращает WKT-строку:

```
PROJCS["_MI_0",GEOGCS[ ,DATUM["North_American_Datum_1983",SPHEROID["Geodet
ic Reference System of
1980",6378137,298.2572221009113],AUTHORITY["EPSG","6269"]],PRIMEM["Greenw
ich",0],UNIT["degree",0.0174532925199433]],PROJECTION["Transverse_Mercato
r"],PARAMETER["latitude_of_origin",0],PARAMETER["central_meridian",-
123],PARAMETER["scale_factor",0.9996],PARAMETER["false_easting",500000],P
ARAMETER["false_northing",0],UNIT["METER",1]]
```

См. также:

предложение CoordSys

Функция Cos()

Назначение

Вычисляет косинус числа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Cos ( num_expr )
```

num_expr – численное выражение угла в радианах.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Cos()** вычисляет косинус числа, полученного в результате вычисления выражения *num_expr*. Функция **Cos()** возвращает значения в диапазоне от 1 до -1.

Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG.

Примечание: Программа должна содержать оператор Include "MAPBASIC.DEF" чтобы использовать значения DEG_2_RAD или RAD_2_DEG.

Пример

```
Include "MAPBASIC.DEF"  
Dim x, y As Float  
x = 60 * DEG_2_RAD  
y = Cos(x)  
  
' y will now be equal to 0.5  
' since the cosine of 60 degrees is 0.5
```

См. также:

[функция Acos\(\)](#), [функция Asin\(\)](#), [функция Atn\(\)](#), [функция Sin\(\)](#), [функция Tan\(\)](#)

Оператор Create Adornment

Назначение

Создает и показывает элементы оформления карты (такие как Масштабная линейка). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Adornment
From Window map_window_id
Type adornment_type
[ Position {
  [ Fixed [ ( x, y ) [ Units paper_units ] ] ] |
  [ win_position [ Offset (x, y) ] [ Units paper_units ] ]
} ]
[ Layout Fixed Position { Frame | Geographic } ]
[ Size [ Width win_width ] [ Height win_height ] [ Units paper_units ]
]
[ Background [ Brush ... ] [ Pen ... ] ]
[ < SCALEBAR CLAUSE > ]

```

где *SCALEBAR CLAUSE*:

```

[ BarType type ]
[ Ground Units distance_units ]
[ Display Units paper_units ]
[ BarLength paper_length ]
[ BarHeight paper_height ]
[ BarStyle [ Pen .... ] [ Brush ... ] [ Font ... ] ]
[ Scale [ { On | Off } ] ]
[ Auto [ { On | Off } ] ]

```

Параметр *adornment_type* может принимать значение **scalebar** (масштабной линейки).

Значения (*x*, *y*) в предложении **Fixed** задают смещение элемента оформления от левого верхнего угла карты, для которого смещения равно (0, 0). При этом способе разметки элементов оформления, элемент оформления сохранит свое положение на карте при изменении размера карты. Например, положение в точке (3, 3) дюйма на карте может быть в нижнем правом углу карты крупного масштаба или в центре карты мелкого масштаба. При изменении размера карты, положение элемента оформления будет стремиться сохраниться внутри выведенной на экран карты.

paper_units – по умолчанию "бумажные" единицы измерения MapBasic. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#).

win_position – задает один из следующих кодов (описаны в файле MAPBASIC.DEF).

```
ADORNMENT_INFO_MAP_POS_TL (0)
ADORNMENT_INFO_MAP_POS_TC (1)
ADORNMENT_INFO_MAP_POS_TR (2)
ADORNMENT_INFO_MAP_POS_CL (3)
ADORNMENT_INFO_MAP_POS_CC (4)
ADORNMENT_INFO_MAP_POS_CR (5)
ADORNMENT_INFO_MAP_POS_BL (6)
ADORNMENT_INFO_MAP_POS_BC (7)
ADORNMENT_INFO_MAP_POS_BR (8)
```

Смещение (*x*, *y*) в предложении **Offset** отсчитывается от выбранной позиции. Например, если *win_position* равно ADORNMENT_INFO_MAP_POS_TL (верхний левый угол), то *x* отсчитывается вправо, а *y* вниз. Если *win_position* равно ADORNMENT_INFO_MAP_POS_BR, то *x* отсчитывается влево, а *y* вверх. В позиции "центр-право" (ADORNMENT_INFO_MAP_POS_CL) и "центр-лево" (ADORNMENT_INFO_MAP_POS_CR) значение *y* игнорируется. В центральной позиции (ADORNMENT_INFO_MAP_POS_CC) игнорируются оба значения смещения. В позиции "верх-центр" (ADORNMENT_INFO_MAP_POS_TC) и "низ-центр" (ADORNMENT_INFO_MAP_POS_BC) игнорируется значение *x*. О значениях ADORNMENT_INFO_MAP_POS_... см. выше, в описании переменной *win_position*.

Переменные *win_width* и *win_height* определяют размер элемента оформления. В случае масштабной линейки MapInfo Pro игнорирует эти параметры, так как размер масштабной линейки задается переменной *BarLength*.

win_position – задает один из следующих кодов (описаны в файле MAPBASIC.DEF).

```
SCALEBAR_INFO_BARTYPE_CHECKEDBAR (0)
SCALEBAR_INFO_BARTYPE_SOLIDBAR (1)
SCALEBAR_INFO_BARTYPE_LINEBAR (2)
SCALEBAR_INFO_BARTYPE_TICKBAR (3)
```



0 Шашки, 1 Промежутки, 2 Линии или 3 Риски

distance_units – единица измерения, которые представляет масштабная линейка:

distance value	Представляет
"ch"	чейны
"cm"	сантиметры

distance value	Представляет
"ft"	футы (иногда называется "международным футом"; один "международный фут" равен точно 30,48 см)
"in"	дюймы
"km"	километры
"li"	линии
"m"	метры
"mi"	мили
"mm"	миллиметры
"nmi"	морские мили (1 морская миля = 1852 метрам)
"rd"	роды
"survey ft"	геодезический фут США геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам) геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам)
"yd"	ярды

Параметр *paper_length* в переменной *paper_units* задает длину линейки в отчете. Можно задавать максимальную длину линейки в 34 дюйма или 86.3 см на распечатке.

Параметр *paper_height* в переменной *paper_units* высоту линейки в отчете. Установите высоту элемента оформления, максимум 44 дюйма или 111.76 см на печатной карте.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цитеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Описание

Масштабная линейка имеет длину *paper_length* в единицах *paper_units*.

Предложение **Position** может либо принимать значение **Fixed** и тогда элемент оформления будет помещен в левый верхний угол окна карты, либо значение будет отсчитываться относительно некой точки в окне карты, заданной переменной *win_position*.

Offset – расстояние, на которое элемент оформления будет смещен на карте в одном из заданных положений *win_position*.

Предложение **Layout Fixed Position** задает **фиксированное** положение элемента оформления на листе отчета. Если это параметр **Geographic**, то элемент оформления будет помещен в тоже положение на листе отчета, которое этот элемент оформления занимает на карте, выведенной на экран. Если размеры рамки отчета будут изменены, то элемент оформления изменит свое положение относительно границ рамки, сохраняя свои географические координаты. Если значение этого параметра **Frame**, то элемент оформления останется в заданном относительно рамки положении, подобно описанному выше в разделе о параметре **Position**. Если в предложении **Position** элемент оформления помещен в точку с координатами (1.0, 1.0) дюймов, то этот элемент оформления будет помещен в точку на дюйм правее и ниже верхнего левого угла рамки. Изменение размера рамки не приводит к изменению положения элемента оформления относительно рамки. По умолчанию используется значение параметра **Geographic**.

Offset – расстояние, на которое элемент оформления будет смещен на карте в одном из заданных положений *win_positions*.

Предложение **Background** с параметром **Brush** задает фоновую штриховку, используемую при создании или изменении масштабной линейки.. Параметр **Pen** в этом же предложении задает стиль контура при прорисовке фона.

Brush – корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение создает фон для элемента оформления.

Pen – корректное **предложение Pen**. Из-за особенностей механизма прорисовки окна элемента оформления в окне карты, линии шириной иной, чем 1 пиксель могут быть показаны некорректно. Кроме того, стили линий, которые на являются непрозрачными, могут показываться некорректно. Это предложение задает стиль рамки элемента украшения.

Font – корректное **предложение Font**.

Если в предложении **Auto** задано значение **On**, то значения, показываемые в масштабной линейке, будут автоматически округляться. Если задано **Off**, то эти значения округляться не будут, как это было в прошлых версиях. Стандартное значение – **Auto Off**.

Можно задать режим **Scale On** для показа дробного представления масштаба. В MapInfo Pro, масштабная линейка обычно не показывает числового представления масштаба, такого как 1:63,360 или 1:1,000,000, называемого **картографическим масштабом**

Пример

Если *paper_length* равно 1, в *paper_unit* измеряется в дюймах, то длина масштабной линейки равна 1 дюйму. В ней имеется надпись, показывающая соотношение *distance_unit* и *paper_unit*, и эта надпись пересчитывается при изменении окна карты (увеличение или сдвиг). Стандартное значение *distance_unit* – это стандартная единица измерения для окна карты. Параметр *paper_height* определяет высоту линейки. Предложения **Pen** и **Brush** задают оформление линейки, а **Font** задает стиль текста в ней. Параметр *Scale* задает показ картографического масштаба.

Следующий пример демонстрирует задание стандартных режимов:

```
create adornment
  from window 261763624
  type scalebar
  position 6 offset (0.000000, 0.000000) units "in"
  background Brush (2,16777215,16777215) Pen (1,2,0)
  bartype 0 ground units "mi" display units "in"
  barlength 1.574803 barheight 0.078740
  barstyle Pen (1,2,0) Brush (2,0,16777215) Font ("Arial",0,8,0)
  scale on
```

См. также:

[Оператор Set Adornment](#)

Оператор Create Arc

Назначение

Создает объект типа “дуга”. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Arc
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 )
start_angle end_angle
[ Pen... ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

$x1, y1$ – координаты одного угла минимального прямоугольного покрытия (МПП), прямоугольника, описывающего дугу;

$x2, y2$ – координаты противоположного по диагонали угла МПП дуги;

start_angle – значение начального угла дуги, в градусах;

end_angle – значение конечного угла дуги, в градусах;

Слово **Pen** начинает стандартное предложение для назначения стиля линии.

Описание

Результатом действия оператора **Create Arc** является новый объект типа "дуга".

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно, объект помещается на заданное место в окне (например, на изменяемый слой). Если **Into** вообще нет в операторе, MapBasic попытается создать объект в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры X и Y задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется **оператор Set CoordSys**. При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: X -координата – это расстояние от левого края листа до точки, и Y -координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**. Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Необязательный параметр **Pen** начинает предложение, задающее стиль линии. Если не задан параметр **Pen**, в операторе **Create Arc** будет использован стандартный стиль линии MapInfo Pro (заданный в диалоге команды **Стиль линий**).

См. также:

оператор Insert, **предложение Pen**, **оператор Update**, **оператор Set CoordSys**

Оператор Create ButtonPad

Назначение

В 64-разрядной версии MapInfo Pro эта команда создаёт новые вкладки или группы на существующих вкладках ленты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create ButtonPad { title_string | ID pad_num } As
  button_definition [ button_definition ... ]
  [ Title title_string ]
  [ Tab tab_name | tab_caption ]
  [ Width width ]
  [ Position ( x, y ) [ Units unit_name ] ]
  [ ToolbarPosition ( row, column ) ]
  [ { Show | Hide } ]
  [ { Fixed | Float | Top | Left | Right | Bottom } ]

```

title_string — заголовок инструментальной панели; Любой пробел в заголовке будет пропущен, а специальные символы заменены.

pad_num — идентификатор стандартной инструментальной панели:

- 1 — для панели Операции
- 2 — для панели Пенал
- 3 — для панели Программы
- 4 — для панели Команды
- 5 — для панели СУБД
- 6 — для панели Web-службы
- 7 — зарезервировано

tab_name, *tab_caption*. Когда создаётся новая вкладка, можно задавать и имя вкладки, и заголовок, подпись, разделяя их с помощью символа вертикальной черты (|). Например, "MyNewTab | MyNewCaption"

Для того чтобы получить имена (NAMES) существующих элементов управления ленты, создайте и запустите модуль ProSampleAddin, который включён в MapBasic, и используйте метод Dump Object Model.

(MapBasic\Samples\RIBBONINTERFACE\DotNet\ProSampleAddIn)

width — ширина панели, измеряется количеством кнопок. Не поддерживается в 64-разрядной версии MapInfo Pro.

x, *y* — позволяет задавать в "бумажных" единицах положение панели, когда она является плавающей. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#). Не поддерживается в 64-разрядной версии MapInfo Pro.

unit_name — имя, название "бумажной" единицы (например, "in" — дюйм, "cm" — сантиметр). Не поддерживается в 64-разрядной версии MapInfo Pro.

row, *column* — координаты инструментальной панели, когда она находится в закреплённом состоянии, например, координаты 0, 0 означают расположение строки инструментов, прижатой к левому и верхнему краям рабочего окна, а 0, 1 задают положение панели второй строкой). Не поддерживается в 64-разрядной версии MapInfo Pro.

- Значение *row* отсчитывается сверху вниз. Это значение высчитывается относительно других строк, находящемся выше или ниже. Если в той же позиции расположена строка меню, то отсчёт ведётся относительно строки меню. Когда панель инструментов находится прямо под строкой меню, номером её строки является значение равное 0. Если панель находится выше строки меню, то её значение равняется -1.
- Значение *column* отсчитывается слева направо. Это значение высчитывается относительно других инструментальных строк, находящемся правее или левее. Например, если панель инструментов закреплена слева и строка меню закреплена в левом положении, то номером столбца левее меню будет -1. Номером колонки, которая располагается правее меню, будет 0.

Каждый параметр *button_definition* является либо ключевым словом **Separator**, либо группой предложений следующего синтаксиса:

```
{ PushButton | ToggleButton | ToolButton }
  Calling { procedure | menu_code | OLE methodname | DDE server, topic }

  [ ID button_id ]
  [ Small | ExtraSmall | Large Icon icon_code [ File file_spec ] ]
  [ Cursor cursor_code [ File file_spec ] ]
  [ DrawMode dm_code ]
  [ HelpMsg msg ]
  [ ModifierKeys { On | Off } ]
  [ { Enable | Disable } ]
  [ { Check | Uncheck } ]
```

procedure — имя процедуры-обработчика, вызываемой при нажатии на кнопку;

menu_code — стандартный код меню MapInfo Pro из файла MENU.DEF (например, M_FILE_OPEN); MapInfo Pro выполняет соответствующую команду меню, когда пользователь нажимает на кнопку.

methodname — строковая величина, задающая имя метода OLE.

server, topic — строковая величина, задающая DDE-сервер и имя темы (topic);

ID *button_id* задаёт кнопке уникальный номер. Этот номер можно будет потом использовать как ее идентификатор, в случае, если несколько кнопок вызывают один обработчик. Его также использует [оператор Alter Button](#).

Icon *icon_code* — задает пиктограмму, которая будет на кнопке. Здесь *icon_code* может быть одним из специальных кодов файла ICONS.DEF, например, MI_ICON_RULER(11). Предложение File задает файл ресурсов изображений; в этом случае параметр *icon_code* должен быть целочисленным идентификатором одного из ресурсов заданного файла. Размер кнопки можно задать с помощью идентификатора файла ресурсов *icon_code* для кнопок малого размера и *icon_code*+1 для кнопок большого размера; то есть одна кнопка представлена в двух вариантах с id-номерами *icon_code* и *icon_code*+1.

Примечание: Для параметра *Icon Code* теперь можно использовать Uri-синтаксис в токене File.

Small | **ExtraSmall** | **Large** — позволяют задавать размер иконок-пиктограмм.

Cursor *cursor_code* — задает форму указателя мыши, если пользователь выберет кнопку инструмента (тип `ToolButton`); коды *cursor_code*, такие как `MI_CURSOR_ARROW` (0), задаются в файле `ICONS.DEF`. Это предложение используется только для кнопок инструментов. Предложение **File** задает имя файла ресурсов изображений; в этом случае параметр *cursor_code* должен быть целочисленным идентификатором одного из ресурсов заданного файла.

В 64-разрядной версии MapInfo Pro вид курсора поддерживается тремя способами:

1. В виде строки, описанной в файле `Icons.def`: `SetRbnToolBtnCtrlCursor(MyToolButton, "138")`
2. В виде строки с ключевым словом `FILE` и именем `DLL`, где хранятся курсоры:
`SetRbnToolBtnCtrlCursor(MyToolButton, "136 FILE gcsres32.dll")`
3. Как целое число с идентификатором курсора MapInfo в `Icons.def`:
`SetRbnToolBtnCtrlCursorId(MyToolButton, MI_CURSOR_FINGER_LEFT)`

DrawMode *dm_code* задаёт возможность инструмента рисовать (использование возможности передвигать мышку с нажатой клавишей) или только указывать (использование только возможности нажимать на клавишу мышки), при этом параметр *dm_code* должен быть одним из специальных кодов из файла `ICONS.DEF` (Предложение **DrawMode** применяется только к инструментальным панелям).

HelpMsg *msg* задает текст подсказки, которая появляется в строке сообщений при указании на кнопку, а также может задавать текст для плавающей подсказки `ToolTip`. Первая часть строки *msg* показывается в строке сообщений. Если величина *msg* включает в себя литеры `\n`, то текст, следующий за `\n`, используется подсказкой `ToolTip`.

ModifierKeys управляет использованием клавиш **SHIFT** и **CTRL** в режиме рисования, сопровождающемся прорисовкой образа объекта ("rubberband"), инструментом кнопки типа `ToolButton`. По умолчанию используется режим **OFF**, не использующий клавиши **SHIFT** и **CTRL**.

Описание

Оператор **Create ButtonPad** используется для создания новой инструментальной панели. Создав панель инструментов, можно ее изменять, используя **оператор Alter Button** и **оператор Alter ButtonPad**.

Каждая инструментальная панель может быть скрыта. Чтобы инструментальная панель была изначально скрыта, задайте слово **Hide**.

Ключевые слова **Fixed** и **Float** задают состояние панели, в котором она показывается на экране: закрепленной в верхнем краю окна или плавающей в виде вспомогательного окна. Пользователь может управлять положением инструментальной панели и помещать её мышкой на экран, либо прикреплять к краю экрана. Дополнительные возможности управления положением панели дают ключевые слова **Top** (то же самое, что и **Fixed**), **Left**, **Right** и **Bottom**.

Если панель "плавающая", ее изначальное положение задается предложением **Position**; если же она закреплена, то положение задается предложением **ToolbarPosition**.

Более подробно инструментальные панели описаны *Руководстве пользователя MapBasic*.
См. также раздел: **Оператор Alter ButtonPad**.

О размерах пиктограмм

В версиях MapInfo Pro до 10.0 пользователю разрешалось создавать пиктограммы (иконки) двух размеров: 18X16 — для малых пиктограмм и 26x24 — для больших. В то же время Инструментальных панелях и меню MapInfo Pro 10.0 использовались только квадратные иконки: 16X16 — для малых и 24x24 — для больших. Если иконки были не квадратными, то в MapInfo Pro версий до 10.0 включительно иконки показывались искаженными. Искажение состояло в том, что программа MapInfo Pro обрезала боковые участки иконок, чтобы добиться квадратного вида. Такие иконки могли выглядеть на экране непонятно.

Лучше всего создавать пиктограммы – иконки размерами 16x16 и 24x24. Но в версиях до 10.0.1 такие иконки будут вызывать ошибку.

Для использования в версиях до 10.0.1, придется создавать иконки размером 18x16 и 26x24. Такие иконки будут искажаться в версии 10.0 и обрезаться по бокам в версиях, старше 10.0.

О пиктограммах для пунктов меню

Ниже описано, как MapInfo Pro 10.0 использует пиктограммы (иконки) в пунктах меню:

- Пиктограммы для пунктов меню динамически формируются из пиктограмм инструментальных панелей.
- Пиктограмма для кнопки инструментальной панели может быть использована для пунктов меню по следующим правилам:
 - Кнопка инструментальной панели и пункт меню должны быть объявлены в одном и том же файле MBX.
 - То же правило должно соблюдаться в отношении пары обработчик/номер userID; каждый пункт меню, вызывающий тот же обработчик/номер userID из MBX-программы, ассоциируется с этой иконкой.
 - Тот же номер userID, вызывающий другие обработчики, ассоциируется с этой иконкой.
 - Тот же обработчик без номера userID ассоциируется с этой иконкой.
 - Номер идентификатора ID должен быть меньшим или равным 32767.
- Для встроенных обработчиков приоритет отдается иконкам со встроенных Инструментальных панелей. Если таковые не найдены, то поиск проводится среди иконок, объявленных в MBX-файлах.
- Иконки показываются рядом с командами меню.
- Когда MBX-программа выгружается из памяти, все ассоциированные с ней иконки также выгружаются и их уже нельзя использовать в меню.

Ниже описано, как MapInfo Pro использует пиктограммы (иконки) в пунктах меню:

- Все остальные пункты меню снабжаются иконками, объявленным в MapBasic-программе, по правилам, описанным выше.

Режимы предложения Calling

Предложение **Calling** задает, что должно случиться, если пользователь нажмет на кнопку инструментальной панели. В таблице приведены возможные примеры использования этого предложения:

Примеры	Описание
<code>Calling M_FILE_NEW</code>	Если за словом Calling идет целочисленный код из файла MENU.DEF, запускается соответствующая стандартная команда MapInfo Pro (например, Файл > Новая таблица).
<code>Calling my_procedure</code>	Если задано имя sub-процедуры, MapInfo передает управление этой процедуре. Эта процедура должна быть частью программы MapBasic, создавшей эту инструментальную панель.
<code>Calling OLE "methodname"</code>	Только для Windows. MapInfo Pro управляет событиями, обращаясь с именем метода к объекту OLE Automation, установленному SetCallback-методом MapInfo. Детали см. в <i>Руководстве пользователя MapBasic</i> .
<code>Calling DDE "server", "topic"</code>	MapInfo Pro управляет событиями, присоединяясь через DDE к паре "сервер тема" и посылая выполняемое сообщение на DDE-сервер.

В последних двух случаях строка послания для OLE или DDE сервера должна начинаться с трех символов "MI:" для того, чтобы сервер смог определить, что послание пришло от MapInfo. Остальная часть этой строки состоит из разделенного запятыми списка значений возврата функций с CommandInfo(1) по CommandInfo(8). Полное описание синтаксиса этой строки см. в *Руководстве пользователя MapBasic*.

Поведение традиционной вкладки, если не указано имя пользовательской вкладки или группы

Каждая панель инструментов создаст в ленте новую группу на вкладке **ИНСТРУМЕНТЫ** со всеми определениями кнопок в этой вновь созданной группе ленты с размером, заданным большим.

Каждая панель инструментов с предопределённым идентификатором ID будет переназначена в группу ленты на вкладке **ИНСТРУМЕНТЫ**.

1. для Операции [символ] группы Операции
2. для Пенал [символ] группы Пенал
3. для Программы [символ] группы Программы
4. для Стандартные [символ] группы Стандартные

5. для СУБД [символ] группы СУБД
6. для Веб-служб [символ] группы Веб-службы
7. Зарезервировано [символ] группы Зарезервировано

Другие изменения традиционного поведения

1. *title_string* [Symbol] имя инструментальной панели или идентификатор заранее определённой панели инструментов.

Примечание: Имена с пробелами и без пробелов будут рассматриваться как одинаковые, например, “pad 1” и “pad1” будет считаться одинаковыми.

2. Имя инструментальной панели не должно содержать никаких специальных символов, поскольку такие символы будут скрыты, как не поддерживаемые элементами управления WPF.
3. Предложение с заголовком [символ] названия инструментальной панели. Если такое предложение не задано, то название группы будет создано автоматически.
4. Width [Symbol] — **не поддерживается**
5. Position [Symbol] — **не поддерживается**
6. Units [Symbol] — **не поддерживается**
7. ToolBarPosition [Symbol] — **не поддерживается**
8. Show/Hide [Symbol] переключатель видимости группы ленты — **поддерживается**.
9. Fixed | Float | Top | Left | Right | Bottom [Symbol] — **не поддерживается**

Примеры

Создание инструментальной панели для дополнительных программ:

```
Create ButtonPad "Utils" As
  PushButton
    HelpMsg "Choose this button to display query dialog"
    Calling button_sub_proc
    Icon MI_ICON_ZOOM_QUESTION
  ToolButton
    HelpMsg "Use this tool to draw a new route"
    Calling tool_sub_proc
    Icon MI_ICON_CROSSHAIR
    DrawMode DM_CUSTOM_LINE
  ToggleButton
    HelpMsg "Turn proximity checking on/off"
    Calling toggle_prox_check
    Icon MI_ICON_RULER
    Check
  Title "Utilities"
  Width 3
  Show
```

Создает кнопку, которая запускает диалог Свойства окна Списка, которому соответствует ID-номер меню 222:

```
Create ButtonPad "Prefs" As
  PushButton
  HelpMsg "Browser Preferences.\nBrowser Prefs"
  Calling 222
  Icon 99
```

См. также:

[оператор Alter Button](#), [оператор Alter ButtonPad](#), [функция ButtonPadInfo\(\)](#)

Оператор Create ButtonPad As Default

Назначение

Восстанавливает исходный вид стандартной инструментальной панели. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create ButtonPad { title_string | ID pad_num } As Default [ Tab tab_name
]
```

title_string — название инструментальной панели (например, "Операции" или "Программы").

pad_num — идентификатор стандартной инструментальной панели:

Tab_name — имя вкладки, которое заменяет панель инструментов.

Примечание: Если имя вкладки не присваивается, то эта команда удаляет группу ленты с именем, совпадающим с *title_string* или с идентификатором ID панели инструментов на вкладке **ИНСТРУМЕНТЫ** в ленте.

- 1 – для панели Операции
- 2 – для панели Пенал
- 3 – для панели Программы
- 4 – для панели Команды
- 5 – для панели СУБД
- 6 – для панели Web-службы
- 7 – зарезервировано

Для инструментальных панелей, созданных пользователем, используют только *title_string*.

Описание

Восстанавливает исходный вид инструментальных панелей MapInfo Pro (таких как: "Операции" или "Программы"). Инструментальные панели, созданные пользователем, удаляются.

См. также:

[оператор Alter Button](#), [оператор Alter ButtonPad](#), [оператор Create ButtonPad](#), [оператор Create ButtonPads As Default](#)

Оператор Create ButtonPads As Default

Назначение

Восстанавливает стандартный вид и состав панелей инструментов. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create ButtonPads As Default
```

Описание

Восстанавливает исходный вид инструментальных панелей MapInfo Pro (таких как: "Операции" или "Программы"). Инструментальные панели, созданные пользователем, удаляются.

Используйте этот оператор с осторожностью. Оператор **Create ButtonPads As Default** убирает все панели инструментов, построенные приложением, и восстанавливает три стандартные панели в их прежнем виде: Операции, Пенал и Программы.

Примечание: Эта команда удаляет все кнопки на вкладке ИНСТРУМЕНТЫ в ленте и удаляет вкладку ИНСТРУМЕНТЫ только из ленты.

См. также:

[оператор Alter Button](#), [оператор Alter ButtonPad](#), [оператор Create ButtonPad](#), [оператор Create ButtonPad As Default](#)

Оператор Create Cartographic Legend

Назначение

Позволяет создать и отобразить стиль картографических легенд, также как и тематических легенд для активного окна карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Cartographic Legend
[ From Window map_window_id ]
[ Behind ]
[ Position ( x, y ) [ Units paper_units ] ]
[ Width win_width [ Units paper_units ] ]
[ Height win_height [ Units paper_units ] ]
[ Window Title { legend_window_title }
[ ScrollBars { On | Off } ]
[ Portrait | Landscape | Custom ]
[ Style Size { Small | Large }
[ Default Frame Title { def_frame_title } [ Font... ] } ]
[ Default Frame Subtitle { def_frame_subtitle } [ Font... ] } ]
[ Default Frame Style { def_frame_style } [ Font... ] } ]
[ Default Frame Border Pen [ [ pen_expr ]
Frame From Layer { map_layer_id | map_layer_name
[ Using
[ Column { column | Object } [ FromMapCatalog { On | Off }]]
[ Label { expression | Default } ]
[ Position ( x, y ) [ Units paper_units ] ]
[ Title { frame_title [ Font... ] }
[ SubTitle { frame_subtitle [ Font... ] } ]
[ Border Pen pen_expr ]
[ Style [ Font... ] [ Norefresh ] [ Text { style_name }
{ Line Pen... | Region Pen... Brush... | Symbol Symbol... } |
Collection [ Symbol ... ]
[ Line Pen... ] [ Region Pen... Brush ... ] } ]
[ , ... ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются функция **FrontWindow()** и функция **WindowID()**.

x – определяет требуемое расстояние от верхнего края рабочего стола MapInfo до верхнего края окна.

y – определяет требуемое расстояние от левого края рабочего стола MapInfo до левого края окна.

Примечание: Здесь под рабочим столом подразумевается область с данными на экране (без строк заголовка и состояния программы и панелей инструментов).

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цистеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

win_width – ширина окна.

win_height – высота окна.

legend_window_title – это строковое выражение, соответствующее заголовку окна легенды, по умолчанию это "Legend of xxx" где xxx это заголовок окна карты (название может быть другим в русской версии);

def_frame_title – это строковая величина, которая по умолчанию определяет заголовок раздела легенды. Эта величина может включать специальный символ "##", который будет замещаться именем текущего слоя.

def_frame_subtitle – это строковая величина, которая по умолчанию определяет заголовок подраздела легенды. Эта величина может включать специальный символ "##", который будет замещаться именем текущего слоя.

def_frame_style – это строковая величина, которая определяет каждый символ в каждом разделе легенды. Символ "##" будет замещаться именем текущего слоя. Символ %% будет замещаться словами "Линия", "Точка", "Полигон", которые будут соответствовать типу географических объектов (эти названия могут быть другими в русской версии). Например, "% of #" развернёт название до "Region of States" слоя STATES.TAB.

pen_expr – это выражение, возвращающее объект типа Pen, например, *MakePen(width, pattern, color)*. Если по умолчанию линия рамки определена, то она останется по умолчанию такой же для раздела легенды. Если предложение Pen для рамки задано для раздела легенды, то оно будет определять тип линии для рамки вместо заданного по умолчанию.

map_layer_id или *map_layer_name* определяют слой карты; это может быть целая величина Smallint (например, используйте 1 для определения самого верхнего слоя, не считая косметического) или строковая величина, соответствующая имени таблицы, отображенной на карте. Для тематического слоя необходимо определять параметр *map_layer_id*.

frame_title – строковая переменная, определяющая заголовок раздела легенды. Если будет определено предложение заголовка раздела **Title**, то будет использоваться эта величина вместо величины *def_frame_title*.

frame_subtitle – строковая переменная, определяющая подзаголовок раздела легенды. Если предложение **Subtitle** определено, то оно будет использоваться вместо имени, задаваемого по умолчанию величиной *def_frame_subtitle*.

column – имя атрибутивной колонки из раздела слоя таблицы.

style_name – строковая переменная, которая указывает к какому типу: символу, линии или области относится объект в разделе легенды.

Описание

Оператор **Create Cartographic Legend** позволяет создать и отобразить стиль картографических легенд, также как и тематических легенд для активного окна карты. Каждый стиль картографической и тематической легенды будет связан только с одним окном карты, так что одновременно может быть открыто несколько окон легенды.

Можно создать раздел легенды для каждого картографического или тематического слоя, который вы захотите включить в легенду. Картографические и тематические разделы будут включать заголовки и подзаголовки легенды. Картографические разделы легенды показывают стили слоев карты; разделы легенды отражают цвета, символы и их размер для тематических слоев. Можно создать разделы, которые имеют стили, основанные на стиле окна карты или создать ваши собственные разделы легенды.

Как минимум, требуется задать одно предложение **Frame**.

Все предложения, относящиеся ко всей легенде (Scrollbars, Width, и др.) должны следовать за первым предложением **Frame**.

Предложение **From Layer** должно быть первым предложением после **Frame**.

Если задано предложение **Behind**, легенда размещается под окном тематической карты.

Предложение **Position** (необязательное) контролирует положение окна легенды в рабочем окне. Верхний левый угол рабочего окна MapInfo Pro имеет координаты 0, 0. Дополнительные предложения **Width** и **Height** позволяют задать другой размер окна. В параметре "Position" используются "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры). "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя [оператор Set Paper Units](#). Оператор **Create Cartographic Legend** может переопределить единицы измерения; для этого надо включить предложение **Units** в предложения **Position**, **Width** и/или **Height**.

Предложение **ScrollBars** управляет показом строки (полосы) прокрутки в окне карты.

Portrait и **Landscape** описывают ориентацию разделов легенды в окне. **Portrait** – это книжная ориентация. **Landscape** – это альбомная ориентация.

Если указано **Custom**, то вы можете задать ваше собственное предложение **Position** для позиционирования раздела.

Предложение **Position** для рамки задает положение рамки в случае использования настраиваемых параметров **Custom**. X-координата отсчитывается от левой стороны легенды, а Y-координата от нижней стороны окна легенды (начало координат (0,0) находится в левом нижнем углу окна легенды).

Предложение **Style Size** управляет размером образцов в окне легенды. Если вы задали режим **Style Size Small**, в окне легенды будут показаны малые образцы. Если вы задали режим **Style Size Large**, в окне легенды будут показаны большие образцы.

Предложения **Position**, **Title**, **SubTitle**, **Border Pen** и **Style** для разделов легенды используются только для слоев карты. Они не используются для тематических слоев. Для тематических слоев вся необходимая информация задаётся автоматически при их создании.

Параметром этого оператора **Font** можно задать стиль оформления текста. Если по умолчанию заголовок раздела, подзаголовок или имя стиля объекта имеют определенный шрифт, то этот же шрифт по умолчанию будет использоваться и для раздела легенды. Если на уровне раздела определены предложения **Title**, **SubTitle** или **Style**, и в этих предложениях используется предложение **Font**, то будет использоваться шрифт, заданный этим предложением. Если ни на каком уровне шрифт не определен, то будет применяться текущий шрифт с размером букв 10, 9 и 8 для заголовка, подзаголовка и имени стиля соответственно.

Предложение **Style** и ключевое слово **NoRefresh** позволяют вам создавать собственные разделы легенды, которые не будут изменяться при обновлении легенды. Если ключевое слово **NoRefresh** используется в предложении **Style**, то стили не извлекаются из таблицы. В этом случае предложение **Style** должно содержать ваш собственный список стилей, используемых в разделе легенды. Это достигается заданием предложения **Text** и соответствующих предложений **Line**, **Region** или **Symbol**. Объекты типа группа точек обрабатываются как точечные объекты.

Объекты типа коллекция обрабатываются отдельно. Когда MapInfo Pro создает легенду на основе типов объектов, то сначала рисуются точечные символы, затем линии, а затем области. Объекты типа коллекция изображаются последними. В пределах объекта типа коллекция порядок рисования таков: точки, линии, объекты.

Если задано предложение *Column*, то значение **column** должно представлять имя атрибутивной колонки в таблице, соответствующей слою, представленному в разделе, либо должно стоять слово **Object**, обозначающее объектную колонку (это значит, что стили Легенды основываются на уникальных стилях MAP-файла). По умолчанию используется **Object**.

FromMapCatalog ON отыскивает стили из MapCatalog для таблицы прямого доступа. Если таблица не является таблицей прямого доступа, MapBasic возвращается к заданному по умолчанию поведению для таблицы с непрямым доступом вместо того, чтобы допустить ошибку. Стандартное поведение для таблицы с отключённым доступом это **FromMapCatalog Off** (то есть, используются стили карты).

FromMapCatalog OFF отыскивает уникальные стили Карты для таблицы прямого доступа сервера. Эта таблица должна быть таблицей прямого доступа, которая поддерживает стилизацию каждой записи. Если таблица прямого доступа не поддерживает стили записей, то применяется заданное по умолчанию поведение для таблиц прямого доступа – используются заданные по умолчанию стили из MapCatalog (**FromMapCatalog ON**).

Предложение *Label* – это либо выражение **expression** для подписи, либо слово **Default** (последнее означает, что при создании текста используется стиль оформления рамки по умолчанию, если только предложение стилизации не описывает текст). Стандартным значением является **Default**.

Сначала, для каждого слоя, порождающего раздел легенды, исследуется соответствующий ТАВ-файл на предмет наличия значений метаданных для заголовка легенды, подзаголовка, структуры колонок и т.п. Если для колонки не заданы метаданные, то по умолчанию используется **Object**. Если в метаданных нет данных для предложения Label, то по умолчанию используются текущие установки стиля раздела. Если же в метаданных есть значения, но их нужно заменить своими, то это можно сделать с помощью операторов MapBasic.

Пример

В следующем примере показано, как создать раздел в картографической легенде для окна карты. У окон легенды есть одна особенность: чтобы создать раздел в окне легенды, нужно использовать предложение **Title** вместо **From Window**.

```
Dim i_layout_id, i_map_id As Integer
Dim s_title As String
' here, you would store the Map window's ID in i_map_id,
' and store the Layout window's ID in i_layout_id.
' To obtain an ID, call FrontWindow() or WindowID().
s_title = "Legend of " + WindowInfo(i_map_id, WIN_INFO_NAME)
Set CoordSys Layout Units "in"
Create Frame
  Into Window i_layout_id
  (1,2) (4, 5)
  Title s_title
```

Так создается раздел в окне картографической легенды. Чтобы создать раздел для тематической легенды, замените заголовок следующим образом:

```
S_title="Theme Legend of " + WindowInfo (I_map_id, WW_INFO_NAME)
```

См. также:

[оператор Set Cartographic Legend](#), [оператор Alter Cartographic Frame](#), [оператор Add Cartographic Frame](#), [оператор Remove Cartographic Frame](#), [оператор Create Legend](#), [оператор Set Window](#), [функция WindowInfo\(\)](#)

Функция CreateCircle()

Назначение

Возвращает объект типа "окружность". Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CreateCircle( x, y, radius )
```

x – X-координата центра окружности (или широта), действительное число;

y – Y-координата центра окружности (или долгота), действительное число;

radius – действительное число, назначающее радиус окружности.

Возвращаемая величина

Объект

Описание

Функция **CreateCircle()** возвращает графический объект типа "окружность".

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется **оператор Set CoordSys**.

Примечание: При этом следует учитывать, что MapBasic игнорирует координатную систему самого окна карты.

Параметр *radius* назначается в тех единицах измерения, которые были назначены MapBasic до выполнения этой функции. По умолчанию MapBasic использует мили, а **оператор Set Distance Units** может задать другие единицы измерения в среде MapBasic.

Окружность заполняется штриховкой текущего стиля. Чтобы создать окружность с заданной штриховкой, выполните **оператор Set Style** перед **CreateCircle()**. Вместо вызова **CreateCircle()** можно выполнить **оператор Create Ellipse**, в котором, если надо, задаются **предложение Pen** и **предложение Brush**.

Графический объект, созданный функцией **CreateCircle()**, может быть присвоен объектной переменной, которая задает значение для уже существующей строки таблицы **оператор Update**) или вновь созданной (используя **оператор Insert**).

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Ошибки

ERR_FCN_ARG_RANGE (644) произведен, если аргумент за пределами действительного диапазона.

Примеры

В примере используется **оператор Insert** для создания новой строки в таблице SITES. Функция **CreateCircle()** вложена в оператор Insert, и при его выполнении создается объект "Окружность", данные которого содержатся в новой строке (записи).

```
Open Table "sites"
Insert Into sites (obj)
  Values ( CreateCircle(-72.5, 42.4, 20) )
```

В следующем примере используется таблица TOWERS, которая имеет три колонки: "Xcoord", "Ycoord" и "Radius". Колонки "Xcoord" и "Ycoord" содержат значения долготы и широты, где находятся радиостанции, а колонка "Radius" – значения радиусов областей их вещания. Каждая запись в таблице описывает радиотрансляционную башню, а колонка "Radius" задает область распространения сигнала каждой башни.

Сначала встроенная в **оператор Update** функция **CreateCircle()** создает окружности для каждой строки таблицы. Затем оператор Update присваивает окружность каждой записи таблицы Towers. Каждая окружность радиус из колонки "Radius", а центр будет задан координатами из колонок "Xcoord" и "Ycoord".

```
Open Table "towers"
Update towers
  Set obj = CreateCircle(xcoord, ycoord, radius)
```

См. также:

оператор Create Ellipse, оператор Insert, оператор Update

Оператор Create Collection

Назначение

Объединяет точечные, линейные и площадные объекты в один объект. В окне списка объект "Коллекция" представлен одной записью. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Collection [ num_parts ]
[ Into { Window window_id | Variable var_name } ]
Multipoint
[ num_points ]
( x1, y1 ) ( x2, y2 ) [ ... ]
[ Symbol... ]
```

```

Region
  num_polygons
  [ num_points1 ( x1, y1 ) ( x2, y2 ) [ ... ] ]
  [ num_points2 ( x1, y1 ) ( x2, y2 ) [ ... ] ... ]
  [ Pen... ]
  [ Brush... ]
  [ Center ( center_x, center_y ) ]
Pline
  [ Multiple num_sections ]
  num_points
  ( x1, y1 ) ( x2, y2 ) [ ... ]
  [ Pen... ]
  [ Smooth... ]

```

num_parts – число непустых частей в коллекции. Это число от 0 до 3 является дополнительным кодом MapBasic (но для MIF задание этого кода обязательно).

num_polygons – число полигонов объекта Коллекция.

num_sections – указывает, сколько сегментов содержит многосегментная полилиния.

Pen – корректное [предложение Pen](#), задающее стиль линии.

Brush – корректное [предложение Brush](#), задающее штриховку.

Пример

```

create collection multipoint 2 (0,0) (1,1) region 3 3 (1,1) (2,2) (3,4)
4
(11,11) (12,12) (13,14) (19,20) 3 (21,21) (22,22) (23,24) pline 3 (-1,1)
(3,-2) (4,3)
dim a as object
create collection into variable a multipoint 2 (0,0) (1,1) region 1 3
(1,1) (2,2) (3,4) pline 3 (-1,1) (3,-2) (4,3)
insert into test (obj) values (a)
create collection region 2 4 (-5,-5) (5,-5) (5,5) (-5,5) 4 (-3,-3) (3,-3)
(3,3) (-3,3) pline multiple 2 2 (-6,-6) (6,6) 2 (-6,6) (6,-6) multipoint
6
(2,2) (-2,-2) (2,-2) (-2,2) (4,1) (-1,-4)

```

См. также:

[оператор Create MultiPoint](#), [оператор Set Combine Version](#)

Оператор Create Cutter

Назначение

Назначает объект, который будет использоваться как секущий объект в операции разрезания, а также изменяемый объект или группу изменяемых объектов. Необходимо назначить изменяемый объект, а также задать секущий объект в виде набора полилиний до того, как запустите операцию разрезания. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Cutter Into Target
```

Описание

Перед использованием оператора Create Cutter должны быть выбраны один или более объектов полилиний и должен существовать изменяемый объект. Это задаётся либо командой **Выбрать изменяемый объект**, либо с помощью оператора **Set Target**. Объекты-полилинии, содержащиеся в выборке, должны быть представлены одним непрерывным объектом, без разрывов и самопересечений.

Полилиния должна пересекать МОП (минимальный описывающий прямоугольник) изменяемого объекта, иначе разрезания не произойдет. Полилиния, однако, сама не должна пересекать изменяемый объект. Например, если изменяемый объект представляет группу Гавайских островов, то полилиния может использоваться для разделения островной гряды на 2 части, при этом не касаясь ни одного отдельного острова. Если МОП изменяемого объекта не пересекает полилинию, то тогда такой объект будет удален из списка изменяемых объектов.

После выделения изменяемых объектов, рассчитывается минимальный описывающий прямоугольник всех этих объектов; сам МОП представляет пространство, которое будет разделено оператором. Затем полилиния продолжается, если необходимо, так что она попадает в область МОП. Это достигается тем, что: полилиния продолжается до пересечения с МОП в направлении отрезков между двумя последними точками на каждом конце этой полилинии. Растянутая полилиния разделяет пространство изменяемых объектов на две части. В результате будет создан и возвращен объект Область, представляющий одну из этих двух частей.

Этот оператор будет возвращать обработанное множество изменяемых объектов и новый секущий объект типа "Область". Объект-область будет вставлен в таблицу с изменяемыми объектами (которая должна быть изменяемой). Исходный объект(ы) полилиния останется, но перестанет быть выбранным. Выбранным станет новый объект типа "Область". Если

получившийся объект-область удовлетворяет нашим нуждам, то можно сразу же проводить операцию разрезания, так как изменяемые объекты и секущий объект уже выбраны.

Примечание: Секущий объект останется на одном слое с изменяемыми объектами. Можно удалить его вручную.

Пример

```
Open Table "C:\MapInfo_data\TUT_USA\USA\STATES.TAB"
Open Table "C:\MapInfo_data\TUT_USA\USA\US_HIWAY.TAB"
Map from States, Us_hiway
select * from States where state = "NY"
Set target On
select * from Us_hiway where highway = "I 90"
Create Cutter Into Target
Objects Split Into Target
```

См. также:

[функция OverlayNodes\(\)](#), [оператор Set Target](#)

Оператор Create Designer Legend

Назначение

Создаёт новое окно **Легенда**, в котором отображаются картографические легенды векторных, растровых и тематических слоёв активного окна **Карта**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Designer Legend
[ From Window map_window_id ]
[ WINDOW_CLAUSE ]
[ DEFAULT_FRAME_CLAUSE ]
[ LEGEND_TEXT_FRAME_CLAUSE ]
[ FRAME_FROM_LAYER_CLAUSE ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#). Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

WINDOW_CLAUSE смотрите [Настройки окон](#)

DEFAULT_FRAME_CLAUSE смотрите [Стандартные параметры раздела](#)

LEGEND_TEXT_FRAME_CLAUSE смотрите [Настройки текстового раздела легенды](#)

FRAME_FROM_LAYER_CLAUSE смотрите [Параметры оператора для раздела слоя](#)

Описание

Оператор **Create Designer Legend** позволяет создавать легенды векторных, растровых и тематических слоёв активного окна **Карта**. Можно создавать более одного окна **Легенда** для каждого окна **Карта**, каждый слой карты может составлять один раздел окна **Легенда**.

Только для форматов файлов, которые поддерживаются в расширенном варианте растрового модуля, имеется возможность создания растровых легенд. Список форматов файлов, поддерживаемых в расширенном варианте растрового модуля, можно найти в MapInfo Pro на вкладке **PRO**, если нажать кнопку **Дополнительно**. В разделе **MapInfo Pro растр** нажмите кнопку **Настройки**. В диалоге **Настройки для растра** выполните команду **Растровые форматы**. Только форматы, которые не относятся к **Изображениям / Палитрам изображений** поддерживаются для создания легенд. За исключением сжатых с помощью вейвлет-метода ERDAS Enhanced Compressed Wavelets (.ecw), новых маркированных USGS DOQ (.doq) и навигационных карт BSB NOAA Nautical Chart (.kap) для всех остальных форматов поддерживается создание легенды.

О том как создавать легенды, в которых представлены индивидуальные стили, с помощью оператора Create Designer Legend, смотрите раздел: [Параметры оператора для раздела слоя](#).

При использовании этого оператора требуется по крайней мере одно предложение **Frame From Layer** (*FRAME_FROM_LAYER_CLAUSE*). Это предложение создаёт легенду в окне **Легенда**. Все предложения, относящиеся к общему окну **Легенда**, например, *WINDOW_CLAUSES*, должны начинаться с предложения **Frame From Layer**.

Примечание: Используя [оператор Shade](#) и оператор [Set Legend](#), по-прежнему можно создавать и изменять тематические легенды, но добавлять тематические разделы в окна **Легенда** можно только с помощью оператора **Create Designer Legend** (задавая идентификатор тематического слоя с помощью параметра **Frame From Layer id**).

См. также:

[оператор Alter Designer Frame](#), [оператор Add Designer Frame](#), [оператор Remove Designer Frame](#), [оператор Set Designer Legend](#), [оператор Set Window](#), [функция WindowInfo\(\)](#)

Настройки окон

Предложения, перечисленные в списке *WINDOW_CLAUSES*, являются дополнительными, необязательными предложениями, с помощью которых настраиваются свойства окна **Легенда**.

Синтаксис

Create Designer Legend

```
[ From Window map_window_id ]
[ WINDOW_CLAUSES ]
[ DEFAULT_FRAME_CLAUSES ]
[ LEGEND_TEXT_FRAME_CLAUSE ]
[ FRAME_FROM_LAYER_CLAUSE ]
```

где *WINDOW_CLAUSES*:

Create Designer Legend

```
[ From Window map_window_id ]
[ Position ( x, y ) [ Units paper_units ] ]
[ Width win_width [ Units paper_units ] ]
[ Height win_height [ Units paper_units ] ]
[ Min | Max | Restore | Floating | Docked | Tabbed | AutoHidden ]
[ Window Title { legend_window_title }
[ Portrait | Landscape | Custom ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#). Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

y – определяет требуемое расстояние от левого края рабочего стола MapInfo до левого края окна.

x – определяет требуемое расстояние от верхнего края рабочего стола MapInfo до верхнего края окна.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также циперо. Список "бумажных" единиц измерения приведен в разделе: [Оператора Set Paper Units](#).

win_width – ширина окна.

win_height – высота окна.

legend_window_title – это строковое выражение, соответствующее заголовку окна легенды, по умолчанию это "Легенда моё окно" где *моё окно* это заголовок окна **Карта**;

Описание

Предложение **Position** (необязательное) контролирует положение окна легенды в рабочем окне. Верхний левый угол рабочего окна MapInfo Pro имеет координаты 0, 0.

Дополнительные предложения **Width** и **Height** позволяют задать другой размер окна. В параметре "Position" используются "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя [оператор Set Paper Units](#). Оператор **Create Designer**

Legend может переназначать текущие единицы измерения, включением в предложение **Units** значений **Position**, **Width** и/или **Height**.

Предложение **Floating** – ответственно за плавающее состояние окна. (Это применимо только к 64-разрядной версии MapInfo Pro.)

Предложение **Docked** – ответственно за прикрепленное состояние окна, которое, обычно, используется по умолчанию. (Это применимо только к 64-разрядной версии MapInfo Pro.)

Предложение **Tabbed** – ответственно за состояние прикрепленных окон в виде отдельных вкладок. В этом состоянии окно называется документом. (Это применимо только к 64-разрядной версии MapInfo Pro.)

Предложение **AutoHidden** – такое состояние прикрепления, когда окно автоматически скрывается. (Это применимо только к 64-разрядной версии MapInfo Pro.)

Ключевые слова **Portrait** и **Landscape** описывают способ размещения разделов легенды. **Portrait** – это книжная ориентация, когда окно расположено сверху вниз. **Landscape** – это альбомная ориентация, когда окно расположено слева направо.

В режиме **Custom** Вы можете в предложении **Position** задавать положение каждого раздела.

Стандартные параметры раздела

Стандартные параметры раздела – *DEFAULT_FRAME_CLAUSES* (параметры **Default Frame**) являются дополнительными, необязательными и позволяют задавать стандартные параметры разделов легенд, которые используются по умолчанию. Если они не заданы, то применяются стандартные параметры программы (при этом действуют минимальные и максимальные ограничения). Стандартные параметры раздела используются, когда пропущены настройки отдельных разделов.

Синтаксис

```
Create Designer Legend
[ From Window map_window_id ]
[ WINDOW_CLAUSES ]
[ DEFAULT_FRAME_CLAUSES ]
[ LEGEND_TEXT_FRAME_CLAUSE ]
[ FRAME_FROM_LAYER_CLAUSE ]
[ , ... ]
```

где *DEFAULT_FRAME_CLAUSES*:

```
Create Designer Legend
[ From Window map_window_id ]
[ Default Frame Title { def_frame_title } [ Font... ] } ]
[ Default Frame Subtitle { def_frame_subtitle } [ Font... ] } ]
[ Default Frame Style { def_frame_style } [ Font... ] } ]
[ Default Frame Line Width width [ Units paper_units ] ]
```

```
[ Default Frame Region Width width [ Units paper_units ] ]
[ Default Frame Region Height height [ Units paper_units ] ]
[ Default Frame Auto Font Size { On | Off } ]
[ Default Frame Bar [ Thickness t_value ] [ Length l_value ]
[ Units paper_units ] [ Horizontal | Vertical ] ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются **функция FrontWindow()** и **функция WindowID()**. Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

def_frame_title – это строковая величина, которая по умолчанию определяет заголовок раздела легенды. Эта величина может включать специальный символ "##", который будет замещаться именем текущего слоя.

def_frame_subtitle – это строковая величина, которая по умолчанию определяет заголовок подраздела легенды. Эта величина может включать специальный символ "##", который будет замещаться именем текущего слоя.

def_frame_style – строковая величина, которая отображается рядом с каждым символом в каждом разделе легенды. Символ "##" будет замещаться именем текущего слоя. Символ %% будет замещаться словами "Линия", "Точка", "Полигон", которые будут соответствовать типу географических объектов (эти названия могут быть другими в русской версии). Например, "#" развернёт название до "Region of States" слоя STATES . TAB.

Примечание: Для того чтобы использовать символы # и % в тексте строки легенды, добавьте дополнительный управляющий символ обратной косой черты (\) перед каждым символом: \# или \%.

Width – значение ширины условного знака площадных объектов в "бумажных" единицах измерения.

height – значение высоты условного знака площадных объектов в "бумажных" единицах измерения.

t_value – толщина полоски цветов растрового раздела легенды. Единицами измерения являются дюймы, если не заданы другие единицы измерения с помощью оператора **Units**.

l_value – длина полоски цветов растрового раздела легенды. Единицами измерения являются дюймы, если не заданы другие единицы измерения с помощью оператора **Units**.

Описание

Предложения **Default Frame Title**, **Default Frame SubTitle** и **Default Frame Style** позволяют задавать стандартные заголовки, подзаголовки и текст строк каждого раздела, если в разделах не заданы индивидуальные параметры, которые переопределяют их.

Default Frame Line Width – задает толщину образцов условных знаков линий.

Default Frame Region Width – задает ширину образцов площадных условных знаков областей, участков.

Default Frame Region Height – задает высоту образцов площадных условных знаков областей, участков.

paper_units – строка, представляющая "бумажные" единицы измерения. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#).

Default Frame Auto Font Size – включает или выключает автоматическое изменение размеров условных знаков в легенде, используя размер шрифта в предложении **Style**.

Примечание: Предложения **Default Frame Title**, **Default Frame Subtitle** и **Default Frame Style** никогда не применяются к легендам тематических слоёв. Для легенд тематических слоев эти значения извлекаются автоматически из структур, описывающих тему.

Default Frame Bar описывает стандартные размеры и ориентацию полосы цветов растровой легенды и должен содержать по крайней мере один из следующих параметров: **Thickness**, **Length** либо ориентацию (**Horizontal** или **Vertical**). Если добавить только выражение **Units**, опустив другие параметры, то это приведёт к ошибке при выполнении (ошибка некорректного выражения). Если предложение **Default Frame Bar** пропущено, MapInfo Pro использует собственные стандартные значения в 0.25 дюйма для толщины – **Thickness**, 2.0 дюйма для длины – **Length** или ширины – **Vertical**, задаваемые при выборе определённой ориентации.

Настройки текстового раздела легенды

Предложение **LEGEND_TEXT_FRAME_CLAUSE** является дополнительным, необязательным предложением, с помощью которого создаётся текстовый раздел в окне **Легенда**. Оно позволяет задавать содержимое раздела, положение в легенде, стиль текста и единицы измерения, в которых отображаются элементы легенды. Можно использовать более одного предложений Legend Text Frame.

Синтаксис

```
Create Designer Legend
[ From Window map_window_id ]
[ WINDOW_CLAUSES ]
[ DEFAULT_FRAME_CLAUSES ]
[ LEGEND_TEXT_FRAME_CLAUSE ]
[ FRAME_FROM_LAYER_CLAUSE ]
```

Где **LEGEND_TEXT_FRAME_CLAUSE**:

```
Create Designer Legend
[ From Window map_window_id ]
[ Legend Text Frame [ Text { frame_text [ Font... ] }
  [ Position ( x, y ) [ Units paper_units ] ] ] ]
```

map_window_id – целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#). Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

frame_text – текст для заголовка, подзаголовка легенды, либо обычный текст (например, сообщение об авторских правах).

x, *y* – позволяют задавать положение текстового раздела легенды: *x* – от левого края окна **Легенда**, а *y* – от верхнего края окна **Легенда**. Начало координат (0,0) находится в левом верхнем углу окна **Легенда**).

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цигеро. Список "бумажных" единиц измерения приведен в разделе: **Оператора Set Paper Units**.

Описание

Предложение **Text** содержит текст текстового раздела. В нём могут быть любые элементы, например, заголовок легенды, подзаголовок или текст описания (например, описание карты или сведения об авторских правах). Если в предложении **Text** не задан параметр шрифта **Font**, то используется стандартный шрифт.

Предложение **Position** на уровне раздела позволяет задавать положение раздела, если определён параметр **Custom** в MapBasic-операторе, который определяет окно.

Параметры оператора для рамки слоя

Как минимум, требуется задать одно предложение *FRAME_FROM_LAYER_CLAUSE*. Оно задаёт стиль и оформление рамки в **Легенда**.

Синтаксис

```
Create Designer Legend
[ From Window map_window_id ]
[ WINDOW_CLAUSES ]
[ DEFAULT_FRAME_CLAUSES ]
[ LEGEND_TEXT_FRAME_CLAUSE ]
[ FRAME_FROM_LAYER_CLAUSE ]
```

Где *FRAME_FROM_LAYER_CLAUSE*:

```
Create Designer Legend
[ From Window map_window_id ]
Frame From Layer { map_layer_id | map_layer_name
[ Position ( x, y ) [ Units paper_units ] ]
[ Title { frame_title [ Font... ] }
[ SubTitle { frame_subtitle [ Font... ] } ]
[ Columns number_of_columns ] |
[ Height frame_height [ Units paper_units ] ]
[ Border Pen .... ] [ Brush ... ]
[ Priority priority_n ]
[ Name frame_name ]
```

```

[ Region [ Height region_height [ Units paper_units ] ] ]
[ Region [ Width region_width [ Units paper_units ] ] ]
[ Line [ Width line_width [ Units paper_units ] ] ]
[ Auto Font Size { On | Off } ]
[ Order { Default | Ascending | Descending |
  { Custom id | id : id [ , id | id : id ... ] ... } } ]
[ Using
  [ Column { column | Object } [ FromMapCatalog { On | Off } ] ]
  [ Label { expression | Default } ] ]
[ Display { Gap | Palette | Bar
  [ [ Thickness t_value ] [ Length l_value ]
    [ Units paper_units ] [ Horizontal | Vertical ] ] ] ]
[ Filter Max max_value ]
[ Sort Alpha ]
[ Into ID frame_id ]
[ Style [ Font... ] [ Norefresh ] [ Count { On | Off } ] ]
[ Text { style_name }
  { [Line Pen...] | [ Region Pen... ] [ Brush... ] | [ Symbol... ] |
    Collection [ Symbol... ] [ Line Pen... ] [ Region Pen... ] [
Brush... ] }
  [ Count count_value ]
  [ Display { On | Off } ] ]
[ Text . . . ] ]

```

map_window_id — целочисленный идентификатор окна, для получения которого используются [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#). Если этот идентификатор указывает на карту в окне **Отчёт**, то легенда будет создана в окне.

map_layer_id или *map_layer_name* определяют слой карты; это может быть целая величина Smallint (например, используйте 1 для определения самого верхнего слоя, не считая косметического) или строковая величина, соответствующая имени таблицы, отображенной на карте. Для тематического слоя необходимо определять параметр *map_layer_id*.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "cm" для сантиметров). Список "бумажных" единиц измерения приведен в разделе: [Оператора Set Paper Units](#).

frame_title — строковая переменная, определяющая заголовок раздела легенды. Если будет определено предложение заголовка раздела **Title**, то будет использоваться эта величина вместо величины *def_frame_title*.

frame_subtitle — строковая переменная, определяющая подзаголовок раздела легенды. Если предложение **Subtitle**, определяющее подзаголовок, задано для раздела легенды, то оно будет использоваться вместо параметра *def_frame_subtitle* из предложения **Default Frame Subtitle** (смотрите раздел: [Стандартные параметры раздела](#)).

number_of_columns — количество колонок, которые будут показаны в разделе.

frame_height — используется для запоминания высоты рамки легенды, изначально задаваемой предложением **Column**. Представляет высоту рамки в "бумажных" единицах. Значение записывается в WOR-файл, если размер рамки был изменен вручную.

priority_n — целочисленное значение, обозначающее порядок сортировки разделов легенды, которая встраивается в окно **Отчёт**.

framename — строка, определяющая название раздела легенды в окне **Отчёт**.

Значения *region_height* и *region_width* представляют новые высоту и ширину условных знаков площадных объектов. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров.

Значение *line_width* представляет ширину линейного условного знака. Можно задавать от 12 до 144 пунктов, от 1 до 12 пик, от 0.666667 до 2 дюймов, от 4.23333 до 50.8 миллиметров или от 4.23333 до 50.8 сантиметров.

column — имя атрибутивной колонки из раздела слоя таблицы.

Expression — корректное MapBasic-выражение.

t_value — толщина цветовой полосы для растровой легенды. Единицами измерения являются дюймы, если другие единицы измерения не заданы с помощью оператора **Units**.

l_value — длина цветовой полосы для растровой легенды. Единицами измерения являются дюймы, если другие единицы измерения не заданы с помощью оператора **Units**.

max_value — максимальное количество цветов, которое отображается в легенде раstra.

Frame_id — идентификатор рамки (ID от 1 до общего количества рамок) для пустой рамки отчёта в текущем окне отчёта.

style_name — строковая переменная, которая указывает к какому типу: символу, линии или области относится объект в разделе легенды.

count_value — значение счётчика определённой строки растровой легенды.

Описание

Как минимум, требуется задать одно предложение **Frame From Layer**. Каждый параметр **Frame From Layer** представляет одну легенду, создаваемую в окне **Легенда**.

Все предложения, относящиеся ко всему окну **Легенда** и стандартным параметрам, должны задаваться после первого предложения **Frame**.

Примечание: Используя **оператор Shade** и **оператор Set Legend**, по-прежнему можно создавать и изменять тематические легенды, но добавлять тематические разделы в окна **Легенда** можно только с помощью оператора **Create Designer Legend** (задавая идентификатор тематического слоя с помощью параметра **Frame From Layer id**).

Предложение **Position** позволяет задавать положение раздела, если определён параметр **Custom** в MapBasic-операторе, который определяет окно. X-координата измеряется от левой

стороны окна **Легенда**, а Y-координата от верхней стороны окна **Легенда**. Начало координат (0,0) находится в левом верхнем углу окна **Легенда**.

Если у раздела имеется заголовок, задаваемый параметром **Title**, то этот параметр переопределяет стандартный заголовок, который, обычно, задаётся параметром **Default Frame Title**. Если для оформления шрифта заголовка используется параметр **Font**, то стандартный шрифт заголовка раздела переопределяется, если такой параметр существует.

Если у раздела имеется подзаголовок, задаваемый параметром **Subtitle**, то этот параметр переопределяет стандартный заголовок, который, обычно, задаётся параметром **Default Frame Subtitle**. Если для оформления шрифта подзаголовка используется параметр **Font**, то стандартный шрифт подзаголовка раздела переопределяется, если такой параметр существует.

Предложение **Height** используется вместо предложения **Columns**, если пользователь изменял размер раздела. Если наличествуют обе, то используется предложение **Columns**.

Параметры **Columns** и **Height** можно применять и к тематическим, и к картографическим легендам. Однако, все свойства тематических легенд должны быть заданы с помощью **оператора Set Legend**.

Предложение **Border** используется только для того, чтобы задавать рамочку и фон легенды, когда она встраивается в окно **Отчёт**.

Параметр **Pen** всегда завершается корректным **предложением Pen**. Это предложение позволяет включать (плотное заполнение цветом) или выключать (штриховка) и задавать цвет рамочки вокруг раздела, который встраивается в окно **Отчёт**.

Brush — корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение используется только для того, чтобы задавать цвет фона легенды, когда она встраивается в окно **Отчёт**.

Предложение **Priority** позволяет задавать обратный порядок сортировки разделов легенды, которые встраиваются в окно **Отчёт**.

Предложение **Name** позволяет задавать название легенды, которая встраивается в окно **Отчёт**.

Предложение **Region Height** позволяет задавать высоту условных знаков легенды и переопределяет стандартное значение, заданное для раздела.

Предложение **Region Width** позволяет задавать ширину условных знаков легенды и переопределяет стандартное значение, заданное для раздела.

Предложение **Line Width** позволяет задавать толщину условных знаков линий легенды и переопределяет стандартное значение, заданное для раздела.

Auto Font Size — позволяет включать и выключать режим подстройки размера шрифтов условных знаков на основе размеров шрифта, заданных параметром **Style** стиля оформления.

С помощью параметра **Order** добавлена возможность либо включать сортировку, либо изменять способ сортировки рядов разделов легенд. Легенды карты можно сортировать либо по стилям, либо задавая индивидуальный порядок сортировки. Сортировать можно по следующим

значениям параметра сортировки: **Default**, **Ascending**, **Descending** или **Custom**. Параметр **Order** записывается в рабочем наборе, только если легенда отсортирована в возрастающем, убывающем или индивидуальном порядке.

Порядок сортировки по умолчанию **Default** — это порядок сортировки, который устанавливает **Конструктор легенды**. Если легенда создаётся по уникальным стилям оформления объектов карты, то ряды условных знаков раздела будут перечислены в порядке, в котором объекты используются на карте. Если задан индивидуальный порядок сортировки условных знаков раздела легенды карты — **Custom**, то значение идентификатора *id* начинается с 1, увеличивается сверху вниз. Если проверить порядок перечисления условных знаков раздела в диалоге **Свойства раздела легенды**, то первый условный знак имеет идентификатор равный 1, номера следующих увеличиваются по мере перечисления в списке. Более подробные сведения содержит раздел **Настройка порядка сортировки** и раздел об **операторе Alter Designer Frame**.

Into ID позволяет вставлять легенду в пустую рамку отчёта. Если рамка не пустая, то появится сообщение об ошибке. Если *frame_id* равняется 1 и рамка не пустая, то появится сообщение об ошибке "Рамка отчёта 1 должна быть пустой". Параметр **Into ID** является дополнительным, необязательным и любые легенды созданные или добавленные без него автоматически располагаются вдоль стороны карты, что является обычной стратегией. Окно **Карта** должно располагаться на той же странице отчёта, на которой размещается легенда: добавление легенды на другую страницу отчёта или, когда карта отсутствует в окне **Отчёт**, приводит к ошибке: "Родительская карта должна быть внедрена на текущей странице окна отчёта.." Идентификатор этой ошибки 1067.

Когда используется предложение **Into ID**, другие предложения пропускаются и не приводят к появлению сообщений об ошибках. Такими предложениями могут быть: **Position**, **Units**, **Height**, **Border**, **Brush**, **Priority** и **Name**. Размер рамки задаётся таким, чтобы в ней оптимально поместилась легенда.

Параметр **Using** позволяет определять, как создаётся легенда векторного слоя. Он ссылается на элементы таблицы при помощи предложений **Column** и **Label**.

Если задано предложение **Column**, то значение этого параметра должно представлять имя атрибутивной колонки в таблице, соответствующей слою, представленному в разделе, либо должно стоять слово **Object**, обозначающее объектную колонку (это значит, что стили легенды основываются на уникальных стилях MAP-файла). По умолчанию используется **Object**.

FromMapCatalog ON отыскивает стили из MapCatalog для таблицы прямого доступа. Если таблица не является таблицей прямого доступа, MapBasic возвращается к заданному по умолчанию поведению для таблицы с непрямым доступом вместо того, чтобы допустить ошибку. Стандартное поведение для таблицы с отключённым доступом это **FromMapCatalog Off** (то есть, используются стили карты).

FromMapCatalog OFF отыскивает уникальные стили Карты для таблицы прямого доступа сервера. Эта таблица должна быть таблицей прямого доступа, которая поддерживает стилизацию каждой записи. Если таблица прямого доступа не поддерживает стили записей, то применяется заданное по умолчанию поведение для таблиц прямого доступа — используются заданные по умолчанию стили из MapCatalog (**FromMapCatalog ON**).

Предложение **Label** — это либо MapBasic-выражение *expression* для подписи, либо слово **Default** (последнее означает, что при создании текста используется стиль оформления рамки по умолчанию, если только предложение стилизации не описывает текст). Стандартным значением является **Default**.

Примечание: Сначала, для каждого векторного слоя, порождающего раздел легенды, исследуется соответствующий TAB-файл на наличие значений метаданных для заголовка легенды, подзаголовка, структуры колонок и т.п. Если метаданные присутствуют, то ими переопределяются значения, заданные предложением **Using**. Смотрите разделы: [функция GetMetadata\\$\(\)](#) и [функция Metadata](#).

Примечание: Когда MapInfo Pro создает легенду на основе типов объектов, то сначала рисуются точечные символы, затем линии, а затем области. Объекты типа коллекция изображаются последними. В пределах объекта типа коллекция порядок рисования таков: точки, линии, объекты.

Предложение **Display { Gap | Palette | Bar }** применяется только к растровым легендам.

Display Bar — позволяет отображать легенду непрерывного растра с помощью цветовой полосы. Для такого предложения существуют параметры, опции, позволяющие задавать размер полосы и ориентацию: толщина — **Thickness**, длина — **Length**, единицы измерения — **Units** и для ориентации — (**Horizontal** или **Vertical**). Параметр **Units**, задающий единицы измерения, является дополнительным, необязательным. Если от не задаётся, то MapInfo Pro использует дюймы. Если пропущен параметр, отвечающий либо за толщину, либо длину, либо ориентацию, MapInfo Pro использует значения, заданные предложением **Default Frame Bar**.

Display Palette — позволяет отображать легенду непрерывного растра в виде условных знаков, содержащих цвета заливок, с подписями. Размер условных знаков зависит от параметров, которыми задаются ширина и высота разделов, либо, которые используются в качестве стандартных, принимаемых по умолчанию.

Display Gap — также позволяет отображать условные знаки цветов заливки, но для тематических, классифицированных растровых слоёв. Если предложение **Display** пропущено, а исходный растровый слой является:

- непрерывным, MapInfo Pro использует предложение **Display Bar**, когда имеется более 20 цветов, в остальных случаях используется **Display Palette**;
- тематическим, классифицированным — MapInfo Pro использует предложение **Display Gap**.

Предложение **Filter Max** позволяет задавать максимальное количество строк или цветов, создаваемых для легенды растра. Это предложение является дополнительным, необязательным и является корректным только для разделов растровых легенд. Стандартным значением является 100. Если это предложение не используется, то MapInfo Pro отобразит в растровой легенде 100 оттенков цвета. Возможные значения — от 1 до 1600.

Предложение **Sort Alpha** используется с тематическими, классифицированными разделами растровых легенд. Оно позволяет сортировать подписи в алфавитном порядке.

Предложение **Style** и ключевое слово **NoRefresh** позволяют создавать собственные легенды, которые не будут изменяться при обновлении легенды. Однако, предложение **Style** должно

содержать собственный, индивидуальный список стилей, используемых в легенды. Это выполняется с помощью параметров операторов в предложении **Text** и, соответственно, в предложениях **Line**, **Region**, **Symbol** или **Collection**.

Предложение **Font** позволяет задавать стиль оформления текста подписей строк и переопределяет параметры шрифта, заданные оператором **Default Frame Style**.

Параметр счётчика **Count { On | Off }** применяется только к растровым легендам и позволяет управлять показом значения количества записей счётчика строк. По умолчанию этот режим **выключен**. Этот параметр игнорируется, когда растровая легенда отображается в иде полоски цветов.

Каждая строка легенды начинается с предложения **Text**, за которым следуют предложения **Line**, **Region**, **Symbol** или **Collection**.

В конце предложения **Text** можно применять дополнительное выражение **Count**, которое применяется только к растровым легендам. Если растровый слой содержит счётчик записей, то они будут сохранены и использоваться в каждой строке легенды. Это не следует менять, поэтому **оператор Alter Designer Frame** не поддерживает изменения такого значения.

В конце каждого предложения **Style** может применяться дополнительный оператор **Display**. **Display On** — делает ряд видимым, а **Display Off** — невидимым. Если параметр **Display** отсутствует, то ряд с условным знаком видим. Параметр **Display** записывается в рабочем наборе только для невидимых в разделе легенды условных знаков.

Примеры

```
Open Table ApplicationDirectory$() + "MyTable.tab" As mytable
Open Table ApplicationDirectory$() + "States.tab" As states
Open Table ApplicationDirectory$() + "City_125.tab" As cities
Open Table ApplicationDirectory$() + "Us_hiway.tab" As hiway
Map From cities, hiway, states, mytable
Create Designer Legend From Window FrontWindow()
Position (8, 0) Units "in"
Width 3 Units "in"
Height 6 Units "in"
Window Title "Legend Designer Window Test"
Custom
Default Frame Title "# Legend" Font ("Calibri",1,12,16711680)
Default Frame Subtitle "#" Font ("Arial",2,10,255)
Default Frame Style "%" Font ("Lucida Calligraphy",0,8,16732240)
Frame From Layer Cities
Using column object
label default
  Position (.5, 0) Units "in"
  Subtitle "125 Largest Cities"
Frame From Layer hiway
Using column object
label default
  Position (.5, .75) Units "in"
  Title "US Highways"
```

```

Subtitle "Interstates"
Frame From Layer states
Using column object
label default
Position (.5, 1.5) Units "in"
Title "United States"
Subtitle "State Boundaries"
Frame From Layer mytable
Using column object
label default
Position (.5, 2.5) Units "in"
Title "MyTable Objects"
Subtitle "Special Objects"
Columns 4

```

В следующем примере показано, как пользоваться оператором **Create Designer Legend** для настройки порядка перечисления строк легенды. Каждое предложение **Text** описывает отдельную строку легенды, а строки легенды записываются в рабочем наборе с стандартным порядком перечисления, где каждое предложение **Text** пронумеровано, начиная с единицы (1) в порядке увеличения.

```

Create Designer Legend Frame From Layer 1
  Order Custom 2, 3, 4, 1
  Style Font ("Arial",0,8,0)
Text "LIBRARY"
  Symbol (41,8388608,12,"MapInfo Miscellaneous",0,0)
Text "PRE-SCHOOL - PUBLIC"
  Symbol (38,16711935,12,"MapInfo Real Estate",0,0)
Text "PRIMARY SCHOOL - PRIVATE"
  Symbol (38,32768,12,"MapInfo Real Estate",0,0)
Text "PRIMARY SCHOOL - PUBLIC"
  Symbol (38,32768,12,"MapInfo Real Estate",0,0)

```

Предложение **Order Custom**, из примера выше, позволяет задавать индивидуальный порядок перечисления строк, описывая, как организовать идентификаторы строк. **Order Custom 2,3,4,1** – передаёт легенде порядок перечисления, начиная со строки номер два (2), за которой следует третья (3) и четвёртая (4), а первое предложение **Text** в конце списка:

2. Text "PRE-SCHOOL - PUBLIC" Symbol (38,16711935,12,"MapInfo Real Estate",0,0)
3. Text "PRIMARY SCHOOL - PRIVATE" Symbol (38,32768,12,"MapInfo Real Estate",0,0)
4. Text "PRIMARY SCHOOL - PUBLIC" Symbol (38,32768,12,"MapInfo Real Estate",0,0)
1. Text "LIBRARY" Symbol (41,8388608,12,"MapInfo Miscellaneous",0,0)

В результате текст легенды будет выглядеть следующим образом:

```

PRE-SCHOOL - PUBLIC
PRIMARY SCHOOL - PRIVATE
PRIMARY SCHOOL - PUBLIC
LIBRARY

```

Создание легенд в окне конструктора отчёта

После того, как карта вставлена в окно **Отчёт**, любые легенды, созданные для карты, также добавляются в тоже окно **Отчёт**. Можно создавать единственный набор легенд для отдельной карты. Набор легенд состоит из легенд для каждого слоя карты. Если с единственной картой связано несколько окон легенды, то только разделы первого окна легенды, созданного на карте, помещаются в отчёт.

Некоторые предложения оператора **Create Designer Legend**, разрешенные синтаксисом оператора, не применяются при создании рамки с легендой карты, включенной в окно **Отчёт**. Предложения **Window Title**, и **Legend Text Frame** игнорируются.

Если повторно использовать оператор **Create Designer Legend** для той же карты в окне **Отчёт**, появится сообщение о том, что карта уже содержит одну или несколько легенд. В этом случае следует использовать **оператор Add Designer Frame**.

Создание легенд растрового слоя

Можно использовать растровый слой в качестве параметров *map_layer_id* и *map_layer_name*, когда растровый слой поддерживает легенды. Только для форматов файлов изображений, которые поддерживаются в расширенном варианте растрового модуля, имеется возможность создания растровых легенд. Список форматов файлов, поддерживаемых в расширенном варианте растрового модуля, можно найти на вкладке **PRO**, если нажать кнопку **Дополнительно**. В разделе **MapInfo Pro растр** нажмите **Настройки**, чтобы проверить список поддерживаемых растровых форматов.

Большая часть оператора **Create Designer Legend** может использоваться для создания растровых легенд, кроме предложения **Using**. Когда в предложении **Frame From Layer** содержится предложение **Using**, растровый слой возвращает сообщение об ошибке и легенда не создаётся. Предложение **Using** не применяется к растровым разделам легенд, поскольку оно ссылается на элементы векторной таблицы, такие как колонки и выражения для подписей.

Настройка порядка сортировки

Оба **оператора Create Designer** и **Alter Designer Frame** теперь содержат параметр Order, с помощью которого можно задавать индивидуальный порядок сортировки – Custom.

```
Custom id | id : id [ , id | id : id ... ]
```

Где диапазон идентификаторов рядов условных знаков задаётся в возрастающем порядке (Id2 > Id1):

```
Id1 : Id2
```

Изменения порядка перечисления списка

Синтаксис настройка порядка сортировки такой же, как и в предложении **Order**, которое описано в разделе об **операторе Set Map**. Этот параметр довольно легко использовать, когда требуется изменить порядок перечисления в начале списка, но возникают сложности, если требуется изменить порядок перечисления ближе к концу списка. Например, если требуется изменить порядок перечисления трех используемых в легенде рядов условных знаков на обратный:

```
Order Custom 3, 2, 1
```

Остальные ряды условных знаков можно не трогать. Если имеется 10 рядов условных знаков, а требуется изменить порядок перечисления двух последних, то придется перечислять идентификаторы в таком порядке:

```
Order Custom 1, 2, 3, 4, 5, 6, 7, 8, 10, 9
```

Для того чтобы сократить запись, можно использовать следующий синтаксис:

```
Order Custom 1:8, 10, 9
```

Разметка диапазонов значений

С помощью двоеточия (:) вы можете размечать диапазоны значений:

```
Long form:
Order Custom 2, 5, 6, 7, 8, 9, 10, 1, 3, 4
```

```
Short form:
Order Custom 2, 5:10, 1, 3, 4
Order Custom 2, 5:10, 1, 3:4 (same as above but also valid)
Order Custom 2, 5:10, 1 (same as above but also valid)
```

```
Long form:
Order Custom 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18,
19, 20, 11
```

```
Short form:
Order Custom 1:10, 12:20, 11
```

В списке не должно быть повторений – они порождают ошибки. Следующая запись ошибочна:

```
Order Custom 1:5, 8, 4:7
```

потому что в ней повторяются значения 4 и 5. Чтобы это увидеть, перепишем список полностью:

```
Order Custom 1, 2, 3, 4, 5, 8, 4, 5, 6, 7
```

Альтернативный синтаксис можно использовать при создании и изменении легенды в окне **Легенда**. В рабочих наборах короткий вариант синтаксиса используется, когда разделы в окне **Легенда** содержат более 50 рядов условных знаков, перечисленных в индивидуальном порядке.

Оператор Create Ellipse

Назначение

Создает эллипсы и окружности. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Ellipse
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 )
[ Pen... ]
[ Brush... ] [ Priority n ] [ Name framename ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

x1, *y1* – координаты одного угла прямоугольника, описывающего эллипс;

x2, *y2* – координаты противоположного по диагонали угла прямоугольника;

Pen – корректное **предложение Pen**, задающее стиль линии.

Brush – корректное **предложение Brush**, задающее штриховку.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окне **Отчёт**.

Framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Результатом действия оператора **Create Ellipse** является новый объект типа "Эллипс" или "Окружность". MapBasic создает объект, вписывая его в прямоугольник, задаваемый координатами двух противоположных углов. Если оператор задает квадрат, то будет создана окружность, иначе – эллипс.

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно, объект помещается на заданное место в окне (например, на изменяемый слой). Если предложения **Into** вообще нет в операторе, MapBasic попытается создать эллипс в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система не объявлялась, то координаты будут принимать значения широты и долготы. Используйте **оператор Set CoordSys** для того, чтобы задать в среде MapBasic другую координатную систему. При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: *X*-координата – это расстояние от левого края листа до точки, и *Y*-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**. Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Необязательный параметр **Pen** начинает предложение, задающее стиль линии. Если не задан параметр **Pen**, в операторе **Create Ellipse** будет использован стандартный стиль линии MapInfo Pro (заданный в диалоге команды **Стиль линий**). Аналогично, **предложение Brush** задает стиль штриховки.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте **предложение Priority**, для того чтобы присваивать значение *Z-Order* новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

См. также:

предложение Brush, **функция CreateCircle()**, **оператор Insert**, **предложение Pen**, **оператор Update**

Оператор Create Frame

Назначение

Создает новый объект "Рамка" в окне отчета. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Frame
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 )
[ Pen... ]
[ Brush... ]
[ Priority n ]
```



```
[ Name framename ]
[ Title title ]
[ From Window contents_win_id ]
[ FillFrame { On | Off } ]
[ Metadata SetKey keyname To keyvalue ]
```

window_id — идентификатор окна.

var_name — имя объектной переменной;

x1, y1 — координаты одного угла рамки;

x2, y2 — координаты другого угла рамки;

Pen — корректное предложение **Pen**, задающее стиль линии.

Brush — корректное предложение **Brush**, задающее штриховку.

n — целое число, указывающее на значение Z-Order объектов (рамки) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

Frame_name — строка, определяющая имя рамки с изображением в окне конструктора отчёта. Если рамке с изображением в окне **Отчёт** присваивается имя, то оно будет сохранено в рабочем наборе. Когда параметр **Name** записывается в рабочем наборе, то номер версии WOR-файла меняется на 1500.

title — строка, задающая заголовок окна, изображение из которого будет помещено в рамку (например, "WORLD Map"; не имеет смысла использовать, если в операторе используется предложение **From Window**).

contents_win_id — идентификатор окна, изображение из которого будет помещено в рамку, целое число.

key_name — строковая величина, представляющая собой имя ключа метаданных.

keyvalue — строка, представляющая значение, которое требуется присвоить ключу.

Описание

Результатом действия оператора **Create Frame** является новый объект типа "Рамка" в окне отчета. Этот оператор работает и с окнами **Отчёт**, и с классическими окнами отчёта. Если не задан параметр *layout_win_id*, то создается новая рамка в самом верхнем окне отчета.

Если используются координаты план-схемы (X, Y), чтобы разместить рамку, появится сообщение об ошибке. Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Параметр *window_id* задаёт окно отчёта, в которое будет добавлена новая рамка, а параметр *contents_win_id* задаёт окно, которое появится в новой рамке. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после

открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Рамки в окне **Отчёт** не могут размещаться в положении, имеющем отрицательные координаты выше верхней границы и левее левого края макета отчёта. При размещении рамки MapInfo Pro преобразует отрицательные значения X и Y в нулевые (0). (Классическое окно отчёта допускает отрицательные значения координат.)

Стиль линии для рамки задает **предложение Pen**, а стиль штриховки — **предложение Brush**.

Предложение **Title** является альтернативным способом задания окна, изображение из которого будет показано в рамке. Например, для окна карты, в котором показаны данные таблицы WORLD, предложение **Title** будет таким: **Title "WORLD Map"**. Если параметр *title* не указывает на существующее окно или если *title* является пустой строкой (""), оператор создаст пустую рамку. Если в операторе присутствуют сразу два предложения: **Title** и **From Window**, то будет исполнено только последнее из них.

Предложение **From Window** указывает, содержимое какого окна будет показано в рамке. Например, чтобы в рамке показывалось окно карты, задайте оператора **From Window i_map**, где *i_map* является идентификатором окна карты. Для этого необходимо знать идентификатор окна. Окно должно быть уже открыто.

При создании рамки для окна карты, можно применять предложение **FillFrame**, чтобы управлять заполнением рамки картой. Задание **FillFrame On** помещает всю карту в рамку. Если задать режим **FillFrame Off** или опустить все предложения **FillFrame**, пропорции рамки в Отчете будут созданы с учетом пропорций экрана, и далее пропорции окна отчёта будут изменяться вслед за изменениями пропорций окна карты.

MapInfo Pro запоминает установки в окне отчета, вставляя оператор **Create Frame** в файл рабочего набора. Чтобы увидеть, как записывается оператор **Create Frame**, создайте отчет, сохраните Рабочий набор и изучите его текст в текстовом редакторе.

Примечание: При работе с существующим отчётом идентификатор окна рамки с картой, входящей в отчёт, можно получить, вызвав **функцию LayoutItemInfo()** с атрибутом LAYOUT_ITEM_INFO_WIN.

Предложение **Metadata SetKey** помогает управлять метаданными, которые хранятся в рамке отчёта.

Пример

Следующие примеры показывают, как можно создавать разделы для картографических и для тематических легенд.

Тематические легенды обрабатываются особым образом. Чтобы создать раздел в окне тематической легенды, нужно использовать предложение **Title** вместо **From Window**.

```
Dim i_layout_id, i_map_id As Integer
Dim s_title As String
```

```
' here, you would store the Map window's ID in i_map_id,
' and store the Layout window's ID in i_layout_id.
' To obtain an ID, call FrontWindow() or WindowID().

s_title = "Theme Legend of " + WindowInfo(i_map_id, WIN_INFO_NAME)
Set CoordSys Layout Units "in"
Create Frame
  Into Window i_layout_id
  (1,2) (4, 5)
  Title s_title
```

Для создания картографической легенды в том случае, если несколько карт сопровождаются легендами, нужно использовать предложение **From Window** с указанием того, какая из легенд (если их несколько) обрабатывается.

```
Dim i_cartlgnd_id As Integer

' here, you would store the Cartographic Legend window's ID
' in i_cartlgnd_id,
' To obtain an ID, call FrontWindow() or WindowID().

Create Frame
  Into Window i_layout_id
  (1,2) (4, 5)
  From Window i_cartlgnd_id
```

См. также:

[предложение Brush](#), [оператор Insert](#), [оператор Layout](#), [предложение Pen](#), [оператор Set CoordSys](#), [оператор Set Layout](#), [оператор Update](#)

Создание пустых рамок

Оператор **Create Frame** может также использоваться для создания пустых рамок окна **Отчёт**. Пустые рамки используются в качестве шаблонов отчётов, которые резервируют место рамкой любого размера, которую можно заполнить впоследствии.

Синтаксис

```
Create Frame
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 )
[ Pen... ]
[ Brush... ]
[ Priority n ]
[ Name framename ]
```

Если *window_id* ссылается на окно **Отчёт**, будет создана новая пустая рамка. Если этот параметр ссылается на классическое окно **Отчёт**, будет создана традиционная пустая рамка. О фоне классических окон **Отчёт** смотрите раздел: [Работа с классическим окном отчёта](#).

Если параметр **Name** пропущен, то он будет заменён пустой строкой текста ("" – пустая строка). Имена будут отображаться в центре рамки. Длинные имена (имена длиннее ширины пустой рамки) будут отображаться с переносом.

Если параметр **Pen** пропущен, то будет использоваться стандартный стиль оформления линий – сплошная черная линия толщиной в один пиксель.

Если параметр **Brush** пропущен, то будет использоваться стандартный стиль оформления областей без заполняющей штриховки. Если параметр **Priority** пропущен, то рамка будет добавлена самой верхней.

Пустая рамка пересчитывается как любая другая и добавляется в счётчик, значение которого возвращается `LayoutInfo()`. Свойства **Pen**, **Brush**, **Name** и **Priority** можно менять после создания рамки с помощью оператора **Alter Designer Frame**. Положение и размер также можно менять либо с помощью оператора **Alter Designer Frame**, либо в графическом интерфейсе программы. Граница рамки и заполнение её штриховкой (параметры **pen** и **brush**) можно менять в графическом интерфейсе программы в диалоге **Свойство рамки**.

Добавление контента в пустые рамки

Повторное использование пустой рамки

В пустые рамки можно вставлять только карты и списки. Существует два способа, которыми можно вставлять карту или список в пустую рамку:

1. С помощью оператора **Set Designer Frame**

```
Set Designer Frame
[ Window layout_window_id ]
[ ID empty_frame_id ]
From { Window map_or_browser_window_id }
```

Если **ID** является пустой рамкой, то, когда после него следует предложение *From Window*, пустая рамка заполняется копией исходной карты или списка.

2. С помощью операторов **Map From** и **Browse From**

Используйте эти операторы, чтобы вставлять в пустую рамку открытые таблицы. Например,

```
Map From World, World_Cities Into Window layout_win_id Id empty_frame_id
```

При этом будут созданы карты из открытых таблиц, которые называются **World** и **World_Cities**, и вставлены в пустую рамку окна конструктора отчётов. Карта будет отображаться в стандартном виде. Будут использованы свойства рамки, заданные ещё для пустой рамки (стили линии и штриховки, положение, размер, имя и параметр **z-order**).

Прежде повторно использовать уже заполненную рамку, её требуется очистить от содержимого.

```
Set Designer Frame [ Window layout_window_id ] Id { frame_id } Clear
```

После выполнения такого оператора остаётся пустая рамка того же размера и в том же месте, где была заполненная рамка, присвоенное которой имя сохраняется. Контент рамки будет уничтожен; карта или список пропадут. Другие свойства, например, линия, используемая для оформления границы, заполняющая штриховка и порядок размещения останутся неизменными.

Списки, карты, легенды, фигуры, изображения и текст можно удалять с помощью этой команды. Команду также можно использовать на пустой рамке. Если очищается рамка с картой, которая имеет ассоциированные с ней рамки легенд, рамки с легендами будут удалены из отчёта.

Оператор Create Grid

Назначение

Создает грид-файл, который MapBasic показывает в виде растровой таблицы в окне карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Grid
From tablename
With expression [ Ignore value_to_ignore ]
Into filespec [ Type grid_type ]
[ Coordsys... ]
[ Clipping { Object obj } | { Table tablename } ]
Inflect num_inflections [ By Percent ] at
color : inflection_value [ color : inflection_value ...]
[ Round rounding_factor ]
{[ Cell Size cell_size [ Units distance_unit ] ] | [ Cell Min n_cells
]}
[ Border numcells ]
Interpolate With interpolator_name Version version_string
Using num_parameters parameter_name : parameter_value
[ parameter_name : parameter_value ... ]
```

tablename – это псевдоним имени открытой таблицы, из которой берутся точки для расчета поверхности (грида).

expression – это выражение, которое выделяет необходимую часть таблицы, например, имя колонки.

value_to_ignore – значение, которое будет проигнорировано; это обычно ноль. Строка, в которой присутствует игнорируемое значение, не будет учитываться при создании регулярной поверхности.

filespec – определяет полный путь и новое имя грид-файла. У этого нового файла будет расширение MIG.

grid_type – строка, в которой задан тип файла создаваемой регулярной поверхности. По умолчанию это MIG-файл, создаваемый с помощью модуля MIG.GHL. Если вы используете собственный программный модуль создания поверхности, то задайте его явно его имя, например, "DTED.GHL".

Coordsys – необязательный параметр CoordSys, определяющий координатную систему создаваемой регулярной поверхности. Если этот параметр не используется, то файл регулярной поверхности будет иметь ту же систему координат, что и исходная таблица. См. подробности в разделе: [Предложение CoordSys](#).

obj – это объект, который обрезает ячейки грида. Будет отображена только часть ячеек внутри такого объекта. Если ячеек грида внутри такого объекта нет, то никаких значений для ячеек записано не будет, и запишутся ячейки с нулевыми значениями.

tablename – имя таблицы, содержащей объекты типа полигонов, которые будут объединены в единый полигон и будут использованы для обрезания ячеек грида;

num_inflections – числовое выражение, определяющее значения для пары *color:inflection_value*.

color – это выражение для обозначения цвета для пары *color:value inflection*;

inflection_value – числовое выражение, определяющее числовое значение для пары *color:inflection_value* в виде числа или процента.

rounding_factor – это числовое выражение, задающее величину округления при вычислении переломного значения.

cell_size – числовое выражение, определяющее размер ячейки грида в единицах расстояния;

distance_unit – строковое выражение, определяющее единицы измерения размера ячейки. Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

n_cells – числовое выражение, которое определяет высоту и ширину грида в количестве ячеек;

numcells – число ячеек, которое будет добавлено вокруг границы грида. Число *numcells* задается для увеличения размер грида сверху, снизу, слева и справа;

interpolator_name – строковое выражение, определяющее имя интерполятора, который используется для создания грида; В MapInfo встроены интерполяторы "IDW" и "TIN."

version_string – это строковое выражение, определяющее версию интерполятора. (IDW-интерполятор версии 100 поставляется с MapInfo Pro 5.0, а интерполяторы TIN и IDW версии 200 – с MapInfo Pro 5.5 и выше.)

num_parameters – это числовое выражение, определяющее число параметров интерполятора - т.е. число пар *parameter_name: parameter_value*.

- *parameter_name* – строковое выражение, определяющее имя в паре *parameter_name: parameter_value*.
- *parameter_value* – числовое выражение, значение в паре *parameter_name: parameter_value*.

Описание

Тематическая карта поверхности представляет собой непрерывную растровую поверхность, полученную путем интерполяции точечных данных. Оператор **Create Grid** берет данные из колонок, которые находятся в таблице с точечными данными, и передает координаты этих точек и их числовые значения интерполятору. Интерполятор создает растровый грид-файл, который MapBasic отображает в виде растровой таблицы в окне карты.

Оператор **Create Grid** считывает значения (x, y, z) из таблицы, определенной в предложении **From**. Он получает значения z, которые указаны в относящемся к данной таблице предложении **With**.

Размеры сетки могут быть определены двумя способами. Первый способ определяет размер ячейки грида, выраженной в единицах расстояния, например, милях. Другой способ заключается в задании количества ячеек грида по ширине и высоте. Например, если вы хотите получить грид размером не менее 200 ячеек по ширине и 200 ячеек по высоте, то надо будет определить "cell min 200". В зависимости от площади, покрываемой гридом, действительный размер грида будет не менее 200 на 200 ячеек.

Предложение **By Percent at** определяет, что пары *color:Inflection_value* состоят из значения цвета и процента. Если оно не задано, то значения *Inflection_value* представляют числа, такие как высоты или температуры.

Более подробно о регулярных поверхностях (гридах) рассказывается в разделе: [Описание поверхностей](#). Интерполяторы подробнее описаны в разделах: [Интерполятор IDW](#) и [Интерполятор TIN](#).

Пример

```
Open Table "C:\States.tab" Interactive
Map From States
Open Table "C:\Us_elev.tab" Interactive
Add Map Auto Layer Us_elev
set map redraw off
Set Map Layer 1 Display Off
set map redraw on

create grid
from Us_elev
with Elevation_FT
into "C:\Us_elev_grid"
clipping table States
inflect 5 at
  RGB(0, 0, 255) : 13
  RGB(0, 255, 255) : 3632.5
```

```

RGB(0, 255, 0) : 7252
RGB(255, 255, 0) : 10871.5
RGB(255, 0, 0) : 14491
cell min 200
interpolate
with "IDW" version "100"
using 4
  "EXPONENT": "2"
  "MAX POINTS": "25"
  "MIN POINTS": "1"
  "SEARCH RADIUS": "100"

```

См. также:

функция [GetGridCellValue\(\)](#), [GridTableInfo\(\)](#), функция [IsGridCellNull\(\)](#), функция [OverlayNodes\(\)](#), функция [RasterTableInfo\(\)](#), оператор [Set Map](#)

Построение поверхности по сетке

Тематическая карта поверхности представляет собой непрерывную растровую поверхность, полученную путем интерполяции точечных данных. Оператор Create Grid берет данные из колонок, которые находятся в таблице с точечными данными, и передает координаты этих точек и их числовые значения интерполятору. Интерполятор создает растровый грид-файл, который MapBasic отображает в виде растровой таблицы в окне карты. Оператор Create Grid считывает значения (x, y, z) из таблицы, определенной в предложении **From** *tablename*. Он получает значения z, которые указаны в относящемся к данной таблице предложении **With** и вычисляет значение для поверхности с использованием режимов выбранного интерполятора.

Размеры сетки могут быть определены двумя способами. Первый способ определяет размер ячейки грида, выраженной в единицах расстояния, например, милях. Другой способ заключается в задании количества ячеек грида по ширине и высоте. Например, если вы хотите получить грид размером не менее 200 ячеек по ширине и 200 ячеек по высоте, то надо будет определить "cell min 200". В зависимости от площади, покрываемой гридом, действительный размер грида будет не менее 200 на 200 ячеек.

Пример IDW-интерполяции с значениями переломов на заданных высотах:

```

Open Table "C:\MyData\States.tab" Interactive
Map From States
Open Table "C:\MyData\Us_elev.tab" Interactive
Add Map Auto Layer Us_elev
Set map redraw off
Set Map Layer 1 Display Off
Set map redraw on
Create Grid
  From Us_elev
  with Elevation_FT
  ignore 0
  into "C:\MapData\Us_elev_grid.mig"
  Type "mig.ghl"

```



```

CoordSys Earth Projection 1, 74
clipping table States
Inflect 6 at
  RGB(64, 0, 128) : -32808
  RGB(0, 128, 192) : -16404
  RGB(151, 255, 239) : -98
  RGB(254, 248, 199) : 33
  RGB(244, 171, 100) : 6566
  RGB(235, 95, 1) : 32808
  round 100
  cell min 200
interpolate
  with "IDW" version "100"
  using 7
    "AGGREGATION METHOD": "1"
    "BORDER": "0"
    "CELL SIZE": "8"
    "EXPONENT": "2"
    "MAX POINTS": "25"
    "MIN POINTS": "1"
    "SEARCH RADIUS": "800"

```

Пример TIN-интерполяции с переломами на заданных процентных значениях:

```

inflect 5 by Percent at
  RGB(0, 0, 255) : 0
  RGB(0, 255, 255) : 25
  RGB(0, 255, 0) : 50
  RGB(255, 255, 0) : 75
  RGB(255, 0, 0) : 100
Round 10
cell min 200
interpolate
  with "TIN" version "100"
  using 8
    "BORDER": "0"
    "CELL SIZE": "15"
    "DISTANCE": "80"
    "EXPONENT": "2"
    "FEATURE ANGLE": "45"
    "SEARCH RADIUS": "53"
    "TOLERANCE": "0.01 "
    "READ ONLY": "T"

```

Раскраска поверхности и переломные значения

После вычисления значений в ячейке, MapInfo Pro группирует их и назначает группам разные цвета в зависимости от максимального и минимального значений и способа раскраски.

Можно управлять раскраской, задавая цвета и переломы – пороговые значения, на которых происходит их смена. Допускается количество переломов от 2 до 255. В вычислениях пороговых

значений может быть также задействован режим округления. При вычислении интерполированных значений, у вас есть четыре способа вычисления переломов:

- одинаковое количество ячеек – переломы распределяются так, чтобы происходить через равное количество ячеек.
- одинаковые диапазоны – переломы распределяются по всему диапазону значений от минимума до максимума.
- заданное количество ячеек – количество ячеек, раскрашенных на основе одного цвета, задает пользователь.
- заданные диапазоны – диапазоны задает пользователь.

При использовании оператора Create Grid вы можете подставлять свои значения перелома и цветов.

Шаблоны поверхностей

При создании тематической карты поверхности в MapInfo Pro используется стандартный шаблон, варьирующий два цвета – от синего минимального до красного максимального. Минимальные и максимальные значения образуют диапазон, который можно разбить на интервалы, границы которых можно задавать значениями или процентными отношениями к диапазону. Границам интервалов можно сопоставить новые цвета, что отражается в легенде. Если значение в ячейке находится на границе (пороге) интервала, то она раскрашивается в соответствующий интервалу цвет. Если значения в ячейке попадает между двумя порогами интервалов, то для нее подбирается цвет, составленный из пороговых цветов.

См. также:

интерполятор IDW, интерполятор TIN

Интерполятор IDW

Интерполятор IDW лучше всего приспособлен для данных, значения которых мало зависят друг от друга и не оказывают друг на друга влияния, например, для данных о населении. Этот метод хорошо также работает с данным с большим разбросом значений. Сначала алгоритм IDW-интерполяции вычисляет размер ячеек сетки, которые должны покрыть область карты. Каждая точка из исходной таблицы вносит свой взвешенный вклад в вычисление значения для ячейки в зависимости от расстояния этой точки до центра ячейки. Как следует из названия метода – обратной (инверсной) взвешенной дальности, – чем дальше точка находится от ячейки, тем меньшее влияние она оказывает на результирующее значение в этой ячейке. Построение сетки в MapInfo Pro начинается с определения минимального прямоугольного покрытия (MBR) исходной таблицы. Сетка составляется из квадратных ячеек одного размера. Например, применяя готовый шаблон Поверхность стандартная к таблице "States" из демонстрационных данных MapInfo Pro, мы получаем сетку из 200 x 303 ячейки. Зная количество ячеек в сетке и размеры минимального прямоугольного покрытия, MapInfo Pro вычисляет размер ячеек: это квадраты 18.1 x 18.1 миль. Размер ячейки зависит от текущих единиц измерения в окне карты. Изменить их можно в диалоге **Карта > Режимы > Единицы измерения**.)

Режимы IDW-интерполятора настраиваются в диалоге, появляющемся при нажатии кнопки **Настройки** в диалоге **Создать тематическую Карту - Шаг 3 из 3** в процессе создания поверхности. В этом диалоге представлены размеры ячейки, которые используются в алгоритме интерполяции. Любое изменение этого количества изменяет и размер ячеек и частоту сетки. Имея размер ячейки и исходные данные, MapInfo Pro вычисляет значение для каждой ячейки. Это значение определяется вычислением усредненного вклада точек, лежащих внутри заданного радиуса поиска. Вклады от этих точек вычисляются в обратной зависимости от их расстояния до центра ячейки. В алгоритме IDW-интерполятора задается также значение экспоненты, которая регулирует влияние точек на результат. Чем выше экспонента, тем большее влияние оказывают более приближенные точки на значение ячейки. Значение экспоненты варьируется от 1 до 10. Вы также можете выбрать метод обобщения значений для z-значений тех точек из исходного массива, которые попали в область одной и той же ячейки. Методами обобщения являются усреднение, подсчет количества точек, сумма значений, минимум или максимум. Стандартным методом является усреднение.

Режимы IDW-интерполятора в операторе Create Grid

Для IDW-интерполятора MapInfo Pro вы можете задавать следующие параметры в операторе Create Grid:

```
Using num_parameters parameter_name : parameter_value
[ parameter_name : parameter_value ... ]
```

Можно задавать следующие пары *parameter_name* : *parameter_values*.

- "EXPONENT": "число" – минимум 1, максимум 10, стандартное значение 2. Чем выше это число, тем большее влияние оказывают на вычисления близкие точки.
- "SEARCH RADIUS": "число" – значение должно быть больше нуля, стандартное значение 10.
- "MIN POINTS": "число" – значение должно быть больше нуля, стандартное значение 1. Минимальное количество точек, нужных для вычисления значения в ячейке.
- "MAX POINTS": "число" – значение должно быть больше нуля, стандартное значение 25. Максимальное количество точек, нужных для вычисления значения в ячейке.
- "GRID HEIGHT": "число" – значение должно быть больше нуля, стандартное значение 100. Минимальное количество ячеек сетки в длину. На результат также влияет режим сохранения пропорций, но количество ячеек в длину не должно превышать этот минимум.
- "GRID WIDTH": "число" – значение должно быть больше нуля, стандартное значение 100. Минимальное количество ячеек сетки в ширину. На результат также влияет режим сохранения пропорций, но количество ячеек в ширину не должно превышать этот минимум.
- "CELL SIZE": "число" – размер ячеек сетки в текущих единицах измерения. Значение должно быть больше нуля, стандартное значение 100.
- "BORDER": "число" – количество ячеек за пределами граничного прямоугольника таблицы, на которые будет продолжена сетка. Это бывает полезно сделать, когда мы интерполируем поверхность по данным из таблицы точек, а результат представляем в области-врезке.

- "AGGREGATION METHOD": "число" – метод обобщения, для выбора одного значения, если в ячейку попадает несколько точек. По умолчанию используется метод усреднения значений, код 0. Большинство параметров интерполирования могут быть заданы в диалоге настройки интерполяции. Следующие коды передаются оператору вместо имен методов обобщения:

Среднее	0
Счетчик (count)	1
Минимум	2
Максимум	3
Сумма	4

Если пара *parameter name* : *parameter* не задана, используется стандартное значение. Если нужно задать все настройки интерполятора стандартными, установите **Using num_parameters** на 0.

См. также:

оператор Create Grid, параметр READ ONLY для TIN- и IDW-интерполяторов

Интерполятор TIN

Интерполятор TIN лучше всего работает с топографическими данными и с массивами, в которых данные меняются непрерывно, без резких скачков, такими как высоты и температуры. Алгоритм TIN-интерполятора создает сеть из треугольников на основе имеющегося набора точек с известными высотными отметками; он лучше подходит для построения цифровых моделей рельефа, чем IDW-интерполятор. Поверхность представляет собой массив состыкованных треугольников, близких по форме к равносторонним. Высоты точек, представляющих вершины треугольников, интерполируются на основе значений высот из базовой таблицы. При интерполяции не учитываются значения в соседних точках сетки, что исключает паразитные эффекты, свойственные IDW-интерполяции.

Настройки TIN-интерполяции могут варьироваться, с целью получения более или менее детального результата. С помощью параметра **Допуск** можно управлять учетом близко расположенных точек. Близость точек определяется долей длины диагонали покрывающего эти точки прямоугольника. Параметром **Расстояние** можно управлять результатами. Если оно не равно нулю, то при построении поверхности участвуют только те ребра и треугольники, которые попадают в сферу заданного радиуса вокруг узла. Этот режим полезно задавать, чтобы при триангуляции не терялись из виду впадины. Параметром **Угол** можно задавать величину острого угла. С помощью этого режима можно влиять на гладкость получаемой

поверхности. Если угол между двумя смежными полигонами больше этого значения, то их общее ребро считается "острым".

Режимы IDW-интерполятора настраиваются в диалоге, появляющемся при нажатии кнопки **Настройки** в диалоге **Создать тематическую Карту - Шаг 3 из 3** в процессе создания поверхности. В этом диалоге можно задавать количество ячеек. Любое изменение этого количества изменяет и размер ячеек и частоту сетки.

Режимы TIN-интерполятора в операторе Create Grid

Для TIN-интерполятора MapInfo Pro вы можете задавать следующие параметры в операторе Create Grid:

```
Using num_parameters parameter_name : parameter_value
[ parameter_name : parameter_value ... ]
```

Можно задавать следующие пары *parameter_name* : *parameter_values*:

- "TOLERANCE": "число" – принимает значения в диапазоне от .0001 до .01 включительно, стандартное значение равно .005.
- "DISTANCE": "число" – должно быть больше нуля и меньше, чем размер сетки.
- "FEATURE ANGLE": "число" – принимает значения в диапазоне от нуля до 180 включительно, стандартное значение равно 25 градусам.
- "MIN POINTS": "число" – значение должно быть больше нуля, стандартное значение 1. Минимальное количество точек, нужных для вычисления значения в ячейке.
- "MAX POINTS": "число" – значение должно быть больше нуля, стандартное значение 25. Максимальное количество точек, нужных для вычисления значения в ячейке.
- "GRID HEIGHT": "число" – значение должно быть больше нуля, стандартное значение 100. Минимальное количество ячеек сетки в длину. На результат также влияет режим сохранения пропорций, но количество ячеек в длину не должно превышать этот минимум.
- "GRID WIDTH": "число" – значение должно быть больше нуля, стандартное значение 100. Минимальное количество ячеек сетки в ширину. На результат также влияет режим сохранения пропорций, но количество ячеек в ширину не должно превышать этот минимум.
- "CELL SIZE": "число" – размер ячеек сетки в текущих единицах измерения. Значение должно быть больше нуля, стандартное значение 100.

Примечание: Если пара *parameter name* : *parameter* не задана, используется стандартное значение. Если нужно задать все настройки интерполятора стандартными, установите **Using num_parameters** на 0.

См. также:

параметр READ ONLY для TIN- и IDW-интерполяторов

Параметр **READ ONLY** для **TIN-** и **IDW-интерполяторов**

Если требуется, чтобы поверхность нельзя было изменить или перерисовать, задайте режим "READ ONLY": "T" в паре *parameter name : parameter value* либо для TIN, либо и IDW-интерполятора MapInfo Pro. Если этот режим задан, то следующие метаданные:

```
"\Interpolator\Parameter\READ ONLY" = "0"
```

будут записаны в TAB-файл и пользователь не сможет ни изменить, ни обновить поверхность через пользовательский интерфейс.

Оператор Create Index

Назначение

Создает индекс для колонки в открытой таблице. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Index On table ( column )
```

table – имя открытой таблицы;

column - имя колонки в открытой таблице.

Описание

Оператор **Create Index** создает индекс для определенной колонки открытой таблицы. MapInfo Pro использует проиндексированные колонки при выполнении таких операций, как: **Найти**. Индексы также улучшают выполнение запросов.

Примечание: MapInfo Pro не индексирует колонки таблицы, в которой есть несохраненные изменения. Для сохранения изменений используется **оператор Commit Table**.

Пример

Создается индекс колонки "Столица" в таблице WORLD:

```
Open Table "world" Interactive
Create Index on World(Capital)
```

См. также:

[оператор Alter Table](#), [оператор Create Table](#), [оператор Drop Index](#), [оператор Commit Table](#)

Оператор Create Legend

Назначение

Открывает новое окно тематической легенды для определенного окна карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Начиная с версии 5.0 MapInfo Pro оператор Create Cartographic Legend позволяет создать и отобразить легенду карты. См. подробности в разделе: [Оператор Create Cartographic Legend](#).

Синтаксис

```
Create Legend
[ From Window window_ID ]
[ { Show | Hide } ]
```

window_ID – целочисленный идентификатор открытого в MapInfo Pro окна карты.

Описание

Этот оператор создает дополнительное, "плавающее" окно тематической легенды в рабочем окне MapInfo Pro. (Последнее вы можете открыть, выполнив оператор Open Window Legend.)

Оператор **Create Legend** может быть вам полезен, если необходимо вывести на экран легенду карты, когда окно карты открыто, но не активно. Этот оператор полезен в приложениях, использующих т.н. "интегрированную картографию", в которых окно MapInfo вставляется в другую программу, например, созданную при участии Visual Basic. Концепция и описание интегрированной картографии содержатся в *Руководстве пользователя MapBasic*.

Если добавить предложение **From Window**, новая тематическая легенда считается порожденной определенным в этом предложении окном; иначе новое окно легенды считается порожденным последним активным окном карты.

Если в операторе используется ключевое слово **Hide**, то окно будет создано в скрытом состоянии. Вывести на экран скрытое окно можно, использовав оператор [Set Window...Show](#).

После выполнения оператора **Create Legend** определить идентификатор нового окна можно будет вызовом функции `WindowID(0)`. Этим идентификатором Вы можете пользоваться в последующих операторах (таких как [оператор Set Window](#)).

Новая тематическая легенда создается в соответствии с режимами наследования и стилизации, для задания которых используется [оператор Set Next Document](#).

См. также:

оператор [Create Cartographic Legend](#), оператор [Open Window](#), оператор [Set Next Document](#), оператор [Set Window](#)

Функция `CreateLine( )`

Назначение

Возвращает объект "линия". Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CreateLine( x1, y1, x2, y2 )
```

x1 – X-координата начальной точки линии (например, долгота), действительное число;

y1 – Y-координата начальной точки линии (например, широта), действительное число;

x2 – X-координата конечной точки линии, действительное число;

y2 – Y-координата конечной точки линии, действительное число.

Возвращаемая величина

Объект

Описание

Функция **CreateLine()** возвращает графический объект типа "Линия". Параметры X и Y задают координаты концов отрезка прямой линии в той координатной системе, которая была объявлена MapBasic ранее. По умолчанию, MapBasic использует долготу и широту. Используйте [оператор Set CoordSys](#) для задания другой системы.

Создаваемый линейный объект будет нарисован с использованием текущей установки стиля линии Pen. Чтобы нарисовать линию в другом стиле, нужно выполнить [оператор Set Style](#) перед вызовом **CreateLine()** или выполнить [оператор Create Line](#), задав при этом [предложение Pen](#).

Графический объект, созданный **функцией CreateLine()**, может быть присвоен объектной переменной, которая определяет значение уже существующей строки в таблице ([оператор Update](#)) или вновь созданной ([оператор Insert](#)). Перед созданием объекта в окне отчета не забудьте выполнить оператор [Set CoordSys Layout](#).

Пример

В примере используется **оператор Insert** для создания новой строки в таблице ROUTES. Функция **CreateLine()** используется внутри оператора Insert.

```
Open Table "Routes"
Insert Into routes (obj)
  Values (CreateLine(-72.55, 42.431, -72.568, 42.435))
```

См. также:

оператор Create Line, оператор Insert, оператор Update

Оператор Create Line

Назначение

Создает объект "прямая линия". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Line
[ Into { Window window_id | Variable var_name } ]
( x1, y1) ( x2, y2)
[ Pen... ] [ Priority n ] [ Name framename ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

x1, *y1* – координаты начала отрезка прямой;

x2, *y2* – координаты конца отрезка прямой линии.

Слово **Pen** начинает стандартное предложение для назначения стиля линии.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**.

Framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Результатом действия оператора **Create Line** является новый графический объект типа "Линия".

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно, объект помещается на

заданное место в окне (например, на изменяемый слой). Если **Into** вообще нет в операторе, MapBasic попытается создать объект в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется **оператор Set CoordSys**. При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: *X*-координата – это расстояние от левого края листа до точки, и *Y*-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Если явно задано **предложение Pen**, то оно определяет стиль линии; см. подробности в разделе: **Предложение Pen**. Если предложение *Pen* не задано оператор **Create Line** использует текущий в среде MapInfo Pro стиль линии.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение *Z-Order* новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

См. также:

функция CreateLine(), оператор Insert, предложение Pen, оператор Update

Оператор Create Map

Назначение

Изменяет структуру существующей таблицы, разрешая сопоставлять ее записям графические объекты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Map
For table [ BlockSize block_size ]
[ CoordSys... ] Using from_table]
```

table – имя открытой таблицы;

from_table – имя открытой таблицы, из которой будет извлечена координатная система.

block_size – значение в диапазоне от 0 до 32767. Если этот параметр не задан, то умолчанию используется значение равное 16384. Если значение меньше 512, то оно будет округлено до 512.

Описание

Оператор **Create Map** присоединяет географические объекты к открытой таблице, после чего их можно видеть в окне карты. Этот оператор не открывает новое окно карты. Чтобы открыть новое окно карты, используйте **оператор Map**.

Не следует применять оператор **Create Map** к таблице, которая уже имеет присоединенные географические объекты; поступая так, вы удаляете все географические объекты из таблицы. Если таблица уже имеет прикрепленные географические объекты, и Вам надо постоянно изменять проекцию карты, используйте оператор **Commit Table As**. Кроме того, если вы хотите временно изменить проекцию, в которой отображается карта, используйте **оператор Set Map** с предложением **CoordSys**. Оператор **Create Map** не работает со связанными таблицами. Чтобы присоединить к связанной таблице географические объекты, используйте **оператор Server Create Map**.

В предложении **BlockSize** можно использовать значение от нуля (0) до 32767. Если этот параметр не задан, то умолчанию используется значение равное 16384. Если значение меньше 512, то оно будет округлено до 512. Как правило, чем больше размер блока, тем выше скорость передачи. Когда выполняется преобразование из формата NativeX (MapInfo Extended) в формат Native (MapInfo), сохраняется размер блока файла NativeX. Пользователи могут задавать новый размер блока при фиксации (выполнении операции commit) из NativeX в формат Native.

CoordSys – предложение о координатной системе; смотрите: **предложение CoordSys clause** и в следующем разделе: **Определение системы координат**.

Определение системы координат

Используйте один из следующих методов для задания системы координат:

- Используйте имя уже открытой таблицы с географическими объектами как значение параметра *from_table* предложения **Using**. В этом случае, используемая система координат будет такой же, как и используемая в *from_table*. Параметр *from_table* должен задавать уже открытую таблицу, причем с географическими объектами, иначе появится сообщение об ошибке.
- Детально задает координатную систему параметра **CoordSys** (заданным по умолчанию). Если опущены оба параметра и **CoordSys**, и **Using**, то в таблице будет использована текущая координатная система заданная MapBasic'ом.

Обратите внимание на то, что **предложение CoordSys** может влиять на точность карты. Для этого **предложение CoordSys** включает в себя предложение **Bounds**, которое устанавливает допустимые значения минимальных и максимальных координат, которые могут быть размещены

на карте. Если предложение **Bounds** пропущено, MapInfo Pro использует стандартные значения, охватывающие всю Землю (в этом случае, координаты имеют точность в миллионную долю градуса, или приблизительно 4 дюйма). Если имеется информация, что карта создана в ограниченном районе, то можно увеличить точность координат карты, задав ее границы. Подробно синтаксис **CoordSys** описан в разделе: [Предложение CoordSys](#).

См. также:

[оператор Commit Table](#), [предложение CoordSys](#), [оператор Create Table](#), [оператор Drop Map](#), [оператор Map](#), [оператор Server Create Map](#), [оператор Set Map](#)

Оператор Create Map3D

Назначение

Создает 3D-карту с определенными параметрами. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Map3D
[ From Window window_id | MapString mapper_creation_string ]
[ Camera [ Pitch angle | Roll angle | Yaw angle | Elevation angle ] |
[ Position ( x, y, z ) | FocalPoint ( x, y, z ) ] |
[ Orientation ( vu_1, vu_2, vu_3, vpn_1, vpn_2, vpn_3,
  clip_near, clip_far )]]
[ Light [ Position ( x, y, z ) | Color lightcolor ] ]
[ Resolution ( res_x, res_y ) ]
[ Scale grid_scale ]
[ Background backgroundcolor ]
[ Units unit_name ]

```

window_id – идентификатор окна карты, содержащего слой поверхности. Если такой слой не найден, появится сообщение об ошибке.

mapper_creation_string - указывает командную строку, которая создает изображение по гриду.

Camera – определяет позицию и ориентацию камеры.

angle – угол, измеряемый в градусах. Горизонтальный угол в диалоге может изменяться от 0 до 360 и вращает карту вокруг центральной точки грида. Вертикальный угол в диалоге изменяется от 0 до 90 и измеряет вращение в вертикальной плоскости от стартовой точки прямо над картой.

Pitch – регулирует поворот камеры вокруг оси X

Roll – регулирует поворот камеры вокруг оси Z

Yaw – регулирует поворот камеры вокруг оси Y

Elevation – определяет поворот камеры (находящейся в точке фокуса) относительно оси X;

Position – регулирует позицию камеры/источника света

FocalPoint – регулирует позицию фокуса камеры/источника света

Orientation – определяет ориентацию камеры: ViewUp (*vu_1, vu_2, vu_3*), ViewPlane Normal (*vpn_1, vpn_2, vpn_3*) и Clipping Range (*clip_near, clip_far*) (фиксированное отображение);

Resolution – количество ячеек поверхности (по X, Y); Эти значения могут быть увеличены до максимально возможного разрешения для данной поверхности. (максимального размера ячейки x, y). Если размер поверхности 200x200, то значения разрешения могут лежать в пределах 200x200. Нельзя увеличивать разрешение регулярной сети, грида, можно изменить только разрешение его изображения.

grid_scale – определяет вертикальный масштаб (в направлении Z-координаты). Значение, большее 1 подчеркивает рельеф, значение, меньшее 1 его огрубляет (в направлении Z-координаты).

backgroundcolor – это цвет, используемый для фона; его задает **функция RGB()**.

unit_name – определяет единицы измерения точек поверхности. Не надо указывать единицы для поверхностей-графиков, например, для трехмерного представления температурного поля или плотности. Эту настройку надо делать к моменту создания грида. Нельзя изменять (настраивать) их с использованием **оператора Set Map3D** или диалога **Свойства**.

Описание

Окно 3D-карты всегда показывается отдельно. Так как оно строится по тем же правилам, что и обычное окно карты, то при изменении или обновлении базовых таблиц или рабочих наборов, окно 3D-карты перерисовывается автоматически. Ошибка создания будет иметь место в том случае, если *window_id* не окно карты или карта не содержит слой поверхности. Если окно карты содержит несколько поверхностей, каждое будет представлено в окне 3D-карты.

3D-карта хранит информацию (строку) об окне карты для генерации текстуры. Эта строка будет также "преобладать" в рабочем наборе, если окно 3D-карты присутствует в данном рабочем наборе. При инициализации прочитываются данные грид-слоя и создаются трехмерные и топологические объекты.

Пример

```
Create Map3D Resolution(75,75)
```

Создает окно 3D-карты для последнего окна карты. Операция не будет выполнена, если окно карты не содержит слоя поверхности. Другой пример:

```
Create Map3D From Window FrontWindow() Resolution(100,100) Scale 2
Background RGB(255,0,0) Units "ft".
```

Создает окно 3D-карты с красным фоном, единицы по оси Z -футы, масштабный коэффициент - 2, и разрешение 100x100.

См. также:

[оператор Set Map3D](#)

Оператор Create Menu

Назначение

Создает новое меню или переопределяет уже существующее. В 64-разрядной версии MapInfo Pro эта команда создаёт новое меню в группе команд **Меню** ленты на вкладке **ИНСТРУМЕНТЫ**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Create Menu newmenuname [ ID menu_id ] [ Context ] As
  menuitem [ ID menu_item_id ] [ HelpMsg help ]
  { Calling handler | As menuname }
  [ , menuitem ... ]
```

Синтаксис 2

```
Create Menu newmenuname As Default
```

Примечание: В 64-разрядной версии MapInfo Pro команда из примера **Синтаксис 2**, приведённого выше, удалит меню из группы меню.

newmenuname – заголовок нового меню или стандартное имя переопределяемого меню, тип String;

menuitem – имя элемента списка меню, тип String;

Слово Context зарезервировано для служебного пользования; запрещено использовать в MapBasic-программах.

menu_id – идентификатор стандартного элемента меню, число типа SmallInt ID от 1 до 15.

menu_item_id – целочисленный идентификатор созданного элемента меню.

help – текст, который будет показываться в строке сообщений при указании на элемент в строке меню.

handler – либо имя процедуры-обработчика, либо код стандартной команды MapInfo, либо строка специального синтаксиса для управления событиями в системе меню через механизмы

OLE или DDE; см. [Режимы предложения Calling](#). Если задаётся код команды меню MapInfo Pro (такой как M_WINDOW_STATISTICS), то аргумент *menuitem* должен начинаться с восклицательного знака и содержать внутри символ каретки (^), чтобы удовлетворять синтаксису команд типа Показать/Скрыть.

menuname – заголовок меню, которое будет включено в список как подменю.

Описание

Если параметр *newmenuname* является именем одного из существующих меню (например, **File**), то оператор переопределяет это меню. Если параметр *newmenuname* имеет какое-то уникальное значение, то оператор **Create Menu** создает новое меню. Список стандартных имен меню MapInfo Pro приводится в разделе [Оператор Alter Menu](#).

Оператор **Create Menu** не показывает новое или переопределяемое меню. Для вывода меню на экран используются [оператор Alter Menu Bar](#) или [оператор Create Menu Bar](#). Однако, если оператор **Create Menu** переопределяет меню, которое существует в строке меню, то изменения меню отображаются немедленно.

Примечание: MapInfo Pro может обслуживать одновременно не более 96 определителей меню, включая меню, автоматически определённые MapInfo Pro (**Файл** и т.п.). Этот предел не зависит от количества пунктов меню, отображаемых одновременно в строке меню.

Параметр *menuitem* определяет имя элемента меню. В нём могут использоваться специальные управляющие символы, определяющие состояние данного элемента меню (например, доступность команды в меню). Смотрите таблицу ниже.

Следующие знаки являются зарезервированными и выполняют специальную роль: косая черта (slash – /), обратная косая черта (back slash – \) и знак меньше (less than – <). Если требуется использовать эти знаки в тексте меню или строке подсказки, то необходимо поставить обратную косую черту в строке *menuitem* или в строке *help*. Например, следующая инструкция создаёт пункт меню "Клиент\Сервер":

```
Create Menu "Data" As
  "Client\Server" Calling cs_proc
```

Если текст параметра *menuitem* начинается с символа @, то пользовательское меню делится на две колонки. Элемент, начинающийся с @, помещается первым во вторую колонку.

Полезно учитывать при использовании этой команды в 64-разрядной версии MapInfo Pro:

1. Если меню с таким идентификатором уже существует на вкладке ИНСТРУМЕНТЫ, то все элементы управления инструмента-дополнения из этого меню будут удалены и будет создано меню с новой структурой.
2. Эта команда не оказывает влияния на контекстные меню.
3. Если одно меню добавлено в двух разных местах на вкладке с традиционными программами, только структура последнего меню будет изменена на новое определение.

4. Показ/скрытие текста с помощью переключателей элементов меню не поддерживается.
5. Если нескольким элементам меню присвоена одна и та же комбинация клавиш, то оба элемента управления не будут вызываться с помощью этой комбинации клавиш.
6. Имена меню должны быть корректными именами Windows Presentation Forms (WPF) (например, без специальных символов, не поддерживаются цифры в начале имён меню — такие символы отображаются символом подчёркивания).
7. Название меню, уже добавленного в ленту, не будет меняться с помощью оператора **Create menu**. Необходимо удалять меню и повторно добавлять его.
8. Если одно и то же меню добавлено в двух разных местах, запоминается только последнее меню.
9. Меню с одним и тем же именем, что и предопределённые меню MapInfo Pro, будут вести себя неправильно до тех пор, пока при вызове оператора **Create menu** не будет указан нужный идентификатор.

Обработчики для новых команд меню

Большинство элементов меню определяются с помощью предложения **Calling handler**, где *handler* либо имя процедуры на MapBasic, либо код стандартной команды MapInfo Pro (например, код `M_FILE_SAVE` соответствует команде **Сохранить таблицу**). Если для элемента определен обработчик и если пользователь выберет в меню этот элемент, то MapBasic автоматически вызовет процедуру-обработчик. Если для элемента меню определен код, такой как `M_FILE_SAVE`, то ваша программа должна в начале текста содержать оператор `Include "MENU.DEF"`, подключающий файл стандартных определений для кодов команд.

С помощью необязательного предложения **ID** вы можете задать элементу меню уникальный идентификатор. Можно связать один и тот же обработчик с разными элементами меню. В этом случае идентификатор может оказаться полезным для определения того, какой именно командой была вызвана эта процедура. Определить идентификатор элемента меню из процедуры-обработчика, которую он вызвал, можно функцией: **CommandInfo** (`((CMD_INFO_MENUITEM))`). Идентификатором элемента меню пользуется также **оператор Alter Menu Item**. Если за параметром *handler* не задано ни команды, ни подменю, ни вызова процедуры *menuname*, то соответствующий элемент меню будет неактивным. Такие элементы в меню выполняют косметическую роль (например, горизонтальная линия, разделяющая список меню на части).

Создание иерархически-подчиненного меню

Чтобы включить иерархическое меню в качестве нового элемента, задайте предложение **As** вместо **Calling**. Предложение **As** должно задавать один из существующих элементов меню. В следующем примере создается новое меню из одной "простой" пользовательской и одной команды и стандартной команды с иерархической структурой.

```
Create Menu "Special" As
  "Configure" Calling config_sub_proc,
  "Objects" As "Objects"
```


Если добавляется иерархическое меню, то оно замещает элемент меню *menuItem*.

Свойства элементов меню

Команды в меню могут быть активны и отключены; последние показываются серым цветом. Можно добавлять галочку к имени элемента меню. Элемент меню с галочкой может быть активирован или деактивирован.

Следующие коды должны предшествовать имени элемента меню, поведение которого мы желаем сделать каким-то особенным:

Код	Эффект
(	Элемент меню недоступен для выбора. Пример: (Close
(-	Горизонтальная разделительная линия; такой элемент не может иметь обработчика. Пример: (-
(\$	Код для помещения в меню Файл (или File) имен четырех последних открывавшихся файлов. Его можно применять в системе меню один раз и его нельзя использовать в быстрых меню. Чтобы удалить список последних открывавшихся файлов из меню Файл (или File), либо удалите этот код из файла MAPINFOW.MNU, либо переопределите меню Файл оператором Create Menu .
(>	Код для представления в меню списка открытых окон. Его можно применять в системе меню один раз.
!	Пункт меню можно выделять, но изначально он не отмечен. Пример: !Confirm Deletion
! ... ^ ...	Если в тексте появляется знак ^, то при выборе элемента он заменяется на другой текст, например (Show... vs. Hide...^)). Текст перед знаком "^" показывается, если элемент уже был выбран. Пример: !Hide Status Bar^Show Status Bar
!+	Элемент меню при выборе может снабжаться галочкой и на момент создания галочка присутствует. Пример: !+Confirm Deletions

Задание комбинаций клавиш

Элементам меню могут быть назначены клавишные сокращения (акселераторы), то есть пользователь может вызвать команду с клавиатуры без использования мыши. Клавишные сокращения бывают двух типов.

Клавишные сокращения первого типа имеют и имена меню, и имена элементов меню. Например, для того, чтобы открыть окно карты, в рабочем окне MapInfo Pro пользователь нажимает сначала клавишное сокращение **ALT+O**, которым открывается список команд меню **Окно**, и только потом клавишу **K** (или **ALT+K**) для запуска команды **Карта**. Для того, чтобы определить буквы для клавишного сокращения в меню и элементах меню, используйте знак амперсанда (&) в текстах параметров `newmenuitem` и `menuitem` (например, текст "&Map" в параметре `menuitem` оператора **Create Menu**). Амперсанд должен располагаться перед буквой, которая будет использоваться в клавишном сокращении.

Другой тип комбинаций клавиш позволяет вызвать команду немедленно, не открывая меню. Если элемент меню имеет комбинацию клавиш, например, **Alt+F5**, то активировать меню можно, нажав **Alt+F5**. Для того чтобы создать такую комбинацию клавиш быстрого вызова, используйте следующие комбинации клавиш.

Примечание: Управляющие коды, использованные для задания сочетаний клавиш, приведены в следующей таблице.

Управляющий код	Эффект
<code>/W {letter %number}</code>	Задаёт комбинацию клавиш для Windows; команда активизируется клавишей, заданной буквой, или клавишей, заданной ее числовым кодом. Примеры: Zap /WZ или Zap /W%120.
<code>/W# {letter %number}</code>	Задаёт комбинацию клавиш для Windows; команда активизируется клавишей, заданной буквой, или клавишей, заданной ее числовым кодом, с нажатой клавишей SHIFT . Примеры: Zap /W#Z или Zap /W#%120.
<code>/W@ {letter %number}</code>	Задаёт комбинацию клавиш для Windows; команда активизируется клавишей, заданной буквой, или клавишей, заданной ее числовым кодом, с нажатой клавишей ALT . Примеры: Zap /W@Z или Zap /W@%120.
<code>/W^ {letter %number}</code>	Задаёт комбинацию клавиш для Windows; команда активизируется клавишей, заданной буквой, или клавишей, заданной ее числовым кодом, с нажатой клавишей CTRL . Примеры: Zap /W^Z или Zap /W^%120.

В Windows для использования клавиш Fn используется возможность задавать комбинации клавиш символом %, за которым следует численный код, номер клавиши. Так, для клавиши **F1** номером является 112, для **F2** – 113 и так далее.

Примечание: Заметим, что оператор **Create Menu Bar As Default** убирает и переопределяет все пользовательские меню, заданные оператором **Create Menu**. Чтобы выборочно отменить одно из созданных пользователем меню, можно воспользоваться оператором **Create Menu menuname As Default**.

Аналогично, чтобы выборочно отменить одно из измененных стандартных меню, можно воспользоваться тем же оператором **Create Menu menuname As Default**.

Режимы предложения *Calling*

Предложение **Calling** задает, что должно случиться, если пользователь в меню выполнит команду, построенную приложением. В таблице приведены возможные примеры использования этого предложения:

Примеры	Описание
<code>Calling M_FILE_NEW</code>	Если за словом Calling идет целочисленный код из файла MENU.DEF, MapInfo Pro запускает на выполнение соответствующую стандартную команду MapInfo Pro (например, Файл > Новая).
<code>Calling my_procedure</code>	Если задано имя процедуры, MapInfo Pro передает управление этой процедуре.
<code>Calling OLE "methodname"</code>	Только для Windows. MapInfo Pro обрабатывает события, обращаясь с именем метода к объекту OLE Automation, установленному SetCallback-методом.
<code>Calling DDE "server","topic"</code>	Только для Windows. MapInfo Pro обрабатывает события, присоединяясь через DDE к паре "сервер тема" и посылая выполняемое сообщение на DDE-сервер.

В последних двух случаях строка послания для OLE или DDE сервера должна начинаться с трех символов "MI:" для того, чтобы сервер смог определить, что сообщение пришло от MapInfo Pro. Остальная часть этой строки состоит из разделенного запятыми списка значений, которые возвращает [функция CommandInfo\(\)](#). Полное описание синтаксиса этой строки см. в *Руководстве пользователя MapBasic*.

Примеры

В следующем примере оператор **Create Menu** создает новое меню и добавляет его в строку меню MapInfo Pro. Сначала удаляется меню Окно (ID 6) и меню Справка (ID 7), после чего добавляется новое меню, а затем восстанавливаются меню Окно и меню Справка. Так мы добиваемся того, что заголовки "Окно" и "Справка" всегда будут последними в строке меню.

```
Declare Sub Main
Declare Sub addsub
Declare Sub editsub
Declare Sub delsub
Sub Main
  Create Menu "DataEntry" As
```

```

"Add" Calling addsub,
"Edit" Calling editsub,
"Delete" Calling delsub

Alter Menu Bar Remove ID 6, ID 7
Alter Menu Bar Add "DataEntry", ID 6, ID 7
End Sub

```

В этом фрагменте создается сокращенная версия меню Файл (или File). Управляющий символ "(" делает команды **Заккрыть**, **Сохранить** и **Печать** недоступными для выбора. Командам "Открыть" и "Сохранить" заданы сочетания клавиш Windows (**Ctrl+O** и **Ctrl+S**). Обратите внимание, что в тексте определения элемента меню **Открыть** и **Сохранить** используется знак табуляции – **Chr\$(9)** – для отделения сочетания клавиш вправо от текста.

```

Include "MENU.DEF"

Create Menu "File" As
  "New" Calling M_FILE_NEW,
  "Open" +Chr$(9)+"Ctrl+O/W^O" Calling M_FILE_OPEN,
  "(-",
  "(Close" Calling M_FILE_CLOSE,
  "(Save" +Chr$(9)+"Ctrl+S /W^S" Calling M_FILE_SAVE,
  "(-",
  "(Print" Calling M_FILE_PRINT,
  "(-",
  "Exit" Calling M_FILE_EXIT

```

Если не хотите, чтобы пользователь мог открывать быстрые меню, используйте оператор **Create Menu** и переопределите эти меню одним только разделителем: " (-". В следующем примере быстрое меню окна карты делается недоступным.

```
Create Menu "MapperShortcut" As " (-"
```

См. также:

[оператор Alter Menu Item](#), [оператор Create Menu Bar](#)

Оператор Create Menu Bar

Назначение

Перестраивает строку заголовков меню, используя стандартные и ранее определенные меню. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

Create Menu Bar As

```
{ menu_name | ID menu_number }  
[ , { menu_name | ID menu_number } ... ]
```

Примечание: В 64-разрядной версии MapInfo Pro эта команда создаёт новую группу на вкладке **ИНСТРУМЕНТЫ**, где каждое меню представляет собой раскрывающийся список. Также, если подобное меню уже существует в группе меню, то эта команда заменит существующую команду новой.

Синтаксис 2

Create Menu Bar As Default

Примечание: В 64-разрядной версии MapInfo Pro команда из примера **Синтаксис 2**, приведённого выше, удалит всю группу меню.

Примечание: Меню WinSpecific больше не поддерживается. Необходимо присвоить индивидуальные имена меню, для того чтобы добавить их в меню группы.

menu_name – заголовок стандартного для MapInfo Pro меню или заголовок специально определенного меню, для создания которого ранее выполняется **оператор Create Menu**.

menu_number – номер стандартного меню (например, 1 для меню Файл).

Описание

Оператор **Create Menu Bar** говорит MapInfo Pro, какие меню должны быть помещены в строку меню и в каком порядке. Если оператор задает не полный список стандартных меню, то результатом будет строка меню с сокращенным списком MapInfo Pro. Если в список меню включено одно или несколько специально определенных имен меню (для их создания используется **оператор Create Menu**), оператор **Create Menu Bar** создаст строку меню с расширенным списком.

Меню, как стандартное меню, так и специально определенное, может задаваться именем (например, "Файл"). Каждое стандартное меню также имеет номер (идентификатор), который может использоваться при задании. Например, меню Файл (или File) имеет идентификатор 1.

Список стандартных имен меню MapInfo Pro см. в разделе: **Оператор Alter Menu Item**.

После того как система меню была изменена, можно оператором:

```
Create Menu Bar As Default
```

вернуть ее к стандартному виду. При этом оператор **Create Menu Bar As Default** удаляет все изменения в строке меню, включая и те, которые были созданы другими приложениями.

Поэтому следует быть внимательным, употребляя этот оператор. Используйте команду **Create Menu Bar As Default** осторожно, чтобы случайно не удалить из меню пользовательские программы MapBasic.

Примеры

В следующем примере меню сокращено и включает только пункты меню Файл, Правка и Запрос.

```
Create Menu Bar As
  "File", "Edit", "Query"
```

В стандартной строке меню такие меню, как карта и список, не показываются, если окна карты или списка соответственно не являются активными. Следуя этой логике, MapInfo Pro показывает меню список (Browse), если активным окном является список. Следующий оператор помещает эти меню в строку меню так, что они не зависят от того, есть ли на экране окна карты и списка. Но при этом, если пользователь откроет меню карта, когда на экране нет активного окна карты, то он увидит, что все команды этого меню недоступны (показаны серым шрифтом). Аналогично работает меню Список.

```
Create Menu Bar As
  "File", "Edit", "Query", "Map", "Browse"
```

В следующем фрагменте создается пользовательское меню, содержащее три пункта: **Файл, Правка и Данные**

```
Declare Sub AddSub
Declare Sub EditSub
Declare Sub DelSub

Create Menu "DataEntry" As
  "Add" calling AddSub,
  "Edit" calling EditSub,
  "Delete" calling DelSub

Create Menu Bar As
  "File", "Edit", "DataEntry"
```

См. также:

[оператор Alter Menu Bar](#), [оператор Create Menu](#), [оператор Menu Bar](#)

Оператор Create MultiPoint

Назначение

Объединяет множество точек в единый объект – группу точек. Все точки обозначаются одним и тем же символом. Объект Multipoint отображается в окне списка как одна запись. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Multipoint
[ Into { Window window_id | Variable var_name } ]
[ num_points ]
( x1, y1 ) ( x2, y2 ) [ ... ]
[ Symbol... ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

num_points – номер точки в объекте Multipoint;

x y – координаты точки.

Предложение **Symbol** определяет стиль символа.

Примечание: Для всех точек, содержащихся в объекте, используется один символ.

В настоящее время MapInfo Pro использует четыре варианта синтаксиса для определения символа, используемого для точек.

Синтаксис (вариант 2 - версия для символов MapInfo Pro 3.0):

```
Symbol ( shape, color, size )
```

shape – целое число, 31 или больше, задающее, какой из символов стандартного набора MapInfo Pro следует использовать. Символы MapInfo 3.0 были представлены в MapInfo для версии Windows 3.0 и поддерживаются всеми последующими версиями MapInfo Pro. Для создания невидимого символа используйте значение 31. Стандартный набор условных знаков включает символы от 31 до 67, но пользователь может отредактировать любой символ с помощью программы Символ.

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

Синтаксис 2 (для шрифтов TrueType)

```
Symbol ( shape, color, size, fontname, fontstyle, rotation )
```

shape – целое, имеющее значение 31 или больше, определяющее, какой используется символ из шрифтов TrueType. Для создания невидимого символа используйте значение 31.

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

fontname – строка с именем шрифта TrueType (например, "WingDings");

fontstyle – целое, код, задающий стилизацию, например, курсивный текст;

rotation – вещественное число, угол поворота в градусах.

Синтаксис 3 (для пользовательских символов Custom Bitmap File)

```
Symbol ( filename, color, size, customstyle )
```

filename – строка до 31 символа длиной с именем растрового файла. Файл должен находиться в папке CUSTSYMB (если только [оператор Reload Symbols](#) не применен для указания другой папки).

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

customstyle – целочисленный код типа Integer, управляющий цветом и фоном символа. Смотрите таблицу ниже.

Синтаксис 5

```
Symbol symbol_expr
```

symbol_expr – выражение для Symbol, которое может быть или именем переменной символа или вызовом функции, которая возвращает значение символа, например, [функция MakeSymbol\(\)](#).

Пример

```
Create Multipoint 7 (0,0) (1,1) (2,2) (3,4) (-1,1) (3,-2) (4,3)
```

См. также:

[оператор Objects Combine](#), [оператор Set Combine Version](#)

Оператор Create Object

Назначение

Создает один или несколько объектов типа "Область", используя географические операции Buffer, Merge, Intersect, Union, Voronoi или Isogram. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Object As { Buffer | Isogram | Union | Intersect | Merge |
ConvexHull | Voronoi }
From fromtable
[ Into { Table intotable | Variable varname } ]
[ Data column = expression [ , column = expression... ] ]
[ Group By { column | RowID } ]
```

fromtable – имя открытой таблицы, содержащей один или более графических объектов.

intotable – имя открытой таблицы, в которую будут помещаться созданные объекты.

varname – имя новой объектной переменной;

column - имя колонки в таблице.

expression – выражение для расчета значений *column*;

Описание

Оператор **Create Object** создает одну или более областей на базе уже существующих объектов, используя географические операции (Buffer, Merge, Intersect, Union, ConvexHull, Voronoi или Isogram).

Предложение **Into** задает, куда будет помещен объект, полученный в результате. Для сохранения результатов в таблице, укажите **Into Table**. Для сохранения результатов в переменной Object, укажите **Into Variable**. Если пропущено предложение **Into**, результаты сохранятся в исходной таблице.

Примечание: Если используется предложение **Group By** чтобы объединить данные, то результаты надо сохранять в таблице, а не в переменной.

Ключевое слово, которое следует за ключевым словом **As**, определяет, какой тип объектов будет создан. **Buffer** и **Isogram** обсуждаются в разделах: [Создание буферных объектов](#) и [Создание объектов-изограмм](#).

Union

В режиме **Union** комбинирование удаляет любые области перекрытия. Если осуществляется операция объединения (Union) двух перекрывающихся регионов (каждый из которых содержит один полигон), то результат будет в виде объекта-региона, содержащего один полигон.

Операции объединения (Union) и слияния (Merge) похожи, но их поведение сильно различается в том случае, если объекты полностью содержат в себе другие объекты. В этом случае, операция слияния (Merge) удаляет область малого объекта из большого объекта, оставляя дыру там, где был малый объект. Операция объединения (Union) не будет удалять область меньшего объекта.

Оператор **Create Objects As Union** похож на **оператор Objects Combine** тем, что Оператор **Objects Combine** удаляет исходные объекты и вставляет новый комбинированный объект. **Create Objects As Union** только добавляет объекты в группу, но не удаляет исходные объекты. Комбинирование с использованием изменяемого объекта и различных таблиц возможно только, применяя **оператор Objects Combine**. Оператор **Objects Combine**, использующий предложение **Column**, допустим только при использовании **Create Objects As Union** вместе с предложением **Group By**.

Если оператор **Create Object As Union** не включает в себя предложение **Group By**, MapInfo Pro создает один комбинированный объект для всех объектов в таблице. Если этот оператор включает в себя предложение **Group By**, то должно быть задано имя колонки в таблице, что позволит MapInfo Pro сгруппировать исходные объекты в соответствии с содержанием колонки и создать комбинированный объект для каждой группы объектов.

Если задать предложение **Group By**, MapInfo Pro сгруппирует все записи, имеющие одинаковые значения, и осуществит операцию (например, слияния) в группу.

Если задать предложение **Data**, MapInfo Pro осуществит объединение данных. Например, если осуществляется слияние или объединение, может возникнуть необходимость использовать предложение **Data**, чтобы присвоить значение данным, выполняя функции объединения **Sum()** или **Avg()**.

Intersect

В режиме **Intersect** создается объект, представляющий собой пересечение других объектов (например, если два региона пересекаются, то пересечение - это область, покрывающая сразу эти два объекта).

Merge

В режиме **Merge** создается объект, представляющий собой комбинированную площадь из объектов-источников. Операция Merge создает результирующий объект, который содержит все полигоны, относящиеся к исходным объектам. Если исходные объекты перекрываются, то операция слияния (merge) не удаляет перекрытия. Таким образом, если объединяются два перекрывающихся региона (каждый из которых содержит один полигон), то окончательный результат будет в виде одного региона, состоящего из двух перекрывающихся полигонов. Чтобы избежать этого, можно использовать объединение в режиме **Union**.

Convex Hull

В режиме **ConvexHull** оператор будет создавать полигон, представляющий оконтуривающий объект вокруг набора точек. Такой полигон можно получить, если натянуть резинку, охватывающую все узлы. Он составлен из минимального количества точек (т.е. точки входного объекта лежат на границах или внутри полигона). Полигон получается выпуклый – то есть все внешние углы созданного полигона-контура больше, чем 180 градусов.

Точки, используемые для оконтуривающего региона, могут быть любыми узлами объектов типа "Область", "Полилиния" или "Точка" таблицы *fromtable*. Если оператор **Create Object As ConvexHull** не включает в себя предложение **Group By**, MapInfo Pro создаст один оконтуривающий полигон. Если оператор включает в себя предложение **Group By**, указывающее имя колонки в таблице, то MapInfo Pro сгруппирует исходные объекты исходя из содержания указанной колонки, затем создаст оконтуривающий полигон для каждой группы объектов. Если оператор включает в себя предложение **Group By RowID**, MapInfo Pro создаст один оконтуривающий полигон для каждого объекта в исходной таблице.

Voronoi

Предложение **Voronoi** используется для создания регионов, представляющих полигоны Вороного для исходных точек. Значения данных их исходных точек могут быть присвоены результирующему полигону, для этого в предложении надо указать источник таких точек.

Пример

Следующий пример показывает слияние объектов-регионов из таблицы PARCELS и хранит результирующие регионы в таблице ZONES. Так как оператор **Create Object** включает в себя предложение **Group By**, MapBasic группирует регионы таблицы PARCELS, а затем произведет слияние для каждой группы. Таким образом, таблица ZONES будет иметь по одному объекту-региону для каждой группы объектов в таблице PARCELS. Каждая группа будет состоять из тех частей, которые имеют одинаковые значения в колонке *zone_id*.

Оператор **Create Object** накапливает в колонке *parcelcount* число однородных объектов таблицы ZONES и показывает, сколько элементов сливается вместе, образуя единую зону. Колонка *zonevalue* в таблице ZONES будет показывать сумму элементов, которые включает в себя эта зона.

```
Open Table "PARCELS"
Open Table "ZONES"
Create Object As Merge
  From PARCELS Into Table ZONES Data
  parcelcount=Count(*), zonevalue=Sum(parcelvalue)
  Group By zone_id
```

Следующий пример представляет, как создается оконтуривающий полигон, состоящий из многих объектов, созданный оператором **Create Object As**.

```
Create Object As ConvexHull from state_caps into Table dump_table
```

См. также:

функция **OverlayNodes()**, оператор **Set Buffer Version**, оператор **Set Combine Version**

Оператор **Create Object As Buffer**

Назначение

Создаёт полигон вокруг точек, линий или других контуров. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Object As Buffer
From fromtable
[ Into { Table intotable | Variable varname } ]
[ Width bufferwidth [ Units unitname ] ]
[ Type { Spherical | Cartesian } ] ]
[ Resolution smoothness ]
[ Data column = expression [ , column = expression... ] ]
[ Group By { column | RowID } ]
[ Concurrency { All | Aggressive | Intermediate | Moderate | None } ]
```

bufferwidth – число, определяющее смещение при выполнении операции построения буферной зоны; если число отрицательное, а исходный объект – замкнутый объект, то полученная буферная зона будет меньше исходного объекта. Если задано отрицательное значение *width*, а объект является линейным (линией, полилинией или дугой) или же точкой, то используется абсолютные значение *width* и создается буфер больший чем объект.

unitname – наименование единиц расстояния (например, “km” для километров).

smoothness – целое от 2 до 100, указывающее число сегментов для каждой зоны в операции буферизации;

Описание

Если оператор **Create Object** работает в режиме **Buffer**, то он может включать в себя предложения **Width** и **Resolution**. Предложение **Width** определяет ширину буферной зоны. Добавочное предложение **Units** позволяет указать имя единиц измерения расстояния (например, “km” для километров) относящихся к предложению **Width**. Если предложение **Width** не включает в себя предложение **Units**, то ширина буферной зоны будет интерпретироваться в текущих единицах измерения MapBasic. По умолчанию, MapBasic использует мили для измерения расстояний; чтобы изменить стандартные единицы измерения, для замены которых выполняется **оператор Set Distance Units**.

Type позволяет указать тип алгоритма вычисления расстояний при создании буферной зоны. Он может быть либо сферическим (**Spherical**), либо декартовым (**Cartesian**). Если координатная система таблицы *intotable* – План-Схема, то вычисления производятся декартовыми методами

вне зависимости от настроек оператора; если же координатная система *intotable* – это Долгота/Широта, то вычисления производятся сферическими методами вне зависимости от настроек оператора.

Предложение **Type** позволяет определять способ вычислений расстояний во время создания буфера. Если используется тип **Spherical**, то все вычисления производятся с учетом шарообразности поверхности Земли, то есть в координатах Долгота/Широта. Если используется **Cartesian** (декартовый) тип вычислений, то подразумевается, что вычисления производятся на карте, на которую нанесены объекты на плоскости (т.е. на плане). Если предложение **Width** не включает предложения **Type**, то по умолчанию используется тип **Spherical**. Если данные представлены в проекции Долгота/Широта, то используются сферические вычисления независимо от установок в предложении **Type**. Если данные в проекции план-схема, то, независимо от установки параметра **Type**, будут использованы декартовы вычисления.

Параметр **Resolution** позволяет определить число сегментов, из которых состоит каждая окружность в составе буферной зоны. Стандартное значение *smoothness* равно 12; это означает, что при рисовании буфера из простой окружности будет использовано 12 сегментов. Если задать большую величину *smoothness*, то получится более сглаженная буферная зона. Обратите внимание, что чем больше значение *smoothness*, тем дольше работает оператор **Create Object** и тем больше дискового пространства занимает результат.

Дополнительное, необязательное предложение **Concurrency** позволяет распределить вычисления между несколькими ядрами процессора, чтобы повысить производительность. Когда значение параметра **Concurrency** установлено равным *none*, а в компьютере используется процессор с несколькими ядрами, то остальные ядра не используются. Это предложение позволяет задавать уровень распараллеливания, используемый при выполнении этой операции на карте. Параметр **Concurrency** может принимать одно из следующих значений:

- **All**: полная параллельная многопоточность. Все процессоры системы используются для выполнения операций. Это стандартный режим MapInfo Pro.
- **Aggressive**: многопоточность, распараллеливание используется агрессивно, при выполнении операций используется 75% процессоров системы.
- **Intermediate**: промежуточный вариант использования многопоточности, распараллеливания, при выполнении операций используется 50% процессоров системы.
- **Moderate**: многопоточность, распараллеливание используется умеренно, при выполнении операций используется 25% процессоров системы.
- **None**: многопоточность, распараллеливание не используется. Операцию выполняет единственный процессор. Эта опция предопределяет наименьшую скорость обработки.

Например:

```
Create Object As Buffer
From zipcodes
Width 1
Units "mi"
Type Spherical
Resolution 12
```

```
Into Table zipcodes
Group by Rowid
Concurrency Aggressive
Data...
```

В дополнение к пяти возможным вариантам использования многопоточности, также можно указать количество задействуемых ядер, например, восемь (8). Если компьютер имеет меньшее, чем указанное количество ядер, MapBasic по умолчанию использует все доступные ядра на этой машине. Нулевое значение (0), отрицательное число или некорректное текстовое значение, которое не может быть интерпретировано, как одно из пяти допустимых вариантов многопоточности, приведёт к ошибке.

Следующая команда вызывает сообщение об ошибке:

```
Create Object As Buffer
From zipcodes
Width 1
Units "mi"
Type Spherical
Resolution 12
Into Table zipcodes
Group by Rowid
Concurrency 0
```

```
Create Object As Buffer
From zipcodes
Width 1
Units "mi"
Type Spherical
Resolution 12
Into Table zipcodes
Group by Rowid
Concurrency -5
```

Примечание: Многопоточность используется только тогда, когда задан вариант **Один буфер для каждого объекта** в диалоге **Буферные объекты**. Если этот вариант обработки буферных объектов не выбран, и в команде оператора присутствует предложение **Concurrency**, то этот параметр пропускается и сообщение об ошибке не появляется.

Если оператор **Create Object As Buffer** не включает в себя предложение **Group By**, MapInfo Pro создаст один буферный регион. Если оператор включает в себя предложение **Group By**, которое является именем колонки в таблице, MapInfo Pro сгруппирует исходные объекты в соответствии с содержанием колонки, затем создаст по одному буферному региону для каждой группы объектов. Если оператор включает в себя предложение **Group By RowID**, MapInfo Pro создаст по одному буферному региону для каждого объекта в исходной таблице.

Пример

Следующий пример создает объект-регион, представляющий зону шириной в четверть мили вокруг выбранных объектов. Объект-буфер хранится в объектной переменной `corridor`. Применяв далее **оператор Update** или **оператор Insert**, можно скопировать объект в таблицу.

```
Dim corridor As Object
Create Object As Buffer
  From Selection
  Into Variable corridor
  Width 0.25 Units "mi"
  Resolution 60
```

См. также:

оператор Set Buffer Version

Create Object As Isogram

Синтаксис

```
Create Object As Isogram
  From fromtable
  [ Into { Table intotable } ]
  [ Data column = expression [ , column = expression... ] ]
  Connection connection_handle
  [ Distance dist1 [[ Brush ... ] [ Pen ... ]]
    [ , dist2 [ Brush ... ] [ Pen ... ]]
    [ , distN [ Brush ... ] [ Pen ... ] [ ,... ]
  Units dist_unit ]
  [ Time time1 [[ Brush ... ] [ Pen ... ]]
    [ , time2 [ Brush ... ] [ Pen ... ]]
    [ , timeN [ Brush ... ] [ Pen ... ] [ ,... ]
  Units time_unit ]
```

connection_handle— численное выражение, уникальный номер соединения, которое возвращает **оператор Open Connection**.

dist1, dist2, distN — численные значения представляющие расстояние для зон транспортной доступности, рассчитываемых в **dist_unit**;

Brush — корректное **предложение Brush**, задающее штриховку.

Pen — корректное **предложение Pen**, задающее стиль линии.

dist_unit — наименование единиц расстояния (например, "km" для километров). См. полный список возможных значений в разделе: **Оператор Set Distance Units**.

time1, time2, timeN - численные значения, представляющие время для зон транспортной доступности, рассчитываемых в **time_units**.

time_unit - строка, представляющая единицы времени. Допустимые значения: "hr", "min", или "sec".

Описание

Если оператор **Create Object** используется для создания изолиний, необходимо передать *connection_handle* соответствующий открытому соединению, который создал **оператор Open Connection**. Необходимо указать предложение **Distance** или **Time** чтобы указать требуемый размер зоны доступности. Предложение **Distance** может включать одно или несколько выражений для расстояния (зон), с возможностью указать стиль оформления для каждой зоны. Если не задано **предложение Brush** или **предложение Pen**, используются текущие установки штриховки или линии. Зон транспортной доступности по времени можно создавать много, но всегда должно присутствовать предложение **Units** с единицами времени, которые будут использоваться в вычислениях.

Указав предложение **Time**, можно создать зоны транспортной доступности основываясь на времени движения, и указать для каждой зоны стиль графического оформления с использованием **предложения Brush** и/или **предложения Pen**. Если не задано **предложение Brush** или **предложение Pen**, используются текущие установки штриховки или линии. Зон транспортной доступности по времени можно создавать много, но всегда должно присутствовать предложение **Units** с единицами времени, которые будут использоваться в вычислениях. Максимальное допустимое значение 50. Каждое значение создаёт отдельную зону, которая может отображать транспортную доступность либо по времени, либо по расстоянию. Чем больше значение, тем больше время потребуется на операцию создания зон. На скорость обработки влияют много факторов, но в целом **оператор Set Connection Isogram** с параметром **MajorRoadsOnly** работает значительно быстрее, чем при использовании всей транспортной сети. MapBasic допускает расстояние 35 миль для параметра **MajorRoadsOnly Off** и 280 миль для **MajorRoadsOnly On**. Аналогично, максимальное время – 0.5 часа для параметра **MajorRoadsOnly Off** и 4 часов для параметра **MajorRoadsOnly On**.

См. также:

функция Buffer(), **функция ConvexHull()**, **оператор Objects Combine**, **оператор Objects Erase**, **оператор Objects Intersect**, **оператор Open Connection**

Оператор Create Pline

Назначение

Создает объект типа "полилиния". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Pline
[ Into { Window window_id | Variable var_name } ]
[ Multiple num_sections ]
num_points ( x1, y1 ) ( x2, y2 ) [ ... ]
[ Pen... ]
[ Smooth ] [ Priority n ] [ Name framename ]

```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

num_points – определяет число узлов для полилинии;

num_sections – указывает, сколько сегментов содержит многосегментная полилиния.

Каждая пара *x*, *y* задает координаты для одного узла полилинии.

Слово **Pen** начинает стандартное предложение для назначения стиля линии.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окне **Отчёт**.

framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Результатом действия оператора **Create Pline** является новый объект типа "Полилиния". Если требуется создать полилинию, но неизвестно, сколько в ней будет узлов, сначала оператором **Create Pline** создайте объект вообще без узлов, а затем с помощью **оператора Alter Object** добавьте к нему узлы.

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно **Window**, объект помещается на заданное место в окне (например, в изменяемый слой). Если предложения **Into** вообще нет в операторе, MapInfo Pro попытается создать прямоугольник в самом верхнем окне; если это невозможно, то объект не будет создан.

Параметры *x* и *y* задаются в той координатной системе, которая была объявлена MapBasic ранее (по умолчанию используется Долгота/Широта). Если объект создается для окна отчета, параметры *x* и *y* интерпретируются как координаты на листе в объявленных ранее "бумажных" единицах измерения листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы в качестве "бумажных" единиц расстояния. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Layout**. Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Необязательный параметр **Pen** начинает предложение, задающее стиль линии. Если не задан параметр **Pen** оператор **Create Pline** использует текущий стиль линии (то есть, тот, который задан в диалоге MapInfo Pro **НастройкиСтиль линии**). **Smooth** – сгладит линию так, что она будет выглядеть без узлов.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение Z-Order новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

Однофрагментный объект типа "Полилиния" может содержать до 134,217,724 узлов. Полилиния, составленная из нескольких фрагментов, может иметь до 24,403,223 сегментов.

См. также:

[оператор Alter Object](#), [оператор Insert](#), [предложение Pen](#), [оператор Set CoordSys](#), [оператор Update](#)

Функция CreatePoint()

Назначение

Возвращает объект "Точка". Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CreatePoint( x, y )
```

x – X-координата точки (или широта), вещественное число;

y – Y-координата точки (или долгота), вещественное число.

Возвращаемая величина

Объект

Описание

Функция **CreatePoint()** возвращает графический объект типа "Точка".

Параметры *X* и *Y* задают координаты центра окружности в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется [оператор Set CoordSys](#). При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты.

Для рисования точечного объекта используется текущий стиль символа. Точечный объект будет создаваться в соответствии с установкой стиля символа для точечных объектов, для чего нужно выполнить **оператор Set Style** до вызова функции `CreatePoint( )`. Также можно вместо создания объекта функцией **CreatePoint()** использовать **оператор Create Point**, задав в нем предложение **Symbol** для определения стиля символа точки.

Графический объект, созданный функцией **CreatePoint()**, может быть присвоен переменной объекта, которая определяет значение в существующей строке таблицы (с помощью **оператора Update**) или вновь созданной (с помощью **оператора Insert**).

Примечание: Перед созданием объекта в окне Отчета не забудьте выполнить **оператор Set CoordSys**.

Примеры

В примере используется **оператор Insert** для создания новой строки в таблице SITES. Функция **CreatePoint()** вложена в оператор Insert, для создания объекта типа "Точка", данные которого будут помещены в этой строке.

```
Open Table "sites"
Insert Into sites (obj)
Values ( CreatePoint(-72.5, 42.4) )
```

В следующем примере производится обращение к таблице SITES, которая имеет колонки Xcoord и Ycoord, представляющие долготу и широту объектов. Оператор **Update** использует **функцию CreatePoint()** для сопоставления точечного объекта каждой строке таблицы. Следующий оператор Update присваивает каждой записи в таблице STATES точечный объект. Каждый точечный объект располагается в зависимости от значений в колонках Xcoord и Ycoord.

```
Open Table "sites"
Update sites
Set obj = CreatePoint(xcoord, ycoord)
```

В приведенном выше примере подразумевается, что колонки "Xcoord" и "Ycoord" содержат значения широты и долготы.

См. также:

оператор Create Point, оператор Insert, оператор Update

Оператор Create Point

Назначение

Создает объект типа "точка". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Point
[ Into { Window window_id | Variable var_name } ]
( x, y )
[ Symbol... ] [ Priority n ] [ Name framename ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

x, *y* – координаты точки.

Предложение **Symbol** определяет стиль символа.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**.

Framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Оператор **Create Point** создает новый точечный объект.

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно, объект помещается на заданное место в окне (например, на изменяемый слой). Если **Into** вообще нет в операторе, MapBasic попытается создать объект в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система не объявлялась, то координаты будут принимать значения широты и долготы. Используйте **оператор Set CoordSys** для того, чтобы задать в среде MapBasic другую координатную систему. При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: *X*-координата – это расстояние от левого края листа до точки, и *Y*-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Необязательное **предложение Symbol**, задает стиль символа; см. описание в разделе: **Предложение Symbol**. Если в операторе это **предложение Symbol** не участвует оператор **Create Point** использует установку соответствующего режима для стиля линии в MapInfo (стиль линии можно изменить командой **НастройкиСтиль символов**).

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение Z-Order новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

См. также:

функция CreatePoint(), оператор Insert, предложение Symbol, оператор Update

Оператор Create PrismMap

Назначение

Создает карту-призму. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create PrismMap
[ From Window window_ID | MapString mapper_creation_string ]
{ layer_id | layer_name }
With expr
[ Camera [Pitch angle | Roll angle | Yaw angle | Elevation angle] |
[ Position ( x, y, z ) | FocalPoint ( x, y, z ) ] |
[ Orientation(vu_1, vu_2, vu_3, vpn_1, vpn_2, vpn_3,
clip_near, clip_far) ]
]
[ Light Color lightcolor ]
[ Scale grid_scale ]
[ Background backgroundcolor ]

```

window_id – идентификатор окна для окна Карты, которое содержит слой полигонов. Если слой с полигонами не найден, появится сообщение об ошибке.

mapper_creation_string – определяет командную строку, которая создает текстуру для Карты-призмы;

layer_id – идентификатор слоя Карты;

layer_name – это имя слоя Карты;

expr – выражение, которое вычисляется для каждой записи таблицы;

Camera – определяет позицию и ориентацию камеры.

angle – угол, измеряемый в градусах. Горизонтальный угол в диалоге может изменяться от 0 до 360 и вращает карту вокруг центральной точки грида. Вертикальный угол в диалоге изменяется от 0 до 90 и измеряет вращение в вертикальной плоскости от стартовой точки прямо над картой.

Pitch – угловая величина определяет текущее положение точки наблюдения по оси X;

Roll – угловая величина определяет текущее положение точки наблюдения по оси Z;

Yaw – угловая величина определяет текущее положение точки наблюдения по оси Y;

Elevation – определяет поворот камеры (находящейся в точке фокуса) относительно оси X;

Position – определяет позицию камеры и/или источника освещения;

FocalPoint – определяет фокальную точку камеры и/или источника освещения;

Orientation – определяет для камеры значение параметров ViewUp (*vu_1*, *vu_2*, *vu_3*), ViewPlane Normal (*vpn_1*, *vpn_2*, *vpn_3*) и Clipping Range (*clip_near* и *clip_far*), используется для компенсации инерции зрительного восприятия).

grid_scale – определяет вертикальный масштаб (в направлении Z-координаты). Значение, большее 1 подчеркивает рельеф, значение, меньшее 1 его огрубляет (в направлении Z-координаты).

backgroundcolor – это цвет, используемый для фона; его задает **функция RGB ()**.

Описание

Оператор **Create PrismMap** создает окно карты-призмы. С помощью карты-призмы можно отображать несколько переменных для одного объекта. Например, цвет, ассоциированный с полигоном, может иметь тематическое значение одной колонки, а высота призмы может отражать значение другой колонки. Оператор **Create PrismMap** соответствует команде MapInfo Pro **Карта-призма**.

Между сеансами работы MapInfo Pro сохраняет настройки карты-призмы, добавляя оператор **Create PrismMap** в файл рабочего набора. Таким образом, чтобы увидеть пример оператора **Create PrismMap**, создайте карту, выполните команду **Добавить тематику** на вкладке **КАРТА**, сохраните рабочий набор (например, `PRISM.WOR`) и изучите текст рабочего набора текстовом редакторе. После этого скопируйте оператор **Create PrismMap** в вашу программу MapBasic. Аналогично можно увидеть примеры операторов **Create PrismMap** при открытии окна **MapBasic** перед выполнением команды **Добавить тематику**.

Каждый оператор **Create PrismMap** должен определять предложение для выражения *expr*. MapInfo Pro оценивает это выражение для каждого объекта в таблице, из которой строятся

призмы; обрабатывая действия оператора **Create PrismMap**, MapInfo Pro выбирает стиль отображения каждого объекта, основываясь на значении записи *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Дополнительное предложение *window_id* определяет, какая карта послужит основой карты-призмы; если предложение *window_id* отсутствует, MapBasic создаст призмы для самого верхнего окна карты. Оператор **Create PrismMap** указывает, какой слой надо использовать, даже если окно карты имеет только один слой. Слой может быть идентифицирован по номеру (*layer_id*), где самый верхний слой карты имеет *layer_id* равный 1, следующий слой имеет *layer_id* равный 2, и т.д. Или же оператор **Create PrismMap** может идентифицировать слой карты по имени (например, "world").

Пример

```
Open Table "STATES.TAB" Interactive
Map From STATES
Create PrismMap From Window FrontWindow() STATES With Pop_1980 Background
RGB(192,192,192)
```

См. также:

[оператор Set PrismMap](#), [функция PrismMapInfo\(\)](#)

Оператор Create Query

Назначение

Создает таблицу на основе содержимого заданного окна списка. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Query
  From Window window_id
  Into query_name
```

window_id – целочисленный ID-номер окна Списка.

query_name – имя создаваемой таблицы.

Ошибки

ERR_TABLE_ALREADY_OPEN порождается, если уже открыта таблица с именем query_name
window_id – целочисленный ID-номер окна списка.

Описание

Можно применить к содержимому окна списка фильтр или отсортировать его, после чего результат можно использовать как таблицу в других операциях. Оператор Create Query позволяет создать таблицу запроса, представляющую собой содержимое окна списка. Результирующая таблица представляет набор отфильтрованных строк. Эту таблицу могут использовать другие операторы MapBasic, такие как **оператор Update**, **оператор Commit Table** и **оператор Map**.

Пример

```
Browse * From World
Set Browse Filter Where (Continent = "North America" Or Continent =
"South
America")
Create Query From Window Frontwindow() into Americas
Map From Americas
```

См. также:

оператор Set Browse, **оператор Select**

Оператор Create Ranges

Назначение

Вычисляет значения диапазонов для условного (тематического) выделения методом выделения диапазонов и помещает объекты в массив переменных, который может впоследствии использовать **оператор Shade**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Ranges
  From table
  With expr
  [ Use { "Equal Ranges" | "Equal Count" | "Natural Break" | "StdDev" }
  ]
```

```
[ Quantile Using q_expr ]
[ Number num_ranges ]
[ Round rounding_factor ]
Into Variable array_variable
```

table – имя открытой таблицы, объекты которой будут участвовать в тематическом выделении;

expr – выражение, которое вычисляется для каждой записи таблицы;

q_expr – выражение, используемое при квантовании;

num_ranges – задает число диапазонов (по умолчанию 4);

rounding_factor – шаг, по которому округляются диапазоны при разделении (например, значение 10 задает округление до ближайшего десятка);

array_variable – массив численных переменных типа Float, в который будут помещены значения для диапазонов.

Описание

Оператор **Create Ranges** вычисляет значения диапазонов, которые впоследствии использует **оператор Shade** для создания тематического слоя на карте методом выделения диапазонов. Подробную информацию о тематических картах смотрите в документации по MapInfo Pro.

Предложение **Use** задает метод, каким будут разделены данные на диапазоны. Задав "Equal Ranges", разделите диапазоны, исходя из разброса значений (например, 0-25, 25-50, 50-75, 75-100). Задав "Equal Count", разделите диапазоны, исходя из количества записей в таблице, т.е. в диапазоны попадет примерно одинаковое количество записей. Задав "Natural Break", разделите диапазоны, исходя из естественно близких групп значений. Задав "StdDev", сначала делите диапазоны по среднему значению, а затем добавляете один диапазон со значениями выше среднего, но не далее величины дисперсии от среднего, а также еще один диапазон со значениями ниже среднего, но не далее величины дисперсии от среднего. MapInfo Pro использует стандартное отклонение ($N - 1$).

Предложение **Into Variable** определяет имя массива переменных типа Float, в который будет помещен результат. Не нужно следить за размерами этого массива; MapInfo Pro автоматически изменяет его размер. Размер этого массива будет всегда в два раза больше количества диапазонов, так как в нем помещаются как нижняя, так и верхняя граница диапазона.

После выполнения оператора **Create Ranges** выполните **оператор Shade**, чтобы создать тематическую карту; включите дополнительный параметр **From Variable** в оператор Shade, с помощью которого можно прочитать массив границ диапазонов. Обычно оператор Shade использует ту же таблицу и то же выражение для колонки, что и оператор **Create Ranges**.

Квантование диапазонов

Предложение **Quantile Using** отключает предложение **Use** и назначает разделение диапазонов квантованием, которое задается выражением *q_expr*.

Квантование лучше всего проиллюстрировать следующим примером. Оператор вычисляет границы диапазонов покупательной способности населения США (BPI), квантуя их по значениям населения штатов.

```
Create Ranges From states
  With BPI_1990 Quantile Using Pop_1990
  Number 5
  Into Variable f_ranges
```

В этом примере создается пять диапазонов (Number 5).

Штаты с высокой покупательной способностью населения (With BPI_1990) помещаются в "высшие" диапазоны (темные оттенки цвета), а с низкой – в "низшие" (светлые оттенки цвета).

Так как предложение `Quantile Using Pop_1990` использует колонку `Pop_1990`, то при вычислении границ промежуточных диапазонов используется метод квантования по значениям населения штата. Границы диапазонов устанавливаются следующим образом: так как предложение **Quantile Using** задает колонку "Pop_1990", то MapInfo Pro сначала вычисляет общее количество населения США (около 250 миллионов). Затем MapInfo Pro делит результат на количество диапазонов (в нашем случае, 5) и получается пятьдесят миллионов. После этого MapInfo Pro делит диапазоны так, чтобы суммарное население штатов, охватываемых диапазоном, приближалось (но не превосходило) пятьдесят миллионов.

MapInfo Pro собирает штаты по порядку возрастания покупательной способности BPI, и в первый, "низший" диапазон попадают штаты с самой низкой покупательной способностью. MapInfo Pro продолжает добавлять штаты в первый диапазон до тех пор, пока суммарное значение численности населения не достигнет или не превысит пятидесяти миллионов. В этот момент MapInfo Pro решает, что первый диапазон готов и приступает к обсчету следующего. Во второй диапазон MapInfo Pro помещает записи о штатах, пока суммарное значение численности населения не достигнет или не превысит сто миллионов; это означает, что второй диапазон заполнен; и т.д.

Пример

```
Include "mapbasic.def"

Dim range_limits() As Float, brush_styles() As Brush
Dim col_name As Alias

Open Table "states" Interactive

Create Styles
  From Brush(2, CYAN, 0) 'style for LOW range
  To Brush (2, BLUE, 0) 'style for HIGH range
  Vary Color By "RGB"
  Number 5
  Into Variable brush_styles
  ' Store a column name in the Alias variable:
  col_name = "Pop_1990"
```

```

Create Ranges From states
  With col_name
  Use "Natural Break"
  Number 5
  Into Variable range_limits

Map From states

Shade states
  With col_name
  Ranges
    From Variable range_limits
    Style Variable brush_styles

' Show the theme legend window:
Open Window Legend

```

См. также:

[оператор Create Styles](#), [оператор Set Shade](#), [оператор Shade](#)

Оператор Create Rect

Назначение

Создает объект "прямоугольник" или "квадрат". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Rect
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 )
[ Pen ... ]
[ Brush ... ] [ Priority n ] [ Name framename ]

```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

x1, *y1* – координаты начального угла прямоугольника;

x2, *y2* – координаты противоположного по диагонали угла прямоугольника;

Pen – корректное [предложение Pen](#), задающее стиль линии.

Brush – корректное [предложение Brush](#), задающее штриховку.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**.

framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Если в операторе **Create Rect** присутствует предложение **Into Variable**, созданный объект объявляется значением объектной переменной *var_name*. Если слово **Into** указывает окно **Window**, объект помещается на заданное место в окне (например, в изменяемый слой). Если **Into** вообще нет в операторе, MapBasic попытается создать объект в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется **оператор Set CoordSys**. При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: *X*-координата – это расстояние от левого края листа до точки, и *Y*-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы сменить единицы измерения, выполните **оператор Set Paper Units**.

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Если явно задано **предложение Pen**, то оно определяет стиль линии; см. подробности в разделе: **Предложение Pen**. Если в операторе не участвует **предложение Pen** оператор **Create Rect** использует текущий стиль линии в MapInfo (который задан в диалоге команды **Стиль линий**). Аналогично, **предложение Brush** обозначает стиль штриховки; подробнее см. раздел: **Предложение Brush**.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение Z-Order новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

См. также:

предложение Brush, **оператор Create RoundRect**, **оператор Insert**, **предложение Pen**, **оператор Update**

Оператор Create Redistricter

Назначение

Открывает окно районирования. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Redistricter source_table By district_column
With
  [ Layer <layer_number> ]
  [ Count ]
  [ , Brush ] [ , Symbol ] [ , Pen ]
  [ , { Sum | Percent } ( expr ) ] [ , { Sum | Percent } ( expr ) ... ]

  [ Percentage From expr ]
  [ Percentage from { column | row } ]
  [ Order { "MRU" | "Alpha" | "Unordered" } ]

```

source_table – имя открытой таблицы, объекты которой будут участвовать в районировании;

district_column – имя колонки; начальный набор районов создается на базе значений из этой колонки и в нее же помещается новая структура районов;

layer_number – номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

Предложение **Count** предопределяет показ в списке районов колонки с количеством объектов в каждой группе;

Предложение **Brush** назначает показ колонки с образцами штриховки объектов;

Предложение **Symbol** назначает показ колонки с образцами символов;

Предложение **Pen** назначает показ колонки с образцами линий объектов;

expr – числовое выражение;

Предложение **Percentage From** задает режим вычисления процентов в строках.

Предложение **Order** задает порядок строк в списке районов (по алфавиту, произвольно или же все затронутые изменения строки помещаются в начало списка; последний режим (MRU) используется по умолчанию).

Описание

Оператор **Create Redistricter** начинает сеанс районирования. Этот оператор соответствует выполнению команды MapInfo Pro **Районирование** на вкладке **ГЛАВНАЯ**. Правила работы с районами подробно описаны в документации MapInfo Pro.

Для управления составом районов используется **оператор Set Redistricter**. Закончить сеанс районирования можно, выполнив **оператор Close Window**, закрывающий окно Районирование.

Если включить слово **Brush**, то в окно районирования будет добавлена колонка с образцами штриховок каждого района. Обратите внимание на то, что это не **предложение Brush**, а отдельное ключевое слово **Brush**. Аналогично **Symbol** и **Pen** – это отдельные ключевые слова, а не **предложение Symbol** или **предложение Pen**. Если в списке районов есть колонки с образцами оформления районов, то пользователь может их изменять, указывая на них мышкой.

Если задано предложение **Percentage From**, то проценты рассчитываются построчным методом. Если предложение **Percentage From** не задано, то проценты рассчитываются по колонкам.

См. также:

оператор Set Redistricter

Оператор Create Region

Назначение

Создает объект типа "область". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Region
[ Into { Window window_id | Variable var_name } ]
num_polygons
[ num_points1 ( x1, y1 ) ( x2, y2 ) [ ... ] ]
[ num_points2 ( x1, y1 ) ( x2, y2 ) [ ... ] ... ]
[ Pen ... ]
[ Brush ... ]
[ Center ( center_x, center_y ) ]
[ Priority n ] [ Name frameName ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

num_polygons – описывает число полигонов в регионе (от нуля и больше);

num_points1 – число узлов в первом полигоне;

num_points2 – число узлов во втором полигоне, и т. д.;

Каждая пара *x*, *y* задает координаты узла полигона.

Pen – корректное [предложение Pen](#), задающее стиль линии.

Brush – корректное [предложение Brush](#), задающее штриховку.

center_x – X-координата центроида области;

center_y – Y-координата центроида области.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**.

framename – строка, определяющая имя объекта в окне конструктора **Отчёт**.

Описание

Результатом действия оператора **Create Region** является новый объект типа "Область".

Параметр *num_polygons* определяет количество многоугольников (полигонов), включенных в область. Если *num_polygons* приравнять нулю, то оператор создаст пустую область (область без полигонов). Позже можно, используя [оператор Alter Object](#), добавлять детали в этот объект.

В некоторых приложениях вам будет удобно разделить создание области на два этапа: сначала при помощи оператора **Create Region** создать объект, не имеющий полигонов, а затем добавлять в объект необходимые элементы, используя оператор **Alter Object**. Если на момент создания объекта нет полной информации о количестве и расположении многоугольников и узлов в будущей области, используйте для добавления узлов. Подробнее см. [Оператор Alter Object](#).

Если оператор использует предложение **Into Variable**, то созданный объект объявляется как значение объектной переменной. Если слово **Into** указывает окно, объект помещается на заданное место в окне (например, на изменяемый слой). Если **Into** вообще нет в операторе, MapBasic попытается создать объект в самом верхнем окне. Если это невозможно (например, поверх всех окон лежит окно графика), то объект не будет создан.

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется [оператор Set CoordSys](#). При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: X-координата – это расстояние от левого края листа до точки, и Y-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#). По умолчанию, MapBasic использует дюймы. Чтобы использовать другие единицы, следует использовать [оператор Set Paper Units](#).

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор [Set CoordSys Layout](#).

Необязательный параметр **предложение Pen** задает стиль линии контура объекта; подробнее см. раздел: **Предложение Pen**. Если параметр Pen не задан оператор **оператор Create RoundRect** использует текущий стиль линии (который задается в диалоге **Стиль Линии**. Аналогично, предложение Brush назначает стиль штриховки; подробнее см. раздел: **Предложение Brush**.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение Z-Order новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

Однофрагментный объект типа "Полигон" может содержать до 134,217,724 узлов. Общее количество фрагментов в полигоне (или коллекции) не должно превышать 20,648,881.

Пример

```
Dim obj_region As Object
Dim x(100), y(100) As Float
Dim i, node_count As Integer

' If you store a set of coordinates in the
' x() and y() arrays, the following statements
' will create a region object that has a node
' at each x,y location:

' First, create an empty region object
Create Region Into Variable obj_region 0

' Now add nodes to populate the object:
For i = 1 to node_count
  Alter Object obj_region Node Add ( x(i), y(i) )
Next

' Now store the object in the Sites table:
Insert Into Sites (Object) Values (obj_region)
```

См. также:

оператор Alter Object, предложение Brush, оператор Insert, предложение Pen, оператор Update

Оператор Create RoundRect

Назначение

Создает объект типа "Скругленный прямоугольник". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create RoundRect
[ Into { Window window_id | Variable var_name } ]
( x1, y1 ) ( x2, y2 ) rounding
[ Pen ... ]
[ Brush ... ] [ Priority n ] [ Name framename ]
```

window_id – идентификатор окна, целое число.

var_name – имя объектной переменной;

x1, *y1* – координаты начального угла прямоугольника;

x2, *y2* – координаты противоположного по диагонали угла прямоугольника;

rounding – действительное число, задающее в единицах измерения окна (например, в дюймах для отчета или в градусах для карты) диаметр скругления угла прямоугольника.

Pen – корректное [предложение Pen](#), задающее стиль линии.

Brush – корректное [предложение Brush](#), задающее штриховку.

n – целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**.

Framename – строка, определяющая имя объекта в окне **Отчёт**.

Описание

Результатом действия оператора **Create RoundRect** является новый объект типа "Скругленный прямоугольник" (прямоугольник с сглаженными углами).

Параметры *X* и *Y* задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется [оператор Set CoordSys](#). При этом надо учесть, что MapBasic игнорирует координатную систему самого окна карты. Если объект создается для окна отчета, измерения производятся в объявленных ранее единицах измерения листа: *X*-координата – это расстояние от левого края листа до точки, и *Y*-координата – расстояние от верхнего края листа. "Бумажные" единицы измерения

описаны в разделе: **Оператор Set Paper Units**. По умолчанию, MapBasic использует дюймы. Чтобы сменить единицы измерения, выполните **оператор Set Paper Units**.

Примечание: Перед созданием объекта в окне отчета не забудьте выполнить оператор **Set CoordSys Layout**.

Необязательный параметр **предложение Pen** задает стиль линии контура объекта; подробнее см. раздел: **Предложение Pen**. Если параметр Pen не задан оператор **оператор Create RoundRect** использует текущий стиль линии (который задается в диалоге **Стиль Линии**). Аналогично, предложение Brush назначает стиль штриховки; подробнее см. раздел: **Предложение Brush**.

При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1. Используйте предложение **Priority**, для того чтобы присваивать значение Z-Order новой созданной рамке окна **Отчёт**.

С помощью параметра **Name** можно присвоить имя рамке окна **Отчёт**. Если присвоено имя, то оно будет записано в файл рабочего набора (WOR-файл), а номер версии рабочего набора будет повышен до 1500.

См. также:

предложение Brush, **оператор Create Rect**, **оператор Insert**, **предложение Pen**, **оператор Update**

Оператор Create Styles

Назначение

Создает массив из значений стилей линии, штриха или символа. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Styles
From { Pen ... | Brush ... | Symbol ... }
To { Pen ... | Brush ... | Symbol ... }
Vary { Color By { "RGB" | "HSV" } | Background By { "RGB" | "HSV" } |
      Size By { "Log" | "Sqrt" | "Constant" } }
[ Number num_styles ]
[ Inflect At range_number With { Pen ... | Brush ... | Symbol ... } ]
Into Variable array_variable
```

num_styles – число создаваемых значений стиля, по умолчанию 4;

range_number – число типа SmallInt, задающее номер диапазона последнего перед переломом;

array_variable – массив переменных Pen, Brush или Symbol.

Pen – корректное **предложение Pen**, задающее стиль линии.

Brush – корректное **предложение Brush**, задающее штриховку.

Symbol – корректное **предложение Symbol**, задающее символ для точечного объекта.

Описание

Оператор **Create Styles** создает множество значений стиля линии, штриха или символа и присваивает его массиву переменных соответствующего типа. Массив этих значений в дальнейшем может использовать **оператор Shade** для создания тематических карт. О создании тематических карт смотрите документацию MapInfo Pro.

Предложения **From** и **To** задают первое и последнее значения стиля: Pen, Brush или Symbol. Если массив позже будет использован в условном (тематическом) выделении, то стилем из предложения **From** будут выделены объекты самого "нижнего" диапазона, а объекты самого "верхнего" диапазона будут выделены стилем из предложения **To**.

Оператор **Create Styles** строит множество переходных значений от значения, начиная с заданного в предложении **From**, до значения стиля, заданного в предложении **To**. Например, после слова **From** может быть задано **предложение Brush** – насыщенный синий для морских глубин, а после слова **To** задан – светло-голубой для отмелей. В этом случае MapInfo Pro построит последовательность значений стиля штриха (тип Brush), где первое будет иметь насыщенный синий цвет, последнее значение стиля – светло-голубой, а промежуточные значения – переходные оттенки.

Предложение **Number** задает количество значений стиля, включая заданные предложениями **To** и **From** первое и последнее значения. Это число должно соответствовать числу диапазонов условного (тематического) выделения, если затем выполняется **оператор Shade**.

Предложение **Vary** задает порядок, согласно которому атрибуты значений стиля будут меняться в создаваемом ряду. Для задания изменения цвета штриховок и линий переднего плана используется предложение **Color**. Для задания изменения цвета фона используется предложение **Background**. В обоих случаях задается перебор значений диапазона цветов окраски по шкале RGB или по шкале HSV. Если создаете ряд для стиля символа, то можно использовать предложение **Size** для интерполяции по размеру символа. Для стиля линии (тип Pen) предложение **Size** задает изменение толщины линии.

Если используется предложение **Inflect At**, то задаете переломную (или пороговую) точку в вычислении ряда значений стилей между значениями, заданными предложением **From**, и значениями, заданными предложением **To**. При этом MapInfo Pro создает две последовательности значений: одну от **From** до значения перелома **Inflect At**, и другую от **Inflect** до **To**. Например, переломное значение используется при построении условной карты прибылей и убытков. Области, где предприятия получают прибыль, будут окрашены разными оттенками зеленого, а области, где предприятия терпят убытки, будут окрашены разными оттенками красного. Переломное значение можно использовать только при раскраске разными цветами.

Предложение **Into Variable** задает имя массива переменных (без скобок). Нет необходимости заботиться о размерности массива, MapBasic автоматически подгонит размерность массива по числу значений, если это необходимо. Значения в массиве *array_variable* (Pen, Brush или Symbol) должны соответствовать по стилю тем, которые заданы в предложениях **From - To**.

Пример

Следующий пример иллюстрирует применение оператора **Create Styles**.

```
Dim brush_styles() As Brush

Create Styles
  From Brush(2, CYAN, 0) 'style for LOW range
  To Brush (2, BLUE, 0) 'style for HIGH range
  Vary Color By "RGB"
  Number 5
  Into Variable brush_styles
```

Этот оператор **Create Styles** создает ряд из пяти значений стиля штриха. Значения помещаются в ячейки массива "b_ranges". Следующий [оператор Shade](#) может создать условную карту, в которой будут использованы стили из массива "b_ranges". См. пример в разделе: [Оператор Create Ranges](#).

См. также:

[оператор Create Ranges](#), [оператор Set Shade](#), [оператор Shade](#)

Оператор Create Table

Назначение

Создает новую таблицу. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Create Table table_name
[ ( column columntype [ , ... ] ) | Using from_table ]
[ File filespec ]
[ {
  Type { NATIVE [ Charset char_set ] [ Version version_pro ] [ Hide ] |
    NATIVEX [ Charset char_set ] [ Locale locale ] [ Hide ] } |
  Type DBF [ CharSet char_set ] [ Version version_pro ] |
  Type { Access | ODBC }
```

```

Database database_filespec [ Version version_msaccess ]
Table db_table_name [ Password pwd ] [ CharSet char_set ]
[ Version version_pro ] |
Type TILESERVER
  TileType { LevelRowColumn | QuadKey }
  URL url
  [ AttributionText "attributiontext" ] [ Font font_clause ]
  [ StartTileNum { 0 | 1 } ]
  [ Minlevel min_level ]
  MaxLevel max_level
  [ Origin { "SW" | "NW" } ]
  Height tile_height [Width tile_width]
  [ ReadTimeout read_time_out ]
  [ RequestTimeout request_time_out]
  CoordSys coordsys
Type TILESERVER
  TileType { WMTS }
  URL url
  Layer layer
  TileMatrixSet tileMatrixSet
  Style style
  Format imageFormat
  [ Override Coordinate Order { On | Off } ]
  [ ReadTimeout read_time_out]
  [ RequestTimeout request_time_out]
  [ CoordSys coordsys ]
  [ PassThrough "name1" "Value1", "name2" "Value2", ... ]
} ]

```

table_name — имя для новой таблицы MapInfo Pro.

column — имя колонки в новой таблице. Имя колонки может быть длиной до 31 символов, и может содержать буквы, числа, и знак подчеркивания. Имя колонки не может начинаться с цифры.

columntype — тип данных, связанных с колонкой. Каждый параметр *columntype* задает тип данных колонки и имеет следующий синтаксис:

```

Char( width ) | Float | Integer | SmallInt |
Decimal( width, decplaces ) | Date | DateTime | Time | Logical

```

Где **Decimal** *width* может принимать значения от 1 до 20, а **Decimal** *decplaces* — от 0 до 16. Параметр *decplaces* должен быть либо нулём (0), либо минимум на 2 меньше *width*; если *width* равняется 10, *decplaces* может до 8.

from_table — имя открытой таблицы, в которой находится та колонка, которую надо сохранить в новой таблице. Таблица *from_table* должна быть базовой и должна содержать колонку с данными. Таблицы запросов и растровые таблицы не могут использоваться для этих целей, их применение даст сообщение об ошибке. Структура колонок в новой таблице будет идентичной той, которая задана в исходной таблице.

filespec — определяет место, где создавать файлы TAB, MAP и ID (и, в случае с Access, AID-файлов). Если вы пропускаете предложение **File**, то файлы создаются в текущем каталоге.

version — равно 100 (при создании таблицы, которая может читаться ранними версиями MapInfo Pro) или 300 (формат MapInfo 3.0). Этот параметр не применяется при создании таблицы Access; версия для таблицы Access управляется DAO.

Type DBF:

char_set — имя набора символов; см. раздел [Предложение CharSet](#).

Type { Access | ODBC }:

database_filespec — строка, определяющая доступную базу данных Access. Если такая база не существует, MapInfo Pro создаст новый Access-файл MDB или ACCDB;

version_msaccess — выражение, определяющее версию базы данных Microsoft Jet, используемую для новой базы данных. Допускаются значения 4.0 (для Access 2000) или 3.0 (для Access '95/'97). Если пропущен, то стандартной версией является 2019. Если база данных, в которой создаётся таблица, уже существует, заданная версия базы данных игнорируется.

db_table_name — строковая переменная, определяющая имя таблицы, которая появится в Access;

pwd — пароль на уровне базы данных, определяемый при включении защиты базы данных.

char_set — имя набора символов; см. раздел [Предложение CharSet](#).

Type TILESERVER:

url — полный URL-адрес, либо `http://<server>`, либо `https://<server>`, сервера тайлов. Если URL-адрес не содержит следующих значений, то оператор Create Table не сработает:

- Если для *tile_type* задано **QuadKey**, то URL-адрес должен содержать {QUADKEY}.
- Если для *tile_type* задано **LevelRowColumn**, то URL-адрес должен содержать компоненты {LEVEL}, {ROW} и {COLUMN}, в которые программно подставляются конкретные значения. Разные серверы поддерживают теги {ROW} и {COL} по-разному; иногда нужно заменять их на парой строка-колонка (или парой X, Y).
- Если параметром типа тайлов *tile_type* является **WMTS**, то URL-адрес должен содержать тэги {TileMatrix}, {TileRow} и {TileCol}.

attributiontext — атрибутивный текст, который приходит вместе с тайлами и показывается на карте. Текст должен быть заключен в кавычки (" . . . ").

font_clause — необязательный параметр, задающий стиль шрифта атрибутивного текста. Это выражение для работы со шрифтами, например, **MakeFont** (*fontname*, *style*, *size*, *fgcolor*, *bcolor*).

```
Font ("Verdana", 1, 24, 0, 255)
Font MakeFont("Verdana", 1, 24, 0, 255)
```

Подробности содержат разделы: [Предложение Font](#) и [Функция MakeFont\(\)](#).

Значение *min_level* задает минимальное количество тайлов, высылаемых с сервера. Это значение должно быть нулем или положительным числом, меньшим чем **max_level**. По умолчанию это 0.

max_level задает максимальное количество тайлов, высылаемых с сервера. Это значение должно быть позитивным.

tile_height — высота тайла в пикселах. Это значение должно быть позитивным.

tile_width — ширина тайла в пикселах. Это значение должно быть позитивным. Если его не задать, ширина тайла будет равна высоте.

layer — идентификатор WMTS-слоя, который требуется отобраз.

tileMatrixSet — идентификатор заданной на WMTS-слое матрицы тайлов.

style — стиль оформления слоя, который требуется выбрать из стилей, поддерживаемых WMTS-слоем.

imageFormat — формат изображений тайлов, который требуется выбрать из форматов, поддерживаемых WMTS-слоем.

read_time_out — количество секунд, за которые должна пройти операция считывания тайла (стандартное значение – 300). Это значение должно быть позитивным.

request_time_out — количество секунд, потраченных на ожидание ответа от сервера тайлов (стандартное значение – 100). Это значение должно быть позитивным.

Параметр *coordsys* задает стандартную координатную систему сервера тайлов. Так как MapInfo Pro не может перепроецировать тайлы, то перепроецированию в координатную систему сервера тайлов подвергается сама карта. Карту, содержащую слой с тайлами, уже невозможно перепроецировать заново. Границы координатной системы также задают границы первого тайла (на минимальном уровне одного тайла). Так MapInfo Pro узнает, как вычислять границы тайлов на остальных уровнях.

Дополнительное, необязательное ключевое слово **Hide** используется, для того чтобы создать таблицу, которая не перечислена списке таблиц пользовательского интерфейса. В `TableInfo()` атрибут `TAB_INFO_ISHIDDEN` возвращает логическую единицу – True.

Описание

Оператор **Create Table** создает новую пустую таблицу размером до 250 колонок. Определив параметр **ODBC**, можно создать новую таблицу на сервере СУБД.

Предложение **Using** позволяет создавать новую таблицу в ходе операции объединения объектов по колонке. Таблица *from_table* должна быть базовой и должна содержать колонку с данными. Таблицы запросов и растровые таблицы не могут использоваться для этих целей, их применение даст сообщение об ошибке. Структура (колонок) создаваемой новой таблицы будет идентична структуре данной таблице.

Добавочное предложение **File** определяет место на диске, где создается новая таблица. Если предложение **File** не используется, таблица создается в текущем каталоге (папке).

Предложение **Type** определяет формат данных таблицы. По умолчанию это формат MapInfo (**NATIVE**), но может быть также и **DBF**. Формат MapInfo (**NATIVE**) занимает меньше места, чем формат DBF, зато формат DBF читается в любых dBASE-совместимых системах управления базами данных. Таким же образом создаются новые таблицы на серверах СУБД с помощью предложения **ODBC Type** оператора **Create Table**. Предложение **NATIVEX** позволяет задавать расширенный MapInfo Extended TAB-файл, в котором поддерживаются наборы символов Unicode, например, UTF-8 или UTF-16.

Предложение **Locale**, используемое совместно с предложением **NATIVEX**, позволяет специфицировать локаль создаваемого индекса, что влияет на сортировку таблицы. Для этого используется строковое значение, например, "en-US". Если этот параметр не задан, то используется стандартная системная локаль. Это предложение применяется только к таблицам NativeX (расширенный MapInfo Extended TAB) с наборами символов (charset) UTF-8 или UTF-16.

Предложение **CharSet** определяет набор символов. Параметр *char_set* должен быть строковой константой, такой как "MacRoman" или "WindowsLatin1". Если предложение **CharSet** не определено, MapBasic использует по умолчанию текущий набор символов Windows. Подробнее смотрите описание параметра **CharSet**.

В колонке с короткими целыми типа **SmallInt** на каждое значение резервируется два байта, таким образом в колонке могут храниться значения от -32,767 до +32,767. В колонке с целыми числами типа **Integer** на каждое значение резервируется четыре байта; таким образом колонка может содержать значения от -2,147,483,647 до +2,147,483,647.

Предложение **TileType** позволяет задавать тип тайл-сервера, который будет использовать данная таблица: **QuadKey**, **LevelRowColumn** или **WMTS**. Эти режимы задают способ отсчета тайлов на сервере. Эти способы таковы:

QuadKey — сервер использует алгоритм четвертичного дерева; изображения при этом поделены на квадраты 256 x 256 пикселей. Каждый тайл обозначается уникальной строкой символов в диапазоне 0 – 3 (отсюда термин **QuadKey**), которая описывает положение и масштаб тайла.

LevelRowColumn — мир разделен на квадраты, которым сопоставлены идентификаторы в виде списка, содержащего масштаб, строку и колонку тайла. Формат идентификатора тайлов может различаться от сервера к серверу, в частности, теги {ROW} и {COL} могут меняться местами.

WMTS — сервер, на котором реализуются стандарты, опубликованные консорциумом OGC для WMT-служб (Web Map Tile – картографических веб-служб тайлов).

Предложение **StartTileNum** не обязательно. Это номер стартового тайла: либо 0, либо 1. По умолчанию нумерация тайлов начинается с 0.

Предложение **Origin** не обязательно задавать для таблиц типа TileServer. Если оно задано, нужно задать одно из значений: "NW" или "SW". Значение "SW" задает Юго-Западную точку отсчета, в этом случае фрагменты в таблице TileServer укладываются снизу вверх, начиная от нижнего левого угла. Значение "NW" задает Северо-западную точку отсчета; фрагменты укладываются начиная от верхнего левого угла. Если значение не задано, принимается значение "NW".

Предложение **Override Coordinate Order** является дополнительным, необязательным, которое применяется, когда в матрице тайлов WMTS-сервера встречается некорректный порядок перечисления координат.

PassThrough — это дополнительный, необязательный набор пар "имя/значение", пересылаемых на WMTS-сервер. Он может использоваться для индивидуальных параметров, поддерживаемых сервером, например, дополнительных размерностей. В предложении можно использовать несколько пар "имя/значение".

Предложение **Version** контролирует формат таблицы. Если задать `Version 100`, MapInfo Pro создаст таблицу в формате, читаемом ранними версиями MapInfo Pro. Если задаётся параметр `Version 300`, MapInfo Pro создаёт таблицу в формате, который использовался в программе MapInfo Pro 3.0. Обратите внимание, что объекты-области и объекты-полилинии, содержащие более 8 000 узлов, и объекты-полилинии, состоящие из нескольких сегментов, требуют использования спецификаций версии 300. Если опустить использование параметра, задаваемого предложением **Version**, создаётся таблица в формате версии 300.

Сообщения, сопровождающие создание таблицы, содержащей тайлы

Если при обращении к серверу тайлов происходят ошибки, то причины их могут быть следующими:

- Некорректный URL-адрес сервера.
- Сервер тайлов недоступен.
- Сервер не отвечает в заданное время.
- Аутентификация на срабатывает из-за несовместимых режимов сетевой защищенности серверов.

Возможные причины сбоев при загрузке тайлов:

- Не найдено свойство в файле конфигурации.
- Не найден файл конфигурации.

Если ошибки возникают при создании файла конфигурации, который по своей природе есть XML-файл, проверьте, доступен ли каталог, в котором он должен быть создан.

Пример

Следующий пример показывает, как создать таблицу TMS тайл-сервера, в которой фрагменты откладываются от Юго-Западного угла.

```
Dim sPath As String
sPath="D:\MapInfo\Tables\TMS.tab"
Create Table PathToTableName$(sPath)
File sPath
Type TILESERVER
TileType "LevelRowColumn"
URL "http://INSERT_TILE_SERVER_NAME/{LEVEL}/{ROW}/{COL}.png"
```

```

    AttributionText "required attribution text" Font
("Arial",256,10,0,16777215)
    MinLevel 0
    MaxLevel 15
    Height 256
    CoordSys Earth Projection 20, 109, 7, 5.387638889, 52.156160556,
0.9999079, 155000, 463000 Bounds (-285401.92, 22598.08) (595401.92,
903401.92)
    Origin "SW"

```

Следующий пример показывает, как создать таблицу TOWNS, и в ней будет 3 поля: символьное поле, названное townname, целочисленное поле, названное population, и десятичное поле, названное median_income. Файл будет создан в подкаталоге C:\MAPINFO\DATA. Поскольку использовано предложение **Type**, таблица будет создана в формате dBASE.

```

Create Table Towns
( townname Char(30),
  population SmallInt,
  median_income Decimal(9,2) )
File "C:\MAPINFO\TEMP\TOWNS"
Type DBF

```

В следующем примере создаётся нативная таблица MapInfo (TAB-файл) с названием test и набором символов "WindowsGreek".

```

Create Table "test" (Field1 Char(10)) file "test.TAB" TYPE NATIVE Charset
"WindowsGreek"

```

В следующем примере создаётся расширенная таблица MapInfo Extended (TAB-файл) с набором символов UTF-16 и английской локалью для США ("en-US").

```

Create Table "pangramsNativeX_utf_16" (fid LargeInt, language
Char(254), textval Char(254), number_of_utf16_chars
LargeInt, number_of_utf8_chars LargeInt) file "pangramsNativeX_utf-16.TAB"
TYPE NATIVEX Charset "UTF-16" Locale "en-US"

```

Примеры создания таблиц тайлов

В следующих примерах показано, как обращаться к серверам тайлов для создания таблиц тайлов. Имена таблиц и файлов в этих примерах условны, замените их своими. Параметр AttributionText должен содержать тексты, требующиеся для провайдера сервера.

Создание таблицы тайлов, базирующейся на сервере **MapInfo Developer**:

```

Create Table MDev_TileServer
File "MDev_TileServer"
Type TILESERVER
TileType "LevelRowColumn"
URL

```

```
"http://INSERT_SERVER_NAME_HERE/MapTilingService/MapName/{LEVEL}/{ROW}
:{COL}/tile.gif"
AttributionText "required attribution text"
Font("Verdana",255,16,0,255)
StartTileNum 1
MaxLevel 20
Height 256
CoordSys Earth Projection 10, 157, 7, 0 Bounds(-20037508.34,-
20037508.34) (20037508.34,20037508.34)
```

Создание таблицы тайлов, базирующей на сервере **MapXtreme.NET**:

```
Create Table MXT_TileServer
File "MXT_TileServer"
Type TILESERVER
TileType "LevelRowColumn"
URL
"http://INSERT_SERVER_NAME_HERE/TileServer/MapName/{LEVEL}/{ROW};{COL}
/tile.png"
AttributionText "required attribution text"
Font("Calibri",255,16,0,255)
MaxLevel 20
Height 256
RequestTimeout 90
ReadTimeout 60
CoordSys Earth Projection 10, 157, 7, 0 Bounds(-20037508.34,-
20037508.34) (20037508.34,20037508.34)
```

Создание таблицы тайлов, базирующей на сервере **OpenStreetMap**:

```
Create Table OSM_TileServer
File "OSM_TileServer"
Type TILESERVER
TileType "LevelRowColumn"
URL
"http://INSERT_OPEN_STREET_MAP_SERVER_NAME_HERE/{LEVEL}/{ROW}/{COL}.pn
g"
AttributionText "required attribution text" Font("Arial",255,16,0,255)

MinLevel 0
MaxLevel 15
Height 256
CoordSys Earth Projection 10, 157, 7, 0 Bounds(-20037508.34,-
20037508.34) (20037508.34,20037508.34)
```

Создание таблицы тайлов, базирующей на сервере тайлов **WMTS**:

```
Create Table WMTS_TileServer
File "WMTS_TileServer"
Type TILESERVER
TileType "WMTS"
```

```

URL "http://ENTER_CAPABILITIES_URL_HERE"
Layer "layer_id"
TileMatrixSet "tileMatrixSet_id"
Style "style_name"
Format "image/png"
RequestTimeout 90
ReadTimeout 60
CoordSys Earth Projection 10, 157, 7, 0 Bounds(-20037508.34,-
20037508.34) (20037508.34,20037508.34)

```

См. также:

[оператор Alter Table](#), [оператор Create Index](#), [оператор Create Map](#), [оператор Drop Table](#), [оператор Export](#), [оператор Import](#), [оператор Open Table](#)

Функция CreateText()

Назначение

Возвращает текстовый объект, созданный в определенном окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CreateText( window_id, x, y, text, angle, anchor, offset )
```

window_id – целочисленный идентификатор окна карты;

x, y – координаты, задающие закрепленное положение подписи, действительные числа;

text – строка с текстом подписи, представляющим текстовый объект;

angle – угол поворота подписи в градусах, действительное число; для горизонтального текста он равен нулю;

anchor – целое число типа Integer от 0 до 8, контролирующее расположение подписи относительно места привязки. Ниже перечислены возможные значения кодов, которые описаны в MAPBASIC.DEF.

```

LAYER_INFO_LBL_POS_CC (0)
LAYER_INFO_LBL_POS_TL (1)
LAYER_INFO_LBL_POS_TC (2)
LAYER_INFO_LBL_POS_TR (3)
LAYER_INFO_LBL_POS_CL (4)
LAYER_INFO_LBL_POS_CR (5)
LAYER_INFO_LBL_POS_BL (6)

```

```
LAYER_INFO_LBL_POS_BC (7)
LAYER_INFO_LBL_POS_BR (8)
```

Двухбуквенное окончание определяет ориентацию подписи: T=Top, B=Bottom, C=Center, R=Right, L=Left. Например, для размещения подписи ниже и правее места привязки, укажите код LAYER_INFO_LBL_POS_BR, или укажите значение 8.

offset – целое число от 0 до 50, расстояние в точках от подписи до точки привязки на подписываемом объекте; *offset* игнорируется, если значение кода привязки равно 0.

Возвращаемая величина

Объект

Описание

Функция **CreateText()** возвращает величину типа Object, являющуюся текстовым объектом.

Текстовый объект использует текущий стиль текста. Для создания текстового объекта с определенным стилем, используйте **оператор Set Style** перед **CreateText()**.

В тот момент, когда текст создается, его высота контролируется размером текущего шрифта. Таким образом, после создания текстового объекта его высота зависит от размера окна карты; изменение масштаба ведет к соответственному изменению размера текста.

Возвращаемый объект имеет тип Object и сохраняется в существующей строке таблицы (используя **оператор Update**) или вставляется в новую строку таблицы (используя **оператор Insert**).

Пример

Следующий пример создает текстовый объект и помещает его на косметический слой карты (целочисленная переменная *i_map_id* содержит идентификатор окна карты).

```
Insert Into Cosmetic1 (Obj)
Values ( CreateText(i_map_id, -80, 42.4, "Sales Map", 0,0,0) )
```

См. также:

оператор AutoLabel, **оператор Create Text**, **предложение Font**, **оператор Insert**, **оператор Update**

Оператор Create Text

Назначение

Создает объект типа "Текст" (например, заголовок) для окна карты, окна **Отчёт** или классического окна отчёта. Не все предложения поддерживаются в окне **Отчёт**, смотрите раздел: *Описание*. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Create Text
[ Into { Window window_id | Variable var_name } ]
  text_string
( x1, y1 ) ( x2, y2 )
[ Font... ]
[ Label Line { Simple | Arrow } ( label_x, label_y ) Pen (pen_expr) ]

[ Spacing { 1.0 | 1.5 | 2.0 } ]
[ Justify { Left | Center | Right } ]
[ Angle text_angle ]
[ Pen .... ] [ Brush ... ] [ Priority n ] [ Name framename ]

```

window_id — идентификатор окна.

var_name — имя объектной переменной;

text_string — текст длиной до 2047 символов (многострочный текст содержит символ Chr\$(10)).

x1, *y1* — координаты одного угла прямоугольника, заполненного текстом, действительные числа;

x2, *y2* — координаты противоположного по диагонали угла прямоугольника, действительные числа; Координаты по-прежнему необходимы при создании текста в окне карты, но являются дополнительными, необязательными, когда текст создаётся в окне конструктора отчёта.

Параметром этого оператора **Font** можно задать стиль оформления текста. Размер шрифта в предложении Font игнорируется для окна карты; о размере шрифта в окне отчета сказано ниже.

label_x, *label_y* — координаты места, к которому прикреплен текст;

Pen — задает предложение Pen для указок в классическом окне отчета (но не в окне **Отчёт**).

text_angle — угол поворота текста в градусах, действительная величина.

n — целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

framename — строка, определяющая название рамки с изображением в окне конструктора отчёта. Если рамке с изображением в окне **Отчёт** присваивается имя, то оно будет сохранено в рабочем наборе. Когда параметр Name записывается в рабочем наборе, то номер версии WOR-файла меняется на 1500.

Описание

Однако, не все параметры будут работать в окнах **Отчёт**, если предложение **Label Line** пропущено. Можно пользоваться предложением Pen для сохранения стилей оформления линий в классическом окне **Отчёт**.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Параметры X и Y задаются в той координатной системе, которая была объявлена MapBasic ранее. Если система координат не была объявлена, то координаты будут принимать значения широты и долготы. Для смены координатной системы в среде MapBasic используется **оператор Set CoordSys**. Перед созданием объекта в окне отчета (в окне **Отчёт** или в классическом окне отчёта) не забудьте выполнить оператор **Set CoordSys Layout**.

Параметры *x1*, *y1*, *x2* и *y2* задают прямоугольник текста. Если текстовый объект создается в окне **Карта**, текст будет заполнять заданную текстовую область так, чтобы длина строки была равна ширине текстовой области. Размер шрифта, заданный **предложением Font**, будет проигнорирован. В окне конструктора отчёта координаты (*x2*, *y2*) являются дополнительными, необязательными. Когда они не включены, созданная текстовая рамка будет иметь размер, который позволяет уместить весь текст с использованием заданного шрифта. Когда же координаты (*x2*, *y2*) включены, то создаётся текстовая рамка заданного двумя парами координат размера. Текст может переноситься на новую строку, если размер рамки слишком мал, или может иметь дополнительные пустые поля, если размер рамки больше, чем длина строки текста.

Brush — корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение создает фон для элемента оформления.

Pen — корректное **предложение Pen**. Это предложение задает стиль рамки элемента украшения.

Пример

Когда пользователь создает подпись с указкой в окне отчета, предложение Create Text Label Line Pen срабатывает и версия рабочего набора повышается до 950:

```
!Workspace
!Version 950
```



```

!Charset WindowsLatin1
Open Table "Data\Introductory_Data\World\WORLD" As WORLD Interactive
Map From WORLD
    Position (0.0520833,0.0520833) Units "in"
    Width 6.625 Units "in" Height 4.34375 Units "in"
Set Window FrontWindow()
Set Map
    CoordSys Earth Projection 1, 104
    Center (35.204159,-25.3575215)
    Zoom 18063.92971 Units "mi"
    Preserve Zoom Display Zoom
    Distance Units "mi" Area Units "sq mi" XY Units "degree"
Set Map
    Layer 1
    Display Graphic
    Global Pen (1,2,0) Brush (2,16777215,16777215) Symbol (35,0,12)
    Line (1,2,0) Font ("Arial",0,9,0)
    Label Line None Position Center Font ("Arial",0,9,0) Pen (1,2,0)

```

См. также:

[функция FrontWindow\(\)](#), [оператор AutoLabel](#), [функция CreateText\(\)](#), [предложение Font](#), [оператор Insert](#), [оператор Update](#), [функция WindowID\(\)](#)

Функция CurDate()

Назначение

Возвращает текущую дату. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurDate ( )
```

Возвращаемая величина

Date

Описание

Функция **Curdate()** возвращает текущее значение даты. Дата в формате ГГГГММДД, где ГГГГ – год, ММ – месяц в диапазоне от 01 (январь) до 12 (декабрь), ДД – день месяца в диапазоне от 01 до 31.

Для того чтобы преобразовать возвращаемое значение в формат локальной системы, используется [функция FormatDate\\$ \(\)](#) или [функция Str\\$ \(\)](#).

Пример

```
Dim d_today As Date
d_today = CurDate()
```

См. также:

[функции даты и времени](#), [функция CurDateTime\(\)](#), [функция CurTime\(\)](#), [функция Day\(\)](#), [функция Format\\$\(\)](#), [функция Month\(\)](#), [функция StringToDate\(\)](#), [функция Timer\(\)](#), [функция Weekday\(\)](#), [функция Year\(\)](#)

Функция CurDateTime

Назначение

Возвращает текущие значения даты и времени. Значения типа Date и Time возвращают [функция GetDate \(\)](#) и [функция GetTime \(\)](#). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurDateTime ( )
```

Возвращаемая величина

ДатаВремя

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim X as datetime
X = CurDateTime()
Print X
```

См. также:

[функция Day\(\)](#) function, [функция Format\\$\(\)](#), [функция Month\(\)](#), [функция StringToDate\(\)](#), [функция Timer\(\)](#), [функция Weekday\(\)](#), [функция Year\(\)](#), [функция CurDate\(\)](#)

Функция CurrentBorderPen()

Назначение

Возвращает текущее значение стиля линии границы региона. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentBorderPen ( )
```

Возвращаемая величина

Pen

Описание

Функция **CurrentBorderPen()** возвращает значение текущей установки стиля линии контура. MapInfo Pro изменяет текущий стиль линии к любому площадному объекту, нарисованному пользователем. Если программа MapBasic создает объект, используя **оператор Create Region** или подобный, но в нем не задано **предложение Pen**, то в объекте будет использован текущий стиль BorderPen.

Возвращаемое значение может быть присвоено переменной типа Pen, или может быть использовано как параметр внутри оператора, который использует параметр типа Pen (например, **оператор Set Map**).

Для извлечения отдельных атрибутов стиля линии (таких как цвет) используется **функция StyleAttr()**. Более подробно настройки стиля линии описаны в разделе: **Предложение Pen**.

Пример

```
Dim p_user_pen As Pen p_user_pen = CurrentBorderPen()
```

См. также:

функция CurrentPen(), **предложение Pen**, **оператор Set Style**, **функция StyleAttr()**

Функция CurrentBrush()

Назначение

Возвращает значение установленного на данный момент стиля штриховки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentBrush ( )
```

Возвращаемая величина

Штриховка

Описание

Функция **CurrentBrush()** возвращает значение текущей установки стиля штриховки. В MapInfo Pro это значение изменяется в диалоге команды **Стиль областей**. Когда вы рисуете в окне такие объекты, как эллипс, прямоугольник, сглаженный прямоугольник или регион, MapInfo Pro заполняет его заданной штриховкой. Когда программа на MapBasic создаёт такие объекты (например, используя **оператор Create Region** или подобный), но не задано **предложение Brush**, объект заштриховывается в соответствии с текущей установкой стиля штриховки в MapInfo.

Величина, возвращаемая **функцией CurrentBrush()** может быть присвоена переменной типа Brush или использована как параметр в операторах, в которых используется установка стиля штриха (таких, как **оператор Set Map** или **оператор Shade**).

Для вывода отдельных характеристик стиля штриха (например, цвета) используется **функция StyleAttr()**.

Более подробно о стиле можно прочитать в разделе: **Предложение Brush**.

Пример

```
Dim b_current_fill As Brush  
b_current_fill = CurrentBrush()
```

См. также:

предложение Brush, **функция MakeBrush()**, **оператор Set Style**, **функция StyleAttr()**

Функция CurrentFont()

Назначение

Возвращает значение шрифта, используемого в данный момент в окне карты или отчета. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentFont ( )
```

Возвращаемая величина

Шрифт

Описание

Функция **CurrentFont()** возвращает значение текущей установки стиля шрифта. В MapInfo Pro это значение изменяется в диалоге команды **Стиль текста** в окне карты или отчета. MapInfo Pro присвоит значение шрифта всем текстовым объектам, которые будут нанесены на карту. Когда программа на MapBasic создает такие объекты (например, используя **оператор Create Text** , но не включает **предложение Font**, объект заштриховывается в соответствии с текущей установкой стиля текста.

Величина, возвращаемая функцией **CurrentFont()**, может быть присвоена переменной типа Brush или использована как параметр в операторах, в которых используется установка стиля штриховки (таких, как **оператор Set Legend**).

Для вывода отдельных характеристик стиля шрифта (например, цвета) используется **функция StyleAttr()**.

Более подробно о стиле можно прочитать в разделе: **Предложение Font**.

Пример

```
Dim f_user_text As Font  
f_user_text = CurrentFont()
```

См. также:

предложение Font, **функция MakeFont()**, **оператор Set Style**, **функция StyleAttr()**

Функция CurrentLinePen()

Назначение

Возвращает значение шрифта, используемого в данный момент в окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentLinePen ( )
```

Возвращаемая величина

Pen

Описание

Функция **CurrentLinePen()** возвращает значение текущей установки стиля шрифта. MapInfo Pro присваивает текущий стиль линии любой линии или полилинии, нарисованной пользователем. Если программа MapBasic создает объект, используя **оператор Create Line**, или подобный ему, но при этом сам оператор не включает в себя параметр Pen, то объект будет использовать текущий стиль линии. Возвращаемое значение может быть присвоено переменной типа Pen, или может быть использовано как параметр внутри оператора, который использует параметр типа Pen (например, **оператор Set Map**).

Для извлечения отдельных атрибутов стиля линии (таких как цвет) используется **функция StyleAttr()**. Более подробно настройки стиля линии описаны в разделе: **Предложение Pen**.

Пример

```
Dim p_user_pen As Pen p_user_pen = CurrentPen()
```

См. также:

функция CurrentBorderPen(), **предложение Pen**, **оператор Set Style**, **функция StyleAttr()**

Функция CurrentPen()

Назначение

Возвращает текущий стиль линии Pen. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentPen ( )
```

Возвращаемая величина

Pen

Описание

Функция **CurrentPen()** возвращает значение текущей установки стиля линии. MapInfo Pro присваивает текущий стиль линии любой линии или полилинии, нарисованной пользователем. Если программа MapBasic создает объект, используя **оператор Create Line** или подобный ему, но при этом сам оператор не включает в себя параметр Pen, то объект будет использовать текущий стиль линии. Если нужно применить текущий стиль линии, не переустанавливая стиль границы региона, то используется **функция CurrentLinePen()**.

Возвращаемое значение может быть присвоено переменной Pen, или может быть использовано как параметр внутри оператора, который использует настройку стиля линии (например, **оператор Set Map**).

Для извлечения отдельных атрибутов стиля линии (таких как цвет) используется **функция StyleAttr()**. Более подробно настройки стиля линии описаны в разделе: **Предложение Pen**.

Пример

```
Dim p_user_pen As Pen  
p_user_pen = CurrentPen()
```

См. также:

функция MakePen(), **предложение Pen**, **оператор Set Style**, **функция StyleAttr()**

Функция CurrentSymbol()

Назначение

Возвращает значение установленного на данный момент стиля символа для точечного объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurrentSymbol ( )
```

Возвращаемая величина

Символ

Описание

Функция **CurrentSymbol()** возвращает значение текущей установки стиля символа для точечного объекта. В MapInfo Pro это значение может изменяться в диалоге команды **Стиль символов**. Когда вы рисуете в окне точечный объект, MapInfo Pro обозначает его установленным символом. Когда программа MapBasic создает такой объект, используя **оператор Create Point**, но не использует параметр Symbol, объект изображается в соответствии с текущей установкой стиля символа в MapInfo.

Величина, возвращаемая **функцией CurrentSymbol()** может быть присвоена переменной типа Symbol или использована как в операторах, в которых используется установка стиля символа (таких, как **оператор Set Map** или **оператор Shade**).

Для вывода отдельных характеристик стиля символа (например, цвета) используется **функция StyleAttr()**. Более подробно о стиле можно прочитать в разделе: **Предложение Symbol**.

Пример

```
Dim sym_user_symbol As Symbol  
sym_user_symbol = CurrentSymbol()
```

См. также:

функция MakeSymbol(), **оператор Set Style**, **функция StyleAttr()**, **предложение Symbol**

Функция CurrTime()

Назначение

Возвращает текущее значение времени: часы, минуты, секунды и миллисекунды. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
CurTime( )
```

Возвращаемая величина

Время

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim Y as time  
Y = CurTime()  
Print Y
```

См. также:

[функция Day\(\)](#) function, [функция Format\\$\(\)](#), [функция Month\(\)](#), [функция StringToDate\(\)](#), [функция Timer\(\)](#), [функция Weekday\(\)](#), [функция Year\(\)](#), [функция CurDate\(\)](#), [функция CurDateTime\(\)](#)

Функция DateWindow()

Назначение

Возвращает текущую дату, установленную как целое в диапазоне от 0 до 99, или (-1) если режим управления окном даты отключен. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
DateWindow( context )
```

context – короткое целое (SmallInt), либо DATE_WIN_CURPROG (2), либо DATE_WIN_SESSION (1).

Описание

Поведение функции зависит от того, какой ей передается контекст (*context*). Если контекст это DATE_WIN_SESSION (1), то возвращаются текущие настройки компьютера. Если контекст это DATE_WIN_CURPROG (2), то возвращаются локальные настройки MapBasic. Если контекст пропущен, то возвращаются настройки текущего сеанса.

Пример

В следующем примере переменная *Date1* = 19990120, *Date2* = 20141203 и *MyYear* = 2014.

```
DIM Date1, Date2 as Date
DIM MyYear As Integer
Set Format Date "US"
Set Date Window 75
Date1 = StringToDate("1/20/99")
Date2 = StringToDate("12/3/14")
MyYear = Year("12/30/14")
```

См. также:

[оператор Set Date Window](#)

Функция Day

Назначение

Возвращает номер дня (от 1 до 31) из даты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Day( date_expr )
```

date_expr – выражение, в результате которого получается дата.

Возвращаемая величина

Короткое целое – SmallInt от 1 до 31.

Описание

Функция **Day()** возвращает целочисленное значение от одного до тридцати одного, являющееся номером дня в месяце. Например, для даты 12/17/92 функция **Day()** будет возвращать 17.

Пример

```
Dim day_var As SmallInt, date_var As Date
date_var = StringToDate("05/23/2014")
day_var = Day(date_var)
```

См. также:

[функция CurDate\(\)](#), [функция Day\(\)](#), [функция Minute\(\)](#), [функция Month\(\)](#), [функция Second\(\)](#), [функция Timer\(\)](#), [функция Weekday\(\)](#), [функция Year\(\)](#)

Оператор DDEExecute

Назначение

Посылает по каналу DDE-связи (динамического обмена данными) в другую программу выполняемую команду.

Синтаксис

```
DDEExecute channel, command
```

channel – номер открытого канала DDE-связи, целое число, возвращаемое функцией DDEInitiate();

command – команда, посылаемая для выполнения в другую программу (DDE-сервер), строковая величина.

Описание

Оператор **DDEExecute** посылает исполняемую команду в другую программу, присоединенную к каналу DDE-связи.

Номер канала задает **функция DDEInitiate()**, которая открывает канал динамического обмена данными.

Параметр `command` должен представлять команду, воспринимаемую DDE-сервером (пассивной стороной связи). Правильный формат команды зависит от правил, принятых в подсоединенной программе. Вы должны изучить документацию другой программы, прежде чем налаживать DDE-связь.

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

ERR_NO_RESPONSE_FROM_APP (697), если программа-сервер не отвечает.

Пример

Программными средствами MapBasic Вы можете открыть канал DDE-связи с Microsoft Excel и обращаться к нему как к программе-серверу. Если канал был открыт для объекта (topic) "System", можно с помощью оператора **DDEExecute** выполнить как команду, так и макрофункцию Excel. Чтобы Excel выполнил макрофункцию, ее нужно заключить в квадратные скобки. В следующем примере открывается рабочая таблица "TRIAL.XLS" в программе Excel:

```
Dim i_chan As Integer
i_chan = DDEInitiate("Excel", "System")
DDEExecute i_chan, "[OPEN(""C:\DATA\TRIAL.XLS"")] "
```

См. также:

функция DDEInitiate() function, оператор DDEPoke, функция DDERequest\$()

Функция DDEInitiate()

Назначение

Открывает новый канал DDE-связи и возвращает его номер.

Синтаксис

```
DDEInitiate( appl_name, topic_name )
```

appl_name – имя подсоединяемой программы (например, "MapInfo"), строковая величина;

topic_name – имя документа или некоторого объекта (topic) программы (например, "System"), строковая величина.

Возвращаемая величина

Целое – Integer

Описание

Функция **DDEInitiate()** иницирует один канал DDE-связи (динамического обмена данными), назначая ему номер.

Этот канал связи позволяет соединить две программы, работающие в среде Microsoft Windows, для пересылки информации. Как только канал открыт, MapBasic может читать информацию из документа другой программы (**функция DDERequest\$()**) и записывать в этот документ (**оператор DDEPoke**). После информационного обмена каналы связи рекомендуется закрыть, используя **оператор DDETerminate** или **оператор DDETerminate All**.

Примечание: DDE-связь возможна только в среде Microsoft Windows. Если Ваше приложение будет обращаться к услугам DDE в среде другой вычислительной платформы, то MapBasic выдаст ошибку. Для обхода такого рода конфликта используется **функция SystemInfo()**.

Параметр *appl_name* задает имя программы, которое понятно для DDE-связи. Например, для установления связи с Microsoft Excel, присвойте параметру *appl_name* значение "Excel". Windows-программа, с которой связывается MapBasic, заданная в параметре *appl_name*, должна быть уже загружена; для этого в программе MapBasic используется **оператор Run Program**. Не все программы Windows поддерживают DDE-связь. Для получения информации о поддержке DDE смотрите документацию этих программ.

Параметр *topic_name* задает некоторый объект в программе, с которым возможен обмен информацией. Каждая программа обладает некоторым набором таких объектов. О списке объектов, поддерживаемых определенной программой, Вы можете прочитать в соответствующей документации. Для многих программ именем объекта, обменивающегося информацией, является имя файла документа. Например, пусть "ORDERS.XLS" – имя файла рабочей таблицы программы Microsoft Excel, тогда оператор

```
Dim i_chan As Integer
i_chan = DDEInitiate("Excel", "C:\ORDERS.XLS")
```

открывает канал связи с ним.

Многие программы поддерживают специальный объект "System". Если начать сеанс связи с объектом "System", то **функция DDERequest\$()** поможет получить список строк, представляющих допустимые объекты для связи (например, список открытых файлов). Получив список открытых файлов, можно начать новый сеанс DDE-связи с любым документом. Смотрите пример ниже.

В следующей таблице приводится список некоторых программ и их объектов для использования в функции **DDEInitiate()**.

Вызов DDEInitiate()	Что происходит при DDE-обмене
<code>DDEInitiate("Excel", "System")</code>	Функция DDERequest\$() может извлечь системную информацию Excel, такую, как список рабочих таблиц, загруженных сейчас в среде Excel, а оператор DDEExecute может послать команду на выполнение в Excel.
<code>DDEInitiate("Excel", <i>wks</i>)</code>	Если <i>wks</i> – это имя документа Excel (например "Sheet1" или "May.xls"), то следующий оператор DDEPoke может поместить какую-нибудь величину в таблицу Excel и функция DDERequest\$() может прочитать информацию из нее.
<code>DDEInitiate("MapInfo", "System")</code>	Функция DDERequest\$() может извлечь информацию, как список выполняемых сейчас в MapInfo Pro программ MapBasic.
<code>DDEInitiate("MapInfo" <i>mbx</i>)</code>	Если <i>mbx</i> – это имя программы на MapBasic, которая сейчас выполняется (например, "C:\GRIDS.MBX"), то следующий оператор DDEPoke может назначить какую-нибудь величину глобальной переменной в этой программе, а функция DDERequest\$() может прочитать текущее значение такой переменной.

Когда программа MapBasic выполняет функцию **DDEInitiate()**, то MapBasic выступает как "клиент" в DDE-связи. Другие программы Windows выступают как программы-"серверы". В пределах одной связи клиент всегда активен, а сервер только отвечает на запросы клиента. MapBasic может поддерживать одновременно сразу столько каналов, сколько могут позволить ресурсы памяти и системы Вашего компьютера. Прикладная программа одновременно может быть как клиентом в сеансе одной связи, выполняя такие операторы как **DDEInitiate()**, так и сервером для другого сеанса связи (когда определена стандартная **процедура RemoteMsgHandler**).

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

ERR_INVALID_CHANNEL (696), если неправильно задан номер канала.

Пример

Следующий фрагмент текста программы иллюстрирует DDE-связь с Microsoft Excel версий 4 и выше. Целью является передача простого сообщения (вроде "Привет от MapInfo!") в первую ячейку первой рабочей таблицы, но только если она пуста. Если ячейка не пуста, ничего записываться не будет, содержимое ячейки будет показано пользователю MapInfo.

```
Dim chan_num, tab_marker As Integer
Dim topiclist, topicname, cell As String

chan_num = DDEInitiate("EXCEL", "System")
If chan_num = 0 Then
    Note "Excel is not responding to DDE conversation."
    End Program
End If

' Get a list of Excel's valid topics
topiclist = DDERequest$(chan_num, "topics")

' If Excel 4 is running, topiclist might look like:
' ": Sheet1 System"
' (if spreadsheet is still "unnamed"), or like:
' ": C:Orders.XLS Sheet1 System"
'
' If Excel 5 is running, topiclist might look like:
' "[Book1]Sheet1 [Book2]Sheet2 ..."
'
' Next, extract just the first topic (for example, "Sheet1")
' by extracting the text between the 1st & 2nd tabs;
' or, in the case of Excel 5, by extracting the text
' that appears before the first tab.

If Left$(topiclist, 1) = ":" Then
    ' ...then it's Excel 4.
    tab_marker = Instr(3, topiclist, Chr$(9) )
    If tab_marker = 0 Then
        Note "No Excel documents in use! Stopping."
        End Program
    End If
    topicname = Mid$(topiclist, 3, tab_marker - 3)
Else
    ' ... assume it's Excel 5.
    tab_marker = Instr(1, topiclist, Chr$(9) )
    topicname = Left$( topiclist, tab_marker - 1)
End If

' open a channel to the specific document
' (e.g., "Sheet1")
DDETerminate chan_num
chan_num = DDEInitiate("Excel", topicname)
If chan_num = 0 Then
    Note "Problem communicating with " + topicname End Program
```

```

End If

' Let's examine the 1st cell in Excel.
' If cell is blank, put a message in the cell.
' If cell isn't blank, don't alter it -
' just display cell contents in a MapBasic NOTE.
' Note that a "Blank cell" gets returned as a
' carriage-return line-feed sequence:
' Chr$(13) + Chr$(10).
cell = DDERequest$( chan_num, "R1C1" )
If cell <> Chr$(13) + Chr$(10) Then
  Note
    "Message not sent; cell already contains:" + cell
Else
  DDEPoke chan_num, "R1C1", "Hello from MapInfo!"
  Note "Message sent to Excel,"+topicname+ ",R1C1."
End If
DDETerminateAll

```

Примечание: Приведенный пример не гарантирует от ошибок. Например, в Excel может в момент вызова быть активно окно графика ("Chart1" или подобные ему); тогда обращение к ячейке R1C1 будет ошибочным. Кроме того, если Вы работаете с русской версией Excel, то обращаться нужно не к R1C1, а к C1K1 и т. д.

См. также:

оператор **DDEExecute**, оператор **DDEPoke**, функция **DDERequest\$()**, оператор **DDETerminate**, оператор **DDETerminate All**

Оператор DDEPoke

Назначение

Посылает данные в программу-сервер по каналу DDE-связи.

Синтаксис

```
DDEPoke channel, itemname, data
```

channel – номер открытого канала DDE-связи, целое число; его возвращает **функция DDEInitiate()**.

itemname – имя элемента объекта, документа программы-сервера, строковая величина;

data – символьная строка, посылаемая в itemname.

Описание

Оператор **DDEPoke** посылает данные по каналу DDE-связи в объект программы-сервера.

Параметр `channel` – это целое число, которое возвращает [функция DDEInitiate\(\)](#).

Параметр `itemname` задает элемент, который соответствует определенному каналу. Каждая программа, поддерживающая DDE-связь, обладает некоторым набором объектов. О списке объектов, поддерживаемых определенной программой, Вы можете прочитать в соответствующей документации.

Для DDE-связи с документом программы Excel именем элемента является строка, такая как "R1C1" (или "C1K1" в русской версии Excel) для ячейки в первой строке и первой колонке. Именем элемента может быть также имя глобальной переменной из другой программы, если с этой программой открыт канал связи.

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

ERR_INVALID_CHANNEL (696), если неправильно задан номер канала.

Пример

В этом примере посылается сообщение ("Привет от MapInfo!") в первую ячейку рабочей таблицы.

```
Dim i_chan_num As Integer
i_chan_num = DDEInitiate("EXCEL", "Sheet1")
DDEPoke i_chan_num, "R1C1", "Hello from MapInfo!"
```

Этот пример выполняется при условии, что Excel уже загружен и рабочая таблица (Sheet1) открыта и находится в режиме редактирования ячеек. Оператор **DDEPoke** используется для изменения глобальной переменной Address.

```
i_chan_num = DDEInitiate("MapInfo", "C:\DISPATCH.MBX")
DDEPoke i_chan_num, "Address", "23 Main St."
```

См. также:

[оператор DDEExecute](#), [функция DDEInitiate\(\)](#), [функция DDERequest\\$\(\)](#)

Функция DDERequest()

Назначение

Возвращает данные, запрошенные через канал DDE-связи.

Синтаксис

```
DDERequest$( channel, itemname )
```

channel – номер открытого канала DDE-связи, целое число; его возвращает **функция DDEInitiate()**.

itemname – имя объекта программы-сервера, который будет возвращать информацию, строковая величина.

Возвращаемая величина

Строка – String

Описание

Функция **DDERequest\$()** возвращает информацию через канал DDE-связи. Если информация недоступна, **DDERequest\$()** возвращает пустое значение ("").

Параметр *channel* – это целое число, которое возвращает **функция DDEInitiate()**.

Параметр *itemname* задает элемент, который соответствует определенному каналу. Каждая программа, поддерживающая DDE-связь, обладает некоторым набором объектов. О списке объектов, поддерживаемых определенной программой, Вы можете прочитать в соответствующей документации.

В следующей таблице в первой колонке содержатся значения, которые были использованы функцией DDEInitiate() при открытии канала связи с Excel, как сервером, а во второй колонке соответствующие значения для параметра *itemname* в функции DDERequest\$().

Имя объекта обмена	Значения itemname и результат функции
"System"	"Systems" для получения списка имен элементов, которые по каналу обмена для "System"; "Topics" для получения списка объектов для DDE-связи с Excel, включая имена всех открытых рабочих таблиц; "Formats" для получения списка форматов для системного буфера (Clipboard), поддерживаемых Excel (например, "TEXT, BITMAP ...").
<i>wks</i> (имя электронной таблицы)	"RnCn" для получения строки с содержимым ячейки (здесь первый символ n должен быть номером строки, а второй – номером колонки).

Примечание: С помощью функции **DDERequest\$()** можно считывать значения глобальных переменных одной программы в другую, выполняющуюся в то же время. Следующая таблица приводит комбинации значений itemname и объектов обмена, с которыми открыта связь, для MapInfo Pro как сервера.

Имя объекта обмена	Значения имени элемента и результат функции DDERequest\$()
"System"	"Systems" для получения списка имен элементов, которые по каналу обмена для "System"; "Topics" для получения списка имен объектов для DDE-связи с MapInfo Pro, включая имена выполняемых в данный момент прикладных программ; "Formats" для получения списка форматов для системного буфера (Clipboard), поддерживаемых MapInfo Pro (например, "TEXT"); "Version" для получения числа, по которому можно определить версию MapInfo (например, "300" означает MapInfo 3.0) .
<i>mbx</i> (имя MBX-программы)	"{items}" для получения списка имен глобальных переменных выполняющейся программы; задание имени глобальной переменной позволяет функции DDERequest\$() вернуть ее значение

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

ERR_INVALID_CHANNEL (696), если неправильно задан номер канала;

ERR_CANT_INITIATE_LINK (698), если MapBasic не связан с этим объектом обмена.

Пример

В этом примере функция **DDERequest\$()** читает содержимое первой ячейки рабочей таблицы. Пример работает, если Excel уже загружен.

```
Dim i_chan_num As Integer
Dim s_cell As String
i_chan_num = DDEInitiate("EXCEL", "Sheet1")
s_cell = DDERequest$(i_chan_num, "R1C1")
```

Следующий пример подразумевает, что уже действует другая MapBasic-программа под названием "Dispatch", в которой определена глобальная переменная "Address". С помощью функции **DDERequest\$()** мы можем получить значение этой переменной.

```
Dim i_chan_num As Integer, s_addr_copy As String
i_chan_num = DDEInitiate("MapInfo", "C:\DISPATCH.MBX")
s_addr_copy = DDERequest$(i_chan_num, "Address")
```

См. также:

[функция DDEInitiate\(\)](#)

Оператор DDETerminate

Назначение

Закрывает один сеанс DDE-обмена.

Синтаксис

```
DDETerminate channel
```

channel – номер открытого канала DDE-связи, целое число; его возвращает [функция DDEInitiate\(\)](#).

Описание

Оператор **DDETerminate** закрывает канал динамического обмена данными.

Номер канала возвращает [функция DDEInitiate\(\)](#) при ее открытии. Проведя необходимые действия по обмену, необходимо завершить его закрытием канала, используя либо **оператор DDETerminate**, либо [оператор DDETerminate All](#).

Примечание: Каждая из прикладных программ может открывать по одному или несколько DDE-каналов независимо от других программ. Каждая из программ, однако,

может закрывать только свои каналы. Программа MapBasic не может закрыть DDE-канал, открытой другой программой MapBasic.

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

ERR_INVALID_CHANNEL (696), если неправильно задан номер канала.

Пример

```
DDETerminate i_chan_num
```

См. также:

функция **DDEInitiate()**, оператор **DDETerminate All**

Оператор DDETerminateAll

Назначение

Закрывает все сеансы DDE-обмена, открытые программой MapBasic.

Синтаксис

```
DDETerminateAll
```

Описание

Оператор **DDETerminateAll** закрывает все каналы DDE-связи, которые были открыты "клиентом", программой MapBasic. Программы MapBasic могут быть запущены одновременно, и в каждой может быть открыто множество каналов. Каждая из программ, однако, может закрывать только свои каналы. Программа MapBasic не может закрыть DDE-канал, открытой другой программой MapBasic.

Проделав необходимые действия по обмену, необходимо завершить его закрытием канала, используя **оператор DDETerminate** или **DDETerminateAll**.

Ошибки

ERR_CMD_NOT_SUPPORTED (642), если программа выполняется не в среде Windows.

См. также:

функция **DDEInitiate()**, оператор **DDETerminate**

Оператор Declare Function

Назначение

Объявляет имя функции и список ее параметров.

Предупреждение!

Этот оператор не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Вызов внешних функций (во втором варианте синтаксиса) зависит от вычислительной платформы. К DLL-библиотекам могут обращаться только Windows-программы.

Синтаксис 1

```
Declare Function fname
( [ [ ByVal ] parameter [ , parameter... ] As var_type ]
  [ , [ ByVal ] parameter [ , parameter... ] As var_type... ] )
As return_type
```

fname — имя функции;

parameter — имя параметра функции.

var_type — стандартный для MapBasic тип данных для параметра или определенный с помощью оператора Type;

return_type — стандартный для MapInfo скалярный тип для величины, полученной в результате.

Синтаксис 2 (внешние рутинны в Windows DLL)

```
Declare Function fname
Lib "file_name" [WideChars] [ Alias "function_alias" ]
( [ [ ByVal ] parameter [ , parameter... ] As var_type ]
  [ , [ ByVal ] parameter [ , parameter... ] As var_type... ] )
As return_type
```

fname — имя вызываемой функции;

file_name — имя DLL-файла;

WideChars — является флагом, который, когда задан, обозначает, что строка параметров и тип возвращаемой из окон внешних DLL строк должны интерпретироваться как длинные строки, иначе, типом возвращаемой строки считается MultiByte в текущем системном наборе символов.

function_alias — настоящее имя внешней функции;

parameter — имя параметра функции.

var_type — стандартный для MapBasic тип данных для параметра или определенный с помощью оператора Type;

return_type — стандартный для MapInfo скалярный тип для величины, полученной в результате.

Описание

Оператор **Declare Function** объявляет об использовании функции, написанной на языке MapBasic, или внешней функции, а также объявляет типы параметров и результата функции.

Для написания функции на MapBasic используется **оператор Function...End Function**. Каждая такая функция должна быть объявлена оператором **Declare Function**. Для более подробной информации о создании пользовательских функций, см. раздел **Оператор Function...End Function**.

Параметры могут пересылать значения функции двумя способами: ссылкой ("by-reference") или значением ("by-value", явно задавая ключевое слово **ByVal**). О различии в природе этих двух типов параметров можно прочитать в *Руководстве пользователя MapBasic*.

Обращение к внешней функции

Второй вариант синтаксиса используется для объявления внешней функции оператором **Declare Function**. Внешняя функция может быть написана на другом языке (например, C или Pascal) и может храниться в отдельном файле. Если программа использует внешнюю функцию, то она должна быть объявлена так же, как и внутренняя функция программы MapBasic.

Если оператор **Declare Function** объявляет внешнюю функцию, то параметр *file_name* должен задавать имя файла, содержащего его. Файл с внешними процедурами должен быть доступен во время выполнения программ.

Каждая внешняя функция имеет имя, которое ей дает автор при создании. Обычно, при задании функции оператором **Declare Function** параметр *fname* явно задает имя функции, извлекаемой из внешнего файла. Однако, в оператор **Declare Function** можно включить предложение **Alias** и с его помощью обращаться к внешней функции по произвольному имени. Обычно предложение **Alias** используется, если Вы хотите избежать конфликта с уже имеющимся именем стандартной функции MapBasic.

Если оператор **Declare Function** содержит предложение **WideChars**, то все строковые параметры и возвращаемые функциями окон, которые управляются внешними DLL, считаются длинными строками. Используйте только широкий формат строки для параметров, когда включается предложение WideChars.

Если оператор **Declare Function** включает предложение **Alias**, то параметр *function_alias* должен совпадать с настоящим именем функции, а параметр *fname* должен задавать имя, по которому к ней обращается MapBasic.

Ограничения для параметров внешней функции Windows DLL

Можно использовать разные типы значений для передачи в DLL. Специально настроенный тип может быть определён с помощью дополнительного, необязательного ключевого слова **Align[n]** в операторе **Type**, чтобы соответствовать "структуре упаковки" целевой DLL. Информация о переменных в разделе: **Оператор Type**. См. подробности в *Руководстве пользователя MapBasic*.

Пример

В этом примере определяется функция пользователя под названием "CubeRoot", которая вычисляет кубический корень из числа.

```
Declare Sub Main
Declare Function CubeRoot(ByVal x As Float) As Float
Sub Main
    Note Str$( CubeRoot(23) )
End Sub
Function CubeRoot(ByVal x As Float) As Float
    CubeRoot = x ^ (1 / 3)
End Function
```

См. также:

оператор Declare Sub, оператор Function...End Function

Оператор Declare Method

Назначение

Задаёт имя и список аргументов методу или функции в файле сборки .Net, чтобы их можно было вызывать из среды MapBasic.

Ограничения!

Этот оператор не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Declare Method fname Class "class_name" Lib "assembly_name"
[ Alias function_alias ]
( [ [ ByVal ] parameter [ , parameter... ] As var_type ]
  [ , [ ByVal ] parameter [ , parameter... ] As var_type... ] )
[ As return_type ]
```

fname – имя, по которому функцию можно вызывать; если опущено предложение Alias, fname должен быть настоящим именем метода/функции .Net. Это имя не должно быть длиннее 31 символа.

class_name – имя класса .Net, из которого будет вызываться функция, с упоминанием пространства имен (например, System.Windows.Forms.MessageBox).

assembly_name – имя файла сборки .Net, например *filename.dll*. Если этот файл загружается из глобального кеша (GAC), параметр *assembly_name* должен быть полностью определен по правилам использования GAC.

function_alias – оригинальное имя метода/функции .Net (такое, каким оно задано в файле сборки .Net). Внимание: предложение Alias используется только тогда, когда может иметь место конфликт имен при вызове метода.

parameter – имя параметра функции.

var_type – тип данных MapBasic, например, Integer.

return_type – стандартный для MapBasic скалярный тип для величины, полученной в результате, например Integer. Если опущено предложение As, программа MapBasic вызывает метод, как Sub-программу (используя **оператор Call**).

Описание

Оператор **Declare Method** позволяет программе MapBasic вызывать метод (функцию или процедуру) из файла сборки .Net. Файл сборки .Net можно создавать на разных языках, например на C# или VB.Net. Подробное описание вызова методов .Net из MapBasic см. в *Руководстве пользователя MapBasic..*

Программы MapBasic могут вызывать только "статические" методы или функции .Net. В языке C# такие методы принято называть "статическими" (static), в языке VB.Net их называют "общедоступными" (shared).

Во время исполнения программы аргумент *assembly_name* задает полное описание файла сборки и, если файл сборки зарегистрирован в кеше Global Assembly Cache (GAC), то MapInfo Pro загрузит файл сборки из GAC. Другой способ указания на файл сборки состоит в том, что он помещается в тот же каталог, что и MBX-файл (в этом случае, *assembly_name* задает имя этого файла, такое как "FILENAME.DLL"). Таким образом, Вы можете как зарегистрировать файл сборки в GAC, так и обойтись без этого.

Примеры

Вот пример класса на языке C#, в котором определен static-метод:

```
namespace MyProduct
{
    class MyWrapper
    {
        public static int ShowMessage(string s)
        {
            System.Windows.Forms.MessageBox.Show(s);
            return 0;
        }
    }
}
```

На языке VB.Net определение класса будет выглядеть так:

```
Namespace MyProduct
    Public Class MyWrapper
        Public Shared Function ShowMessage(ByVal s As String) As Integer
            System.Windows.Forms.MessageBox.Show(s)
            Return 0
        End Function
    End Class
End Namespace
```

В следующем примере приводится подходящий к этому случаю синтаксис MapBasic:

```
Declare Method ShowMessage
Class "MyProduct.MyWrapper"
Lib "MyAssembly.DLL" (ByVal str As String) As Integer
. . .
Dim retval As Integer
retval = ShowMessage("Here I am")
```

Следующий пример показывает, как объявляются методы из файлов сборки, зарегистрированных в глобальном кеше GAC. Обратите внимание на то, что в предложении задается полное описание файла сборки. Чтобы составить полное описание файла сборки, вы можете воспользоваться разными вспомогательными средствами, например, утилитой GACUTIL, поставляемой Microsoft в составе Visual Studio.

```
' Declare a method from the System.Windows.Forms.dll assembly:
Declare Method Show
    Class "System.Windows.Forms.MessageBox"
        Lib "System.Windows.Forms, Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=b77a5c561934e089"
        (ByVal str As String, ByVal caption As String)
' Declare a method from the mscorlib.dll assembly:
Declare Method Move
    Class "System.IO.File"
```

```

    Lib "mscorlib, Version=2.0.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089"
    (ByVal sourceFileName As String, ByVal destFileName As String)
    ' Display a .NET MessageBox dialog box with both a message and a caption:

Call Show("Table update is complete.", "Tool name")
' Call the .NET Move method to move a file
Call Move("C:\work\pending\entries.txt", "C:\work\finished\entries.txt")

```

Оператор Declare Sub

Назначение

Объявляет имя и типы параметров подпрограммы.

Ограничения

Этот оператор не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Вызов внешних функций (во втором варианте синтаксиса) зависит от вычислительной платформы. К DLL-библиотекам могут обращаться только Windows-программы.

Синтаксис 1

```

Declare Sub sub_proc
  [ ( [ ByVal ] parameter [ , parameter... ] As var_type [ , ... ] ) ]

```

где *sub_proc* — имя подпрограммы;

parameter — имя параметра из списка подпрограммы;

var_type — тип значения параметра, стандартный или результат оператора Type.

Синтаксис 2 (внешние процедуры в Windows DLL)

```

Declare Sub sub_proc Lib "file_name" [WideChars] [ Alias "sub_alias" ]
  [ ( [ ByVal ] parameter [ , parameter... ] As var_type [ , ... ] ) ]

```

sub_proc — имя внешней процедуры;

file_name — строка, имя DLL-файла;

WideChars — является флагом, который, когда задан, обозначает, что строка параметров для DLL-процедур внешних окон должна интерпретироваться как длинная строка, иначе, типом строки параметров считается MultiByte в текущем системном наборе символов.

sub_alias — оригинальное имя процедуры;

parameter — имя параметра из списка подпрограммы;

var_type — стандартный для MapBasic тип данных для параметра или определенный с помощью оператора Type;

Описание

Оператор **Declare Sub** объявляет имена процедур и списки их параметров. Обычно, каждому оператору **Declare Sub** соответствует одна Sub-процедура, которая находится ниже соответствующего оператора.

Такая процедура создается в программе, используя **оператор Sub...End Sub** и называется внутренней. Каждой такой процедуре должен предшествовать оператор **Declare Sub**. Более подробно о создании процедур см. в разделе: **Оператор Sub...End Sub**.

Для того чтобы объявить параметр значением, надо перед его именем использовать ключевое слово **ByVal**. По умолчанию параметр объявляется ссылкой.

Вызов внешних процедур

Второй вариант синтаксиса используется для объявления оператором **Declare Sub** внешней процедуры. Внешняя функция может быть написана на другом языке (например, C или Pascal) и может храниться в отдельном файле. Если программа использует внешнюю функцию, то она должна быть объявлена так же, как и внутренняя функция программы MapBasic.

Если **Declare Sub** объявляет внешнюю процедуру, то имя файла, в которой она содержится, должно быть задано в параметре *file_name*. Файл с внешними процедурами должен быть доступен во время выполнения программ.

Каждая внешняя функция имеет имя, которое ей дает автор при создании. Обычно, при задании процедуры оператором **Declare Sub** параметр *sub_proc* явно задает имя функции, извлекаемой из внешнего файла. Однако, в оператор **Declare Sub** можно включить предложение **Alias** и с его помощью обращаться к внешней процедуре по произвольному имени. Обычно предложение **Alias** используется, если требуется избежать конфликта с уже имеющимся именем стандартной процедурой MapBasic.

Если оператор **Declare Sub** содержит предложение **WideChars**, то все строки параметров к внешним процедурам, считаются длинными строками. Используйте только широкий формат строки для параметров, когда включается предложение **WideChars**.

Если оператор **Declare Sub** включает предложение **Alias**, то параметр *sub_alias* должен содержать оригинальное имя процедуры во внешнем файле, а параметр *sub_proc* должен задавать имя, которое использует для обращения к ней MapBasic. Можно использовать разные типы значений для передачи в DLL. Специально настроенный тип может быть определён с

помощью дополнительного, необязательного ключевого слова **Align[n]** в операторе **Type**, чтобы соответствовать "структуре упаковки" целевой DLL. Информация о переменных в разделе: **Оператор Type**.

Пример

```
Declare Sub Main
Declare Sub Cube(ByVal original As Float, cubed As Float)

Sub Main
  Dim x, result As Float
  Call Cube(2, result)
  ' result now contains the value: 8 (2 x 2 x 2)
  x = 1
  Call Cube(x + 2, result)
  ' result now contains the value: 27 (3 x 3 x 3)
End Sub
Sub Cube (ByVal original As Float, cubed As Float)
  '
  ' Cube the "original" parameter value, and store
  ' the result in the "cubed" parameter.
  '
  cubed = original ^ 3
End Sub
```

См. также:

оператор Call, оператор Sub...End Sub

Оператор Define

Назначение

Назначает имя для постоянной величины.

Предупреждение!

Оператор **Define** не может быть использован в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Define identifier definition
```

identifier – имя, слово не более 31 символа длиной, начинающееся с буквы или символа подчеркивания (_);

definition – значение, которое MapBasic подставит вместо каждого identifier в тексте программы.

Описание

Оператор **Define** задает идентификатор для постоянной величины. Перед компиляцией MapBasic сначала заменяет каждый идентификатор identifier на текст, который определен параметром definition. Примеры применения оператора **Define** можно увидеть в файле MAPBASIC.DEF.

Строчные и прописные символы в именах идентификаторов, заданные оператором **Define**, не различаются. То есть, если с помощью оператора **Define** задан некоторой величине идентификатор FOO, то в тексте программы можно использовать и Foo, и foo. В операторе **Define** нельзя применять для имен слова, используемые как ключевые, например, **Set** или **Create**. Список "запретных" слов приведен в разделе: [Оператор Dim](#).

Примеры

Оператор Define делает программу более понятной, т. к. можно задать осмысленные имена константам. Например, в программе используется число "пи", которое примерно равно 3.141593. Можно присвоить этой константе имя "PI" и использовать его в тексте программы. Для этого в начале программы напишите:

```
Define PI 3.141593
```

Теперь можно использовать слово "PI" вместо того, чтобы каждый раз набивать цифры 3.141593.

В операторе **Define** можно использовать кавычки. Например, в следующем операторе создается ключевое слово с определителем, включающим кавычки:

```
Define FILE_NAME "World.tab"
```

Следующий оператор **Define** входит в состав файла стандартных определений MAPBASIC.DEF. Выполнение этого **Define**-определения приводит к открытию пустого окна "Сообщение":

```
Define CLS Print Chr$(12)
```

Функция DeformatNumber()

Назначение

Очищает строку, представляющую число, от форматирующих символов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
DeformatNumber$ ( numeric_string )
```

numeric_string – строковая величина, представляющая число, такое как “12,345,678”

Возвращаемая величина

Строка – String

Описание

Функция возвращает строку, представляющую число. При этом из исходного значения удаляются разделители тысяч, если они были в значении параметра *numeric_string*. Также заменяется знак отделения десятичной части на точку, даже если в исходной строке он не точка.

Примеры

В следующем примере **функция Val()** используется для превращения строки в число. Перед тем, как выполнить **функцию Val()**, в этом примере используется **функция DeformatNumber\$()**, удаляющая запятые из строки, потому что **функция Val()** воспринимает и разделители тысяч как просто текст.

```
Dim s_number As String
Dim f_value As Float

s_number = "1,222,333.4"
s_number = DeformatNumber$(s_number)

' the variable s_number now contains the
' string: "1222333.4"

f_value = Val(s_number)

Print f_value
```

См. также:

функция FormatNumber\$(), функция Val()

Оператор Delete

Назначение

Удаляет один или более графических объектов. Или же удаляет одну или более строк из таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Delete [ Object ] From table
[ Where [ Rowid = id_number ] | expr ]
[ DropIndex ( Auto | On | Off ) ]
```

table – имя открытой таблицы;

id_number – номер строки, целое число от 1 и более.

expr – выражение для колонки или записи;

Описание

Оператор **Delete** удаляет графический объект или всю запись, соответствующую этому объекту в таблице. По умолчанию, оператор **Delete**:

- Удаляет все записи таблицы. Если в операторе указано ключевое слово **Object**, то, удалив только графические объекты, MapBasic не удалит записи, к которым эти объекты были присоединены.
- Воздействует на все записи таблицы. Для этого задается предложение **Where Rowid =**, в котором будет указано, какие строки обрабатываются (или удаляются) оператором **Delete**.
- Воздействует на все записи таблицы. Для этого задается предложение **Where expr...**, в котором будет указано, какие строки обрабатываются (или удаляются) оператором **Delete**.

Следует отметить разницу между тем, как действует оператор `Delete Object From` и **оператор Drop Map**. Оператор `Delete Object From` воздействует только на объекты или записи и не влияет на структуру таблицы. Оператор **Drop Map** меняет структуру таблицы, и может случиться так, что Вы не сможете после его применения добавить графические объекты в таблицу.

Если в предложении **DropIndex** задать состояние **On**, то во время выполнения операции обновление индексов транзакций приостанавливается и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции

удаляются или изменяются записи и таблица содержит более 1000 записей, из которых обновляется более 0.1% от общего количества записей. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Оператор Delete поддерживает условные выражения с помощью предложения *WHERE*. Если оператор содержит условие *WHERE*, то обрабатываются только записи, которые удовлетворяют условию выражения.

```
Delete Object From clients Where Status = 0
```

Если в условном выражении *WHERE* требуется использовать идентификатор RowId, применяйте такую конструкцию:

```
"4 > RowId OR 2 < RowId" instead of "RowId > 4 OR RowId < 2"
```

Примеры

В этом примере оператор **Delete** удаляет все записи из таблицы. После выполнения этой операции сама таблица не удаляется, а становится пустой, похожей на вновь созданную таблицу.

```
Open Table "clients"
Delete From clients
Table clients
```

Следующий оператор **Delete** удаляет только графический объект, присоединенный к десятой записи таблицы.

```
Open Table "clients"
Delete Object From clients Where Rowid = 10
Table clients
```

Следующий оператор **Delete** удаляет только объект со статусом = 0 (неактивный):

```
Open Table "clients"
Delete Object From clients Where Status = 0
Table clients
```

См. также:

[оператор Drop Map](#), [оператор Insert](#)

Оператор Dialog

Назначение

Создает новое диалоговое окно.

Предупреждение!

Оператор **Dialog** не может быть использован в окне **MapBasic**.

Синтаксис

```
Dialog
[ Title title ]
[ Width w ] [ Height h ] [ Position x, y ]
[ Calling handler ]
Control control_clause
[ Control control_clause... ]
```

title – строковая величина, которая помещается в строку заголовка диалогового окна;

h – задает высоту диалогового окна в специальных единицах измерения высоты диалога (высота одного символа в диалоге равна 8 единицам);

w – задает ширину диалогового окна в специальных единицах измерения ширины диалога (ширина одного символа в диалоге равна 4 единицам);

x, y – координаты верхнего левого угла диалога в пикселах относительно верхнего левого угла рабочей области окна MapInfo Pro; если не задано предложение **Position**, диалог будет расположен в середине окна.

handler – имя процедуры, которая выполняется перед выводом диалога на экран; обычно в эти процедуры помещается **оператор Alter Control** (или несколько операторов).

Каждый параметр *control_clause* может быть одной из конструкций, которая начинается одним из следующих ключевых слов:

- Кнопка
- OKButton – кнопка "ОК"
- CancelButton – кнопка "Отмена"
- EditText – окошко ввода текста
- StaticText – неизменяемый текст
- PopupMenu – "всплывающее" меню
- CheckBox – флажки
- MultiListBox – список множественного выбора

- **GroupBox** – группа
- **RadioGroup** – группа кнопок-переключателей
- **PenPicker** – диалог выбора стиля линии
- **BrushPickerr** – диалог выбора стиля штриховки
- **FontPickerr** – диалог выбора шрифта
- **SymbolPickerr** – диалог выбора символа
- **ListBox** – список

Каждому управляющему элементу посвящен отдельный раздел этого настоящего Справочника (например, элементу **CheckBox** посвящен раздел Предложение **Control CheckBox**; элементу **Picker** – раздел Предложения **Control PenPicker/BrushPicker/SymbolPicker/FontPicker**; и т.п.).

С помощью этих ключевых слов можно создавать следующие управляющие элементы диалога:

- **Button / OKButton / CancelButton** – кнопки
- **CheckBox** – флажки
- **GroupBox** – группа
- **RadioGroup** – группа кнопок-переключателей
- **EditText** – окошко ввода текста
- **StaticText** – неизменяемый текст
- **PenPicker / BrushPicker / SymbolPicker / FontPicker** – кнопки выбора стилей
- **ListBox / MultiListBox** – окошки списка
- **PopupMenu** – "всплывающее" меню

Описание

Оператор **Dialog** создает диалоговое окно произвольного вида для организации диалога программы с пользователем. Это так называемый модальный диалог; другими словами, пользователь должен закрыть диалог (например, нажав на кнопку **ОК** или **Cancel/Отмена**) для того, чтобы продолжить работу с MapInfo Pro. Более подробно о создании диалоговых окон написано в *Руководстве пользователя MapBasic*.

Все, что находится внутри диалогового окна, называется управляющими элементами диалога или элементами. В каждом диалоге должен быть задан хотя бы один управляющий элемент. Каждый управляющий элемент описывается в своем разделе Справочника (например, элементу **CheckBox** посвящен раздел Предложение **Control CheckBox**). Обычно, каждый диалог должен содержать определения элементов **OKButton** и/или **CancelButton** чтобы его можно было хотя бы закрыть.

Оператор **Dialog** позволяет создавать Вам диалоговые окна произвольного вида. Если требуется показать стандартный диалог (например, диалог, вызываемый командой **Открыть**), используйте следующие операторы или функции: функцию **Ask()**, оператор **Note**, оператор **ProgressBar**, функцию **FileOpenDlg()**, функцию **FileSaveAsDlg()** или функцию **GetSeamlessSheet()**.

Информацию об основных концепциях построения диалогового окна в MapBasic можно найти в *Руководстве пользователя MapBasic*.

Размер и расположение элементов диалога

В операторе **Dialog** размер и расположение элемента диалога измеряются в долях шрифта диалога. Каждая единица измерения ширины в диалоге равна одной четвертой ширины символа (координата X) и каждая единица измерения высоты в диалоге – одной восьмой высоты символа (координата Y). Так, если элемент имеет ширину 40 и высоту 40, то это значит, что в элемент можно уместить слово в десять букв и список в пять строк. Центром координат для определения места элементов диалога взят верхний левый угол окна диалога. Он имеет координаты (0,0).

Предложения **Position**, **Height** и **Width** не являются обязательными. Если они опущены, то MapBasic разместит элементы по умолчанию в порядке следования соответствующих предложений Control в операторе.

Заккрытие диалога

Диалоговое окно, открытое MapBasic оператором **Dialog**, можно закрыть одним из четырех способов:

- пользователь нажимает кнопку элемента управления OKButton (если этот элемент присутствует в диалоге);
- пользователь нажимает кнопку элемента управления CancelButton (если этот элемент присутствует в диалоге);
- пользователь указывает на элемент диалога, обработчик которого выполняет **оператор Dialog Remove**; или
- пользователь отменяет диалог, используя системное меню диалогового окна или клавишу **ESC**.

Чтобы диалог оставался на экране после нажатия кнопок **ОК** или **Отмена**, требуется создать специальные процедуры-обработчики нажатия этих кнопок и указать эти процедуры в описаниях элементов OKButton или CancelButton. Для достижения нужного "эффекта присутствия" в этих обработчиках должен использоваться **оператор Dialog Preserve**.

Чтение введенных значений

После оператора **Dialog** часто используется **функция CommandInfo()**, чтобы определить, как был закрыт диалог пользователем: кнопкой **ОК** или **Отмена**. Если пользователь нажал на кнопку **ОК**, то следующая функция вернет значение TRUE:

```
CommandInfo(CMD_INFO_DLG_OK)
```

Для определения, какие значения ввел пользователь в окошки диалога и какие режимы выбрал, есть два способа: в состав оператора включается **Dialog** предложение Into или используется **функция ReadControlValue()** из процедуры-обработчика.

Если используется предложение **Into**, а диалог был закрыт кнопкой ОК, MapInfo Pro присвоит финальные значения из элементов диалога соответствующим переменным.

Примечание: MapInfo Pro обновляет переменную после нажатия на **ОК**. Также MapInfo Pro обновляет переменную после закрытия диалога.

Чтобы прочитать значения из диалога, пока он не закрыт, можно только из процедуры-обработчика, используется **функция ReadControlValue()**.

Задание клавишных сокращений для элементов диалога

Если прикладная программа выполняется в среде MapInfo для Windows, то элементам диалога могут быть назначены клавишные сокращения, которые позволяют пользователю обращаться к элементам диалога с клавиатуры.

В тексте заголовка перед нужным символом надо поставить знак амперсанда (&). Символ для комбинации клавиш выбирается из заголовков и подписей элементов, которые задаются элементам в предложении **Title**. Например, следующая конструкция создает элемент диалога Button с комбинацией клавиш **Alt+R**:

```
Control Button
  Title "&Reset"
```

Знак & (амперсанд), задаваемый в предложении **Title**, на экране не отображается. Если требуется вывести сам знак амперсанда в заголовок элемента, (например, чтобы создать следующую подпись элемента "Найти & Заменить"), то используйте в предложении **Title** два амперсанда подряд:

```
Title "Find && Replace"
```

Если задаётся элемент StaticText непосредственно до или после элемента EditText, то определенная комбинация клавиш в тексте StaticText можно использовать для перехода в окошко элемента EditText.

Порядок элементов диалога для TAB-клавиши

Для изменения фокуса (точки ввода) в диалоге при помощи клавиатуры пользователь может использовать клавишу табулятора **TAB**. Фокус перемещается от элемента к элементу при каждом нажатии на TAB.

Порядок перемещения фокуса задается последовательностью, в которой соответствующие предложения **Control** следуют в операторе **Dialog**. Так, если фокус находится на третьем по порядку элементе, то нажатие на TAB перемещает фокус на следующий за ним в описании,

четвертый элемент и т.д.. Изменить этот порядок можно, переставив в описании диалога предложения **Control**.

Примеры

Следующий оператор строит диалог с окошком для ввода текста пользователем. Поскольку ни одно из предложений **Control** не использует подпредложения **Position**, MapBasic размещает элементы управления автоматически.

```
Dialog
  Title "Search"
  Control StaticText
    Title "Enter string to find:"
  Control EditText
    Value gs_searchfor 'this is a Global String variable
    Into gs_searchfor
  Control OKButton
  Control CancelButton
If CommandInfo(CMD_INFO_DLG_OK) Then
  ' ...then the user clicked OK, and the variable
  ' gs_searchfor contains the text the user entered.
End If
```

В следующем фрагменте можно найти применение элементов диалога всех типов.

```
Include "mapbasic.def"
Declare Sub reset_sub `resets dialog to default settings
Declare Sub ok_sub ' notes values when user clicks OK.
Declare Sub Main
Sub Main
  Dim s_title As String 'the title of the map
  Dim l_showlegend As Logical 'TRUE means include legend
  Dim i_details As SmallInt '1 = full details; 2 = partial
  Dim i_quarter As SmallInt '1=1st qrtr, etc.
  Dim i_scope As SmallInt '1=Town;2=County; etc.
  Dim sym_variable As Symbol

  Dialog
    Title "Map Franchise Locations"

    Control StaticText
      Title "Enter Map Title:"
      Position 5, 10

    Control EditText
      Value "New Franchises, FY 95"
      Into s_title
      ID 1
      Position 65, 8 Width 90
    Control GroupBox
      Title "Level of Detail"
```

```

    Position 5, 30 Width 70 Height 40

Control RadioGroup
    Title "&Full Details;&Partial Details"
    Value 2
    Into i_details
    ID 2
    Position 12, 42 Width 60

Control StaticText
    Title "Show Franchises As:" Position 95, 30
Control SymbolPicker
    Position 95, 45
    Into sym_variable
    ID 3

Control StaticText
    Title "Show Results For:"
    Position 5, 80
Control ListBox
    Title "First Qrtr;2nd Qrtr;3rd Qrtr;4th Qrtr"
    Value 4
    Into i_quarter
    ID 4
    Position 5, 90 Width 65 Height 35
Control StaticText
    Title "Include Map Layers:"
    Position 95, 80
Control MultiListBox
    Title "Streets;Highways;Towns;Counties;States"
    Value 3
    ID 5
    Position 95, 90 Width 65 Height 35
Control StaticText
    Title "Scope of Map:"
    Position 5, 130
Control PopupMenu
    Title "Town;County;Territory;Entire State"
    Value 2
    Into i_scope
    ID 6
    Position 5, 140
Control CheckBox
    Title "Include &Legend"
    Into l_showlegend
    ID 7
    Position 95, 140
Control Button
    Title "&Reset"
    Calling reset_sub
    Position 10, 165
Control OKButton
    Position 65, 165

```

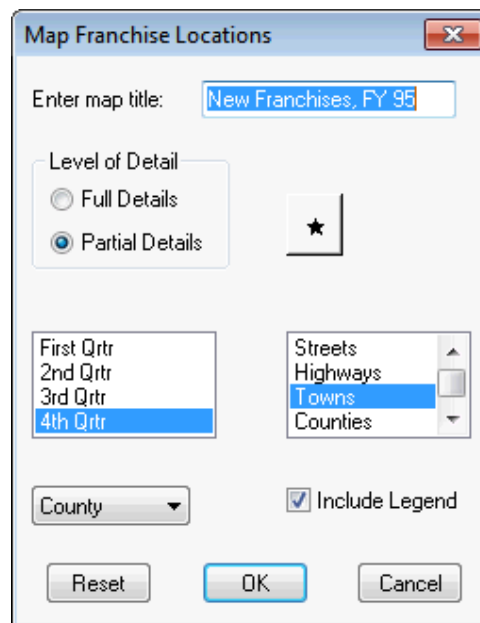
```

Calling ok_sub

Control CancelButton
Position 120, 165
If CommandInfo(CMD_INFO_DLG_OK) Then
' ... then the user clicked OK.
Else
' ... then the user clicked Cancel.
End If
End Sub
Sub reset_sub
' here, you could use Alter Control statements
' to reset the controls to their original state.
End Sub
Sub ok_sub
' Here, place code to handle user clicking OK
End Sub

```

При выполнении этого фрагмента программы будет показан следующий диалог:



См. также:

оператор **Alter Control**, функция **Ask()**, оператор **Dialog Preserve**, оператор **Dialog Remove**, функция **FileOpenDlg()**, функция **FileSaveAsDlg()**, оператор **Note**, функция **ReadControlValue()**

Оператор Dialog Preserve

Назначение

Открывает диалог снова после нажатия клавиш типа **ОК** или **Отмена**.

Синтаксис

```
Dialog Preserve
```

Предупреждение!

Этот оператор может быть использован только в процедуре обработчика кнопок типа OKButton и CancelButton. Этот оператор не может быть использован в окне **MapBasic** в среде MapInfo Pro.

Описание

Оператор **Dialog Preserve** позволяет приостановить закрытие диалога, который создает программу [оператор Dialog](#)) нажатием кнопок типа Ок или Отмена.

Оператор **Dialog Preserve** позволяет подтверждать "закрытие" диалога. Например, пользователь нажимает на кнопку **Отмена**. Обработчик кнопки вызывает диалог, спрашивающий: "Вы согласны отменить все изменения?" Если пользователь выберет кнопку "Нет", обработчик восстановит первоначальный диалог. Такого поведения можно добиться, включив оператор **Dialog Preserve** в тело процедуры-обработчика нажатия на кнопку "Нет", задаваемую элементом CancelButton.

Пример

Следующий отрывок может быть процедурой-обработчиком кнопки Отмена.

```
Sub confirm_cancel
  If Ask("Do you really want to lose your changes?",
    "Yes", "No") = FALSE Then
    Dialog Preserve
  End If
End Sub
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [оператор Dialog Remove](#), [функция ReadControlValue\(\)](#)

Оператор Dialog Remove

Назначение

Закрывает диалог, созданный оператором Dialog.

Синтаксис

```
Dialog Remove
```

Ограничения!

Оператор работает только в процедуре обработчика элемента диалога. Этот оператор не может быть использован в окне **MapBasic** в среде MapInfo Pro.

Описание

Оператор **Dialog Remove** закрывает диалог, который создал последний **оператор Dialog**. Автоматически диалоговое окно закрывается по нажатию на кнопки ОК или Отмена. Оператор **Dialog Remove**, включенный в тело процедуры-обработчика элемента диалога, удаляет диалог с экрана, не дожидаясь того, что пользователь нажмет **ОК** или **Отмена**. Это полезно, если, например, в диалоге выбрать двойным щелчком мышкой из окошка списка какую-нибудь строчку и требуется, чтобы диалог после этого закрывался автоматически.

Примечание: **Dialog Remove** посылает диалогу сигнал на закрытие после того, как процедура-обработчик закончит свою работу. (То есть диалог не закрывается немедленно после выполнения этого оператора в процедуре-обработчике).

Пример

Следующая процедура – часть программы NVIEWS.MB. Она исполняет роль обработчика спискового элемента (ListBox) диалога **Показ Карт** (Named Views). Если пользователь указывает в список, используя одно нажатие на клавишу мышки, обработчик оставляет возможность закрыть диалог при помощи кнопок. Если пользователь указывает в список, используя двойное нажатие на клавишу мышки, обработчик использует оператор **Dialog Remove** для закрытия диалогового окна.

Примечание: MapInfo Pro вызывает этот обработчик и тогда, когда клавиша мышки нажимается один раз, и тогда, когда пользователь использует двойное нажатие.

```
Sub listbox_handler
  Dim i As SmallInt
  Alter Control 2 Enable
  Alter Control 3 Enable
  If CommandInfo(CMD_INFO_DLG_DBL) = TRUE Then
    '
    ' ... then the user double-clicked.
    '
    i = ReadControlValue(1)
    Dialog Remove
    Call go_to_view(i)
  End If
End Sub
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [оператор Dialog Preserve](#), [функция ReadControlValue\(\)](#)

Dim statement

Purpose

Defines one or more variables. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Restrictions

When you issue **Dim** statements through the **MapBasic** window, you can only define one variable per **Dim** statement, although a **Dim** statement within a compiled program may define multiple variables. You cannot define array variables using the **MapBasic** window.

Syntax

```
Dim var_name [ , var_name ... ] As var_type
[ , var_name [ , var_name ... ] As var_type ... ]
```

var_name is the name of a variable to define.

var_type is a standard or custom variable Type.

Description

A **Dim** statement declares one or more variables. The following table summarizes the types of variables which you can declare through a **Dim** statement.

Таблица 4. Location of Dim Statements and Scope of Variables

Variable Type	Description
LargeInt	Whole numbers from –9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 (inclusive); stored in 8 bytes.
SmallInt	Whole numbers from -32768 to 32767 (inclusive); stored in 2 bytes.
Integer	>Whole numbers from -2,147,483,648 to +2,147,483,647 (inclusive); stored in 4 bytes.
Float	Floating point value; stored in eight-byte IEEE format.
String	Variable-length character string, up to 32767 bytes long.
String * length	Fixed-length character string (where <i>length</i> dictates the length of the string, in bytes, up to 32767 bytes); fixed-length strings are padded with trailing blanks.
Logical	TRUE or FALSE, stored in 1 byte: zero=FALSE, non-zero=TRUE.
Date	Date
DateTime	DateTime
Time	Time
Object	Graphical object (Point, Region, Line, Polyline, Arc, Rectangle, Rounded Rectangle, Ellipse, Text, or Frame).
Alias	Column name.
Pen	Pen (line) style setting.
Brush	Brush (fill) style setting.
Font	Font (text) style setting.

Variable Type	Description
Symbol	Symbol (point-marker) style setting.
IntPtr	A platform specific type that is used to represent a pointer or a handle. This helps to write applications that will work with both the 32-bit and 64-bit versions of MapInfo Pro. The 64-bit version of MapInfo Pro treats this like a LargeInt, and the 32-bit version of MapInfo Pro treats this like an Integer.
This	Represents a reference to a .NET object, You can use this to hold a .NET reference type and call method/properties on it. Simple Integer value stored in 4 bytes.
RefPtr	Represents a reference to a .NET object. You can use this to hold a .Net reference type and pass to method in .NET code. Simple integer value stored in 4 bytes.

The **Dim** statement which defines a variable must precede any other statements which use that variable. **Dim** statements usually appear at the top of a procedure or function.

If a **Dim** statement appears within a **Sub...End Sub statement** or within a **Function...End Function statement**, the statement defines variables that are local in scope. Local variables may only be accessed from within the procedure or function that contained the **Dim** statement.

If a **Dim** statement appears outside of any procedure or function definition, the statement defines variables that are module-level in scope. Module-level variables can be accessed by any procedure or function within a program module (for example, within the .MB program file).

To declare global variables (variables that can be accessed by any procedure or function in any of the modules that make up a project), use the **Global statement**.

Declaring Multiple Variables and Variable Types

A single **Dim** statement can declare two or more variables that are separated by commas. You also can define variables of different types within one **Dim** statement by grouping like variables together, and separating the different groups with a comma after the variable type:

```
Dim jointer, i_min, i_max As Integer, s_name As String
```

Array Variables

MapBasic supports one-dimensional array variables. To define an array variable, add a pair of parentheses immediately after the variable name. To specify an initial array size, include a constant integer expression between the parentheses.

The following example declares an array of ten float variables, then assigns a value to the first element in the array:

```
Dim f_stats(10) As Float
f_stats(1) = 17.23
```

The number that appears between the parentheses is known as the subscript. The first element of the array is the element with a subscript of one (as shown in the example above).

To re-size an array, use the **ReDim statement**. To determine the current size of an array, use the **UBound() function**. If the **Dim** statement does not specify an initial array size, the array will initially contain no members; in such a case, you will not be able to store any data in the array until re-sizing the array with a **ReDim statement**. A MapBasic array can have up to 32,767 items.

String Variables

A string variable can contain a text string up to 32 kilobytes in length. However, there is a limit to how long a string constant you can specify in a simple assignment statement. The following example performs a simple string variable assignment, where a constant string expression is assigned to a string variable

```
Dim status As String
status = "This is a string constant ... "
```

In this type of assignment, the constant string expression to the right of the equal sign has a maximum length of 256 characters.

MapBasic, like other BASIC languages, pads fixed-length string variables with blanks. In other words, if you define a 10-byte string variable, then assign a five-character string to that variable, the variable will actually be padded with five spaces so that it fills the space allotted. (This feature makes it easier to format text output in such a way that columns line up).

Variable-length string variables, however, are not padded in this fashion. This difference can affect comparisons of strings; you must exercise caution when comparing fixed-length and variable-length string variables. In the following program, the **If...Then statement** would determine that the two strings are not equal:

```
Dim s_var_len As String
Dim s_fixed_len As String * 10
s_var_len = "testing"
s_fixed_len = "testing"
If s_var_len = s_fixed_len Then
    Note "strings are equal" ' this won't happen
Else
    Note "strings are NOT equal" ' this WILL happen
End If
```

.NET Specific Variables

This - Represents a reference to a .NET object, You can use this to hold a .NET reference type and call method/properties on it. Simple Integer value stored in four bytes.

Example

If you have a .NET class *ClassShow* with instance method *Show()*, then in .NET

```
namespace ClassShow
{
    public class ClassShow
    {
        public void Show(){}
    }
}
ClassShow obj = new ClassShow();
obj.Show();
```

In MapBasic:

```
`Declaration of ClassShow constructor
Declare Method New_ClassShow Class "ClassShow.ClassShow" Lib
"ClassShow.dll" Alias Ctor_CreateInstance() as This
`Declaration of ClassShow Show method
Declare Method Show Class "ClassShow.ClassShow " Lib "ClassShow.dll"
Alias Show(ByVal classShowInstance as This)
Dim obj as This
obj = New_ClassShow()
Call Show(obj)
`Assigning NULL_PTR to C# reference type will destroy and allow GC to
collect them.
obj = NULL_PTR
```

Passing *obj* as the first parameter to *Show()* is similar to calling *obj.Show()*, but make sure that the first parameter is of type *This*.

RefPtr - Represents a reference to a .NET object. You can use this to hold a .NET reference type and pass it to the method in .NET code. Simple integer value stored in 4 bytes. After using this reference object, MapBasic developers should assign NULL_PTR to the object.

RefPtr is similar to *This*, just that it does not allow you call the class instance methods on the object.

Restrictions on Variable Names

You may not use a MapBasic keyword as a variable name, a table name, or column name. Restrictions on reserved words also apply when uploading a table to a DBMS, such as Oracle, PostGIS, or SQL Server, or converting to another third-party format like SHP. These systems have reserved words with restrictions on their use as table and column names. Avoid using any word for table and column names that could be a command syntax: do not declare variables with names such as, If, Then, Select, Open, Close, or Count.

Variable names are case-insensitive. Thus, if a **Dim** statement defines a variable called abc, the program may refer to that variable as abc, ABC, or Abc.

Each variable name can be up to 31 characters long, and can include letters, numbers, and the underscore character (_). Variable names can also include the punctuation marks \$, %, &, !, #, and @, but only as the final character in the name. A variable name may not begin with a number.

Many MapBasic language keywords, such as **Open**, **Close**, **Set**, and **Do**, are reserved words which may not be used as variable names. If you attempt to define a variable called **Set**, MapBasic will generate an error when you compile the program. The table below summarizes the MapBasic keywords which may not be used as variable names.

Add	Alter	Browse	
Call	Close	Create	DDE
DDEExecute	DDEPoke	DDETerminate	DDETerminateAll
Declare	Delete	Dialog	Dim
Do	Drop	Else	Elseif
End	Error	Event	Exit
Export	Fetch	Find	For
Function	Get	Global	Goto
Graph	If	Import	Insert
Layout	Map	Menu	Note
Objects	OnError	Open	Pack
Print	PrintWin	ProgressBar	Put
ReDim	Register	Reload	Remove
Rename	Resume	Rollback	Run
Save	Seek	Select	Set

Shade	StatusBar	Stop	Sub
Type	Update	While	

In some BASIC languages, you can dictate a variable's type by ending the variable with one of the punctuation marks listed above. For example, some BASIC languages assume that any variable named with a dollar sign (for example, LastName\$) is a string variable. In MapBasic, however, you must declare every variable's type explicitly, through the **Dim** statement.

Initial Values of Variables

MapBasic initializes numeric variables to a value of zero when they are defined. Variable-length string variables are initialized to an empty string, and fixed-length string variables are initialized to all spaces.

Object and style variables are not automatically initialized. You must initialize Object and style variables before making references to those variables.

Example

```
' Below is a custom Type definition, which creates
' a new data type known as Person
Type Person
  Name As String
  Age As Integer
  Phone As String
End Type

' The next Dim statement creates a Person variable
Dim customer As Person

' This Dim creates an array of Person variables:
Dim users(10) As Person

' this Dim statement defines an integer variable
' "counter", and an integer array "counters" :
Dim counter, counters(10) As Integer

' the next statement assigns the "Name" element
' of the first member of the "users" array
users(1).Name = "Chris"
```

See also:

[Global statement](#), [ReDim statement](#), [Type statement](#), [UBound\(\) function](#)

Функция Distance()

Назначение

Возвращает расстояние между двумя точками, заданными координатами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Distance ( x1, y1, x2, y2, unit_name )
```

x1 и *x2* – координаты по оси X (например, долгота).

y1 и *y2* – координаты по оси Y (например, широта).

unit_name – строка, определяющая единицы измерения расстояния (к примеру, “км.”).

Возвращаемая величина

Вещественное – Float

Описание

Функция **Distance()** вычисляет расстояние между двумя определенными точками.

Она возвращает значение в указанных единицах *unit_name*; например, чтобы получить расстояние в милях, задайте “mi” в качестве *unit_name*. Список доступных единиц измерения смотрите в разделе **оператор Set Distance Units**.

Координаты по осям X и Y должны быть заданы в текущей системе координат MapBasic. Координаты X и Y понимаются MapBasic относительно заданной системы координат. Если система координат не объявлялась, то используется система "Широта/Долгота". Можно установить систему координат по умолчанию, выполнив **оператор Set CoordSys**.

Если текущей системой координат является система координат Земли, то функция **Distance()** возвращает расстояние между двумя точками по дуге большого земного сечения. Расстояние вычисляется по большой окружности на сфере (большая окружность получается в результате сечения земного шара плоскостью, заданной этими двумя точками и центром Земного шара).

В большинстве случаев MapInfo Pro проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система - план, то выполняются декартовы вычисления.

Пример

```
Dim dist, start_x, start_y, end_x, end_y As Float
Open Table "cities"
Fetch First From cities
start_x = CentroidX(cities.obj)
start_y = CentroidY(cities.obj)
Fetch Next From cities
end_x = CentroidX(cities.obj)
end_y = CentroidY(cities.obj)
dist = Distance(start_x, start_y, end_x, end_y, "mi")
```

См. также:

функция Area(), функция ObjectLen(), оператор Set CoordSys, оператор Set Distance Units

Оператор Do Case...End Case

Назначение

Выполняет одну из групп операторов, условия для которой истинны.

Ограничения!

Оператор **Do Case** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Do Case do_expr
  Case case_expr [ , case_expr ]
    statement_list
  [ Case ... ]
  [ Case Else
    statement_list ]
End Case
```

do_expr – выражение;

case_expr – выражение, результат которого будет сравниваться со значением *do_expr*;

statement_list – группа операторов.

Описание

Оператор **Do Case** действует также, как и **оператор If...Then**. **Do Case** проверяет истинность определенных выражений и, в зависимости от результата, выполняет одну из групп операторов. Оператор **Do Case** в языке MapBasic аналогичен оператору **Select Case** в языке BASIC. (Так как в MapBasic есть другой **оператор Select**, имя оператора изменено для того, чтобы не возникало путаницы).

Выполняя оператор **Do Case**, MapBasic вычисляет выражение **Case case_expr**. Если результат **Case case_expr** равен результату вычисления выражения **do_expr**, то MapBasic выполнит группу операторов **statement_list**, которая расположена за первым словом **Case** и после этого управление программой передается оператору, следующему после ключевых слов **End Case**.

Если ни одно из выражений **Case case_expr** не дает того же результата, что **do_expr**, MapBasic переходит к следующему выражению **Case case_expr**. Так MapBasic перебирает все выражения **Case case_expr** пока либо не добьется совпадения, либо пока не переберет все.

MapBasic выполняет операторы **statement_list** из оператора **Do Case** ровно один раз или ни разу. Так, обнаружив совпадение, MapBasic выполняет операторы **statement_list** и сразу переходит на первый оператор после **End Case**.

Если же ни одно выражение **case_expr** не дает того же результата, что выражение **do_expr**, то операторы из списка **statement_list** не выполняются. Однако, в оператор **Do Case** можно включить предложение **Case Else**, и тогда если ни одно из выражений **Case case_expr** не даст совпадения с условием, то MapBasic выполнит операторы из раздела **Case Else**.

Заметим, что оператор **Do Case** такой формы:

```
Do Case expr1
  Case expr2
    statement_list1
  Case expr3, expr4
    statement_list2
  Case Else
    statement_list3
End Case
```

работает так же, как **оператор If...Then** следующей конструкции:

```
If expr1 = expr2 Then
  statement_list1
ElseIf expr1 = expr3 Or expr1 = expr4 Then
  statement_list2
Else
  statement_list3
End If
```

Пример

В примере выбираются текстовые строки: "Первый квартал", "Второй квартал" и т.п., в зависимости от текущего месяца.

```
Dim cur_month As Integer, msg As String
cur_month = Month( CurDate() )
Do Case cur_month
  Case 1, 2, 3
    msg = "First Quarter"
  Case 4, 5, 6
    msg = "Second Quarter"
  Case 7, 8, 9
    msg = "Third Quarter"
  Case Else
    msg = "Fourth Quarter"
End Case
```

См. также:

[оператор If...Then](#)

Оператор Do...Loop

Назначение

Выполняет последовательность действий по циклу до тех пор, пока выполнится (или, наоборот, не выполнится) некое условие.

Предупреждение!

Оператор **Do Loop** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Do
  statement_list
Loop [ { Until | While } condition ]
```

Синтаксис 2

```
Do [ { Until | While } condition ]
  statement_list
Loop
```

statement_list – группа операторов для одного шага цикла;

condition – условие, задающее предложение или окончание цикла.

Описание

Оператор **Do...Loop** задает цикл. Цикл **Do...Loop** повторяет группу операторов *statement_list* до тех пор, пока предложение **While** сохраняет значение TRUE (или, наоборот, цикл повторяет операторы *statement_list* пока условие **Until** не примет значение TRUE).

Если цикл **Do...Loop** не содержит слов **Until** / **While**, то цикл повторяется бесконечно. В этом случае управление выполнением цикла ведется из операторов самого цикла, используя оператор **Goto** или оператор **Exit Do**, которые прекращают цикл. Выполнив MapBasic-оператор **Exit Do** останавливает цикл **Do...Loop** немедленно, не взирая на условия, заданные предложениями **Until** / **While**, и передает управление на первый оператор после предложения **Loop**.

Предложения **Until** / **While** могут находиться как после слова **Do**, так и после слова **Loop**. В зависимости от положения слов **Until** / **While**, проверка выполнения условия происходит до или после выполнения операторов *statement_list*. Это важно для первой итерации. Следующая конструкция:

```
Do
  statement_list
Loop While condition
```

выполняет операторы *statement_list* и потом вычисляет условие condition. Если условие – condition равно TRUE, MapBasic вновь выполняет *statement_list* пока условие condition не станет равным FALSE. Так, цикл **Do...Loop** в представленном выше варианте, хотя бы раз выполняет набор операторов *statement_list*.

В следующей конструкции оператор **Do...Loop** выполнит операторы набора *statement_list*, если условие condition равно TRUE.

```
Do While condition
  statement_list
Loop
```

Пример

В цикле **Do...Loop** читаются первые десять записей таблицы:

```
Dim sum As Float, counter As Integer
Open Table "world"
Fetch First From world
counter = 1
Do
  sum = sum + world.population
  Fetch Next From world
```

```
counter = counter + 1
Loop While counter <= 10
```

См. также:

[оператор Exit Do](#), [оператор For...Next](#)

Функция DotDensityThemeInfo()

Назначение

Возвращает информацию об определённом на карте тематическом слое плотности точек. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
DotDensityThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id Идентификатор окна карты.

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический слой плотности точек карты.

attribute Код – короткое целое `SmallInt`, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	DotDensityThemeInfo() возвращает
DOTDENSITYTHEME_INFO_CONCE	1	Вещественный (<code>Float</code>) результат – число, соответствующее каждой точке при использовании метода плотности точек.
DOTDENSITYTHEME_INFO_SQUARE	2	Логический (<code>Logical</code>) результат (T or F) – T, если точки квадратные. F, если точки круглые.

код атрибута	ID	DotDensityThemeInfo() возвращает
DOTDENSITYTHEME_INFO_SIZE	3	Целочисленный (<i>Integer</i>) результат – размер одной точки в пикселах при использовании метода плотности точек.
DOTDENSITYTHEME_INFO_COLOR	4	Целочисленный (<i>Integer</i>) результат, представляющий значение RGB для цвета точек при использовании тематического метода плотности точек.

Оператор Drop Index

Назначение

Удаляет индекс из таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Drop Index table( column )
```

table – имя открытой таблицы;

column – имя колонки в этой таблице.

Описание

Оператор **Drop Index** отменяет индексирование колонки в открытой таблице. Удаление индекса освобождает место на диске, где расположена таблица. (Для создания индекса вновь используется [оператор Create Index.](#))

Примечание: MapInfo Pro не может отменить индексирование в таблице с несохраненными изменениями. Для сохранения изменений используется [оператор Commit Table.](#)

Используйте оператор **Drop Index** для удаления сразу всех графических объектов открытой таблицы, изменяя структуру таблицы так, что к ее записям уже не могут присоединяться графические объекты. Отменить результат выполнения оператора **Drop Index** с помощью команд **Восстановить таблицу** или **Отменить** невозможно. Аналогично [оператор Rollback](#) не может отменить действие оператора **Drop Index**.

Пример

Следующий код отменяет индексирование колонки "Name" в таблице WORLD:

```
Open Table "world"
Drop Index world(name)
```

См. также:

[Оператор Create Index](#)

Оператор Drop Map

Назначение

Удаляет все графические объекты из таблицы. Не может быть применен к связанным таблицам. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Drop Map table
```

table — имя открытой таблицы;

Описание

После выполнения оператора **Drop Map**, соответствующая таблица более не может быть показана в окне Карты. Все ее графические объекты пропадают.

Примечание: Оператор **Drop Map** не имеет обратного действия и не требует последующего сохранения изменений на диске. Отменить результат выполнения оператора **Drop Map** с помощью команд **Восстановить таблицу** или **Отменить** невозможно. Аналогично, MapBasic-[оператор Rollback](#) также не может отменить действие оператора **Drop Map**. Поэтому необходимо быть очень внимательными, используя оператор **Drop Map**.

После выполнения оператора **Drop Map** соответствующая таблица более не может быть показана в окне Карты; оператор **Drop Map** удаляет графические объекты навсегда и структура таблицы изменяется так, что она уже не может присоединять их. Для изменения структуры таблицы так, чтобы к ней можно было вновь присоединять графические объекты, используйте (однако, [оператор Create Map](#) не восстанавливает графические объекты, уничтоженные оператором **Drop Map**). Оператор **Drop Map** не меняет количества записей в таблице. Оператор **Drop Map** не влияет на данные в записях таблицы, которые можно выводить в окно Списка.

Если требуется удалить все графические объекты без изменения структуры таблицы, воспользуйтесь, для того чтобы добавить новые графические объекты оператором **Delete Object** вместо **Drop Map**.

Оператор **Drop Map** не работает со связанными таблицами.

Пример

```
Open Table "clients"  
Drop Map clients
```

См. также:

[оператор Create Map](#), [оператор Create Table](#), [оператор Delete](#)

Оператор Drop Table

Назначение

Удаляет таблицу полностью. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Drop Table table
```

table – имя открытой таблицы;

Описание

Оператор **Drop Table** используется для удаления таблицы с диска. Таблица при этом должна быть открыта.

Отметим, что если таблица основана на файле базы данных или электронной таблице, то оператор **Drop Table** удалит их вместе с файлами-компонентами, которые добавляет к ним MapInfo Pro. Другими словами, **Drop Table** может удалять файлы, которые не создавались непосредственно в среде MapInfo Pro.

Оператор **Drop Table** имеет непосредственный эффект и не имеет обратного действия. Отменить результат выполнения оператора **Drop Table** с помощью команд **Восстановить таблицу** или **Отменить** невозможно. Аналогично, MapBasic-[оператор Rollback](#) не восстанавливает таблицу после применения **Drop Table**. С оператором **Drop Table** нужно быть осторожным!

Примечание: Многие операции с таблицами в MapInfo помещают результат во временные таблицы (например, "Query1" или "Запрос1"). Временная таблица удаляется автоматически с закрытием сеанса работы MapInfo Pro и нет необходимости применять оператор **Drop Table** для удаления временной таблицы.

Оператор **Drop Table** также может использоваться для удаления таблиц GeoPackage. При удалении таблицы GeoPackage удаляется *.tab файл и все связанные файлы компонентов. Если геопакет GeoPackage содержит единственную таблицу, то с помощью этой команды будет удалена таблица и связанные с ней файлы, но не файл геопакета GeoPackage.

Оператор **Drop Table** нельзя применять для удаления таблиц, которые по природе своей являются представлениями ("view"). Например, таблицы формата StreetInfo (такие, как SF_STRTS) являются представлениями, то есть представляют на экране комбинацию других таблиц (SF_STRT1 и SF_STRT2). Поэтому таблицу SF_STRTS нельзя удалить оператором **Drop Table**.

Пример

```
Open Table "clients"
Drop Table clients
```

См. также:

[оператор Create Table](#), [оператор Delete](#), [оператор Kill](#)

Оператор End MapInfo

Назначение

Прекращает выполнение MapInfo Pro.

Синтаксис

```
End MapInfo [ Interactive ]
```

Описание

Оператор **End MapInfo** используется для остановки выполнения прикладной программы и завершения сеанса работы MapInfo Pro.

Программа может содержать процедуру-обработчик завершения работы, для которой в MapBasic зарезервировано имя **EndHandler**. Если программа выполняет оператор **End MapInfo**,

то MapInfo Pro автоматически выполнит перед закрытием все процедуры **EndHandler**.
Подробнее смотрите: [Процедура EndHandler](#).

Если оператор **End MapInfo** выполнен тогда, когда были открыты таблицы и в них были произведены изменения, не сохраненные в файлах, MapInfo Pro выдаст запрос о сохранении этих изменений.

Если оператор использует ключевое слово **Interactive** и если были открыты окна карт с объектами на косметическом слое или с не сохраненными тематическими объектами, то MapInfo Pro выдаст пользователю сообщения, предлагающие сохранить эти объекты. Однако, если в MapInfo Pro включен режим автоматического сохранения рабочего набора `MAPINFOW.WOR` перед закрытием, то сообщения выдаваться не будут. Если слово **Interactive** опущено, то предложения сохранить косметические и тематические объекты выдаваться не будут.

Для остановки MapBasic-программы без завершения сеанса работы MapInfo Pro, используйте [оператор End Program](#).

См. также:

[оператор End Program](#), [процедура EndHandler](#)

Оператор End Program

Назначение

Прекращает выполнение прикладной программы, но не останавливает MapInfo.

Предупреждение!

Оператор **End Program** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
End Program
```

Описание

Оператор **End Program** используется для остановки выполнения прикладной программы MapBasic. Прикладная программа на MapBasic может добавлять свои команды и даже заголовки в строку меню. Обычно, такие команды вызывают дополнительные процедуры из программы MapBasic. Такие процедуры неактивны, пока соответствующая команда не будет выбрана пользователем.

Если какая-либо процедура в рамках данной программы MapBasic выполнит оператор **End Program**, то вся программа прекращает работу, в том числе и все ее неактивные подпрограммы. Когда программа завершается с помощью этого оператора, MapInfo Pro автоматически удаляет все созданные ею команды и заголовки из меню.

Если в программе есть процедура с именем **EndHandler**, MapInfo Pro автоматически выполнит ее при остановке программы, каким бы образом эта остановка не совершалась.

См. также:

[оператор End MapInfo](#), [процедура EndHandler](#)

Процедура EndHandler

Назначение

Специальная подпрограмма, Sub-процедура, автоматически выполняющаяся при завершении выполнения программы.

Синтаксис

```
Declare Sub EndHandler
Sub EndHandler
    statement_list
End Sub
```

statement_list – список операторов.

Описание

EndHandler – зарезервированное имя для процедуры MapBasic.

Процедура **EndHandler** называется обработчиком завершения программы. Когда пользователь запускает программу, в которой есть обработчик завершения, **EndHandler** то эта процедура будет автоматически вызвана при завершении программы. Обработчик сработает как при закрытии MapInfo Pro, так и если программу останавливает [оператор End Program](#).

Примечание: Одновременно в состоянии ожидания могут находиться несколько программ. Если эти программы снабжены обработчиками завершения **EndHandler**, то при закрытии MapInfo Pro каждая из них сработает по очереди.

См. также:

[процедура RemoteMsgHandler](#), [процедура SelChangedHandler](#), [процедура ToolHandler](#), [процедура WinChangedHandler](#), [процедура WinClosedHandler](#)

Функция EOF()

Назначение

Возвращает FALSE, если MapBasic читает запись из файла, и TRUE, если пробует прочитать что-то после конца файла.

Синтаксис

```
EOF( filenum )
```

filenum – номер открытого файла, который возвращает **оператор Open File**.

Возвращаемая величина

Логическое – Logical.

Описание

Функция **EOF()** позволяет определить, достигнут ли конец файла, открытого в программе под номером *filenum*.

Если оператор **Get** попыталась прочитать запись, следующую после последней, то функция **EOF()** возвращает логическое "Да" (TRUE); иначе, если была прочитана запись файла, **EOF()** возвращает FALSE.

Функция **EOF()** работает с открытыми файлами (текстовыми или произвольной природы); если нужно отследить конец чтения из открытой таблицы MapInfo, то используется **функция EOT()**.

Пример использования функции **EOF()** можно посмотреть в тексте программы `NVIEWS.MB`.

Ошибки

В результате выполнения функции можно получить код ошибки `ERR_FILEMGR_NOTOPEN` (366), если файл не был открыт.

См. также:

функция EOT(), оператор Open File

Функция EOT()

Назначение

Возвращает FALSE, если MapBasic читает запись из таблицы, и TRUE, если программа пробует прочитать что-то после конца таблицы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
EOT( table )
```

table – имя открытой таблицы;

Возвращаемая величина

Логическое – Logical.

Описание

Функция **EOT()** Возвращается логическое "Да" (TRUE), если MapBasic попытался прочитать запись, следующую после последней, и логическое "Нет" (FALSE), если прочитана строка таблицы. Параметр table определяет имя таблицы, из которой производится чтение.

Ошибки

В результате выполнения оператора может генерироваться код ошибки RR_TABLE_NOT_FOUND (405), если не найдена данная таблица.

Пример

Здесь результат функции **EOT()** является критерием прекращения выполнения цикла. Цикл последовательно прочитывает записи из таблицы до тех пор, пока функция **EOT()** не просигнализирует о достижении конца таблицы.

```
Dim f_total As Float
Open Table "customer"
Fetch First From customer
Do While Not EOT(customer)
    f_total = f_total + customer.order
    Fetch Next From customer
Loop
```

См. также:

функция EOF(), оператор Fetch, оператор Open File, оператор Open Table

Функция EPSGToCoordSysString()

Назначение

Конвертирует строку с описанием географической системы координат (Spatial Reference System) в **предложение CoordSys**, которое может быть использовано функцией или оператором MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
EPSGToCoordSysString$( srs_string )
```

srs_string – строка, описывающая поддерживаемую географическую систему координат (SRS). Дескрипторы SRS иногда называют дескрипторами EPSG (European Petroleum Survey Group) (например, epsg:2600). Полный список кодов EPSG, используемых MapInfo Pro, можно найти в файле MAPINFOW.PRJ в каталоге, в который была установлена программа MapInfo Pro. Коды EPSG можно определить по префиксу “\p”, за которым следует число.

Описание

Функция **EPSGToCoordSysString\$()** преобразует строку SRS в значение параметра CoordSys, который можно применять в любой функции или операторе MapBasic, где этот параметр CoordSys может быть использован.

Пример

В следующем примере задается стандартная система координат – проекция "Широта/Долгота (WGS 84)" (в форме параметров выражения: Earth Projection 1, 104).

```
run command("Set Map " + EPSGToCoordSysString$("EPSG:4326"))
```

См. также:

предложение CoordSys

Функция Erase()

Назначение

Возвращает объект, полученный из другого удалением части, которая перекрывается другим объектом. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Erase( source_object, eraser_object )
```

source_object – объект любой природы, кроме точечного или текстового;

eraser_object – замкнутый объект, выполняющий роль "ластика".

Возвращаемая величина

Объект, который остается от объекта *source_object* после удаления из него части *eraser_object*.

Описание

Функция **Erase()** возвращает объект, полученный в результате удаления части объекта другим.

Объект *source_object* может быть линейным (прямой Линией, Полилинией или Дугой) или замкнутым объектом (Область, Прямоугольник, Скругленный Прямоугольник или Эллипс), но не может быть точечным или текстовым объектом. Объект *eraser_object* должен быть замкнутым. Объект, полученный в результате, унаследует стиль оформления (тип линии, штриховки и цвет) от объекта *source_object*.

Пример

```
' In this example, o1 and o2 are Object variables
' that already contain Object expressions.
If o1 Intersects o2 Then
  If o1 Entirely Within o2 Then
    Note "Cannot Erase; nothing would remain."
  Else
    o3 = Erase( o1, o2 )
  End If
Else
  Note "Cannot Erase; objects do not intersect."
End If
```

См. также:

оператор **Objects Erase**, оператор **Objects Intersect**

Функция Err()

Назначение

Возвращает числовой код, соответствующий текущей ошибке. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Err ( )
```

Возвращаемая величина

Целое – Integer

Описание

Функция **Err()** возвращает целочисленный код, соответствующий последней из происшедших ошибок.

Обычно, когда в прикладной программе генерируется ошибка, выводится сообщение о ней и выполнение программы прекращается. Если в программе есть процедура обработчика ошибки, вход в которую обеспечивает **оператор OnError**, выполнение программы передается обработчику, как только какой-нибудь оператор, находящийся после OnError, возвращает ошибку. В обработчике ошибок можно применять функцию **Err()** для определения происшедшей ошибки.

Функция **Err()** возвращает код ошибки, только если она вызвана из процедуры-обработчика. После того, как программа выполнит **оператор Resume** для возвращения из обработчика в процедуру, вызвавшую ее, код ошибки в программе обнуляется. Поэтому не имеет смысла использовать функцию **Err()** вне обработчика ошибок, т. к. функция, вызванная после обработчика, вернет ноль.

В этой книге после описаний некоторых операторов и функций следует раздел "Ошибки", в котором перечисляются коды некоторых возможных ошибок, генерируемых описываемым оператором или функцией. Но это не все возможные ошибки, а только связанные с наиболее часто встречающимися ситуациями.

Некоторые коды ошибок MapBasic генерируются только в специфических, редко встречающихся ситуациях. Например, код ERR_INVALID_CHANNEL (696) генерируется только функциями и

операторами, обслуживающими DDE-связь. Если оператор может генерировать такой "специальный" код, то он также приводится в разделе "Ошибки" для этого оператора.

Другие ошибки MapBasic, напротив, порождаются часто и многими функциями. Например, **функция Area()** и **функция ObjectInfo()** используют выражение типа Object как параметр. Этими функциями может генерироваться код ошибки ERR_FCN_OBJ_FETCH_FAILED (650), если задан параметр в форме *ttablename.obj*, тогда как текущая строка этой таблицы не имеет присоединенного объекта. Другими словами, каждая функция, обрабатывающая объекты, может породить ошибку ERR_FCN_OBJ_FETCH_FAILED (650). Так как ошибка ERR_FCN_OBJ_FETCH_FAILED (650) может случиться во многих и разнообразных ситуациях, многие функции не рапортуют о ней как о явной причине сбоя.

Аналогично, есть две ошибки – ERR_FP_MATH_LIB_DOMAIN (911) и ERR_FP_MATH_LIB_RANGE (912) – которые возникают при некорректных математических операциях. Эти ошибки могут вызываться вызовом любой из следующих функций: **функция Acos()**, **функция Asin()**, **функция Atn()**, **функция Cos()**, **функция Exp()**, **функция Log()**, **функция Sin()**, **функция Sqr()** или **функция Tan()**.

Полный список кодов ошибок MapBasic находится в файле ERRORS.DOC.

См. также:

оператор Error, **функция Error\$()**, **оператор OnError**

Оператор Error

Назначение

Генерирует ошибку определенного вида. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Error error_num
```

error_num – целочисленный код, соответствующий стандартной ошибке в MapBasic.

Описание

Оператор **Error** генерирует ошибку по заданному коду. Обычно он используется при отладке.

Если в программе до этого была определена процедура обработчика ошибок (смотрите: **Оператор OnError**), то MapBasic передает управление процедуре-обработчику ошибок. Если обработчик отсутствует в программе, то после выполнения оператора **Error** MapBasic выводит сообщение о ней и останавливает выполнение программы.

См. также:

[функция Err\(\) function](#), [функция Error\\$\(\)](#), [оператор OnError](#)

Функция Error\$()

Назначение

Возвращает сообщение о текущей ошибке. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Error$ ( )
```

Возвращаемая величина

Строка

Описание

Функция **Error\$ ()** возвращает текст сообщения о последней зафиксированной ошибке. Если ошибок не было, то функция **Error\$ ()** возвращает пустую строку.

Функция **Error\$ ()** может быть вызвана только из процедуры-обработчика ошибок. Подробнее смотрите: [Функция Err \(\)](#).

См. также:

[функция Err \(\)](#), [оператор Error](#), [оператор OnError](#)

Оператор Exit Do

Назначение

Останавливает действующий [оператор Do...Loop](#).

Предупреждение!

Оператор **Exit Do** не может быть выполнен в окне **MapBasic**.

Синтаксис

Exit Do

Описание

Оператор **Exit Do** прекращает выполнение цикла, который начал **оператор Do...Loop**. После выполнения **Exit Do**, MapBasic передает управление оператору, перед которым стоит **оператор Do...Loop**. Оператор **Exit Do** действителен только внутри **оператора Do...Loop**.

Оператор **Do...Loop** может быть вложенным, т. е. один **оператор Do...Loop** может быть вложен в другой **оператор Do...Loop**. Оператор **Exit Do** прекращает только **оператор Do...Loop**, внутри которого употребляется. Так в конструкции:

```
Do While condition1
:
  Do While condition2
  :
  If error_condition
    Exit Do
  End If
  :
  Loop
  :
  Loop
```

Оператор **Exit Do** прерывает вложенный цикл (организованный вторым оператором (**Do While condition2**), не останавливает первый цикл (**Do While condition1**).

См. также:

оператор Do...Loop, **оператор Exit For**, **оператор Exit Sub**

Оператор Exit For

Назначение

Останавливает действующий **оператор For...Next**.

Предупреждение!

Оператор **Exit For** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

Exit For

Описание

Оператор **Exit For** прекращает выполнение цикла, который начал **оператор For...Next**. После выполнения **Exit For**, MapBasic передает управление оператору, перед которым стоит **оператор For...Next**. Оператор **Exit For** действителен только внутри **оператора For...Next**.

Оператор **For...Next** может быть вложенным, т.е. один **оператор For...Next** может быть вложен в другой **оператор For...Next**. Оператор **Exit For** прекращает только **оператор For...Next**, внутри которого употребляется. Так в конструкции:

```
For x = 1 to 5
:
For y = 2 to 10 step 2
:
If error_condition
Exit For
End If
:
Next
:
Next
```

оператор **Exit For** прерывает выполнение внутреннего цикла (For y = 2 to 10 step 2) без оказывания влияния на внешний цикл (For x = 1 to 5).

См. также:

оператор Exit Do, **оператор For...Next**

Оператор Exit Function

Назначение

Останавливает подпрограмму, которую запустил **оператор Function...End Function**.

Предупреждение!

Оператор **Exit Function** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Exit Function
```

Описание

Оператор **Exit Function** прекращает выполнение функции на MapBasic. Оператор **Exit Function** нельзя выполнить в **операторе Function...End Function**.

Вызовы функции могут быть вложенными, то есть одна функция может вызывать другую. Каждый оператор **Exit Function** прекращает выполнение только той функции, внутри которой употребляется.

См. также:

Оператор Function...End Function

Оператор Exit Sub

Назначение

Останавливает подпрограмму, которую запустил **оператор Sub...End Sub**.

Предупреждение!

Оператор **Exit Sub** не может быть выполнен в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Exit Sub
```

Описание

Оператор **Exit Sub** осуществляет выход из текущей Sub-процедуры. Оператор **Exit Sub** может выполняться в теле только таких процедур.

Оператор **Sub...End Sub** может быть вложенным; т.е. может вызывать другую Sub-процедуру, которая, в свою очередь, вызывает третью и т. д. Каждый оператор **Exit Sub** прекращает выполнение только той подпрограммы, внутри которой употребляется.

См. также:

оператор Call, оператор Sub...End Sub

Функция Exp()

Назначение

Вычисляет значение экспоненты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Exp ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Математическая функция **Exp()** вычисляет значение числа *e*, возведенного в степень *num_expr*. Число *e* иррационально и приблизительно равно 2.7182818.

Примечание: Операция возведения в степень производится с помощью математического оператора (^).

Пример

```
Dim e As Float  
e = Exp(1)  
' the local variable e now contains  
' approximately 2.7182818
```

См. также:

[функция Cos\(\)](#), [функция Sin\(\)](#), [функция Log\(\)](#)

Оператор Export

Назначение

Экспортирует таблицу в файл другого формата. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис (вариант 1 - для экспорта в формат MIF/MID, DBF или ASCII)

```
Export table
Into file_name
[ Type
{ "MIF" |
"DBF" [ Charset char_set ] |
"ASCII" [ Charset char_set ] [ Delimiter "d " ] [ Titles ] |
"CSV" [ Charset char_set ] [ Titles ] } ]
[ Overwrite ]
```

Синтаксис (вариант 2 - для экспорта в формат DXF)

```
Export table
Into file_name
[ Type "DXF" ]
[ Overwrite ]
[ Preserve
[ AttributeData ] [ Preserve ] [ MultiPolygonRgns [ As Blocks ] ] ]
[ { Binary | ASCII [ DecimalPlaces decimal_places ] } ]
[ Version { 12 | 13 } ]
[ Transform
( MI_x1, MI_y1 ) ( MI_x2, MI_y2 )
( DXF_x1, DXF_y1 ) ( DXF_x2, DXF_y2 ) ]
```

table – имя открытой таблицы (в этом операторе кавычки при задании имени таблицы не используются);

file_name – строка с именем файла, в которую будут экспортироваться данные (если строка не включает DOS-маршрут, то файл будет создан в рабочем каталоге);

char_set – имя набора символов; см. раздел [Предложение CharSet](#).

d – символ, который будет использован как разделитель для ASCII-файла;

decimal_places – целое число типа Small Integer (короткое целое) от 0 до 16 (по умолчанию 6), задающее количество позиций после десятичной точки для экспорта числа с плавающей точкой в текстовый формат ASCII;

MI_x1, *MI_y1* и т. п. – пары чисел, представляющие собой координаты в таблице MapInfo Pro;
DXF_x1, *DXF_y1* и т. п. – пары чисел, представляющие собой координаты в файле DXF.

Описание

Оператор **Export** копирует содержимое таблицы MapInfo в отдельный файл другого формата для работы с этим файлом в других программах и операционных средах. Например, можно экспортировать содержимое таблицы в DXF-файл, чтобы импортировать его в AutoCAD или другую программу, которая работает с форматом DXF. Оператор **Export** не изменяет исходный файл.

Задание формата файла

Формат будущего файла объявляется при помощи предложения **Type**.

Предложение Type	Формат будущего файла
Type "MIF"	Формат обмена MapInfo (MapInfo Interchange File format). Описание формата MIF смотрите в документации MapInfo Pro.
Type "DXF"	Формат DXF (графический формат программ CAD, таких как AutoCAD).
Type "DBF"	Формат dBASE и других совместимых баз данных. Примечание: Графические объекты не экспортируются при использовании этого формата.
Type "ASCII"	Текстовый файл Примечание: Графические объекты не экспортируются при использовании этого формата.
Type "CSV"	Формат данных, разделенных запятыми Примечание: Графические объекты не экспортируются при использовании этого формата.

Если предложение **Type** отсутствует, MapInfo Pro пытается определить формат экспорта по расширению в имени файла. Например, если в операторе задается имя "PARCELS.DXF", то MapInfo Pro создаст файл формата DXF.

Если файл с заданным именем уже существует и в операторе используется слово **Overwrite**, MapInfo Pro создаст экспортный файл, записав его поверх старого. Если слово **Overwrite**

опущено и файл с заданным именем уже есть на диске, MapInfo Pro не будет переписывать файл.

Экспорт в текстовые ASCII-файлы

Если таблица экспортируется в текстовый ASCII-файл, то последний будет содержать разделители. Разделителем называется специальный символ, разделяющий данные полей в строке записи. Файлы в формате CSV используют запятую (,) в качестве разделителя автоматически. Другие разделители для экспорта в формат CSV задавать нельзя.

Стандартным разделителем для текстовых ASCII-файлов является символ табуляции (Chr\$(9)). Для этого случая можно задать в предложении **Delimiter** другой символ-разделитель. В следующем примере символом разделителя назначается двоеточие (:):

```
Export sites Into "sitedata.txt" Type "ASCII"
  Delimiter ":" Titles
```

При экспорте в файлы ASCII или CSV, Вы можете использовать слово **Titles**. Если задано слово **Titles**, то первая строка в тексте будет содержать заголовки колонок таблицы. Если слово **Titles** не задано, то названия колонок в текстовый файл не попадут (что может создать путаницу, если Вы планируете экспортировать данный файл).

Экспорт в DXF-файл

Для получения файла в формате DXF используется второй вариант синтаксиса оператора **Export**, который может включать приведенные ниже специфические для формата DXF предложения и ключевые слова.

Предложение **Preserve AttributeData** используется для экспорта табличных данных в атрибуты DXF-файла.

Предложение **Preserve MultiPolygonRgns As Blocks** используется для того, чтобы MapInfo Pro экспортировала область, состоящую из нескольких полигонов, в качестве блока DXF. Если это предложение будет опущено, то каждый полигон сложносоставной области будет представлен отдельным объектом.

Ключевое слово **Binary** используется для экспорта таблицы в бинарный файл DXF; предложение **ASCII** задает текстовый вид файла DXF. Если не используется ни один из этих вариантов, то MapInfo Pro создаст текстовый (ASCII) файл DXF. Бинарный файл DXF обычно компактнее, чем текстовый, и поэтому обрабатывается быстрее. При создании тестового файла можно задать число знаков после десятичной точки для экспорта численных данных, имеющих тип Float в таблице MapInfo. Параметр `decimal_places` может принимать значения от 0 до 16, по умолчанию 6.

Предложения **Version 12** и **Version 13** указывают, для какой версии программы AutoCAD будет создан файл DXF. Если предложение не задано, MapInfo Pro будет экспортировать в DXF-файл 12-ой версии.

Transform используется для задания координатного искажения. После слова **Transform** задаются координатные пары минимума и максимума таблицы MapInfo и какие пары координат будут соответствовать им в файле DXF.

Пример

Здесь открывается таблица "FACILITY " и экспортируется в файл FACIL.DXF.

```
Open Table "facility"

Export facility
Into "FACIL.DXF"
Type "DXF"
Overwrite
Preserve AttributeData
Preserve MultiPolygonRgn As Blocks
ASCII DecimalPlaces 3
Transform (0, 0) (1, 1) (0, 0) (1, 1)
```

Ссылки, связанные с данной

Оператор Import на стр. 505

Информация по теме

Ограничения на имена переменных на стр. 383

Функция ExtractNodes()

Назначение

Возвращает полилинию или область из подмножества узлов существующего объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ExtractNodes( object, polygon_index, begin_node, end_node, b_region )
```

object – объект типа "Полилиния" или "Область";

polygon_index – короткое целое число, 1 или больше; Этот индекс указывает, какой полигон (для регионов) или сегмент (полилиний) опрашивается.

begin_node – короткое целое число, 1 или больше, которое задает первый узел для выбираемого подмножества узлов;

end_node – короткое целое число, 1 или больше, которое задает последний узел для выбираемого подмножества узлов;

b_region – логическая величина, управляющая типом объекта, полученным в результате: для области – логическое "Да" (TRUE), для полилинии – логическое "Нет" (FALSE).

Возвращаемая величина

Объект типа "Полилиния" или "Область". Величина типа Object. MapBasic присваивает все стили (цвет и т. п.) оригинального объекта объекту, полученному в результате. Если каких-то стилей недостает, то берутся текущие в MapBasic значения.

Описание

Если значение параметра *begin_node* равно или больше значения *end_node*, то MapBasic будет отсчитывать узлы в следующем порядке:

- *begin_node* (начальный узел) - следующий узел -... - последний узел полигона;
- первый узел полигона - ... - конечный узел *end_node*.

Если параметр *object* задает область и оба параметра *begin_node* и *end_node* равны 1 (единице), то MapBasic вернет полное множество узлов одного полигона (многоугольника). Это простой механизм для выделения одного полигона из многокомпонентной области. Для определения числа узлов полигона из области используется [функция ObjectInfo\(\)](#).

Ошибки

Функция может вернуть код ошибки ERR_FCN_ARG_RANGE (644), если значение параметра *b_region* равно FALSE и подмножество узлов состоит менее чем из двух узлов или если значение параметра *b_region* равно TRUE и подмножество узлов состоит менее чем из трех узлов.

См. также:

[функция ObjectNodeX\(\)](#), [функция ObjectNodeY\(\)](#)

Оператор Farthest

Назначение

Находит объект в таблице, который находится от исходного объекта на максимальном расстоянии. Результатом является объект полилинии, представляющая наибольшее расстояние. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Farthest [ N | All ]
From { Table fromtable | Variable fromvar }
To totable Into intotable
[ Type { Spherical | Cartesian } ]
[ Ignore [ Contains ] [ Min min_value ]
  [ Max max_value ] Units unitname ]
[ Data clause ]

```

N – параметр (необязательный), представляющий число наиболее удаленных объектов, которые будут найдены. По умолчанию это 1. Если используется параметр **All**, то объект-расстояние будет создан для каждой комбинации.

fromtable – таблица объектов, от которых требуется найти наиболее удаленные объекты.

fromvar – переменная MapBasic для объекта, от которого ищется самое большое расстояние;

totable – таблица объектов, до которых будут искаяться самые дальние расстояния;

intotable – таблица, в которую будут записаны полученные результаты;

min_value – минимальная допустимая дистанция для результата;

max_value – максимальная допустимая дистанция для результата;

unitname – единицы измерения параметров *min_value* и/или *max_value* (например, “km”).

Описание

Оператор **Farthest** находит все объекты из таблицы *fromtable*, которые расположены далее всего от заданного объекта. Рассматривается каждый объект в таблице *fromtable*. Для каждого объекта в таблице *fromtable* будет найден наиболее самый дальний объект в таблице *totable*. Если задан параметр *N*, то будет найдено *N* наиболее дальних объектов в таблице *totable*. Объект-полилиния, соединяющий наиболее удаленные точки между объектом *fromtable* и выбранным объектом *totable*, помещается в таблицу *intotable*. Если задан параметр **All**, то объект помещается в таблицу *intotable*, представляющую расстояние между объектом *fromtable* каждым объектом таблицы *totable*.

Если имеется несколько объектов в таблице *totable*, которые имеют одинаковое расстояние до заданного объекта *fromtable*, только один из них может быть возвращён в виде результата. Если возвращается несколько объектов (задана величина *N*, большая чем 1), то объекты с одинаковым расстоянием будут записаны последовательно. Если существует второй объект с таким же расстоянием, а запрашивается 3 объекта, то искомым объектом станет третий по счёту объект.

Типы объектов в таблицах *fromtable* и *totable* могут быть любыми кроме текстов. Например, если обе таблицы содержат объекты-регионы, то находится минимальное расстояние между объектами-регионами – создаётся объект-полилиния, соединяющий две точки каждого объекта, использованных для расчёта расстояния. Если объекты-регионы пересекаются, минимальное

расстояние будет нулевым, и объект-полилиния будет вырожден до двух точек на пересечении, имеющими одни и те же координаты.

Расстояние, рассчитанное этой функцией, не принимается во внимание при расчёте маршрутов в транспортных задачах. Это расстояние можно сравнить с "перелётом птицы".

Предложение **Ignore** может использоваться для ограничения искомого расстояния, и может определять, сколько объектов таблицы *totable* найдено для каждого объекта таблицы *fromtable*. С помощью параметра **Min** можно исключать нулевые расстояния. Это может быть полезно в случае двух таблиц с точками, чтобы не рассчитывать расстояния от одинаковых точек. Например, если две таблицы с точками, представляют города, и мы хотим найти ближайшие города, мы можем исключить из поиска одинаковые города.

С помощью параметра **Max** можно ограничить число объектов, рассматриваемых в *totable*. Это полезно при совместном использовании с параметрами *N* или **All**. Например, надо найти 5 аэропортов, находящихся ближе остальных для нескольких городов (где *fromtable* это набор городов, а *totable* это набор аэропортов), но мы не будем рассматривать аэродромы, удалённые более чем на 100 км. Результат может оказаться меньшим, чем пять аэропортов. Этот параметр целесообразно использовать совместно с параметром **All**, так можно найти все аэропорты на расстоянии 100 км от города.

Применение параметра **Max** может улучшить поведение оператора **Farthest**, поскольку он эффективно ограничивает число искомых объектов *totable*.

Найденные расстояния находятся так, что они строго больше чем *min_value* и меньше или равны *max_value*:

```
min_value < distance <= max_value
```

Можно сделать так, чтобы расстояния возвращались в виде нескольких проходов оператора **Farthest**. Например, первый прогон может вернуть все объекты на расстоянии от 0 до 100 км, второй проход может вернуть все объекты на расстоянии от 100 до 200 км, а результаты не должны содержать повторяющихся записей (то есть, расстояние 100 встретилось только в первом проходе, а во втором уже не встречается).

Type – метод, используемый для расчета расстояния между объектами. Он может быть либо сферическим (Spherical), либо декартовым (Cartesian). Тип алгоритма вычислений должен соответствовать системе координат результирующей таблицы *intotable* или произойдет ошибка. Если система координат результирующей таблицы *intotable* не "Широта/Долгота" а метод расчёта сферический, то будет выдано сообщение об ошибке. Если система координат результирующей таблицы *intotable* – "Широта/Долгота", а метод расчёта декартовый, то будет выдано сообщение об ошибке.

Предложение **Ignore** ограничивает возвращаемые расстояния. Любые расстояния, найденные по запросу, которые меньше или равны *min_value* или больше *max_value* - игнорируются; *min_value* и *max_value* являются расстояниями и измеряются в единицах, определенных параметром *unitname*. Если *unitname* не соответствует единицам измерения, будет выдано сообщение об ошибке. Список доступных единиц измерения смотрите в разделе **оператор**

Set Distance Units. Предложение **Ignore** не является обязательным, равно как и подпредложения **Min** и **Max**.

Если один объект находится внутри другого, то расстояние между ними считается равным нулю. Например, если таблица *fromtable* это WORLDCAPS, а таблица *totable* - это WORLD, то расстояние между Москвой и Россией будет равно нулю. Если же задан режим **Contains** в предложении **Ignore**, то расстояние не будет обязательно нулевым. Вместо этого результатом будет расстояние между Москвой и границей России. Все замкнутые объекты, типа полигонов (регионов) будут обработаны как полилинии для выполнения этой операции.

Предложение **Data** используется для того, чтобы отметить от каких объектов из таблиц *fromtable* и *totable* будут получен результат.

Предложение Data

```
Data IntoColumn1=column1, IntoColumn2=column2
```

Переменная *IntoColumn* в левой части уравнения должна корректно задавать колонку в *intotable*. Имя *колонки* справа от знака равенства должно быть именем колонки либо таблицы *totable*, либо *fromtable*. Если одинаковое имя колонки существует в обеих таблицах *totable* и *fromtable*, то будет использоваться колонка из таблицы *totable* (т.е. в таблице *totable* в первую очередь ищутся имена колонок, а это правая часть "уравнения").

Чтобы избежать конфликтов такого типа, имена колонок можно заменять псевдонимами таблиц. Например:

```
Data name1=states.state_name, name2=county.state_name
```

Чтобы заполнить колонку в таблице *intotable* значениями расстояний, нужно выполнить команду **Обновить колонку** или использовать **оператор Update**.

См. также:

оператор Nearest, **функция CartesianObjectDistance()**, **функция ObjectDistance()**, **функция SphericalObjectDistance()**, **функция CartesianConnectObjects()**, **функция ConnectObjects()**, **функция SphericalConnectObjects()**

Оператор Fetch

Назначение

Задание текущей позиции курсора в таблице (т. е. какая строка таблицы должна быть текущей). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Fetch { First | Last | Next | Prev | Rec n } From table
```

n – номер записи для чтения;

table – имя открытой таблицы;

Описание

Оператор **Fetch** используется для позиционирования записи в открытой таблице. Выполняя оператор **Fetch**, программа помещает курсор на определенную строку таблицы. Это состояние записи в таблице называется “текущим”.

Примечание: Термин "курсор" используется здесь для отражения расположения строки в таблице. Он никак не связан с визуальным курсором экрана.

После оператора **Fetch** приложение MapBasic может извлекать данные из текущей записи, используя выражения:

Синтаксис	Пример
<i>table.column</i>	World.Country
<i>table.col#</i>	World.col1
<i>table.col(number)</i>	World.col(1)

Оператор **Fetch First** переводит курсор на первую неудаленную запись таблицы.

Оператор **Fetch Last** переводит курсор на последнюю неудаленную запись таблицы.

Оператор **Fetch Next** переводит курсор на следующую неудаленную запись таблицы.

Оператор **Fetch Prev** переводит курсор на предыдущую неудаленную запись таблицы.

Оператор **Fetch Rec** *n* переводит текущую позицию на определенную строку, даже если она удалена.

Примечание: Если запись удалена, то оператор генерирует ошибку выполнения с номером 404

Надо учитывать, что многие операторы MapBasic и команды MapInfo Pro сбрасывают текущее положение курсора в таблице (например оператор Select оператор Update, перерисовка экрана). Используйте обращение к текущей строке сразу после выполнения оператора **Fetch**.

Определение конца таблицы

После выполнения оператора **Fetch** функция **EOT()** возвращает логическую величину (TRUE или FALSE) в зависимости от того, достигнут ли предел таблицы после последней записи.

Если оператор **Fetch** помещает курсор на действительную запись таблицы, то функция **EOT()** возвращает значение FALSE.

Если оператор **Fetch** попытается установить текущей запись после последней записи таблицы, то функция **EOT()** вернет значение TRUE.

В следующем примере оператор **Fetch Next** проходит по циклу по всем строкам таблицы. После каждого применения оператора **Fetch Next** вызывается функция **EOT()** и, если она вернет истинное значение, то цикл будет прерван.

```
Dim i As Integer

i = 0
Fetch First From world
Do While Not EOT(world)
    i = i + 1
    Fetch Next From world
Loop

Print "Number of undeleted records: " + i
```

Запрос данных из таблиц GeoPackage с непоследовательными строками

Для таблиц MapInfo всегда считается, что строки от первой до n-ой перечисляются последовательно. Однако, таблицы GeoPackage не всегда содержат последовательно перечисленные данные; могут встречаться пустые строки, строки, отсчёт которых начинается не с единицы (1), и пропуски в идентификаторах записей-строк. На это следует обратить внимание, когда операторы MapBasic используются для запроса данных таблиц GeoPackage.

Для таблиц GeoPackage значение MapBasic **RowId** является первичным значением ключа, а не значением-счётчиком номера строки. Команда **Fetch Rec 3** возвращает строку со значением **primary_key** равным трём (3), если такое существует. Она не возвращает третью запись таблицы. Не выполняйте запрос по определённому идентификатору записи-строки. Используйте **Fetch First**, **Fetch Next**, **Fetch Last** вместо **Fetch Rec #**.

В следующем примере показан цикл по непоследовательно перечисленной таблице:

```
Dim iCount as integer
fetch first from USA_NonSeq
For iCount = 1 to TableInfo(USA_NonSeq, TAB_INFO_NROWS)
```

```
'COL5 is an integer column in USA_NonSeq table
Print USA_NonSeq.COL5
fetch next from USA_NonSeq
next
```

Примеры

В этом примере показано, как подвести курсор к 3-ей записи в таблице STATES:

```
Open Table "states"
Fetch Rec 3 From states 'position at 3rd record
Note states.state_name 'display name of state
```

Оператором **Fetch** можно извлекать значения из временных таблиц (например, из таблицы Selection).

```
Select * From states Where pop_1990 < pop_1980
Fetch First From Selection
Note Selection.col1 + " has negative net migration"
```

См. также:

[функция EOT\(\)](#), [оператор Open Table](#)

Функция FileAttr()

Назначение

Возвращает информацию об открытом файле. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FileAttr( filenum, attribute )
```

filenum – целое число, присвоенное файлу оператором Open File.

attribute – код, указывающий, какой атрибут следует вернуть; смотрите таблицу ниже.

Возвращаемая величина

Большое целочисленное значение (LargeInt)

Описание

Функция **FileAttr()** используется для получения информации об открытом в файле MapInfo. Параметр *attribute* должен принимать одно из следующих значений:

параметр <i>attribute</i>	ID	Возвращаемая величина
FILE_ATTR_MODE	1	Короткое целое, величина типа Small Integer, соответствующая коду режима, в котором был открыт файл. Значения: • MODE_INPUT (0) • MODE_OUTPUT (1) • MODE_APPEND (2) • MODE_RANDOM (3) • MODE_BINARY (4)
FILE_ATTR_FILESIZE	2	Целое число, величина типа Integer, размер файла в байтах.

Ошибки

ERR_FILEMGR_NOTOPEN (366), если файл не был открыт.

См. также:

[функция EOF\(\)](#), [оператор Get](#), [оператор Open File](#), [оператор Put](#)

Функция FileExists()

Назначение

Возвращает логическую величину (TRUE или FALSE), в зависимости от того, существует файл или нет. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FileExists( filespec )
```

filespec – строка с полным именем файла, включая DOS-маршрут.

Возвращаемая величина

Логическое: TRUE, если файл существует: иначе – FALSE.

Пример

```
If FileExists("C:\MapInfo\TODO.TXT") Then  
  Open File "C:\MapInfo\TODO.TXT" For INPUT As #1  
End If
```

См. также:

Функция TempFileName()

Функция FileOpenDlg()

Назначение

Вызывает диалог **Открыть файл**, чтобы открыть таблицу и возвращает имя файла, выбранное пользователем. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FileOpenDlg ( path, filename, filetype, prompt )
```

path – строковая величина, определяющая начальное значение выбранного каталога;

filename – строковая величина, определяющая начальное значение выбранного файла в каталоге;

filetype – строковая величина в три символа или меньше, определяющая тип файла;

prompt – строковая величина, задающая заголовок диалога.

Возвращаемая величина

Строка с именем файла или пустая строка, если диалог был отменен пользователем. Величина типа String.

Описание

Функция **FileOpenDlg()** открывает диалоговое окно подобно команде MapInfo **ФайлОткрыть таблицу**.

Чтобы выбрать файл из списка, представленного в диалоге, Вы можете указать на имя файла и нажать на кнопку **ОК**, или просто дважды указать мышкой на имя файла. В этом случае функция **FileOpenDlg()** возвращает полное имя выбранного файла, включая DOS-маршрут. Альтернативно, если нажата кнопка **Отмена**, то возвращаемая строка будет пуста ("").

Заметим, что функция **FileOpenDlg()** не открывает выбранный файл; она только показывает диалог, в котором файл можно выбрать. Для того чтобы открыть этот файл, программе необходим дополнительный оператор, например, **оператор Open Table**. Если же требуется, чтобы в программе выбранный файл **открывался** автоматически, используйте **оператор Run Menu Command** с аргументами M_FILE_OPEN или M_FILE_ADD_WORKSPACE.

Параметр *path* назначает каталог, в котором помещены файлы для выбора. Значение каталога является начальным, и ничто не мешает пользователю выбрать другой каталог в диалоговом окне. Если параметр не задан (пустое значение), то каталогом выбора будет текущий рабочий каталог.

Параметр *filename* принимает строковые значения и задает предварительное название файла.

Параметр *filetype* принимает строковые значения обычно длиной в три символа и определяет тип файлов для выбора в диалоговом окне. Диалог может автоматически реагировать на параметр *filetype*; так если, например, параметр *filetype* имеет значение "TAB", то диалог покажет список файлов таблиц MapInfo, если же задано значение *filetype* "WOR", то диалог покажет список файлов с рабочими наборами MapInfo.

Существуют и другие трехсимвольные значения *filetype*; они приведены в следующей таблице. Если надо вывести список всех файлов, то в качестве параметра *filetype* используется маска "*.*".

Параметр <i>filetype</i>	Типы файлов
"TAB"	Таблицы MapInfo.
"WOR"	Рабочие наборы MapInfo.
"MIF"	Формат обмена данными MapInfo (MapInfo Interchange Format), используется для импорта/экспорта карт в/из ASCII-текстовых файлов.
"DBF"	Формат dBASE или других совместимых баз данных.
"WKS", "WK1"	Табличные файлы Lotus.

Параметр <i>filetype</i>	Типы файлов
"XLS", "XLSX"	Табличные файлы Excel.
"DXF"	Формат обмена данными AutoCAD.
"MDB", "ACCDB"	Файлы базы данных Microsoft Access
"MMI", "MBI"	Формат обмена данными с MapInfo для DOS.
"MB"	Файлы с текстами программ MapBasic.
"MBX"	Файлы с откомпилированными программами MapBasic.
"TXT"	Текстовые файлы.
"BMP"	Формат растрового образа Windows.
"WMF"	Метафайлы Windows.

В Windows каждому типу соответствует определенное DOS-расширение имени файла. Другими словами, тип *filetype* файла "WOR" говорит MapBasic, чтобы был показан список файлов с расширением "WOR", т. к. имена с таким расширением имеют файлы Рабочих наборов в MapInfo для Windows.

Если требуется, чтобы программу можно было использовать в любой системе, используйте только трехсимвольную установку параметра *filetype*. В Windows некоторые специальные значения типа файла соответствуют строчке меню в нижнем левом углу диалога. Если в функции **FileOpenDlg()** параметр *filetype* задает значение не из приведенного выше списка, то в диалоге в окошке списка типов файлов будет показано пустое место.

Пример

```
Dim s_filename As String
s_filename = FileOpenDlg("", "", "TAB", "Open Table")
```

См. также:

функция FileSaveAsDlg(), оператор Open File, оператор Open Table

Функция FileSaveAsDlg()

Назначение

Вызывает диалоговое окно команды **Создать копию** в MapInfo и возвращает имя файла, введенное пользователем. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FileSaveAsDlg( path, filename, filetype, prompt )
```

path – строковая величина, определяющая начальное значение выбранного каталога;

filename – строковая величина, определяющая начальное значение выбранного файла в каталоге;

filetype – строковая величина в три символа или меньше, определяющая тип файла;

prompt – строковая величина, задающая заголовок диалога.

Возвращаемая величина

Строка с заданным именем файла (пустая строка, если введенные в диалоге данные были отменены).

Описание

Функция **FileSaveAsDlg()** открывает диалоговое окно **Создать копию** подобно команде MapInfo **ФайлСоздать копию**.

Пользователь может ввести с клавиатуры имя файла или выбрать файл из списка, представленного в диалоге. Для этого можно указать на имя файла и нажать кнопку ОК, или просто дважды указать мышкой на имя файла. Если файл уже существует на диске, MapBasic спросит, действительно ли необходимо заменить старое содержимое файла на новое.

После того, как пользователь выберет файл и нажмет на кнопку **ОК** функция **FileSaveAsDlg()** возвращает полное имя введенного или выбранного файла, включая DOS-маршрут. Если вы нажали на кнопку **Cancel**, то возвращаемая строка будет пуста ("").

Параметр *path* назначает каталог, в котором должен быть сохранен файл. Значение каталога является начальным, и пользователь может изменить каталог в диалоговом окне. Если параметр не задан (пустое значение), то каталогом будет текущий рабочий каталог.

Параметр *filename* принимает строковые значения и задает предварительное название файла.

Значение *filetype*, определяет расширение файла (до трех букв), то есть его тип. Например, если параметр *filetype* принимает значение "TAB", то диалог покажет список файлов таблиц MapInfo, а если – "WOR", то диалог покажет список файлов с рабочими Наборами MapInfo. Все специальные значения параметра *filetype* приведены в таблице в разделе: [Функция FileOpenDlg\(\)](#).

Функция **FileSaveAsDlg()** не сохраняет выбранный файл, но использует диалог соответствующей команды для получения полного имени, под которым надо сохранить файл. После этого для сохранения выбранного файла можно выполнить [оператор Commit Table](#).

См. также:

[оператор Commit Table](#), [функция FileOpenDlg\(\)](#)

Оператор Find

Назначение

Поиск объекта или пересечения объектов таблицы на карте. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Find address [ , region ] [ Interactive ]
```

address – строковая величина, определяющая адрес или имя искомого объекта карты; для поиска пересечения двух улиц параметр использует синтаксис: *streetname* && *streetname*.

region – имя области, в которой будет осуществляться поиск.

Описание

Оператор **Find** осуществляет поиск на Карте именованных элементов, заданных параметром *address*. Результаты поиска сохраняются в системных переменных, значения которых использует [функция CommandInfo\(\)](#). Если оператор **Find** включает ключевое слово **Interactive** и если MapBasic не может сразу обнаружить адрес, то поиск сопровождается диалоговым окном, в котором можно выбирать из нескольких "похожих" адресов.

Оператор **Find** может просматривать только таблицы, имеющие присоединенные графические объекты. Таблица при этом должна быть открыта. Оператор **Find** проводит поиск в той колонке, которая назначена для поиска. Программа MapBasic может выполнить [оператор Find Using](#), чтобы узнать, в какой колонке производить поиск. Если перед оператором **Find** не был выполнен оператор **Find Using**, MapBasic будет вести поиск, пользуясь установками, которые были заданы ранее в первом диалоге команды MapInfo Pro **ЗапросНайти**.

Оператор **Find** может уточнять поиск, если Вы зададите необязательный параметр `region` – обычно это имя области, в которой должен содержаться искомый объект. Другими словами, Вы можете просто попробовать найти город по названию (например, "Albany") в таблице городов или сузить область поиска, задав имя города и область, где он может находиться (например, "Albany", "CA"). Оператор **Find** не отмечает найденный объект символом, так как это делает команда поиска. Если необходимо показать пользователю результаты поиска, используется [функция CreatePoint\(\)](#) или [оператор Create Point](#); смотрите пример ниже.

Анализ результата поиска

Если после выполнения оператора **Find** изменить функцию **CommandInfo(CMD_INFO_FIND_RC)**, то Вы сможете узнать результат поиска. Если поиск оказался успешным, вызывайте функцию **CommandInfo(CMD_INFO_X)**, чтобы определить х-координату полученного в запросе объекта, и **CommandInfo(CMD_INFO_Y)** – для получения у-координаты. Если после оператора **Find** вызвать функцию **CommandInfo(CMD_INFO_FIND_ROWID)**, то она вернет номер строки, которой соответствует найденный объект.

Оператор **Find** может завершиться либо нахождением объекта, либо выбором ближайшего по смыслу объекта, либо объект не будет найден. Если поиск оператором **Find** точно находит объект, то функция **CommandInfo(CMD_INFO_FIND_RC)** возвратит единицу. Если был выбран вариант приблизительного совпадения, то функция **Find** возвратит число большее единицы. В случае неудачи поиска функция **Find** возвращает отрицательное значение.

В следующей таблице собраны различные варианты кодов, возвращаемых функцией **CommandInfo(CMD_INFO_FIND_RC)**. Каждое значение задается не более чем тремя

значащими цифрами, каждая из которых представляет результат одного из трех возможных этапов поиска.

Значения единиц	Значение
xx1	Точное совпадение.
xx2	Был задействован файл сокращений.
xx3 (-)	Точного совпадения не найдено.
xx4 (-)	Не было задано имя объекта; точного совпадения не найдено.
xx5 (+)	Пользователь выбрал имя в интерактивном диалоговом окне.
x1x	Не была определена сторона улицы.
x2x (+ / -)	Номер дома попал в заданные пределы для данной улицы.
x3x (+ / -)	Номер дома оказался вне заданных пределов для данной улицы.
x4x (+ / -)	Номер дома не задан.
x5x (-)	Улицы не пересекаются.

Значения единиц	Значение
x6x (-)	Найденной строке не сопоставлен объект на карте.
x7x (+)	Пользователь выбрал номер дома в интерактивном диалоговом окне.
1xx (+ / -)	Имя найдено только один раз вне уточняющей области.
2xx (-)	Имя найдено в нескольких регионах, но не в уточняющей области.
3xx (+ / -)	Имя найдено только один раз, но уточняющая область не была задана.
4xx (-)	Имя найдено несколько раз, но уточняющая область не была задана.
5xx (+)	Имя найдено несколько раз в уточняющей области.
6xx (+)	Пользователь выбрал имя области в интерактивном диалоговом окне.

Определить, какая цифра находится в определенном разряде числа, поможет арифметический оператор Mod. Например, чтобы определить последнюю цифру в числе, используйте деление по модулю `number Mod 10` (число). Чтобы определить две последние цифры в числе, используйте деление по модулю `number Mod 100`(число) и т.д.

Разницу между точным совпадением и приблизительным можно пояснить на следующем примере. Если таблица, содержащая имена городов, содержит одну запись для города "Albany" оператор Find Using задает поиск без уточняющей области, а оператор **Find** использует значение "Albany" как параметр address, то результатом будет полное совпадение. Следующий за таким оператором **Find** вызов функции **CommandInfo(CMD_INFO_FIND_RC)** возвратит 1 (единицу), что означает точное совпадение.

Теперь представим, что оператор действует в режиме использования уточняющей области; другими словами оператор **Find** ожидает, что название города будет дополнено названием штата (например, "Albany", "NY"). Если программа выполнит оператор **Find** со значениям "Albany" для адреса и нулевой строкой в качестве уточняющего названия штата, то с технической точки зрения точного совпадения не будет, потому что MapBasic ожидает, что название города должно сопровождаться названием штата.. Однако, так как город "Albany" упоминается только в одной записи таблицы, то MapBasic найдет эту запись. Следующий оператор **CommandInfo(CMD_INFO_FIND_RC)** возвратит 301. Цифра 1 обозначает совпадение адреса с названием города из таблицы, а цифра 3 означает частичный успех при поиске уточняющей области.

Если таблица улиц содержит запись для улицы "Main St" и оператор **Find** пытается найти улицу "Main Street", MapBasic оценит результат как частичное совпадение (принимая во внимание использование файла сокращений, смотрите также раздел об [операторе Find Using](#)). Строго говоря, строки "Main Street" и "Main St" не совпадают. Однако MapBasic полагает эти строки равными, поскольку находит подстановку в файле сокращений (MAPINFOW.ABB). Следующая за подобной группой операторов **Find** функция **CommandInfo(CMD_INFO_FIND_RC)** возвратит 2.

Если операция поиска сопровождается диалогом, в котором пользователь вводит свой текст, то код возврата будет иметь 1 (единицу) в разряде миллионов.

Пример

```
Include "mapbasic.def"
Dim x, y As Float, win_id As Integer
Open Table "states" Interactive
Map From States
win_id = FrontWindow()
Find Using states(state)
Find "NY"
If CommandInfo(CMD_INFO_FIND_RC) >= 1 Then
  x = CommandInfo(CMD_INFO_X)
  y = CommandInfo(CMD_INFO_Y)
  Set Map
  Window win_id
  Center (x, y)
  ' Now create a symbol at the location we found.
  ' Create the object in the Cosmetic layer.
  Insert Into
  WindowInfo( win_id, WIN_INFO_TABLE) (Object)
  Values ( CreatePoint(x, y) )
Else
  Note "Location not found."
End If
```

См. также:

[функция CommandInfo\(\)](#), [оператор Find Using](#), [функция OverlayNodes\(\)](#)

Оператор Find Using

Назначение

Устанавливает, какие таблицы и какие колонки будут рассматриваться при последующем поиске оператором Find. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Find Using table ( column )
[ Refine Using table ( column ) ]
[ Options
[ Abbrs { On | Off } ]
[ ClosestAddr { On | Off } ]
[ OtherBdy { On | Off } ]
[ Symbol symbol_style ]
[ Inset inset_value { Percent | Distance Units dist_unit } ]
[ Offset value ] [ Distance Units dist_unit ] ]
```

table – имя открытой таблицы;

column - имя колонки в таблице.

symbol_style – величина типа Symbol для задания стиля символа, которым будет отмечен найденный объект, если пользователь выполнит команду **ЗапросНайти**.

inset_value – положительное целое значение, определяющее величину смещения от концов улиц для размещения адреса;

value – определяет значение отступа Offset (расстояние от улицы - в глубину от улицы);

dist_unit – строка, определяющая название единиц измерения расстояний (например, "mi" для миль или "m" для метров).

Описание

Оператор **Find Using** определяет таблицу (таблицы) и колонку (колонки), в которых будет осуществлять поиск **оператор Find**. Колонки должны быть индексированы.

Предложение **Refine** используется для назначения второй таблицы, уточняющей поиск. Вторая таблица должна содержать объекты типа "Область". Заданная для уточнения поиска колонка должна быть индексирована. Если пропустить параметр **Refine**, то последующие операторы будут искать просто название (например, "Portland"). Если же параметр **Refine** задан, то последующие операторы Find будут искать название объекта и название района (например, "Portland", "OR").

Предложение **Abbrs** определяет, будет ли применяться файл сокращений в процедурах поиска. По умолчанию режим применения файла сокращений включен (**On**); чтобы отключить его, нужно явно задать предложение **Abbrs Off**.

Предложение **ClosestAddr** задает режим автоматической подстановки ближайшего адреса в случае не достижения точного совпадения. По умолчанию режим подстановки ближайшего адреса выключен (**Off**); чтобы включить его, нужно явно задать предложение **ClosestAddr On**.

Предложение **OtherBdy** задает режим автоматической подстановки адреса из другого региона, если в заданном уточняющем регионе ничего подходящего не было найдено. По умолчанию этот режим выключен (**Off**); чтобы включить его, нужно явно задать предложение **OtherBdy On**.

MapInfo Pro сохраняет параметры **Inset** и **Offset**, заданные при выполнении операции оператора **Find Using**. Таким образом, последние настройки становятся параметрами по умолчанию для следующего сеанса работы.

Если указан параметр **Percent**, то смещение определяется как процент от длины улицы. Для **Percent** возможны следующие значения *inset_value*: от 0 до 50. Если указан параметр **Distance Units**, то *inset_value* представляет расстояние от концов улицы для размещения адреса. В этом случае возможны следующие значения *inset_value*: от 0 до 32,767. Этот параметр используется для того, чтобы символы найденных объектов, расположенных близко к концу улицы сдвигались на экране, более равномерно распределяясь к центру улицы.

Значение **Offset** задает глубину размещения символа найденного от улицы, чтобы также избежать кучности близко расположенных символов. *Value* – положительная целая величина, определяющая как далеко "отступать" от улицы для размещения адреса. Возможные значения - от 0 до 32,767.

Пример

```
Find Using city_1k(city)
  Refine Using states(state)

Find "Albany", "NY"
```

См. также:

[оператор Create Index](#), [оператор Find](#)

Функция Fix()

Назначение

Возвращает целое число, полученное из целой части действительного числа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Fix( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Целое – Integer

Описание

Функция **Fix()** отсекает дробную часть от действительного числа, и возвращает целую часть. Функция **Fix()** и **Int()** похожи, но не идентичны.. Функции различаются способом удаления дробной части отрицательного числа. Для отрицательного числа функция **Fix()** возвращает ближайшее целое, больше или равное оригиналу. Например:

```
Fix(-2.3)
```

возвращает значение -2. Функция же **Int()** возвращает ближайшее целое, меньше или равное оригиналу. Например:

```
Int(-2.3)
```

возвращает значение -3.

Пример

```
Dim i_whole As Integer
i_whole = Fix(5.999)
' i_whole now has the value 5.

i_whole = Fix(-7.2)
' i_whole now has the value -7.
```

См. также:

[функция Int\(\)](#), [функция Round\(\)](#)

Предложение Font

Назначение

Определяет шрифт для текстов. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Font font_expr
```

font_expr – описание шрифта, величина или имя переменной типа Font, вызов функции, такой как:

```
MakeFont( fontname, style, size, fgcolor, bgcolor )
```

Описание

Параметром этого оператора **Font** можно задать стиль оформления текста. **Font** не является отдельным оператором, а используется в других операторах, где требуется задать стиль оформления текста (шрифта). Например, **оператор Create Text** позволяет установить атрибуты шрифта будущего текстового объекта: имя шрифта, написание, размер. Если в операторе Create Text опустить параметр **Font**, то для новых текстовых объектов будет использован текущий шрифт MapInfo Pro. За словом **Font** следовать выражение, имеющее значение типа Font.

Это выражение может быть переменной Font:

```
Font font_var
```

или вызовом функций, возвращающих такой тип данных (например, **функция CurrentFont()** или **функция MakeFont()**):

```
Font MakeFont("Helvetica", 1, 12, BLACK, WHITE)
```

Некоторые операторы MapBasic (например, **оператор Set Legend**) требуют, чтобы за словом **Font** следовали заключенные в скобки пять параметров, описывающих стиль: fontname, style, point_size, foreground_color, background_color:

```
Font("Helvetica", 1, 12, BLACK, WHITE)
```

В следующей таблице приводится описание этих параметров шрифта:

Компонента стиля	Описание
font name	Строковое значение, имя шрифта. Зависит от набора шрифтов, используемых операционной системой, в которой будет работать приложение MapBasic.
style	Целое число. Управляет написанием: жирным шрифтом, курсивом или подчеркнутым. Смотрите следующую таблицу.
size	Целое число, величина типа Integer, размер шрифта в пунктах. Размер 12 пунктов соответствует высоте буквы, равной одной шестой дюйма.
foreground color	Целое число, величина типа Integer, задающее цвет букв по шкале RGB. Смотрите: функция Rnd() .
background color	Целочисленный код цвета фона штриха. Целое число, величина типа Integer, задающее цвет фона или каймы текста по шкале RGB. Для задания прозрачного фона текста просто опустите этот параметр в предложении Font . Например <code>Font("Helvetica", 1, 12, BLACK)</code> . Если используется функция MakeFont() , пятый параметр должен быть равен 1 для задания прозрачного фона текста.

Следующая таблица содержит значения для параметра style, задающего написание шрифта:

Значения style	Стиль написания
0	Обычный
1	Жирный
2	Курсив
4	Подчеркнутый
8	Перечеркнутый
32	Тень
256	Кайма
512	Капитель

Значения style	Стиль написания
1024	Разрядка

Чтобы скомбинировать два и более эффектов, числа нужно сложить. Например, для написания жирным шрифтом и только заглавными буквами параметр style должен иметь значение 513.

Пример

```
Include "MAPBASIC.DEF"
Dim o_title As Object
Create Text
  Into Variable o_title
  "Your message could go HERE"
  (73.5, 42.6) (73.67, 42.9)
  Font MakeFont("Helvetica",1,12,BLACK,WHITE)
```

См. также:

[оператор Alter Object](#), [функция Chr\\$\(\)](#), [оператор Create Text](#), [функция RGB\(\)](#)

Оператор For...Next

Назначение

Выполняет последовательность действий в цикле с заданным числом итераций.

Ограничения!

Оператор **For...Next** нельзя выполнить в окне **MapBasic**.

Синтаксис

```
For var_name = start_expr To end_expr [ Step inc_expr ]
  statement_list
Next
```

var_name – имя переменной, выполняющей роль счетчика цикла;

start_expr – численное выражение;

end_expr – численное выражение;

inc_expr – численное выражение;

statement_list – группа операторов, выполняющихся за одну итерацию цикла.

Описание

Оператор **For...Next** задает цикл. Управление циклом происходит при помощи целочисленной переменной *var_name*, выступающей в роли счетчика. Оператор **For...Next** последовательно выполняет группу операторов *statement_list* заданное число раз, либо пропускает всю группу *statement_list* целиком. Выражения *start_expr*, *end_expr* и *inc_expr* определяют, сколько раз будет выполнена группа операторов *statement_list*.

Выполнение цикла **For...Next** начинается с присваивания значения *start_expr* переменной *var_name*. Если это значение меньше или равно *end_expr*, MapBasic выполняет набор операторов *statement_list*, после чего прибавляет значение *inc_expr* к начальному значению. Если предложение **Step** отсутствует в операторе, то счетчик увеличивается на одну единицу. Затем MapBasic сравнивает это значение со значением *end_expr*, если счетчик меньше или равен *end_expr*, MapBasic снова повторяет операторы из списка *statement_list*. Если же *var_name* больше, чем *end_expr*, MapBasic останавливает цикл, и передает управление оператору, следующему за оператором **Next**.

Оператор **For...Next** может также работать в обратную сторону, используя отрицательное значение шага **Step**. В этом случае, каждый проход цикла уменьшает значение *var_name* и MapBasic повторяет операции цикла до тех пор, пока *var_name* остается большим или равным *end_expr*.

Каждое предложение **For** должно заканчиваться парным **Next**. Все операторы, которые находятся между операторами **For** и **Next**, являются операторами тела цикла *statement_list* и выполняются за одну итерацию (проход) цикла.

Используйте **оператор Exit For** для немедленного прекращения цикла, независимо от значения счетчика *var_name*. Оператор **Exit For** просто передает управление программы следующему оператору после слова **Next**.

MapBasic позволяет изменять величину переменной *var_name* в теле цикла и тем самым влиять на выполнение цикла. Но для соблюдения хорошего стиля программирования переменную *var_name* изменять не рекомендуется.

Пример

```
Dim i As Integer

' the next loop will execute a Note statement 5 times
For i = 1 to 5
    Note "Hello world!"
Next

' the next loop will execute the Note statement 3 times
For i = 1 to 5 Step 2
    Note "Hello world!"
Next
```

```
' the next loop will execute the Note statement 3 times
For i = 5 to 1 Step -2
  Note "Hello world!"
Next

' MapBasic will skip the following For statement
' completely, because the initial start value is
' already larger than the initial end value
For i = 100 to 50 Step 5
  Note "This note will never be executed"
Next
```

См. также:

[оператор Do...Loop](#), [оператор Exit For](#)

Процедура ForegroundTaskSwitchHandler

Назначение

Процедура, автоматически выполняющаяся при изменении фокуса окна программы MapInfo Pro относительно других программ, выполняющихся в среде Windows.

Синтаксис

```
Declare Sub ForegroundTaskSwitchHandler

Sub ForegroundTaskSwitchHandler
  statement_list
End Sub
```

statement_list – список операторов процедуры.

Описание

Когда пользователь запускает программу, в которой есть процедура **ForegroundTaskSwitchHandler**, MapInfo Pro вызывает эту процедуру каждый раз, когда MapInfo Pro теряет или приобретает фокус. В процедуре может быть использована [функция CommandInfo\(\)](#) для определения, приобретает ли MapInfo Pro фокус или теряет его.

Пример

```

Sub ForegroundTaskSwitchHandler

  If CommandInfo(CMD_INFO_TASK_SWITCH)
    = SWITCHING_INTO_MAPINFO Then

    ' ... then MapInfo just became active
  Else
    ' ... another app just became active
  End If

End Sub

```

См. также:

функция [CommandInfo\(\)](#)

Функция Format\$

Назначение

Возвращает строковое представление числа в заданном формате. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Format$ ( value, pattern )
```

value – численное выражение;

pattern – шаблонная строка, задающая формат.

Возвращаемая величина

String

Описание

Функция **Format\$()** преобразует число в строковую величину по шаблону. Например, число 12345.67 функция **Format\$()** может преобразовать в строку "\$12,345.67".

Параметр *value* задает число, которое нужно отформатировать.

Параметр *pattern* задает шаблон в виде строки, определяющий правила форматирования. Строка-pattern должна включать один или более специальных символов форматирования, таких как #, 0, %, запятая (,), точка (.), точка с запятой (;); эти символы контролируют, как будет выглядеть результат. Роль каждого из этих символов в формировании результирующей строки приведена в следующей таблице:

Символ в управляющей строке-pattern	Какую роль играет для результата функции
#	Соответствует одной или нескольким цифрам в value.

Символ в
управляющей
строке-pattern

Какую роль играет для результата функции

	Если шаблон содержит один или более символов # левее десятичного знака, а форматируемое число меньше единицы, но больше нуля, то в результате ноль в целых частях не будет отображен, строка будет начинаться с точки.
0	Символ-заместитель цифры задает место (подобно #), но только для одного численного символа. Если управляющая строка содержит один или более символов 0 левее десятичной точки и значение числа меньше единицы и больше нуля, то результирующая строка будет иметь ноль перед десятичной точкой. См. примеры ниже.
. (period)	Соответствует десятичной точке. Число символов правее точки определяет число десятичных разрядов, то есть, сколько будет выведено символов после десятичной точки. Результирующая строка будет включать в себя тот десятичный разделитель, который установлен в системе. Чтобы назначить в качестве десятичного разделителя точку, даже если в системе принят другой знак, используется оператор Set Format .
, (comma)	Разделитель тысяч. Если управляющая строка содержит запятую перед знаком решетки, определяющей целую часть числа, то в результате каждые три цифры будут отделены запятой. Например, десять миллионов будут показаны так: "10,000,000". Результирующая строка будет включать в себя тот десятичный разделитель, который установлен в системе. Чтобы явно задать запятую в качестве разделителя тысяч, используйте оператор Set Format.
%	Процентное представление числа. Если управляющая строка содержит знак процента, то результатом будет строка с числом, умноженным на сто и знаком процента справа. Например, число 0.75 будет преобразовано в строку "75%". Если не требуется преобразовывать число в проценты, но хотите иметь в строке знак %, используйте его в комбинации со знаком "\" (обратная косая – обратный слэш).
E+	Представление числа в научном (экспоненциальном) формате. Например, число 1234 будет преобразовано в строку "1.234e+03". Если экспонента положительна, то после символа "e" будет помещен плюс. Если экспонента отрицательна, после символа "e" будет помещен знак минус.
E-	Представление числа в научном (экспоненциальном) формате. От предыдущего отличается тем, что в случае положительной экспоненты знак плюс не отображается.
; (semi-colon)	

Символ в управляющей строке-pattern

Какую роль играет для результата функции

	Включив точку с запятой в шаблон, вы можете указать один формат для положительных чисел, а другой для отрицательных. Так, в управляющей строке сначала определяется формат для представления положительных значений числа. Потом через точку с запятой задается другой формат для отрицательных значений. Если необходимо иметь знак минус, включите “-” во второй набор форматирующих символов для отрицательных значений.
\	Обратная косая (обратный слэш) инструктирует MapBasic не обрабатывать символ, следующий непосредственно за ним. Если обратный слэш стоит перед специальным символом (например, перед %), то MapBasic будет рассматривать этот символ в шаблоне как текстовый символ.

Ошибки

В результате выполнения функции можно получить код ошибки ERR_FCN_INVALID_FMT (643), если неправильно задан параметр управляющих символов – pattern.

Примеры

Следующие примеры показывают, какие результаты получаются при применении различных строковых шаблонов. Результаты показаны в комментариях.

Примечание: Внимание: в локализованных системах Вы можете получить те же результаты, но немного в другом виде.

```
Format$( 12345, ",#") ' returns "12,345"
Format$(-12345, ",#") ' returns "-12,345"
Format$( 12345, "$#") ' returns "$12345"
Format$(-12345, "$#") ' returns "-$12345"

Format$( 12345.678, "$,#.##") ' returns "$12,345.68"
Format$(-12345.678, "$,#.##") ' returns "-$12,345.68"

Format$( 12345.678, "$,#.##;($,#.##)") 'returns "$12,345.68"
Format$(-12345.678, "$,#.##;($,#.##)") 'returns "($12,345.68)"
Format$(12345.6789, ",#.###") ' returns "12,345.679"
Format$(12345.6789, ",#.##") ' returns "12,345.7"
Format$(-12345.6789, "#.###E+00") ' returns "-1.235e+04"
Format$( 0.054321, "#.###E+00") ' returns "5.432e-02"

Format$(-12345.6789, "#.###E-00") ' returns "-1.235e04"
Format$( 0.054321, "#.###E-00") ' returns "5.432e-02"
```

```
Format$(0.054321, "#.##%") ' returns "5.43%"
Format$(0.054321, "#.##\%") ' returns ".05%"
Format$(0.054321, "0.##\%") ' returns "0.05%"
```

См. также:

[функция Str\\$](#)

Функция FormatDate()

Назначение

Возвращает дату в укороченном формате, в том стиле, который определен в Панели управления. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Примечание: Функция **FormatDate\$()** не позволяет задавать другие представления даты, в то время как функция **FormatTime\$()** позволяет менять вид значения.

Синтаксис

```
FormatDate$( value )
```

value – число, представляющее дату в формате ГГГГММДД, где ГГГГ – год, ММ – месяц в диапазоне от 01 (январь) до 12 (декабрь), ДД – день месяца в диапазоне от 01 до 31.

Возвращаемая величина

Строка – String

Описание

Функция **FormatDate\$()** возвращает строковую величину, представляющую дату с локальным системном формате, как определено в Панели управления (Windows Control Panel).

Если вы зададите год двумя цифрами (например, 96), то MapInfo Pro припишет этот год к веку, который задает **оператор Set Date Window()**.

В большинстве случаев, год можно задавать двумя цифрами. (Для более точной настройки используется Date window ("годовой диапазон")). Смотрите: [функция DateWindow\(\)](#).

Примеры

В этих примерах принимается, что в **Панели управления** заданы День/Месяц/Год в качестве порядка перечисления дат, '-' – в качестве символа-разделителя и "dd-МММ-уууу" – в качестве короткого формата даты:

```
Dim d_Today As Date
d_Today = CurDate()
Print d_Today 'returns "19970910"
Print FormatDate$( d_Today ) 'returns "10-Sep-1997"
Dim s_EnteredDate As String
s_EnteredDate = "03-02-61"
Print FormatDate$( s_EnteredDate ) 'returns "03-Feb-1961"
s_EnteredDate = "12-31-61"
Print FormatDate$( s_EnteredDate ) ' returns ERROR: not d/m/y ordering
s_EnteredDate = "31-12-61"
Print FormatDate$( s_EnteredDate ) ' returns 31-Dec-1961"
```

См. также:

функции даты и времени, функция FormatTime\$(), функция DateWindow(), оператор Set Date Window()

Функция FormatNumber()

Назначение

Форматирование числа с использованием символов десятичной точки и разделителя тысяч – тех, которых в данное время использует система, в которой выполняется приложение. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FormatNumber$( num )
```

num – численная или строковая величина, представляющее число, например, "1234.56".

Возвращаемая величина

Строка – String

Описание

Функция возвращает строку, представляющую число. Если число достаточно велико, то функция вставит разделители тысяч. MapInfo Pro прочитывает конфигурацию пользовательской системы и определяет, какие символы используются для десятичной точки и разделителя тысяч.

Примеры

Следующая таблица демонстрирует, как работает функция **FormatNumber\$ ()** в компьютере, в котором точка обозначает десятичную точку, а запятая – разделитель тысяч (стандартная установка в США):

Вызов функции	Результат
<code>FormatNumber\$ ("12345.67")</code>	"12,345.67" (добавлен разделитель тысяч)
<code>FormatNumber\$ ("12, 345.67")</code>	"12,345.67" (без изменений)

Если в компьютере знак запятой зарезервирован за десятичной точкой, а точка разделяет тысячи, то функция будет работать так:

Вызов функции	Результат
<code>FormatNumber\$ ("12345.67")</code>	"12.345,67" (добавлен разделитель тысяч и изменен знак десятичной точки)
<code>FormatNumber\$ ("12, 345.67")</code>	"12.345,67" (изменены оба знака)

См. также:

[функция DeformatNumber\(\)](#)

Функция FormatTime\$()

Назначение

Возвращает строку с временем в формате, заданным вторым аргументом. Эту функцию можно вызывать в окне **MapBasic** в среде MapInfo Pro.

Примечание: Функция **FormatDate\$()** не позволяет задавать другие представления даты, в то время как функция **FormatTime\$()** позволяет менять вид значения.

Синтаксис

```
FormatTime$( time, string )
```

time – значение времени.

string – формат, который используется в возвращаемом строковом значении (смотрите раздел описания).

Возвращаемая величина

Строка – String

Описание

Форматирующая строка *string* должна удовлетворять стандартам Microsoft на настройки местного времени.

Часы	Значение
h	Часы без нулей перед значениями часов (12-часовой-отсчет).
hh	Часы с нулями перед значениями часов (12-часовой-отсчет).
H	Часы без нулей перед значениями часов (24-часовой-отсчет).
HH	Часы с нулями перед значениями часов (24-часовой-отсчет).
Минуты	Значение

Часы	Значение
m	Минуты без нулей перед значениями минут.
mm	Минуты с нулями перед значениями минут.
Секунды	Значение
s	Секунды без нулей перед значениями секунд.
ss	Секунды с нулями перед значениями секунд.
Маркер времени	Значение
t	Строка отметки времени, состоящая из единственного символа. Примечание: В некоторых языках не применяется, например, японском (Япония). В случае использования такого формата отметка времени, заданная системными параметрами LOCALE_S1159 (AM) и LOCALE_S2359 (PM), сокращается до единственного символа. Поэтому приложение может задать неправильное форматирование, когда одна и та же строка будет использоваться и для значений AM, и для PM.
tt	Строка отметки времени из нескольких символов.

Источник: <https://msdn.microsoft.com/en-us/library/dd318148.aspx>

В предыдущих форматах буквы m, s и t обязательно должны вводиться в нижнем регистре; буква h в нижнем регистре предназначена для 12-часового отсчета, в то время, как в верхнем регистре указывает на 24-часовой.

Используемый нами код отвечает правилам, установленным для системного местного формата времени. Кроме того, допускается использование f, ff или fff для десятых, сотых долей секунды или миллисекунд.

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim Z as time
Z = CurTime()
Print FormatTime$(Z, "hh:mm:ss.fff tt")
```

См. также:

функции даты и времени, функция `FormatDate$()`, функция `GetTime()`, функция `NumberToDateTime()`

Оператор FME Refresh Table

Назначение

Обновляет таблицу Universal Data Source (FME) значениями исходной таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FME Refresh Table alias
```

alias - псевдоним, выбранный для зарегистрированной таблицы Universal Data Source (FME).

Пример:

Следующий пример демонстрирует обновление локальной таблицы под названием watershed.

```
FME Refresh Table watershed
```

Функция FrontWindow()

Назначение

Возвращает идентификатор активного окна. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
FrontWindow ( )
```

Возвращаемая величина

Целое – Integer

Описание

Функция **FrontWindow()** возвращает целочисленный идентификатор самого верхнего окна документа: (**Карта**, **Список**, **Отчёт**, **График**). Этот оператор полезно применять сразу после создания нового окна (**Карта**, **Список**, **Отчёт** или **График**), чтобы запомнить значение идентификатора самого верхнего окна.

Пример

```
Dim map_win_id As Integer
Open Table "states"
Map From states
map_win_id = FrontWindow()
```

См. также:

[функция NumWindows\(\)](#), [функция WindowID\(\)](#), [функция WindowInfo\(\)](#)

Оператор Function...End Function

Назначение

Объявляет новую функцию.

Предупреждение!

Оператор **Function...End Function** нельзя выполнить в окне **MapBasic**.

Синтаксис

```
Function name ( [ [ ByVal ] parameter As datatype ]
  [ , [ ByVal ] parameter As datatype... ] ) As return_type
  statement_list
End Function
```

name – имя функции;

parameter – имя параметра функции.

datatype – тип данных, стандартный в MapInfo, или тип, созданный при помощи оператора **Type**;

return_type – стандартный для MapInfo скалярный тип для величины, полученной в результате.

statement_list – группа операторов, составляющая тело функции.

Описание

Оператор **Function...End Function** позволяет создавать свои функции. Созданные таким образом функции являются внутренними и могут использоваться в программе, в которой находятся наравне со стандартными функциями MapInfo Pro.

Каждое имя функции, созданной при помощи оператора **Function...End Function**, а также типы параметров и результата должны быть ранее объявляет **оператор Declare Function**.

Внутренняя функция подобна процедуре, сделанной при помощи оператора **Sub**, но в отличие от процедуры функция возвращает значение. Функция по сравнению с процедурой в применении более гибка, т.к. можно составлять выражения, содержащие вызов к одной и более функциям. Например, в следующем выражении дважды вызывается **функция Proper\$()**:

```
fullname = Proper$(firstname) + " " + Proper$(lastname)
```

В конструкции оператора **Function...End Function** параметр name, определяющий имя функции, трактуется как имя переменной. Этой переменной и присваивается значение, возвращаемое функцией. Если переменной name не было ничего присвоено функция возвращает нулевое значение для числовых функций, FALSE для логических функций или нулевую строку ("") для строчных функций.

Ограничения при передаче параметров

Результат функции может быть только скалярным. Другими словами функция не может возвращать массивы и данные сложных типов, созданных оператором Type. По умолчанию, каждый параметр во внутреннюю функцию передается "ссылкой" ("by-reference"). Это означает, что оператор, вызывающий функцию, должен в качестве параметров использовать имена переменных. Если функция изменяет значение параметра, передаваемого ссылкой, то изменяется и переменная в вызывающей процедуре.

Любой или все параметры функции могут передаваться "значением", если перед ними явно указано ключевое слово **ByVal** в определении оператора **Function...End Function**. В этом случае оператор, вызывающий функцию, может использовать в качестве параметра целое выражение, а не одно только имя переменной. Однако, если изменить значение такого параметра, новое значение нельзя будет вернуть в вызывающий модуль. Вы не можете передавать в виде значения (**ByVal**) во внутреннюю функцию массивы, а также переменные, заданные оператором. Однако, все такие данные можно передавать параметрами-ссылками. Если функция не должна иметь параметров, то оператор **Function...End Function** может, включать пустые скобки или не использовать их вообще. Тем не менее, каждый вызов функции должны включать в себя пару скобок, независимо от того, принимает ли функция параметры. Например, задавая функцию Foo оператором **Function...End Function**, можно использовать запись:

```
Function Foo()  
' ... statement list goes here ...  
End Function
```

которая идентична конструкции:

```
Function Foo
' ... statement list goes here ...
End Function
```

Но при вызове этой функции использование скобок обязательно:

```
var_name = Foo()
```

Доступность функций пользователя

Пользователь не может помещать вызовы к внутренним функциям выполняющейся прикладной программы при заполнении стандартных диалогов в MapInfo Pro. Однако сама программа может это делать. Так, пользователь не может обратиться к определенной программой функции из диалога команды **SQL-запрос**, однако, в скомпилированной MapBasic-программе может содержаться **оператор Select**, использующий эту функцию.

Внутреннюю функцию можно вызывать только в той программе, в которой она была определена. Поэтому, если требуется использовать такую функцию в другой программе, то придется полностью скопировать определение **Function...End Function** в текст другой программы.

Имена функций

Имя функции, определенное оператором **Function...End Function**, может совпадать с именем стандартной функции MapBasic, таким как **Abs** или **Chr\$**. Подобная функция полностью замещает одноименную стандартную в пределах прикладной программы, в которой она определена. Например, если создать свою функцию **Abs**, то все вызовы функции **Abs** будут обращаться к Вашей функции, а стандартная **функция Abs()** использоваться не будет.

Если произошло описанное переопределение, то на другие программы оно не действует. Так, если вы пишете несколько программ и планируете, чтобы в каждой была своя особая **функция Distance()**, то в каждой из программ необходимо отдельно задавать оператор **Function...End Function**.

Когда в программе MapBasic переопределяется стандартная функция, то это переопределение распространяется на всю программу. В каждой процедуре программы вызовы будут адресоваться к переопределенной пользователем функции, а не к стандартной.

Пример

В этом примере определяется функция пользователя под названием "CubeRoot", которая вычисляет кубический корень из числа. Так как функция CubeRoot вызывается в тексте

программы до определения самой функции, то в этом примере используется **оператор Declare Function**, определяющий аргумент функции и тип возвращаемого значения.

```
Declare Function CubeRoot(ByVal x As Float) As Float
Declare Sub Main

Sub Main
  Dim f_result As Float
  f_result = CubeRoot(23)
  Note Str$(f_result)
End Sub

Function CubeRoot(ByVal x As Float) As Float
  CubeRoot = x ^ 0.333333333333
End Function
```

См. также:

оператор Declare Function, оператор Declare Sub, оператор Sub...End Sub

Оператор Geocode

Назначение

Проводит операцию геокодирования таблицы или отдельной записи с помощью удаленной службы геокодирования, для соединения с которой использовался **оператор Open Connection**, а для настройки **оператор Set Connection Geocode**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Geocode connection_number
Input
[ Table input_tablename ]
[ Country = Country_expr
[ Street = Street_expr,
  [ IntersectingStreet = IntersectingStreet_expr ],
  Municipality = Municipality_expr,
  CountrySubdivision = CountrySubdiv_expr,
  PostalCode = PostalCode_expr,
  CountrySecondarySubdivision = CountrySecondarySubdiv_expr,
  SecondaryPostalCode = SecondaryPostalCode_expr,
  Placename = Placename_expr,
  Street2 = Street2_expr,
  MunicipalitySubdivision = MunicipalitySubdiv_expr ] ]
```

```

[ PassThrough name value, name value, ... ]
Output
[ Into
[ Table out_tablename [ Key out_keycolumn = in_keyexpr ] ] |
[ Variable variable_name ] ]
[ Point [ On | Off ] [ Symbol Symbol_expr ], ]
[ Street_column = Street,
Municipality_column = Municipality,
CountrySubdiv_column = CountrySubdivision,
PostalCode_column = PostalCode,
CountrySecondarySubdiv_column = CountrySecondarySubdivision,
SecondaryPostalCode_column = SecondaryPostalCode,
Placename_column = Placename,
MunicipalitySubdiv_column = MunicipalitySubdivision,
Country_column = Country,
ResultCode_column = ResultCode,
Latitude_column = Latitude,
Longitude_column = Longitude
Columns colname = geocoder_keyname [ , ] ... ]
[ Interactive [ On
[ Max Candidates candidates_expr | All ]
[ CloseMatchesOnly [ On | Off ] ]
| Off [ First | None ] ] ]

```

connection_number – число полученное при создании соединения. См. [Оператор Open Connection](#).

input_tablename синоним открытой таблицы, включая результаты запросов и выборки (selection);

Country_expr – строка, определяющая трехбуквенный код страны по списку ISO;

Street_expr – строка, задающая улицу адреса;

IntersectingStreet_expr – выражение, определяющее улицу, пересекающую улицу *Street_expr*;

Municipality_expr – выражение, в котором задано название муниципального образования;

CountrySubdivision_expr – выражение, в котором задано название территориальной или административной единицы страны самого старшего уровня. Например, в США здесь может быть задано название штата. В Канаде – название провинции.

PostalCode_expr – выражение, в котором задан код почтового индекса;

CountrySecondarySubdiv_expr – выражение, в котором задано название территориальной или административной единицы страны следующего после самого старшего уровня. Например, в США здесь можно задать название графства, а в Канаде – название переписного участка.

SecondaryPostalCode_expr – выражение, в котором задан код почтового индекса второго уровня; В США этому параметру соответствует код дополнительная часть формы ZIP+4 кода почтового индекса ZIP.

Placename_expr – выражение, в котором можно задать название хорошо известной достопримечательности, например здания, которое может иметь несколько адресов;

Street2_expr – выражение, в котором задан второй уровень адреса улицы (в англоязычной практике указывается на отдельной строке почтового отправления);

MunicipalitySubdiv_expr – выражение, в котором задано название муниципального образования младшего уровня;

out_tablename – синоним таблицы, которая будет использоваться для хранения результатов геокодирования;

out_keycolumn – строка, в которой задаётся имя ключевого поля таблицы с результатами, в котором будут храниться "ключевые" значения исходных записей; Используется для выделения записей по происхождению исходных данных геокодирования;

in_keyexpr – выражение из (исходной таблице), значение которого будет вставлено в геокодированную запись;

variable_name – имя переменной с единственным геометрическим примитивом;

Symbol_expr – выражение для символа условного знака, который будет использоваться для отображения точек из колонки с геометрическими примитивами; Подробнее см. [Предложение Symbol](#).

Street_column – синоним имени колонки, в которой будут храниться результаты с названиями улиц;

Municipality_column – строка с именем колонки, в которой будут храниться результаты с названиями муниципальных образований;

CountrySubdiv_column – строка с именем колонки, в которой будут храниться результаты с названиями территориальных или административных образований страны;

PostalCode_column – строка с именем колонки, в которой будут храниться результаты с кодами почтовых индексов;

CountrySecondarySubdiv_column – строка с именем колонки, в которой будут храниться результаты с названиями территориальных или административных образований второго уровня;

SecondaryPostalCode_column – строка с именем колонки, в которой будут храниться результаты с дополнительные коды почтовых индексов;

Placename_column – строка с именем колонки, в которой будут храниться результаты с названиями достопримечательностей;

MunicipalitySubdiv_column – строка с именем колонки, в которой будут храниться результаты с названиями младших муниципальных образований;

Country_column – строка с именем колонки, в которой будут храниться результаты с названиями стран;

ResultCode_column – строка с именем колонки, в которой будут храниться коды результатов порожденные геокодером;

Latitude_column – строка с именем вещественной или десятичной колонки, в которой будут храниться "Широты" результатов;

Longitude_column – строка с именем вещественной или десятичной колонки, в которой будут храниться "Долготы" результатов;

colname – строка с именем колонки для результатов, характерных для используемого геокодера;

geocoder_keyname – строка с названием параметра страны, использованным при геокодировании. Эти параметры описаны в документации каждого геокодера.

candidates_expr – выражение для количества кандидатур, которые будут возвращены в интерактивном сеансе.

Описание

Каждый оператор **Geocode** должен обязательно содержать и выражение **Input**, и выражение **Output**. Параметр, заданный выражением *input_tablename*, является необязательным, однако, если уж таблица не задана, то геокодирование будет проводиться по набору исходных строк (переменных или констант), вследствие чего по каждому запросу будет геокодирован единственный адрес. Параметр, заданный выражением *output_tablename* также является необязательным. Смотрите [Разница между табличной. и нетабличной формой исходных и выходных данных](#) ниже.

Предложение Input

Предложение **Input** необходимо, поскольку для геокодирования требуются некоторые данные на входе.

Аргумент параметра **Country** должен быть указан либо явно, либо в виде ссылки на колонку в *input_tablename*. Когда геокодируются данные для единственной страны, то этот параметр может быть константой, если отсутствуют дополнительные сведения. Необходимо использовать стандартные трехбуквенные коды стран, одобренные ISO.

Список полей *input_table*, по которым следует выполнять геокодирование должен содержать хотя бы одно значение. Чем больше выражений составляют запрос, тем более точными будут результаты геокодирования.

PassThrough – это набор пар "имя/значение", пересылаемых на сервер геокодирования. Эти пары специфичны для серверов геокодирования и документируются для каждого отдельного сервера. For now, this is only used by "Global Geocoder" and "Precisely Global Geocoder".

Предложение Output.

Предложение **Output** обязательно, поскольку при его отсутствии оператора не возвратит ничего.

Into Table определяет, что предложение **Output** ссылается на колонки не защищенной от записи таблицы *output_table*. Если не объявлено явно, это предложение обращается к *input_table*. Обратите внимание, если требуется, чтобы исходные колонки были обновлены, то необходимо объявлять их **ОДНОВРЕМЕННО** и в аргументе *input*, и в аргументе *output*.

Аргумент **Key** используется вместе с **Into Table**. Этим аргументом устанавливается связь между ключевыми колонками исходной таблицы и таблицы на выходе.

Аргумент **Variable** определяет, что геометрические примитивы после выполнения геокодирования будут храниться в переменной заданной аргументом *variable_name*. Обратите внимание, что в случае, когда в качестве исходной используется таблица, и заданы дополнительные параметры для результатов, будет обработана только первая запись, а все остальные пропущены.

Аргумент **Point** указывает, что географический результат геокодирования будет храниться либо в таблице, либо в переменной. Если используется таблица, то для неё должна быть предусмотрена возможность присоединять географическую информацию.

Для того чтобы хранить точки в колонке объектов, используйте либо аргумент **Point**, либо **Point On** (по умолчанию используется вариант *On*). Будет использоваться стандартный символ условного знака. Для того чтобы задать определенный символ условного знака, используйте последовательность аргументов **Point On Symbol** *symbol_expr*. Если не требуется хранить точки в колонке объектов, используйте аргумент **Point Off**. В любом случае: будет создан объект или нет, можно будет хранить координаты *x* и *y* в численных колонках. Для этого, определите такие колонки в аргументах **Latitude** = *latitude_column* и **Longitude** = *longitude_column*.

Остальные аргументы результатов определяют колонки выходной таблицы, в которых заранее известные геокодеры будут хранить полученные результаты. В целом, эти аргументы более своеобразны, чем аргументы исходных параметров. Например, можно геокодировать адрес, используя только коммерческое название "MapInfo" и код почтового индекса "12180". Однако, в результате будет получено гораздо больше данных. Дополнительный аргумент **Columns** позволяет использовать характерные для геокодера данные. Пользователь должен заранее знать имена ключей, определяемых геокодером.

Разница между табличной и нетабличной формой исходных и выходных данных

Оператор **Geocode** может использоваться с исходными и выходными данными в форме таблиц или в не-табличной форме. Если в качестве исходных данных используется таблица, то результаты можно хранить в существующей или новой таблице или в переменных. Если для результатов используется переменная, то обрабатывается только первая запись, а сохраняться будет только географический объект.

Если в качестве исходных, используются данные не в форме таблицы, то аргументы оператора должны быть либо выражениями (не имена колонок), либо переменными, либо константами, а результаты либо могут храниться в таблице, либо будут присвоены переменной.

Предложение *Interactive*

Interactive [**On** | **Off**] – параметры дополнительного аргумента, с помощью которого можно вызывать диалог, когда возвращаются несколько кандидатур каждого адреса. В этом случае, пользователь сможет выбрать, оформить запрос по-другому, пропустить или вовсе отменить обработку исходных данных.

Если сомневаетесь, пропустите обработку данных. Если используется параметр **On** – при возникновении описанной ситуации, появится диалог. Если используется параметр **Off**, и будет выявлено несколько совпадений, то будут использоваться либо первый полученный вариант, либо все варианты будут пропущены. По умолчанию все полученные результаты будут пропущены.

Случай, когда аргумент **Interactive** пропущен, эквивалентен заданию аргумента **Interactive Off None** без дополнительных параметров. Если используется аргумент **Interactive**, по умолчанию параметру присваивается значение **On**.

- **Interactive** – эквивалентно **Interactive On**. Если не предусмотрено никакого значения параметра **Max** – по умолчанию будет возвращено три (3) кандидатуры.
- **Interactive On Max Candidates All** – будут возвращены все кандидатуры
- **Interactive On Max Candidates** $4 * \text{myMBVariable} / 6$ – будут возвращены кандидатуры в количестве, определенном выражением.
- **Interactive Off** – эквивалентно **Interactive Off None**.
- **Interactive Off First** – будет возвращена первая кандидатура списка.

Параметр **CloseMatchesOnly** – передает серверу команду возвращать только близкие совпадения, по правилам заданным для этого сервера. Если параметр **CloseMatchesOnly** установлен в состояние **Off**, – все возвращенные результаты вплоть до количества, определенного параметром аргумента **Max Candidates** вместе со считающимися близкими совпадениями, будут отмечены как близко совпадающие.

Примеры

В следующем примере приведен запрос на геокодирование по таблице `nystreets`, из которой используются колонки "city, Streetname, state и postcode".

```
Geocode connectionHandle Input Table nystreets municipality=city,
street=StreetName, countrysubdivision=state, postcode=zip,
country="usa"
OUTPUT StreetName=street, address=municipality
```

В этом примере приведен запрос на геокодирование по таблице `nystreets` и задан символ условного знака для полученных результатов.

```
Geocode connectionHandle Input Table nystreets street=StreetName,
country="usa"
Output Point Symbol MakeFontSymbol(65, 255 ,24,"MapInfo
Cartographic",32,0), StreetName=street
```


В этом примере запрос содержит аргумент `Interactive` в состоянии `On`, а возвращенные результаты будут помещены в колонку `"street"`.

```
Geocode connectionHandle Input Table nystreets street=StreetName,
country="usa"
Output Point Symbol MakeFontSymbol(65, 255 ,24,"MapInfo
Cartographic",32,0),
StreetName=street Interactive on Max Candidates 5
```

В следующем примере показан вариант запроса на геокодирование без использования таблиц, с сохранением результатов в переменной:

```
Geocode connectionHandle Input street="1 Global View", country="usa",
countrysubdivison="NY", municipality="Troy"
Output Variable outvar
```

См. также:

[оператор Open Connection](#)

Функция GeocodeInfo()

Назначение

Возвращает любой или все атрибуты соединения, для создания которого использовался [оператор Set Connection Geocode](#). Кроме того, **GeocodeInfo()** может использоваться для получения некоторых параметров состояния после выполнения последней команды геокодирования по каждому из соединений. Также можно задавать атрибут, управляющий максимальным количеством адресов, которое допускается единовременно передавать от сервера службе. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GeoCodeInfo( connection_handle, attribute )
```

connection_handle – целое.

attribute – это целое, определяющее, какого типа информация будет возвращена.

Возвращаемая величина

Вещественное (Float), целое (Integer), короткое целое (SmallInt), логическое (Logical), или строка символов (String), в зависимости от параметра атрибута.

Описание

Функция **GeoCodeInfo()** возвращает характеристики соединения, используемые по умолчанию, или параметры, заданные оператором **Set GeoCode**. Подобно многим другим функциям MapBasic этого типа, возвращаемые значения могут зависеть от параметра *attribute*. Коды всех значений перечислены в `MAPBASIC.DEF`.

Значение атрибута	ID	Возвращаемое значение GeoCodeInfo()
GEOCODE_STREET_NAME	1	логическое, определяющее задано или нет совпадение по названиям улиц <code>StreetName</code> .
GEOCODE_STREET_NUMBER	2	логическое, определяющее задано или нет совпадение по номерам улиц <code>StreetNumber</code> .
GEOCODE_MUNICIPALITY	3	логическое, определяющее задано или нет совпадение по названиям муниципальных образований.
GEOCODE_MUNICIPALITY2	4	логическое, определяющее задано или нет совпадение по <code>MunicipalitySubdivision</code> .
GEOCODE_COUNTRY_SUBDIVISION	5	логическое, определяющее задано или нет совпадение по <code>CountrySubdivision</code> .
GEOCODE_COUNTRY_SUBDIVISION2	6	логическое, определяющее задано или нет совпадение по <code>CountrySecondarySubdivision</code> .
GEOCODE_POSTAL_CODE	7	логическое, определяющее задано или нет совпадение по <code>PostalCode</code> .
GEOCODE_DICTIONARY	9	короткое целое (<code>SmallInt</code>), принимающее одно из пяти значений: • <code>DICTIONARY_ALL</code> • <code>DICTIONARY_ADDRESS_ONLY</code> • <code>DICTIONARY_USER_ONLY</code> • <code>DICTIONARY_PREFER_ADDRESS</code> • <code>DICTIONARY_PREFER_USER</code>
GEOCODE_BATCH_SIZE	10	целое, определяющее объем пакета.
GEOCODE_FALLBACK_GEOGRAPHIC	11	логическое, определяющее как геокодеру использовать или нет приведение к географическому центроиду, когда все остальные попытки поиска не дали результата.

Значение атрибута	ID	Возвращаемое значение GeoCodeInfo()
GEOCODE_FALLBACK_POSTAL	12	логическое, определяющее как геокодеру использовать или нет приведение к центроиду почтового участка, когда все остальные попытки поиска не дали результата.
GEOCODE_OFFSET_CENTER	13	вещественное, определяющее расстояние от середины дороги, на котором будут возвращены точки.
GEOCODE_OFFSET_CENTER_UNITS	14	строка символов, определяющая единицы измерения середины дорог.
GEOCODE_OFFSET_END	15	вещественное, определяющее расстояние от конца дороги, на котором будут возвращены точки.
GEOCODE_OFFSET_END_UNITS	16	строка символов, определяющая единицы измерения отступа от концов дорог
GEOCODE_MIXED_CASE	17	логическое, определяющее как MapInfo Pro будет обрабатывать строки возвращенные в смешанном регистре: оставить как есть или переформатировать в верхний регистр. Для отдельных стран нет возможности выполнить эту команду. Этот параметр использует специфичные для каждой страны алгоритмы, в которых определено какая часть адреса должна быть отформатирована в верхнем регистре букв, а какая – в нижнем.
GEOCODE_RESULT_MARK_MULTIPLE	18	логическое, определяющее должна ли MapInfo Pro изменить код результата, возвращенный сервером, добавив к коду результата индикатор того, что результат получен произвольным выбором между несколькими близкими совпадениями. Этот флаг может иметь значение только при следующих обстоятельствах: 1. геокодирование было не интерактивным, поэтому диалог с выбором кандидатур не демонстрировался 2. был установлен флаг обязательного и безусловного выбора первой возвращаемой кандидатуры (см. параметр "First" в описании команды Этим можно заставить MapInfo Pro принять единственный вариант. 3. В ответ на запрос были получены несколько вариантов с похожими на запрашиваемую запись значениями.
GEOCODE_COUNT_GEOCODED	19	целое, определяющее число геокодированных за последнюю операцию значений.
GEOCODE_COUNT_NOTGEOCODED	20	целое, определяющее число не геокодированных за последнюю операцию значений.

Значение атрибута	ID	Возвращаемое значение GeoCodeInfo()
GEOCODE_UNABLE_TO_CONVERT_DATA	21	логическое, определяющее была или нет обновлена после выполнения геокодирования колонка, может быть из-за несовпадения типов. Типичным примером такого несовпадения, является попытка задать числовую колонку для нечисловых почтовых кодов.
GEOCODE_MAX_BATCH_SIZE	22	целое, определяющее максимальное количество записей (адресов), которое может единовременно передаваться службе для обработки.
GEOCODE_PASSTHROUGH	100	целое, определяющее число транзитных элементов, заданное для этого соединения. Для каждой пары существует два элемента. Это значение используется для определения момента прекращения подсчета количества значений без ошибок.
GEOCODE_PASSTHROUGH + <i>n</i>	100	Строка символов, определяющая имя и значение для каждой транзитной пары. <i>n</i> – является значимым вплоть до значения возвращенного вызовом GEOCODE_INFO_PASSTHROUGH.

Пример

Этот фрагмент кода MapBasic печатает в окне сообщений MapInfo Pro сведения от сервиса Envinsa Location Utility Constraints:

```

Include "MapBasic.Def"
declare sub main
sub main
dim iConnect as integer
Open Connection Service Geocode Envinsa
    URL
"http://envinsa_server:8066/LocationUtility/services/LocationUtility"
    User "john"
    Password "green"
into variable iConnect
Print "Geocode Max Batch Size: " +
GeoCodeInfo(iConnect,GEOCODE_MAX_BATCH_SIZE)
end sub

```

См. также:

оператор Open Connection, оператор Set Connection Geocode

Оператор Get

Назначение

Читает данные из файла, открытого в бинарном режиме (BINARY) или в режиме произвольного доступа (RANDOM).

Синтаксис

```
Get [#] filenum, [ position ], var_name
```

filenum – целое число, присвоенное файлу **оператором Open File**.

Position – значение LargeInt, обозначающее нужное положение файла, из которого требуется прочитать.

var_name – имя переменной, в которой MapBasic сохранит результат.

Описание

Оператор **Get** читает данные из открытого файла. Способ чтения оператором **Get** определяется режимом, который задает **оператор Open File** при открытии файла.

Если в **операторе Open File** задан режим **Random**, то в операторе **Get** параметр *Position* определяет строку данных, с которой начинается чтение. Сразу после открытия файла позиция для записи – это первая запись файла. По мере чтения оператором **Get** позиция считывания автоматически наращивается, и нет нужды каждый раз менять значение *Position*. Однако, если требуется перейти не к следующей по порядку строке, то придется явно задать значение *Position*.

Если в **операторе Open File** задан режим **Binary**, одним оператором **Get** читается одно значение *var_name*. То, как читаются данные, зависит от порядка задания байтов в исходном файле и типа переменной *var_name*. Если переменная имеет тип Integer, то прочитываются 4 байта из двоичного файла и превращаются в переменную MapBasic. Переменные заполняются при чтении по следующим правилам:

Тип переменной	Хранение в файле
Логическое – Logical.	Однobaйтовое значение: либо 0, либо другое ненулевое число.
Короткое целое – SmallInt	Двухбайтовое значение, целое число.

Тип переменной	Хранение в файле
Целое – Integer	Четырехбайтовое значение, целое число.
Вещественное – Float	Восьмибайтовое число в формате IEEE.
Строка – String	Длина строки плюс один байт для нулевого значения, обозначающего конец строки.
Дата – Date	4 байта: короткое целое (SmallInt) для года, один байт для месяца и один байт для дня.
Другие типы	Чтение не производится.

В режиме чтения двоичных кодов (BINARY) параметр *position* используется для задания позиции считывания на определенное значение смещения в файле. Сразу после открытия файла значение *position* устанавливается на единицу (начальный байт файла). По мере выполнения оператора Position, каждый новый оператор **Get** продолжает чтение с того места, на котором оно остановилось в прошлый раз. Если предложение Position опущено оператор **Get** начинает чтение с того места, где завершилось предыдущее чтение.

Примечание: В операторе **Get** нужно поставить две запятые, если параметр *position* опущен.

Если файл был открыт в режиме BINARY оператор **Get** не может заполнять строку (переменную типа String) неопределенной длины; любая переменная типа String в операторе **Get** должна иметь определенную длину.

См. также:

[оператор Open File](#), [оператор Put](#)

Функция GetCurrentPath()

Назначение

Возвращает дисковый маршрут, используемый в диалоге **Файл MapInfo Pro**. Каждый служебный файл или таблица MapInfo имеет свой маршрут доступа.

Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetCurrentPath$( current_path_id )
```

current_path_id принимает следующие значения:

```
PREFERENCE_PATH_TABLE (0)
PREFERENCE_PATH_WORKSPACE (1)
PREFERENCE_PATH_MBX (2)
PREFERENCE_PATH_IMPORT (3)
PREFERENCE_PATH_SQLQUERY (4)
PREFERENCE_PATH_THEMETHEMPLATE (5)
PREFERENCE_PATH_MIQUERY (6)
PREFERENCE_PATH_NEWGRID (7)
PREFERENCE_PATH_CRYSTAL (8)
PREFERENCE_PATH_GRAPHSSUPPORT (9)
PREFERENCE_PATH_REMOTETABLE (10)
PREFERENCE_PATH_SHAPEFILE (11)
PREFERENCE_PATH_WFSTABLE (12)
PREFERENCE_PATH_WMSTABLE (13)
```

Возвращаемая величина

Строка – String

Описание

По заданному ID-номеру специального каталога MapInfo функция GetCurrentPath\$() возвращает маршрут к этому каталогу. Ниже приводится пример поиска маршрута к каталогу, в который MapInfo помещает новые таблицы.

Примечание: Возвращаемое значение меняется и в случаях, когда пользователь меняет маршруты в диалоговом окне и закрывает диалог с подтверждением (отмена во внимание не принимается).

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
include "mapbasic.def"
declare sub main
sub main
dim sMiPrfFile as string
sMiPrfFile = GetCurrentPath$( PREFERENCE_PATH_WORKSPACE)
Print sMiPrfFile
end sub
```

См. также:

функция `GetPreferencePath$ ( )`

Функция `GetCurrentWorkspacePath$()`

Назначение

Возвращает маршрут директории файла текущего рабочего набора. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetCurrentWorkspacePath$ ( )
```

Возвращаемая величина

Строка

Описание

Эта функция позволяет отображать маршрут директории, в которой находится файл открытого рабочего набора. Если файл рабочего набора не открыт, то выполнение этой функции возвращает пустую строку "".

Функция `GetDate( )`

Назначение

Возвращает компоненту даты **DateTime**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetDate ( DateTime )
```

Возвращаемая величина

Date

Пример

В этом примере настраивается дата и время строкой формата dd/mm/уууу (где dd – день, mm – месяц и уууу – год) и hh:mm:ss.fff (где hh – часы, ss – секунды, mm – минуты и fff – десятичные доли в миллисекундах), которая затем преобразуется с помощью функции `GetDate()`.

```
dim dtX as datetime
dim Z as date
dtX = "03/07/2007 12:09:09.000 AM"
Z = GetDate(dtX)
Print FormatDate$(Z)
```

См. также:

[функции даты и времени](#), [функция GetTime\(\)](#)

Функция GetFolderPath\$()

Назначение

Возвращает путь к специальным папкам MapInfo Pro или Windows. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetFolderPath$( folder_id )
```

folder_id – может принимать одно из следующих значений:

```
FOLDER_MI_APPDATA (-1)
FOLDER_MI_LOCAL_APPDATA (-2)
FOLDER_MI_PREFERENCE (-3)
FOLDER_MI_COMMON_APPDATA (-4)
FOLDER_APPDATA (26)
FOLDER_LOCAL_APPDATA (28)
FOLDER_COMMON_APPDATA (35)
FOLDER_COMMON_DOCS (46)
FOLDER_MYDOCS (5)
FOLDER_MYPICS (39)
```

Возвращаемая величина

Строка – String

Описание

Получив идентификатор одной из определенных папок MapInfo или Windows функция **GetFolderPath\$ ()** возвращает путь к этой папке. Примером специальной папки Windows является папка My Documents. Другим примером специальной папки MapInfo Pro является папка с настройками.

Расположение многих из этих папок варьируется в различных версиях Windows. Оно также может отличаться для отдельных пользователей. Заметим, что FOLDER_MI_APPDATA (-1), FOLDER_MI_LOCAL_APPDATA (-2) и FOLDER_MI_COMMON_APPDATA (-4) могут не существовать. Перед попыткой войти в одну из этих папок, убедитесь в их существовании, для чего используется **функция FileExists()**. FOLDER_MI_PREFERENCE (-3) существует всегда.

Идентификаторы папок, начинающиеся с FOLDER_MI, возвращают путь к папкам, специфическим для MapInfo Pro. Остальные возвращают путь для папок Windows и соответствуют названиям, определенным для функции WIN32 API, а именно, SHGetFolderPath. Самые общие из этих названий можно использовать в приложениях MapBasic. Любые из названий, возможных для SHGetFolderPath, будут работать и с функцией **GetFolderPath\$ ()**.

Пример

```
include "mapbasic.def"
declare sub main
sub main
dim sMiPrfFile as string
sMiPrfFile = GetFolderPath$(FOLDER_MI_PREFERENCE)
Print sMiPrfFile
end sub
```

См. также:

функция LocateFile()

Функция GetGridCellValue()

Назначение:

Определяет значение в ненулевой (non-null) ячейке сетки, задающей поверхность.

Синтаксис

```
GetGridCellValue( table_id, x_pixel, y_pixel )
```

table_id – либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0. таблица должна описывать сетку (grid).

x_pixel – целочисленный номер X-координаты ячейки сетки. Нумерация пикселей начинается с 0. Максимальное значение координаты пикселя (*pixel_width* – 1), можно выяснить при помощи вызова:

```
RasterTableInfo(...RASTER_TAB_INFO_WIDTH) .
```

y_pixel – целочисленный номер Y-координаты ячейки сетки. Нумерация пикселей начинается с 0. Максимальное значение координаты пикселя (*pixel_height* – 1), можно выяснить при помощи вызова:

```
RasterTableInfo(...RASTER_TAB_INFO_HEIGHT) .
```

Возвращаемая величина

Величина типа Float, представляющая значение в заданной ячейке, если только ячейка не имеет атрибут null. Предварительно должна быть вызвана функция IsGridCellNull() для определения наличия в ячейке значения или атрибута null.

Получение значений ячейки регулярной сети по координатам

Для того чтобы получить значения ячеек регулярной сети в координатах, используйте программу-инструмент GridTools.MBX, которая установлена в папке MapInfo\Professional\Tools. С помощью этой программы-инструмента можно получить информацию о значениях регулярной сети по географическим положению.

Как воспользоваться MapBasic, чтобы получить значений ячеек регулярной сети по координатам, смотрите следующий пример. Географические координаты (x,y) конвертируются в координатную систему таблицы регулярной сети. Затем эти координаты используются для вычисления строки и столбца регулярной сети.

- Если в результате выполнения первой строки получено логическое значение "истина" (true, then) (строка 2: оператор Then), параметр **Перепроецирование** (задаётся в диалоге **Режимы окна карты** в разделе **Обработка растра**) устанавливается в режим **Нет**. В этом случае проекция окна **Карта** зависит от параметров регулярной сети, поэтому можно использовать те же самые, уже определённые значения x и y.
- Если в результате выполнения первой строки получено значение "не истина" (not true), то (строка 5: оператор Else) проекция окна **Карта** пересчитывается, что приведёт к тому, что координаты (x,y) пересчитываются в координатную систему таблицы регулярной сети прежде, чем выполнять математические операции с регулярной сетью.

Оператор `Else` (строка 10) преобразует x/y значения из координатной системы окна карты в координатную систему таблицы регулярной сети.

```
If (MapperInfo(MapWindowID, MAPPER_INFO_REPROJECTION) = "None") And
iVisibleImages = 1
Then
    x_grid = x
    y_grid = y
Else
    If MapperInfo(MapWindowID, MAPPER_INFO_COORDSYS_CLAUSE) =
TableInfo(sTable, TAB_INFO_COORDSYS_CLAUSE) Then
        x_grid = x
        y_grid = y
    Else
        csysConverter = CreatePoint(x, y)
        Set CoordSys Table sTable
        x_grid = CentroidX(csysConverter)
        y_grid = CentroidY(csysConverter)
    End If
End If
```

См. также:

оператор `Create Grid`, `GridTableInfo()`, функция `IsGridCellNull()`, функция `OverlayNodes()`, функция `RasterTableInfo()`

Функция `GetLicenseKind( )`

Назначение

Возвращает значение, представляющее тип используемой лицензии MapInfo Pro. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetLicenseKind( )
```

Возвращаемая величина

Короткое целое (`SmallInt`):

Описание

Значение возвращается в виде целого числа. Возможные возвращаемые значения вместе со смысловым описанием приводятся в таблице ниже. Возвращаемые значения, приведённые в левом столбце, определяются в файле `MAPBASIC.DEF`.

Возвращаемая величина:	ID	Значение
<code>LICENSE_KIND_NOLICENSE</code>	0	Нет лицензии
<code>LICENSE_KIND_REGULAR</code>	1	Обычная лицензия
<code>LICENSE_KIND_ADVANCE</code>	2	Расширенная лицензия

Пример:

```

If GetLicenseKind() = LICENSE_KIND_ADVANCE Then
  Print "Advanced License"
End If

If GetLicenseKind() = LICENSE_KIND_NOLICENSE Then
  Print "No License"
End If

End Sub

If GetLicenseKind() = LICENSE_KIND_REGULAR Then
  Print "Regular License"
End If

```

Функция GetMetadata()

Назначение

Извлекает из таблицы метаданные. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetMetadata$( table_name, key_name )
```

table_name – имя открытой таблицы (например, "World") или её синонима (например, World);

key_name – строковая величина, представляющая собой имя ключа метаданных.

Возвращаемая величина

Величина типа String, строка длиной до 239 байт. Если ключ не существует или соответствующее значение пусто, то MapInfo Pro вернет пустую строку.

Описание

Эта функция возвращает метаданные из таблицы. Более подробное описание процесса извлечения метаданных из таблицы см. в разделе: [Оператор Metadata](#) или *Руководство пользователя MapBasic*.

Пример

Если в таблице PARCELS есть ключ метаданных "\Copyright", то следующий оператор распечатывает соответствующее значение метаданных:

```
Print GetMetadata$(Parcels, "\Copyright")
```

См. также:

[оператор Metadata](#)

Функция GetPreference()

Назначение

Возвращает значение заданного параметра. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetPreference ( segment, ordinal )
```

segment – положительное целое, представляющее одну из опций или параметров, которые можно задавать через интерфейс пользователя. Имена сегментов похожи, но не идентичны именам, используемым в интерфейсе пользователя.

ordinal – положительное целое кодовое число, обозначающее параметр, входящий в сегмент, который требуется получить.

Возвращаемая величина

Возвращаемый тип зависит от запрошенного параметра.

Описание

Учитывая сегмент и порядковый номер в пределах этого сегмента, возвращается параметр. Возвращаемый тип зависит от запрошенного параметра.

Параметр *attribute* определяет вид возвращаемой функцией информации. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код сегмента	ID	Значение
PREFERENCE_SYSTEM	0	Системные настройки.
PREFERENCE_TABLES	1	Предпочтения
PREFERENCE_BROWSER	3	Настройки окна списка
PREFERENCE_MAPPER	4	Настройки окна карты
PREFERENCE_INTERNATIONAL	6	Порядок записи адреса
PREFERENCE_PATHS	11	Каталоги MapInfo
PREFERENCES_LEGEND	12	Настройки окна легенды
PREFERENCES_OUTPUT	13	Настройки режимов вывода
PREFERENCES_PRINTER	14	Настройки принтера
PREFERENCES_WEBSERVICES	15	Настройки Web-служб
PREFERENCES_IMAGE_PROCESSING	16	Настройки обработки раstra
PREFERENCES_LAYOUT	17	Настройки окна отчёта
PREFERENCES_PDF	18	Настройки PDF-принтера
PREFERENCES_PERFORMANCE	19	Настройки производительности

код сегмента	ID	Значение
PREFERENCES_MANAGED_WORKSPACE	50	Настройки рабочего набора

Параметр *attribute* определяет вид возвращаемой функцией информации. Имена для режимных кодов определены в файле MAPBASIC.DEF.

Возвращаются целочисленные значения `SmallInteger`, если не задано иное.

Примечание: Некоторые режимы опущены, поскольку задаваемые ими целочисленные параметры более не используются или не поддерживаются.

ordinal	ID	GetPreference() возвращает
PREF_SYSTEM_COLORDEFAULT	0	0 = монитор, 1 = чёрно-белый, 2 = цветной
PREF_SYSTEM_AUTOSAVEWORKSPACE	3	Автоматическое сохранение рабочих наборов: 0=ON, 1=OFF
PREF_SYSTEM_COPYTEXTCLIPBOARD	7	Копия текста в буфер обмена: 0=ON, 1=OFF
PREF_SYSTEM_COPYGRAPHICSClipboard	8	Копия BMP в буфер обмена: 0=ON, 1=OFF
PREF_SYSTEM_PAPERUNITS	9	11 = пункты (Points) 12 = циперо (Picas) 2 = дюймы 5 = миллиметры 6 = сантиметры
PREF_SYSTEM_UNDO_COUNT	10	Количество объектов в операции отмены
PREF_SYSTEM_SUBSTITUTETTFONT	12	Отображение символов в формате до версии 4
PREF_SYSTEM_COPY_EMF_CLIPBOARD	13	Копия EMF в буфер обмена: 0=ON, 1=OFF
PREF_SYSTEM_COPY_OLEOBJECT	14	Копия картографических объектов MapInfo Map Object в буфер обмена: 0=ON, 1=OFF — отключено в x64
PREF_SYSTEM_SAVEQUERYWORKSPACE	18	Сохранение запросов в рабочих наборах: 0=ON, 1=OFF

ordinal	ID	GetPreference() возвращает
PREF_SYSTEM_DATEWINDOW	19	Короткое целое число – SmallInt: 1 = использовать текущее столетие, иначе – значение года
PREF_SYSTEM_UNDO_SIZE	20	объём памяти в байтах, выделяемый для операций отмены
PREF_SYSTEM_RESOLUTION	21	Разрешение при экспорте в dpi
PREF_SYSTEM_VMOPTION	22	0 = GRID, 1= Raster, 2 = стандартно
PREF_SYSTEM_DISTANCEUNITS	23	Возвращает константу, определяющую единицы измерения расстояний 2 = дюймы 30 = линки 3 = футы 8 = геодезические футы США 4 = ярды 32 = роды 31 = чейны 0 = мили 9 = морские мили 5 = миллиметры 6 = сантиметры 7 = метры 1 = километры

ordinal	ID	GetPreference() возвращает
PREF_SYSTEM_AREAUNITS	24	Возвращает константу, определяющую единицы измерения площадей 16 = квадратные дюймы 33 = квадратные линки 17 = квадратные футы 18 = квадратные ярды 35 = квадратные роды 36 = перчи 34 = квадратные чейны 37 = руды 28 = акры 14 = квадратные мили 23 = квадратные морские мили 19 = квадратные миллиметры 20 = квадратные сантиметры 21 = квадратные метры 29 = гектары 15 = квадратные километры
PREF_SYSTEM_COPY_OLEOBJECT_NEW	25	Копия EMF в буфер обмена: 0=ON, 1=OFF
PREF_SYSTEM_COPY_ENHANCED_EMF_TYPE	26	0 = EMF, 1= EMF+, 2 = EMF+Dual
PREF_SYSTEM_CROSSHAIRCURSOR	28	138 = перекрестие среднего размера, 164 = большое перекрестие, 165 = стандартное
PREF_SYSTEM_COPY_WMF_CLIPBOARD	29	Копия WMF в буфер обмена: 0=ON, 1=OFF
PREF_SYSTEM_TRUECOLORCURSORS	30	Разрешение на использование режима цветности true color для курсоров: 0=ON, 1=OFF
PREF_SYSTEM_WRITEWINDOWSTOWORKSPACE	31	Сохранение положения инструментальных окон в рабочих наборах: 0=ON, 1=OFF
PREF_SYSTEM_MULTIPLETOUCH	32	Поддержка сенсорного экрана: 0=ON, 1=OFF

ordinal	ID	GetPreference() возвращает
PREF_SYSTEM_DEFAULTFILEFORMAT	33	Короткое целое — SmallInt, обозначающее стандартный тип формата для новых таблиц. 5 для MapInfo (Native) 14 для расширенной версии MapInfo Extended (NativeX)
PREF_SYSTEM_DEFAULTCHARSETFORNATIVE	34	Строковый результат, представляющий стандартный набор символов, используемый при создании расширенных таблиц MapInfo Extended (например, UTF-8 или WindowsLatin).
PREF_SYSTEM_WRITEUNICODEDATA	35	Короткий целый результат (SmallInt) – 1, если включен режим Использовать Юникод для раб.наборов и файлов Tab .

Числовые определители режимов рабочих наборов	ID	GetPreference возвращает
PREF_WKS_PROMPTTOSAVE	1	0 = OFF, 1=ON
PREF_WKS_PROMPTTOSAVEOPTION	2	0 = автоматически, 1 = только, если WOR-файл открыт

Числовые определители режимов таблиц	ID	GetPreference возвращает
PREF_TABLES_FILEOPENWHAT	0	Последний использованный режим показа
PREF_TABLES_OPENCHOICESCREEN	1	Диалог QuickStart
PREF_TABLES_CONNNNAME	2	Строка, величина типа String: Стандартное имя соединения СУБД
PREF_TABLES_CONNDRIVER	3	Строка, величина типа String: Имя драйвера СУБД
PREF_TABLES_CONNSTRING	4	Строка, величина типа String: Имя соединения СУБД

Числовые определители режимов списков	ID	GetPreference возвращает
PREF_BROWSER_RDTSORTORDER	0	Порядок сортировки при районировании 0=MRU, 1=в алфавитном порядке, 2=без определённого порядка
PREF_BROWSER_ALTROWCOLORRED	1	Короткое целое число – SmallInt: Красный цвет компоненты RGB
PREF_BROWSER_ALTROWCOLORGREEN	2	Короткое целое число – SmallInt: Зелёный цвет компоненты RGB
PREF_BROWSER_ALTROWCOLORBLUE	3	Короткое целое число – SmallInt: Синий цвет компоненты RGB

Числовые определители режимов карт	ID	GetPreference возвращает
PREF_MAPPER_PRESERVESCALEONRESIZE	0	0=уместить в карте, 1 = сохранить масштаб
PREF_MAPPER_HIGHLIGHT_BRUSH	2	Предложение, определяющее вид штриховки для выборки
PREF_MAPPER_SNAPTONODES	3	Привязка к узлам: 0=Off, 1=ON
PREF_MAPPER_MOVEDUPLICATENODES	4	0 = ни одного слоя, 1= тот же слой
PREF_MAPPER_PIXELTOLERANCE	5	Короткое целое число – SmallInt: допустимое отклонение в пикселях
PREF_MAPPER_TARGET_PEN	7	Предложение, определяющее вид линии изменяемых объектов
PREF_MAPPER_TARGET_BRUSH	8	Предложение, определяющее вид штриховки изменяемых объектов
PREF_MAPPER_HIGHLIGHT_PEN	9	Предложение, определяющее вид линии для выборки
PREF_MAPPER_SCROLLBARS	10	Показ полос прокрутки: 0=Off, 1=ON

Числовые определители режимов карт	ID	GetPreference возвращает
PREF_MAPPER_SAVECOSMETIC	11	Предупреждение о потере косметических объектов: 0=Off,1=ON
PREF_MAPPER_SAVELABELS	12	Предупреждение о потере подписей на карте: 0=Off,1=ON
PREF_MAPPER_SAVETHEMATIC	13	Предупреждение о потерях на тематических слоях: 0=Off,1=ON
PREF_MAPPER_SHOWINFOTIPS	14	Показ подсказок на слоях: 0=Off,1=ON
PREF_MAPPER_DISTANCETYPE	15	Измерения проводятся: '0=на сфере, 1=декартовый
PREF_MAPPER_SHOWDEGREESMINUTESSECONDS	16	Короткое целое число – SmallInt: Возвращает определители MAPPER_INFO_DISPLAY_DMS
PREF_MAPPER_LASTTHEMETYPE	17	Строка, величина типа String: Последний использованный тематический тип
PREF_MAPPER_GRIDOUTPUTFORMATHANDLER	18	Строка, величина типа String: Новые регулярные поверхности
PREF_MAPPER_CLIPREGIONTYPE	19	Короткое целое число – SmallInt: Возвращает определители MAPPER_INFO_CLIP_TYPE
PREF_MAPPER_3DHARDWARE	20	Включение аппаратного ускорения для окон трёхмерных карт, 0=Off,1=ON
PREF_MAPPER_AUTORASTERZOOMLAYER	21	Масштабирование слоёв с растрами: 0=Off,1=ON
PREF_MAPPER_AUTOGRIDZOOMLAYER	22	Масштабирование слоёв с регулярными поверхностями: 0=Off,1=ON
PREF_MAPPER_POINT_HIGHLIGHT_BRUSH	23	Предложение, определяющее вид штриховки для выборок с группами точек

Числовые определители режимов карт	ID	GetPreference возвращает
PREF_MAPPER_POINT_TARGET_BRUSH	24	Предложение, определяющее вид штриховки для изменяемых объектов с группами точек
PREF_MAPPER_AUTOSCROLL	25	Автоматическая прокрутка: 0=Off, 1=ON
PREF_MAPPER_DEFAULTPROJECTION	26	Строка, величина типа String: Режимы, определяющие проекции таблиц
PREF_MAPPER_DEFAULTMBPROJECTION	27	Строка, величина типа String: Режимы, определяющие проекции таблиц в сеансе
PREF_MAPPER_DISPLAYSNAPRADIUS	28	Короткое целое число – SmallInt: Значение радиуса привязки
PREF_MAPPER_CARTOGRAPHICSCALE	29	Картографический масштаб: 0=Off, 1=ON
PREF_MAPPER_DISPLAYLAYERCONTROLTOOLTIPS	30	Подсказки инструментов управления слоями: 0=Off, 1=ON
PREF_MAPPER_AUTONODETOLERANCE	31	Короткое целое число – SmallInt: Значение допусков автоматической привязки к узлам
PREF_MAPPER_DRAWLAYERSUNDERTHEMES	32	Прорисовка слоёв под тематическими: 0=Off, 1=ON
PREF_MAPPER_READDEFAULTTHEMEMETADATA	33	Считывание стандартных метаданных при открытии таблиц. 0=Off, 1=ON
PREF_MAPPER_FINDSELECTIONAFTERPASTE	34	Перемещение к выборке после добавления объекта: 0=Off, 1=ON
PREF_MAPPER_ZOOMONFINDSELECTION	35	Масштабирование по границам выбранных объектов: 0=Off, 1=ON
PREF_MAPPER_ALPHAHIGHLIGHT	36	Значение полупрозрачности, где 255 =100% непрозрачный, 0= прозрачный
PREF_MAPPER_ZOOMMOUSEWHEELONCURSOR	37	Масштабирование колесом мыши вокруг положения курсора: 0=Off, 1=ON

Числовые определители режимов карт	ID	GetPreference возвращает
PREF_MAPPER_ADORNAUTOMATICSCALING	38	Использование автоматического определения размеров для визуальных эффектов: 0=Off, 1=ON
PREF_MAPPER_ADORN_BACKGROUND_BRUSH	39	Предложение о форме штриховки фона визуальных эффектов
PREF_MAPPER_ADORN_BACKGROUND_PEN	40	Предложение о форме штриха линии фона визуальных эффектов

Числовые определители международных режимов	ID	GetPreference возвращает
PREF_INTERNATIONAL_ADDRMATCH	0	0=номер дома ПЕРЕД названием улицы, 1=номер дома ПОСЛЕ названия улицы
PREF_INTERNATIONAL_INSET	1	Значение вставки в единицах измерения расстояний
PREF_INTERNATIONAL_OFFSET	2	Значение смещения в единицах измерения расстояний
PREF_INTERNATIONAL_DISTUNIT	3	Возвращает определитель единиц измерения расстояний. Смотрите: PREF_SYSTEM_DISTANCEUNITS
PREF_INTERNATIONAL_ENABLEDISTANCESPECIFIED	4	Короткое целое — SmallInt значения процентов внутреннего смещения адреса
PREF_INTERNATIONAL_INSET_DIST	5	Короткое целое – SmallInt расстояния внутреннего смещения
PREF_INTERNATIONAL_INSET_DIST_DISTUNIT	6	Возвращает определитель единиц измерения расстояния внутреннего смещения. Смотрите: PREF_SYSTEM_DISTANCEUNITS
PREF_INTERNATIONAL_USER_CLEAN	7	Подстановки из файла MAPINFOW.ABB: 0=Off, 1=ON

Числовые определители международных режимов

ID

GetPreference возвращает

PREF_INTERNATIONAL_CLOSE_ADDR	8	Использование номера ближайшего адреса: 0=Off,1=ON
PREF_INTERNATIONAL_ALT_BDY	9	Использование совпадений в других пределах: 0=Off,1=ON

Числовые определители системных адресов

ID

GetPreference возвращает

PREF_PATH_MITABLEPATH	0	Каталог для новых таблиц
PREF_PATH_MIWORKSPACEPATH	1	Каталог для новых рабочих наборов
PREF_PATH_MIPROGRAMPATH	2	Каталог для MapBasic-программ
PREF_PATH_MIDATAPATHS	3	Альтернативный адрес поиска, если PREF_PATH_USETRAVERSAL =1
PREF_PATH_USETRAVERSAL	4	Короткое целое число – SmallInt: 0= без альтернативного адреса поиска, 1 = использовать альтернативный адрес поиска
PREF_PATH_CUSTOMSYMBOLPATH	5	Каталог с растровыми символами
PREF_PATH_IMPORTPATH	6	Каталог для импорта файлов
PREF_PATH_SQLQUERYPATH	7	Каталог SQL-запросов
PREF_PATH_THEMETEMPLATEPATH	8	Каталог для тематических шаблонов
PREF_PATH_MIQUERYPATH	9	Каталог для сохранённых запросов
PREF_PATH_NEWGRIDPATH	10	Каталог для новых файлов регулярных поверхностей
PREF_PATH_REMOTETABLEPATH	13	Каталог для таблиц СУБД

Числовые определители системных адресов	ID	GetPreference возвращает
PREF_PATH_TABLESFROMSHAPEFILES	14	Каталог для таблиц шейп-файлов
PREF_PATH_TABLESFROMWFS	15	Каталог для WFS-таблиц
PREF_PATH_TABLESFROMWMS	16	Каталог для WMS-таблиц
PREF_PATH_LAYOUTIMAGES	17	Каталог изображений отчётов

Числовые определители легенд	ID	GetPreference возвращает
PREF_LEGEND_FNTTITLE	0	Предложение Font, определяющее шрифт заголовка легенды
PREF_LEGEND_FNTSUBTITLE	1	Предложение Font, определяющее шрифт подзаголовка легенды
PREF_LEGEND_FNTSTYLE	2	Предложение, определяющее категорию шрифта легенды
PREF_LEGEND_PENBORDER	3	Предложение, определяющее категорию стиля линии легенды
PREF_LEGEND_TITLEPATTERN	4	Предложение, определяющее стиль штриховки фона заголовка легенды
PREF_LEGEND_SUBTITLEPATTERN	5	Предложение, определяющее стиль штриховки фона подзаголовка легенды
PREF_LEGEND_STYLEPATTERN	6	Предложение, определяющее стиль штриховки категории легенды
PREF_LEGEND_STYLESIZE	7	Размер условных знаков легенды 0=Малые условные знаки, 1=крупные условные знаки
PREF_LEGEND_PROMPTSAVEWORKSPACE	8	Сообщение об изменениях легенд в рабочих наборах: 0=Off, 1=On

Числовые определители легенд	ID	GetPreference возвращает
PREF_LEGEND_CREATEPRE1150LEGENDS	9	Создание легенд версии 1150.0 Не используется в x64
PREF_LEGEND_PAPERUNITS	10	Возвращается PREF_SYSTEM_PAPERUNITS
PREF_LEGEND_SYNCDISPLAY	11	Размера образца увеличивается для соответствия шрифту. 0=Off, 1=On
PREF_LEGEND_LINEWIDTH	12	Короткое целое – SmallInt, задающее толщину линии условного знака
PREF_LEGEND_REGIONWIDTH	13	Короткое целое – SmallInt, задающее длину площадного условного знака
PREF_LEGEND_REGIONHEIGHT	14	Короткое целое SmallInt, задающее высоту площадного условного знака
PREF_LEGEND_DISPLAYGRID	15	Показ сетки: 0=Off, 1=On
PREF_LEGEND_SNAPGRID	16	Показ сетки линий привязки: 0=Off, 1=On
PREF_LEGEND_AUTOUPDATE	17	Автоматическое обновление легенды: 0=Off, 1=On
PREF_LEGEND_GRIDCOLORALPHA	18	Короткое целое число – SmallInt: где 255 =100% непрозрачный, 0= прозрачный
PREF_LEGEND_GRIDCOLORRED	19	Короткое целое число – SmallInt: Компонент красного цвета RGB-значения
PREF_LEGEND_GRIDCOLORGREEN	20	Короткое целое число – SmallInt: Компонент зелёного цвета RGB-значения
PREF_LEGEND_GRIDCOLORBLUE	21	Короткое целое число – SmallInt: Компонент синего цвета RGB-значения
PREF_LEGEND_GUIDECOLORALPHA	22	Короткое целое число – SmallInt: где 255 =100% непрозрачный, 0= прозрачный

Числовые определители легенд	ID	GetPreference возвращает
PREF_LEGEND_GUIDECOLORRED	23	Короткое целое число – SmallInt: Компонент красного цвета RGB-значения
PREF_LEGEND_GUIDECOLORGREEN	24	Короткое целое число – SmallInt: Компонент зелёного цвета RGB-значения
PREF_LEGEND_GUIDECOLORBLUE	25	Короткое целое число – SmallInt: Компонент синего цвета RGB-значения
PREF_LEGEND_AUTOFONTSIZESETTING	26	Увеличение размера шрифта условного знака: 0=Off, 1=On

Числовые определители, относящиеся к выводу на печать и экспорту	ID	GetPreference возвращает
PREF_OUTPUT_USETRUECOLORDISPLAY	0	Полноцветный растр True Color: 0=Off, 1=On
PREF_OUTPUT_DITHERMETHODDISPLAY	1	1=полутона, 2 = диффузия цветов
PREF_OUTPUT_PRINTMETHOD	2	0=напрямую на устройство, 1=используя EMF-метод, 2=OSBM
PREF_OUTPUT_PRINTBORDER	3	Рамка при печати: 0=Off, 1=On
PREF_OUTPUT_TRANSPWHENPRINTVECTOR	5	Внутренняя обработка при печати прозрачных векторов: 0=Off, 1=On
PREF_OUTPUT_USETRUECOLORPRINT	6	Полноцветная печать в режиме True Color: 0=Off, 1=On
PREF_OUTPUT_DITHERMETHODPRINT	7	Полутонная печать: 1=полутона, 2 = диффузия цветов
PREF_OUTPUT_EXPORTBORDER	8	Рамка при экспорте: 0=Off, 1=On
PREF_OUTPUT_TRANSPWHENEXPORTRASTER	9	Внутренняя обработка при экспорте прозрачных растровых изображений: 0=Off, 1=On

Числовые определители, относящиеся к выводу на печать и экспорту	ID	GetPreference возвращает
PREF_OUTPUT_TRANSPWHENEXPORTVECTOR	10	Внутренняя обработка при экспорте прозрачных векторов: 0=Off, 1=On
PREF_OUTPUT_USETRUECOLOREXPORT	11	Экспорт в полноцветном режиме True Color: 0=Off, 1=On
PREF_OUTPUT_DITHERMETHODEXPORT	12	Экспорт в полутонах: 1=полутон, 2 = диффузия цветов
PREF_OUTPUT_ROPTRANSPWHENPRINTINGRASTER	13	Использование ROP-метода при печати растровых изображений: 0=Off, 1=On
PREF_OUTPUT_ROPTRANSPWHENEXPORTINGRASTER	14	Использование ROP-метода при экспорте растровых изображений: 0=Off, 1=On
PREF_OUTPUT_SCALEPATTERNSPRINT	15	Масштабирование штриховок при печати: 0=Off, 1=On
PREF_OUTPUT_USEANTIALIASING	16	Использование сглаживания: 0=Off, 1=On
PREF_OUTPUT_ANTIALIASINGTHRESHOLD	17	Значение допуска, используемого при сглаживании
PREF_OUTPUT_ANTIALIASINGMASKSIZE	18	Значение размера маски при сглаживании: 0=Off, 1=On
PREF_OUTPUT_ANTIALIASINGFILTER	19	Смотрите определители возвращаемых параметров WIN_INFO_EXPORT_FILTER
PREF_OUTPUT_DISPLAYPOLYGONMODE	20	Использование полигонального режима при отображении: 0=Off, 1=On
PREF_OUTPUT_PRINTINGPOLYGONMODE	21	Использование полигонального режима при печати: 0=Off, 1=On
PREF_OUTPUT_EXPORTINGPOLYGONMODE	22	Использование полигонального режима при экспорте: 0=Off, 1=On
PREF_OUTPUT_SUBDIVIDEPRINTING	23	Разделение на части при выводе на печать: 0=Off, 1=On

Числовые определители, относящиеся к выводу на печать и экспорту	ID	GetPreference возвращает
PREF_OUTPUT_CLARIFYTEXTTOOVERWYSIWYG	24	Экспорт осветлённого текста: 0=Off, 1=On
PREF_OUTPUT_DISPLAYPOINTSFAST	25	Использование быстрого рендеринга символов: 0=Off, 1=On

Числовые определители настроек печати	ID	GetPreference возвращает
PREF_PRINTER_USESPECIFICPRINTER	0	0=стандартный принтер Windows, 1 = принтер MapInfo
PREF_PRINTER_PAPERSIZE	1	Возвращаемые значения смотрите в PaperSize.Def, например: 1=формат Letter, 2=формат Legal
PREF_PRINTER_ORIENTATION	2	Смотрите WIN_INFO_PRINTER_ORIENT
PREF_PRINTER_PRINTERNAME	3	Строка, величина типа String: Имя принтера
PREF_PRINTER_SAVEPRINTERINTOWORKSPACE	4	Сохранение информации о принтере в рабочем наборе WOR: 0=Off, 1=On
PREF_PRINTER_RESTOREPRINTERFROMWORKSPACE	5	Восстановление информации о принтере из рабочего набора WOR: 0=Off, 1=On

Числовые определители настроек веб-служб	ID	GetPreference возвращает
PREF_WEBSERVICES_WFSCONNECTTIMEOUT	0	Короткое целое число – SmallInt: задержка соединения в секундах
PREF_WEBSERVICES_WFSSENDTIMEOUT	1	Короткое целое число – SmallInt: задержка передачи в секундах
PREF_WEBSERVICES_WFSRECEIVETIMEOUT	2	Короткое целое число – SmallInt: задержка получения в секундах

Числовые определители настроек веб-служб	ID	GetPreference возвращает
PREF_WEBSERVICES_WMSCONNECTTIMEOUT	3	Короткое целое число – SmallInt: задержка соединения в секундах
PREF_WEBSERVICES_WMSENDDTIMEOUT	4	Короткое целое число – SmallInt: задержка передачи в секундах
PREF_WEBSERVICES_WMSRECEIVETIMEOUT	5	Короткое целое число – SmallInt: задержка получения в секундах
PREF_WEBSERVICES_WMSMAXWIDTH	6	Короткое целое число – SmallInt: максимальная ширина WMS в пикселях
PREF_WEBSERVICES_WMSMAXHEIGHT	7	Короткое целое число – SmallInt: максимальная высота WMS в пикселях
PREF_WEBSERVICES_ACCESSTYPE	8	Тип прокси-доступа: 0=системный, 1=напрямую, 2=используя прокси
PREF_WEBSERVICES_PROXYSERVER	9	Строка, величина типа String: IP прокси-сервера
PREF_WEBSERVICES_PORT	10	Короткое целое число – SmallInt: Порт прокси-сервера
PREF_WEBSERVICES_BYPASSLOCAL	11	Использование обходного пути: 0=Off, 1=On
PREF_WEBSERVICES_BYPASSLIST	12	Строка, величина типа String: список адресов, использующих обходной путь
PREF_WEBSERVICES_WFSREFRESHUSINGMAP	13	Обновление картографических фильтров на текущей карте: 0=Off, 1=On
PREF_WEBSERVICES_WEBGEOCODECONNECTTIMEOUT	14	Короткое целое число – SmallInt: задержка соединения в секундах
PREF_WEBSERVICES_WEBGEOCODESENDDTIMEOUT	15	Короткое целое число – SmallInt: задержка передачи в секундах
PREF_WEBSERVICES_WEBGEOCODERECEIVETIMEOUT	16	Короткое целое число – SmallInt: задержка получения в секундах

Числовые определители настроек веб-служб	ID	GetPreference возвращает
PREF_WEBSERVICES_STREETOFFSET	17	Смещение от дороги
PREF_WEBSERVICES_CORNEROFFSET	18	Смещение от перекрёстка
PREF_WEBSERVICES_UNITOFFSET	19	Возвращает PREF_SYSTEM_DISTANCEUNITS
PREF_WEBSERVICES_GEOCODEDICTIONARY	20	Короткое целое число – SmallInt: Смотрите возвращаемые значения GeocodeInfo()
PREF_WEBSERVICES_GEOCODEBATCHSIZE	21	Короткое целое число – SmallInt: Точки для пакетного геокодирования
PREF_WEBSERVICES_WEBROUTINGCONNECTTIMEOUT	22	Короткое целое число – SmallInt: Задержка соединения в секундах
PREF_WEBSERVICES_WEBROUTINGSENDTIMEOUT	23	Короткое целое число – SmallInt: Задержка передачи в секундах
PREF_WEBSERVICES_WEBROUTINGRECEIVETIMEOUT	24	Короткое целое число – SmallInt: Задержка получения в секундах
PREF_WEBSERVICES_DRIVEREGIONBATCHSIZE	25	Короткое целое число – SmallInt: Точки для пакетного определения зон транспортной доступности
PREF_WEBSERVICES_APPLYMULTIRESULTCODE	26	Отметка нескольких совпадений в автоматическом режиме
PREF_WEBSERVICES_READTIMEOUTWEBTILESERVER	27	Считывание задержек в секундах для тайл-серверов
PREF_WEBSERVICES_REQUESTTIMEOUTWEBTILESERVER	28	Запрос задержек в секундах для тайл-серверов
PREF_WEBSERVICES_LIBRARYSERVICEACCESSMODE	29	0=отключено, 1=доступ к каталогу, 2=доступ к каталогу и библиотеке
PREF_WEBSERVICES_OPENMANAGEDTABLEEDITOR	30	Открытие записей для просмотра/изменения после сохранения управляемых таблиц

Числовые определители настроек веб-служб	ID	GetPreference возвращает
PREF_WEBSERVICES_BINGKEYFROMUSER	31	Строка, величина типа String: Возвращение произвольного ключа лицензии Bing, если требуется.

Числовые определители настроек обработки раstra	ID	GetPreference возвращает
PREF_IMAGEPROCESSING_REPROJECTIONUSEINDEX	0	0=никогда, 1=всегда, 2=оптимально
PREF_IMAGEPROCESSING_RESAMPLEMETHODINDEX	1	0=кубическая свёртка, 1=ближайшее соседство

Числовые определители настроек отчётов	ID	GetPreference возвращает
PREF_LAYOUT_PROMPTSAVEWORKSPACE	3	0=OFF, 1=ON

Числовые определители настроек PDF	ID	GetPreference возвращает
PREF_PDF_LAYEREDPDF	0	Многослойный PDF: 0=OFF, 1=ON
PREF_PDF_SAVEGEOINFO	1	Создание геопривязанного PDF: 0=OFF, 1=ON
PREF_PDF_SAVEATTRIBUTEDATA	2	Сохранение атрибутивных данных: 0=OFF, 1=ON
PREF_PDF_AUTOOPENPDF	3	Автоматическое открытие PDF: 0=OFF, 1=ON
PREF_PDF_POSTSCRIPTSIM	4	Использование расширенного режима подготовки PDF: Включение эмуляции Postscript
PREF_PDF_SAVEGEOINFOINNATIVEUNITS	5	Единицы измерения координат геопривязки: 0=единицы карты, 1=градусы

Числовые определители настроек производительности	ID	GetPreference возвращает
PREF_PERFORMANCE_CONCURRENCYSETTING	0	0=нет,1=умеренно,2=средне,3=агрессивно,4=полностью
PREF_PERFORMANCE_DROPINDEXSETTING	1	SmartIndexing: 0=автоматически,1=On,2=Off

Если требуется использовать коды, включите в начало программы оператор Include "MAPBASIC.DEF".

Условия появления сообщения об ошибке

ERR_FCN_ARG_RANGE (644) — сообщение об ошибке создаётся, если неправильно задано значение аргумента.

Пример

```
Include "MAPBASIC.DEF"
Dim i As SmallInt
Dim s As String
i = GetPreference(PREFERENCE_SYSTEM, PREF_SYSTEM_DEFAULTFILEFORMAT)
s = GetPreference(PREFERENCE_SYSTEM, PREF_SYSTEM_DEFAULTCHARSETFORNATIVE)
```

См. также:

функция [GetPreferencePath\\$\(\)](#)

Функция GetPreferencePath\$()

Назначение

Возвращает дисковый маршрут к специальным таблицам и файлам MapInfo Pro, заданным в диалоге настроек. Этот маршрут используется для инициализации диалога настроек MapInfo Pro при первом использовании в сеансе и может измениться в диалоге **Настройки**.

Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetPreferencePath$ ( preference_path_id )
```

preference_path_id принимает следующие значения:

- PREFERENCE_PATH_TABLE (0)
- PREFERENCE_PATH_WORKSPACE (1)
- PREFERENCE_PATH_MBX (2)
- PREFERENCE_PATH_IMPORT (3)
- PREFERENCE_PATH_SQLQUERY (4)
- PREFERENCE_PATH_THEMETHEMPLATE (5)
- PREFERENCE_PATH_MIQUERY (6)
- PREFERENCE_PATH_NEWGRID (7)
- PREFERENCE_PATH_CRYSTAL (8)
- PREFERENCE_PATH_GRAPH SUPPORT (9)
- PREFERENCE_PATH_SHAPEFILE (10)
- PREFERENCE_PATH_REMOTETABLE (11)
- PREFERENCE_PATH_WFSTABLE (12)
- PREFERENCE_PATH_WMSTABLE (13)

Возвращаемая величина

Строка

Описание

Функция `GetPreferencePath$( )` возвращает строку с дисковым маршрутом одного из служебных каталогов MapInfo Pro. Ниже приводится пример поиска маршрута к каталогу, в который MapInfo помещает новые таблицы.

Пример

```
include "mapbasic.def"
declare sub main
sub main
dim sMiPrfFile as string
sMiPrfFile = GetPreferencePath$( PREFERENCE_PATH_WORKSPACE)
Print sMiPrfFile
end sub
```

См. также:

[функция `GetCurrentPath\( \)`](#), [функция `LocateFile\( \)`](#)

Функция GetSeamlessSheet()

Назначение

Выдает диалоговый запрос, в котором пользователь должен выбрать одну из таблиц-компонент сшитой таблицы и возвращает имя выбранной таблицы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetSeamlessSheet( table_name )
```

table_name – имя открытой сшитой таблицы.

Возвращаемая величина

Строка с заданным именем таблицы (пустая строка, если введенные в диалоге данные были отменены).

Описание

Эта функция показывает диалоговое окно со списком таблиц, составляющих сшитую таблицу. Если пользователь выберет таблицу и нажмет на кнопку **ОК**, то функция вернет имя выбранной таблицы. Если пользователь отменит диалог, то результатом будет пустая строка.

Пример

```
Sub Browse_A_Table(ByVal s_tab_name As String)
  Dim s_sheet As String

  If TableInfo(s_tab_name, TAB_INFO_SEAMLESS) Then
    s_sheet = GetSeamlessSheet(s_tab_name)
    If s_sheet <> "" Then
      Browse * From s_sheet
    End If
  Else
    Browse * from s_tab_name
  End If

End Sub
```

См. также:

оператор Set Table, оператор TableInfo()

Функция GetTime()

Назначение

Возвращает компоненту времени из переменной типа **DateTime**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GetTime ( DateTime )
```

Возвращаемая величина

Время

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim dtX as datetime
dim Z as time
dtX = "03/07/2007 12:09:09.000 AM"
Z = GetTime(dtX)
Print FormatTime$(Z, "hh:mm:ss.fff tt")
```

См. также:

[функции даты и времени](#), [функция FormatDate\\$\(\)](#), [функция FormatTime\\$\(\)](#), [функция NumberToDateTime\(\)](#), [функция GetDate\(\)](#)

Оператор Global

Назначение

Объявляет имена и типы глобальных переменных.

Синтаксис

```
Global var_name [ , var_name... ] As var_type
[ , var_name... ] As var_type... ]
```

var_name – имя глобальной переменной;

var_type – тип данных: Integer, Float, Date, Logical, String или тип, созданный при помощи оператора Type.

Описание

Оператор **Global** объявляет одну или более глобальных переменных. **Global** употребляется вне текста процедур и функций.

Оператор **Global** и **оператор Dim** используют одинаковый синтаксис; отличие в том, что переменные, объявленные оператором **Global**, глобальны, а **оператор Dim** объявляет локальные переменные. Локальные переменные могут использоваться только в процедурах, в которых они объявлены. Значения глобальных переменных могут использоваться и изменяться во всех процедурах программы. В этом случае локальные переменные, заданные в подпрограмме, используются в первую очередь (например, в подпрограмме обращение к переменной подразумевает использование локальной переменной, а не глобальной с тем же именем). Для изменения размерности массива глобальных переменных используется **оператор ReDim**. В Windows значения глобальных переменных выполняющей программы доступны другим программам, поддерживающим DDE-связь.

Пример

```
Declare Sub testing()
Declare Sub Main()
Global gi_var As Integer
Sub Main()
  Call testing
  Note Str$(gi_var) ' this displays "23"
End Sub

Sub testing()
  gi_var = 23
End Sub
```

См. также:

оператор Dim, оператор ReDim, оператор Type, функция UBound()

Оператор Goto

Назначение

Передаёт управление программой оператору с меткой.

Предупреждение!

Оператор **Goto** не может выполняться в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Goto label
```

label – имя метки в процедуре.

Описание

Оператор **Goto** определяет безусловный переход выполнения программы. Выполнение продолжается с оператора, отмеченного меткой *label*. Метка *label* в тексте программы представляет собой произвольное слово перед оператором, отделённое двоеточием; слово *label* используется в операторе **Goto** без двоеточия.

Оператор **Goto** не рекомендуется использовать для выхода из цикла. Для этого используются **оператор Exit Do** и **оператор Exit For**. Также **Goto** нельзя использовать для входа в тело цикла.

Оператор **Goto** можно использовать для перехода на метку *label* только внутри одной процедуры.

Пример

```
Goto endproc  
  
...  
  
endproc: End Program
```

См. также:

оператор Do Case...End Case, **оператор Do...Loop**, **оператор For...Next**, **оператор OnError**, **оператор Resume**

Функция GraduatedThemeInfo()

Назначение

Возвращает информацию об определённом на карте тематическом слое градуированных значков. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
GraduatedThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id *map_window_id* – идентификатор окна карты;

thematic_layer_id Короткое целое (*SmallInt*), которое идентифицирует тематический слой градуированных символов (значков) карты.

attribute Код – короткое целое (*SmallInt*), см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	GraduatedThemeInfo() возвращает
GRADUATEDTHEME_INFO_SYMBOL	1	Предложение Symbol , описывающее стиль оформления градуированных условных знаков.
GRADUATEDTHEME_INFO_VALUE	2	Вещественный (<i>Float</i>) результат, представляющий численное значение, которое соответствует символу, использованному выше (GRADUATEDTHEME_INFO_SYMBOL) Метод GRADUATEDTHEME_INFO_GRAD_ROOT(0) "SQRT" (квадратного корня), который присваивает возрастающие размеры условных знаков точек по мере увеличения квадратных корней значений GRADUATEDTHEME_INFO_GRAD_CONST(1) – постоянный размер символов GRADUATEDTHEME_INFO_GRAD_LOG(2) – изменяет размер по логарифмам значений

код атрибута	ID	GraduatedThemeInfo() возвращает
GRADUATEDTHEME_INFO_SHOW_NEG	3	Логический (<i>Logical</i>) результат (Т или F), которым определяется, следует ли отображать отрицательные значения или нет.
GRADUATEDTHEME_INFO_SHOW_NEG_SY	4	Предложение Symbol , описывающее стиль оформления градуированных условных знаков отрицательных значений.
GRADUATEDTHEME_INFO_GRADUATION	5	Короткий целочисленный (<i>SmallInt</i>) результат, представляющий, как разность численных значений соответствуют разнице размеров символов условных знаков. Поддерживается три метода.

Функция GridTableInfo()

Назначение

Возвращает информацию об открытой таблице поверхности.

Синтаксис

```
GridTableInfo( table_id, attribute )
```

table_id – либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0. таблица должна описывать сетку (grid).

attribute – целочисленный код, уточняющий тему запроса.

Возвращаемая величина

Величина типа String, Integer, SmallInt или Logical, в зависимости от значения параметра attribute.

Параметр attribute определяет вид возвращаемой функцией информации. Коды в левом столбце (например, GRID_TAB_INFO_) определены в файле стандартных определений MAPBASIC.DEF.

код атрибута	ID	GridTableInfo() возвращает:
GRID_TAB_INFO_MIN_VALUE	1	Вещественное (Float) число, представляющее минимальное значение в ячейках сетки.
GRID_TAB_INFO_MAX_VALUE	2	Вещественное (Float) число, представляющее максимальное значение в ячейках сетки.
GRID_TAB_INFO_HAS_HILLSHADE	3	Логическое (Logical) значение, TRUE, если грид-файл содержит информацию о затенении и рельефности. Это значение не зависит от того, применяется ли затенение (hillshading) при показе.

См. также:

оператор [Create Grid](#), функция [GetGridCellValue\(\)](#), функция [IsGridCellNull\(\)](#), функция [RasterTableInfo\(\)](#)

Функция [GroupLayerInfo\(\)](#)

Назначение

Эта функция возвращает информацию о заданной группе слоёв карты.

Синтаксис

```
GroupLayerInfo ( map_window_id, group_layer_id, attribute )
```

map_window_id – идентификатор окна Карты;

group_layer_id – номер группы слоев (например, 1 для самой верхней группы) или название группы. Количество групп слоев на карте возвращает [функция MapperInfo\(\)](#).

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

В качестве атрибутов можно использовать:

attribute-код	ID	Описание
GROUPLAYER_INFO_NAME	1	Возвращает строку с именем слоя.
GROUPLAYER_INFO_LAYERLIST_ID	2	Возвращает число, номер группы слоев в списке слоев.
GROUPLAYER_INFO_DISPLAY	3	Возвращает логическое значение: - GROUPLAYER_INFO_DISPLAY_ON ("правда", если слой видимый) - GROUPLAYER_INFO_DISPLAY_OFF ("ложь", если слой невидимый)
GROUPLAYER_INFO_LAYERS	4	Возвращает количество графических слоев в группе. Группировка слоев игнорируется и пересчитываются все все слои, в том числе и вложенные.
GROUPLAYER_INFO_ALL_LAYERS	5	Возвращает количество графических слоёв и групп слоёв, включая все вложенные слои и группы слоёв.
GROUPLAYER_INFO_TOPLEVEL_LAYERS	6	Возвращает количество первых графических слоёв в группах.
GROUPLAYER_INFO_PARENT_GROUP_ID	7	Возвращает идентификатор группы слоёв содержащей группу слоёв. Если группа слоёв самая верхняя, возвращается ноль.

ID-номера групп слоев меняются от 0 до n, где n - количество групп слоев, а 0 представляет собой верхний ("корневой") слой в списке. Все атрибуты информации о группах слоёв кроме GROUPLAYER_INFO_DISPLAY доступны и для "корневого" уровня слоёв.

GROUPLAYER_INFO_NAME возвращает название карты (совпадает с заголовком окна).

Косметический слой добавляется в сумму при любом варианте подсчета графических слоев.

Назначение ID-номеру значения 0 возвращает имя окна карты, а также имя Косметического слоя в виде "cosmetic1", "cosmetic2".

См. также:

функция [LayerInfo\(\)](#), функция [MapperInfo\(\)](#)

Функция HomeDirectory()

Назначение

Возвращает личный каталог пользователя. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
HomeDirectory$( )
```

Возвращаемая величина

Строка

Описание

Функция **HomeDirectory\$()** возвращает строку, представляющую личный каталог пользователя.

Значение, которое будет возвращено функцией, определяется системной платформой.

Следующая таблица обобщает правила назначения личных каталогов в разных платформах:

Система	Определение "личного каталога"
Окна	Каталог, заданный по правилам Windows и

Пример

```
Dim s_home_dir As String
s_home_dir = HomeDirectory$( )
```

См. также:

функция [ApplicationDirectory\(\)](#), функция [ProgramDirectory\(\)](#), функция [SystemInfo\(\)](#)

Функция HotlinkInfo()

Назначение

Возвращает информацию о геолинке на карте. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
HotlinkInfo ( map_window_id, layer_number, hotlink_number, attribute )
```

map_window_id – идентификатор окна Карты;

layer_number – номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

hotlink_number – индекс запрашиваемого геолинка. Первое определение геолинка на карте имеет номер индекса 1.

attribute - допускаются следующие значения атрибута:

Имя геолинка	ID	Описание
HOTLINK_INFO_EXPR	1	Возвращает выражение для файла, на который указывает геолинк.
HOTLINK_INFO_MODE	2	Возвращает текущий режим геолинка, одно из следующих предопределенных значений: • HOTLINK_MODE_LABEL (0) • HOTLINK_MODE_OBJ (1) • HOTLINK_MODE_BOTH (2)
HOTLINK_INFO_RELATIVE	3	Возвращает "Да" (TRUE), если маршрут в геолинке задан в относительной форме.
HOTLINK_INFO_ENABLED	4	Возвращает "Да" (TRUE), если геолинк активирован.
HOTLINK_INFO_ALIAS	5	Возвращает TRUE, если геолинк является псевдонимом.

См. также:

оператор **Set Map**, функция **LayerInfo()**

Функция Hour()

Назначение

Возвращает часовую компоненту времени (переменной типа Time) в виде целого числа в диапазоне 0-23. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Hour ( Time )
```

Возвращаемая величина

Короткое целое – SmallInt

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim Z as time  
dim iHour as integer  
Z = CurDateTime()  
iHour = Hour(Z)  
Print iHour
```

См. также:

функция **Minute()**, функция **Second()**

оператор If...Then

Назначение

Условное выполнение той или иной группы операторов.

Синтаксис

```

If if_condition Then
  if_statement_list
  [ ElseIf elseif_condition Then
    elseif_statement_list ]
  [ ElseIf ... ]
  [ Else
    else_statement_list ]
End If

```

condition – выражение, результат которого есть логическая величина (TRUE или FALSE).

statement_list – группа операторов, количество которых может быть нулевым.

Предупреждение!

Оператор **If...Then** нельзя выполнить в окне **MapBasic**.

Описание

Оператор **If...Then** позволяет выбрать, какую группу операторов выполнить при удовлетворении определенных условий.

Возможна простая форма оператора **If** без предложений **Elseif** и **Else**:

```

If if_condition Then
  if_statement_list
End If

```

В этом случае значение выражения *if_condition* вычисляется при выполнении программы. Если выражение *if_condition* равно логическому значению TRUE, то MapBasic выполнит операторы *if_statement_list*; иначе MapBasic пропустит группу операторов *if_statement_list*.

Второй вариант формы **If** включает конструкцию **Else**:

```

If if_condition Then
  if_statement_list
Else
  else_statement_list
End If

```

Здесь MapBasic вычисляет либо группу операторов *if_statement_list* (если условие condition TRUE, т.е. истинно), либо группу *else_statement_list* (если condition равно FALSE).

Третий вариант формы **If** включает предложение **Elseif**, следующее за **If** (а затем можно добавить предложение **Else**):

```
If if_condition Then
    if_statement_list
ElseIf elseif_condition Then
    elseif_statement_list
Else
    else_statement_list
End If
```

В этой форме оператора MapBasic проверяются два или более условий до тех пор, пока либо какое-нибудь условие не равно TRUE, либо пока не достигнут конец предложения **Else**, либо пока не встретится **End If**. Если условие *if_condition* равно TRUE, MapBasic выполнит группу *if_statement_list* и передаст управление первому оператору, следующему за **End If**. Если условие равно FALSE, MapBasic переходит к вычислению условия *elseif_condition* и, если оно оказывается истинным (TRUE), выполняет группу операторов *elseif_statement_list*.

Оператор **If** может содержать несколько конструкций **Elseif** позволяющих рассматривать любое количество возможных состояний. Но если требуется, чтобы программа была написана в хорошем стиле, используйте **оператор Do Case...End Case** вместо оператора **If** с несколькими предложениями **Elseif**.

Пример

```
Dim today As Date
Dim today_mon, today_day, yearcount As Integer

today = CurDate() ' get current date
today_mon = Month(today) ' get the month value
today_day = Day(today) ' get the day value (1-31)

If today_mon = 1 And today_day = 1 Then
    Note "Happy New Year!"
    yearcount = yearcount + 1
ElseIf today_mon = 2 And today_day = 14 Then
    Note "Happy Valentine's Day!"
ElseIf today_mon = 12 And today_day = 25 Then
    Note "Merry Christmas!"
Else
    Note "Good day."
End If
```

См. также:

оператор Do Case...End Case

Функция IIf()

Назначение

Такая функция выполняет выражения. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IIf( expression, truepart, falsepart )
```

expression — выражение, которое требуется выполнить.

truepart — значение или оценённое выражение, возвращённые, когда *expression* находится в состоянии True.

falsepart — значение или исполненное выражение, которое возвращается, когда условие в выражении *expression* находится в состоянии False.

Описание

Все параметры выражения *expression*, *truepart* и *falsepart* являются обязательными.

Mapbasic оценивает выражение прежде чем передать функции IIF, поэтому оба параметра *truepart* и *falsepart* проходят оценку, даже если возвращается только один из них.

Замечание о IIF() и IfThen

Функция IfThen оценивает первый аргумент и, в зависимости от результата, переходит к обработке второго или третьего аргумента. MapBasic предварительно оценивает выражения до передачи их функции IIF для оценки всех трёх аргументов и получает сообщения об ошибках исполнения (например, когда встречается деление на ноль). Это показано в следующем примере:

Когда делитель не равняется нулю (0), функция IIf() возвращает четыре (4).

```
Dim divisor, number as Integer
divisor = 3
number = 12
Print IIf(divisor <> 0, number \ divisor, 0)
```

Если делитель равен нулю (0), IIf() вызывает ошибку исполнения, разделив число на ноль (0), поскольку MapBasic предварительно оценивает все три аргумента до передачи их функции.

```
divisor = 0
Print IIf(divisor <> 0, number \ divisor, 0)
```


Пример

Простая подпись карты Мира, где континент либо Азия, либо нет.

```
Set Map Window <mapID> Layer 1 Label With IIf(Continent="Asia", "Asia",
"Other Continent")
```

Обновляя колонку Division таблицы States по категории, зависящей от того является ли значение в колонке Population 1990 большим 6 миллионов или нет.

```
Update States Set Division = IIF(Pop_1990>6000000, "LargePop", "SmallPop")
DropIndex Auto
```

Выбирая из колонки States все значения, в которых в колонке плотности населения (Pop_Density) меньше 10 или нет.

```
Select * from States where IIF(Pop_Density<10, Pop_Density, 0) into
Selection
```

Создавая тематическую карту индивидуальных, отдельных значений, где, если название штата равняется Алабама State_Name = Alabama, то такой штат будет закрашен зелёным, а иначе, во всех остальных случаях штаты закрашиваются красным.

```
shade window <mapID> 1 with Int(IIF(State_Name="Alabama", 1, 0)) values
0 Brush (2,16711680,16777215) Pen (1,2,0) ,1 Brush (2,65280,16777215)
Pen (1,2,0) default Brush (1,0,16777215) Pen (1,2,0)
```

См. также:

[Функция COND\(\)](#)

Оператор Import

Назначение

Создает новую таблицу MapInfo Pro при импорте файла, такого как GML или DXF. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

См. разделы: [Импорт файлов MIF/MID, PICT, или MapInfo для DOS](#), [Импорт DXF-файлов](#), [Импорт GML-файлов](#) и [Импорт из файлов GML 2.1](#).

Импорт из MIF/MID, PICT или MapInfo для DOS

Синтаксис

```
Import file_name
[ Type file_type ]
[ Into table_name [ Type { NATIVE | NATIVEX } ] ]
[ Overwrite ]
```

file_name – строка, определяющая имя импортируемого файла;

file_type – строка, определяющая формат импортируемого файла (MIF, MBI, MMI, IMG или PICT);

table_name – определяет имя новой создаваемой таблицы;

Описание

Оператор **Import** создает новую таблицу MapInfo, импортируя содержимое существующего файла.

Примечание: Для создания таблицы MapInfo, оснований на электронной таблице или файле базы данных, используйте оператор **Register Table**, а не оператор **Import**.

Дополнительное предложение **Type** определяет формат файла при импорте. Предложение **Type** может иметь один из следующих вариантов:

Предложение Type	Формат будущего файла
Тип "DXF"	Формат DXF (графический формат программ CAD, таких как AutoCAD). См. Импорт из DXF .
Тип "MIF"	Файлы MIF / MID, создаваемые при экспорте таблиц MapInfo.
Тип "MBI"	Файл MapInfo Boundary Interchange, созданный в MapInfo для DOS.
Тип "MMI"	Файл MapInfo Map Interchange, созданный в MapInfo для DOS.
Тип "IMG"	Файл MapInfo Image, созданный в MapInfo для DOS.
Тип "GML"	GML-файлы. См. Импорт из GML-файлов .
Тип "GML21"	Файлы формата GML 2.1. Смотрите: Импорт из файлов GML 2.1 .

Если пропущено предложение **Type**, предполагается, что формат файла определяется расширением. Например, файл с именем "PARCELS.DXF" будет считаться файлом DXF. (Об импорте из DXF см. в разделе: [Импорт из DXF](#)).

Предложение **Into** позволяет переписать имя и положение создаваемой таблицы MapInfo. Если предложение Into не определено, то новая таблица создается в том же каталоге, что и исходный файл, с соответствующим именем. Например, если в среде Windows импортируется текстовый файл "WORLD.MIF", то новая таблица по умолчанию будет иметь имя "WORLD.TAB".

Используйте предложение **Into** с предложением **Type**, чтобы задать режим импорта либо в традиционном формате MapInfo (**NATIVE**), либо в расширенном формате MapInfo Extended (**NATIVEX**). Формат **NATIVEX** поддерживает кэш-таблицы, размер которых превышает 2GB, и наборы символов UTF-8 и UTF-16. Дополнительное предложение **Type** должно следовать за предложением **Into**, если присутствует это предложение. Если не задано, то подразумевается использование традиционного формата **NATIVE**. Обратите внимание, что предложение **Type** отличается от предложения **Type**, используемого при импорте типа файла.

Если используется дополнительное ключевое слово **Overwrite**, MapInfo Pro создаст новую таблицу, независимо от того, существует или нет таблица с таким именем; новая таблица заменит существующую. Если ключевое слово **Overwrite** опущено, а таблица с таким именем уже существует, то MapInfo Pro не будет переписывать таблицу.

Пример

Пример импорта с применением текущего стиля MapInfo:

```
Import "D:\midata\GML\test.gml" Type "GML" layer "TopographicLine" style
auto off Into "D:\midata\GML\test_TopographicLine.TAB" Overwrite
```

Пример импорта в файл MIF (MapInfo Interchange Format):

```
Import "WORLD.MIF"
Type "MIF"
Into "world_2.tab"

Map From world_2
```

Импорт DXF-файлов

Синтаксис

```
Import file_name
[ Type "DXF" ]
[ Into table_name ]
[ Overwrite ]
[ Warnings { On | Off } ]
```

```
[ Preserve
  [ AttributeData ] [ Preserve ] [ Blocks As MultiPolygonRgns ] ]
[ CoordSys... ]
[ Autoflip ]
[ Transform
  ( DXF_x1, DXF_y1 ) ( DXF_x2, DXF_y2 )
  ( MI_x1, MI_y1 ) ( MI_x2, MI_y2 ) ]
[ Read
  [ Integer As Decimal ] [ Read ] [ Float As Decimal ] ]
[ Store [ Handles ] [ Elevation ] [ VisibleOnly ] ]
[ Layer DXF_layer_name
  [ Into table_name ]
  [ Preserve
    [ AttributeData ] [ Preserve ] [ Blocks As MultiPolygonRgns ] ] ]
[ Layer... ]
```

file_name – строка, определяющая имя импортируемого файла;

table_name – определяет имя новой создаваемой таблицы;

DXF_x1, DXF_y1 и т.д. – числа, представляющие координаты в DXF файле;

MI_x1, MI_y1 и т.д. – числа, представляющие координаты в таблице MapInfo;

DXF_layer_name – строка, имя слоя в DXF файле;

Описание

Если импортируется DXF-файл, то оператор **Import** может включать следующие предложения, определяющие настройки DXF.

Примечание: Порядок предложений крайне важен; помещение предложений в неправильном порядке приведет к ошибкам компиляции.

Warnings On или **Warnings Off** – контролируют, появляются ли предупреждающие сообщения во время импорта. По умолчанию установлено Warnings Off.

Preserve AttributeData – включите это предложение, если надо, чтобы MapInfo Pro сохраняла атрибутивные данные из DXF.

Preserve Blocks As MultiPolygonRgns – Включите это предложение, если надо, чтобы MapInfo Pro преобразовала все полигоны из блока записи DXF в единый объект из множества полигонов. Если это предложение пропущено, каждый полигон DXF становится отдельным объектом-регионом MapInfo Pro.

CoordSys – контролирует проекцию и систему координат таблицы. Дополнительная информация содержится в разделе [Предложение CoordSys](#).

Autoflip – включите этот режим, если надо повернуть X-координаты карты относительно центральной линии карты. Команда применима только к плановым (не-мировым) координатам.

Transform используется для задания координатного искажения. В предложении **Transform** задается минимум и максимум X- и Y-координат импортируемого файла, и задается минимум и максимум тех координат, которые Вам понадобятся в таблице MapInfo.

Read Integer As Decimal – используйте это предложение, если надо хранить целочисленные значения из файла DXF в десятичной колонке новой таблицы. Это предложение можно использовать только при применении предложения **Preserve AttributeData**.

Read Float As Decimal – используйте это предложение, если надо хранить вещественные значения из файла DXF в десятичной колонке новой таблицы. Это предложение можно использовать только при применении предложения **Preserve AttributeData**.

Store [Handles] [Elevation] [VisibleOnly] – если определено предложение **Handles**, то таблица MapInfo хранит уникальные ID-номера графических объектов в колонке, с именем “_DXFHandle”. Если определено предложение **Elevation**, MapInfo Pro хранит высоту центра каждого объекта в колонке с именем “_DXFElevation”. (Для линий, MapInfo Pro хранит высоты центров линий; для регионов, MapInfo Pro хранит среднее значение высоты объекта) Если включено предложение **VisibleOnly**, MapInfo Pro игнорирует невидимые объекты.

Предложение Layer – если не задано ни одного предложения **Layer**, все объекты из файла DXF импортируются в одну таблицу MapInfo. Если используется одно или более предложений **Layer**, каждый слой DXF, который перечислен, будет преобразован в отдельную таблицу MapInfo.

Если файл DXF содержит много слоев, и оператор **Import** содержит одно или несколько предложений **Layer**, то MapInfo Pro будет импортировать слои в соответствии с именами таблиц. Например, пусть файл DXF содержит четыре слоя (под номерами 0, 1, 2 и 3).

Следующий оператор **Import** будет импортировать все четыре слоя в одну таблицу MapInfo:

```
Import "FLOORS.DXF"
Into "FLOORS.TAB"
Preserve AttributeData
```

Следующий оператор будет импортировать слои 1 и 3, но не будет импортировать слои 0 или 2:

```
Import "FLOORS.DXF"
Layer "1"
Into "FLOOR_1.TAB"
Preserve AttributeData
Layer "3"
Into "FLOOR_3.TAB"
Preserve AttributeData
```

Импорт GML-файлов

Синтаксис

```
Import file_name
[ Type "GML" ]
[ Layer layer_name ]
[ Into table_name ]
[ Style Auto [ On | Off ] ]
```

file_name – строка, определяющая имя импортируемого файла;

layer_name – строка, имя слоя в GML файле.

table_name – определяет имя новой создаваемой таблицы;

Описание

Type принимает значение “GML” для GML-файлов.

MapInfo Pro поддерживает импорт GML-файлов от OSGB (Ordnance Survey of Great Britain). Поддерживаются объекты типа Cartographic Symbol, Topographic Point, Topographic Line, Topographic Area и Boundary Line; тип Cartographic Text не поддерживается. Topographic Area может иметься в двух формах; MapInfo Pro поддерживает нетопологическую форму. Если файл содержит XLINKS, MapInfo Pro импортирует только атрибутивные данные и не импортирует пространственные объекты. Связи типа XLINK хранятся в файле GML как "xlink:href=". Если топологические объекты включены в файл, то появится предупреждающее сообщение о том, что пространственные объекты не могут быть импортированы. Просмотреть атрибутивные данные можно в окне списка.

Пример:

Следующий пример импортирует GML-файл:

```
Import "D:\midata\GML\est.gml" Type "GML" layer "LandformArea" style
auto
on Into "D:\midata\GML\est_LandformArea.TAB" Overwrite
```

Импорт из файлов GML 2.1

Синтаксис

```
Import file_name
[ Type "GML21" ]
[ Layer layer_name]
```

```
[ Into table_name ]
[ Overwrite ]
[ CoordSys... ]
```

file_name – имя файла GML 2.1 для импорта.

layer_name – это имя слоя GML;

table_name – имя таблицы MapInfo;

Описание

Type – “GML21” для файлов формата GML 2.1;

Overwrite действует для автоматической перезаписи TAB-файла. Если не задано предложение **Overwrite**, а TAB-файл существует, то будет выдано сообщение об ошибке.

Предложение **Coordsys** не обязательно. Если GML-файл содержит поддерживаемую проекцию, а предложение **Coordsys** не задано, то используется проекция из GML-файла. Если GML-файл содержит поддерживаемую проекцию и задано предложение **Coordsys**, то используется проекция из предложения **Coordsys**. Если GML-файл не содержит поддерживаемую проекцию, нужно явно задать предложение **Coordsys**.

Примечание: Если проекция в предложении **Coordsys** не совпадает с проекцией, заданной в GML-файле, импорт данных может быть некорректным. Координатная система должна соответствовать координатной системе данных в файле GML. Трансформации данных из одной проекции в другую не происходит.

Пример

Следующий пример импортирует GML21-файл:

```
Import "D:\midata\GML\GML2.1\mi_usa.xml" Type "GML21" layer "USA" Into
"D:\midata\GML\GML2.1\mi_usa_USA.TAB" Overwrite CoordSys Earth Projection
1, 104
```

Ссылки, связанные с данной

Оператор Export на стр. 409

Информация по теме

Ограничения на имена переменных на стр. 383

Оператор Include

Назначение

Объединяет содержимое отдельного текстового файла с текстом программы MapBasic. Этот оператор нельзя выполнить в окне **MapBasic**.

Синтаксис

```
Include "filename"
```

filename – имя текстового файла.

Предупреждение!

Нельзя использовать оператор **Include** в окне **MapBasic**.

Описание

Когда MapBasic компилирует программный файл с оператором **Include**, текст определенного в этом операторе файла рассматривается как часть текста программы. Файл, заданный оператором **Include**, должен состоять из допустимых операторов MapBasic.

Если имя файла в параметре *filename* задано без каталога и этот файл не находится в активном каталоге, то компилятор языка MapBasic делает попытку найти его в том каталоге, в котором расположен программный файл MapBasic. Это позволяет не копировать каждый раз файлы стандартных определений (такие, как MAPBASIC.DEF) в каталог разрабатываемой программы.

Оператор **Include** используется для подсоединения к файлам стандартных определений, таких, как MAPBASIC.DEF. В них с помощью оператора Define целочисленным кодам заданы имена (например, TRUE и FALSE), использование которых в Вашей программе делает ее текст более понятным.

Однако, если вы изменили содержимое текстового файла, подсоединенного оператором **Include**, то для внесения этих изменений в программу ее необходимо перекомпилировать.

Пример:

```
Include "MAPBASIC.DEF"
```


Функция IndividualThemeInfo()

Назначение

Возвращает информацию об определённом на карте тематическом слое отдельных значений. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IndividualThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id *map_window_id* – идентификатор окна Карты;

thematic_layer_id Короткое целое (SmallInt), которое идентифицирует тематический слой индивидуальных значений на карте.

attribute Код – короткое целое (SmallInt), см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	IndividualThemeInfo() возвращает
INDIVTHEME_INFO_ITEMS_COUNT	1	Короткий целочисленный результат (SmallInt), в котором представляется счётчик количества тематических элементов (за исключением диапазона со "всеми остальными" значениями – All others).
INDIVTHEME_INFO_VARY_STYLE	2	Короткий целочисленный результат (SmallInt), в котором представляется, как применяются стили. Возможны следующие значения. INDIVTHEME_INFO_VARY_BY_ALL(0) – используются все атрибуты стилей. INDIVTHEME_INFO_VARY_BY_COLOR(1) – используются только стили, связанные с цветным оформлением. INDIVTHEME_INFO_VARY_BY_SIZE(2) – используются только стили, связанные с размерами.

код атрибута	ID	IndividualThemeInfo() возвращает
INDIVTHEME_INFO_REPLACE_STYLE	3	Логический результат (<i>Logical</i> – Т или F), определяющий заменяет ли стиль оформления тематических индивидуальных значений стили подлежащих объектов или нет.
INDIVTHEME_OTHERS_INFO_LINE	4	Предложение, описывающее стиль линии, которое используется для отображения объектов, принадлежащих элементам, входящим в список остальных объектов – All others . Если тематический слой построен не на линейной тематике, возвращается стандартный стиль линии с цветом преобладающих геометрических примитивов (областей или точек) стиля исключительного элементов (All others).
INDIVTHEME_OTHERS_INFO_PEN	5	Предложение Pen , описывающее стиль линии границы объектов, которое используется для отображения объектов, принадлежащих элементам, входящим в список остальных объектов – All others . Если тематический слой построен не на тематике областей, возвращается стандартный стиль линии Pen(1, 2, 0).
INDIVTHEME_OTHERS_INFO_BRUSH	6	Предложение Brush , описывающее стиль штриховки, которое используется для отображения объектов, принадлежащих элементам, входящим в список остальных объектов – All others . Если тематический слой построен не на тематике областей, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (точек или линий) стиля исключительного диапазона All others .
INDIVTHEME_OTHERS_INFO_SYMBOL	7	Предложение, описывающее стиль символа, которое используется для отображения объектов, принадлежащих элементам, входящим в список остальных объектов – All others . Если тематический слой построен не на точечной тематике, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (областей или линий) стиля исключительного диапазона All others .

См. также:

Функция **IndividualThemeItemInfo()** на стр. 515

Функция IndividualThemeItemInfo()

Назначение

Возвращает информацию об определённой тематической характеристике карты на тематическом слое отдельных значений. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IndividualThemeItemInfo ( map_window_id, thematic_layer_id, item_id,  
attribute )
```

map_window_id **map_window_id** – идентификатор окна карты;

thematic_layer_id Короткое целое **SmallInt**, которое идентифицирует тематический слой карты.

item_id Целое (**Integer**), которое идентифицирует тематический элемент.

attribute **SmallInt***attribute* – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	IndividualThemeItemInfo() возвращает
INDIVTHEME_ITEM_INFO_LINE	1	Предложение стиля линии, использующееся для отображения объектов, принадлежащих заданному по идентификатору элементу. Если тематический слой построен не на линейной тематике, возвращается стандартный стиль линии с цветом color преобладающих геометрических примитивов стилей элементов (областей или точек).

код атрибута	ID	IndividualThemeItemInfo() возвращает
INDIVTHEME_ITEM_INFO_PEN	2	Предложение Pen , описывающее стиль линии, которое используется для отображения объектов, принадлежащих заданному по идентификатору элементу. Если тематический слой построен не на тематике областей, возвращается стандартный стиль линии Pen(1, 2, 0).
INDIVTHEME_ITEM_INFO_BRUSH	3	Предложение Brush , описывающее стиль штриховки, используемое для отображения объектов, принадлежащих заданному по идентификатору элементу. Если тематический слой построен не на тематике областей, возвращается стандартный стиль с цветом color преобладающих геометрических примитивов (точек или линий) стиля диапазона.
INDIVTHEME_ITEM_INFO_SYMBOL	4	Предложение, описывающее стиль символа, которое используется для отображения объектов, принадлежащих заданному по идентификатору элементу. Если тематический слой построен не на точечной тематике, возвращается стандартный стиль с цветом color преобладающих геометрических примитивов (областей или линий) стиля диапазона .
INDIVTHEME_ITEM_INFO_VALUE	5	Возвращается значение (Integer, или Decimal, или String, или Float, или Time, или Date или Datetime и т.п.), представляющее заданный идентификатор элемента.

См. также:

Функция [IndividualThemeInfo\(\)](#) на стр. 513

Оператор Input

Назначение

Читает данные из файла и помещает их в переменные.

Синтаксис

```
Input # filenum, var_name [ , var_name... ]
```

filenum – номер открытого файла, который возвращает [оператор Open File](#).

var_name – имя объявляемой переменной.

Описание

Оператор **Input #** читает данные из файла, открытого в режиме последовательного доступа (INPUT или APPEND), и присваивает их как значения одной или более переменным MapBasic.

Оператор **Input #** читает данные последовательно от верхней строки к последней, присваивая каждую порцию одной переменной *var_name*. Оператор считывает порции данных из файла, используя в качестве разделителей запятую и символ конца строки. Поэтому, чтобы прочитать одну строку с запятыми в одну строковую переменную, надо использовать **оператор Line Input**.

MapBasic автоматически преобразовывает считанные данные в соответствующие типы. Когда оператор **Input #** читает пустую строку, то значением строковой переменной будет пустая строка (""). Если переменная была объявлена численным типом, то **Input #** преобразует пустую строку в ноль.

После выполнения оператора **Input #** применяется **функция EOF()**, чтобы определить, корректно ли были прочитаны данные. Если чтение произошло успешно, **функция EOF()** возвращает логическое "нет" (FALSE); если же был прочитан признак конца файла, **функция EOF()** вернет логическое "Да" (TRUE).

Пример использования оператора **Input #** можно увидеть в программе NVIEWS ("Имена Карт") из стандартной поставки MapBasic.

Следующие типы данных нельзя считывать оператором **Input #**:

- Alias
- Pen
- Brush
- Font
- Symbol
- Object

См. также:

функция EOF(), оператор Line Input, оператор Open File, оператор Write #

Оператор Insert

Назначение

Добавляет новую строку в открытую таблицу. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Insert Into table
[ ( columnlist ) ]
{ Values ( exprlist ) | Select columnlist From table }
[ DropIndex ( Auto | On | Off ) ]
```

table – имя открытой таблицы;

columnlist – разделенной запятыми список колонок;

exprlist – одно или несколько выражений для колонок, разделенных запятыми.

Описание

Оператор **Insert** вставляет новую строку в открытую таблицу. Этот оператор применяется в двух формах, позволяющих либо добавлять таблице по одной строке, либо добавлять группы строк из другой таблицы (с помощью предложения **Select**). В каждом случае порядок и количество выражений, которые будут вставлены в поля новой строки, должны соответствовать порядку колонок. Если колонки не перечислены, то будут рассматриваться все поля в новой строке. Если требуется сохранить новые строки в таблице, то после ввода используйте **оператор Commit Table**.

Если точно известно, сколько колонок в таблице, и если известно, какие именно величины в какие поля надо поместить, то можно опустить предложение *columnlist*.

В следующем примере результатом оператора будет новая строка в таблице, состоящей из четырех колонок ("Имя", "Адрес", "Город", "Область"), и каждому полю будет определено значение:

```
Insert Into customers
Values ("Mary Ryan", "23 Main St", "Dallas", "TX")
```

Результатом этого оператора может быть ошибка, если таблица будет иметь больше или меньше колонок, чем четыре. Если точно не известно, сколько колонок составляют таблицу, или не знаете, в каком порядке расположены колонки в таблице, необходимо в операторе использовать список имен колонок *columnlist*.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Когда в предложении **DropIndex** задано либо состояние **Off**, либо **Auto**, операция выполняется без приостановки и повторного создания индекса транзакций.

Примеры

В следующем примере в таблицу клиентов вставляется новая строка с одним значением поля, а остальные колонки остаются пустыми. Заполняется только поле имени с помощью

предложения **Values**, при этом можно не заботиться о том, какова структура таблицы и какое место в ней занимает колонка "Имя".

```
Insert Into customers (Name)
Values ("Steve Harris")
```

Следующий оператор создает точечный объект и вставляет его в поле "Obj" новой записи в таблице SITES. "Obj" - это имя специальной колонки для хранения графических объектов. Таким образом, мы создаем новую чистую запись с присоединенным точечным объектом.

```
Insert Into sites (Obj)
Values ( CreatePoint(-73.5, 42.8) )
```

Следующий пример иллюстрирует, как оператором **Insert** можно добавить записи из одной таблицы в другую. Допустим, что в таблице NY_ZIPS содержатся ZIP-коды штата Нью-Йорк и в таблице NJ_ZIPS содержатся ZIP-коды штата Нью-Джерси. Добавим все ZIP-коды из одной таблицы в другую. (Такая операция иногда необходима для поиска, так как оператор Find может работать только с одной таблицей.)

Аналогично оператор **Insert** добавляет все записи из таблицы штата New Jersey в таблицу штата New York.

```
Insert Into NY_ZIPS
Select * From NJ_ZIPS
```

В следующем примере берутся все объекты из таблицы WORLD и каждый подсоединяется к новой строке таблицы OUTLINE.

```
Open Table "world"
Open Table "outline"
Insert Into outline (Obj)
Select Obj From World
```

См. также:

[оператор Commit Table](#), [оператор Delete](#), [оператор Rollback](#)

Функция InStr()

Назначение

Возвращает позицию первого символа подстроки в строке. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
InStr( position, string, substring )
```

position – стартовая позиция для поиска, положительное целое число;

string – строковое выражение;

substring – строковое выражение, искомая подстрока.

Возвращаемая величина

Integer

Описание

Функция **InStr**() проверяет, входит ли в состав строки *string* подстрока *substring*. MapBasic просматривает исходную строку, начиная с символа с номером *position*; если *position* равно единице, то MapBasic начинает поиск с начала строки *string*.

Если *string* не содержит *substring* функция **InStr**() возвращает ноль.

Если строка *string* содержит подстроку *substring*, то функция **InStr**() возвращает позицию начала подстроки в строке. Например, если *substring* находится в самом начале строки *string* функция **InStr**() возвращает единицу.

Если строка *substring* – пустая строка, то результатом функции **InStr**() будет 0 (ноль).

Функция **InStr**() различает строчные и прописные буквы. Другими словами функция **InStr**() не найдет подстроку "ИЯ" в строке "Россия" и вернет ноль.

Ошибки

ERR_FCN_ARG_RANGE (644), если значение аргумента выходит за пределы, заданные при его определении.

Пример

```
Dim fullname As String, pos As Integer
fullname = "New York City"
pos = InStr(1, fullname, "York")
' pos will now contain a value of 5 (five)

pos = InStr(1, fullname, "YORK")
' pos will now contain a value of 0;
' YORK is uppercase, so InStr will not locate it
' within the string "New York City"
```

См. также:

функция Mid\$

Функция Int()

Назначение

Возвращает целое число, полученное из целой части действительного числа. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Int( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Целое – Integer

Описание

Функция **Int()** отсекает дробную часть от действительного числа, полученного в результате вычисления выражения *num_expr*, и возвращает целую часть. Функция **Fix()** и **Int()** похожи, но не идентичны.. Функции различаются способом удаления дробной части отрицательного числа. Для отрицательного дробного аргумента **функции Fix()** возвратит ближайшее целое число, большее или равное исходному, например: **Fix(-2.3)** возвратит -2. Функция же **Int()** возвращает ближайшее целое, меньше или равное оригиналу. Поэтому вызов функции **Int(-2.3)** возвращает значение -3.

Пример

```
Dim whole As Integer
whole = Int(5.999)
' whole now has the value 5

whole = Int(-7.2)
' whole now has the value -8
```

См. также:

функция Fix(), функция Round()

Функция IntersectNodes()

Назначение

Вычисляет точки пересекающихся объектов и возвращает полилинию, имеющую узлы в точках пересечения. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IntersectNodes( object1, object2, points_to_include )
```

object1 и *object2* – объектные выражения, которые могут представлять объекты любого типа, кроме точечного и текстового;

points_to_include – короткое целое значение идентификатора SmallInt (ID), смотрите таблицу далее.

Возвращаемая величина

Полилиния, составленная из точек пересечения. Величина типа Object.

Описание

Функция **IntersectNodes()** возвращает объект типа "полилиния", узлы которого лежат в точках пересечения объектов. Существует несколько атрибутов, которые **IntersectNodes()** может вернуть. Коды определены в MAPBASIC.DEF.

Код attribute	ID	Возвращаемое значение IntersectNodes()
INCL_CROSSINGS	1	INCL_CROSSINGS – функция возвращает узлы, в которых сегменты обоих объектов пересекаются;
INCL_COMMON	6	INCL_COMMON – функция возвращает узлы отрезков, на которые накладываются сегменты обоих объектов;
INCL_ALL	7	INCL_ALL – функция возвращает узлы, в которых сегменты пересекаются и накладываются.

Функция IsGridCellNull()

Назначение

Возвращает логическое (Logical) значение. Возвращает TRUE, если в соответствующей ячейке сетки (x, y) находится значение null (то есть никакого значения ячейке не присвоено).

Возвращает FALSE, если ячейка содержит значение, не равное null. Далее для извлечения значения из такой ячейки применяется функция GetCellValue().

Синтаксис

```
IsGridCellNull( table_id, x_pixel, y_pixel )
```

table_id – либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0. таблица должна описывать сетку (grid).

x_pixel – целочисленный номер X-координаты ячейки сетки. Нумерация пикселей начинается с 0. Максимальное значение координаты пикселя (*pixel_width* – 1), можно выяснить при помощи вызова:

```
RasterTableInfo(...RASTER_TAB_INFO_WIDTH)
```

y_pixel – целочисленный номер Y-координаты ячейки сетки. Нумерация пикселей начинается с 0. Максимальное значение координаты пикселя (*pixel_height* – 1), можно выяснить при помощи вызова:

```
RasterTableInfo(...RASTER_TAB_INFO_HEIGHT) .
```

Возвращаемая величина

Возвращает логическое (Logical) значение, показывающее, приписано ли данной ячейке значение. Если возвращается FALSE, то далее для извлечения значения из такой ячейки применяется функция GetCellValue().

См. также:

[оператор Create Grid](#), [функция GetGridCellValue\(\)](#), [GridTableInfo\(\)](#), [функция RasterTableInfo\(\)](#)

Функция IsogramInfo()

Назначение

Возвращает любой или все атрибуты соединения, для создания которого использовался **оператор Set Connection Isogram**. Включая атрибуты, задающие максимальное количество записей на сервере, значение времени и расстояния. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IsogramInfo( connection_handle, attribute )
```

connection_handle – целое, определяющее число соединений, которое возвращает **оператор Open Connection**.

attribute – это целое, определяющее, какого типа информация будет возвращена.

Возвращаемая величина

Вещественное (Float), логическое (Logical), или строка символов (String), в зависимости от параметра *атрибута*.

Описание

Эта функция возвращает стандартные параметры соединения или, если применялся **оператор Set Connection Isogram**.

Существует несколько атрибутов, которые **IsogramInfo()** может вернуть. Коды определены в MAPBASIC.DEF.

Параметры	ID	Возвращаемое значение IsogramInfo()
ISOGRAM_BANDING	1	Логическое, определяющее оконтуривание
ISOGRAM_MAJOR_ROADS_ONLY	2	Логическое, определяющее выбор только основных дорог MajorRoadsOnly
ISOGRAM_RETURN_HOLES	3	Логическое, определяющее следует или нет возвращать полигоны с пустотами.

Параметры	ID	Возвращаемое значение IsogramInfo()
ISOGRAM_MAJOR_POLYGON_ONLY	4	Логическое, определяющее выбор только основного полигона возвращаемой области.
ISOGRAM_MAX_OFF_ROAD_DISTANCE	5	Вещественное, определяющее значение максимального удаления от дороги Maximum off Road Distance.
ISOGRAM_MAX_OFF_ROAD_DISTANCE_UNITS	6	Строка символов, определяющая единицу измерения расстояний
ISOGRAM_SIMPLIFICATION_FACTOR	7	Вещественное, определяющее параметр упрощения (Simplification Factor), относительное значение от 0 до 1.
ISOGRAM_DEFAULT_AMBIENT_SPEED	8	Вещественное, определяющее скорость потока.
ISOGRAM_DEFAULT_AMBIENT_SPEED_DISTANCE_UNIT	9	Строка символов, определяющая единицу измерения расстояний ("mi", "km").
ISOGRAM_DEFAULT_AMBIENT_SPEED_TIME_UNIT	10	Строка символов, определяющая единицу измерения времени ("hr", "min", "sec").
ISOGRAM_DEFAULT_PROPAGATION_FACTOR	11	Определяет процентное отношение проселочных дорог в оценке остатка (расстояний) допустимое при вычислении полос дальностей. Дороги, не включенные в сеть дорог, могут быть в том числе и проездами или подъездными путями. Параметр движения – процентное отношение оценки при вычислении дальности от начальной точки до заданной на заданном расстоянии. По умолчанию этот параметр имеет значение 0.16.
ISOGRAM_BATCH_SIZE	12	Целое, определяющее максимальное количество записей, которое может одновременно передаваться службе для обработки.
ISOGRAM_POINTS_ONLY	13	Логическое, определяющее, пропускать или нет точечные объекты.
ISOGRAM_RECORDS_INSERTED	14	Целое, в котором отражено количество записей вставленное последней командой.

Параметры	ID	Возвращаемое значение IsogramInfo()
ISOGRAM_RECORDS_NOTINSERTED	15	Целое, в котором отражено количество записей не вставленное последней командой.
ISOGRAM_MAX_BATCH_SIZE	16	Целое, определяющее максимальное количество записей (например, точек), которое можно одновременно передавать службе для обработки.
ISOGRAM_MAX_BANDS	17	Целое число – максимальное число зон времени или расстояний.
ISOGRAM_MAX_DISTANCE	18	Вещественное, определяющее максимальную удаленность зоны запроса расстояний. Расстояние измеряется в единицах, указанных параметром ISOGRAM_MAX_DISTANCE_UNITS.
ISOGRAM_MAX_DISTANCE_UNITS	19	Строка задающая единицы измерения, используемые в ISOGRAM_MAX_DISTANCE.
ISOGRAM_MAX_TIME	20	Вещественное, определяющее максимальную удаленность по времени зоны запроса расстояний. Время измеряется в единицах, указанных параметром ISOGRAM_MAX_TIME_UNITS.
ISOGRAM_MAX_TIME_UNITS	21	Строка, задающая единицы времени, используемые в ISOGRAM_MAX_TIME.

Пример

Следующий фрагмент кода MapBasic печатает в окне сообщений MapInfo Pro сведения от Envinsa Routing Constraints:

```

Include "MapBasic.Def"
declare sub main
sub main
dim iConnect as integer
Open Connection Service Isogram
  URL "http://envinsa_server:8062/Route/services/Route"
  User "john"
  Password "green"
  into variable iConnect
Print "Isogram_Max_Batch_Size: " +
IsogramInfo(iConnect,Isogram_Max_Batch_Size)
Print "Isogram_Max_Bands: " + IsogramInfo(iConnect, Isogram_Max_Bands)

```

```
Print "Isogram_Max_Distance: " + IsogramInfo(iConnect,
Isogram_Max_Distance)
Print "Isogram_Max_Distance_Units: " + IsogramInfo(iConnect,
Isogram_Max_Distance_Units)
Print "Isogram_Max_Time: " + IsogramInfo(iConnect,Isogram_Max_Time)
Print "Isogram_Max_Time_Units: " +
IsogramInfo(iConnect,Isogram_Max_Time_Units)
Close Connection iConnect
end sub
```

См. также:

[оператор Create Object](#), [оператор Open Connection](#), [оператор Set Connection Isogram](#)

Функция IsPenWidthPixels()

Назначение

Определяет в каких величинах измеряется ширина линии – в пикселах или в пунктах. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
IsPenWidthPixels( penwidth )
```

penwidth – это короткое целое, определяющее ширину линии.

Возвращаемая величина

TRUE, если ширина линии задана в пикселах. FALSE, если ширина задана в пунктах.

Описание

Функция **IsPenWidthPixels()** возвратит истинное значение (TRUE), если ширина линии задана в пикселах. Ширину линии может определить, используя [функцию \(StyleAttr\(\)](#).

Пример

```
Include "MAPBASIC.DEF"
Dim CurPen As Pen
Dim Width As Integer
Dim PointSize As Float
CurPen = CurrentPen()
```

```
Width = StyleAttr(CurPen, PEN_WIDTH)
If Not IsPenWidthPixels(Width) Then
  PointSize = PenWidthToPoints(Width)
End If
```

См. также:

[функция CurrentPen\(\)](#), [функция MakePen\(\)](#), [предложение Pen](#), [функция PenWidthToPoints\(\)](#), [функция StyleAttr\(\)](#)

Оператор Kill

Назначение

Удаляет файл. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Kill filespec
```

filespec – строка, представляющая имя файла и, если необходимо, дисковый маршрут.

Возвращаемая величина

Строка

Описание

Оператор **Kill** удаляет файл с диска. Действие оператора **Kill** нельзя отменить. Поэтому применяйте **Kill** с осторожностью.

Пример

```
Kill "C:\TEMP\JUNK.TXT"
```

См. также:

[оператор Open File](#)

Функция LabelFindByID()

Назначение

Инициализирует внутренний указатель подписи; после этого можно запросить подпись из определенной строки в слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LabelFindByID( map_window_id, layer_number, row_id, table, b_mapper )
```

map_window_id – ID-номер окна Карты;

layer_number – номер слоя в текущем окне Карты (например, 1 для верхнего слоя);

row_id – положительное целочисленное значение, указывающее номер строки, в которой запрашивается подпись;

table – имя таблицы или пустой строки (""): когда Вы делаете запрос к таблице, входящей в сшитую таблицу, укажите имя такой таблицы, входящей в сшитую; в противном случае, укажите пустую строковую переменную;

b_mapper – логическая величина, "Да" (TRUE) при запросе подписей, которые появляются при активном окне карты; "Нет" (FALSE) при запросе подписей, которые появляются, когда карта помещена в отчёт.

Возвращаемая величина

Логическая величина: "Да" (TRUE) означает, что подпись для определенной строки существует.

Описание

Вызывайте **LabelFindByID()**, когда требуется запросить подпись из определенной строки в слое карты. Если величина, полученная в результате – "Да" (TRUE), то подпись существует в строке, и ее может запросить [функцию LabelInfo\(\)](#).

Пример

Следующий пример изображает карту Мира, автоматически ее подписывает и затем определяет, была ли изображена подпись для определенной строки таблицы.

```
Include "mapbasic.def"
Dim b_morelabels As Logical
Dim i_mapid As Integer
Dim obj_mytext As Object
Open Table "World" Interactive As World
Map From World
i_mapid = FrontWindow()
Set Map Window i_mapid Layer 1 Label Auto On
' Make sure all labels draw before we continue...
Update Window i_mapid
' Now see if row # 1 was auto-labeled
b_morelabels = LabelFindByID(i_mapid, 1, 1, "", TRUE)
If b_morelabels Then
' The object was labeled; now query its label.
obj_mytext = LabelInfo(i_mapid, 1, LABEL_INFO_OBJECT)
' At this point, you could save the obj_mytext object
' in a permanent table; or you could query it by
' calling ObjectInfo() or ObjectGeography().
End If
```

См. также:

функция [LabelFindFirst\(\)](#), функция [LabelFindNext\(\)](#), функция [LabelInfo\(\)](#)

Функция LabelFindFirst()

Назначение

Инициализирует внутренний указатель подписи, так что можно запросить первую подпись на слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LabelFindFirst( map_window_id, layer_number, b_mapper )
```

map_window_id – ID-номер окна Карты;

layer_number – номер слоя в текущем окне Карты (например, 1 для верхнего слоя);

b_mapper – логическая величина, “Да” (TRUE) при запросе подписей, которые появляются при активном окне карты; “Нет” (FALSE) при запросе подписей, которые появляются, когда карта помещена в отчёт.

Возвращаемая величина

Логическая величина: “Да” (TRUE) означает, что подпись существует на определенном слое (либо подписи видимы в данный момент, либо пользователь их редактировал, либо эти отредактированные подписи в данный момент невидимы).

Описание

Вызовите функцию **LabelFindFirst()**, когда требуется запросить подписи на слое карты в цикле. Процедура запроса подписей состоит из двух шагов:

1. Установите внутренний указатель подписи MapBasic. Для этого используется **функция LabelFindFirst()**, **функция LabelFindNext()** или **функция LabelFindByID()**.
2. Если функция, которую была вызвана на шаге 1, не возвращает “Нет” (FALSE), можно запросить текущую подпись, используя **функция LabelInfo()**.

Для продолжения запроса дополнительных подписей, вернитесь к шагу 1.

Пример

См. пример в разделе о **функции LabelInfo()**.

См. также:

функция LabelFindByID(), **функция LabelFindNext()**, **функция LabelInfo()**

Функция LabelFindNext()

Назначение

Перемещает внутренний указатель подписи, так что можно запрашивать следующую подпись на слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LabelFindNext( map_window_id, layer_number )
```

map_window_id – ID-номер окна Карты;

layer_number – номер слоя в текущем окне Карты (например, 1 для верхнего слоя);

Возвращаемая величина

Логическая величина: “Да” (TRUE) означает, что указатель подписи передвинут к следующей подписи; “Нет” (FALSE) означает, что на этом слое более нет подписей.

Описание

После того, как вызвана **функция LabelFindFirst()** можно вызывать функцию **LabelFindNext()**, чтобы переместиться к следующей подписи на этом же слое

Пример

См. пример в разделе о **функции LabelInfo()**.

См. также:

функция LabelFindByID(), функция LabelFindFirst(), функция LabelInfo()

Функция LabelInfo()

Назначение

Возвращает информацию о подписи на карте. LabelInfo возвращает подписи как текстовые объекты. Эти текстовые объекты могут быть либо размещены вдоль кривой, либо повернуты под углом. Однако, если подписи выполнены вдоль кривой, то они будут возвращены как повернутый текст вдоль прямой. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Labelinfo( map_window_id, layer_number, attribute )
```

map_window_id – ID-номер окна Карты;

layer_number – номер слоя в текущем окне Карты (например, 1 для верхнего слоя);

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра attribute.

Описание

Функция **Labelinfo()** возвращает информацию о подписи в окне Карты.

Примечание: Подписи и текстовые объекты – это не одно и то же. Для запроса текстовых объектов используется **функция ObjectInfo()** или **функция ObjectGeography()**.

Перед вызовом **Labelinfo()** необходимо инициализировать внутренний указатель подписей MapBasic. Для этого используются **функция LabelFindFirst()**, **функция LabelFindNext()** или **функция LabelFindByID()**. Смотрите пример ниже.

Параметр **attribute** должен принимать одно из следующих значений в таблице; коды определены в **MAPBASIC.DEF**.

код атрибута	ID	Возвращаемое значение Labelinfo()
LABEL_INFO_OBJECT	1	Возвращает текстовый объект, заменяющий подпись. Эта операция позволяет конвертировать подпись в текстовый объект, который Вы можете сохранить в постоянной таблице. Примечание: LABEL_INFO_OBJECT возвращает текстовый объект, но если подпись выполнена вдоль кривой, то она будет возвращена в неискривленном виде. MapBasic не обеспечивает работу с подписями вдоль кривой как с текстовыми объектами.
LABEL_INFO_POSITION	2	Целочисленное значение от 0 до 8, показывающее положение подписи относительно места привязки. Величина, полученная в результате соответствует одному из следующих кодов: • LAYER_INFO_LBL_POS_AUTO (-1), • LAYER_INFO_LBL_POS_CC (0), • LAYER_INFO_LBL_POS_TL (1), • LAYER_INFO_LBL_POS_TC (2), • LAYER_INFO_LBL_POS_TR (3), • LAYER_INFO_LBL_POS_CL (4), • LAYER_INFO_LBL_POS_CR (5), • LAYER_INFO_LBL_POS_BL (6), • LAYER_INFO_LBL_POS_BC (7), • LAYER_INFO_LBL_POS_BR (8). Например, если подпись ниже и правее места привязки, ее позиция обозначается кодом 8; если подпись располагается выше и левее места привязки, ее код будет равен 1. Если подпись позиционируется автоматически, то ей соответствует код -1.

код атрибута	ID	Возвращаемое значение Labelinfo()
LABEL_INFO_ANCHORX	3	Величина типа Float, указывает X-координату места привязки подписи.
LABEL_INFO_ANCHORY	4	Величина типа Float, указывает Y-координату места привязки подписи.
LABEL_INFO_OFFSET	5	Целочисленное значение от 0 до 200, показывающее расстояние (в точках) от подписи до места прикрепления.
LABEL_INFO_ROWID	6	Целочисленное значение, представляющее ID-номер строки, в которой расположена подпись; возвращает 0, если подписи не существует.
LABEL_INFO_TABLE	7	Строчковая величина, представляющая имя таблицы, в которой находится данная подпись. Используется при работе со сшитой таблицей, в том случае, если Вам надо узнать в какой из составляющих ее таблиц находится данная подпись.
LABEL_INFO_EDIT	8	Логическая величина; "Да" (TRUE) если подпись редактировалась.
LABEL_INFO_EDIT_VISIBILITY	9	Логическая величина; "Да" (TRUE) если выбран режим видимости подписи "Скрыть" (OFF).
LABEL_INFO_EDIT_ANCHOR	10	Логическая величина; "Да" (TRUE) если подпись перемещалась.
LABEL_INFO_EDIT_OFFSET	11	Логическая величина; "Да" (TRUE) если расстояние в точках от подписи до места прикрепления изменилось.
LABEL_INFO_EDIT_FONT	12	Логическая величина; "Да" (TRUE) если шрифт подписи изменился.
LABEL_INFO_EDIT_PEN	13	Логическая величина; "Да" (TRUE) если стиль линии указки изменился.
LABEL_INFO_EDIT_TEXT	14	Логическая величина; "Да" (TRUE) если текст подписи изменился.
LABEL_INFO_EDIT_TEXTARROW	15	Логическая величина; "Да" (TRUE) если стрелка указки подписи изменилась.
LABEL_INFO_EDIT_ANGLE	16	Логическая величина; "Да" (TRUE) если угол наклона подписи изменился.

код атрибута	ID	Возвращаемое значение Labelinfo()
LABEL_INFO_EDIT_POSITION	17	Логическая величина; "Да" (TRUE) если позиция подписи (относительно места прикрепления) изменилась.
LABEL_INFO_EDIT_TEXTLINE	18	Логическая величина; "Да" TRUE если указка переместилась.
LABEL_INFO_SELECT	19	Логическая величина; "Да"(TRUE) если подпись выбрана.
LABEL_INFO_DRAWN	20	Логическая величина; "Да" (TRUE) если подпись в текущий момент видима.
LABEL_INFO_ORIENTATION	21	Возвращает короткое целое, описывающее ориентацию подписи. Подпись создается одной из следующих функций: LabelFindFirst, LabelFindByID или LabelFindNext. Результатом может быть один из следующих кодов: • LABEL_INFO_LABEL_ORIENT_HORIZONTAL (подпись под углом равным 0) • LABEL_INFO_LABEL_ORIENT_PARALLEL (подпись под любым углом отличным от 0) • LABEL_INFO_LABEL_ORIENT_CURVED (подпись по кривой)

Пример

Следующий пример показывает, как в цикле проводится поиск подписи среди других подписей и используется функция **Labelinfo()**, чтобы запросить каждую подпись.

```
Dim b_morelabels As Logical
Dim i_mapid, i_layernum As Integer
Dim obj_mytext As Object
' Here, you would assign a Map window's ID to i_mapid,
' and assign a layer number to i_layernum.
b_morelabels = LabelFindFirst(i_mapid, i_layernum, TRUE)
Do While b_morelabels
  obj_mytext = LabelInfo(i_mapid, i_layernum, LABEL_INFO_OBJECT)
  ' At this point, you could save the obj_mytext object
  ' in a permanent table; or you could query it by
  ' calling ObjectInfo() or ObjectGeography().
  b_morelabels = LabelFindNext(i_mapid, i_layernum)
Loop
```

См. также:

функция LabelFindByID(), функция LabelFindFirst(), функция LabelFindNext()

Функция LabelOverrideInfo()

Возвращает информацию о режиме единообразного подписывания.

Синтаксис

```
LabelOverrideInfo (
    window_id, layer_number, labeloverride_index, attribute )
```

window_id – целочисленный идентификатор окна Карты.

layer_number – номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

labeloverride_index – целочисленный индекс (начинающийся с единицы 1) описания режима единообразного подписывания на данном слое. Каждое описание режима единообразного подписывания привязано к масштабному коэффициенту и упорядочено так, что единообразный режим показа для минимального масштаба находится сверху (индекс равен 1).

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип величины зависит от значения attribute.

Описание

Функция LabelOverrideInfo() возвращает описание режимов единообразного подписывания на слое в существующем окне карты. Число *layer_number* должно корректно задавать стиль (1 представляет верхний табличный стиль и т.д.). Параметр *attribute* должен принимать одно из следующих значений в таблице; коды определены в MAPBASIC.DEF.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_NAME	1	Название изменения подписи.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_VISIBILITY	2	Короткое целое (SmallInt), индикатор режима видимости подписей. Результатом может быть один из следующих кодов: • LBL_OVR_INFO_VIS_OFF (0) единообразные подписи не показывается никогда • LBL_OVR_INFO_VIS_ON (1) единообразные подписи видны на карте • LBL_OVR_INFO_VIS_OFF_ZOOM (2) единообразные подписи не видны на карте, потому что задан другой масштаб
LBL_OVR_INFO_ZOOM_MIN	3	Число (тип Float), минимальный порог для масштабного эффекта (в текущих в MapBasic единицах измерения расстояний), при котором измененная подпись показывается.
LBL_OVR_INFO_ZOOM_MAX	4	Число (тип Float), максимальный порог для масштабного эффекта (в текущих в MapBasic единицах измерения расстояний), при котором измененная подпись показывается.
LBL_OVR_INFO_EXPR	5	Строка: выражение, создающее подписи.
LBL_OVR_INFO_LT	6	Целочисленный код (тип SmallInt), показывающий, какой тип указки используется. Возвращаются следующие значения: • LAYER_INFO_LBL_LT_NONE (0) нет линии • LAYER_INFO_LBL_LT_SIMPLE (1) простая линия • LAYER_INFO_LBL_LT_ARROW (2) линия со стрелкой
LBL_OVR_INFO_FONT	7	Стиль шрифта для подписей.
LBL_OVR_INFO_PARALLEL	8	Логическая величина: "Да" (TRUE), если на слое разрешено поворачивать подписи.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_POS	9	Целочисленный код (тип SmallInt), показывающий ориентацию подписи. Литера Т обозначает верх, литера В – низ, С – центр, R – право, L – лево. • LAYER_INFO_LBL_POS_CC (0) • LAYER_INFO_LBL_POS_TL (1) • LAYER_INFO_LBL_POS_TC (2) • LAYER_INFO_LBL_POS_TR (3) • LAYER_INFO_LBL_POS_CL (4) • LAYER_INFO_LBL_POS_CR (5) • LAYER_INFO_LBL_POS_BL (6) • LAYER_INFO_LBL_POS_BC (7) • LAYER_INFO_LBL_POS_BR (8)
LBL_OVR_INFO_OVERLAP	10	Логическая величина: "Да" (TRUE), если допускается пересечение подписей.
LBL_OVR_INFO_DUPLICATES	11	Логическая величина: "Да" (TRUE), если допускается дублирование подписей.
LBL_OVR_INFO_OFFSET	12	Целочисленная величина типа Smallint от 0 до 50, смещение от подписи до центроида. Измеряется в точках (points).
LBL_OVR_INFO_MAX	13	Целое число, максимальное количество подписей, для которых можно изменять данный режим показа. Если такое число не назначено, то возвращается число 2 147 483 647.
LBL_OVR_INFO_PARTIALSEGS	14	Логическая величина; TRUE, если для этого слоя установлен флажок Подписывать части объектов.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_ORIENTATION	15	LAYER_INFO_LABEL_ORIENTATION – возвращает короткое целое, описывающее ориентацию автоматически создаваемых подписей. Результатом будет один из следующих кодов: • LAYER_INFO_LABEL_ORIENT_HORIZONTAL подписи показаны под углом 0 градусов • LAYER_INFO_LABEL_ORIENT_PARALLEL подписи показаны под ненулевым углом • LAYER_INFO_LABEL_ORIENT_CURVED подписи по кривой Если возвращается LBL_OVR_LABEL_ORIENT_PARALLEL, то LAYER_INFO_INFO_PARALLEL возвращает TRUE.
LBL_OVR_INFO_ALPHA	16	Короткое целое (SmallInt) представляющее альфа-фактор для подписей на данном слое. • 0 = полностью прозрачные. • 255 = полностью непрозрачные. Для задания режима полупрозрачности или альфы-слоя используйте предложение Label оператора Set Map. См. раздел Управление настройками подписей на слое.
LBL_OVR_INFO_AUTODISPLAY	17	Логическое значение: TRUE, если единообразный режим предписывает показывать подписи автоматически.
LBL_OVR_INFO_POS_RETRY	18	Логическое значение: TRUE, если подписям разрешено накладываться друг на друга, при этом предварительно пробуются разные варианты размещения подписей без наложения.
LBL_OVR_INFO_LINE_PEN	19	Стиль линии, используемой для указки.
LBL_OVR_INFO_PERCENT_OVER	20	Короткое целое (SmallInt), максимальный процент размера закругленной подписи, выступающий за полилинию.
LBL_OVR_INFO_AUTO_POSITION	21	Логическое значение: индикатор режима уточненного подписывания регионов.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_AUTO_SIZES	22	Целое, задает количество размеров шрифта, которые будут использованы при попытке поместить подпись в область в режиме единообразного подписывания. Значение меняется в диапазоне от 1 до 10. Значение 0 (ноль) соответствует режиму Default , при котором MapInfo Pro самостоятельно выбирает размеры шрифтов.
LBL_OVR_INFO_SUPPRESS_IF_NO_FIT	23	Логическое значение –TRUE, если включен этот режим подписей On . Если этот режим действует, но после уменьшения размера шрифта подпись все еще не помещается в область, то она не показывается.
LBL_OVR_INFO_AUTO_SIZE_STEP	24	Целое, задает долю в процентах для уменьшения шрифта в режиме автоматического уменьшения шрифта. Если размер шрифта изначально был равен 24 пунктов и шаг уменьшения был задан значением 66, то уменьшенный шрифт будет на 66% меньше исходного (то есть, 66% от 24 pt = 8 пунктов).
LBL_OVR_INFO_CURVED_BEST_POSITION	25	Логическое значение, индикатор режима автоматического размещения подписей по кривой.
LBL_OVR_INFO_CURVED_FALLBACK	26	Логическое значение, индикатор режима замены повернутой подписи.
LBL_OVR_INFO_USE_ABBREVIATION	27	Логическое значение, индикатор режима сокращения подписей.
LBL_OVR_INFO_ABBREVIATION_EXPR	28	Возвращает выражение, по которому создается сокращенная подпись.
LBL_OVR_INFO_AUTO_CALLOUT	29	Логическое значение: TRUE, если установлен режим уточненного подписывания для указок.
LBL_OVR_INFO_EXPR_EDITED	30	Логическое значение, описывающее состояние свойства выражения, – редактировалось оно или нет (в сравнении с базовым, основным слоем).
LBL_OVR_INFO_LT_EDITED	31	Логическое значение, описывающее состояние свойств типа линии, – редактировалось оно или нет.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_FONT_EDITED	32	Логическое значение, описывающее состояние свойства шрифта, – редактировалось оно или нет.
LBL_OVR_INFO_PARALLEL_EDITED	33	Логическое значение, описывающее состояние свойства ориентации, – редактировалось оно или нет.
LBL_OVR_INFO_POS_EDITED	34	Логическое значение, описывающее состояние свойства местоположения, – редактировалось оно или нет.
LBL_OVR_INFO_OVERLAP_EDITED	35	Логическое значение, описывающее состояние свойства наложения, – редактировалось оно или нет.
LBL_OVR_INFO_DUPLICATES_EDITED	36	Логическое значение, описывающее состояние свойства дублей подписей, – редактировалось оно или нет.
LBL_OVR_INFO_OFFSET_EDITED	37	Логическое значение, описывающее состояние свойства смещения подписей, – редактировалось оно или нет.
LBL_OVR_INFO_MAX_EDITED	38	Логическое значение, описывающее состояние свойств максимума подписей, – редактировалось оно или нет.
LBL_OVR_INFO_PARTIALSEGS_EDITED	39	Логическое значение, описывающее состояние свойств сегментов, – редактировалось оно или нет.
LBL_OVR_INFO_ORIENTATION_EDITED	40	Логическое значение, описывающее состояние свойства ориентации, – редактировалось оно или нет.
LBL_OVR_INFO_ALPHA_EDITED	41	Логическое значение, описывающее состояние альфа-свойства, – редактировалось оно или нет.
LBL_OVR_INFO_AUTODISPLAY_EDITED	42	Логическое значение, описывающее состояние свойства автоматического отображения подписей, – редактировалось оно или нет.
LBL_OVR_INFO_POS_RETRY_EDITED	43	Логическое значение, описывающее состояние свойства повторного размещения, – редактировалось оно или нет.

Код атрибута	ID	Возвращаемое значение LabelOverrideInfo()
LBL_OVR_INFO_LINE_PEN_EDITED	44	Логическое значение, описывающее состояние свойств штриховки линии, – редактировалось оно или нет.
LBL_OVR_INFO_PERCENT_OVER_EDITED	45	Логическое значение, описывающее состояние свойства процентного отношения, – редактировалось оно или нет.
LBL_OVR_INFO_AUTO_POSITION_EDITED	46	Логическое значение, описывающее состояние свойства автоматического размещения, – редактировалось оно или нет.
LBL_OVR_INFO_AUTO_SIZES_EDITED	47	Логическое значение, описывающее состояние свойства автоматического размера подписей, – редактировалось оно или нет.
LBL_OVR_INFO_SUPPRESS_IF_NO_FIT_EDITED	48	Логическое значение, описывающее состояние свойства автоматического усечения, если подписи не помещаются, – редактировалось оно или нет.
LBL_OVR_INFO_AUTO_SIZESTEP_EDITED	49	Логическое значение, описывающее состояние свойства автоматического шага изменения размера – редактировалось оно или нет.
LBL_OVR_INFO_CURV_BEST_POS_EDITED	50	Логическое значение, описывающее состояние свойства размещения по кривой, – редактировалось оно или нет.
LBL_OVR_INFO_CURV_FALLBACK_EDITED	51	Логическое значение, описывающее состояние свойства свески при размещении по кривой, – редактировалось оно или нет.
LBL_OVR_INFO_USE_ABBREV_EDITED	52	Логическое значение, описывающее состояние свойства замены аббревиатурами, – редактировалось оно или нет.
LBL_OVR_INFO_ABBREV_EXPR_EDITED	53	Логическое значение, описывающее состояние свойства замены выражениями с аббревиатурами, – редактировалось оно или нет.
LBL_OVR_INFO_AUTO_CALLOUT_EDITED	54	Логическое значение, описывающее состояние свойства автоматических сносок для подписей, – редактировалось оно или нет.

Пример

```
LabelOverrideInfo(nMID, nLayer, nOverride, LBL_OVR_INFO_ORIENTATION)
```

См. также:

[функция StyleOverrideInfo\(\)](#), [функция LayerStyleInfo\(\)](#), [оператор Set Map](#), [функция LayerInfo\(\)](#)

Функция LayerControlInfo()

Назначение

Возвращает информацию об окне Управление слоями.

Синтаксис

```
LayerControlInfo( attribute )
```

attribute – целочисленный код, см. таблицу ниже.

Описание

Значение параметра *attribute* показаны в таблице ниже. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код атрибута	ID	TableInfo() возвращает
LC_INFO_SEL_COUNT	1	Короткое целое (Smallint result), количество выбранных объектов.

Пример

```
LayerControlInfo(LC_INFO_SEL_COUNT)
```

См. также:

[функция LayerControlSelectionInfo\(\)](#)

Функция LayerControlSelectionInfo()

Назначение

Возвращает информацию о выбранном объекте в окне Управление слоями.

Синтаксис

```
LayerControlSelectionInfo( selection_index, attribute )
```

selection_id – номер выбранного объекта в окне Управление слоями.

attribute – целочисленный код, см. таблицу ниже.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код атрибута	ID	LayerControlSelectionInfo() возвращает
LC_SEL_INFO_NAME	1	Строка, представляющая имя выбранного объекта.
LC_SEL_INFO_TYPE	2	Короткое целое (Smallint), тип выбранного объекта. Результатом будет один из следующих кодов: • LC_SEL_INFO_TYPE_MAP (0) • LC_SEL_INFO_TYPE_LAYER (1) • LC_SEL_INFO_TYPE_GROUPLAYER (2) • LC_SEL_INFO_TYPE_STYLE_OVR (3) • LC_SEL_INFO_TYPE_LABEL_OVR (4) • LC_SEL_INFO_TYPE_THEME_ITEM (5)
LC_SEL_INFO_MAPWIN_ID	3	Целое число, ID-номер окна карты, ассоциированной с выбранным объектом.
LC_SEL_INFO_LAYER_ID	4	Короткое целое, ID-номер слоя, ассоциированного с выбранным объектом. Если запросить это значение в тот момент, когда выбрана карта, будет возвращена -1. Для элемента тематической категории этот идентификатор слоя, относящийся к какому-либо тематическому слою.

код атрибута	ID	LayerControlSelectionInfo() возвращает
LC_SEL_INFO_OVR_ID	5	Короткое целое, индекс описания единообразного оформления выбранного объекта. Если запросить это значение в тот момент, когда выбрана карта, слой или группа слоев, будет возвращена -1.

Пример

```
LayerControlSelectionInfo(layer_number, LC_SEL_INFO_NAME)
```

См. также:

функция [LayerControlInfo\(\)](#)

Функция LayerInfo()

Назначение

Возвращает информацию о слое в окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LayerInfo( window_id, layer_number, attribute )
```

window_id — целочисленный идентификатор окна Карты.

layer_number — номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

attribute — целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип величины зависит от значения attribute.

Предупреждение!

Многие коды из приведенных выше не применимы к косметическому слою, тематическому слою и слою с растровым изображением. Функцию [LayerInfo\(\)](#) с аргументом

LAYER_INFO_TYPE используют для определения, каким является слой. Пример смотрите ниже.

Описание

Функция **LayerInfo()** возвращает информацию об определенном слое в окне карты. Число *layer_number* должно корректно задавать стиль (1 представляет верхний табличный стиль и т.д.). Параметр *attribute* должен быть одним из кодов из следующей таблицы; коды определяются в файле MAPBASIC.DEF. Здесь также можно выполнить запрос об опциях геолинков, проверяя атрибуты LAYER_HOTLINK_.*.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_NAME	1	Строка (тип String), содержащая имя таблицы, данные которой представлены на этом слое. Если слой косметический, то строка будет именоваться "Cosmetic1"; это имя может быть использовано в других операторах (см., например, Оператор Select).
LAYER_INFO_EDITABLE	2	Логическая величина (тип Logical), TRUE если слой изменяем.
LAYER_INFO_SELECTABLE	3	Логическая величина (тип Logical), TRUE если слой доступен.
LAYER_INFO_ZOOM_LAYERED	4	Логическая величина (Logical), TRUE если применяется ли масштабный эффект для данного слоя.
LAYER_INFO_ZOOM_MIN	5	Число (тип Float), минимальный порог для масштабного эффекта (в текущих в MapBasic единицах измерения расстояний). Для установки единиц измерения расстояний используйте оператор Set Distance Units .
LAYER_INFO_ZOOM_MAX	6	Число (тип Float), максимальный порог для масштабного эффекта.
LAYER_INFO_COSMETIC	7	Логическая величина (тип Logical). TRUE, если данный слой косметический.
LAYER_INFO_PATH	8	Строка (тип String), содержащая DOS-маршрут, по которому находится файл таблицы, данные из которой представлены на этом слое.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_DISPLAY	9	Короткое целое число (SmallInt), задающее режим показа слоя: • LAYER_INFO_DISPLAY_OFF (0) слой не показывается. • LAYER_INFO_DISPLAY_GRAPHIC (1) объекты на слое показываются в собственном стиле, то есть в том, который сохранен в таблице. • LAYER_INFO_DISPLAY_GLOBAL (2) объекты на слое показываются в стиле, заданном в диалоге Управление слоями. • LAYER_INFO_DISPLAY_VALUE (3) объекты на слое показываются в стиле, заданном режимам тематического выделения.
LAYER_INFO_OVR_LINE	10	Стиль линии, используемый для отображения линейных объектов. Если в свойствах слоя используются комбинированные стили, то возвращается стиль первой отрисовываемой линии.
LAYER_INFO_OVR_PEN	11	Стиль линии, используемый для отображения контуров объектов, имеющих площадь. Если в свойствах слоя используются комбинированные стили, то возвращается стиль первой отрисовываемой линии.
LAYER_INFO_OVR_BRUSH	12	Стиль штриха, используемый для отображения объектов, имеющих площадь. Если в свойствах слоя используются комбинированные стили, то возвращается стиль первой отрисовываемой штриховки.
LAYER_INFO_OVR_SYMBOL	13	Стиль символа, используемый для отображения точечных объектов. Если в свойствах слоя используются комбинированные стили, то возвращается стиль первого отрисовываемого символа.
LAYER_INFO_OVR_FONT	14	Стиль шрифта, используемый для отображения текстовых объектов. Если в свойствах слоя используются комбинированные стили, то возвращается стиль первого отрисовываемого шрифта.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_LBL_EXPR	15	Строка: выражение, создающее подписи.
LAYER_INFO_LBL_LT	16	Целочисленный код (тип SmallInt), показывающий, какой тип указки используется. Возвращаются следующие значения: • LAYER_INFO_LBL_LT_NONE (0) нет линии • LAYER_INFO_LBL_LT_SIMPLE (1) простая линия • LAYER_INFO_LBL_LT_ARROW (2) линия со стрелкой
LAYER_INFO_LBL_CURFONT	17	В приложениях, откомпилированных в среде MapBasic версии 3.x, результатом будет: Логическая величина: "Да" (TRUE), если слой использует текущую установку для стиля шрифта, или "Нет" (FALSE), если слой использует специальную установку для стиля шрифта (смотрите LAYER_INFO_LBL_FONT). В приложениях, откомпилированных с MapBasic версии 4.0 и выше, результат всегда будет FALSE.
LAYER_INFO_LBL_FONT	18	Стиль шрифта для подписей.
LAYER_INFO_LBL_PARALLEL	19	Логическая величина: "Да" (TRUE), если на слое разрешено поворачивать подписи.
LAYER_INFO_LBL_POS	20	Целочисленный код (тип SmallInt), показывающий ориентацию подписи. Литера T обозначает верх, литера B – низ, C – центр, R – право, L – лево. • LAYER_INFO_LBL_POS_TL (1) • LAYER_INFO_LBL_POS_TC (2) • LAYER_INFO_LBL_POS_TR (3) • LAYER_INFO_LBL_POS_CL (4) • LAYER_INFO_LBL_POS_CC (0) • LAYER_INFO_LBL_POS_CR (5) • LAYER_INFO_LBL_POS_BL (6) • LAYER_INFO_LBL_POS_BC (7) • LAYER_INFO_LBL_POS_BR (8)

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_ARROWS	21	Логическая величина: TRUE, если в линейных объектах используется стрелка.
LAYER_INFO_NODES	22	Логическая величина: TRUE, если показываются узлы объектов слоя.
LAYER_INFO_CENTROIDS	23	Логическая величина: TRUE, если показываются центры тяжести объектов слоя.
LAYER_INFO_TYPE	24	Целочисленный код (тип SmallInt), показывающий тип слоя: • LAYER_INFO_TYPE_NORMAL (0) для обычного слоя • LAYER_INFO_TYPE_COSMETIC (1) для косметического слоя • LAYER_INFO_TYPE_IMAGE (2) для растрового слоя • LAYER_INFO_TYPE_THEMATIC (3) для тематического слоя • LAYER_INFO_TYPE_GRID (4) для слоя поверхности • LAYER_INFO_TYPE_WMS (5) для слоя, полученного от службы Web Map Service • LAYER_INFO_TYPE_TILESERVER (6) для слоя, полученного от Tile Server • LAYER_INFO_TYPE_GRIDLINE (7) для слоя линий сетки
LAYER_INFO_LBL_VISIBILITY	25	Целочисленная величина, типа Smallint соответствующая режиму показа подписей на слое; смотрите описание предложения Visibility в разделе Оператор Set Map. Результатом будет один из следующих кодов: • LAYER_INFO_LBL_VIS_ON (3) подписи показываются всегда • LAYER_INFO_LBL_VIS_OFF (1) подписи не показываются • LAYER_INFO_LBL_VIS_ZOOM (2) подписи показываются в определенном масштабном диапазоне

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_LBL_ZOOM_MIN	26	Действительная величина (тип Float), минимальное значение показа подписей при масштабном эффекте.
LAYER_INFO_LBL_ZOOM_MAX	27	Действительная величина (тип Float), максимальное значение показа подписей при масштабном эффекте.
LAYER_INFO_LBL_AUTODISPLAY	28	Логическая величина: TRUE, если слой подписывается автоматически. Смотрите описание предложения Auto в разделе Оператор Set Map .
LAYER_INFO_LBL_OVERLAP	29	Логическая величина: "Да" (TRUE), если допускается пересечение подписей.
LAYER_INFO_LBL_DUPLICATES	30	Логическая величина: "Да" (TRUE), если допускается дублирование подписей.
LAYER_INFO_LBL_OFFSET	31	Целочисленная величина типа Smallint от 0 до 50, смещение от подписи до центроида. Измеряется в точках (points).
LAYER_INFO_LBL_MAX	32	Целочисленная величина типа Integer, максимальное число подписей, разрешенное для этого слоя. Если такое число не назначено, то возвращается число 2 147 483 647.
LAYER_INFO_LBL_PARTIALSEGS	33	Логическая величина; TRUE, если для этого слоя установлен флажок Подписывать части объектов .
LAYER_INFO_HOTLINK_EXPR	34	Возвращает выражение для файла, на который указывает геолнк. Допускается пустая строка ("")
LAYER_INFO_HOTLINK_MODE	35	Возвращает текущий режим геолнका, один из следующих: • HOTLINK_MODE_LABEL (0), стандартное значение • HOTLINK_MODE_OBJ (1) • HOTLINK_MODE_BOTH (2)
LAYER_INFO_HOTLINK_RELATIVE	36	Возвращает TRUE, если маршрут в геолнке записан в относительной форме, и FALSE в обратном случае. По умолчанию – FALSE.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_HOTLINK_COUNT	37	Позволяет задавать несколько геолинков на слое карты.
LAYER_INFO_LBL_ORIENTATION*	38	LAYER_INFO_LABEL_ORIENTATION — возвращает короткое целое, описывающее ориентацию автоматически создаваемых подписей. Результатом будет один из следующих кодов: • LAYER_INFO_LABEL_ORIENT_HORIZONTAL подписи показаны под углом 0 градусов • LAYER_INFO_LABEL_ORIENT_PARALLEL подписи показаны под ненулевым углом • LAYER_INFO_LABEL_ORIENT_CURVED подписи по кривой Если возвращается LBL_OVR_LABEL_ORIENT_PARALLEL, то LAYER_INFO_INFO_PARALLEL возвращает TRUE.
LAYER_INFO_LAYER_ALPHA	39	Короткое целое (SmallInt), представляющее альфа-фактор для данного слоя. • 0 = полностью прозрачные. • 255 = полностью непрозрачные. Для задания режима полупрозрачности или альфы для слоя используйте оператор Set Map.
LAYER_INFO_LAYER_TRANSLUCENCY	40	Короткое целое (SmallInt), представляющее процент полупрозрачности для данного слоя. • 100 = полностью прозрачные. • 0 = полностью непрозрачные. Для задания режима полупрозрачности или альфы для слоя используйте оператор Set Map.
LAYER_INFO_LABEL_ALPHA	41	Короткое целое (SmallInt) представляющее альфа-фактор для подписей на данном слое. • 0 = полностью прозрачные. • 255 = полностью непрозрачные. Для задания режима полупрозрачности или альфы-слоя используйте оператор Set Map с предложением Layer.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_LAYERLIST_ID	42	Возвращает общий числовой идентификатор слоя в текущем списке слоёв. Например, слой может быть первым в группе в списке слоёв карты (групповой идентификатор этого слоя будет 1), но он может четвертым в общем списке. Таким образом идентификатором списка слоёв будет 4. Этот идентификатор может использоваться функцией LayerListInfo() .
LAYER_INFO_PARENT_GROUP_ID	43	Возвращает групповой идентификатор слоя непосредственно в группе, в которую входит этот слой, будет возвращено значение равное 0, если слой первый в общем списке
LAYER_INFO_OVR_STYLE_COUNT	44	Возвращает короткое целое (Smallint), количество единообразных стилей.
LAYER_INFO_OVR_LBL_COUNT	45	Возвращает короткое целое (Smallint), количество единообразных режимов подписывания.
LAYER_INFO_OVR_STYLE_CURRENT	46	Возвращает короткое целое Smallint, индекс изменённых стилей оформления при текущем масштабе показа, 0 означает, что изменений стилей нет.
LAYER_INFO_OVR_LBL_CURRENT	47	Возвращает короткое целое (Smallint), номер режима единообразного подписывания в данном масштабном диапазоне, 0 означает, что единообразный режим не задан.
LAYER_INFO_OVR_LINE_COUNT	48	Возвращает короткое целое (Smallint), количество стилей Pen, заданных для показа линейных объектов в базовом наборе свойств стиля.
LAYER_INFO_OVR_PEN_COUNT	49	Возвращает короткое целое (Smallint), количество стилей Pen, заданных для показа границ площадных объектов в базовом наборе свойств стиля.
LAYER_INFO_OVR_BRUSH_COUNT	50	Возвращает короткое целое (Smallint), количество стилей штриховки (Brush)), заданных для показа площадных объектов в базовом наборе свойств стиля.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_OVR_SYMBOL_COUNT	51	Возвращает короткое целое (Smallint), количество стилей символов, заданных для показа точечных объектов в базовом наборе свойств стиля.
LAYER_INFO_OVR_FONT_COUNT	52	Возвращает короткое целое (Smallint), количество шрифтов, заданных для показа текстовых объектов в базовом наборе свойств стиля. Это всегда 1, потому что шрифты не обрабатываются механизмом комбинированных стилей.
LAYER_INFO_TILE_SERVER_LEVEL	53	Короткое целое (Smallint), представляющее уровень тайл-сервера, представляющий слой в текущем окне карты. Значение -1 представляет слои, взятые не с тайл-сервера.
LAYER_INFO_LBL_AUTO_POSITION	54	Логическое значение: индикатор режима уточненного подписывания.
LAYER_INFO_LBL_AUTO_SIZES	55	Целое, задает количество размеров шрифта, которые будут использованы при попытке поместить подпись в область в режиме единообразного подписывания. Количество шрифтов может быть от 1 до 10. Значение 0 (нулевое) обозначает, что выбран стандартный вариант Default , когда программа MapInfo Pro самостоятельно определяет количество используемых шрифтов.
LAYER_INFO_LBL_SUPPRESS_IF_NO_FIT	56	Логическое значение, определяет, включен ли режим автоматического уменьшения подписей (On или Off). Если этот режим действует, но после уменьшения размера шрифта подпись все еще не помещается в область, то она не показывается.
LAYER_INFO_LBL_AUTO_SIZE_STEP	57	Целое, задает долю в процентах для уменьшения шрифта в режиме автоматического уменьшения шрифта. Если размер шрифта изначально был равен 24 пунктам, шаг уменьшения был задан значением 66, то уменьшенный шрифт будет на 66% меньше исходного (минимальный шрифт будет 8 пунктов).
LAYER_INFO_LBL_CURVED_BEST_POSITION	58	Логическое значение, индикатор режима автоматического размещения подписей по кривой.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_LBL_CURVED_FALLBACK	59	Логическое значение, индикатор режима замены повернутой подписи.
LAYER_INFO_LBL_USE_ABBREVIATION	60	Логическое значение, индикатор режима сокращения подписей.
LAYER_INFO_ABBREVIATION_EXPR	61	Возвращает выражение, по которому создается сокращенная подпись.
LAYER_INFO_LBL_AUTO_CALLOUT	62	Логическое значение: TRUE, если установлен режим уточненного подписывания для указок.
LAYER_INFO_LBL_ORDER	63	Короткое целое (Smallint), порядок подписывания на слое.
LAYER_INFO_FRIENDLYNAME	64	Строковое значение с "понятным" именем, присвоенным слою.
LAYER_INFO_GRIDLINE_COORDSYS	65	Строковое значение, содержащее название координатной системы, использованной при создании сетки линий. Пустое значение, если слой не является сеткой линий.
LAYER_INFO_GRIDLINE_SPACE_HORIZONTAL	66	Вещественное (Float) значение, в котором возвращается текущий, актуальный шаг карты сетки линий на экране, а не запрашиваемое значение. Если используется автоматический шаг сетки, то этот параметр соответствует горизонтальному шагу. Если линии ближе чем минимальный разрешенный шаг 'бумажных' единиц, то в этом параметре возвращается горизонтальный шаг, с которым выполняется отображение на карте (которое может быть большим чем запрашиваемое значение). Возвращается 0.0, когда слой не является слоем сетки линий.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_GRIDLINE_SPACE_VERTICAL	67	Вещественное (Float) значение, в котором возвращается текущий, актуальный шаг карты сетки линий на экране, а не запрашиваемое значение. Если используется автоматический шаг сетки, то этот параметр соответствует вертикальному шагу. Если линии ближе чем минимальный разрешенный шаг 'бумажных' единиц, то в этом параметре возвращается вертикальный шаг, с которым выполняется отображение на карте (которое может быть большим чем запрашиваемое значение). Возвращается 0.0, когда слой не является слоем сетки линий.
LAYER_INFO_GRIDLINE_AUTO	68	Логическое значение, которое принимает значение TRUE, если слой является слоем с автоматическим шагом сетки линий.
LAYER_INFO_GRIDLINE_MIN_HORIZONTAL	69	Вещественное (Float) значение, с которым допускается минимальный шаг 'на бумаге' между горизонтальными линиями сетки, когда используется автоматический шаг линий. Значение в текущих 'бумажных' единицах измерения (смотрите: Оператор Set Paper Units). Возвращается 0, когда слой не является слоем сетки линий.
LAYER_INFO_GRIDLINE_MIN_VERTICAL	70	Вещественное (Float) значение, с которым допускается минимальный шаг 'на бумаге' между вертикальными линиями сетки, когда используется автоматический шаг линий. Значение в текущих 'бумажных' единицах измерения (смотрите: Оператор Set Paper Units). Возвращается 0, когда слой не является слоем сетки линий.
LAYER_INFO_GRIDLINE_PAPER_HORIZONTAL	71	Вещественное (Float) значение, в котором возвращается запрашиваемое расстояние 'на бумаге', если шаг линий является автоматическим, и актуальное 'бумажное' расстояние, если шаг линий не автоматический. Значение в текущих 'бумажных' единицах измерения (смотрите: Оператор Set Paper Units). Возвращается 0, когда слой не является слоем сетки линий.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_GRIDLINE_PAPER_VERTICAL	72	Вещественное (Float) значение, в котором возвращается запрашиваемое расстояние 'на бумаге', если шаг линий является автоматическим, и актуальное 'бумажное' расстояние, если шаг линий не автоматический. Значение в текущих 'бумажных' единицах измерения (смотрите: Оператор Set Paper Units). Возвращается 0, когда слой не является слоем сетки линий.
LAYER_INFO_GRIDLINE_OFFSET	73	Вещественное (Float) значение, представляющее текущее расстояние в 'бумажных' единицах подписей линий от края карты (внутрь). Значение в текущих 'бумажных' единицах измерения (смотрите: Оператор Set Paper Units). Возвращается -1, когда слой не является слоем сетки линий.
LAYER_INFO_LABEL_POS_RETRY	74	Логическое значение – TRUE, если задан параметр для подписей Пробовать другое положение , иначе – FALSE.
LAYER_INFO_GRIDLINE_HORIZONTAL_LABEL_POS	75	Вещественное (Float) значение текущего положения подписей горизонтальных линий сетки. Возвращаются следующие значения: • -1, когда слой не является слоем сетки линий. • GRIDLINE_HORIZONTAL_LABEL_NONE (0), когда у горизонтальных линий сетки нет подписей. • GRIDLINE_HORIZONTAL_LABEL_LEFT (1), когда подписи горизонтальных линий сетки только слева. • GRIDLINE_HORIZONTAL_LABEL_RIGHT (2), когда подписи горизонтальных линий сетки только справа. • GRIDLINE_HORIZONTAL_LABEL_BOTH (3), когда подписи горизонтальных линий сетки и слева и справа.

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_GRIDLINE_VERTICAL_LABEL_POS	76	Вещественное (Float) значение текущего положения подписей вертикальных линий сетки. Возвращаются следующие значения: • -1, когда слой не является слоем сетки линий. • GRIDLINE_VERTICAL_LABEL_NONE (0), когда у вертикальных линий сетки нет подписей. • GRIDLINE_VERTICAL_LABEL_TOP (1), когда подписи вертикальных линий сетки только сверху. • GRIDLINE_VERTICAL_LABEL_BOTTOM (2), когда подписи вертикальных линий сетки только снизу. • GRIDLINE_VERTICAL_LABEL_BOTH (3), когда подписи вертикальных линий сетки и сверху и снизу.
LAYER_INFO_GRIDLINE_LABEL_DIRECTION	77	Вещественное (Float) значение, определяющее направление подписей линий сетки. Возвращаются следующие значения: • -1, когда слой не является слоем сетки линий. • GRIDLINE_LABEL_DIRECTION_COMPASS (1), когда координатная система направляющих Широта/Долгота и подписи направляющих дополняются компасным направлением: N — на Север, S — на Юг, E — на Восток или W — на Запад. Числа всегда отображаются в виде положительных цифр. • GRIDLINE_LABEL_DIRECTION_PLUSMINUS (0), когда подписями сетки линий являются либо положительные, либо отрицательные числа.
LAYER_INFO_ALIAS	78	Строковое значение для имени слоя, которое перечисляется в списке слоёв MalpInfo Pro.
LAYER_INFO_LBL_LINE_PEN	79	Строковое значение для предложения, описывающего стиль линии, используемой в качестве линии-выноски подписи. Если растровый слой не поддерживает подписи, то возвращается стиль линии, используемый в сеансе работе.
LAYER_INFO_RASTER_CONTRAST	80	Короткое целое (SmallInt) значение, представляющее контрастность в процентах (в диапазоне 0-100%).
LAYER_INFO_RASTER_BRIGHTNESS	81	Короткое целое (SmallInt) значение, представляющее яркость в процентах (в диапазоне 0-100%).

Код атрибута	ID	Возвращаемое значение LayerInfo()
LAYER_INFO_RASTER_ALPHA	82	Короткое целое (SmallInt) значение, представляющее альфа-фактор для прозрачности растра (в диапазоне 0-255).
LAYER_INFO_RASTER_GREYSCALE	83	Логическое (Logical) значение, обозначающее использование тонов серого вместо стандартных цветов изображения.
LAYER_INFO_LABEL_EXPR_TABLE	84	Строковое значение, представляющее имя таблицы, используемой для выражений подписей.
LAYER_INFO_LBL_CURVED_PERCENT_OVER	85	Целое (Integer) значение, обозначающее значение в процентах, на которое подпись может быть продлено за пределы конца линии (например).
LAYER_INFO_LEGEND	86	Логическая величина: Возвращает логическую единицу — True, если слой поддерживает легенду

О геолинках

Для совместимости с предыдущими версиями поддерживается набор атрибутов из версий до 10.0, и возвращаются значения для первого определения геолинки слоя. Если геолинки не заданы, то эта функция будет возвращать следующие значения:

LAYER_INFO_HOTLINK_EXPR — пустая строка ("")

LAYER_INFO_HOTLINK_MODE — возвращает стандартное значение HOTLINK_MODE_LABEL

LAYER_INFO_HOTLINK_RELATIVE — возвращает стандартное значение FALSE

Примеры

Многие коды из приведенных выше не применимы к косметическому слою, тематическому слою и слою с растровым изображением. Функцию **LayerInfo()** с аргументом LAYER_INFO_TYPE используют для определения, каким является слой.

Для того чтобы определить, является ли слой карты обычным слоем, можно использовать параметры LAYER_INFO_TYPE, как показано ниже:

```
i_layer_type = LayerInfo( map_id, layer_number, LAYER_INFO_TYPE)

If i_layer_type = LAYER_INFO_TYPE_NORMAL Then
    ' ... then this is a "normal" layer
```

```
,
End If
```

Следующий пример иллюстрирует использование приоритетов подписей. Если приоритеты подписей расставлены следующим образом:

```
Set Map Window FrontWindow() LabelPriority 5, 1, 2, 4, 3
```

То следующая функция вернет 1, потому что стандартный порядок подписывания тот же, что и порядок слоев карты (снизу вверх).

```
LayerInfo(FrontWindow(), 5, LAYER_INFO_LBL_ORDER)
```

См. также:

[функция GroupLayerInfo](#), [функция LayerListInfo](#), [функция MapperInfo\(\)](#), [оператор Set Map](#)

Функция LayerListInfo()

Назначение

С помощью этой функции можно пронумеровать список графических слоёв и групп слоёв карты.

Синтаксис

```
LayerListInfo( map_window_id, numeric_counter, attribute )
```

map_window_id – идентификатор окна карты;

numeric_counter – значения от нуля (0) до MAPPER_INFO_ALL_LAYERS, номер слоя в окне Карты, исключая Косметический. См. описание значений MAPPER_INFO_ALL_LAYERS в разделе: [Функция MapperInfo\(\)](#).

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Эту функцию можно использовать итеративно для всех элементов списка слоёв карты, для которых номер слоя *numeric_counter* принимает значение от 0 до MAPPER_INFO_ALL_LAYERS.

В качестве атрибутов можно использовать:

attribute-код	ID	Описание
LAYERLIST_INFO_TYPE	1	Тип слоя: • LAYERLIST_INFO_TYPE_LAYER (0) • LAYERLIST_INFO_TYPE_GROUP (1)
LAYERLIST_INFO_NAME	2	Возвращает строку, имя слоя или группы слоев.
LAYERLIST_INFO_LAYER_ID	3	Возвращает число, ID-номер слоя. В дальнейшем этот номер может использовать функция LayerInfo() .
LAYERLIST_INFO_GROUPLAYER_ID	4	Возвращает число, ID-номер группы слоев. В дальнейшем этот номер может использовать функция GroupLayerInfo() .

Если обращение к типу возвращает графический слой, используйте функцию LayerInfo для получения атрибутов. Если обращение к типу возвращает группу слоев, используйте функцию GroupLayerInfo для получения атрибутов. Для перебора всех слоев в цикле, задавайте в качестве его предела значение MAPPER_INFO_ALL_LAYERS.

Задав ID-номер, равный нулю (0), Вы получите информацию о Косметическом слое.

См. также:

[функция GroupLayerInfo](#), [функция LayerInfo\(\)](#), [функция MapperInfo\(\)](#)

Функция LayerStyleInfo()

Возвращает информацию о стилях, составляющих комбинированный стиль.

Синтаксис

```
LayerStyleInfo (
    window_id, layer_number, override_index, pass_index, attribute )
```

window_id – целочисленный идентификатор окна Карты.

layer_number – номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

override_index – целочисленный номер, начиная с нуля, где 0 задает базовый набор компонентов стиля слоя, а числа 1 и далее задают наборы единообразного оформления.

pass_index – номер шага прорисовки комбинированного стиля ("прохода"), начинающийся с 1. Первый "проход" прорисовывается на экране в первую очередь, поверх него рисуется второй и т. д.

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип величины зависит от значения *attribute*.

Описание

Функция `LayerStyleInfo()` возвращает информацию о стилях, составляющих комбинированный стиль. Комбинированный стиль образуется из одного или нескольких описаний стилей. Например, линия может быть стилизована комбинацией из двух стилей: тонкая красная линия рисуется поверх более толстой темно-красной линии, что на языке MapBasic задается следующим оператором:

```
Line (7,2,12582912), Line (3,2,16736352)
```

Такой синтаксис предписывает нарисовать толстую темно-красную линию первой.

Число *layer_number* должно корректно задавать стиль (1 представляет верхний табличный стиль и т.д.). Параметр *attribute* должен принимать одно из следующих значений в таблице; коды определены в `MAPBASIC.DEF`.

Код атрибута	ID	Возвращаемое значение <code>LayerStyleInfo()</code>
STYLE_OVR_INFO_LINE	10	Стиль линии, используемый при заданном шаге прорисовки линейных объектов.
STYLE_OVR_INFO_PEN	11	Стиль линии, используемый при заданном шаге прорисовки границ площадных объектов.
STYLE_OVR_INFO_BRUSH	12	Стиль штриховки, используемый при заданном шаге прорисовки площадных объектов.
STYLE_OVR_INFO_SYMBOL	13	Стиль символа, используемый при заданном шаге прорисовки точечных объектов.
STYLE_OVR_INFO_FONT	14	Шрифт, используемый при заданном шаге прорисовки текстовых объектов.

Пример

```
LayerStyleInfo(nMID, nLayer, nOverride, nPass, STYLE_OVR_INFO_PEN)
```

См. также:

функция [StyleOverrideInfo\(\)](#), функция [LabelOverrideInfo\(\)](#), оператор [Set Map](#), функция [LayerInfo\(\)](#)

Функция LayoutInfo()

Назначение

Возвращает информацию об окне **Список**, в частности, ширину и высоту макета отчёта, поля, масштаб показа и положение центра отчёта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LayoutInfo( window_id, attribute )
```

window_id — идентификатор окна.

attribute — целочисленный код типа информации, которая будет возвращена. Значения его представлено в таблице ниже.

Возвращаемая величина

Зависит от параметра *attribute*, смотрите далее в описании.

Описание

Функция LayoutInfo() возвращает информацию об окне **Отчёт**. Эта функция не применима к окну Районирования.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Функция LayoutInfo() может возвращать различные значения параметров заданного окна **Отчёт**. Параметр *attribute* сообщает функции LayoutInfo(), какие данные об окне **Отчёт** возвращаются. Параметр *attribute* должен принимать одно из следующих значений в таблице; коды определены в MAPBASIC.DEF.

параметр attribute	ID	Возвращаемая величина
LAYOUT_INFO_NUM_ITEMS	1	Целое число типа Integer: возвращает количество элементов отчёта.
LAYOUT_INFO_WIDTH	2	Вещественное значение – Float: ширина макета в "бумажных" единицах.
LAYOUT_INFO_HEIGHT	3	Вещественное значение – Float: высота макета в "бумажных" единицах.
LAYOUT_INFO_LEFT_MARGIN	4	Вещественное значение – Float: ширина левого поля макета в "бумажных" единицах.
LAYOUT_INFO_RIGHT_MARGIN	5	Вещественное значение – Float: ширина правого поля макета в "бумажных" единицах.
LAYOUT_INFO_TOP_MARGIN	6	Вещественное значение – Float: ширина верхнего поля макета в "бумажных" единицах.
LAYOUT_INFO_BOTTOM_MARGIN	7	Вещественное значение – Float: ширина нижнего поля макета в "бумажных" единицах.
LAYOUT_INFO_ZOOM	8	Вещественное значение – Float: величина уровня масштабирования показа макета в процентах, по умолчанию используется значение равное 100.
LAYOUT_INFO_CENTER_X	9	Вещественное значение – Float: x-координата центра отчёта в "бумажных" единицах.
LAYOUT_INFO_CENTER_Y	10	Вещественное значение – Float: y-координата центра отчёта в "бумажных" единицах.
LAYOUT_INFO_NUM_PAGES	11	Целое число типа Integer: возвращает количество элементов отчёта.
LAYOUT_INFO_CUR_PAGE	12	Целое число типа Integer: номер видимой страницы или текущей страницы.
LAYOUT_INFO_GRID_VISIBILITY	13	Логическая величина: TRUE, если регулярная сеть видима, иначе — FALSE.
LAYOUT_INFO_GIRD_SNAP	14	Логическая величина: TRUE, если включено совмещение по ячейкам, иначе — FALSE.

параметр attribute	ID	Возвращаемая величина
LAYOUT_INFO_GRID_SIZE	15	Вещественное значение – Float: размер линий сетки или интервалов между линиями в текущих "бумажных" единицах измерения.
LAYOUT_INFO_SMART_GUIDES	16	Логическая величина: TRUE, если смарт-направляющие включены, иначе — FALSE.

По умолчанию "бумажные" единицы измерения — дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

См. также:

функция FrontWindow(), функция LayoutItemInfo(), оператор Set Paper Units, функция WindowID()

Функция LayoutItemID()

Назначение

Возвращает идентификатор рамки окна **Отчёт**. В многостраничном отчёте эта функция осуществляет поиск только среди выбранных элементов на текущей странице. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LayoutItemID( window_id, frame_name, frame_type )
```

window_id – идентификатор окна.

frame_name – строка, указывающая имя рамки. Используйте пустую строку "" для рамок без имени.

Frame_type – короткое целое SmallInt, определяющее тип искомой рамки отчёта. Используйте одно перечисленных ниже значений переменной LAYOUT_ITEM_INFO_TYPE.

Далее перечислены типы рамок, возвращаемые LAYOUT_ITEM_INFO_TYPE.

Тип рамки	ID	Описание рамки
LAYOUT_ITEM_TYPE_ANY	-1	Первая рамка с заданным именем независимо от того, какого типа рамка.
LAYOUT_ITEM_TYPE_EMPTY	0	Пустая рамка: рамка, которая не содержит контента.
LAYOUT_ITEM_TYPE_MAPPER	1	Рамка с картой.
LAYOUT_ITEM_TYPE_BROWSER	2	Рамка со списком.
LAYOUT_ITEM_TYPE_LEGEND	3	Рамка с легендой.
LAYOUT_ITEM_TYPE_TEXT	4	Рамка с текстом.
LAYOUT_ITEM_TYPE_SHAPE	5	Рамка с фигурами такими, как линии, прямоугольники, скруглённые прямоугольники или эллипсы.
LAYOUT_ITEM_TYPE_IMAGE	6	Рамка с изображением.

По умолчанию "бумажные" единицы измерения – дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

Возвращаемая величина

Идентификатор типа рамки возвращается в виде целого числа . Если объект не найден, возвращается значение -1.

Описание

Функция LayoutItemID() возвращает идентификатор рамки ID окна **Отчёт**. Эта функция не применима к окну Районирования.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

В многостраничном отчёте эта функция осуществляет поиск только среди выбранных элементов на текущей странице.

См. также:

функция FrontWindow(), функция LayoutInfo(), функция LayoutItemInfo(), оператор Set Paper Units, функция WindowID()

Функция LayoutItemInfo()

Назначение

Возвращает информацию о рамке окна **Отчёт**. В многостраничном отчёте эта функция возвращает информацию о рамке на текущей странице. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Эта функция очень похожа на функцию **LayoutPageItemInfo()**, которую можно использовать, если страница, которую требуется запросить, не является видимой в окне **Отчёт**.

Синтаксис

```
LayoutItemInfo( window_id, frame_id, attribute )
```

window_id — идентификатор окна **Отчёт**, целое число.

frame_id — число, указывающее какую рамку в окне Отчёт требуется опросить. Разделы пронумерованы от 1 до n, где n это номер в Легенде. Именами рамок являются строки максимальной длины в 255 символов, которые задаются предложением Name, либо когда создаётся рамка, либо когда рамка создаётся с помощью оператора Alter Designer Frame, либо когда изменяется имя поля в диалоге Свойства рамки отчёта.

attribute — это целочисленный код, указывающий, какой тип информации возвращается. Значения его представлено в таблице ниже.

Возвращаемая величина

Зависит от параметра *attribute*, смотрите далее в описании.

Описание

Функция LayoutItemInfo() возвращает информацию об окне **Отчёт**. Эта функция не применима к окну Районирования. В многостраничном отчёте эта функция возвращает информацию о рамке на текущей странице.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Функция LayoutItemInfo() может возвращать различные значения параметров заданного окна **Отчёт**. Параметр *attribute* сообщает функции LayoutItemInfo(), какие данные об окне **Отчёт**

возвращаются. Параметр `attribute` должен принимать одно из следующих значений в таблице; коды определены в `MAPBASIC.DEF`.

Параметр Attribute	ID	Возвращаемое значение LayoutItemInfo()
LAYOUT_ITEM_INFO_POS_X	1	Вещественное значение – Float: возвращает положение элемента в виде расстояния в "бумажных" единицах от левого края отчёта.
LAYOUT_ITEM_INFO_POS_Y	2	Вещественное значение – Float: возвращает положение элемента в виде расстояния в "бумажных" единицах от верхнего края отчёта.
LAYOUT_ITEM_INFO_WIDTH	3	Вещественное значение – Float: возвращает ширину элемента отчёта в "бумажных" единицах.
LAYOUT_ITEM_INFO_HEIGHT	4	Вещественное значение – Float: возвращает высоту элемента отчёта в "бумажных" единицах.
LAYOUT_ITEM_INFO_WIN	5	Целое число типа Integer: возвращает идентификатор окна элемента отчёта (рамки). Либо возвращает 0, если ассоциированного окна не существует.
LAYOUT_ITEM_INFO_SELECTED	6	Логическая величина: возвращает True, если элемент выбран.
LAYOUT_ITEM_INFO_ACTIVATED	7	Логическая величина: возвращает True, если элемент активирован (например, карта активирована после нажатия клавиши Alt и щелчка по рамке).
LAYOUT_ITEM_INFO_EMPTY	8	Логическая величина: возвращает True, если элемент является пустой рамкой, иначе — false.
LAYOUT_ITEM_INFO_LEGEND_FRAME_ID	9	Целое число типа Integer: для рамок с легендой. Возвращает целое — идентификатор рамки с легендой в родительском окне Отчёт . Это значение может быть использовано для вызова операторов Legend. Возвращает 0 для рамок без легенды.
LAYOUT_ITEM_INFO_LEGEND_DESIGNER_WINDOW	10	Целое число типа Integer: Для рамок с легендой. Возвращает целое — идентификатор окна для родительского окна Легенда . Это значение может быть использовано для вызова операторов Legend. Возвращает 0 для рамок без легенды.

Параметр Attribute	ID	Возвращаемое значение LayoutItemInfo()
LAYOUT_ITEM_INFO_TYPE	11	Короткое целое (SmallInt): возвращает число описывающее тип рамки. Смотрите таблицу ниже.
LAYOUT_ITEM_INFO_IMAGE_FILE	12	Строковое значение – String: для рамок с изображением. Возвращает полный адрес файла с изображением в рамке. Если рамка с изображением отображается с ошибками, возвращает адрес файла, который должен быть показан в рамке. Возвращает пустую строку для рамок без изображений.
LAYOUT_ITEM_INFO_OBJ	13	Объектная величина: возвращает объект в рамке с текстом или фигурой. Для рамок с текстом возвращает объект. Для рамок с фигурами возвращает линии, прямоугольники, скруглённые прямоугольники или эллипсы. Для рамок без объектов возвращает неинициализированный объект.
LAYOUT_ITEM_INFO_NAME	14	Строковое значение – String: возвращает имя рамки отчёта.
LAYOUT_ITEM_INFO_PEN	15	Величина типа Pen (стиль линии). Возвращает тип линии рамки.
LAYOUT_ITEM_INFO_FILL_BRUSH	16	Величина типа Brush (стиль штриховки). Возвращает тип штриховки рамки. Не может быть применено к рамкам, заштрихованным линиями.
LAYOUT_ITEM_INFO_Z_ORDER	17	Целое число типа Integer: Возвращает параметр z-order объекта. Параметр z-order берётся из нормализованного набора свойств рамки, каждый элемент которого является уникальным.
LAYOUT_ITEM_INFO_ANGLE	18	Вещественное значение, которое возвращает либо поворот рамок с изображением или текстом, либо поворот символа TrueType для рамок с символами. Значение представляет угол в градусах. Если рамки не поддерживают поворот, то возвращается 0.0.
LAYOUT_ITEM_INFO_ADDIN_FILE_NAME	19	Строка, величина типа String: имя MBX-файла заказной дополнительной рамки.
LAYOUT_ITEM_INFO_SERIALIZED_NAME	20	Строка, величина типа String: имя файла нестандартной рамки.

Параметр Attribute	ID	Возвращаемое значение LayoutItemInfo()
LAYOUT_ITEM_INFO_XPS_VIEW_NAME	21	Строка, величина типа String: имя файла нестандартной XPS-рамки.
LAYOUT_ITEM_INFO_NUM_ITEMS_ON_PAGE	22	Целое число типа Integer: количество рамок на этой странице отчёта. Идентификатор рамки игнорируется.
LAYOUT_ITEM_INFO_FRAME_UNIQUE_ID	23	Строка, величина типа String: Уникальный идентификатор рамки отчёта.
LAYOUT_ITEM_INFO_TEXTSTRING	24	Строка, величина типа String: является корректной только для рамок с текстом. Возвращает неформатированную строку, которая содержится в элементе управления рамки с текстом. Это та же самая строка, которая возвращается с помощью атрибута ObjectInfo OBJ_INFO_TEXTSTRING (3) о текстовом объекте LAYOUT_ITEM_INFO_OBJ. Информация о форматировании, например, "All Caps" и "Expanded" не входит в состав возвращаемой строки.
LAYOUT_ITEM_INFO_SMART_TEXTSTRING	25	Строка, величина типа String: является корректной только для рамок с текстом. Возвращает неформатированную строку, которая отображается в элементе управления рамки с текстом; если текст содержит выражения Smart Text, в строке возвращается обработанный результат выполнения выражения. Если при обработке любого динамического текстового выражения Smart Text возникает ошибка, то возвращается сообщение об ошибке. Информация о форматировании, например, "All Caps" и "Expanded" не входит в состав возвращаемой строки.

Далее перечислены типы рамок, возвращаемые LAYOUT_ITEM_INFO_TYPE.

Тип рамки	ID	Описание рамки
LAYOUT_ITEM_TYPE_ANY	-1	Первая рамка с заданным именем независимо от того, какого типа рамка.
LAYOUT_ITEM_TYPE_EMPTY	0	Пустая рамка: рамка без содержимого.
LAYOUT_ITEM_TYPE_MAPPER	1	Рамка с картой.

Тип рамки	ID	Описание рамки
LAYOUT_ITEM_TYPE_BROWSER	2	Рамка со списком.
LAYOUT_ITEM_TYPE_LEGEND	3	Рамка с легендой.
LAYOUT_ITEM_TYPE_TEXT	4	Рамка с текстом.
LAYOUT_ITEM_TYPE_SHAPE	5	Рамка с фигурами, такими как линии, прямоугольники, скруглённые прямоугольники или эллипсы.
LAYOUT_ITEM_TYPE_IMAGE	6	Рамка с изображением.
LAYOUT_ITEM_TYPE_CUSTOM	7	Нестандартная рамка.

По умолчанию "бумажные" единицы измерения — дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

Используйте функцию **LayoutPageItemInfo()**, чтобы получить информацию о любой странице окна **Отчёт**.

См. также:

функция FrontWindow(), функция LayoutInfo(), функция LayoutItemID(), функция LayoutPageItemInfo(), оператор Set Paper Units, функция WindowID()

Функция LayoutItemMetadataValue()

Назначение

Эта функция возвращает значение, соответствующее ключу, хранящемуся в метаданных, которые ассоциированы с рамкой внутри окна **Отчёт**. Эта функция не применима к классическому, традиционному окну **Отчёт**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LayoutItemMetadataValue ( window_id, frame_id, key)
```

window_id — идентификатор окна.

frame_id — число, указывающее какую рамку в окне **Отчёт** требуется опросить. Разделы пронумерованы от 1 до *n*, где *n* это номер в отчёте.

Key — строка, представляющая ключ, по которому выполняется поиск значения метаданных рамки отчёта.

Возвращаемая величина

Строка

Эта функция возвращает значение метаданных, передавая ключ рамке в окне **Отчёт**. Если значение, соответствующее ключу, не найдено, то возвращается пустая строка — "".

Возвращённое значение является значением соответствующим ключу, хранящемуся в метаданных определённой рамки.

Описание

Параметр *window_id* задает идентификатор окна карты. В многостраничном отчёте эта функция осуществляет поиск только среди выбранных элементов на текущей странице.

См. также:

[оператор Create Frame](#), [оператор Alter Designer Frame](#), [функция SaveLayoutItemCustomFrameFiles\(\)](#)

Функция LayoutPageItemInfo()

Назначение

Возвращает информацию о рамке на заданной странице окна **Отчёт**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Эта функция очень похожа на функцию [LayoutItemInfo\(\)](#), которую можно использовать, если страница, которую требуется запросить, является видимой в окне **Отчёт**.

Синтаксис

```
LayoutPageItemInfo( window_id, page_num, frame_id, attribute )
```

window_id — идентификатор окна **Отчёт**, целое число;

Page_num — номер страницы, на которой требуется выполнить запрос идентификатора рамки. Используйте номера в диапазоне от 1 до *n*, где *n* это номер страницы отчёта.

frame_id — число, указывающее какую рамку в окне **Отчёт** требуется опросить. Разделы пронумерованы от 1 до *n*, где *n* это номер в Легенде.

attribute — это целочисленный код, указывающий, какой тип информации возвращается. Значения его представлено в таблице ниже.

Возвращаемая величина

Зависит от параметра *attribute*, смотрите далее в описании.

Описание

Функция `LayoutPageltemInfo( )` возвращает информацию об окне **Отчёт**. Эта функция не применима к окну Районирования.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию `FrontWindow( )`**, которая вызывается сразу после открытия окна, а также **функцию `WindowID( )`**, которую можно вызывать в любой момент после создания окна.

Функция `LayoutPageltemInfo( )` может возвращать различные значения параметров заданного окна **Отчёт**. Параметр *attribute* сообщает функции `LayoutPageltemInfo( )`, какие данные об окне **Отчёт** возвращаются. Параметр *attribute* должен принимать одно из следующих значений в таблице; коды определены в `MAPBASIC.DEF`.

параметр attribute	ID	Возвращаемое значение <code>LayoutPageltemInfo()</code>
LAYOUT_ITEM_INFO_POS_X	1	Вещественное значение – Float: возвращает положение элемента в виде расстояния в "бумажных" единицах от левого края отчёта.
LAYOUT_ITEM_INFO_POS_Y	2	Вещественное значение – Float: возвращает положение элемента в виде расстояния в "бумажных" единицах от верхнего края отчёта.
LAYOUT_ITEM_INFO_WIDTH	3	Вещественное значение – Float: возвращает ширину элемента отчёта в "бумажных" единицах.
LAYOUT_ITEM_INFO_HEIGHT	4	Вещественное значение – Float: возвращает высоту элемента отчёта в "бумажных" единицах.
LAYOUT_ITEM_INFO_WIN	5	Целое число типа Integer: возвращает идентификатор окна элемента отчёта (рамки). Либо возвращает 0, если ассоциированного окна не существует.
LAYOUT_ITEM_INFO_SELECTED	6	Логическая величина: возвращает True, если элемент выбран.

параметр attribute	ID	Возвращаемое значение LayoutPageItemInfo()
LAYOUT_ITEM_INFO_ACTIVATED	7	Логическая величина: возвращает True, если элемент активирован (например, карта активирована после нажатия клавиши Alt и щелчка по рамке).
LAYOUT_ITEM_INFO_EMPTY	8	Логическая величина: возвращает True, если элемент является пустой рамкой, иначе — false.
LAYOUT_ITEM_INFO_LEGEND_FRAME_ID	9	Целое число типа Integer: для рамок с легендой. Возвращает целое — идентификатор рамки с легендой в родительском окне Отчёт . Это значение может быть использовано для вызова операторов Legend. Возвращает 0 для рамок без легенды.
LAYOUT_ITEM_INFO_LEGEND_DESIGNER_WINDOW	10	Целое число типа Integer: для рамок с легендой. Возвращает целое — идентификатор окна родительского окна Легенда . Это значение может быть использовано для вызова операторов Legend. Возвращает 0 для рамок без легенды.
LAYOUT_ITEM_INFO_TYPE	11	Короткое целое (SmallInt): возвращает число описывающее тип рамки. Смотрите таблицу ниже.
LAYOUT_ITEM_INFO_IMAGE_FILE	12	Строковое значение – String: для рамок с изображением. Возвращает полный адрес файла с изображением в рамке. Если рамка с изображением отображается с ошибками, возвращает адрес файла, который должен быть показан в рамке. Возвращает пустую строку для рамок без изображений.
LAYOUT_ITEM_INFO_OBJ	13	Объектная величина: возвращает объект в рамке с текстом или фигурой. Для рамок с текстом возвращает объект. Для рамок с фигурами возвращает линии, прямоугольники, скруглённые прямоугольники или эллипсы. Для рамок без объектов возвращает неинициализированный объект.
LAYOUT_ITEM_INFO_NAME	14	Строковое значение – String: возвращает имя рамки отчёта.
LAYOUT_ITEM_INFO_PEN	15	Величина типа Pen (стиль линии). Возвращает тип линии рамки.

параметр attribute	ID	Возвращаемое значение LayoutPageItemInfo()
LAYOUT_ITEM_INFO_FILL_BRUSH	16	Величина типа Brush (стиль штриховки). Возвращает тип штриховки рамки. Не может быть применено к рамкам, заштрихованным линиями.
LAYOUT_ITEM_INFO_Z_ORDER	17	Целое – Integer. Возвращает параметр z-order объекта. Параметр z-order берётся из нормализованного набора свойств рамки, каждый элемент которого является уникальным.
LAYOUT_ITEM_INFO_ANGLE	18	Вещественное значение, которое возвращает либо поворот рамок с изображением или текстом, либо поворот символа TrueType для рамок с символами. Значение представляет угол в градусах. Если рамки не поддерживают поворот, то возвращается 0.0.
LAYOUT_ITEM_INFO_ADDIN_FILE_NAME	19	Строка, величина типа String: имя MBX-файла заказной дополнительной рамки.
LAYOUT_ITEM_INFO_SERIALIZED_NAME	20	Строка, величина типа String: имя файла заказной серийной рамки.
LAYOUT_ITEM_INFO_XPS_VIEW_NAME	21	Строка, величина типа String: имя файла заказной XPS-рамки.
LAYOUT_ITEM_INFO_NUM_ITEMS_ON_PAGE	22	Целое число типа Integer: количество рамок на этой странице отчёта. Идентификатор рамки игнорируется.

Далее перечислены типы рамок, возвращаемые LAYOUT_ITEM_INFO_TYPE.

Тип рамки	ID	Описание рамки
LAYOUT_ITEM_TYPE_ANY	-1	Первая рамка с заданным именем независимо от того, какого типа рамка.
LAYOUT_ITEM_TYPE_EMPTY	0	Пустая рамка: рамка без содержимого.
LAYOUT_ITEM_TYPE_MAPPER	1	Рамка с картой.
LAYOUT_ITEM_TYPE_BROWSER	2	Рамка со списком.

Тип рамки	ID	Описание рамки
LAYOUT_ITEM_TYPE_LEGEND	3	Рамка с легендой.
LAYOUT_ITEM_TYPE_TEXT	4	Рамка с текстом.
LAYOUT_ITEM_TYPE_SHAPE	5	Рамка с фигурами такими, как линии, прямоугольники, скруглённые прямоугольники или эллипсы.
LAYOUT_ITEM_TYPE_IMAGE	6	Рамка с изображением.
LAYOUT_ITEM_TYPE_CUSTOM	7	Заказная рамка.

По умолчанию "бумажные" единицы измерения — дюймы. Чтобы задать другие "бумажные" единицы измерения, используйте **оператор Set Paper Units**.

См. также:

функция FrontWindow(), функция LayoutInfo(), функция LayoutItemInfo(), оператор Set Paper Units, функция WindowID()

Оператор Layout

Назначение

Открывает новое окно отчёта. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Layout [ Designer ]
[ Position ( x, y ) [ Units paperunits ] ]
[ Width window_width [ Units paperunits ] ]
[ Height window_height [ Units paperunits ] ]
[ { Min | Max | Floating | Docked | Tabbed | AutoHidden } ]
```

x, y — координаты левого верхнего угла окна Отчета в определенных единицах измерения; "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**.

paper_units — строка, представляющая "бумажные" единицы измерения. (например, "cm" для сантиметров).

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки

- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

win_width — ширина окна.

win_height — высота окна.

Все состояния прикрепления окон относятся только к 64-разрядной версии MapInfo Pro:

- *Floating* — соответствует плавающему окну.
- *Docked* — соответствует прикреплённому в стандартном положении окну.
- *Tabbed* — соответствует окнам, оформленным в виде вкладок, которые также, иногда, вызываться как документы.
- *AutoHidden* — соответствует автоматически спрятанному окну.

Описание

Оператор предназначен для обеспечения обратной совместимости со старыми 32-разрядными версиями MapInfo Pro. Если опустить предложение **Designer**, то будет открыто традиционное окно **Отчёт**, которое можно найти только в 32-разрядных версиях MapInfo Pro. При работе в современной версии MapInfo Pro (64разрядная версии), включите предложение **Designer**, для того чтобы открыть окно **Отчёт**, которое обеспечивает больше функциональности. Такое окно обычно называется окном **Конструктор отчёта**. MapInfo Pro работает с отрицательными координатами и адресами расположения и умеет преобразовывать их из старых отчётов в окна **Конструктор отчёта**.

Если задано ключевое слово **Min**, окно отчёта показывается минимизированным. Если задано ключевое слово **Max**, окно отчёта заполняет все рабочее пространство MapInfo Pro.

Предложения **Width** и **Height** определяют ширину и высоту окна отчёта. Но эти характеристики не влияют на размеры страницы макета.

Для назначения размеров страниц и их количества используйте **оператор Set Layout**.

Когда приходится работать с классическими окнами **Отчёт** и картами, созданными с помощью более старой версии чем 12.5, то MapInfo Pro присваивает специальное скрытое имя таблицы каждому классическому окну **Отчёт**, но не современным окнам **Отчёт**. Первому открытому окну отчёта дается имя "Отчёт1", следующему "Отчёт2" и так далее. О фоне классических окон **Отчёт** смотрите раздел: **Работа с классическим окном отчёта**.

Программа, написанная на MapBasic, может создавать, выбирать или изменять объекты в окне отчёта, обращаясь к нему по этому имени. Например, следующий оператор выбирает все объекты в окне **Отчёт**:

```
Select * From Layout1
```


Пример

В следующем примере создается окно **Отчёт** в два дюйма шириной и четыре высотой. Верхний левый угол окна располагается в верхнем левом углу рабочей области MapInfo.

```
Layout Designer Position (0, 0) Width 2 Height 4
```

См. также:

[оператор Open Window](#), [оператор Set Layout](#)

Функция Lcase\$()

Назначение

Возвращает строку, преобразуя все прописные буквы в строчные. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LCase$( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **LCase\$()** возвращает строку, полученную из строки, представленной выражением *string_expr*, преобразованием всех заглавных букв в строчные.

Преобразованию подвергаются только буквы: латинские – от A до Z, и русские – от А до Я. Цифры и другие текстовые символы не преобразуются. Например:

```
LCase$( "A#12a" )
```

возвращает строку "a#12a".

Пример

```
Dim regular, lower_case As String
regular = "Los Angeles"
lower_case = LCase$(regular)
'
' Now, lower_case contains the value "los angeles"
'
```

См. также:

функция **Proper\$()**, функция **UCase\$()**

Функция Left\$()

Назначение

Возвращает левую часть строки, выделяя определенное количество символов из исходной. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Left$( string_expr, num_expr )
```

string_expr – строковое выражение.

num_expr – численное выражение, результат которого ноль или более.

Возвращаемая величина

Строка – String

Описание

Функция **Left\$()** возвращает строковую величину, полученную из самых левых *num_expr* символов строки *string_expr*.

В результате вычисления выражения *num_expr* должно получаться целое положительное число, не равное нулю. Если величина *num_expr* дробная, MapBasic округляет ее до ближайшего целого. Если *num_expr* равно нулю, **Left\$()** возвращает пустую строку. Если численный параметр *num_expr* больше, чем число символов в строке *string_expr*, результат функции **Left\$()** будет равен строке, представленной выражением *string_expr*.

Пример

```
Dim whole, partial As String
whole = "Afghanistan"
partial = Left$(whole, 6)

' at this point, partial contains the string: "Afghan"
```

См. также:

функция [Mid\(\)](#), функция [Right\(\)](#)

Функция LegendFrameInfo()

Назначение

Возвращает информацию о разделе в легенде. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LegendFrameInfo( window_id, frame_id, attribute )
```

window_id – число, указывающее какое окно легенды требуется опросить;

frame_id – число, указывающее какой раздел в окне легенды требуется опросить. Разделы пронумерованы от 1 до *n*, где *n* это номер в легенде.

attribute – это целочисленный код, указывающий, какой тип информации возвращается. Значения его представлено в таблице ниже.

Возвращаемая величина

Зависит от параметра *attribute*, смотрите далее в описании.

Описание

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Функция LegendFrameInfo() может возвращать различные значения параметров заданного окна **Отчёт**. Параметр *attribute* сообщает функции LegendFrameInfo(), какие данные об окне

Отчёт возвращаются. Параметр `attribute` должен принимать одно из следующих значений в таблице; коды определены в `MAPBASIC.DEF`.

Параметр <code>attribute</code>	ID	Возвращаемое значение <code>LegendFrameInfo()</code>
<code>FRAME_INFO_TYPE</code>	1	Возвращает одно из следующих констант, определяющих тип раздела: • <code>FRAME_TYPE_NONE</code> (0) • <code>FRAME_TYPE_STYLE</code> (1) • <code>FRAME_TYPE_THEME</code> (2) • <code>FRAME_TYPE_RASTER</code> (3)
<code>FRAME_INFO_MAP_LAYER_ID</code>	2	Возвращает ID-номер слоя, с которым соотносится раздел.
<code>FRAME_INFO_REFRESHABLE</code>	3	Возвращает <code>TRUE</code> , если раздел был создан без ключевого слова Norefresh . Всегда возвращает <code>TRUE</code> для тематических разделов.
<code>FRAME_INFO_POS_X</code>	4	Возвращает расстояние от верхнего левого угла раздела до левого края окна легенды (в "бумажных" единицах).
<code>FRAME_INFO_POS_Y</code>	5	Возвращает расстояние от верхнего левого угла раздела до верхнего края канвы легенды (в "бумажных" единицах).
<code>FRAME_INFO_WIDTH</code>	6	Возвращает ширину раздела (в "бумажных" единицах). "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
<code>FRAME_INFO_HEIGHT</code>	7	Возвращает высоту раздела (в "бумажных" единицах).
<code>FRAME_INFO_TITLE</code>	8	Возвращает заголовок раздела или тематического раздела.
<code>FRAME_INFO_TITLE_FONT</code>	9	Возвращает шрифт заголовка раздела легенды. Если раздел не имеет заголовка, возвращает шрифт заголовка, используемый по умолчанию.
<code>FRAME_INFO_SUBTITLE</code>	10	Возвращает подзаголовок раздела.
<code>FRAME_INFO_SUBTITLE_FONT</code>	11	То же, что и <code>FRAME_INFO_TITLE_FONT</code> (9)

Параметр attribute	ID	Возвращаемое значение LegendFrameInfo()
FRAME_INFO_BORDER_PEN	12	Возвращает параметры линии, использованной для рамки окон картографической или тематической легенды. Не поддерживается для окон Легенда и возвращают невидимый стиль линии: Pen (0, 1, 0).
FRAME_INFO_NUM_STYLES	13	Возвращает количество стилей в разделе.
FRAME_INFO_VISIBLE	14	Возвращает TRUE, если раздел видимый (тематические разделы могут быть невидимыми).
FRAME_INFO_COLUMN	15	Возвращает атрибуты имени колонки для легенды в виде строки. Возвращает пустую строку, если раздел тематический.
FRAME_INFO_LABEL	16	Возвращает выражение подписи в виде строки. Возвращает пустую строку, если раздел тематический.
FRAME_INFO_COLUMNS	17	Возвращает количество колонок в разделе легенды. Возвращает -1 для окна Картографическая легенда .
FRAME_INFO_NUM_VISIBLE_ROWS	18	Возвращает количество видимых в разделе легенды рядов условных знаков. Для окон Картографических легенд (используемых в более ранних чем 11.5 версиях) возвращает -1.
FRAME_INFO_LINE_SAMPLE_WIDTH	19	Возвращает ширину образцовой линии в "бумажных" единицах. Возвращает -1 для окна Картографическая легенда или раздела растровой легенды с полоской цветов. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
FRAME_INFO_REGION_SAMPLE_WIDTH	20	Возвращает ширину образца площадного условного знака в "бумажных" единицах. Возвращает -1 для окна Картографическая легенда или раздела растровой легенды с полоской цветов. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .

Параметр attribute	ID	Возвращаемое значение LegendFrameInfo()
FRAME_INFO_REGION_SAMPLE_HEIGHT	21	Возвращает высоту образца в "бумажных" единицах. Возвращает -1 для окна Картографическая легенда или раздела растровой легенды с полоской цветов. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
FRAME_INFO_AUTO_FONT_SIZE	22	Возвращает размер используемого шрифта. Возвращает -1 для раздела растровой легенды с полоской цветов.
FRAME_INFO_SHOW_COUNTS	23	Логическое – Logical. Возвращает состояние требуется ли отображать рядом с записями счётчик количества цветов в легенде. Это имеет смысл только для разделов тематических, классифицированных растровых легенд.
FRAME_INFO_DISPLAY_TYPE	24	DT_INTEGER. Возвращает код, который описывает, как отображаются строки легенды (условные знаки и текст) в разделе легенды. Это имеет смысл только для растровых легенд, но для всех разделов будет возвращаться одно из следующих значений: • FRAME_DISPLAY_TYPE_NOT_SPECIFIED (0) – все векторные и тематические разделы легенды. • FRAME_DISPLAY_TYPE_GAP (1) – тематические, классифицированные легенды или растровые легенды, созданные с помощью MapBasic-оператора Display Gap. • FRAME_DISPLAY_TYPE_PALETTE (2) – растровые легенды диапазонов или растровые легенды, созданные с помощью MapBasic-оператора Display Palette. • FRAME_DISPLAY_TYPE_BAR (3) – растровые непрерывные легенды или растровые легенды, созданные с помощью MapBasic-оператора Display Bar.
FRAME_INFO_BAR_THICKNESS	25	DT_FLOAT. Возвращает толщину полоски цветов легенды непрерывного раstra (FRAME_DISPLAY_TYPE_BAR) в текущих "бумажных" единицах измерения, принятых в MapBasic. Возвращает -1 для любых других типов раздела легенды.

Параметр attribute	ID	Возвращаемое значение LegendFrameInfo()
FRAME_INFO_BAR_LENGTH	26	DT_FLOAT. Возвращает длину полосы цветов легенды непрерывного растра (FRAME_DISPLAY_TYPE_BAR) в текущих "бумажных" единицах измерения, принятых в MapBasic. Возвращает -1 для любых других типов раздела легенды.
FRAME_INFO_BAR_IS_HORIZONTAL	27	DT_LOGICAL: возвращает состояние, является ли полоска цветов горизонтальной. Возвращает "истину – true" для горизонтальной полосы цветов. Возвращает "ложь – false" для вертикальной полосы цветов или любых других типов разделов оегенды.

См. также:

[функция FrontWindow\(\)](#), [функция LegendInfo\(\)](#), [функция LegendTextFrameInfo\(\)](#), [функция LegendStyleInfo\(\)](#), [функция WindowID\(\)](#)

Функция LegendInfo()

Назначение

Возвращает информацию о легенде, например, сведения об ориентации, количестве разделов легенды и о размере образца условного знака в легенде — большой или маленький. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LegendInfo( window_id, attribute )
```

window_id — число, указывающее какое окно легенды требуется опросить;

attribute — это целочисленный код, указывающий, какой тип информации возвращается. Значения его представлено в таблице ниже.

Возвращаемая величина

Зависит от параметра *attribute*, смотрите далее в описании.

Описание

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Функция LegendInfo() может возвращать различные значения параметров заданного окна **Отчёт**. Параметр attribute сообщает функции LegendInfo(), какие данные об окне **Отчёт** возвращаются. Параметр attribute должен принимать одно из следующих значений в таблице; коды определены в MAPBASIC.DEF.

Код атрибута	ID	Возвращаемое значение LegendInfo()
LEGEND_INFO_MAP_ID	1	Целое число типа Integer: возвращает идентификатор родительского окна карты (для его получения так же используется функция WindowInfo() с кодом WIN_INFO_TABLE).
LEGEND_INFO_ORIENTATION	2	Целое число типа Integer: Возвращает предопределенное значение, характеризующее ориентацию легенды: • ORIENTATION_PORTRAIT (1) • ORIENTATION_LANDSCAPE (2) • ORIENTATION_CUSTOM (3)
LEGEND_INFO_NUM_FRAMES	3	Целое число типа Integer: Возвращает число разделов в легенде
LEGEND_INFO_STYLE_SAMPLE_SIZE	4	Целое число типа Integer: Возвращает 0 для малых образцов символов легенды и 1 для больших. Возвращает -1 для окна Легенда .
LEGEND_INFO_LINE_SAMPLE_WIDTH	5	Целое число типа Integer: Возвращает ширину образцовой линии в "бумажных" единицах. Возвращает -1 для окна картографической легенды. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
LEGEND_INFO_REGION_SAMPLE_WIDTH	6	Целое число типа Integer: Возвращает ширину образца площадного условного знака в "бумажных" единицах. Возвращает -1 для окна картографической легенды.

Код атрибута	ID	Возвращаемое значение LegendInfo()
LEGEND_INFO_REGION_SAMPLE_HEIGHT	7	Целое число типа Integer: Возвращает высоту образца в "бумажных" единицах. Возвращает -1 для окна картографической легенды.
LEGEND_INFO_NUM_TEXTFRAMES	8	Целое число типа Integer: Возвращает количество текстовых разделов в текущем окне Легенда
LEGEND_INFO_DEFAULT_TITLE	9	Строка, величина типа String: возвращает стандартный заголовок легенды.
LEGEND_INFO_DEFAULT_TITLE_FONT	10	Строка, величина типа String: возвращает шрифт стандартного заголовка легенды.
LEGEND_INFO_DEFAULT_SUBTITLE	11	Строка, величина типа String: возвращает стандартный подзаголовок легенды.
LEGEND_INFO_DEFAULT_SUBTITLE_FONT	12	Строка, величина типа String: возвращает шрифт стандартного подзаголовка легенды.
LEGEND_INFO_DEFAULT_STYLE	13	Строка, величина типа String: возвращает имя стандартного стиля легенды.
LEGEND_INFO_DEFAULT_STYLE_FONT	14	Строка, величина типа String: возвращает имя шрифта стандартного стиля легенды.
LEGEND_INFO_DEFAULT_AUTO_FNT_SZ	15	Логическое: Возвращается стандартное значение, применённое при выполнении условия "Увеличивать условные знаки до размера шрифта".

Пример

```

dim wndLegend, wndMap as integer
for i = 1 to NumWindows()
  If WindowInfo(WindowID(i), WIN_INFO_TYPE) = WIN_MAPPER then
    wndMap = WindowInfo(WindowID(i), WIN_INFO_WINDOWID)
  end if
  if WindowInfo(WindowID(i), WIN_INFO_TYPE) = WIN_LEGEND_DESIGNER then
    wndLegend = WindowInfo(WindowID(i), WIN_INFO_WINDOWID)
  end if
next

```

В следующем примере показывается, как узнать, какой ID-номер карты был использован для создания окна **Легенда**.

```
If LegendInfo(wndLegend, LEGEND_INFO_MAP_ID) = wndMap then
  Print "Map ID: " + str$(MapID) + " was correct"
End if
```

См. также:

[функция FrontWindow\(\)](#), [функция LegendFrameInfo\(\)](#), [функция LegendTextFrameInfo\(\)](#), [функция LegendStyleInfo\(\)](#), [функция WindowID\(\)](#)

Функция LegendStyleInfo()

Назначение

Возвращает информацию о стиле, используемом в разделе легенды. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LegendStyleInfo( window_id, frame_id, style_id, attribute )
```

window_id – число, указывающее какое окно легенды требуется опросить;

frame_id – число, указывающее какой раздел в окне легенды требуется опросить. Разделы пронумерованы от 1 до *n*, где *n* это номер в легенде.

style_id – это число, определяющее, какой стиль внутри раздела Вы хотите опросить; Стили пронумерованы от 1 до *n*, где *n* это число стилей в разделе.

attribute – это целочисленный код, указывающий, какой тип информации возвращается.

Возвращаемая величина

Код атрибута	ID	Возвращаемое значение LegendStyleInfo()
LEGEND_STYLE_INFO_TEXT	1	Возвращает текст стиля.
LEGEND_STYLE_INFO_FONT	2	Возвращает шрифт стиля.

Код атрибута	ID	Возвращаемое значение LegendStyleInfo()
LEGEND_STYLE_INFO_OBJ	3	Возвращает объект стиля. Допустимые значения для разных типов тематических разделов: • Тематическая картодиаграмма диапазонов – прямоугольник, линия или точка • Столбчатая картодиаграмма – прямоугольник • Тематическая поверхность – прямоугольник • Тематическая картодиаграмма значков – точка • Картодиаграмма плотности точек – прямоугольник • Круговая картодиаграмма – прямоугольник
LEGEND_STYLE_INFO_ROW_VISIBLE	4	Возвращает информацию о видимости стиля оформления ряда легенды. Для окон картографических легенд (используемых в более ранних чем 11.5 версиях) возвращает true.
LEGEND_STYLE_INFO_COUNT	5	DT_LARGEINT. Возвращает значение счётчика строк этой легенды. Возвращает -1, если счётчик не может быть применён к этому разделу или строке легенды. Также возвращает -1, если счётчик обращается к растровой легенде.

Ошибки

Генерируется сообщение об ошибке, если раздел не имеет стилей (тематический раздел).

Пример

В следующем примере показано, как вызывать функцию **LegendStyleInfo()**, для того чтобы выяснить информацию о стиле оформлении только тематических разделов окна **Легенда**. В примере подразумевается, что окно **Легенда** является самым верхним. В нем возвращается информация о стиле оформления первого элемента описания первого раздела легенды о тематическом слое диапазонов.

```

dim objStyle as object
dim tBrush as Brush
dim tPen as Pen
dim tSymbol as Symbol
dim tFont as Font
dim strSampleText as string

objStyle = LegendStyleInfo(FrontWindow(), 1, 1, LEGEND_STYLE_INFO_OBJ)
tFont = LegendStyleInfo(FrontWindow(), 1, 1, LEGEND_STYLE_INFO_FONT)
strSampleText = LegendStyleInfo(FrontWindow(), 1, 1,
LEGEND_STYLE_INFO_TEXT)

```

```

Do Case ObjectInfo(objStyle, OBJ_INFO_TYPE)
  Case OBJ_TYPE_POINT
    tSymbol = ObjectInfo(objStyle, OBJ_INFO_SYMBOL)
    Print tSymbol
    'or use StyleAttr() to return specific properties of the Symbol Style,
    for example: StyleAttr(tSymbol, SYMBOL_FONT_NAME)
  Case OBJ_TYPE_LINE
    tPen = ObjectInfo(objStyle, OBJ_INFO_PEN)
    Print tPen
    'or use StyleAttr() to return specific properties of the Pen\Line Style,
    for example: StyleAttr(tPen, PEN_COLOR)
  Case OBJ_TYPE_RECT
    tBrush = ObjectInfo(objStyle, OBJ_INFO_BRUSH)
    Print tBrush
    'or use StyleAttr() to return specific properties of the Brush\Region
    Style, for example: StyleAttr(tBrush, BRUSH_FORECOLOR)
  Case Else
    Note "Unexpected Object type"
End Case

Print tFont 'for example: Font ("Arial",0,8,0,0)

```

Используйте **функцию StyleAttr()**, чтобы получить свойства шрифта, например:

```
StyleAttr(tFont, FONT_NAME)
```

и напечатайте текст первого тематического значения (например, от 5 700 000 до 23 700 000).

```
Print strSampleText
```

См. также:

функция StyleAttr()

Функция LegendTextFrameInfo()

Назначение

Возвращает информацию о текстовом разделе окна **Легенда**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LegendTextFrameInfo( window_id, frame_id, attribute )
```

window_id – число, указывающее какое окно легенды требуется опросить;

frame_id – число, указывающее какой раздел в окне легенды требуется опросить. Разделы пронумерованы от 1 до *n*, где *n* это номер в легенде.

attribute – это целочисленный код, указывающий, какой тип информации возвращается.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

параметр attribute	ID	Возвращаемое значение LegendTextFrameInfo()
FRAME_INFO_POS_X	1	Возвращает расстояние от верхнего левого угла раздела до левого края окна легенды (в "бумажных" единицах).
FRAME_INFO_POS_Y	2	Возвращает расстояние от верхнего левого угла раздела до верхнего края канвы легенды (в "бумажных" единицах).
FRAME_INFO_WIDTH	3	Возвращает ширину раздела (в "бумажных" единицах). "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
FRAME_INFO_HEIGHT	4	Возвращает высоту раздела (в "бумажных" единицах). "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
FRAME_INFO_TEXT	5	Возвращает текст из текстового раздела.
FRAME_INFO_TEXT_FONT	6	Возвращает шрифт текстового раздела.

См. также:

функция LegendInfo(), функция LegendFrameInfo(), функция LegendStyleInfo()

Функция Len()

Назначение

Возвращает количество символов в строке или число байтов в переменной. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Примечание: Для строковой переменной функция **Len()** возвращает только количество символов в строке, а не количество байт. Для того чтобы узнать количество байт в строке, можно пользоваться функцией **StringByteLength()**.

Например, `Len("ABCD属性コ構")` возвращает 8 — количество символов в строке, но `StringByteLength("ABCD属性コ構", WindowsJapanese)` возвращает 12 — количество байт в строке в кодировке WindowsJapanese.

Синтаксис

```
Len ( expr )
```

expr — выражение, результатом которого является число, но не величина типа Pen, Brush, Symbol, Font или Alias.

Возвращаемая величина:

Короткое целое – SmallInt

Описание

Информация, возвращаемая функцией **Len()**, зависит от типа результата выполнения *expr*.

Если выражение *expr* является строкой, то функция **Len()** вернет количество символов в строке.

Если *expr* — переменная MapBasic, то результат функции **Len()** будет размером переменной в байтах. Так, если тип переменной был объявлен как Integer (целое число), то функция **Len()** т. к. переменной типа Integer отводится 4 байта. Для переменной типа SmallInt (короткое целое число), результатом функции **Len()** т. к. переменной типа SmallInt отводится 2 байта.

Пример

```
Dim name_length As SmallInt
name_length = Len("Boswell")
```

теперь значение *name_length* = 7

См. также:

- [функция StringByteLength\(\)](#)
- [Функция ObjectLen\(\)](#)

Функция LibraryServiceInfo()

Назначение

Возвращает информацию о службах библиотек, например, состояние режима обслуживания, номер версии, стандартный URL-адрес библиотечной службы. Также выводит список CSW URL-адресов, который выдает сервер MapInfo Manager.

Синтаксис

```
LibraryServiceInfo( attribute )
```

attribute – целочисленный код, см. таблицу ниже.

Описание

Функция **LibraryServiceInfo()** возвращает информацию о библиотечной службе (Library Services).

Значение параметра *attribute* показаны в таблице ниже. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код атрибута	ID	LibraryServiceInfo() возвращает
LIBSRVC_INFO_LIBSRVCMODE	1	Целое число, обозначающее текущий режим работы службы Library Service.
LIBSRVC_INFO_LIBVERSION	2	Строка, представляющая версию Library Service. Перед вызовом этой функции, должен быть задан стандартный URL-адрес службы Library Service.
LIBSRVC_INFO_DEFURLPATH	3	Строка, представляющая стандартный URL-адрес службы Library Service. По умолчанию – пустая строка.
LIBSRVC_INFO_LISTCSWURL	4	Строка, разделенная точками с запятой (;), представляющая из себя список CSW URL-адресов, который выдает сервер MapInfo Manager.

Пример

Следующий пример иллюстрирует использование этой функции:

```
include "mapbasic.def"
declare sub main
sub main
dim liburlpath as string
dim libversion as string
liburlpath = LibraryServiceInfo(LIBSRVC_INFO_DEFURLPATH) if
StringCompare(liburlpath, "") == 0 then
Set LibraryServiceInfo URL
"http://localhost:8080/LibraryService/LibraryService"
endif
libversion = LibraryServiceInfo(LIBSRVC_INFO_LIBVERSION)
end sub
```

См. также:

[оператор Set LibraryServiceInfo](#)

Функция Like()

Назначение

Возвращает TRUE или FALSE, сравнивая строку с шаблоном. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Like( string, pattern_string, escape_char )
```

string – строка;

pattern_string – шаблон для сравнения, который является строкой, состоящей из регулярных и специальных символов;

escape_char – строковое выражение, задающее символ отмены проверки следующего за ним символа. В этой строке можно использовать "отменяющий" символ "\" в сочетании с символами, обозначающими один или несколько произвольных символов ("% и "). Если отменяющий символ не назначается, то используется пустая строка ("").

Возвращаемая величина

Логическое значение (TRUE , если строка совпадает с *pattern_string*).

Описание

Так же, как оператор **Like()**, выполняет сравнение строк. В отличие от оператора Like, строчные и прописные буквы различаются.

Строка **pattern_string** может содержать следующие символы, обозначающими один или несколько произвольных символов:

_ (символ подчеркивания)	соответствует одному символу
% (символ процента)	соответствует нескольким символам или не одному

Для явного задания знаков подчеркивания и процента они используются вместе с символом *escape_char*. Примеры приведены в следующей таблице:

Критерий совпадения	Используйте функцию с аргументами
начинается с "South"	<pre>Like(string_var, "South%", "")</pre>
кончается на "America"	<pre>Like(string_var, "%America", "")</pre>
содержит "ing"	<pre>Like(string_var, "%ing%", "")</pre>
начинается с подчеркивания	<pre>Like(string_var, "_%", "\")</pre>

См. также:

[функция Len\(\)](#), [функция StringCompare\(\)](#)

Оператор Line Input

Назначение

Читает строку из текстового файла в переменную.

Синтаксис

```
Line Input [#] filenum, var_name
```

filenum – номер открытого файла, целое число.

var_name – имя переменной строчного типа.

Описание

Оператор **Line Input** читает текущую строку из текстового файла в переменную типа String. Текстовый файл должен быть открыт для последовательного доступа (INPUT).

Оператор **Line Input** читает каждую строку полностью. Если строка содержит список выражений, разделенный запятыми и Вы хотите каждое выражение присвоить отдельной переменной, то используйте **оператор Input #** вместо **Line Input**.

Пример

В нижеприведенном коде читается строка за строкой из первого файла и копируется во второй файл.

```
Dim str As String
Open File "original.txt" For Input As #1
Open File "copy.txt" For Output As #2
Do While Not EOF(1)
    Line Input #1, str
    If Not EOF(1) Then
        Print #2, str
    End If
Loop
Close File #1
Close File #2
```

См. также:

оператор Input #, **оператор Open File**, **оператор Print #**

Функция LocateFile()

Назначение

Возвращает маршрут к файлам данных MapInfo. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LocateFile$( file_id )
```

file_id – см. таблицу ниже.

Значение	ID	Описание
LOCATE_PREF_FILE	0	Файл настроек, сделанных заранее
LOCATE_DEF_WOR	1	Стандартный файл рабочего набора (MAPINFOW.WOR).
LOCATE_CLR_FILE	2	Файл раскраски
LOCATE_PEN_FILE	3	Файл штрихов линий и контуров
LOCATE_FNT_FILE	4	Файл символов условных знаков
LOCATE_ABB_FILE	5	Файл сокращений
LOCATE_PRJ_FILE	6	Файл проекций
LOCATE_MNU_FILE	7	Файл меню
LOCATE_CUSTSYMB_DIR	8	Каталог растровых символов
LOCATE_THMTMPLT_DIR	9	Каталог тематических шаблонов
LOCATE_GRAPH_DIR	10	Каталог с шаблонами графиков
LOCATE_WMS_SERVERLIST	11	XML-список WMS-серверов (MIWMSERVERS.XML).

Значение	ID	Описание
LOCATE_WFS_SERVERLIST	12	XML-список WFS-серверов (MIWFSSERVERS.XML).
LOCATE_GEOCODE_SERVERLIST	13	XML-список серверов геокодирования (MIGEOCODESERVERS.XML).
LOCATE_ROUTING_SERVERLIST	14	XML-список серверов маршрутизации (MIROUTINGSERVERS.XML).
LOCATE_LAYOUT_TEMPLATE_DIR	15	Директория шаблонов отчётов (LAYOUTTEMPLATE)

Возвращаемая величина

Строка – String

Описание

Эта функция, используя ID-номер файла данных MapInfo Pro, возвращает маршрут к нему. Программы установки MapInfo Pro помещают эти файлы в пользовательский каталог Application Data, но есть возможность разместить некоторых из этих файлов в некоторых других каталогах, включая каталог с MAPINFOW.EXE. Программы MapBasic, которые используют эти файлы, должны, однако, придерживаться правила всегда использовать функцию **LocateFile\$()**.

Пример

```
include "mapbasic.def"
declare sub main
sub main
dim sGraphLocations as string
sGraphLocations = LocateFile$(LOCATE_GRAPH_DIR)
Print sGraphLocations
end sub
```

См. также:

функция **GetFolderPath\$()**

Функция LOF()

Назначение

Возвращает длину открытого файла.

Синтаксис

```
LOF( filenum )
```

filenum – номер файла, который был присвоен ему при открытии.

Возвращаемая величина

Значение LargeInt

Описание

Функция **LOF()** возвращает размер открытого файла в байтах.

Параметр **file** должен быть целочисленным номером файла; этот номер определяется в предложении **As**, которое использует оператор Open File.

Ошибки

В результате выполнения функции можно получить код ошибки ERR_FILEMGR_NOTOPEN (366), если файл не был открыт.

Пример

```
Dim size As Integer
Open File "import.txt" For Binary As #1
size = LOF(1)
' size now contains the # of bytes in the file
```

См. также:

оператор Open File

Функции Log()

Назначение

Вычисляет натуральный логарифм. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Log ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Log()** возвращает значение натурального логарифма от числа, полученного в результате вычисления выражения *num_expr*.

Функция натурального логарифма обратна функции экспоненты (число *e* в степени *num_expr*). Число *e* иррационально и приблизительно равно 2.7182818.

Логарифм может вычисляться только от положительного числа; функция **Log()** вернет ошибку, если *num_expr* есть отрицательная величина.

Можно вычислить логарифм и по другому основанию (например, 10), используя натуральный логарифм. Для вычисления логарифма по основанию 10 от числа *n* надо разделить натуральный логарифм от числа *n* (**Log(n)**) на натуральный логарифм от 10 (**Log(10)**).

Пример

```
Dim original_val, log_val As Float
original_val = 2.7182818
log_val = Log(original_val)

' log_val will now have a value of 1 (approximately),
' since E raised to the power of 1 equals
' 2.7182818 (approximately)
```

См. также:

функция **Exp()**

Функция LTrim\$()

Назначение

Удаляет пробелы в начале строки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
LTrim$( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **LTrim\$ ()** возвращает строковую величину, полученную из выражения *string_expr* удалением пробелов в начале строки, если они есть.

Пример

```
Dim name As String  
name = " Mary Smith"  
name = LTrim$(name)  
  
' name now contains the string "Mary Smith"
```

См. также:

[Функция RTrim\\$\(\)](#)

Процедура Main

Назначение

Главная процедура, которая выполняется первой при загрузке прикладной программы.

Синтаксис

```
Declare Sub Main
Sub Main
    statement_list
End Sub
```

statement_list – список операторов, составляющих программу.

Описание

Main – стандартное имя процедуры MapBasic. Если текст программы на MapBasic содержит процедуру **Main**, то выполнение программы начнется с этой процедуры. Процедура **Main** может вызывать другие процедуры (смотрите раздел об [операторе Call](#)).

Можно не объявлять процедуру **Main**. В этом случае первый оператор программы понимается как оператор из процедуры **Main**, если только этот оператор не объявляет другую процедуру. MapBasic начинает выполнять программу, как если бы процедура **Main** была объявлена перед этим оператором. Назовем этот случай "неявным" заданием процедуры **Main** (как противоположность "явному" заданию **Main**).

Пример

Прикладная программа на языке MapBasic может состоять из одной строки. Например, эта программа выполняет только один оператор:

```
Note "Testing, one two three."
```

Можно использовать этот оператор как полноценную программу, при этом MapBasic будет неявно считать ее содержанием процедуры Main. При запуске этой программы MapBasic выполняет [оператор Note](#).

В следующем примере явно задана процедура **Main** результат который точно такой же, т. е. срабатывает [оператор Note](#)).

```
Declare Sub Main
Sub Main
```



```
Note "Testing, one two three."
End Sub
```

Следующая программа также содержит неявно заданную процедуру **Main**. Из этой неявной процедуры **Main** процедура под именем "Talk", для чего используется **оператор Call**.

```
Declare Sub Talk(ByVal msg As String)
Call Talk("Hello")
Call Talk("Goodbye")
Sub Talk(ByVal msg As String)
    Note msg
End Sub
```

Следующий пример содержит явную процедуру **Main**, из которой вызывается подпрограмма "Talk". Для этого **Main** использует оператор **Call**.

```
Declare Sub Main
Declare Sub Talk(ByVal msg As String)

Sub Main
    Call Talk("Hello")
    Call Talk("Goodbye")
End Sub

Sub Talk(ByVal msg As String)
    Note msg
End Sub
```

См. также:

процедура EndHandler, **процедура RemoteMsgHandler**, **процедура SelChangedHandler**, **оператор Sub...End Sub**, **процедура ToolHandler**, **процедура WinClosedHandler**

Функция MakeBrush()

Назначение

Возвращает установку стиля штриха. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakeBrush( pattern, forecolor, backcolor)
```

pattern – тип штриха, целое число от 1 до 8 или от 12 до 186. Рисунки штриха приведены в разделе о **предложении Brush**.

forecolor – цвет штриха в системе RGB. Подробнее см. раздел: [Функция RGB\(\)](#).

backcolor – цвет фона в системе RGB. Чтобы сделать фон прозрачным, задайте значение backcolor равное -1 и значение pattern равное 3 или более.

Возвращаемая величина

Штриховка

Описание

Функция **MakeBrush()** возвращает величину типа Brush, определяющую стиль штриховки графического объекта. Возвращаемая величина может быть присвоена переменной типа Brush или использована как параметр оператора (таких как Create Ellipse, Set Map, Set Style или Shade).

См. более подробную информацию в разделе: [Предложение Brush](#).

Пример

```
Include "mapbasic.def"
Dim b_water As Brush
b_water = MakeBrush(64, CYAN, BLUE)
```

См. также:

[предложение Brush](#), [функция CurrentBrush\(\)](#), [функция RGB\(\)](#), [функция StyleAttr\(\)](#)

Функция MakeCustomSymbol()

Назначение

Возвращает символ, созданный из растрового файла. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakeCustomSymbol( filename, color, size, customstyle )
```

filename – строка до 31 символа длиной с именем растрового файла. Файл должен находиться в подкаталоге CUSTSYMB каталога пользователя MapInfo.

color – цвет в системе RGB; см. также раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

customstyle – целочисленный код типа Integer, управляющий цветом и фоном символа. Смотрите таблицу ниже.

Возвращаемая величина

Символ

Описание

Функция **MakeCustomSymbol()** возвращает величину типа Symbol, основанную на растровом файле. Смотрите в разделе: [Предложение Symbol](#) описание других типов символа.

В следующей таблице перечислены возможности настройки растрового символа:

значение customstyle	Стиль символа
0	Не действуют режимы из группы Эффекты диалога Стиль символа, и символ появляется таким, какой он есть. Все белые пиксели растра прозрачны.
1	Действует режим Добавить фон; все белые пиксели растра непрозрачны.
2	Действует режим Покрасить одним цветом; все небелые точки растра закрашены одним цветом.
3	Установлены оба флажка (действуют оба режима).
4	Действует режим Показать реальный размер; растровый символ визуализируется в свою натуральную величину.
5	Действуют режимы Добавить фон и Показать реальный размер.
7	Действуют режимы: Добавить фон, Покрасить одним цветом и Показать реальный размер.

Пример

```
Include "mapbasic.def"
Dim sym_marker As Symbol
sym_marker = MakeCustomSymbol("CAR1-64MP", BLUE, 18, 0)
```

См. также:

функция [CurrentSymbol\(\)](#), функция [MakeFontSymbol\(\)](#), функция [MakeSymbol\(\)](#), функция [StyleAttr\(\)](#), предложение [Symbol](#)

Функция MakeDateTime()

Назначение

Возвращает значение DateTime, составленное из заданных значений Date и Time. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakeDateTime( date, time )
```

Возвращаемая величина

Дата/Время

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim tX as time
dim dX as date
dim dtX as datetime
tX = 105604123
dX = 20070908
dtX = MakeDateTime(dX,tX)
Print FormatDate$(GetDate(dtX))
Print FormatTime$(GetTime(dtX), "hh:mm:ss.fff tt")
```

См. также:

функция [DateWindow\(\)](#), оператор [Set Date Window\(\)](#)

Функция MakeFont()

Назначение

Возвращает величину, являющуюся установкой стиля шрифта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakeFont( fontname, style, size, forecolor, backcolor )
```

fontname – имя шрифта, строковая величина (например, "Arial"); Прописные и строчные буквы для этого параметра различаются.

style – численное выражение, в результате которого получается положительное целое число; подробности приведены в описании предложения Font.

size – размер шрифта, целое число;

forecolor – цвет символов шрифта в системе RGB; Смотрите: [функция RGB\(\)](#).

backcolor – цвет фона или каймы в системе RGB. Чтобы сделать фон прозрачным, задайте значение равное -1.

Возвращаемая величина

Шрифт – Font

Описание

Функция **MakeFont()** возвращает величину типа Font для определения шрифта, который может быть назначен текстовому объекту. Возвращаемая величина может быть присвоена переменной типа Font или использована как параметр в других операторах (таких как [оператор Create Text](#) или [оператор Set Style](#)).

Для дополнительной информации о стиле шрифта см. раздел: [Предложение Font](#).

Пример

```
Include "mapbasic.def"
Dim big_title As Font
big_title = MakeFont("Arial", 1, 20,BLACK,WHITE)
```

См. также:

[функция CurrentFont\(\)](#), [предложение Font](#), [функция StyleAttr\(\)](#)

Функция MakeFontSymbol()

Назначение

Возвращает символ, используя букву (символ) шрифта TrueType. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakeFontSymbol( shape, color, size, fontname, fontstyle, rotation )
```

shape – целое число, величина типа SmallInt, от 31 или больше (31 – значение для невидимого символа), задающая код шрифта TrueType;

color – цвет в системе RGB; см. также раздел: [Функция RGB\(\)](#).

size – целое число, величина типа SmallInt, от 1 до 48, назначающая размер символа в пунктах;

fontname – строка, имя шрифта TrueType (например, "WingDings"). Прописные и строчные буквы для этого параметра различаются.

fontstyle – численный код, управляющий атрибутами шрифта, такими как жирное написание, курсив, контур;

rotation – действительное число, задающее угол поворота символа в градусах.

Возвращаемая величина

Символ

Описание

Функция **MakeFontSymbol()** возвращает величину типа Symbol, используя заданный символ шрифта TrueType. Смотрите в разделе: [Предложение Symbol](#) описание других типов символа.

Следующая таблица приводит значения для параметра *fontstyle* задающего стиль символа шрифта:

fontstyle значение	Стиль символа
0	Обычный
1	Жирный

fontstyle значение	Стиль символа
16	Черная кайма
32	Тень
256	Белая кайма

Чтобы скомбинировать два и более эффектов, числа нужно сложить. Например, для того, чтобы получить символ жирного и оттененного написания, параметр fontstyle должен быть равен 33. Белая и черная кайма взаимно исключают друг друга.

Пример

```
Include "mapbasic.def"
Dim sym_marker As Symbol
sym_marker = MakeFontSymbol(65,RED,24,"WingDings",32,0)
```

См. также:

[функция CurrentSymbol\(\)](#), [функция MakeCustomSymbol\(\)](#), [функция MakeSymbol\(\)](#), [функция StyleAttr\(\)](#), [предложение Symbol](#)

Функция MakePen()

Назначение

Возвращает установку стиля линии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MakePen( width, pattern, color )
```

width – толщина линии;

pattern – тип линии (см. список в описании предложения Pen);

color – цвет линии в системе RGB. См. подробности в разделе: [Функция RGB\(\)](#).

Возвращаемая величина

Pen

Описание

Функция **MakePen()** возвращает величину типа Pen, определяющую стиль линии графического объекта. Возвращаемая величина может быть присвоена переменной типа Pen или использована как параметр в других операторах (таких как **оператор Create Line**, **оператор Create Pline**, **оператор Set Style** или **оператор Set Map**).

См. раздел о **предложении Pen** для дополнительной информации о стиле линии.

Пример

```

Include "mapbasic.def"
Dim p_bus_route As Pen
p_bus_route = MakePen(3, 9, RED)

```

См. также:

функция CurrentPen(), **предложение Pen**, **функция StyleAttr()**, **функция RGB()**

Функция MakeSymbol()

Назначение

Возвращает установку стиля символа. Действует для символов формата MapInfo версии 3. Набор символов версии 3.0, введенный в MapInfo для работы в среде Windows 3.0, поддерживается во всех последующих версиях MapInfo Pro. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

MakeSymbol( shape, color, size )

```

shape – форма символа, целое число от 31 и более (31 для невидимого знака); стандартный набор символов использует коды от 31 до 67 (список смотрите в разделе: **Предложение Symbol**).

color – цвет в системе RGB; см. также раздел: **Функция RGB()**.

size – целое число, величина типа SmallInt, от 1 до 48, назначающая размер символа в пунктах;

Возвращаемая величина

Символ

Описание

Функция **MakeSymbol()** возвращает величину типа Symbol, определяющую стиль отображения точечного объекта. Возвращаемая величина может быть присвоена переменной типа Symbol или использована как параметр в других операторах (таких как [предложение Symbol](#), [оператор Create Point](#), [оператор Set Map](#), [оператор Set Style](#) или [оператор Shade](#)).

Чтобы создать символ из буквы или знака шрифта TrueType, используется [функция MakeFontSymbol\(\)](#).

Чтобы создать символ из растрового файла, используется [функция MakeCustomSymbol\(\)](#).

См. раздел: [Предложение Symbol](#) для дополнительной информации о стиле символа.

Пример

```

Include "mapbasic.def"
Dim sym_marker As Symbol
sym_marker = MakeSymbol(44, RED, 16)

```

См. также:

[функция CurrentSymbol\(\)](#), [функция MakeCustomSymbol\(\)](#), [функция MakeFontSymbol\(\)](#), [функция StyleAttr\(\)](#), [предложение Symbol](#)

Оператор Map

Назначение

Открывает новое окно карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Map From item [ , item ... ]
[ Position ( x, y ) [ Units paperunits ] ]
[ Width window_width [ Units paperunits ] ]
[ Height window_height [ Units paperunits ] ]
[ { Min | Max | Floating | Docked | Tabbed | AutoHidden | Hide } ]
[ Pen .... ] [ Brush ... ] [ Priority n ] [ Name frameName ]

```

```
[ Redraw { On | Off | Suspended } ]
[ Into { Window layout_win_id } { ID empty_frame_id } ]
```

где *item*:

```
table | [GroupLayer ("friendly_name" [ , item ...])
```

item — имя либо открытой таблицы, либо группы слоёв;

friendly_name — для каждой группы слоёв должно присутствовать, но не обязано быть уникальным, группы слоёв могут содержать другие группы, и/или таблицы или быть пустыми (не содержать таблиц);

paperunits — строковая величина, задающая единицу измерения листа (например, "mm");

Floating — соответствует плавающему окну.

Docked — соответствует прикреплённому в стандартном положении окну.

Tabbed — соответствует окнам, оформленным в виде вкладок, которые также, иногда, вызываться как документы.

AutoHidden — соответствует автоматически скрытому окну.

Hide — дополнительное, необязательное ключевое слово, с помощью которого создаётся окно в скрытом состоянии, и работает подобно оператору Set Window id hide. Вывести на экран скрытое окно можно, используя оператор Set Window...Show statement

Примечание: AutoHidden — соответствует автоматически скрытому окну.

Примечание: Все четыре выше перечисленных состояния окна относятся только к 64-разрядной версии MapInfo Pro.

x, *y* — координаты верхнего левого угла окна карты в "бумажных" единицах; "Бумажные" единицы измерения описаны в разделе: **Оператор Set Paper Units**. С помощью предложения **Into Window** задаётся положение относительно верхнего левого угла окна **Отчёт**.

window_width и *window_height* — ширина и высота окна в "бумажных" единицах. С помощью предложения **Into Window** задаются ширина и высота рамки в окне **Отчёт**. Если действительные значения ширины или высоты не заданы, то эти значения параметров рамки будут присвоены автоматически.

n — целое число, указывающее на значение Z-Order объектов (рамок) в окнах **Отчёт**. При выполнении копирования или при сохранении рабочего набора MapInfo Pro нормализует приоритет рамок, создавая набор уникальных значений, начиная с 1.

framename — строка, определяющая название рамки с картой в окне отчёта. Если рамке в окне **Отчёт** присваивается имя, то оно будет сохранено в рабочем наборе. Когда параметр **Name** записывается в рабочем наборе, то номер версии WOR-файла меняется на 1500.

layout_win_id — идентификатор окна **Отчёт**, целое число;

layout_win_id — идентификатор окна **Отчёт**, целое число;

Описание

Оператор **Map** открывает новое окно карты. После открытия карты можно изменять это окно, используя **оператор Set Map**.

Чтобы создавать вложенные группы слоёв, добавлено ключевое слово **GroupLayer**. Группы слоёв это специальный тип слоя, с помощью которого можно организовывать слои карт в группы, аналогично структуре папок с файлами. Группы слоёв полезны при работе с картами, на которых много слоёв. У групп слоёв два преимущества:

1. Организация - список слоёв выглядит более упорядоченным, поскольку слои организованы в группы, названия которых поддаются интерпретации.
2. Эффективность — после того слои объединены в группы, последовательно выполняемые операции, например, "выключить все слои улиц" выполняются проще и быстрее, за меньшее количество действий с мышью/шагов.

Таблица должна быть заранее открыта. Таблица должна быть также картографируемой, то есть устроена так, чтобы к ней можно было присоединять графические объекты. Не обязательно, чтобы в таблице были графические объекты, но она должна иметь структуру, позволяющую иметь объекты, связанные с информацией в записях.

В операторе **Map** должна быть задана хотя бы одна таблица, независимо от того является она слоем группы или нет, поскольку окно Карты должно содержать хотя бы один слой. Если требуется одним оператором **Map** открыть несколько таблиц в многослойной карте, то задайте их списком через запятую. При каждом обновлении карты таблицы отрисовываются в порядке, заданном в операторе **Map**; то есть, первая таблица, упомянутая в операторе **Map**, помещается на верхнем слое. Обычно первой (на верхний слой) оператор **Map** размещает таблицу с точечными объектами, а таблицы с областями (границами) **Map** помещает на нижний слой.

Размер окна по умолчанию равен примерно четверти экрана, и положение его зависит от того, сколько окон уже открыто. Задайте **Position**, **Height** и **Width**, если требуется при создании карты сами определить размер и место окна карты. Предложения **Height** и **Width** задают размеры окна в дюймах. Предложение **Position** задает расположение окна в окне MapInfo Pro, независимо от его расположения на экране.

Если в операторе **Map** задано предложение **Max**, то окно карты занимает при открытии все рабочее пространство окна MapInfo Pro. И, наоборот, включение в оператор **Map** предложения **Min** приводит к показу окна, свернутого в иконку.

Каждая карта может иметь свою проекцию. Открывая окно карты, MapInfo Pro использует проекции таблицы первого слоя. Проекцию карты можно изменить с помощью команды **Настройка карты** на вкладке **КАРТА**. Для изменения проекции карты в MapBasic используется **оператор Set Map**.

Brush — корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет

всё равно подставлен непрозрачный аналог. Это предложение создает фон для элемента оформления.

Pen — корректное [предложение Pen](#). Это предложение задает стиль рамки элемента украшения.

Для дополнительного, необязательного предложения **Redraw** можно использовать три значения — **On**, **Off** и **Suspended**. Эти параметры являются таким же, как параметры Redraw оператора Set Map. Это позволяет создавать окно карты, для которого изначально задан режим обновления. Предложение Redraw управляет автоматической перерисовкой карты в окне. Если программа выполнила оператор Set Map Redraw Off, то последующие операторы, изменяющие карту (такие как Set Map, Add Map Layer, Remove Map Layer), будут выполняться без автоматического обновления изображения в окне карты. После внесения всех нужных изменений в окно карты, выполните оператор Set Map Redraw On, восстанавливающий режим автоматической перерисовки окна. При использовании специального значения параметра Suspended на Карте появится надпись о состоянии окна.

Результатом действия оператора **Into Window** является новая рамка в окне **Отчёт**. Если не задан параметр *layout_win_id*, то создается новая рамка в самом верхнем окне **Отчёт**.

Используйте параметр **ID**, чтобы добавить открытую карту в пустую рамку отчёта. Появится карта стандартного вида, при этом будут использованы свойства рамки, заданные ещё для пустой рамки (стили линии и штриховки, положение, размер, имя и параметр z-order).

Сообщение об ошибке появится, если:

- From Window с идентификатором ID не является картой или списком: 1696 "Ошибка: требуется окно карты или списка."
- Рамка с идентификатором ID не является пустым: 1698 "Рамка ^0 отчёта должна быть пустой."
- From Window не закрывает: 1697 "Не удалось поместить окно в пустую рамку."
- Легенды карты не клонируются: 1708 "Не удалось поместить легенды в рамку карты."

В следующем примере создаётся карта на основе открытых таблиц World и World_Cities, которая добавляется в пустую рамку окна отчёта.

```
Map From World, World_Cities Into Window layout_win_id ID empty_frame_id
```

Прежде повторно использовать уже заполненную рамку, её требуется очистить от содержимого, смотрите описание [оператора Set Designer Frame](#). После выполнения такого оператора остаётся пустая рамка того же размера и в том же месте, где была заполненная рамка, присвоенное которой имя сохраняется.

Примеры

Откроем окно карты в 3 дюйма шириной и 2 дюйма высотой с двумя слоями (косметический слой не считается). Верхний левый угол окна карты будет ниже на 1 и правее на 1 дюйм от верхнего левого угла окна MapInfo. В окне карты будет 2 слоя.

```
Open Table "world"
Open Table "cust1994" As customers
Map from customers, world
  Position (1,1) Width 3 Height 2
```

В следующем примере будет открыта карта с группами слоев, некоторые из которых вложены друг в друга (предполагается, что все таблицы уже открыты).

```
Map From
  GroupLayer (
    "Grid",
    GroupLayer ("Tropics", Tropic_Of_Capricorn, Tropic_Of_Cancer),
    Wgrid15
  ),
  GroupLayer (
    "World Places", WorldPlaces, WorldPlacesMajor, WorldPlaces_Capitals
  ),
  Airports,
  GroupLayer ("World Boundaries", world_Border),
  GroupLayer (
    "Roads", Roads, US_Primary_Roads, US_Secondary_Roads, US_Major_Roads
  ),
  GroupLayer ("Countries" Countries_small, Countries_large),
  Ocean
```

Группы слоев, подобно отдельным слоям, имеют уникальный идентификатор. Идентификатором слоя может либо числа, либо имена таблиц. Если используются числа, то они представляют собой номер слоя в списке (в обратном порядке) от самого верхнего до самого нижнего. Численные идентификаторы групп слоев образуют отдельную последовательность, но также увеличиваются от самой верхней до самой нижней. В примере используются следующие группы и идентификаторы слоев:

Группа слоев	Идентификатор слоя
GroupLayer "Grid"	группа 1
GroupLayer "Tropics"	группа 2
Tropic_Of_Capricorn	слой 1
Tropic_Of_Cancer	слой 2

Группа слоев	Идентификатор слоя
Wgrid15	слой 3
GroupLayer "World Places"	группа 3
WorldPlaces	слой 4
WorldPlacesMajor	слой 5
WorldPlaces_Capitals	слой 6
Airports	слой 7
GroupLayer "World Boundaries"	группа 4
world_Border	слой 8
GroupLayer "Roads"	группа 5
Roads	слой 19
US_Primary_Roads	слой 10
US_Secondary_Roads	слой 11
US_Major_Roads	слой 12
GroupLayer "Countries"	группа 6
Countries_small	слой 13
Countries_large	слой 14
Ocean	слой 15

См. также:

[оператор Add Map](#), [оператор Set Designer Frame](#), [оператор Remove Map](#), [оператор Set Map](#), [оператор Set Shade](#), [оператор Shade](#)

Функция Map3DInfo()

Назначение

Возвращает свойства окна 3D-карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Map3DInfo( window_id, attribute )
```

window_id – идентификатор окна.

attribute – целочисленный код, указывающий, какой тип информации должен быть возвращен.

Возвращаемая величина

Вещественное (Float), целое (Integer), или строка символов (String), в зависимости от параметра атрибута.

Описание

Функция **Map3DInfo()** возвращает информацию об окне карты-призмы.

Параметр *window_id* определяет какое окно 3D-карты опрашивается. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Есть несколько числовых атрибутов, которые **Map3DInfo()** может вернуть для любого окна 3D-Карты. Параметр *attribute* сообщает функции **Map3DInfo()** какие данные об окне Карты надо возвратить. Коды *attribute* определены в файле MAPBASIC.DEF.

attribute-код	ID	Возвращаемая величина
MAP3D_INFO_SCALE	1	Вещественное, масштабный фактор 3D-карты.
MAP3D_INFO_RESOLUTION_X	2	Целое, разрешение по оси X грида в окне 3D-карты.
MAP3D_INFO_RESOLUTION_Y	3	Целое, разрешение по оси Y грида в окне 3D-карты.

attribute-код	ID	Возвращаемая величина
MAP3D_INFO_BACKGROUND	4	Целое, цвет фона, см. функцию RGB.
MAP3D_INFO_UNITS	5	Строка, представляющая сокращение единиц измерения площади, например, "mi" для миль.
MAP3D_INFO_LIGHT_X	6	Вещественное, координата X источника света.
MAP3D_INFO_LIGHT_Y	7	Вещественное (Float), координата Y источника света.
MAP3D_INFO_LIGHT_Z	8	Вещественное (Float), координата Z источника света.
MAP3D_INFO_LIGHT_COLOR	9	Целое (Integer), цвет источника, см. раздел Функция RGB() .
MAP3D_INFO_CAMERA_X	10	Вещественное (Float), координата X камеры.
MAP3D_INFO_CAMERA_Y	11	Вещественное (Float), координата Y камеры.
MAP3D_INFO_CAMERA_Z	12	Вещественное (Float), координата Z камеры.
MAP3D_INFO_CAMERA_FOCAL_X	13	Вещественное (Float), координата X фокальной точки камеры.
MAP3D_INFO_CAMERA_FOCAL_Y	14	Вещественное (Float), координата Y фокальной точки камеры.
MAP3D_INFO_CAMERA_FOCAL_Z	15	Вещественное, координата Z фокальной точки камеры.
MAP3D_INFO_CAMERA_VU_1	16	Вещественное (Float), первое значение нормального вектора точки просмотра.
MAP3D_INFO_CAMERA_VU_2	17	Вещественное (Float), второе значение нормального вектора точки просмотра.
MAP3D_INFO_CAMERA_VU_3	18	Вещественное (Float), третье значение нормального вектора точки просмотра.
MAP3D_INFO_CAMERA_VPN_1	19	Вещественное, первое значение нормального вектора плоскости просмотра.
MAP3D_INFO_CAMERA_VPN_2	20	Вещественное (Float), второе значение нормального вектора плоскости просмотра.

attribute-код	ID	Возвращаемая величина
MAP3D_INFO_CAMERA_VPN_3	21	Вещественное (Float), третье значение нормального вектора плоскости просмотра.
MAP3D_INFO_CAMERA_CLIP_NEAR	22	Вещественное (Float), приближение камеры.
MAP3D_INFO_CAMERA_CLIP_FAR	23	Вещественное (Float), удаление камеры.

Пример

Распечатка текущих значений переменных, определяющих показ окна 3D-карты:

```
include "Mapbasic.def"
Print "MAP3D_INFO_SCALE: " + Map3DInfo(FrontWindow(), MAP3D_INFO_SCALE)
Print "MAP3D_INFO_RESOLUTION_X: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_RESOLUTION_X)
Print "MAP3D_INFO_RESOLUTION_Y: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_RESOLUTION_Y)
Print "MAP3D_INFO_BACKGROUND: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_BACKGROUND)
Print "MAP3D_INFO_UNITS: " + Map3DInfo(FrontWindow(), MAP3D_INFO_UNITS)
Print "MAP3D_INFO_LIGHT_X : " + Map3DInfo(FrontWindow(),
MAP3D_INFO_LIGHT_X )
Print "MAP3D_INFO_LIGHT_Y : " + Map3DInfo(FrontWindow(),
MAP3D_INFO_LIGHT_Y )
Print "MAP3D_INFO_LIGHT_Z: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_LIGHT_Z)
Print "MAP3D_INFO_LIGHT_COLOR: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_LIGHT_COLOR)
Print "MAP3D_INFO_CAMERA_X: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_X)
Print "MAP3D_INFO_CAMERA_Y : " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_Y )
Print "MAP3D_INFO_CAMERA_Z : " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_Z )
Print "MAP3D_INFO_CAMERA_FOCAL_X: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_FOCAL_X)
Print "MAP3D_INFO_CAMERA_FOCAL_Y: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_FOCAL_Y)
Print "MAP3D_INFO_CAMERA_FOCAL_Z: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_FOCAL_Z)
Print "MAP3D_INFO_CAMERA_VU_1: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VU_1)
Print "MAP3D_INFO_CAMERA_VU_2: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VU_2)
Print "MAP3D_INFO_CAMERA_VU_3: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VU_3)
```

```
Print "MAP3D_INFO_CAMERA_VPN_1: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VPN_1)
Print "MAP3D_INFO_CAMERA_VPN_2: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VPN_2)
Print "MAP3D_INFO_CAMERA_VPN_3: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_VPN_3)
Print "MAP3D_INFO_CAMERA_CLIP_NEAR: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_CLIP_NEAR)
Print "MAP3D_INFO_CAMERA_CLIP_FAR: " + Map3DInfo(FrontWindow(),
MAP3D_INFO_CAMERA_CLIP_FAR)
```

См. также:

[оператор Create Map3D](#), [оператор Set Map3D](#)

Функция MapperInfo()

Назначение

Возвращает информацию о координатах или расстояниях в окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MapperInfo( window_id, attribute )
```

window_id — идентификатор окна.

attribute — целочисленный код, указывающий, какой тип информации должен быть возвращен. См. значения в таблице ниже.

Возвращаемая величина

Вещественное (Float), целое (Integer), или строка символов (String), в зависимости от параметра атрибута.

Описание

Функция **MapperInfo()** возвращает информацию об окне карты-призмы.

Параметр *window_id* задает идентификатор окна Карты. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Есть несколько числовых атрибутов, которые **MapperInfo()** может вернуть для любого окна карты. Параметр *attribute* сообщает функции **MapperInfo()** какие данные об окне карты надо вернуть. Коды *attribute* определены в файле `MAPBASIC.DEF`.

Параметры	ID	Возвращаемое значение MapperInfo()
MAPPER_INFO_ZOOM	1	Размер показанной части карты (расстояние от Западного до Восточного края) в единицах измерения расстояния, установленных в MapBasic (смотрите раздел Оператор Set Distance Units).
MAPPER_INFO_SCALE	2	Текущий масштаб окна карты, заданный количеством текущих единиц измерения (например, миль) в одной текущей "бумажной" единице (например, в дюйме). Результат возвращается в терминах текущих единиц измерения расстояний MapBasic.
MAPPER_INFO_CENTERX	3	X-координата центральной точки окна.
MAPPER_INFO_CENTERY	4	Y-координата центральной точки окна.
MAPPER_INFO_MINX	5	Минимальная X-координата части карты, показанной в окне.
MAPPER_INFO_MINY	6	Минимальная Y-координата части карты, показанной в окне.
MAPPER_INFO_MAXX	7	Максимальная X-координата части карты, показанной в окне.
MAPPER_INFO_MAXY	8	Максимальная Y-координата части карты, показанной в окне.
MAPPER_INFO_LAYERS	9	Количество слоев в окне карты в виде короткого целого SmallInt без Косметического слоя и без групп слоев.
MAPPER_INFO_EDIT_LAYER	10	Целое число типа SmallInt, являющееся номером изменяемого слоя. Ноль (0), если объекты изменяются в Косметическом слое; единица (1), если изменяемый слой первый некосметический, и т. д. Если результатом будет минус единица, то ни один слой не находится в изменяемом состоянии.

Параметры	ID	Возвращаемое значение MapperInfo()
MAPPER_INFO_XYUNITS	11	Строковая величина с именем координатной единицы (например, "degree").
MAPPER_INFO_DISTUNITS	12	Имя единицы измерения расстояния (например, "mi").
MAPPER_INFO_AREAUNITS	13	Строковая величина с именем единицы измерения площади (например, "sq mi" – квадратные мили).
MAPPER_INFO_SCROLLBARS	14	Логическая величина, показывающая, есть ли в окне карты полосы прокрутки.
MAPPER_INFO_DISPLAY	15	Целое число типа SmallInt, соответствующее типу информации, которая показывается в строке сообщений окна карты. См. также раздел Set Map Display. Результатом может быть один из следующих кодов: • MAPPER_INFO_DISPLAY_SCALE (0) • MAPPER_INFO_DISPLAY_ZOOM (1) • MAPPER_INFO_DISPLAY_POSITION (2) • MAPPER_INFO_DISPLAY_CARTOGRAPHIC_SCALE (3)
MAPPER_INFO_NUM_THEMATIC	16	Короткое целое число, номер слоя, который является тематическим.
MAPPER_INFO_COORDSYS_CLAUSE	17	Строка, содержащая Предложение CoordSys .
MAPPER_INFO_COORDSYS_NAME	18	Строка с именем координатной системы карты такая, как она обозначена в файле MAPINFOW.PRJ (но без суффикса "\p...", который можно видеть в файле MAPINFOW.PRJ). Возвращает пустую строку, если значение Coord-Sys не найдено в файле MAPINFOW.PRJ.
MAPPER_INFO_MOVE_DUPLICATE_NODES	19	Короткое целое, указывающее надо ли удалять дублирующиеся узлы в режиме Форма окна карты. Если значение равно 0, дублирующиеся узлы не удаляются. Если значение равно 1, все дублирующиеся узлы с одного слоя удаляются. Чтобы восстановить стандартные режимы окна, выполните оператор Set Map Move Nodes Default .

Параметры	ID	Возвращаемое значение MapperInfo()
MAPPER_INFO_DIST_CALC_TYPE	20	Короткое целое, указывающее тип алгоритма вычисления расстояния, длины, периметра и площади. См. также раздел Set Map Distance Type. Возвращает значения: • MAPPER_INFO_DIST_SPHERICAL (0) • MAPPER_INFO_DIST_CARTESIAN (1)
MAPPER_INFO_DISPLAY_DMS	21	Короткое целое число, которым задается способ отображения координат: либо в виде градусов, минут и секунд, либо – десятичных градусов, либо – Military Grid Reference System, либо в формате USNG (US National Grid). Возвращает значения: • MAPPER_INFO_DISPLAY_DECIMAL (0) • MAPPER_INFO_DISPLAY_DEGMINSEC (1) • MAPPER_INFO_DISPLAY_MGRS (2) армейская опорная референсная система • MAPPER_INFO_DISPLAY_USNG_WGS84 (3) Национальная опорная сеть США NAD 83/WGS 84 • MAPPER_INFO_DISPLAY_USNG_NAD27 (4) Национальная опорная сеть США NAD 27
MAPPER_INFO_COORDSYS_CLAUSE_WITH_BOUNDS	22	Строка, содержащая предложение CoordSys для этого окна.
MAPPER_INFO_CLIP_TYPE	23	Тип отсечения, применяемый к карте. Варианты включают: • MAPPER_INFO_CLIP_DISPLAY_ALL (0) • MAPPER_INFO_CLIP_DISPLAY_POLYOBJ (1) • MAPPER_INFO_CLIP_OVERLAY (2)
MAPPER_INFO_CLIP_REGION	24	Возвращает строку, определяющую, используется ли регион для отсечения части карты. Возвращает "on", если применяется регион отсечения. В других случаях, возвращает "off".

Параметры	ID	Возвращаемое значение MapperInfo()
MAPPER_INFO_REPROJECTION	25	Строковое значение, указывающее текущий режим перепроецирования. Возможны следующие значения: • None – никогда не перепроецировать карту. • Always – всегда перепроецировать карту. • Auto – перепроецировать карту или нет, будет решать MapInfo Pro.
MAPPER_INFO_RESAMPLING	26	Строковое значение, указывающее метод расчёта значений пикселей исходного растра при перепроецировании. Возможны следующие значения: • Кубическая свертка • Ближайшее соседство
MAPPER_INFO_MERGE_MAP	27	Строковое значение – String: строка, представляющая собой последовательность операторов MapBasic, которая может потребоваться для включения окна карты в текущее окно карты.
MAPPER_INFO_ALL_LAYERS	28	Возвращает количество слоёв и групп слоёв (включая все вложенные слои и группы слоёв).
MAPPER_INFO_GROUPLAYERS	29	Возвращает количество групп слоёв (включая вложенные группы слоёв).
MAPPER_INFO_LABELS_SELECTABLE	30	Логическое значение, отвечающее за выбор подписей на карте.
MAPPER_INFO_HAS_CUSTOM_LABELS	31	Логическое значение, представляющее наличие на карте индивидуальных подписей или их отсутствие.
MAPPER_INFO_CARTO_SCALE	32	Вещественное значение картографического масштаба, вычисленное MapInfo Pro прежде, чем оно округлено для использования на экране.

Параметры	ID	Возвращаемое значение MapperInfo()
MAPPER_INFO_CARTO_SCALE_EXPR	33	Строковое выражение картографического масштаба, похожее на то, которое отображается в строке состояния окна карты, т.е., если картографический масштаб обозначается в виде числа 945 000, то возвращается строковое выражение "1:945 000". Строка форматируется в соответствии с региональными настройками, принятыми для отображения чисел.
MAPPER_INFO_NUM_ADORNMENTS	200	Возвращает целое, описывающее количество элементов оформления карты. Пользуйтесь любым значением, выходящим за пределы диапазона значений MapperInfo, например, 100.
MAPPER_INFO_ADORNMENT+n	200	Возвращает идентификатор окна WindowID заданного элемента оформления карты.

Когда **MapperInfo()** вызывается для получения значений координат (с указанием MAPPER_INFO_CENTERX в качестве attribute), возвращаемое значение будет координатами в текущей для MapBasic системе координат, которая может отличаться от системы координат в окне карты. Используйте **оператор Set CoordSys** для задания другой системы координат.

Настройки для окна карты и обеспечение поддержки MapBasic можно делать для каждого окна карты. Можно задавать способы совмещения узлов объектов при их перемещении в режиме Форма.

Когда создано новое окно карты, можно настроить режим совмещения при перемещении (Настройки > Режимы > Окно Карты > Совмещать при перемещении).

Существующее окно карты может быть опрошено на предмет параметров режима совмещения при перемещении использованием новых атрибутов функции **MapperInfo()**.

Для изменения текущих настроек используется **оператор Set Map**.

Информация о возвращаемых координатах

MapperInfo() не возвращает координат (например, MINX, MAXX, MINY, MAXY) в единицах измерения, заданных на Карте. Координаты возвращаются либо в единицах внутренней координатной системы MapInfo Pro, действующей в рамках данного сеанса, либо в терминах координатной системы программы MapBasic, вызывающей данную функцию. При задании атрибута MAPPER_INFO_XYUNITS возвращаются единицы, которые используются для показа положения указателя мыши на экране в строке состояний (устанавливается оператором **Set Map Window Frontwindow**).

Информация о врезке региона

Начиная с MapInfo Pro 6.0, существуют 3 метода, применяющихся для создания врезки. Метод MAPPER_INFO_CLIP_OVERLAY (2) является методом, который был единственным вариантом до выхода MapInfo Pro 6.0. Когда используется этот метод, **функция Overlap()** (команда **Удалить внешнюю часть**) используется внутренне. Так как **функция Overlap()** не затрагивает текстовые объекты, тексты этим методом не обрезаются. Точечные объекты обрезаются только в том случае, если они целиком выходят за область врезки. Подписи обрабатываются аналогично: они либо показываются, если подпись целиком попадает в область врезки, либо отбрасываются. Стилизованные элементы (линии с заданной шириной, символы, тексты и пр.) при использовании этого метода не обрезаются.

Метод MAPPER_INFO_CLIP_DISPLAY_ALL (0) использует механизм Windows Display для организации врезки. Этот метод обрезает все типы объектов. Тематика, растры и сетки (гриды) тоже обрезаются. Стили (ширина линий, символы, текст) – тоже. Это стандартный метод выполнения отсечений.

Метод MAPPER_INFO_CLIP_DISPLAY_POLYOBJ (1) есть сочетание системной функции Windows Display и функциональности метода MAPPER_INFO_CLIP_OVERLAY (2). Функция Windows Display Clipping применяется к полигональным и полилинейным объектам и к объектам, которые могут быть преобразованы в таковые (прямоугольники, скругленные прямоугольники, эллипсы, дуги). Вся стилизация этих объектов и связанные с ними символы также обрезаются. Точечные объекты, подписи и тексты обрабатываются также, как и в методе MAPPER_INFO_CLIP_OVERLAY (2). Этот комбинированный метод часто дает лучшие результаты, чем метод MAPPER_INFO_CLIP_OVERLAY (2).

Условия появления сообщения об ошибке

ERR_BAD_WINDOW (590), если нет такого окна;

ERR_FCN_ARG_RANGE (644), если значение аргумента выходит за пределы, заданные при его определении.

ERR_WANT_MAPPER_WIN (313), если окно не карта.

См. также:

функция LayerInfo(), **оператор Set Distance Units**, **оператор Set Map**

Функция Maximum()

Назначение

Возвращает наибольшее. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Maximum( num_expr, num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Maximum()** возвращает наибольшее из двух чисел, заданных численными выражениями.

Пример

```
Dim x, y, z As Float
x = 42
y = 27
z = Maximum(x, y)
```

z равно 42

См. также:

[функция Minimum\(\)](#)

Функция MBR()

Назначение

Возвращает прямоугольный объект, представляющий минимальное прямоугольное покрытие заданного объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MBR( obj_expr )
```

obj_expr – выражение, определяющее объект.

Возвращаемая величина

Величина типа Object. Графический объект типа "прямоугольник".

Описание

Функция **MBR()** возвращает графический объект – наименьший прямоугольник, в который можно вписать объект, заданный выражением *obj_expr*.

Такой прямоугольник представляет наименьший прямоугольник, в который покрывает объект целиком. Например, минимальное прямоугольное покрытие США представляет собой прямоугольник, у которого правая сторона включает в себя самую западную точку границы штата Мен, нижняя сторона – самую южную точку границы Гавайи, и левая и верхняя стороны – самую восточную и самую северную точки границы штата Аляска.

Минимальное прямоугольное покрытие точечного объекта имеет нулевую ширину и нулевую высоту.

Пример

```
Dim o_mbr As Object
Open Table "world"
Fetch First From world
o_mbr = MBR(world.obj)
```

См. также:

[функция Centroid\(\)](#), [функция CentroidX\(\)](#), [функция CentroidY\(\)](#)

Оператор Menu Bar – устаревший

Назначение

Показывает или скрывает строку меню. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Menu Bar { Hide | Show }
```

Описание

Оператор **Menu Bar** управляет отображением строки меню в рабочем окне MapInfo Pro. Используя этот оператор, программа может освободить больше места на экране для окна карты, списка, отчёта или графика.

Чтобы вновь показать строку меню, скрытую оператором **Menu Bar Hide**, используйте оператор **Menu Bar Show**. Оператор **Menu Bar Hide** следует использовать аккуратно, так как пользователь может быть поставлен в тупик, оказавшись без строки меню. За каждым оператором **Menu Bar Hide** по возможности, должен следовать оператор **Menu Bar Show**.

Пока строка меню скрыта, MapInfo Pro будет игнорировать клавишные сокращения для вызова команд. Т.е., например, можно использовать клавиши **Ctrl+O** для вызова диалога команды **Открыть**, но если строка меню скрыта, то нажатие **Ctrl+O** ни к чему не приведет.

См. также:

оператор **Alter Menu Bar**, оператор **Create Menu Bar**

Функция MenuItemEnabled()

Назначение

Возвращает идентификаторы включённых команд меню. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MenuItemEnabled(menuitem_ID)
```

Возвращаемая величина

Функция возвращает TRUE, если команда меню включена, FALSE, если — не включена.

Функция MenuItemInfoByHandler()

Назначение

Возвращает информацию об элементе меню MapInfo Pro.

Синтаксис

```
MenuItemInfoByHandler( handler, attribute )
```

handler – либо строка с именем процедуры-обработчика, заданной для элемента меню предложением **Calling**, либо целое число (тип Integer), код, который был задан в предложении **Calling**.

attribute – код возвращаемого атрибута, см. таблицу ниже.

Описание

Параметр *handler* может быть как строковым, так и численным. Если был выбран строковый вид (имя процедуры), а при этом соответствующую процедуру вызывают два или более элемента меню, то MapInfo Pro возвратит идентификатор первого элемента меню, вызвавшего эту процедуру. Если требуется опросить несколько элементов меню, которые обращаются к одной обрабатывающей процедуре, дайте каждому такому элементу меню ID-номер (например, используя предложение **ID**, когда вызывается **оператор Create Menu**), и далее используется **функция MenuItemInfoById()** вместо **MenuItemInfoByHandler()**.

В следующей таблице в первой колонке приводятся имена кодов, установленных в файле стандартных определений MapBasic MAPBASIC.DEF):

Параметры	ID	Возвращаемая величина:
MENUITEM_INFO_ENABLED	1	Логическая величина: “Да” (TRUE), если элемент меню активен.
MENUITEM_INFO_CHECKED	2	“Да” (TRUE), если элемент меню можно фиксировать и в данный момент он фиксирован (есть галочка); “Да” (TRUE) также, если элемент меню имеет несколько вариантов текста (например, Показать... и Скрыть...), и при этом элемент меню находится в состоянии “Показать”. “Нет” (FALSE) во всех остальных случаях.
MENUITEM_INFO_CHECKABLE	3	Логическая величина: “Да” (TRUE), если элемент меню фиксируется (рядом с именем элемента в меню может появляться галочка).
MENUITEM_INFO_SHOWHIDEABLE	4	TRUE, если пункт меню имеет альтернативные подсказки (например, в случае пары Показать... и Скрыть...). Несколько вариантов текста задаются помещением символа “!” в начало строки описания элемента меню оператор Create Menu или оператор Alter Menu и символа (^) перед началом альтернативного текста.

Параметры	ID	Возвращаемая величина:
MENUITEM_INFO_ACCELERATOR	5	Строка, величина типа String: строковый код акселератора элемента меню (например, “/W^Z” или “/W#%119”) или пустая строка, акселератор не был назначен. Информацию о назначении элементу меню акселератора смотрите в разделе оператор Create Menu .
MENUITEM_INFO_TEXT	6	полный текст, используемый при создании элемента меню (например, когда выполняется оператор Create Menu).
MENUITEM_INFO_HELPMSG	7	подсказка для элемента меню, которая была назначена в предложении HelpMsg (см. раздел Оператор Create Menu), или пустая строка, если подсказка не назначалась.
MENUITEM_INFO_HANDLER	8	Целое число типа Integer, номер обработчика элемента меню. Если при создании элемента меню в предложении Calling был задан код (например, <code>Calling M_FILE_SAVE</code>), то результатом будет соответствующая константа. Если предложение Calling задает “OLE”, “DDE” или имя процедуры, то результатом будет уникальное целое число, которое может быть принимать в качестве аргумента функция MenuItemInfoByHandler() или оператор Run Menu Command .
MENUITEM_INFO_ID	9	Целое – Integer: идентификатор элемента меню, который был назначен предложением ID (см. раздел Оператор Create Menu), или 0, если элемент меню не имеет идентификатора.

См. также:

[функция MenuItemInfoByID\(\)](#)

Функция MenuItemInfoByID()

Назначение

Возвращает информацию об элементе меню MapInfo Pro.

Синтаксис

```
MenuItemInfoByID( menuitem_ID, attribute )
```

menuitem_ID – целое число типа Integer, идентификатор элемента меню, который он получил при создании в предложении ID оператора Create Menu).

attribute – целочисленный код, определяющий тему запроса.

Описание

Эта функция работает так же, как и [функция MenuItemInfoByHandler\(\)](#), за исключением того, в первом параметре задается целочисленный идентификатор.

Эта функция запрашивает элемент меню по его идентификатору. Функция [MenuItemInfoByHandler\(\)](#) запрашивает элемент меню по его обработчику, который запускается, когда пользователь выберет этот элемент меню.

Параметр *attribute* должен быть целочисленным кодом, одним из тех, имена которым присвоены в файле MAPBASIC.DEF, например, MENUITEM_INFO_CHECKED (2). Список возможных значений параметра *attribute* и возвращаемые результаты см. в разделе [функция MenuItemInfoByHandler\(\)](#).

См. также:

[функция MenuItemInfoByHandler\(\)](#)

Оператор Metadata

Назначение

Управление метаданными таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Metadata Table table_name
{ SetKey key_name To key_value |
  DropKey key_name [ Hierarchical ] |
  SetTraverse starting_key_name [ Hierarchical ]
  Into ID traverse_ID_var }
```

table_name – имя открытой таблицы;

key_name – строковая величина, представляющая собой имя ключа метаданных. Она должна начинаться с обратной косой черты ("\") и не должна заканчиваться обратной косой чертой.

key_value – строка до 239 символов длиной, значение, присваиваемое ключу;

starting_key_name – строка, представляющая первое имя ключа для извлечения соответствующего значения из таблицы. Чтобы начать структурное извлечение с самого начала списка ключей, добавьте “\” (обратную косую черту);

traverse_ID_var – имя переменной типа Integer. С помощью этой переменной MapInfo управляет последовательными операторами **Metadata Traverse**....

Синтаксис 2

```
Metadata Traverse traverse_ID
{ Next Into Key key_name_var In key_value_var |
  Destroy }
```

traverse_ID – целое число типа Integer (такое как значение переменной *traverse_ID_var* из предыдущего варианта синтаксиса оператора);

key_name_var – имя строковой переменной (MapInfo помещает в эту переменную название ключа для извлечения);

key_value_var – имя строковой переменной (MapInfo Pro помещает в эту переменную название ключа для извлечения);

Описание

Оператор Metadata управляет метаданными, размещаемыми в таблице MapInfo. Метаданные – это информация, размещаемая в файле таблицы (TAB), а не в строках и столбцах файла данных.

Каждая таблица может иметь ключи метаданных - от нуля и выше. Каждый ключ определяет категорию хранимой информации, такой как имя автора, его права и т.д. Каждому ключу соответствует некое строчное значение. Например, ключу “\Copyright” может соответствовать значение “Copyright 2001 Precisely Corporation.” Более подробно метаданные описываются в *Руководстве пользователя MapBasic*.

Изменение метаданных в таблице

Чтобы создать, поменять или удалить метаданные, используйте Синтаксис 1. При этом используются следующие предложения:

SetKey

Присваивает значение ключу. Если ключ уже существует, то MapInfo Pro присваивает ему новое значение. Если ключ не существует, MapInfo Pro создает его. Если требуется создать метаданные, то они записываются немедленно; не нужно проводить операцию сохранения.

```
MetaData Table Parcels SetKey "\Info\Date" To Str$(CurDate())
```

Примечание: MapInfo Pro автоматически создает ключ метаданных “\IsReadOnly” (со стандартным значением “FALSE”) в первый раз, когда добавляете метаданные в таблицу. Ключ “\IsReadOnly” – это специальный ключ, который MapInfo использует для своих нужд.

DropKey

Удаляет ключ из таблицы. Если добавить слово **Hierarchical**, MapInfo Pro удаляет вместе с ключом всю структуру метаданных, подчиненных этому ключу. Например, если в таблице есть ключи “\Info\Author” и “\Info\Date”, то оба они удаляются следующим оператором:

```
MetaData Table Parcels DropKey "\Info" Hierarchical
```

Чтение метаданных из таблицы

Чтобы прочитать метаданные из таблицы, используется предложение **SetTraverse**, которое инициализирует операцию структурного извлечения, продолжаемую затем одноразовыми извлечениями значений предложением **Next**. Для завершения операции извлечения применяется предложение **Destroy**, которое освобождает память, используемую при этой операции. При этом используются следующие предложения:

SetTraverse

Подготавливает операцию структурного извлечения метаданных, начиная с определенного ключа. Чтобы начать извлечение с самого первого в иерархии ключа, задайте в качестве имени начального ключа “\”. Если будет добавлено слово **Hierarchical**, то операция извлечения пройдет по всей структуре. Если слово **Hierarchical** опущено, то операция извлечения не будет опускаться на уровни ниже начального (т.е. извлечение, стартовавшее с ключа “\Info”, не затронет ключ “\Info\Date”).

Next Into Key... Into Value...

Попытка чтения следующего ключа. Если еще остаются ключи для чтения, то MapInfo Pro помещает ключ в переменную *key_name_var* и значение, соответствующее ключу, в переменную *key_value_var*. Если ключи исчерпаны, MapInfo Pro помещает в обе переменные пустые значения.

Destroy

Завершает операцию структурного извлечения и освобождает занятую под нее память.

Примечание: Операция структурного извлечения может проникать до десятого уровня иерархии (т.е. “\Один\Два\Три\Четыре\Пять\Шесть\Семь\Восемь\Девять\Десять”), начиная с нулевого или корневого (“\”). Если требуется проникнуть глубже, чем на 10 уровней, то начинайте операцию структурного извлечения с ненулевого уровня (например, с уровня “\Один\Два\Три\Четыре\Пять”).

Пример

Следующая процедура извлекает все метаданные из таблицы; имя таблицы определяется вызывающей программой. Все ключи и соответствующие им значения распечатываются в окне Сообщения.

```
Sub Print_Metadata(ByVal table_name As String)
  Dim i_traversal As Integer
  Dim s_keyname, s_keyvalue As String

  ' Initialize the traversal:
  Metadata Table table_name
  SetTraverse "\" Hierarchical Into ID i_traversal

  ' Attempt to fetch the first key:
  Metadata Traverse i_traversal
  Next Into Key s_keyname Into Value s_keyvalue

  ' Now loop for as long as there are key values;
  ' with each iteration of the loop, retrieve
  ' one key, and print it to the Message window.
  Do While s_keyname <> ""
    Print " "
    Print "Key name: " & s_keyname
    Print "Key value: " & s_keyvalue

    Metadata Traverse i_traversal
    Next Into Key s_keyname Into Value s_keyvalue
  Loop

  ' Release this traversal to free memory:
  Metadata Traverse i_traversal Destroy

End Sub
```

См. также:

[функция GetMetadata\(\)](#), [функция TableInfo\(\)](#)

Функция MGRSToPoint()

Назначение

Преобразует строку с координатами в формате MGRS (Military Grid Reference System) в точечные объекты в текущей координатной системе MapBasic. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MGRSToPoint( string )
```

string – строку с координатами в формате MGRS.

Как начальное значение используется стандартная система "Широта/Долгота".

Возвращаемая величина

Объект

Описание

Возвращает точки текущей координатной системе MapBasic, которая по умолчанию есть "Широта/Долгота" (без референц-эллипсоида). Чтобы добиться максимальной точности, задайте координатную систему MapBasic такой же, как и в исходной таблице до вызова **MGRSToPoint()**. Иначе MapInfo Pro проведет ненужные преобразования координатных систем, что негативно отразится на точности.

Пример

Пример 1:

```
dim obj1 as Object
dim s_mgrs As String
dim obj2 as Object
obj1 = CreatePoint(-74.669, 43.263)
s_mgrs = PointToMGRS$(obj1)
obj2 = MGRSToPoint(s_mgrs)
```

Пример 2:

```
Open Table "C:\Temp\MyTable.TAB" as MGRSfile
' When using the PointToMGRS$() or MGRSToPoint() functions,
' it is very important to make sure that the current MapBasic
' coordsys matches the coordsys of the table where the
' point object is being stored.
'Set the MapBasic coordsys to that of the table used
Set CoordSys Table MGRSfile
'Update a Character column (for example COL2) with MGRS strings from
'a table of points
Update MGRSfile
Set Col2 = PointToMGRS$(obj)
'Update two float columns (Col3 & Col4) with
'CentroidX & CentroidY information
'from a character column (Col2) that contains MGRS strings.
Update MGRSfile
```

```
Set Col3 = CentroidX(MGRSToPoint(Col2))
Update mgrstestfile ' MGRSfile
Set Col4 = CentroidY(MGRSToPoint(Col2))
Commit Table MGRSfile
Close Table MGRSfile
```

См. также:

функция **PointToMGRS()**

Функция Mid\$

Назначение

Возвращает строку, извлекая ее из середины другой. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Mid$( string_expr, position, length )
```

string_expr – строковое выражение.

position – целочисленное выражение, результат которого есть номер первого символа, извлекаемого из строки;

length – длина извлекаемой подстроки, целое число.

Возвращаемая величина

Строка – String

Описание

Функция **Mid\$()** возвращает подстроку из строки, заданной выражением *string_expr*.

Mid\$() копирует *length* символов из *string_expr*, начиная с символа, номер которого определен параметром *position*. Если значение этого параметра меньше или равно единице, то копирование будет производиться с самого начала строки *string_expr*.

Если длина строки *string_expr* меньше, чем заданный параметр *length*, то функция **Mid\$()** вернет укороченную строку. Если параметр *position* задает позицию за пределами строки *string_expr*, то функция **Mid\$()** возвратит пустую строку. Если параметр *length* меньше единицы функция **Mid\$()** также возвратит пустую строку.. Все дробные параметры MapBasic округляет до ближайших целых.

Пример

```
Dim str_var, substr_var As String
str_var = "New York City"
substr_var = Mid$(str_var, 10, 4)

' substr_var now contains the string "City"
```

См. также:

функция [InStr\(\)](#), функция [Left\\$\(\)](#), функция [Right\\$\(\)](#)

Функция MidByte()

Назначение

Позволяет извлекать байты из строки, состоящей из двухбайтовых символов (например, Windows Japanese). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
MidByte$( string_expr, position, length )
```

string_expr – строковое выражение.

position – целочисленное выражение, результат которого есть номер первого символа, извлекаемого из строки;

length – целочисленное выражение, задающее количество извлекаемых байт.

Возвращаемая величина

Строка – String

Описание

Функция **MidByte\$ ()** возвращает заданный байт из строки.

Функция **MidByte\$ ()** используется для извлечения нескольких байтов из строки, образованной символами двухбайтовой кодировки. Например, в японской версии Microsoft Windows используется двухбайтовая система.

В операционных системах, в которых используется однобайтовые наборы символов, или в Unicode-версиях MapInfo Pro, результаты, возвращаемые функцией **MidByte\$()** идентичны результатам, возвращаемым функцией [Mid\\$\(\)](#).

См. также:

[функция InStr\(\)](#), [функция Left\\$\(\)](#), [функция Right\\$\(\)](#)

Функция Minimum()

Назначение

Возвращает наименьшее. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Minimum( num_expr, num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное – Float

Описание

Функция **Minimum()** возвращает наименьшее из двух чисел, заданных численными выражениями.

Пример

```
Dim x, y, z As Float  
x = 42  
y = -100  
z = Minimum(x, y)
```

z равно -100

См. также:

[функция Maximum\(\)](#)

Функция Minute()

Назначение

Извлекает минуты из значения времени (Time) в виде целого в диапазоне 0-59. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Minute( Time )
```

Возвращаемая величина

Короткое целое – SmallInt

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim X as time  
dim iMin as integer  
X = CurDateTime()  
iMin = Minute(X)  
Print iMin
```

См. также:

[функция Hour\(\)](#), [функция Second\(\)](#)

Функция Month()

Назначение

Возвращает из даты компоненту, соответствующую номеру месяца в году (1 – 12). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Month( date_expr )
```

date_expr – выражение, результат которого есть дата (величина типа Data).

Возвращаемая величина

Короткое целое число (SmallInt) от 1 до 12, включительно.

Описание

Функция **Month()** возвращает целое число, являющееся номером месяца в дате.

Примеры

Определим, какой сейчас месяц с помощью функции **Month()**:

```
If Month(CurDate()) = 12 Then
    '
    ' ... then it is December...
    '
End If
```

Функцию **Month()** можно использовать в SQL-запросе. В примере подразумевается, что в таблица ORDERS есть колонка с данными типа Date, названная "Order_Date". Предложение Where оператора Select предписывает MapInfo Pro выбирать заказы только за декабрь 2013.

```
Open Table "orders"
Select *
  From orders
 Where Month(orderdate) = 12 And Year(orderdate) = 2013
```

См. также:

[функция CurDate\(\)](#), [функция Day\(\)](#), [функция Minute\(\)](#), [функция Month\(\)](#), [функция Second\(\)](#), [функция Weekday\(\)](#), [функция Year\(\)](#)

Оператор Nearest

Назначение

Ищет объект в таблице, который находится ближе других объектов к нужному. Результат выдаётся в виде объекта-полилинии между 2 точек, и представляет самое кратчайшее расстояние между объектами. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Nearest [ N | All ]
  From { Table fromtable | Variable fromvar }
  To totable Into intotable
  [ Type { Spherical | Cartesian } ]
  [ Ignore [ Contains ] [ Min min_value ] [ Max max_value ]
    Units unitname ] [ Data clause ]
```

N — параметр (необязательный), представляющий число "ближайших" объектов, которые будут найдены. Стандартно, по умолчанию используется число 1. Если используется параметр **All**, то объекты с расстояниями будут создаваться для каждой комбинации.

fromtable — таблица объектов, от которых будут искаться кратчайшие расстояния.

fromvar — переменная MapBasic для объекта, от которого ищется кратчайшее расстояние.

totable — таблица объектов, до которых будут искаться кратчайшие расстояния.

intotable — таблица, в которую будут записаны полученные результаты.

min_value — минимальная допустимая дистанция для результата;

max_value — максимальная допустимая дистанция для результата;

unitname — единицы измерения параметров *min_value* и/или *max_value* (например, "km").

clause — выражение, задающее таблицы *fromtable* и *totable*, из которых извлекаются результаты.

Описание

Оператор **Nearest** находит все объекты из таблицы *fromtable* которые ближе всего к заданному объекту. Рассматривается каждый объект в таблице *fromtable*. Для каждого объекта в таблице *fromtable*, будет найден наиболее близкий объект в таблице *totable*. Если задан параметр *N*, то будет найдено *N* наиболее близких объектов в таблице *totable*. Объект-полилиния, соединяющий наиболее близкие точки между объектом *fromtable* и выбранным объектом *totable*, помещается в таблицу *intotable*. Если задан параметр **All**, то объект помещается в таблицу *intotable*, представляющую расстояние между объектом *fromtable* каждым объектом таблицы *totable*.

Если имеется несколько объектов в таблице *totable*, которые имеют одинаковое расстояние до заданного объекта *fromtable*, только один из них может быть возвращён в виде результата. Если возвращается несколько объектов (задана величина *N*, большая чем 1), то объекты с одинаковым расстоянием будут записаны последовательно. Если существует второй объект с таким же расстоянием, а запрашивается 3 объекта, то искомым объектом станет третий по счёту объект.

Типы объектов в таблицах *fromtable* и *totable* могут быть любыми кроме текстов. Например, если обе таблицы содержат объекты-регионы, то находится минимальное расстояние между объектами-регионами — создаётся объект-полилиния, соединяющий две точки каждого объекта, использованных для расчёта расстояния. Если объекты-регионы пересекаются, минимальное

расстояние будет нулевым, и объект-полилиния будет вырожден до двух точек на пересечении, имеющими одни и те же координаты.

Расстояние, рассчитанное этой функцией, не принимается во внимание при расчёте маршрутов в транспортных задачах. Это расстояние можно сравнить с "перелётом птицы".

Type — метод, используемый для расчета расстояния между объектами. Он может быть либо сферическим (**Spherical**), либо декартовым (**Cartesian**). Тип алгоритма вычислений должен соответствовать системе координат результирующей таблицы *intotable* или произойдет ошибка. Если система координат результирующей таблицы *intotable* не "Широта/Долгота", а метод расчёта сферический, то будет выдано сообщение об ошибке. Если система координат результирующей таблицы *intotable* — "Широта/Долгота", а метод расчёта декартовый, то будет выдано сообщение об ошибке.

Предложение **Ignore** ограничивает возвращаемые расстояния. Любые расстояния, найденные по запросу, которые меньше или равны *min_value* или больше *max_value* - игнорируются; *min_value* и *max_value* являются расстояниями и измеряются в единицах, определенных параметром *unitname*. Если *unitname* не соответствует единицам измерения, будет выдано сообщение об ошибке. С помощью параметра **Min** можно исключать нулевые расстояния. Это может быть полезно в случае двух таблиц с точками, чтобы не рассчитывать расстояния от одинаковых точек. Например, если две таблицы с точками, представляют города, и мы хотим найти ближайшие города, мы можем исключить из поиска одинаковые города. Предложение **Ignore** не является обязательным, равно как и подпредложения **Min** и **Max**.

С помощью параметра **Max** можно ограничить число объектов, рассматриваемых в *totable*. Это полезно при совместном использовании с параметрами *N* или **All**. Например, надо найти 5 аэропортов, находящихся ближе остальных для нескольких городов (где *fromtable* это набор городов, а *totable* это набор аэропортов), но мы не будем рассматривать аэродромы, удалённые более чем на 100 км. Результат может оказаться меньшим, чем пять аэропортов. Этот параметр целесообразно использовать совместно с параметром **All**, так можно найти все аэропорты на расстоянии 100 км от города. Применение параметра **Max** может улучшить поведение оператора **Nearest**, поскольку он эффективно ограничивает число искомых объектов *totable*.

Найденные расстояния находятся так, что они строго больше чем *min_value* и меньше или равны *max_value*:

```
min_value < distance <= max_value
```

Можно сделать так, чтобы расстояния возвращались в виде нескольких проходов оператора **Nearest**. Например, первый прогон может вернуть все объекты на расстоянии от 0 до 100 км, второй проход может вернуть все объекты на расстоянии от 100 до 200 км, а результаты не должны содержать повторяющихся записей (то есть, расстояние 100 встретилось только в первом проходе, а во втором уже не встречается).

Если один объект находится внутри другого, то расстояние между ними считается равным нулю. Например, если таблица *fromtable* это WORLDCAPS, а таблица *totable* - это WORLD, то расстояние между Москвой и Россией будет равно нулю. Если же флаг **Contains** в предложении **Ignore** установлен, то расстояние не будет обязательно нулевым. Вместо этого результатом

будет расстояние между Москвой и границей России. Все замкнутые объекты, типа полигонов (регионов) будут обработаны как полилинии для выполнения этой операции.

Предложение Data

Предложение **Data** используется для того, чтобы отметить от каких объектов из таблиц *fromtable* и *totable* будут получен результат.

```
Data IntoColumn1=column1, IntoColumn2=column2
```

Переменная *IntoColumn* в левой части уравнения должна корректно задавать колонку в *intotable*. В правой части от знака равенства должны также находиться корректно заданные колонки из таблиц *totable* или *fromtable*. Если одинаковое имя колонки существует в обеих таблицах *totable* и *fromtable*, то будет использоваться колонка из таблицы *totable* (т.е. в таблице *totable* в первую очередь ищутся имена колонок, а это правая часть "уравнения"). Чтобы избежать конфликтов такого типа, имена колонок можно заменять псевдонимами таблиц.

```
Data name1=states.state_name, name2=county.state_name
```

Для того чтобы заполнить колонку таблицы *intotable* расстояниями, можно использовать либо команду **Обновить колонку** на вкладке **ТАБЛИЦА**, либо **оператор Update**.

Примеры

Предположим, имеется таблица с точечными объектами, представляющими местоположение банкоматов, в ней имеется как минимум две колонки: колонка с фирмами, то есть с именами фирм, где есть банкоматы, и адресную колонку с почтовым адресом фирмы. Предположим, что текущая выборка представляет наше текущее местоположение. Тогда можно найти точку с ближайшим от нас банкоматом:

```
Nearest From Table selection To atm Into result Data
where=Business,address=Address
```

Если надо найти 5 ближайших банкоматов:

```
Nearest 5 From Table selection To atm Into result Data
where=Business,address=Address
```

Если надо найти все банкоматы в радиусе 5 км:

```
Nearest All From Table selection To atm Into result Ignore Max 5 Units
"mi" Data where=buisness,address=address
```

Предположим, у нас есть таблица с размещением домов (*fromtable*) и таблица с береговой линией (*totable*). Чтобы найти расстояние от данного дома до береговой линии:

```
Nearest From Table customer To coastline Into result Data
who=customer.name,
where=customer.address,coast_loc=coastline.county,type=coastline.designat
ion
```

Мы не будем рассматривать местоположения клиентов, которые расположены дальше чем 30 км:

```
Nearest From Table customer To coastline Into result Ignore Max 30 Units
"mi" Data who=customer.name,
where=customer.address,coast_loc=coastline.county,
type=coastline.designation
```

Предположим, имеется таблица городов (*fromtable*) и другая таблица столиц (*totable*), и надо найти ближайшую столицу для каждого города, но нам надо проигнорировать те случаи, когда город в таблице *fromtable* является также столицей:

```
Nearest From Table uscty_1k To usa_caps Into result Ignore Min 0 Units
"mi" Data city=uscty_1k.name,capital=usa_caps.capital
```

См. также:

оператор **Farthest**, функция **CartesianObjectDistance()**, функция **ObjectDistance()**, функция **SphericalObjectDistance()**, функция **CartesianConnectObjects()**, функция **ConnectObjects()**, функция **SphericalConnectObjects()**

Оператор Note

Назначение

Показывает сообщение в простом диалоговом окне. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Note message
```

message – выражение, результат которого будет показан в окне.

Описание

Оператор **Note** создает простое диалоговое окно для сообщений. В нем есть кнопка **ОК**; сообщение присутствует на экране, пока кнопка **ОК** не будет нажата.

Параметр message может быть выражением, не обязательно строковым. При вычислении выражения message получается величина объектного типа (Object), MapBasic автоматически преобразует его в строку, представляющую тип объекта в диалоге **оператора Note**. Если message – строка, то максимальная длина должна быть не более 300 символов и может занимать только 6 строк.

Пример

```
Note "Total # of records processed: " + Str$( i_count )
```

См. также:

[функция Ask\(\)](#), [оператор Dialog](#), [оператор Print](#)

Функция NumAllWindows()

Назначение

Возвращает количество окон, открытых MapInfo Pro, включая специальные окна, такие как инструментальные панели и окно Информации. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumAllWindows ( )
```

Возвращаемая величина

Короткое целое – SmallInt

Описание

Функция **NumAllWindows()** и возвращает количество окон, открытых MapInfo Pro.

Чтобы определить количество открытых “документальных” окон MapInfo (**Карта**, **Список**, **График** и **Отчётов**), используйте функцию **NumWindows()**.

См. также:

[функция NumWindows\(\)](#), [функция WindowID\(\)](#)

Функция NumberToDate()

Назначение

Возвращает величину типа Date, созданную из величины типа Integer. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumberToDate ( numeric_date )
```

numeric_date – целое число, представляющее дату в формате ГГГГММДД, где ГГГГ – год, ММ – месяц в диапазоне от 01 (январь) до 12 (декабрь), ДД – день месяца в диапазоне от 01 до 31.

Возвращаемая величина

Date

Описание

Функция **NumberToDate()** возвращает дату, величину типа Date, используя восьмизначное целое число. Например, следующая функция выдаст в результате 31 декабря 2014 года:

```
NumberToDate (20151231)
```

Пример

В следующем примере результат применения одной функции Date вычитается из другого. Получается количество дней между двумя датами. *i_elapsed* содержит количество дней, начиная с 1 января 2014.

```
Dim i_elapsed As Integer
i_elapsed = CurDate() - NumberToDate(20160101)
```

См. также:

[функции даты и времени](#), [функция StringToDate\(\)](#), [оператор Set Format](#), [функция Str\\$\(\)](#), [функция NumberToDateTime\(\)](#), [функция NumberToTime\(\)](#)

Функция NumberToDateTime()

Назначение

Возвращает значение Даты/времени (DateTime). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumberToDateTime( numeric_datetime )
```

numeric_datetime – семнадцатизначное целое число типа Integer в форме ГГГГММДДЧЧММССДДД (YYYYMMDDHHMMSSFFF). Например, 20140301214237582 представляет 1 Марта 2007 г. 21:42:37.582.

Возвращаемая величина

Дата/Время

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim fNum as float
dim Y as datetime
fNum = 20140301214237582
Y = NumberToDateTime (fNum)
Print FormatDate$(GetDate(Y))
Print FormatTime$(GetDate(Y), "hh:mm:ss.fff tt")
```

См. также:

функция FormatTime\$(), функция GetTime(), функция NumberToDate(), функция NumberToTime()

Функция NumberToTime()

Назначение

Возвращает значение Времени (Time). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumberToTime( numeric_time )
```

numeric_time – целое число из девяти цифр в форме HHMMSSFFF, где HH – час, MM – минуты, SS – секунды и FFF – десятичная доля. Например, 214237582 представляет 21:42:37.582 или 9:42:37.582 P.M.

Возвращаемая величина

Время

Пример

```
dim fNum as integer
dim Y as time
fNum = 214237582
Y = NumberToTime(fNum)
Print FormatTime$(Y, "hh:mm:ss.fff tt")
```

См. также:

[функции даты и времени](#), [функция FormatTime\\$\(\)](#), [функция GetTime\(\)](#), [функция NumberToDate\(\)](#), [функция NumberToDateTime\(\)](#)

Функция NumCols()

Назначение

Возвращает число колонок таблицы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumCols( table )
```

table – имя открытой таблицы;

Возвращаемая величина

Короткое целое – SmallInt

Описание

Функция **NumCols()** возвращает число колонок, из которых состоит открытая таблица.

В число возвращаемых функцией **NumCols()** колонок не входит специальная колонка "Object" (или "Obj" сокращенно), содержащая ссылки на графические объекты, присоединенные к таблице. Также в число колонок не включается другая специальная колонка RowID, содержащая номера строк таблицы.

Примечание: Если в таблице есть временные колонки (например, которые создает **оператор Add Column**), то число колонок, полученное от функции **NumCols()**, будет включать и временные колонки.

Ошибки

ERR_TABLE_NOT_FOUND (405), если не найдена данная таблица;

Пример

```
Dim i_counter As Integer
Open Table "world"
i_counter = NumCols(world)
```

См. также:

функция ColumnInfo(), функция NumTables(), функция TableInfo()

Функция NumTables()

Назначение

Возвращает число открытых на данный момент таблиц. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumTables ( )
```

Возвращаемая величина

Короткое целое – SmallInt

Описание

Функция **NumTables()** возвращает число открытых на данный момент таблиц.

Если в MapInfo открыта таблица, содержащая карту улиц (StreetInfo), то на самом деле открыты две связанные таблицы. Например, когда открываете таблицу DCWASHS (карта улиц Вашингтона), MapInfo Pro открывает две составляющие таблицы DCWASHS1.TAB и DCWASHS2.TAB. Тем не менее MapInfo Pro считает DCWASHS одной таблицей, поскольку составляющие таблицы являются частями одной карты. Так же и функция **NumTables()** таблицу, содержащую Карту улиц, будет считать одной открытой таблицей, несмотря на то, что практически она состоит из двух.

Пример

```
If NumTables() < 1 Then
  Note "You must open a table before continuing."
End If
```

См. также:

[оператор Open Table](#), [функция TableInfo\(\)](#), [функция ColumnInfo\(\)](#)

Функция NumWindows()

Назначение

Возвращает количество открытых на данный момент окон (**Карта**, **Список**, **График** и **Отчёт**).
Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
NumWindows ( )
```

Возвращаемая величина

Короткое целое – SmallInt

Описание

Функция **NumWindows()** возвращает число открытых на данный момент окон **Карта**, **Список**, **График** и **Отчёт**. Результат функции не зависит от того, в каком состоянии находится окно: свернуто в иконку или нет.

Чтобы определить общее количество выведенных на экран окон, включая вспомогательные (такие как окно Легенды, Информации), используется функция **NumAllWindows()**.

Пример

```
Dim num_open_wins As SmallInt
num_open_wins = NumWindows()
```

См. также:

[функция NumAllWindows\(\)](#), [функция WindowID\(\)](#)

Функция ObjectDistance()

Назначение

Возвращает расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectDistance( object1, object2, unit_name )
```

object1 и *object2* — выражения, указывающие на объекты.

unit_name — строка, определяющая единицы измерения расстояния.

Возвращаемая величина

Вещественное – Float

Описание

ObjectDistance() возвращает минимальное расстояние между объектами *object1* и *object2*, вычисленное в сферических координатах. Возвращаемое значение измеряется в единицах *unit_name*. Если вычисления не могут быть выполнены на сфере (то есть MapBasic Coordinate System — "Широта/Долгота"), то будет использован декартов метод расчёта расстояний на плоскости.

См. также:

[функция SphericalObjectDistance\(\)](#), [функция CartesianObjectDistance\(\)](#), [оператор Nearest](#), [оператор Farthest](#), [функция SphericalConnectObjects\(\)](#), [функция ConnectObjects\(\)](#), [функция CartesianConnectObjects\(\)](#)

Функция ObjectGeography()

Назначение

Возвращает информацию о графическом объекте, определяющую его расположение. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectGeography( object, attribute )
```

object – это выражение для объекта.

attribute – целое, задающее тип возвращаемой информации.

Возвращаемая величина

Вещественное – Float

Описание

Параметр *attribute* должен принимать значения целочисленного кода, управляющего типом возвращаемой функцией информации. В следующей таблице в первой колонке приводятся имена кодов параметра *attribute* функции ObjectGeography(), которые заданы в файле стандартных определений MapBasic MAPBASIC.DEF.

Некоторые значения параметра *attribute* могут адресоваться объектам определенного типа. Например, начальный и конечный угол может быть только у объекта типа "дуга", а текстовые объекты – единственные с атрибутом наклона шрифта. Если объект не поддерживает z- или m-значения или z- или m-значения для данного узла не определены, будет возвращена ошибка.

Параметры	ID	Возвращаемое значение (вещественное – Float)
OBJ_GEO_MINX	1	Минимальная X-координата минимального прямоугольного покрытия объекта, если объект не является линией. Иначе возвратится значение, равное OBJ_GEO_LINEBEGX.
OBJ_GEO_MINY	2	Минимальная Y-координата минимального прямоугольного покрытия объекта, если объект не является линией. Иначе возвратится значение, равное OBJ_GEO_LINEBEGY.

Параметры	ID	Возвращаемое значение (вещественное – Float)
OBJ_GEO_MAXX	3	Максимальная X-координата объекта или его минимального прямоугольного покрытия. Код не применим для объекта типа "точка". Иначе возвратится значение, равное OBJ_GEO_LINEENDX.
OBJ_GEO_MAXY	4	Максимальная Y-координата объекта или его минимального прямоугольного покрытия. Код не применим для объекта типа "точка". Иначе возвратится значение, равное OBJ_GEO_LINEENDY.
OBJ_GEO_ARCBEGANGLE	5	Начальный угол дуги.
OBJ_GEO_ARCENDANGLE	6	Конечный угол дуги.
OBJ_GEO_LINEBEGX	1	X-координата начальной точки прямой. Только для объекта типа "линия".
OBJ_GEO_LINEBEGY	2	Y-координата начальной точки линии. Только для объекта типа "линия".
OBJ_GEO_LINEENDX	3	X-координата конечной точки линии. Только для объекта типа "линия".
OBJ_GEO_LINEENDY	4	Y-координата конечной точки линии. Только для объекта типа "линия".
OBJ_GEO_POINTX	1	X-координата точечного объекта.
OBJ_GEO_POINTY	2	Y-координата точечного объекта.
OBJ_GEO_POINTZ	8	Значение Z объекта "точка".
OBJ_GEO_POINTM	9	Значение M объекта "точка".
OBJ_GEO_ROUNDRAIUS	5	Диаметр окружности, которую можно вписать в закругление угла объекта типа "скругленный прямоугольник". Результат выдается в текущих координатных единицах (например, в градусах).
OBJ_GEO_CENTROID	5	Возвращает точечный объект центроида областей, коллекций, групп точек и полилиний. Обычно используется оператор Alter Object .

Параметры	ID	Возвращаемое значение (вещественное – Float)
OBJ_GEO_TEXTLINEX	5	Координата по оси X конца текстовой строки объекта.
OBJ_GEO_TEXTLINEY	6	Координата по оси Y конца текстовой строки объекта.
OBJ_GEO_TEXTANGLE	7	Угол поворота текстового объекта.

Функция **ObjectGeography()** поддерживает группы точек и коллекции. Для обоих этих типов поддерживаются атрибуты 1 - 4 (координаты минимального описывающего объект прямоугольника (МОП)).

OBJ_GEO_MINX	1	Минимальная X-координата описывающего прямоугольника.
OBJ_GEO_MINY	2	Максимальная Y-координата описывающего прямоугольника.
OBJ_GEO_MAXX	3	Максимальная X-координата описывающего прямоугольника.
OBJ_GEO_MAXY	4	Максимальная Y-координата описывающего прямоугольника.

Пример

Считывание координат начальной точки линии из таблицы City. Далее используется **оператор Set Map** для перемещения центра просмотра карты в начальную точку прямой линии.

```

Include "MAPBASIC.DEF"
Dim i_obj_type As Integer, f_x, f_y As Float
Open Table "city"
Map From city
Fetch First From city
' at this point, the expression:
' city.obj
' represents the graphical object that's attached
' to the first record of the CITY table.
i_obj_type = ObjectInfo(city.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_LINE Then
    f_x = ObjectGeography(city.obj, OBJ_GEO_LINEBEGX)
    f_y = ObjectGeography(city.obj, OBJ_GEO_LINEBEGY)
    Set Map Center (f_x, f_y)
End If

```

См. также:

функция Centroid(), функция CentroidX(), функция CentroidY(), функция ObjectInfo()

Функция ObjectInfo()

Назначение

Возвращает стиль линии, штриха и другие величины, описывающие графический объект, а также его тип. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectInfo( object, attribute )
```

object — это выражение для объекта.

attribute — целое, задающее тип возвращаемой информации.

Возвращаемая величина

Величины типа SmallInt, Integer, String, Float, Pen, Brush, Symbol или Font, в зависимости от параметра attribute.

OBJ_INFO_NPOLYGONS (21) — целое число полигонов в объекте типа "область" или число ломаных компонент в объекте типа "полилиния".

OBJ_INFO_NPOLYGONS+N (21) — целое число узлов в N-ом полигоне "области" или в N-ой ломаной "полилинии".

Примечание: В случае объектов-регионов, MapInfo Pro считает начальный узел дважды (один раз как начальный и второй раз как конечный). Так, для треугольной области функция **ObjectInfo()** возвращает 4.

Описание

Функция **ObjectInfo()** возвращает информацию об одной характеристике графического объекта за одно обращение. Первым параметром должен быть определитель объекта (например, имя переменной объекта, или выражение описывающее объекты таблицы вида *tablename.obj*).

Каждому объекту присвоено несколько атрибутов. Например, каждый объект имеет атрибут типа — "type", с помощью которого идентифицируется является ли объект точкой, линией или областью и т.п. Большинство типов объектов имеет атрибуты Pen и/или Brush, которые определяют внешний вид объектов. Функция **ObjectInfo()** возвращает только один атрибут заданного объекта. Возвращаемый атрибут зависит от использованного параметра атрибута. Поэтому, если требуется определить несколько характеристик объекта, то функцию **ObjectInfo()** следует вызывать несколько раз с разными параметрами в каждом из вызовов.

В таблице ниже перечислены разные атрибуты и возвращаемые значения для каждого из них.

Код атрибута	ID	Возвращаемая величина
OBJ_INFO_TYPE	1	Короткое целое, представляющее тип объекта; возвращаемые значения перечислены в таблице ниже (например, OBJ_TYPE_LINE). Атрибутом из DEF-файла является 1 (ObjectInfo(Object, 1)).
OBJ_INFO_PEN	2	Величина типа Pen. Стил линии для объектов типа "дуга", "эллипс", "линия", "полилиния", "рамка", "область", "прямоугольник" и "скругленный прямоугольник".
OBJ_INFO_BRUSH	3	Величина типа Brush. Стил штриховки объектов типа "эллипс", "рамка", "область", "прямоугольник" и "скругленный прямоугольник".
OBJ_INFO_TEXTFONT	2	Возвращается стиль шрифта — Font; этот запрос используется только текстовых объектов. Примечание: Если текстовый объект принадлежит таблице (а не отчету), то размер шрифта равен нулю, а размер шрифта динамически определяется MapInfo в зависимости от размера окна карты.
OBJ_INFO_SYMBOL	2	Величина типа Symbol. Стил символа точечного объекта.
OBJ_INFO_NPNTS	20	Целое число узлов в полилинии или в многоугольнике области.
OBJ_INFO_SMOOTH	4	Логический признак сглаженности объекта типа "полилиния".
OBJ_INFO_FRAMEWIN	4	Целое число, идентификатор окна, вставленного в рамку Отчета.
OBJ_INFO_FRAMETITLE	6	Строка с заголовком рамки.
OBJ_INFO_NPOLYGONS	21	Короткое целое, число полигонов в объекте типа "область" или число ломаных компонент в объекте типа "полилиния".

Код атрибута	ID	Возвращаемая величина
OBJ_INFO_NPOLYGONS+N	21	Целое число узлов <i>N</i>-ого полигона области или <i>N</i>-ого сегмента полилинии. Примечание: В случае объектов-регионов, MapInfo Pro считает начальный узел дважды (один раз как начальный и второй раз как конечный). Так, для треугольной области функция ObjectInfo() возвращает 4.
OBJ_INFO_TEXTSTRING	3	Величина типа String. Текстовое содержимое объекта типа "текст". Если объект состоит из нескольких строк, то результат будет включать символ конца строки (Chr\$(10) Chr\$(10)).
OBJ_INFO_TEXTSPACING	4	Число 1, 1.5 или 2, определяющее интерлиньяж в текстовом объекте.
OBJ_INFO_TEXTJUSTIFY	5	Число типа SmallInt, определяющее выравнивание текста: 0 = по левому краю, 1 = по центру, 2 = по правому краю.
OBJ_INFO_TEXTARROW	6	Число типа SmallInt, определяющее стиль указки в текстовом объекте: 0 = нет указки, 1 = просто линия, 2 = стрелка.
OBJ_INFO_FILLFRAME	7	Логическое: TRUE, если объект типа "рамка" показывает карту и для него установлен режим "Заполнить Рамку Картой".
OBJ_INFO_NONEMPTY	11	Логическая величина, TRUE, если коллекция имеет узлы, FALSE, если объект пустой.
OBJ_INFO_REGION	8	Возвращает значение соответствующее полигонам, входящим в коллекцию. Если коллекция не содержит полигонов, будет возвращен пустой полигон. Этот запрос действует только для объектов типа "коллекция".
OBJ_INFO_PLINE	9	Возвращает значение соответствующее полилиниям, входящим в коллекцию. Если коллекция не содержит полилиний, будет возвращена пустая полилиния. Этот запрос действует только для объектов типа "коллекция".
OBJ_INFO_MPOINT	10	Возвращает значение соответствующее группам точек, входящим в коллекцию. Если коллекция не содержит групп точек, будет возвращена пустая группа точек. Этот запрос действует только для объектов типа "коллекция".

Код атрибута	ID	Возвращаемая величина
OBJ_INFO_Z_UNIT_SET	12	логическое значение, показывает, что единицы измерения Z-координат заданы.
OBJ_INFO_Z_UNIT	13	Строка, указывающая единицы расстояния для Z-значений. Пустая строка будет возвращена, если единицы измерения Z-координат не заданы.
OBJ_INFO_HAS_Z	14	Logical, логическая величина, показывает, что объект имеет Z-координаты.
OBJ_INFO_HAS_M	15	логическая величина, показывает, что объект имеет m-значения.

Коды в левой колонке (например, OBJ_INFO_TYPE) задаются в файле определителей MapBasic MAPBASIC.DEF. Программа должна включать ссылку на "MAPBASIC.DEF", если предполагается вызывать функцию **ObjectInfo()**.

Каждый графический атрибут может применяться только для определенных типов графических объектов. Например, точечные объекты являются единственными с атрибутами Symbol, а текстовые объекты — с атрибутами Font. Поэтому функция **ObjectInfo()** не может возвращать любой атрибут для конкретного типа объекта.

Если используется код OBJ_INFO_TYPE как значение параметра attribute, то функция **ObjectInfo()** вернет код, соответствующий типу графического объекта. В следующей таблице приведены имена этих кодов из файла стандартных определений.

Таблица 5. Значения OBJ_INFO_TYPE

Значения OBJ_INFO_TYPE	ID	Соответствующий тип
OBJ_TYPE_ARC	1	Объект дуга
OBJ_TYPE_ELLIPSE	2	Объекты эллипс / окружность
OBJ_TYPE_LINE	3	Объект прямая линия
OBJ_TYPE_PLINE	4	Объект полилиния
OBJ_TYPE_POINT	5	Объект точка
OBJ_TYPE_FRAME	6	Объект рамка в окне Отчета

Значения OBJ_INFO_TYPE	ID	Соответствующий тип
OBJ_TYPE_REGION	7	Объект полигон
OBJ_TYPE_RECT	8	Объект прямоугольник
OBJ_TYPE_ROUNDRECT	9	Объект скругленный прямоугольник
OBJ_TYPE_TEXT	10	Текстовый объект
OBJ_TYPE_MPOINT	11	Объект группа точек
OBJ_TYPE_COLLECTION	12	Коллекция объектов, например, точек, полилиний или областей

Условия появления сообщения об ошибке

Сообщение об ошибке "... не удалось выбрать объект из текущей записи (650)" появляется, когда выражение для объекта не имеет объекта. Для того чтобы обработать эту ошибку, либо используйте оператор `OnError GoTo`, либо старайтесь предотвратить ошибку, проверив правильное выражение объекта с помощью теста, такого как:

```
If Str$(myObject) <> "" Then
    ' the object expression does contain an object
    ' so it is safe to call ObjectInfo
End If
```

Пример

```
Include "MAPBASIC.DEF"
Dim counter, obj_type As Integer
Open Table "city"
Fetch First From city
    ' at this point, the expression: city.obj
    ' represents the graphical object that's attached
    ' to the first record of the CITY table.
obj_type = ObjectInfo(city.obj, OBJ_INFO_TYPE)
Do Case obj_type
Case OBJ_TYPE_LINE
    Note "First object is a line."
Case OBJ_TYPE_PLINE
    Note "First object is a polyline..."
    counter = ObjectInfo(city.obj, OBJ_INFO_NPNTS)
    Note "... with " + Str$(counter) + " nodes."
Case OBJ_TYPE_REGION
```

```
Note "First object is a region..."
counter = ObjectInfo(city.obj, OBJ_INFO_NPOLYGONS)
Note ", made up of " + Str$(counter) + " polygons..."
counter = ObjectInfo(city.obj, OBJ_INFO_NPOLYGONS+1)
Note "The 1st polygon has" + Str$(counter) + " nodes"
End Case
```

См. также:

[оператор Alter Object](#), [предложение Brush](#), [предложение Font](#), [функция ObjectGeography\(\)](#), [предложение Pen](#), [предложение Symbol](#)

Функция ObjectLen()

Назначение

Возвращает географическую протяженность объекта линии или полилинии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectLen( expr, unit_name )
```

expr – это выражение для объекта.

unit_name – это строка, представляющая единицы измерения расстояния (например, "km").

Возвращаемая величина

Вещественное – Float

Описание

Функция **ObjectLen()** возвращает длину выражения объекта. Помните, что ненулевые длины имеют только объекты линий и полилиний, для измерения периметра прямоугольника, эллипса или полигона, используется [функция Perimeter\(\)](#).

Возвращаемое функцией **ObjectLen()** значение длины периметра измеряется в единицах длины, определенных параметром *unit_name*; например, для получения длины в милях, укажите "mi" в качестве *unit_name*. Список допустимых единиц измерения смотрите в разделе: [Оператор Set Distance Units](#).

В большинстве случаев MapInfo Pro проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система – план, то выполняются декартовы вычисления.

Пример

```
Dim geogr_length As Float
Open Table "streets"
Fetch First From streets
geogr_length = ObjectLen(streets.obj, "mi")
```

geogr_length теперь представляет длину сегмента улицы в милях.

См. также:

[функция Distance\(\)](#), [функция Perimeter\(\)](#), [функция Set Distance Units](#)

Функция ObjectNodeHasM()

Назначение

Возвращает значение TRUE, если определённый узел в полигоне, полилинии или объекте группа точек имеет значение m. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectNodeHasM( object, polygon_num, node_num )
```

object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Логическое

Описание

Функция **ObjectNodeHasM()** возвращает значение TRUE, если определённый узел в полигоне, полилинии или объекте группа точек имеет значение m.

Параметр *polygon_num* должен иметь значение от единицы и выше. Он указывает, какой полигон (если запрос сделан к региону) или какая секция (если запрос к полилинии) будут опрошены. [Функция ObjectInfo\(\)](#) вызывается для определения числа полигонов или сегментов

объекта. Функция **ObjectNodeHasM()** поддерживает объекты типа "группа точек" и возвращает значение TRUE, если определённая точка в объекте "группа точек" имеет значение m.

Параметр *node_num* должен иметь значение от единицы и выше; он указывает MapBasic, какой из узлов объекта будет опрошен. **Функция ObjectInfo()** может быть использована для определения числа узлов в объекте.

Если объект не поддерживает значение m или значение m для данного узла не определено, возвращается FALSE.

Пример

Опрос первого графического элемента в таблице ROUTES. Если первый объект является полилинией, программа определит, что первый узел содержит z-координату или m-значение, и запросит z-координату и m-значение первого узла полилинии.

```
Dim i_obj_type As SmallInt,
    z, m As Float
hasZ, hasM as Logical
Open Table "routes"
Fetch First From routes
' at this point, the expression:
' routes.obj
' represents the graphical object that's attached
' to the first record of the routes table.
i_obj_type = ObjectInfo(routes.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_PLINE Then
' ... then the object is a polyline...
If (ObjectNodeHasZ(routes.obj, 1, 1)) Then
    z = ObjectNodeZ(routes.obj, 1, 1) ' read z-coordinate
End If
If (ObjectNodeHasM(routes.obj, 1, 1)) Then
    m = ObjectNodeM(routes.obj, 1, 1) ' read m-value
End If
End If
```

См. также:

[информация об объектах карты, функция ObjectInfo\(\)](#)

Функция ObjectNodeHasZ()

Назначение

Возвращает значение TRUE, если определённый узел в полигоне, полилинии или объекте группа точек имеет z-координату. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectNodeHasZ( object, polygon_num, node_num )
```

object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Логическое

Описание

Функция **ObjectNodeHasZ()** возвращает значение TRUE, если заданный узел полигона, полилинии или группы точек имеет z-координату. Параметр *polygon_num* должен иметь значение от единицы и выше. Он указывает, какой полигон (если запрос сделан к региону) или какая секция (если запрос к полилинии) будут опрошены. **Функция ObjectInfo()** вызывается для определения числа полигонов или сегментов объекта. Функция **ObjectNodeHasZ()** поддерживает объекты типа "группа точек" и возвращает значение TRUE, если заданный узел объекта "группа точек" имеет z-координату.

Параметр *node_num* должен иметь значение от единицы и выше; он указывает MapBasic, какой из узлов объекта будет опрошен. **Функция ObjectInfo()** может быть использована для определения числа узлов в объекте.

Если *object* не поддерживает z-координаты или z-координата для этого узла не определена, то будет возвращено значение FALSE.

Пример

Опрос первого графического элемента в таблице ROUTES. Если первый объект является полилинией, программа определит, что первый узел содержит z-координату или m-значение, и запросит z-координату и m-значение первого узла полилинии.

```
Dim i_obj_type As SmallInt,
  z, m As Float
  hasZ, hasM as Logical
Open Table "routes"
Fetch First From routes
' at this point, the expression:
' routes.obj
' represents the graphical object that's attached
' to the first record of the routes table.
```

```

i_obj_type = ObjectInfo(routes.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_PLINE Then
  ' ... then the object is a polyline...
  If (ObjectNodeHasZ(routes.obj, 1, 1)) Then
    z = ObjectNodeZ(routes.obj, 1, 1) ' read z-coordinate
  End If
  If (ObjectNodeHasM(routes.obj, 1, 1)) Then
    m = ObjectNodeM(routes.obj, 1, 1) ' read m-value
  End If
End If

```

См. также:

[информация об объектах карты, функция ObjectInfo\(\)](#)

Функция ObjectNodeM()

Назначение

Возвращает m-значение заданного узла полигона, полилинии или группы точек. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectNodeM( object, polygon_num, node_num )
```

object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Вещественное – Float

Описание

Функция **ObjectNodeM()** возвращает m-значения указанного узла в регионе, полилинии или группе точек.

Параметр *polygon_num* должен иметь значение от единицы и выше. Он указывает, какой полигон (если запрос сделан к региону) или какая секция (если запрос к полилинии) будут

опрошены. **Функция ObjectInfo()** вызывается для определения числа полигонов или сегментов объекта. Функция **ObjectNodeM()** поддерживает объекты "группа точек" и возвращает m-величину заданного узла группы точек.

Параметр *node_num* должен иметь значение от единицы и выше; он указывает MapBasic, какой из узлов объекта будет опрошен. **Функция ObjectInfo()** может быть использована для определения числа узлов в объекте.

Если объект не поддерживает m-значения или m-значения для данного узла не определены, будет возвращена ошибка.

Пример

Опрос первого графического элемента в таблице ROUTES. Если первый объект это полилиния, то программа опросит z-координаты и m-значения первого узла полилинии.

```
Dim i_obj_type As SmallInt,
z, m As Float
Open Table "routes"
Fetch First From routes
' at this point, the expression:
' routes.obj
' represents the graphical object that's attached
' to the first record of the routes table.
i_obj_type = ObjectInfo(routes.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_PLINE Then
' ... then the object is a polyline...
z = ObjectNodeZ(routes.obj, 1, 1) ' read z-coordinate
m = ObjectNodeM(routes.obj, 1, 1) ' read m-value
End If
```

См. также:

[информация об объектах карты, функция ObjectInfo\(\)](#)

Функция ObjectNodeX()

Назначение

Возвращает X-координату заданного узла полигона или полилинии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectNodeX( object, polygon_num, node_num )
```


object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Вещественное – Float

Описание

Функция **ObjectNodeX()** возвращает X-координату заданного узла в полигоне или полилинии. Похожая **функция ObjectNodeY()** возвращает Y-координату.

Параметр *polygon_num* должен иметь значение от единицы и выше. Он указывает, какой полигон (если запрос сделан к региону) или какая секция (если запрос к полилинии) будут опрошены. **Функция ObjectInfo()** вызывается для определения числа полигонов или сегментов объекта. Функция **ObjectNodeX()** поддерживает объекты "группа точек" и возвращает X-координату заданного узла группы точек.

Параметр *node_num* должен иметь значение от единицы и выше; он указывает MapBasic, какой из узлов объекта будет опрошен. **Функция ObjectInfo()** может быть использована для определения числа узлов в объекте. Координата, которую будет получена в результате применения функции **ObjectNodeX()**, будет выведена в системе координат, которая определена как текущая для MapBasic. По умолчанию в MapBasic используется проекция "Широта/Долгота". Подробнее о проекциях и координатных системах смотрите в разделе: **Оператор Set CoordSys**.

Пример

Опрос первого графического элемента в таблице ROUTES. Если первый объект полилиния, то считываются координаты первого узла. На этом месте создается точечный объект.

```
Dim i_obj_type As SmallInt, x, y As Float, new_pnt As Object
Open Table "routes"
Fetch First From routes
' at this point, the expression:
' routes.obj
' represents the graphical object that's attached
' to the first record of the routes table.
i_obj_type = ObjectInfo(routes.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_PLINE Then
' ... then the object is a polyline...
x = ObjectNodeX(routes.obj, 1, 1) ' read longitude
y = ObjectNodeY(routes.obj, 1, 1) ' read latitude
Create Point Into Variable new_pnt (x, y)
```

```
Insert Into routes (obj) Values (new_pnt)
End If
```

См. также:

оператор **Alter Object**, функция **ObjectGeography()**, функция **ObjectInfo()**, функция **ObjectNodeY()**, оператор **Set CoordSys**

Функция ObjectNodeY()

Назначение

Возвращает Y-координату заданного узла полигона или полилинии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ObjectNodeY( object, polygon_num, node_num )
```

object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Вещественное – Float

Описание

Функция **ObjectNodeY()** возвращает Y-координату заданного узла в полигоне или полилинии. Подробнее см. [Функция ObjectNodeX\(\)](#).

Пример

Смотрите: [функция ObjectNodeX\(\)](#).

См. также:

оператор **Alter Object**, функция **ObjectGeography()**, функция **ObjectInfo()**, оператор **Set CoordSys**

Функция ObjectNodeZ()

Назначение

Возвращает z-координату указанного узла полигона, полилинии или объекта "группа точек".

Синтаксис

```
ObjectNodeZ( object, polygon_num, node_num )
```

object – это выражение для объекта.

polygon_num – целое, положительное, определяющее, какой полигон или сегмент опрашиваются. Оно игнорируется для объектов типа "группа точек" (используется только для регионов и полилиний).

node_num – положительное целое число, показывающее какой узел считывается.

Возвращаемая величина

Вещественное – Float

Описание

Функция **ObjectNodeZ()** возвращает z-координату заданного узла полигона, полилинии или группы точек.

Параметр *polygon_num* должен иметь значение от единицы и выше. Он указывает, какой полигон (если запрос сделан к региону) или какая секция (если запрос к полилинии) будут опрошены. [Функция ObjectInfo\(\)](#) вызывается для определения числа полигонов или сегментов объекта. Функция **ObjectNodeZ()** поддерживает объекты "группа точек" и возвращает z-координату заданного узла группы точек.

Параметр *node_num* должен иметь значение от единицы и выше; он указывает MapBasic, какой из узлов объекта будет опрошен. [Функция ObjectInfo\(\)](#) может быть использована для определения числа узлов в объекте.

Если объект не поддерживает z-значения или z-значения для данного узла не определены, будет возвращена ошибка.

Пример

Опрос первого графического элемента в таблице ROUTES. Если первый объект это полилиния, то программа опросит z-координаты и m-значения первого узла полилинии.

```
Dim i_obj_type As SmallInt,
  z, m As Float
Open Table "routes"
Fetch First From routes
' at this point, the expression:
' routes.obj
' represents the graphical object that's attached
' to the first record of the routes table.
i_obj_type = ObjectInfo(routes.obj, OBJ_INFO_TYPE)
If i_obj_type = OBJ_PLINE Then
' ... then the object is a polyline...
z = ObjectNodeZ(routes.obj, 1, 1) ' read z-coordinate
m = ObjectNodeM(routes.obj, 1, 1) ' read m-value
End If
```

См. также:

[информация об объектах карты](#), [функция ObjectInfo\(\)](#)

Оператор Objects Check

Назначение

Проверяет таблицы на предмет некорректности данных, что может вызвать проблемы и ошибочные результаты применения различных операций. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Check From tablename Into Table tablename
[ SelfInt [ Symbol Clause ] ]
[ Overlap [ Pen Clause ] [ Brush Clause ] ]
[ Gap areavalue [ Units Units ] [ Pen Clause ] [ Brush Clause ] ] ]
```

tablename – строка с именем открытой таблицы.

Clause – выражение.

Units – единицы измерения площадей.

areavalue – значение площади, при превышении которого области не обрабатываются алгоритмом как пустоты.

Описание

Objects Check проверяет таблицу, указанную в предложении **From** на наличие некорректных данных, которые могут вызвать проблемы при выполнении различных операций. Будут проверяться только полигональные объекты. Области будут проверены на наличие самопересечений, взаимных пересечений и пустот (брешей).

Самопересечения могут вызвать ошибки при выполнении многих вычислений, в том числе и расчета площади области. Они также могут привести к некорректным результатам при выполнении различных операций над объектами, таких как комбинация, создание буферных зон, вырезание и деление.

Для обозначения самопересечений создаются точечные объекты, которые помещаются в выходную таблицу. Выходная таблица может быть определена в предложении **Into Table**. Если предложение **Into Table** не указано, выходные данные размещаются во входной таблице.

Если используется режим **Selfint**, то таблица будет проверена на самопересечения. Стиль создаваемых точечных объектов определяется через параметр **Symbol**. По умолчанию, это красная булавка размером 28-точек.

Большинство таблиц с областями представляют границы. Файлы `STATES.TAB` и `WORLD.TAB` содержат примеры объектов-полигонов с границами. В таких таблицах границы не должны перекрываться (например, штат Utah не должен перекрывать Wyoming). В режиме **Overlap** таблицы будут проверяться на наличие перекрытия регионов. В выходной таблице будут созданы регионы, представляющие собой области перекрытия. Стиль регионов-перекрытий определяется через **предложение Brush** (заливка) и **предложение Pen** (границы). По умолчанию, для этих регионов предусмотрена сплошная желтая заливка и тонкая черная граница.

Пустоты (бреши) – это замкнутые области, которые не образуют полигонов. В таблице с границами полигоны должны иметь общие границы. В идеальном случае в таблице не должны встречаться перекрытия полигонов и пустоты между ними. В некоторых случаях пустоты между полигонами имеют смысл и право на существование. Например, Великие озера на карте Мира являются "пустотой" между Канадой и США. Однако, в большинстве случаев пустоты являются результатом плохого согласования общих границ между полигонами. Такие бреши обычно имеют малые размеры.

Чтобы отделить допустимые пустоты (например, Великие озера) от заведомо ненужных брешей, используется предложение **Gap areavalue**. Любые пустоты больше заданной площади будут оставаться без изменения. Единицы измерения площади **Gap areavalue** задаются предложением **Units**. Если подпредложение **Unit** не задано, то площадь брешей **Gap areavalue** будет измеряться в текущих единицах измерения MapBasic.

Бреши оформляются стилями, которые задают **предложение Pen** и **предложение Brush**, следующие за словом **Gap**. По умолчанию, эти полигоны изображаются с синей заливкой и тонкой черной границей.

Пример

В этом примере функция **Objects Check** используется для проверки областей таблицы TestFile, результаты сохраняются в таблице DumpFile. Кроме того, с помощью ключевого слова **Overlap** выполняется проверка на наличие перекрытий, изменены стандартные стили для точек (самопересечения) и полигонов (перекрытия). В этом примере значение критической площади для предложения **Gap** равно 100000.

```
objects check from TestFile into table Dumpfile
Selfint Symbol (67,16711680,28)
Overlap Pen (1,2,0) Brush (2,16776960,0)
100000 Units "sq mi" Pen (1,2,0) Brush (2,255,0)
```

См. также:

[функция OverlayNodes\(\)](#), [оператор Objects Enclose](#)

Оператор Objects Clean

Назначение

Корректирует топологию объектов из данной таблицы, дополнительно может удалять перекрытия и закрывать бреши между полигонами. Таблица может быть таблицей выборки (Selection). Все объекты, подлежащие коррекции, должны быть замкнутыми (полигоны, прямоугольники, скругленные прямоугольники или эллипсы). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Clean From tablename
[ Overlap ]
[ Gap Area [ Unit Units ] ]
```

tablename – строка с именем открытой таблицы.

Units – единицы измерения площадей.

Описание

Объекты из исходной таблицы *tablename* проверяются на предмет топологической корректности, например, самопересечений, наложений и пустот. Самопересекающиеся полигоны в форме "восьмерки" будут превращены в два полигона с общей вершиной. Полигоны, содержащие острые выступы подвергаются обработке, при которой часть таких выступов удаляется. Исправленный объект заменит исходный.

Если включено ключевое слово **Overlap**, то области наложения полигонов друг на друга удаляются из полигонов. Часть перекрытия будет удалена из всех перекрывающихся полигонов, кроме того, у которого наибольшая площадь.

Примечание: **Objects Clean** удаляет перекрытия, когда один объект полностью попадает в другой. Обратите внимание, что это исключение из правила "крупнейший объект побеждает". Если объект находится полностью внутри другого объекта, то меньший объект сохраняется, а в большом объекте образуется дырка, таким образом, Результат не содержит никаких перекрытий.

Пустоты (бреши) – это замкнутые области, которые не образуют полигонов. В таблице с границами полигоны должны иметь общие границы. В идеальном случае в таблице не должны встречаться перекрытия полигонов и пустоты между ними. В некоторых случаях пустоты между полигонами имеют смысл и право на существование. Например, Великие озера на карте Мира являются "пустотой" между Канадой и США. Однако, в большинстве случаев пустоты являются результатом плохого согласования общих границ между полигонами. Такие бреши обычно имеют малые размеры.

Чтобы успешно разделить допустимые пустоты (например, Великие озера) от заведомо ненужных брешей, используется предложение **Gap Area**. Любые пустоты больше заданной площади будут оставляться без изменения. Единицы измерения площади пустот в предложении **Gap Area** задаются предложением **Units**. Если подпредложение **Unit** не задано, то **Gap Area** измеряет площадь брешей в текущих единицах MapBasic. Пустоты будут удалены путем их объединения с соседним полигоном, причем именно тем, у которого площадь больше. Чтобы задать подходящую величину площади пустот, используйте **оператор Objects Check**. Любая пустота, которую отметит **оператор Objects Check**, будет удалена **оператором Objects Clean**, аналогично команде "Коррекция топологии" из MapInfo Pro.

Пример

```
Open Table "STATES.TAB" Interactive
Map From STATES
Set Map Layer 1 Editable On
select * from STATES
Objects Clean From Selection Overlap Gap 10 Units "sq m"
```

См. также:

оператор Create Object, **функция OverlayNodes()**, **оператор Objects Disaggregate**, **оператор Objects Check**

Оператор Objects Combine

Назначение

Позволяет объединять объекты таблицы; соответствует команде MapInfo Pro **Объединить** на вкладке **ОБЪЕКТЫ**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Combine
[ Into Target ]
[ Data column = expression [ , column = expression ... ] ]
[ DropIndex ( Auto | On | Off ) ]
```

column – строка с именем колонки изменяемой таблицы.

expression – выражение для расчета значений *column*.

Описание

Оператор **Objects Combine** создает объект, представляющий собой географическое объединение выбранных объектов. С помощью оператора **Objects Combine** можно также обобщать данные, вычислять суммы и средние значения величин из записей, к которым присоединены объекты.

Оператор **Objects Combine** соответствует команде MapInfo Pro **Объединить** на вкладке **ОБЪЕКТЫ**. Описание выполняющейся операции смотрите в описании команды **Объединить** в *Руководстве пользователя MapInfo Pro*. Если в MapInfo Pro выполнить команду **Объединить** на вкладке **ОБЪЕКТЫ** и при этом будет открыто окно **MapBasic**, то в протоколе выполненных действий будет использован оператор **Objects Combine**. Объекты, участвующие в этой операции, должны быть или все замкнутыми (т.е. регионы, прямоугольники, скруглённые прямоугольники или эллипсы) или все линейными (т.е. линии, полилинии или дугу). Невозможно объединять замкнутые и линейные объекты или точечные и текстовые.

Дополнительный параметр **Into Target** допускается, если задан изменяемый объект (либо пользователем, либо используя [оператор Set Target](#)), и этот изменяемый объект состоит из единственной фигуры. Если в операторе есть предложение **Into Target**, то MapInfo Pro будет комбинировать изменяемый объект с выбранными объектами на карте. Объект, полученный в результате комбинирования, заменит изменяемый объект в таблице.

Если выбранные объекты, участвующие в комбинировании в режиме **Into Target**, находятся в той же таблице, что и изменяемый объект, то MapInfo Pro удалит записи, к которым присоединены выбранные объекты.

Если используется предложение **Into Target** и выбранные объекты принадлежат одной таблице, а изменяемый объект – другой, то MapInfo Pro не удаляет выбранные объекты. Если опустить параметр **Into Target**, MapInfo Pro объединит выбранные объекты, не трогая изменяемый (если такой изменяемый объект существует). Выбранные объекты и соответствующие им строки в таблице удаляются, новая строка с объектом, полученным в результате комбинирования, добавляется в конец таблицы.

Предложение **Data** управляет обобщением данных. (Информация об обобщении данных приводится в описании команды **Объединить** в *Руководстве пользователя MapInfo Pro*.) За ключевым словом **Data** должен следовать список определений через запятую. Каждое определение является выражением, по которому будет вычислено (или изменено) значение в определенной колонке для записи, которая будет получена в результате выполнения оператора Objects Combine.

Вычисления или изменения должны производиться в соответствии с типом колонки (числовым, строковым и т. п.)

Выражение	Описание
<code>col_name = col_name</code>	содержимое колонки не меняется.
<code>col_name = value</code>	MapBasic помещает значение value в поле записи объекта, полученного в результате.
<code>col_name = Sum(col_name)</code>	Используется только для числовых колонок. MapBasic помещает сумму значений колонки col_name из всех записей объектов, участвующих в комбинации, в поле записи объекта, полученного в результате.
<code>col_name = Avg(col_name)</code>	Используется только для числовых колонок. MapBasic помещает среднее из значений колонки col_name из всех записей объектов, участвующих в комбинации, в поле записи объекта, полученного в результате.
<code>col_name = WtAvg(colname, wtcolname)</code>	Используется только для числовых колонок. MapInfo Pro помещает взвешенное среднее число из значений колонки col_name из всех записей объектов, участвующих в комбинации, в поле записи объекта, полученного в результате. В качестве коэффициентов веса используются значения из колонки wtcolname.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если в списке **Data** определены не все колонки, то MapBasic разместит пустые значения в

неописанные поля новой записи. Если предложения **Data** нет в операторе, но используется предложение **Into Target**, то MapInfo Pro сохранит значения изменяемого объекта в записи.

Если в операторе не используется ни предложение **Data**, ни предложение **Into Target**, то результирующий объект будет помещен в новую строку и MapInfo Pro разместит нулевые и пустые значения в поля этой записи.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

См. также:

[функция Combine\(\)](#), [оператор Set Combine Version](#), [оператор Set Target](#)

Оператор Objects Disaggregate

Назначение

Разъединяет объект на составляющие его компоненты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Disaggregate [ Into Table name ]
[ All | Collection ]
[ Data column_name = expression [ , column_name = expression ... ]
[ DropIndex ( Auto | On | Off ) ]
```

name – строка с именем таблицы, в которой будут храниться новые разобщенные объекты.

column_name – строка с именем колонки изменяемой таблицы.

expression – выражение, которым задаётся то, что будет записано в колонку *column_name*.

Описание

Если объект включает в себя несколько других объектов, то для каждого объекта, входящего в состав основного объекта, в итоговой таблице создается новый объект.

По умолчанию, каждый составной объект будет разделен на элементарные одиночные объекты. Регион будет разделен на некоторое число полигонов, в зависимости от того, установлен ли режим **All**. Если режим **All** установлен, то будет создано множество отдельных одиночных объектов-полигонов. Для островов (внутренние границы) будут созданы отдельные объекты-полигоны. Если режим **All** отключен, то в результирующих объектах острова сохраняются. Например, если исходный регион содержит 3 полигона и один из них является островом в другом полигоне, то в результате получится 2 объекта-полигона, один из которых содержит остров.

Сложные полилинейные объекты будут разбиты на отдельные полилинии. Из объектов, состоящих из нескольких точек, получается новый точечный объект, в котором каждый исходный точечный объект группы составляет один узел.

Коллекции обрабатываются рекурсивно. Если коллекция содержит регион, то в зависимости от режима **All** будут созданы новые объекты-полигоны. Если она включает сложные полилинии, то из каждой отдельной полилинии будет создан отдельный объект. Если в коллекции содержится группа точек, то будут созданы новые объекты-точки, по одной точке для каждого узла из группы. Все другие типы объектов, включая точки, линии, дуги, прямоугольники, скругленные прямоугольники и эллипсы, которые уже являются простыми объектами, этой операцией не изменяются.

Если регион содержит единственный полигон, то на выходе он останется без изменений. Если сложная полилиния содержит одну полилинию, то на выходе она останется без изменений. Если группа точек содержит единственный узел, то выходящий объект преобразуется в точечный объект, содержащий этот узел. Дуги, прямоугольники, скругленные прямоугольники, эллипсы на выходе остаются без изменений. Все другие типы объектов, например, текстовые не воспринимаются оператором **Objects Disaggregate** и вызовут ошибку.

В режиме **Collection** разделяются только объекты коллекции. Если коллекция содержит регион, то этот регион на выходе станет новым объектом. Если коллекция содержит полилинию, то эта полилиния будет на выходе новым объектом. Если коллекция содержит группу точек, то эта группа точек будет на выходе новым объектом. В этом различие от поведения оператора, режиме, описанном выше, поскольку регион может содержать несколько полигонов, а сложная полилиния может содержать несколько полилиний. При использовании режима, описанного выше, группа точек не будет создана.

В режиме **Collection** на выход передаются без изменений все другие типы объектов, включая точки, группы точек, линии, полилинии, дуги, регионы, прямоугольники, скругленные прямоугольники и эллипсы.

Если не задается **Into Table**, то в качестве таблицы для выходящих данных используется текущая редактируемая таблица. Входящие для обработки командой объекты берутся из текущей выборки.

Дополнительное предложение **Data** определяет, какие значения хранятся в колонках изменяемых объектов. Предложение **Data** может содержать список присваиваемых значений, разделенных точкой с запятой. Все значения, которые могут быть присвоены, описаны в таблице ниже:

Выражение	Эффект
<code>col_name = col_name</code>	Не изменяет величины, хранящиеся в колонке.
<code>col_name = value</code>	Хранит указанные величины в колонке. Если колонка строковая, то значение тоже будет строковым; если колонка числовая, то значение будет числовым.
<code>col_name = Proportion(col_name)</code>	Используется только для числовых колонок; уменьшает число, содержащееся в колонке, пропорционально удаленной площади объекта.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если предложение **Data** содержит назначения только для части колонок, пустые значения будут присвоены тем колонкам, которые не поименованы в предложении **Data**. Если пропустить предложение **Data** целиком, все колонки будут пустыми, содержащими нулевые значения для числовых колонок и пустые значения для строковых колонок.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Пример

```
Open Table "STATES.TAB" Interactive
Map From STATES
Set Map Layer 1 Editable On
select * from STATES
Objects Disaggregate Into Table STATES
```

См. также:

оператор Create Object

оператор Objects Enclose

Назначение

Позволяет создавать области, которые образуются из коллекций полилиний; соответствует команде MapInfo Pro **Замкнуть (полилинии в полигоны)** (на вкладке **ОБЪЕКТЫ** нажать **Полигоны**). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Enclose
[ Into Table tablename ]
[ Region ]
```

tablename – строка с именем таблицы, в которой необходимо создать объекты.

Описание

Objects Enclose создает объекты, представляющие "замыкание" полилиний. Новый регион создается для каждой замкнутой полигональной области. Объекты для обработки командой берутся из текущей выборки. В отличие от случая, когда используется **оператор Objects Combine**, **Objects Enclose** не будет удалять входные объекты-оригиналы. Объединения данных не происходит.

Режим **Region** позволяет использовать в операторе **Objects Enclose** замкнутые объекты (области, прямоугольники, скругленные прямоугольники и эллипсы). В ходе выполнения операции исходные полигоны будут преобразованы в полилинии. Результат эквивалентен следующей последовательности действий: преобразованию всех замкнутых объектов в полилинии и последующему выполнению операции **Objects Enclose**. Все исходные объекты должны быть либо линейными, либо замкнутыми (точки, группы точек, коллекции и текст приведут к появлению ошибки). Если выбраны замкнутые объекты, а ключевое слово **Region** отсутствует, то такие объекты будут пропущены.

Оператор **Objects Enclose** соответствует команде MapInfo Pro **Замкнуть (полилинии в полигоны)**. Описание этой операции смотрите в описании команды в *Руководстве пользователя MapInfo Pro*. Если в MapInfo Pro выполнить команду **Объединить** на вкладке **ОБЪЕКТЫ** и при этом будет открыто окно **MapBasic**, то в протоколе выполненных действий будет использован оператор **Objects Enclose**.

С помощью дополнительного параметра **Into Table** можно сохранить созданные объекты в определенной таблице. Иначе, получаемые объекты будут храниться в той же таблице, откуда берутся исходные объекты.

Пример

Будут выбраны все объекты таблицы testfile, выполнена операция **Objects Enclose**, а полученные результаты сохранены в таблице dump_file.

```
select * from testfile
Objects Enclose Into Table dump_file
```

См. также:

[оператор Objects Check](#), [оператор Objects Combine](#), [оператор Set Combine Version](#)

Оператор Objects Erase

Назначение

Позволяет удалять любую часть целевого объекта, которая пересекается с выбранным объектом; соответствует команде **Удалить часть** на вкладке **ОБЪЕКТЫ**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Erase Into Target
[ Concurrency { All | Aggressive | Intermediate | Moderate | None } ]
[ Data column_name = expression [ , column_name = expression ... ]
[ DropIndex ( Auto | On | Off ) ]
```

column_name – строка с именем колонки изменяемой таблицы.

expression – выражение, которым задаётся то, что будет удалено из колонок *column_name*.

Описание

Оператор **Objects Erase** удаляет часть объекта (или весь объект), который объявлен как изменяемый. Использование оператора **Objects Erase** эквивалентно команде MapInfo Pro **Удалить часть** на вкладке **ОБЪЕКТЫ**. Введение об использовании команды **Удалить часть** смотрите *Руководстве пользователя MapInfo Pro*.

Objects Erase удаляет ту часть изменяемого объекта, которая перекрывает выбранный объект (или объекты). Если изменяемый объект перекрывается полностью, то он удаляется полностью. Если необходимо удалить часть не перекрываемую выбранными объектами, то используйте [оператор Objects Intersect](#).

Перед выполнением оператора **Objects Erase** должен быть выбран изменяемый объект и один или более замкнутых объектов (типа "область", "прямоугольник", "скругленный

прямоугольник" или "эллипс"), играющих роль "ластика". Изменяемый объект может быть назначен командой **Выбрать изменяемый объект** на вкладке **ОБЪЕКТЫ** или с помощью MapBasic-оператора **Set Target**.

Из каждого изменяемого объекта, будет создан единственный объект, состоящий из той части изменяемого объекта, которая не перекрывается "объектами-ластиком". Если изменяемый объект полностью лежит внутри "объектов-ластиков", то никакой объект не будет создан.

Дополнительное, необязательное предложение **Concurrency** позволяет распределить вычисления между несколькими ядрами процессора, чтобы повысить производительность. Когда значение параметра **Concurrency** установлено равным none, а в компьютере используется процессор с несколькими ядрами, то остальные ядра не используются. Это предложение позволяет задавать уровень распараллеливания, используемый при выполнении этой операции на карте. Параметр **Concurrency** может принимать одно из следующих значений:

- **All**: полная параллельная многопоточность. Все процессоры системы используются для выполнения операций. Это стандартный режим MapInfo Pro.
- **Aggressive**: многопоточность, распараллеливание используется агрессивно, при выполнении операций используется 75% процессоров системы.
- **Intermediate**: промежуточный вариант использования многопоточности, распараллеливания, при выполнении операций используется 50% процессоров системы.
- **Moderate**: многопоточность, распараллеливание используется умеренно, при выполнении операций используется 25% процессоров системы.
- **None**: многопоточность, распараллеливание не используется. Операцию выполняет единственный процессор. Эта опция предопределяет наименьшую скорость обработки.

В дополнение к пяти возможным вариантам использования многопоточности, также можно указать количество задействуемых ядер, например, восемь (8). Если компьютер имеет меньшее, чем указанное количество ядер, MapBasic по умолчанию использует все доступные ядра на этой машине. Нулевое значение (0), отрицательное число или некорректное текстовое значение, которое не может быть интерпретировано, как одно из пяти допустимых вариантов многопоточности, приведёт к ошибке.

Дополнительное предложение **Data** определяет, какие значения хранятся в колонках изменяемых объектов. Предложение **Data** может содержать список присваиваемых значений, разделенных точкой с запятой.

Все значения, которые могут быть присвоены, описаны в таблице ниже:

Выражение	Эффект
<code>col_name = col_name</code>	MapBasic не изменяет величину, хранящуюся в колонке.
<code>col_name = value</code>	MapBasic хранит указанные величины в колонке. Если тип колонки символьный, то value должно быть строкой. Если тип колонки числовой, то value должно быть числом.
<code>col_name = Proportion(col_name)</code>	Используется только для числовых колонок; MapBasic уменьшает число, содержащееся в колонке, пропорционально удаленной площади объекта. Так, если была удалена половина объекта, то значение в колонке уменьшится наполовину.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если предложение **Data** содержит назначения только для части колонок, пустые значения будут присвоены тем колонкам, которые не поименованы в предложении **Data**. Если пропустить предложение **Data** целиком, все колонки останутся пустыми, содержащими нулевые значения для числовых колонок и пустые значения для строковых колонок.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Пример

В следующем примере в операторе **Objects Erase** не используется предложение **Data**. Все записи, к которым присоединены изменяемые объекты, теряют свои значения. В этом примере хотя бы один из объектов назначен изменяемым, а один или несколько объектов – выбраны.

```
Objects Erase Into Target
```


Следующий оператор **Objects Erase** имеет предложение **Data**, которое задает выражения для трех колонок ("State_Name", "Pop_1990" и "Med_Inc_80"). Этот оператор присваивает строку "остаток" колонке "State_Name" и определяет, что значения в колонке "Pop_1990" будут уменьшены пропорционально оставшейся после удаления площади. Значения в колонке "Med_Inc_80" сохраняются нетронутыми (какими были до применения оператора **Objects Erase**). Остальные колонки изменяемого объекта очищаются.

```
Objects Erase Into Target
Data
  State_Name = "area remaining",
  Pop_1990 = Proportion( Pop_1990 ),
  Med_Inc_80 = Med_Inc_80
```

Использование предложения **Concurrency**:

```
Objects Erase Into Target
Concurrency All Data...
```

См. также:

[функция Erase\(\)](#), [оператор Objects Intersect](#)[оператор Objects Split](#)

Оператор Objects Intersect

Назначение

Позволяет удалять любую часть целевого объекта, которая пересекается с выбранным объектом; соответствует команде **Удалить внешнюю часть** на вкладке **ОБЪЕКТЫ**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Intersect Into Target
[ Concurrency { All | Aggressive | Intermediate | Moderate | None } ]
[ Data column_name = expression [ , column_name = expression ... ] ]
[ DropIndex ( Auto | On | Off ) ]
```

column_name – строка с именем колонки изменяемой таблицы.

expression – выражение, которым задаётся то, что будет удалено из колонок *column_name*.

Описание

Оператор **Objects Intersect** удаляет часть объекта (или объект полностью), который назначен как изменяемый. Использование оператора **Objects Intersect** эквивалентно команде MapInfo

Pro **Удалить внешнюю часть** на вкладке **ОБЪЕКТЫ**. Введение об использовании команды **Удалить внешнюю часть** смотрите *Руководстве пользователя MapInfo Pro*.

Дополнительное, необязательное предложение **Concurrency** позволяет распределить вычисления между несколькими ядрами процессора, чтобы повысить производительность. Когда значение параметра **Concurrency** установлено равным none, а в компьютере используется процессор с несколькими ядрами, то остальные ядра не используются. Это предложение позволяет задавать уровень распараллеливания, используемый при выполнении этой операции на карте. Параметр **Concurrency** может принимать одно из следующих значений:

- **All**: полная параллельная многопоточность. Все процессоры системы используются для выполнения операций. Это стандартный режим MapInfo Pro.
- **Aggressive**: многопоточность, распараллеливание используется агрессивно, при выполнении операций используется 75% процессоров системы.
- **Intermediate**: промежуточный вариант использования многопоточности, распараллеливания, при выполнении операций используется 50% процессоров системы.
- **Moderate**: многопоточность, распараллеливание используется умеренно, при выполнении операций используется 25% процессоров системы.
- **None**: многопоточность, распараллеливание не используется. Операцию выполняет единственный процессор. Эта опция предопределяет наименьшую скорость обработки.

В дополнение к пяти возможным вариантам использования многопоточности, также можно указать количество задействуемых ядер, например, восемь (8). Если компьютер имеет меньшее, чем указанное количество ядер, MapBasic по умолчанию использует все доступные ядра на этой машине. Нулевое значение (0), отрицательное число или некорректное текстовое значение, которое не может быть интерпретировано, как одно из пяти допустимых вариантов многопоточности, приведёт к ошибке.

Дополнительное предложение **Data** определяет, какие значения хранятся в колонках изменяемых объектов. Предложение **Data** может содержать список присваиваемых значений,

разделенных точкой с запятой. Все значения, которые могут быть присвоены, описаны в таблице ниже:

Выражение	Эффект
<code>col_name = col_name</code>	MapBasic не изменяет величину, хранящуюся в колонке.
<code>col_name = value</code>	MapBasic хранит указанные величины в колонке. Если тип колонки символьный, то value должно быть строкой; если тип колонки числовой, то value должно быть числом.
<code>col_name = Proportion(col_name)</code>	Используется только для числовых колонок; MapBasic уменьшает число, содержащееся в колонке, пропорционально удаленной площади объекта. Так, если была удалена половина объекта, то значение в колонке уменьшится наполовину.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если предложение **Data** содержит назначения только для части колонок, пустые значения будут присвоены тем колонкам, которые не поименованы в предложении **Data**. Если предложение **Data** пропущено целиком, все колонки будут пустыми, содержащими нулевые значения для числовых колонок и пустые значения для строковых колонок.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Оператор **Objects Intersect** очень похож на оператор **Objects Erase**, за одним важным исключением: **Objects Intersect** удаляет те части изменяемых объектов, которые не перекрыты выбранными объектами, а оператор **Objects Erase** удаляет части изменяемых объектов. Из каждого изменяемого объекта, будет создан единственный объект, состоящий из той части изменяемого объекта, которая пресекается с "объектом-ластиком". Например, если изменяемый объект пересекается с тремя выбранными объектами, то будет создано три новых объекта. Части изменяемого объекта за пределами всех выбранных объектов будут удалены. См. также раздел: **Оператор Objects Erase**.

Пример

```
Objects Intersect Into Target
Data
  Field2=Proportion(Field2)
```

Использование предложения **Concurrency**:

```
Objects Intersect Into Target
Concurrency All Data...
```

См. также:

[оператор Create Object](#), [функция IntersectNodes\(\)](#), [функция Overlap\(\)](#), [оператор Objects Erase](#)

Оператор Objects Move

Назначение

Перемещает выбранные в исходной таблице объекты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Move
Angle angle
Distance distance
[ Units unit ]
[ Type { Spherical | Cartesian } ]
```

angle – величина, определяющая угол смещения выбранного объекта.

distance – число, определяющее расстояние смещения выбранного объекта.

unit – единицы измерения расстояний *distance*.

Описание

Objects Move перемещает объекты в пределах исходной таблицы. Исходные объекты получены из текущей выборки. Результирующие объекты заменяют исходные объекты. Объединение данных не проводится, так как данные, связанные с исходными объектами, не изменяются.

Объект перемещается в направлении, заданном параметром *angle*, измеряемым от положительной оси X, указывающей восток (при этом угол измеряется против часовой стрелки),

и смещен на расстояние, заданное параметром расстояния *distance*. Расстояние *distance* измеряется в единицах, указанных параметром *unit*, если он представлен. Если предложение **Units** пропущено, то текущая единица расстояния будет задана по умолчанию. По умолчанию, MapBasic использует мили для измерения расстояний; чтобы изменить стандартные единицы измерения, для замены которых выполняется **оператор Set Distance Units**.

Дополнительная часть предложения **Type** позволяет задать тип расчета расстояния, используемого при смещении объектов. Если используется сферический тип – **Spherical**, то вычисления производятся в координатах "Широта/Долгота", а расстояния *distance* рассчитываются на сфере. Если используется тип Cartesian, то вычисления производятся на плоскости, на которую спроектированные географические данные и расстояния рассчитываются по декартовым алгоритмам. Если часть предложения **Type** не задана, то используется сферический расчет расстояний. Если данные представлены в проекции "Широта/Долгота", то используется сферический тип расчетов независимо от настроек части предложения **Type**. Если данные представлены в проекции план-схема, используются декартовы вычисления независимо от настроек части предложения **Type**.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если система координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
Objects Move Angle 45 Distance 100 Units "mi" Type Spherical
```

См. также:

оператор Objects Offset

Оператор Objects Offset

Назначение

Копирует выбранные объекты, перемещая их. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Offset
[ Into Table intotable ]
Angle angle
Distance distance
[ Units unit ]
[ Type { Spherical | Cartesian } ]
[ Data column = expression [ , column = expression ... ] ]
```

intotable – строка, определяющая таблицу, в которую будут скопированы новые значения.

angle – значение угла сдвига выбранных объектов.

distance – число, определяющее расстояние сдвига выбранного объекта.

unit – единицы измерения расстояний *distance*.

column – строка, определяющая колонку, которой выполняется сдвиг.

expression – выражение, с помощью которого вычисляется сдвиг относительно колонки.

Описание

Objects Offset копирует объекты со сдвигом относительно исходных. Исходные объекты получены из текущей выборки. Новые объекты будут сохранены в таблице *intotable*, если присутствует предложение **Into**. Иначе объекты будут храниться в исходной таблице (например, базовой таблице выбранных объектов).

Объект перемещается в направлении, заданном параметром *angle*, измеряемым от положительной оси X, указывающей восток (при этом угол измеряется против часовой стрелки), и смещен на расстояние, заданное параметром расстояния *distance*. *Distance* измеряется в единицах, указанных параметром *unit*, если он существует. Если предложение **Units** пропущено, то текущая единица расстояния будет задана по умолчанию. По умолчанию, MapBasic использует мили для измерения расстояний; чтобы изменить стандартные единицы измерения, для замены которых выполняется **оператор Set Distance Units**.

Дополнительная часть предложения **Type** позволяет задать тип расчета расстояния, используемого при смещении объектов. Если используется сферический тип – **Spherical**, то

вычисления производятся в координатах "Широта/Долгота", а расстояния distance рассчитываются на сфере. Если используется тип Cartesian, то вычисления производятся на плоскости, на которую спроектированные географические данные и расстояния рассчитываются по декартовым алгоритмам. Если часть предложения **Type** не задана, то используется сферический расчет расстояний. Если данные представлены в проекции "Широта/Долгота", то используется сферический тип расчетов независимо от настроек части предложения **Type**. Если данные представлены в проекции план-схема, используются декартовы вычисления независимо от настроек части предложения **Type**.

Если используется предложение **Data**, приложение выполнит объединение данных.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если система координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
Objects Offset Into Table c:\temp\table1.tbl Angle 45 Distance 100 Units
"mi" Type Spherical
```

См. также:

[функция Offset\(\)](#)

Оператор Objects Overlay

Назначение

Добавляет узлы изменяемому объекту в точках пересечения линий или контуров выбранных объектов. Оператор соответствует команде **Объекты > Добавить узлы**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Overlay Into Target
[ Concurrency { All | Aggressive | Intermediate | Moderate | None } ]
```

Описание

Перед выполнением оператора **Objects Overlay** должен быть назначен изменяемый объект и один или более объектов любого типа, кроме текстового или точечного. Изменяемый объект может быть назначен в меню **Объекты** командой **Выбрать изменяемый объект** или используя **оператор Set Target** из прикладной программы. Более подробная информация приводится в описании команды в *Справке MapInfo Pro*.

Дополнительное, необязательное предложение **Concurrency** позволяет распределить вычисления между несколькими ядрами процессора, чтобы повысить производительность. Когда значение параметра **Concurrency** установлено равным none, а в компьютере используется процессор с несколькими ядрами, то остальные ядра не используются. Это предложение позволяет задавать уровень распараллеливания, используемый при выполнении этой операции на карте. Параметр **Concurrency** может принимать одно из следующих значений:

- **All**: полная параллельная многопоточность. Все процессоры системы используются для выполнения операций. Это стандартный режим MapInfo Pro.
- **Aggressive**: многопоточность, распараллеливание используется агрессивно, при выполнении операций используется 75% процессоров системы.
- **Intermediate**: промежуточный вариант использования многопоточности, распараллеливания, при выполнении операций используется 50% процессоров системы.
- **Moderate**: многопоточность, распараллеливание используется умеренно, при выполнении операций используется 25% процессоров системы.
- **None**: многопоточность, распараллеливание не используется. Операцию выполняет единственный процессор. Эта опция предопределяет наименьшую скорость обработки.

В дополнение к пяти возможным вариантам использования многопоточности, также можно указать количество задействуемых ядер, например, восемь (8). Если компьютер имеет меньшее, чем указанное количество ядер, MapBasic по умолчанию использует все доступные ядра на этой машине. Нулевое значение (0), отрицательное число или некорректное текстовое значение, которое не может быть интерпретировано, как одно из пяти допустимых вариантов многопоточности, приведёт к ошибке.

Пример

Использование предложения **Concurrency**:

```
Objects Overlay Into Target
Concurrency All
```

См. также:

функция **OverlayNodes()**, оператор **Set Target**, оператор **Objects Split**

Оператор Objects Pline

Назначение

Разделяет полилинию на две полилинии. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Pline Split At Node index
[ Into Table name ]
[ Data column_name = expression [ , column_name = expression ... ] ]
```

index – целое, с номером узла, в котором надо разрезать линию.

name – строка с именем таблицы, в которой будут храниться новые объекты.

column_name – строка с именем колонки, в которой будут храниться значения новых объектов.

expression – выражение, которое будет использоваться для создания значений в колонке *column_name*.

Описание

Если объект является полилинией, состоящей из единственного сегмента, то в таблице будут созданы две новые полилинии, состоящие из единственного сегмента. Индекс (**index**) узла (**Node**) должен быть корректным индексом MapBasic для разделяемой полилинии. Если узел (**Node**) является начальным или конечным узлом полилинии, операция разделения не будет выполнена и будет показано сообщение об ошибке.

Дополнительное предложение **Data** определяет, какие значения хранятся в колонках объектов. Предложение **Data** может содержать список присваиваемых значений, разделенных запятыми. Все значения, которые могут быть присвоены, описаны в таблице ниже:

Выражение	Эффект
<code>col_name = col_name</code>	Не изменяет величины, хранящиеся в колонке.
<code>col_name = value</code>	Хранит указанные величины в колонке. Если тип колонки символьный, то значение value должно быть строкой; если тип колонки числовой, то value должно быть числом.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если предложение **Data** содержит назначения только для части колонок, пустые значения будут присвоены тем колонкам, которые не поименованы в предложении **Data**.

Если пропустить предложение **Data** целиком, все колонки будут пустыми, содержащими нулевые значения для числовых колонок и пустые значения для строковых колонок.

Пример

В следующем примере, выбранная полилиния будет разрезана в заданном узле (индекс узла 12). Значения исходных записей выбранной полилинии будут записаны в новые записи разрезанной полилинии без изменений.

```
Objects Pline Split At Node 12 Into Table WORLD Data
Country=Country,Capital=Capital,Continent=Continent,Numeric_code=Numeric_
code,FIPS=FIPS,ISO_2=ISO_2,ISO_3=ISO_3,Pop_1994=Pop_1994,Pop_Grw_Rt=Pop_G
rw_Rt,Pop_Male=Pop_Male,Pop_Fem=Pop_Fem...
```

См. также:

[функция ObjectLen\(\)](#), [функция ObjectNodeX\(\)](#), [функция ObjectNodeY\(\)](#), [оператор Objects Disaggregate](#)

Оператор Objects Snap

Назначение

Выполняет коррекцию объектов данной таблицы и осуществляет различные топологические операции над объектами, включая совмещение узлов разных объектов, прилегающих друг к

другу и генерализацию/разреживание узлов. Таблица может быть таблицей выборки (Selection). Все объекты, подлежащие коррекции, должны быть замкнутыми (полигоны, прямоугольники, скругленные прямоугольники или эллипсы). Одновременно линейные и замкнутые объекты не могут обрабатываться этим оператором, будет выдано сообщение об ошибке. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Snap From tablename
[ Tolerance [ Node node_distance ] [ Vector vector_distance ]
[ Units unit_string ] ]
[ Thin [ Bend bend_distance ] [ Distance spacing_distance ]
[ Units unit_string ] ]
[ Cull Area cull_area [ Units unit_string ] ] ]
```

tablename – строка с именем таблицы, объекты которой должны быть проверены.

node_distance – численное значение радиуса окружностей вокруг концов полилинии.

vector_distance – численное значение радиуса окружностей вокруг промежуточных узлов полилиний.

bend_distance – численное значение допуска коллинеарности серии узлов.

spacing_distance – численное значение минимального расстояния между узлами, которое будет использовано при удалении лишних узлов.

unit_string – строка с названием единиц измерения расстояний.

cull_area – численное значение пороговой площади полигонов, которые будут отсортированы.

unit_string – строка с названием единиц измерения площадей.

Описание

Объекты из исходной таблицы *tablename* проверяются на предмет различных проблем и несоответствий, таких, как самопересечения. Самопересекающиеся полигоны в форме "восьмерки" будут превращены в два полигона с общей вершиной. Полигоны, содержащие острые выступы, подвергаются обработке, при которой часть таких выступов удаляется. Исправленный объект заменит исходный. Если существуют наложения полигонов друг на друга, то такие наложения будут удалены. Удаление взаимных перекрытий обычно состоит из вырезания перекрывающей части из одного объекта, в другом объекте эта часть остается. Область перекрытия состоит из нескольких полигонов. Один полигон представляет собой часть, которая не была перекрыта, а отдельный полигон – каждое перекрытие.

Значения допусков **Tolerance** для **Node** и **Vector** предназначены для совмещения узлов из разных близлежащих объектов и удаляют мелкие пустоты и мелкие пустоты между двумя объектами. Подпредложение **Units** предложения **Tolerance** позволяет задать единицы измерения расстояния (например, "km" для километров), применяемых к значениям **Node** и **Vector**. Если подпредложение **Units** отсутствует, то значения **Node** и **Vector** будут

интерпретироваться в текущих значениях измерения расстояния MapBasic. По умолчанию, MapBasic измеряет расстояния в милях; для того чтобы задать другую единицу измерения, используйте **оператор Set Distance Units**.

Допуск для **Node** — это окружности заданного радиуса вокруг конечных узлов полилиний. Если узлы из других объектов попадают в этот радиус, то один или два узла перемещаются так, что попадают в одну точку (т.е. они совмещаются).

Допуск для **Vector** — это окружности, заданного радиуса вокруг промежуточных узлов полилиний. Его назначение такое же, что и у радиуса **Node**; кроме случаев, когда он используется только для промежуточных точек полилиний. Обратите внимание, что для полигонов не определено понятие конечных точек (значение **Node** не используется) ввиду их замкнутости. Для них используется только величина **Vector**, которая применяется ко всем узлам объекта. Значение допуска **Node** не применяется к полигонам. Для полилиний значение допуска **Node** должно быть больше или равно значению **Vector**.

Значения **Bend** и **Distance** могут использоваться для удобства операций разреживания узлов и обобщения контуров. Они уменьшают число узлов, используемых в объекте, сохраняя основные черты формы объекта. Подпредложение **Units** предложения **Thin** позволяет указать имя единицы измерения расстояния (например, "km" для километров) в которых измеряется значение **Bend** и **Distance**. Если подпредложение **Units** отсутствует, то значения **Bend** и **Distance** будут интерпретироваться в текущих единицах измерения MapBasic.

Значение **Bend** используется для управления коллинеарным отклонением группы из 3 последовательных узлов. Эти 3 узла связываются в треугольник. Измеряется перпендикуляр, опущенный из средней точки на длинную сторону треугольника. Если это расстояние меньше значения **Bend**, то эти три узла рассматриваются как коллинеарные, и второй (средний) узел удаляется из объекта.

Расстояние **Distance** используется для удаления узлов из одного объекта, если узлы расположены слишком близко друг к другу. Измеряется расстояние между двумя соседними точками объекта. Если это расстояние меньше, чем **Distance**, то один из двух узлов будет удален.

Значение **Cull Area** используется для удаления избыточных полигонов, площадь которых меньше некоторого заданного значения. Подпредложение **Units** из предложения **Cull** позволяет настроить единицы измерения площади (например, "кв. км" для квадратных километров), применяемые в значении **Area**. Если подпредложение **Unit** не задано, то площадь брешей **Area** измеряется в текущих единицах MapBasic. По умолчанию, MapBasic измеряет площади в кв. милях; для того чтобы задать другую единицу измерения, используйте **оператор Set Area Units**.

Примечание: Для всех расстояний и площадей, упомянутых выше, всегда используется тип измерений на плоскости. Всегда надо учитывать систему координат и проекцию. Вычисления расстояний и площадей в "Долготе/ Широте" на плоскости осуществляются с невысокой точностью. Перед тем как использовать этот оператор, убедитесь, что Вы работаете в подходящей системе координат (декартовой).

Пример

```
Open Table "STATES.TAB" Interactive
Map From STATES
Set Map Layer 1 Editable On
select * from STATES
Objects Snap From Selection Tolerance Node 3 Vector 3 Units "mi" Thin
Bend
0.5 Distance 1 Units "mi" Cull Area 10 Units "sq mi"
```

См. также:

оператор Create Object, функция OverlayNodes(), функция Overlap()

Оператор Objects Split

Назначение

Позволяет разделить целевые объекты, используя выбранные объекты в качестве шаблона, и соответствует команде **Разрезать** на вкладке **ОБЪЕКТЫ**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Objects Split Into Target
[ concurrency { all | aggressive | intermediate | moderate | none } ]
[ Data column_name = expression [ , column_name = expression ... ]
[ DropIndex ( Auto | On | Off ) ]
```

column_name – строка с именем колонки, в которой будут храниться значения новых объектов.

expression – выражение, которое будет использоваться для создания значений в колонке *column_name*.

concurrency – позволяет использовать несколько ядер для разрезания выбранных объектов карты. Использование нескольких ядер достигается использованием токена concurrency. Если используется токен concurrency, то его необходимо дополнить одним из значений: aggressive, intermediate, moderate, none или число, соответствующее количеству используемых ядер. Также как при использовании оператора **Create Object As Buffer**, если в качестве значения параметра многопоточности после токена concurrency используется нулевое или отрицательное число, появится сообщение об ошибке.

Описание

Оператор **Objects Split** разрезает каждый изменяемый объект на несколько. Использование оператора **Objects Split** эквивалентно команде MapInfo Pro **Разрезать** на вкладке **ОБЪЕКТЫ**. Более подробная информация о разрезах приводится в описании команды в *Справочнике MapInfo Pro*.

Перед выполнением оператора **Objects Split** должен быть выбран изменяемый объект и один или более замкнутых объектов (типа "область", "прямоугольник", "скругленный прямоугольник" или "эллипс"), играющих роль "ластика". Изменяемый объект может быть назначен командой **Выбрать изменяемый объект** на вкладке **ОБЪЕКТЫ** или используя **оператор Set Target**.

Из каждого изменяемого объекта будет создан единственный объект, состоящий из той части изменяемого объекта, которая пресекается с "объектом-ластиком". Например, если изменяемый объект пересекается с тремя выбранными объектами, то будет создано три новых объекта. Кроме того, из всех частей изменяемого объекта, не попадающих внутрь "ластика", будет создан единственный объект. Это эквивалентно выполнению обоих операторов **Objects Erase** и **Objects Intersect** (команда **Удалить внешнюю часть** на вкладке **ОБЪЕКТЫ**).

Дополнительное предложение **Data** определяет, какие значения хранятся в колонках изменяемых объектов. Предложение **Data** может содержать список присваиваемых значений, разделенных точкой с запятой. Все значения, которые могут быть присвоены, описаны в таблице ниже:

Выражение	Эффект
<code>col_name = col_name</code>	Содержимое колонки не меняется; каждый объект, полученный в результате, в своей записи имеет то же значение, что и объект, из которого он был получен.
<code>col_name = value</code>	MapBasic хранит указанные величины в колонке. Если тип колонки символьный, то value должно быть строкой; если тип колонки числовой, то value должно быть числом. Каждый полученный после разреза объект получит заданное значение.
<code>col_name = Proportion(col_name)</code>	Используется только для числовых колонок; MapInfo Pro разделяет значение в колонке объектов по графическим объектам, полученным после разреза. Каждому объекту будет сопоставлено значение пропорциональное площади полученного объекта и исходному значению.

Список в предложении **Data** может состоять из определений для всех колонок таблицы. Если предложение **Data** содержит назначения только для части колонок, пустые значения будут присвоены тем колонкам, которые не поименованы в предложении **Data**. Если предложение

Data пропущено целиком, все колонки будут пустыми, содержащими нулевые значения для числовых колонок и пустые значения для строковых колонок.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Пример

В следующем примере в операторе **Objects Split** не используется предложение **Data**. Все записи, к которым присоединены изменяемые объекты, теряют свои значения.

```
Objects Split Into Target
```

В следующем операторе используется предложение **Data**, которое задает выражения для трех колонок ("State_Name", "Pop_1990" и "Med_Inc_80"). Первая часть предложения **Data** присваивает строку "подразделение" колонке "State_Name"; то есть строка "подразделение" будет помещена в колонку "State_Name" для каждого объекта, являющегося результатом разрезания. Далее в предложении **Data** определяется, что население из колонки "Pop_1990" пропорционально разделяется между объектами, полученными после разреза. В третьей части параметра **Data** задано, что значение "Med_Inc_80" сохраняется для всех объектов.

```
Objects Split Into Target
Data
  State_Name = "sub-division",
  Pop_1990 = Proportion( Pop_1990 ),
  Med_Inc_80 = Med_Inc_80
```

Использование токена concurrency:

```
Objects Split Into Target
concurrency all Data...
```

См. также:

[оператор Alter Object](#)[оператор Create Object As Buffer](#)

Функция Offset()

Назначение

Возвращает копию исходного объекта, смещённого на определенную дистанцию и угол. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Offset( object, angle, distance, units )
```

object – объект, подвергаемый сдвигу.

angle – угол перемещения объекта.

distance – число, определяющее расстояние сдвига выбранного объекта.

units – строка, представляющая единицы измерения расстояния *distance*.

Возвращаемая величина

Объект

Описание

Offset() создаёт новый объект, являющийся копией исходного объекта, смещённой на определённое расстояние *distance* и повернутой на заданный угол *angle* в градусах относительно горизонтали (это нулевой угол, положительные значения отсчитываются против часовой стрелки). Строковое значение *units* представляет собой единицы измерения расстояния, которые используются в **функции ObjectLen()** или **функции Perimeter()**, является значением для параметра *distance*. Для параметра *distance* применяется способ расчета на сфере кроме тех случаев, когда используется план-схема. Для план-схемы автоматически используется декартовый алгоритм расчета расстояний. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если

система координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
Offset(Rect, 45, 100, "mi")
```

См. также:

[оператор Objects Offset](#), [функция OffsetXY\(\)](#)

Функция OffsetXY()

Назначение

Возвращает копию исходного объекта, смещённую на заданное расстояние по X и Y. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
OffsetXY( object, xoffset, yoffset, units )
```

object – объект, подвергаемый сдвигу.

xoffset и *yoffset* – числа, определяющие расстояния вдоль осей X и Y, на которые будет смещен объект;

units – строка, представляющая единицы измерения расстояния distance.

Возвращаемая величина

Объект

Описание

OffsetXY() создает новый объект, который является копией исходного объекта, смещенного на расстояние *xoffset* вдоль оси X и на расстояние *yoffset* вдоль оси Y. Строковое значение *units* представляет единицы измерения расстояния. См. также разделы: [Функция ObjectLen\(\)](#) или [Функция Perimeter\(\)](#). Для параметра distance применяется способ расчета на сфере кроме тех случаев, когда используется план-схема. Для план-схемы автоматически

используется декартовый алгоритм расчета расстояний. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Так, при перемещении объекта по карте, построенной в проекции "Широта/Долгота", его форма не изменится, но поменяется площадь. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если система координат – "Широта/Долгота", преобразование в градусы использует фиксированное число десятичных знаков. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
OffsetXY(Rect, 92, -22, "mi")
```

См. также:

[функция Offset\(\)](#)

Оператор OnError

Назначение

Вызывает процедуру обработки ошибок.

Синтаксис

```
OnError Goto{ label | 0 }
```

label – строка, представляющая метку в тексте некоторой подпрограммы или функции.

Ограничения!

Нельзя использовать оператор **OnError** в окне **MapBasic**.

Описание

Оператор **OnError** используется либо для запуска процедуры-обработчика ошибок, если ошибка имела место, либо для отмены обработки ошибок (форма **OnError Goto 0**). (Процедура-обработчик ошибок представляет собой группу операторов, которые выполняются в случае ошибки).

В отличие от стандартных версий BASIC, оператор **OnError** в MapBasic пишется в одно слово.

Оператор **OnError Goto label** объявляет, что операторы после метки *label* являются обработчиком ошибок. Если один из операторов, следующих за **OnError**, вернет код ошибки, то MapBasic передаст выполнение программы метке *label*. Предполагается, что операторы, выполняемые по метке *label*, должны обработать конфликт, возникший в результате ошибки, так, чтобы он не повлиял на корректность выполнения программы, или предупреждали пользователя о случившейся ошибке, или то и другое. В процедуре-обработчике ошибок используйте **оператор Resume**, чтобы возобновить выполнение программы.

Заметим, что если Ваша программа имеет обработчик ошибок, то Вы должны перед оператором с меткой *label* расположить оператор управления выполнением программы (например, **оператор Exit Sub** или **оператор End Program**). Это не позволяет программе передать управление процедуре-обработчику без наличия ошибки. Смотрите пример ниже.

Оператор **OnError Goto 0** отменяет выполнение установленной до этого процедуры-обработчика ошибок. Если ошибка происходит в программе, где нет обработчика ошибок или он отменен, то MapBasic выводит на экран окно сообщения об ошибке и прекращает выполнение программы.

Операторы обработчика ошибок могут располагаться в отдельной процедуре или функции. Так, подпрограмма может запускать процедуру-обработчик ошибок следующим оператором:

```
OnError Goto recover
```

(при этом подразумевается, что в этой процедуре есть метка "recover"). Если после выполнения такого оператора **OnError** процедура выполнит **оператор Call** и перейдет в другую подпрограмму, то процедура-обработчик по метке "recover" не будет способна реагировать на ошибку, пока действует другая процедура, которую вызывает оператор Call. Это происходит потому, что каждая метка локальна по отношению к процедуре и функции, в которой она задана. Этот прием позволяет каждой функции и каждой процедуре сопоставить собственный обработчик ошибок.

Примечание: Если ошибка возникнет в процессе обработки других ошибок, то выполнение программы прекратится.

Пример

```
OnError GoTo no_states
Open Table "states"
```

```

OnError GoTo no_cities
Open Table "cities"

Map From cities, states
after_mapfrom:
  OnError GoTo 0
  '
  ' ...
  '
End Program

no_states:
  Note "Could not open table States... no Map used."
  Resume after_mapfrom

no_cities:
  Note "City data not available..."
  Map From states
  Resume after_mapfrom

```

См. также:

[функция Err\(\)](#), [оператор Error](#), [функция Error\\$\(\)](#), [оператор Resume](#)

Оператор Open Connection

Назначение

Устанавливает соединение с внешней службой геокодирования или маршрутизации сервера MapMarker, Envinsa или глобальный сервер геокодирования Precisely Global Geocoder. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Open Connection
Service { Geocode [ MapMarker | Envinsa ] | Isogram }
ServiceName servicename_string
URL URLstring
[ User name_string [ Password pwd_string ] ]
[ Interactive [ On | Off ] ]
into variable var_name

```

Servicename_string – строка, обозначающая корректную службу геокодирования. Отныне, мы поддерживаем глобальные серверы геокодирования: "Global Geocoder" и "Precisely Global Geocoder".

URLString – строка, содержащая допустимый URL. *URLString* – корректный URL адрес сервера маршрутизации, если указан **Isogram** или адрес сервера геокодирования, если указан **Geocode**.

name_string – строка, представляющая имя пользователя для сервиса Envinsa или MapMarker.

pwd_string – строка, представляющая собой пароль, соответствующий пользователю *user_name*.

var_name – строка, представляющая переменную, содержащую возвращённый номер соединения.

Описание

Оператор **Open Connection** создаёт соединение с сервером геокодирования или маршрутизации. Каждый оператор должен содержать информацию о сервисе и провайдере, соединение с которым устанавливается. Так как служба маршрутизации представляется только сервером Envinsa, другой провайдер пока указан быть не может. Если устанавливается соединение со службой геокодирования и провайдер не указан, по умолчанию предполагается, что это **Envinsa**.

Ключевые слова **User** и **Password** требуются только, когда используются службы **Envinsa** и **Global Geocoder services**.

Ключевому слову **Into variable** требуется переменная *var_name*, содержащая возвращённый номер соединения, который затем передаётся в другие операторы, такие как **оператор Set Connection Geocode**, **оператор Geocode**, **оператор Set CoordSys** и **оператор Create Object Isogram**.

Interactive определяет, показывать ли диалоговый запрос имени пользователя и пароля в случае, если авторизация не прошла успешно. Если **Interactive** задан в режиме **Off**, диалог не отображается и команда не выполняется, если авторизация не прошла успешно. Если слово **Interactive** задано в режиме **On**, то в случае неудачи с авторизации появляется диалог с запросом имени и пароля.

По умолчанию для этой команды задан режим **Interactive Off**. Если ключевое слово **Interactive** не используется, это эквивалентно установке параметра **Interactive Off**. Однако, если параметр **Interactive** указан без аргументов, это равнозначно команде **Interactive On**.

Примеры

Примечание: Все примеры без указания ключевого слова **MapMarker** подразумевают использование сервера Envinsa.

Открывается соединение с сервером геокодирования без указания ключевого слова **Interactive**. **Interactive** установлен **Off** по умолчанию.

```
Open Connection Into Variable CnctNum Service Geocode URL
"http://EnvinsaServices/LocationUtility/services/LocationUtility"
```

Следующий пример открывает соединение с сервером геокодирования с указанием ключевого слова **Interactive** с параметром **On**.

```
Open Connection Into Variable CnctNum Service Geocode URL
"http://EnvinsaServices/LocationUtility/services/LocationUtility"
Interactive On
```

В этом примере открывается соединение с сервером геокодирования, требующим авторизации.

```
dim baseURLVariable as String
baseURLVariable = "http://EnvinsaServices/"
Open Connection Service Geocode URL baseURLVariable +
"LocationUtility/services/LocationUtility" User "geocodeuser" Password
"GeoMe" Into Variable CnctNum
```

Этот пример открывает соединение с сервером маршрутизации, требующем авторизации.

```
dim baseURLVariable as String
baseURLVariable = "http://EnvinsaServices/"
Open Connection Service IsoGram URL baseURLVariable +
"Route/services/Route" User "isogramuser" Password "ISOMe" Into Variable
CnctNum
```

См. также:

[оператор Close Connection](#), [оператор Set Connection Geocode](#), [оператор Set Connection IsoGram](#)

Оператор Open File

Назначение

Открывает файл для ввода/вывода.

Синтаксис

```
Open File filespec
[ For { Input | Output | Append | Random | Binary } ]
[ Access { Read | Write | Read Write } ]
As [ # ] filenum
[ Len = recordlength ]
[ ByteOrder { LOWHIGH | HIGHLOW } ]
[ CharSet char_set ]
```

filespec – строка с именем создаваемого файла.

filenum – целочисленный номер, который будет присвоен файлу вплоть до завершения работы с ним (например, **оператор Get** или **оператор Put**).

recordlength – число символов в одной записи (включая символ конца строки) для доступа к файлу в режиме произвольного доступа (Random).

char_set – имя набора символов; см. раздел **Предложение CharSet**.

Предупреждение!

Нельзя использовать оператор **Open File** в окне **MapBasic**.

Описание

Оператор **Open File** открывает текстовый файл для операций ввода/вывода прикладной программой.

В MapBasic понятия таблицы и файла существенно различаются. Для операций ввода/вывода таблиц используется один набор операторов (например, **оператор Open Table**, **оператор Fetch** и **оператор Select**), а для файлов других типов другой набор операторов (например, **Open File**, **оператор Get**, **оператор Put**, **оператор Input #**, **оператор Print #**).

Предложение **For** задает режим доступа к данным файла: режим последовательного доступа, режим произвольного доступа или режим бинарного доступа. Типы ввода/вывода описаны ниже. Если предложения **For** нет в операторе, то для открытия файла используется режим **Random** (произвольного доступа).

Файлы последовательного доступа

Если требуется прочитать текст из файла, записи которого имеют разную длину (например, одна строка имеет 55 символов, а следующая – 72 и т. д.), то необходимо использовать режим последовательного доступа. Для задания этого режима в предложении **For** используется ключевое слово **Input**, **Output** или **Append**.

Если используется предложение **For Input**, то можно прочитать из файла, используя **оператор Input #** и **оператор Line Input #**.

Если использовано предложение **For Output** или предложение **For Append**, то для записи файл можно использовать **оператор Print #** и **оператор Write #**.

Если используется предложение **For Input**, то в предложении **Access** можно использовать только ключевое слово **Read**; аналогично, с предложением **For Output** можно использовать в предложении **Access** только ключевое слово **Write**.

Предложение **Len** не должно использоваться в операторе, если задается режим последовательного доступа.

Файлы произвольного доступа

Если требуется прочитать текст из файла, записи которого имеют одинаковую длину (например, каждая строка по 80 символов длиной), то необходимо открыть файл в режиме **произвольного доступа** mode, Для задания этого режима используется параметр. **For Random**.

Для режима **произвольного доступа** необходимо задать длину записи в предложении **Len = recordlength**. Величина в параметре *recordlength* должна задать количество символов одной записи, включая символы конца записи, например, пара символов "возврат каретки" и "новая строка".

Для режима **произвольного доступа** в предложении **Access** Вы можете использовать ключевые слова: **Read** для чтения, **Write** для записи или (**Read Write**). Для чтения из файла и записи в файл, открытый в режиме **произвольного** доступа, используются соответственно **оператор Get** и **оператор Put**.

Двоичные файлы

Если файл открыт в **бинарном** режиме, то MapBasic конвертирует величину переменной MapBasic в бинарную величину в случае записи и наоборот в случае чтения. Хранение числовых данных в бинарном файле более компактно, чем хранение бинарных данных в текстовом файле. Но бинарный файл нельзя показывать и распечатывать как текстовый.

Для открытия файла в бинарном режиме используется предложение **For Binary**.

Для режима **бинарного** доступа в предложении **Access** Вы можете использовать ключевые слова: **Read** для чтения, **Write** для записи или (**Read Write**). Для чтения из файла и записи в файл, открытый в бинарном режиме, используются соответственно **оператор Get** и **оператор Put**.

Предложение **Len** и **предложение CharSet** не должны использоваться в операторе, если задается режим бинарного доступа.

Управление работой с файлом

Предложение **CharSet** определяет набор символов. Параметр *char_set* должен быть строковой константой, такой как "MacRoman" или "WindowsLatin1". Если предложение CharSet опущено, MapInfo Pro будет использовать заданный по умолчанию набор символов. Заметим, что оно используется только в том случае, если файл открывается в режимах **Input**, **Output** или **Random**. Подробнее см. **Предложение CharSet**.

Если открыт файл в режиме произвольного или бинарного доступа (**Random** или **Binary**), предложение **ByteOrder** задает способ представления в файле.

Если прикладная программа действует только в пределах одной вычислительной платформы, то Вам не надо беспокоиться о порядке разрядов байтов в файле. Но если необходимо читать из бинарного файла или писать в бинарный файл, который был создан или будет использоваться в другой платформе, то Вам придется контролировать порядок расположения байтов с помощью предложения **ByteOrder**.

Примеры

```
Open File "cxdata.txt" For INPUT As #1
Open File "cydata.txt" For RANDOM As #2 Len=42
Open File "czdata.bin" For BINARY As #3
```

См. также:

оператор Close File, функция EOF(), оператор Get, оператор Input #, оператор Open Table, оператор Print #, оператор Put, оператор Write #, предложение CharSet

Оператор Open Table

Назначение

Открывает таблицу MapInfo Pro для ввода/вывода. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Open Table filename [ As tablename ]
[ Hide ] [ ReadOnly ] [ Interactive ] [ Password pwd ]
[ NoIndex ] [ MRU ] [ View Automatic ] [ DenyWrite ]
[ VMGrid | VMRaster | VMDefault ]
```

filename — строка, задающая таблицу MapInfo, которую требуется открыть.

tablename — строка с "синонимом" имени, под которым открывается таблица.

pwd — строка, представляющая собой пароль на уровне базы данных, определяемый при включении защиты базы данных. Используется только для таблиц Access.

VMGrid задается для открытия файлов Vertical Mapper GRD как слоев поверхности.

VMRaster задается для открытия файлов Vertical Mapper GRD как растровых слоев.

VMDefault задается для открытия файлов GRD-файлов либо как растровых слоев, либо как слоев-поверхностей, в зависимости от наличия в TAB-файле тега RasterStyle 6 1.

MRU — дополнительный, необязательный токен. Если присутствует, то добавляет имя успешно открытой таблицы в список последних используемых — MRU.

Описание

Оператор **Open Table** открывает уже существующую таблицу. Эффект от этого оператора такой же, как и от команды MapInfo **Открыть таблицу**. Таблица должна быть открыта командой или оператором, прежде чем MapInfo Pro сможет производить какие-либо действия с таблицей.

Примечание: Имя файла, который будет открыт (определяемый параметром *filespec*) должен соответствовать таблице, которая уже существует; для создания новой таблицы используется **оператор Create Table**. Оператор **Open Table** можно применять только к таблицам MapInfo; для того чтобы открыть файлы других форматов, следует применять функции **оператора Register Table** и **оператора Open File**.

Если оператор включает предложение **As**, MapInfo Pro открывает таблицу под именем, задаваемым параметром *tablename*, которое мы будем называть “синонимом” или “псевдонимом” таблицы. Это влияет на то, как имя таблицы будет показано в списках, например, в том списке, который появляется при выполнении команды **Заккрыть**. Более того, если оператор **Open Table** задает синоним имени таблицы, все следующие операции с таблицами, выполняемыми MapBasic'ом (например, **оператором Close Table**) должны использовать синоним имени таблицы, а не постоянное имя таблицы. Синоним имени далее представляет эту таблицу во всех списках MapInfo. Назначение имени-синонима не имеет ничего общего с операцией переименования таблицы.

Если оператор включает предложение **Hide**, то имя таблицы не появится ни в каком диалоге, показывающем список открытых таблиц (например, в диалоге **Заккрыть таблицу**). Используйте предложение **Hide** если требуется открыть таблицу, которая останется скрытой от пользователя. Если оператор включает предложение **ReadOnly**, пользователь не сможет редактировать таблицу.

Дополнительное ключевое слово **Interactive** дает команду MapBasic подсказать пользователю место нахождения таблицы, если она не найдена по указанному пути. Ключевое слово **Interactive** полезно в ситуации, когда Вы не знаете, где находятся нужные файлы. Если оператор включает ключевое слово **NoIndex**, индекс MapInfo не будет встраиваться в таблицу MS Access при ее открытии.

View Automatic — дополнительное предложение оператора **Open Table**, с помощью которого можно запустить в текущем сеансе MapInfo Pro ассоциированные с геолинком либо таблицу MapInfo, либо рабочий набор, либо файл программы, в том числе, начать новый сеанс, если ни одного не запущено. При наличии параметра **View Automatic** MapInfo Pro либо добавит эти данные в существующее окно, либо откроет новое окно карты или списка. Это особенно полезно для геолинков.

DenyWrite — дополнительное предложение параметра только для таблиц MS Access; если задан этот параметр, другие пользователи не смогут редактировать эту таблицу. Если же другой пользователь уже редактирует такую таблицу, то оператор **Open Table** не будет выполнен.

Открытие двух таблиц с одинаковым именем

MapInfo Pro может открыть две отдельные таблицы, которые имеют одно и то же имя. В этом случае, MapInfo Pro должно открыть вторую таблицу под специальным именем чтобы избежать конфликтов. В зависимости от того, включает ли оператор **Open Table** ключевое слово **Interactive**, MapBasic либо присваивает специальное имя таблице автоматически, либо показывает диалог, позволяющий пользователю интерактивно выбрать специальное имя таблицы. \

Например, пользователь может хранить две копии таблицы SITES: одну копию в каталоге 2006 ("C:\2006\SITES.TAB" и другую, возможно более новую, в другом каталоге "C:\2005\SITES.TAB"). Когда пользователь (или приложение) открывает первую таблицу Sites, MapInfo Pro открывает таблицу под ее именем ("Sites"). Если приложение использует оператор **Open Table** для открытия второй таблицы Sites, MapInfo Pro автоматически открывает вторую таблицу под измененным именем (например, "Sites_2") что бы отличать ее от первой таблицы. С другой стороны, если оператор **Open Table** включает предложение **Interactive**, MapInfo Pro откроет диалог, позволяющий пользователю выбрать для таблицы альтернативное имя.

Несмотря на то, использует ли оператор **Open Table** ключевое слово **Interactive**, в результате таблица может быть открыта с нестандартным именем. Вслед за оператором **Open Table**, можно выполнить функцию `TableInfo(0, TAB_INFO_NAME)` для определения имени, под которым MapInfo Pro открыло таблицу.

Открытие таблицы, которая уже открыта

Если таблица уже открыта, и оператор **Open Table...As** пытается заново открыть ту же таблицу под новым именем, MapBasic сгенерирует код ошибки. Одна таблица не может быть открыта под двумя различными именами одновременно.

Однако, если таблица уже открыта, а затем оператор **Open Table** пытается заново открыть таблицу без указания нового имени, MapBasic не будет генерировать код ошибки. Таблица просто остается открытой под ее текущим именем.

Пример

Следующий пример открывает таблицу `STATES.TAB`, затем отображает таблицу в окне карты. Поскольку оператор **Open Table** использует предложение **As** для открытия таблицы под псевдонимом (USA), в окне карты появится заголовок "USA Map", а не "States Map".

```
Open Table "States" As USA
Map From USA
```

В следующем примере функция **TableInfo()** используется вместе с оператором **Open Table**. Если определенная таблица уже открыта (States) и Вы выполняете программу ниже, MapBasic будет открывать эту таблицу "C:States.TAB" с определенным номером (например, "States_2").

Функция TableInfo() будет возвращать номер, под которым таблица "C:STATES.TAB" была открыта.

```
Include "MAPBASIC.DEF"
Dim s_tab As String
Open Table "C:states"
s_tab = TableInfo(0, TAB_INFO_NAME)
Browse * From s_tab
Map From s_tab
```

См. также:

оператор Close Table, оператор Create Table, оператор Delete, оператор Fetch, оператор Insert, функция TableInfo(), оператор Update

Оператор Open Window

Назначение

Открывает или показывает окно. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Open Window window_name
```

window_name – имя окна (например, Ruler) или код окна (например, WIN_RULER), коды которых задаются в файле MAPBASIC.DEF.

Описание

Оператор **Open Window** показывает окно MapInfo Pro. Например, следующий оператор отображает окно статистики, как будто пользователь выполняет команду **Статистика** на вкладке **ГЛАВНАЯ** в группе .

```
Open Window Statistics
```

В следующей таблице описаны значения *window_name*:

Значение	Описание окна
Справка	Окно Справочной системы (WIN_HELP).

Значение	Описание окна
Info	Окно Информация (WIN_INFO).
LayerControl	Окно Управление слоями (WIN_LAYER_CONTROL). В приложениях интегрированной картографии это относится к модальному диалогу.
Legend	Окно Легенда (WIN_LEGEND).
MapBasic	Окно MapBasic (WIN_MAPBASIC).
Message	Окно Сообщение , которое использует оператор Print (WIN_MESSAGE).
MoveMapTo	Окно Переместиться по Карте (WIN_MOVE_MAP_TO).
Ruler	Окно инструмента Линейка (WIN_RULER).
Statistics	Окно Статистика (WIN_STATISTICS).
TableList	Окно Список таблиц (WIN_TABLE_LIST). В приложениях интегрированной картографии это относится к модальному диалогу.

Примечание: ID-номера окон **Список таблиц**, **Управление слоями** и **Переместиться по карте** к игнорируются оператором **Set Window**, функцией **WindowInfo()** и функцией **WindowID()**.

Нельзя открывать окно документа (**Карта**, **Список**, **График**, **Отчёт**) оператором **Open Window**. Для того чтобы открыть окно документа каждого из этих типов, существуют отдельные операторы (смотрите: **оператор Map**, **оператор Graph**, **оператор Browse**, **оператор Layout** и оператор **Create Redistrict**).

См. также:

оператор Close Window, **оператор Print**, **оператор Set Window**

Функция **Overlap()**

Назначение

Возвращает объект, полученный в результате географического пересечения двух объектов; результат похож на получаемый командой MapInfo Pro **Удалить внешнюю часть**. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Overlap( object1, object2 )
```

object1 – параметр, задающий пересекающиеся объекты, ни один из которых не может быть точечным.

object2 – параметр, задающий пересекающиеся объекты, ни один из которых не может быть точечным.

Возвращаемая величина

Объект, являющийся пересечением *object1* и *object2*.

Описание

Функция **Overlap()** вычисляет географическое пересечение двух объектов (площадь перекрытия одним объектом другого), и возвращает объект, представляющий пересечение.

MapBasic передает все стили оформления объекта *object1* результирующему объекту. Если необходимо, стиль оформления объекта можно изменить на текущий в прикладной программе.

Если один из объектов является линейным (например, полилинией), а второй замкнутым (например, область) функция **Overlap()** вернет часть линейного объекта, которая находит на площадь замкнутого.

См. также:

[функция AreaOverlap\(\)](#), [функция Erase\(\)](#), [функция IntersectNodes\(\)](#), [функция OverlayNodes\(\)](#), [оператор Objects Intersect](#)

Функция OverlayNodes()

Назначение

Возвращает объект, созданный на основе существующего, добавлением узлов в точках пересечения со вторым объектом. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
OverlayNodes ( input_object, overlay_object )
```

input_object – объект, на основе которого будет создан результирующий, и который не может быть точечным.

overlay_object – объект, пересекающий объект *input_object* (также не может быть точечным).

Возвращаемая величина

Объект, либо типа "область", либо типа "полилиния"

Описание

Функция **OverlayNodes()** возвращает объект, созданный из узлов первого плюс узлы, полученные пересечением линий или контуров объектов *input_object* и *overlay_object*.

Если объект *input_object* замкнут (область, прямоугольник, скругленный прямоугольник или эллипс), то функция **OverlayNodes()** вернет область. Если объект *input_object* линейный (прямая линия, полилиния или дуга), то функция **OverlayNodes()** вернет полилинию.

MapBasic передает все стили оформления объекта *input_object* результирующему.

Для определения, прибавила ли функция *OverlayNodes()* несколько узлов к тем, которые были у объекта *input_object*, используется [функция ObjectInfo\(\)](#). Заметим, что если объект *input_object*, пересекающийся с другим, уже имеет узлы в точках пересечения, то функция **OverlayNodes()** не будет добавлять новых узлов к имеющимся в *input_object*, то есть результирующий объект будет состоять только из тех узлов, которые были у первого объекта.

См. также:

[функция IntersectNodes\(\)](#), [оператор Objects Overlay](#)

Оператор Pack Table

Назначение

Обеспечивает функциональность, которую предлагает команда MapInfo Pro **Упаковать таблицу** (на вкладке **ТАБЛИЦА** нажмите **Таблица**). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Pack Table table { Graphic | Data | Graphic Data } [ Interactive ]
```

table – имя открытой таблицы, которая не имеет не сохраненные изменения.

Описание

Для упаковки неграфических данных таблицы в операторе используется ключевое слово **Data**. Когда данные сжимаются таким образом, MapInfo Pro физически удаляет все строки, которые были помечены как удаленные.

Для упаковки графических объектов таблицы в операторе используется ключевое слово **Graphic**. При упаковке графики удаляются пустые места из MAP-файла таблицы. Однако, упаковка графических объектов несколько замедляет графические операции.

Оператор **Pack Table** может использовать одновременно и слово **Data**, и слово **Graphic**, или должен включать хотя бы одно.

Оператор **Pack Table** может послужить причиной удаления слоев из окна Карты, а также привести к потере объектов тематических слоев или объектов с Косметического слоя..

Если используется ключевое слово **Interactive**, то MapInfo Pro предложит пользователю сохранить тематические или/и косметические объекты. Этот оператор не может сжимать связанные таблицы. Кроме того, MapInfo не может сжимать таблицу, если она была изменена и эти изменения не были сохранены на диске. Для сохранения на диск таблицы используйте **оператор Commit Table**.

Примечание: Упаковка таблицы может повлиять на подписи, созданные или измененные пользователем и сохраненные в рабочем наборе. Допустим, что подписи были созданы и сохранены в рабочем наборе. Если строки были удалены из таблицы и таблица упакована, то при следующем открытии рабочего набора подписи могут оказаться в неправильных местах. (Это происходит потому, что в Рабочем Наборе подписи пользователя соотносятся с номером строки таблицы; операция упаковки меняет порядок записей потому, что из таблицы исключаются удаленные записи; после упаковки подписи могут появляться не там, где ожидалось и иметь неправильный вид). Если удалить

последние записи таблицы (например, выбрав их в конце окна Списка), то упаковка не повлияет на созданные подписи.

Упаковка таблиц Access

Оператор **Pack Table** скопирует исходную таблицу Microsoft Access без колонок тех типов, которые MapInfo Pro не поддерживает. Если в таблице Microsoft Access имеются колонки типов MEMO, OLE или LONG BINARY, то такие колонки при упаковке пропадут.

Пример

```
Pack Table parcels Data
```

См. также:

[оператор Open Table](#)

Функция PathToDirectory()

Назначение

Возвращает заданный каталог с файлами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PathToDirectory$( filespec )
```

filespec – строка, содержащая полное имя файла (носитель+каталог + имя).

Возвращаемая величина

Строка

Описание

Функция **PathToDirectory\$()** извлекает каталог из записи полного дискового маршрута к файлу.

Имя файла, заданное полностью, состоит из каталога и имени файла. Имя файла, заданное полностью C:\MAPINFO\DATA\WORLD.TAB, содержит каталог "C:\MAPINFO\DATA\".

Пример

```
Dim s_filespec, s_filedir As String
s_filespec = "C:\MAPINFO\DATA\STATES.TAB"
s_filedir = PathToDirectory$(s_filespec)

' s_filedir now contains the string "C:\MAPINFO\DATA\"
```

См. также:

функция [PathToFileName\(\)](#), функция [PathToTableName\(\)](#)

Функция PathToFileName()

Назначение

Извлекает из полного имени файла имя файла. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PathToFileName$( filespec )
```

filespec – строка, содержащая полное имя файла (носитель+каталог + имя).

Возвращаемая величина

Строка

Описание

Функция **PathToFileName\$()** извлекает из полного дискового маршрута к файлу только имя файла.

Имя файла, заданное полностью, состоит из каталога и имени файла. Функция **PathToFileName\$()** возвращает часть полного имени, которая содержит имя файла.

Имя файла, заданное полностью C:\MAPINFO\DATA\WORLD.TAB, состоит из имени каталога ("C:\MAPINFO\DATA\") и имени файла ("WORLD.TAB").

Пример

```
Dim s_filespec, s_filename As String
s_filespec = "C:\MAPINFO\DATA\STATES.TAB"
s_filename = PathToFileName$(s_filespec)

' filename now contains the string "STATES.TAB"
```

См. также:

функция [PathToDirectory\(\)](#), функция [PathToTableName\(\)](#)

Функция PathToTableName()

Назначение

Возвращает синоним имени таблицы (псевдоним, например, "_1995_Data"), полученный из полного имени файла таблицы (например, "C:\MapInfo\Data\1995 Data.tab"). Эту функцию можно вызывать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PathToTableName$( filespec )
```

filespec – строка, содержащая полное имя файла (носитель+каталог + имя).

Возвращаемая величина

Строка (string) длиной до 31 символа.

Описание

Получая полное имя файла с расширением TAB функция возвращает строку, которая может быть для этой таблицы в данный момент псевдонимом (alias). Именно такой синоним видит пользователь в строке заголовка документального окна MapInfo Pro (например, в строке заголовка списка).

Для получения синонима таблицы при ее открытии MapInfo Pro удаляет из полного имени файла имя носителя, каталога и расширение TAB. Любые специальные символы, такие как тире, пробелы и др. заменяются знаками подчеркивания (_). Если имя файла начинается с цифры, то MapInfo Pro вставляет знак подчеркивания в начало имени таблицы. При этом длина строки имени таблицы увеличивается на 1 символ, так 30-символьное строка имени

может стать 31-символьной. Если результирующая строка получается длиннее 31 символа, то MapInfo Pro отсекает лишнее с конца; имена псевдонимов не могут быть длиннее 31 символа.

Заметим, что таблица может быть открыта с именем-псевдонимом, отличающимся от имени файла. Программа MapBasic может открыть таблицу с именем-псевдонимом, используя **оператор Open Table** с предложением **As**. Например, откроем таблицу WORLD под псевдонимом "Earth":

```
Open Table "C:\MapInfo\Data\World.tab" As Earth
```

Могут быть также открыты две одноименные таблицы, расположенные в разных каталогах, и MapInfo Pro автоматически изменит имя одной из таблиц. В таких случаях имя таблицы, возвращаемое функцией **PathToTableName\$ ()**, может не совпадать с именем, под которым она открыта в MapInfo. Для того, чтобы определить синонимическое имя открытой таблицы, используется **функция TableInfo()** с параметром (TAB_INFO_NAME).

Пример

```
Dim s_filespec, s_tablename As String
s_filespec = "C:\MAPINFO\DATA\STATES.TAB"
s_tablename = PathToTableName$(s_filespec)
' s_tablename now contains the string "STATES"
```

См. также:

функция PathToDirectory(), **функция PathToFileName()**, **функция TableInfo()**

Предложение Pen

Назначение

Задаёт стиль контуров графических объектов. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Pen pen_expr
```

pen_expr – это выражение, возвращающее объект типа "Pen", например, **MakePen(width, pattern, color)**.

Описание

Предложение **Pen** входит в состав некоторых операторов, в которых необходимо задавать стиль линии для некоторых графических объектов. Стиль линии представляет собой набор из атрибутов толщины линии, типа линии и цвета.

Предложение **Pen** не является полным оператором. Например, **оператор Create Line** использует предложение **Pen**, когда создается новый объект типа "линия". Слово **Pen** может сопровождаться выражением, возвращающим значение типа **Pen**. Например, выражение такого вида может быть переменной **Pen**:

```
Pen pen_var
```

или вызов функции (например, **функция CurrentPen()** или **функция MakePen()**), которые возвращают значение типа "Pen":

```
Pen MakePen(1, 2, BLUE)
```

Можно создать стиль линии, рисуемый с перекрытиями, прибавляя 128 к значению штриховки. В следующем примере задается двухпиксельная фиолетовая (cyan) линия образца 101, прорисовываемая в режиме перекрытия (101+128=229):

```
Pen MakePen(2, 229, CYAN)
```

В некоторых операторах (например, **оператор Set Map**) после слова **Pen** стиль задается непосредственно набором из трех целочисленных параметров (width, pattern и color), например:

```
Pen(1, 2, BLUE)
```

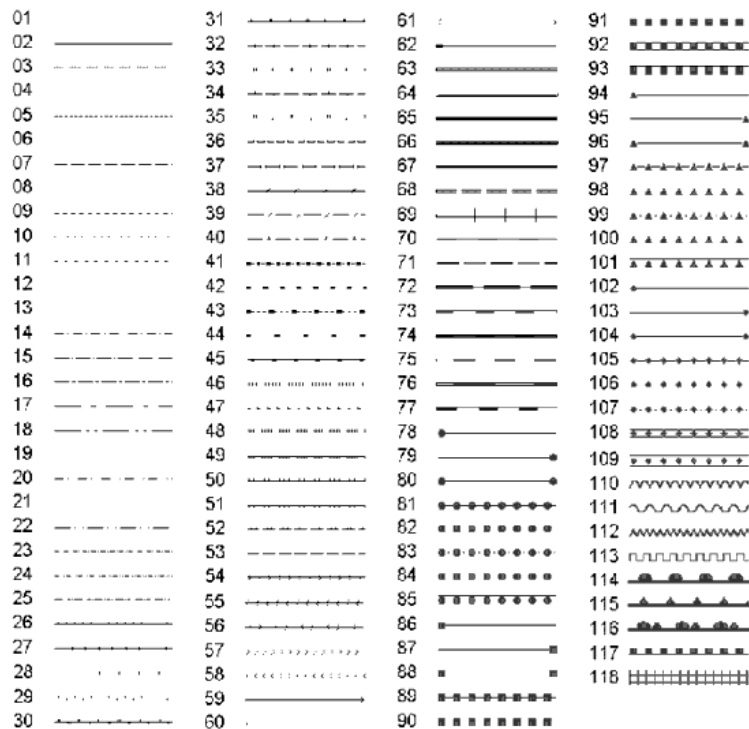
Некоторые операторы MapBasic используют выражения типа стиля контура **Pen** в качестве параметра (например, переменная типа **Pen** не используя при этом предложения **Pen**). Одним из примеров является **оператор Alter Object**.

В следующей таблице приводится описание параметров стиля линии:

Компонента стиля	Описание
width (толщина)	Толщина линии в точках, целочисленная величина типа Integer, от 1 до 7 включительно. Если требуется иметь невидимую линию в объекте, то задайте нулевую толщину. При этом, если параметр будет равен 0, то параметр типа линии pattern должен быть равен 1 (единице). Для задания толщины линии в точках, используется функция PointsToPenWidth(). При вычислениях размер в точках умножается на 10, затем к результату прибавляется еще 10, поэтому толщина линии всегда больше 10.

Компонента стиля	Описание
штриховка	Тип линии, целочисленная величина типа Integer, от 1 до 118 (смотрите таблицу ниже). Значение 1 обозначает невидимую линию. Для задания режима рисования с перекрытиями, прибавьте 128 к номеру типа линии. Не все штриховки, однако, выглядят лучше при рисовании с перекрытием.
color (цвет)	Целочисленный код цвета для рисунка штриха. Может быть заменен вызовом функции; см. раздел: Функция RGB().

Образцы штриховок приведены на рисунке ниже.



Примеры

```

Include "MAPBASIC.DEF"
Dim cable As Object
Create Line
  Into Variable cable
  (73.5, 42.6) (73.67, 42.9)
  Pen MakePen(1, 2, BLACK)

```

Пример назначения единообразного стиля линии к слою карты: толщина линии = 5 points; стиль линии – B17 в окошке выбора стилей линий; тип линии = 66.

```
Set Map Window <windowid>
Layer 1 Display Global
Global Line (60,194,16711680)
' Interleave: 194 = 66 + 128
Set Map Window 234499920
Layer 1 Display Global
Global Line MakePen(PointsToPenWidth(5),66,16711680)
' 5 point line, non-interleaved
```

См. также:

оператор **Alter Object**, функция **CreateLine()**, оператор **Create Pline**, функция **CurrentPen()**, функция **IsPenWidthPixels()**, функция **MakePen()**, функция **PointsToPenWidth()**, функция **PenWidthToPoints()**, функция **RGB()**, оператор **Set Style**

Функция PenWidthToPoints()

Назначение

Функция **PenWidthToPoints** возвращает толщину линии в американских типографских пунктах. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PenWidthToPoints ( penwidth )
```

penwidth – это целочисленное значение, задающее толщину линии.

Возвращаемая величина

Вещественное – Float

Описание

Функция **PenWidthToPoints()** возвращает значение толщины линии в пунктах. Толщину линии, использованную в стиле оформления линии, может возвращать функция **StyleAttr()**. Толщина линии, которую возвращает функция **StyleAttr()** может измеряться в пунктах или пикселях. В пунктах измеряется любая толщина линии, выраженная в единицах более десяти. **PenWidthToPoints()** будет возвращать значение только для линий, толщина которых задана в пунктах. **PenWidthToPoints()** будет возвращать значение только для линий, толщина которых

задана в пунктах. Чтобы определить в каких единицах задана толщина линии: пикселах или пунктах, используется [функция IsPenWidthPixels\(\)](#).

Пример

```
Include "MAPBASIC.DEF"
Dim CurPen As Pen
Dim Width As Integer
Dim PointSize As Float
CurPen = CurrentPen()
Width = StyleAttr(CurPen, PEN_WIDTH)
If Not IsPenWidthPixels(Width) Then
    PointSize = PenWidthToPoints(Width)
End If
```

См. также:

[функция CurrentPen\(\)](#), [функция IsPenWidthPixels\(\)](#), [функция MakePen\(\)](#), [предложение Pen](#), [функция PointsToPenWidth\(\)](#), [функция StyleAttr\(\)](#)

Функция Perimeter()

Назначение

Возвращает периметр графического объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Perimeter( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – строка, определяющая единицы измерения расстояния (к примеру, “км.”).

Возвращаемая величина

Вещественное – Float

Описание

Функция **Perimeter()** вычисляет периметр объекта *obj_expr*. Функция **Perimeter()** определена для следующих типов объектов: эллипсов, прямоугольников, скругленных прямоугольников и полигонов. Другие типы объектов имеют периметр нулевой длины.

Возвращаемое функцией **Perimeter()** значение длины периметра измеряется в единицах длины, определенных параметром *unit_name*; например, для получения длины в милях, укажите "mi" в качестве *unit_name*. Список допустимых единиц измерения смотрите в разделе: [Оператор Set Distance Units](#).

Функция **Perimeter()** возвращает приблизительные результаты, когда применяется к скругленным прямоугольникам. MapBasic вычисляет периметр прямоугольников со скругленными углами как если бы они были обычными прямоугольниками. В большинстве случаев MapInfo Pro проводит либо декартовы, либо сферические вычисления. Обычно выполняются сферические вычисления; если координатная система – план, то выполняются декартовы вычисления.

Пример

Следующий пример показывает как можно использовать функцию **Perimeter()** для определения периметра географического объекта.

```
Dim perim As Float
Open Table "world"
Fetch First From world
perim = Perimeter(world.obj, "km")
```

Переменная *perim* теперь содержит периметр полигона, который соответствует первой записи таблицы *World*.

Можно совместно использовать функцию **Perimeter()** внутри [оператора Select](#). Следующий [оператор Select](#) извлекает информацию из таблицы *States* и сохраняет результаты во временной таблице *Results*.

Поскольку [оператор Select](#) включает функцию **Perimeter()**, таблица результатов будет включать колонку, показывающую периметр каждого штата.

```
Open Table "states"
Select state, Perimeter(obj, "mi")
From states
Into results
```

См. также:

[функция Area\(\)](#), [функция ObjectLen\(\)](#), [оператор Set Distance Units](#)

Функция PieChartThemeInfo()

Назначение

Возвращает информацию об определённом тематическом круговом слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PieChartThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id `map_window_id` – идентификатор окна карты;

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический круговой слой карты.

attribute `SmallInt` *attribute* – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	PieChartThemeInfo() возвращает
PIECHARTTHEME_INFO_GRADUATION	1	Короткий целочисленный (<code>SmallInt</code>) результат, представляющий, как разность численных значений соответствуют разнице диаметров кругов. Поддерживается три метода. PIECHARTTHEME_INFO_GRAD_ROOT(0) PIECHARTTHEME_INFO_GRAD_LOG(1) PIECHARTTHEME_INFO_GRAD_LOG(2)

код атрибута	ID	PieChartThemeInfo() возвращает
PIECHARTTHEME_INFO_ORIENTATION	2	Короткий целочисленный (<code>SmallInt</code>) результат, представляющий положение нижней границы круговой диаграммы относительно центроидов каждого из объектов. Возможны следующие девять значений. PIECHARTTHEME_INFO_ORI_CC(0) – центр, центр. PIECHARTTHEME_INFO_ORI_TL(1) – верхняя граница, левая граница. PIECHARTTHEME_INFO_ORI_TC(2) – левая граница, центр. PIECHARTTHEME_INFO_ORI_TR(3) – верхняя граница, правая граница . PIECHARTTHEME_INFO_ORI_CL (4) – центр, левая граница. PIECHARTTHEME_INFO_ORI_CR(5) – центр, правая граница . PIECHARTTHEME_INFO_ORI_BL(6) – нижняя граница, левая граница. PIECHARTTHEME_INFO_ORI_BC(7) – нижняя граница, центр . PIECHARTTHEME_INFO_ORI_BR(8) – нижняя граница, правая граница.
PIECHARTTHEME_INFO_PEN	3	Предложение Pen, описывающее линию границы кругов диаграмм.
PIECHARTTHEME_GRADUATED	4	Логический (<code>Logical</code>) результат (Т или F), которым определяется, является ли круговая диаграмма размерной, градуированной или нет.
PIECHARTTHEME_HALF_PIE	5	Логический <code>Logical</code> результат (Т или F), которым определяется, будут ли отображаться только полукруги или нет (полные круги).
PIECHARTTHEME_DIAMETER	6	Вещественный (<code>Float</code>) результат, которым определяется численное значение, соответствующее диаметру кругов диаграммы в "бумажных" единицах.
PIECHARTTHEME_DIAMETER_UNIT	7	Строковый (<code>String</code>) результат, которым определяется единица измерения "бумажных" единиц диаметра круговой диаграммы.
PIECHARTTHEME_DIAMETER_VALUE	8	Вещественный (<code>Float</code>) результат, которым определяется численное значение, соответствующее "бумажным" единицам диаметра кругов диаграммы, используемых в .PIECHARTTHEME_DIAMETER.

код атрибута	ID	PieChartThemeInfo() возвращает
PIECHARTTHEME_START_ANGLE	9	Вещественный (Float) результат, которым определяется численное значение, соответствующее начальному углу круговой диаграммы в градусах.
PIECHARTTHEME_CLOCKWISE	10	Логический (Logical) результат (Т или F), которым определяется, в каком направлении от начального угла будут откладываться сектора круговой диаграммы – по часовой или нет.

См. также:

[Функция PieChartThemeItemInfo\(\)](#) на стр. 724

Функция PieChartThemeItemInfo()

Назначение

Возвращает информацию об определённой тематической характеристике на круговом тематическом слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PieChartThemeItemInfo ( map_window_id, thematic_layer_id, item_id,  
attribute )
```

map_window_id *map_window_id* – идентификатор окна карты;

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический слой диапазонов карты.

range_item_id Целое (`Integer`), которое идентифицирует тематическое выражение. Счётчик тематических выражений, использованных на этом слое, можно запрашивать с помощью команды [ThemeInfo\(\)](#).

attribute Код – короткое целое `SmallInt`, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	PieChartThemeItemInfo() возвращает
PIECHARTTHEME_ITEM_INFO_BRUSH	1	Предложение со стилем штриховки, использованной при отображении заданной характеристики с идентификатором ID круговой тематической картодиаграммы: выражение.

См. также:

[Функция PieChartThemeInfo\(\)](#) на стр. 722

Функция PointsToPenWidth()

Назначение

Преобразует значение, заданное в типографских пунктах, в ширину линии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PointsToPenWidth( pointsize )
```

pointsize – это десятичное число, представляющее значение в десятых долях пункта.

Возвращаемая величина

Короткое целое – SmallInt

Описание

Функция **PointsToPenWidth()** преобразует значение заданное в пунктах, в значение ширины линии.

Пример

```
Include "MAPBASIC.DEF"
Dim Width As Integer
Dim p_bus_route As Pen
```

```
Width = PointsToPenWidth(1.7)
p_bus_route = MakePen(Width, 9, RED)
```

См. также:

функция [CurrentPen\(\)](#) function, функция [IsPenWidthPixels\(\)](#), функция [MakePen\(\)](#), предложение [Pen](#), функция [PenWidthToPoints\(\)](#), функция [StyleAttr\(\)](#)

Функция PointToMGRS()

Назначение

Функция `PointToMGRS$( )` конвертирует координаты объекта, представляющего точку, в строковую величину координаты MGRS. Функция поддерживает только точечные объекты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PointToMGRS$( inputobject )
```

inputobject – выражение для точечного объекта.

Описание

MapInfo Pro автоматически преобразует координаты исходной точки в текущей системе координат MapBasic в координаты в проекции "Широта/Долгота" (WGS84) до преобразования в строку координат MGRS. Однако, так как по умолчанию, координатная система MapBasic – "Широта/Долгота" (без проекции), то при использовании такой промежуточной системы координат может возникнуть значительная потеря точности конечного результата, поскольку преобразования без использования сведений о проекции значительно снижают точность. Обычно, в качестве системы координат MapBasic либо задается "Широта/Долгота" (WGS84), либо оставляется исходная система координат таблицы, в этих случаях промежуточное преобразование не выполняется. Смотрите пример 2 ниже.

Возвращаемая величина

Строка – String

Примеры

Следующий пример демонстрирует использование функций `MGRSToPoint( )` и `PointToMGRS( )`

Пример 1:

```

dim obj1 as Object
dim s_mgrs As String
dim obj2 as Object

obj1 = CreatePoint(-74.669, 43.263)
s_mgrs = PointToMGRS$(obj1)
obj2 = MGRSToPoint(s_mgrs)

```

Пример 2:

```

Open Table "C:\Temp\MyTable.TAB" as MGRSfile

' When using the PointToMGRS$() or MGRSToPoint() functions,
' it is very important to make sure that the current MapBasic
' coordsys matches the coordsys of the table where the
' point object is being stored.

'Set the MapBasic coordsys to that of the table used
Set CoordSys Table MGRSfile

'Update a Character column (e.g. COL2) with MGRS strings from
'a table of points

Update MGRSfile
Set Col2 = PointToMGRS$(obj)

'Update two float columns (Col3 & Col4) with
'CentroidX & CentroidY information
'from a character column (Col2) that contains MGRS strings.

Update MGRSfile
Set Col3 = CentroidX(MGRSToPoint(Col2))

Update mgrstestfile ' MGRSfile
Set Col4 = CentroidY(MGRSToPoint(Col2))

Table MGRSfile
Close Table MGRSfile

```

См. также:**функция MGRSToPoint()**

Функция PointToUSNG\$()

Назначение

Конвертирует координаты объекта, представляющего точку, в строковую величину координаты USNG (United States National Grid). Функция поддерживает только точечные объекты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PointToUSNG$(obj, datumid)
```

obj – объект, представляющий точку для преобразования. Это может быть выражение, результатом которого является объект.

datumid – число, представляющее референц-эллипсоид. Допустимы следующие значения:

```
DATUMID_NAD27   (62)  
DATUMID_NAD83   (74)  
DATUMID_WGS84   (104)
```

Примечание: DATUMID_* определены в MapBasic.def. Значения WGS84 и NAD83 трактуются как одинаковые.

Описание

MapInfo Pro автоматически преобразует координаты исходной точки в текущей системе координат MapBasic в координаты в проекции "Широта/Долгота" (WGS84 и NAD27) до преобразования в строку координат USNG. Однако, так как по умолчанию, координатная система MapBasic – "Широта/Долгота" (без проекции), то при использовании такой промежуточной системы координат может возникнуть значительная потеря точности конечного результата, поскольку преобразования без использования сведений о проекции значительно снижают точность. Обычно, в качестве системы координат MapBasic либо задается "Широта/Долгота" (WGS84 и NAD27), либо оставляется исходная система координат таблицы, в этих случаях промежуточное преобразование не выполняется.

Возвращаемая величина

Строка – String

Пример 1

Следующий пример иллюстрирует использование функций `USNGToPoint()` и `PointToUSNG$()`.

```
dim obj1 as Object
dim s_USNG As String
dim obj2 as Object

obj1 = CreatePoint(-74.669, 43.263)
s_USNG = PointToUSNG$(obj1)
obj2 = USNGToPoint(s_USNG)
```

Пример 2

```
Open Table "C:\Temp\MyTable.TAB" as USNGfile

' When using the PointToUSNG$() or USNGToPoint() functions,
' it is very important to make sure that the current MapBasic
' coordsys matches the coordsys of the table where the
' point object is being stored.

'Set the MapBasic coordsys to that of the table used
Set CoordSys Table USNGfile

'Update a Character column (e.g. COL2) with USNG strings from
'a table of points

Update USNGfile
Set Col2 = PointToUSNG$(obj)

'Update two float columns (Col3 & Col4) with
'CentroidX & CentroidY information
'from a character column (Col2) that contains USNG strings.

Update USNGfile
Set Col3 = CentroidX(USNGToPoint(Col2))

Update USNGtestfile ' USNGfile
Set Col4 = CentroidY(USNGToPoint(Col2))

Table USNGfile
Close Table USNGfile
```

См. также:

[USNGToPoint\(string\)](#)

Оператор Print

Назначение

Печатает пояснительный текст или текст сообщения из программы в окне "Сообщение". Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Print message
```

string_expr – выражение, результатом которого является строка.

Описание

Оператор **Print** используется для вывода текста в окне "Сообщение". Это окно является одним из вспомогательных окон в MapInfo. Окно "Сообщение" предназначено для вывода информации из программ, написанных на MapBasic. Эти сообщения можно использовать для комментирования действий программы без ее остановки, например, "Запись удалена" или "Выберите область для анализа". Задать шрифт в окне "Сообщение" можно, используя **оператор Set Window**. Программа MapBasic может перед выполнением оператора **Print** открыть окно сообщений, используя **оператор Open Window**.

Если оператор **Print** выполняется, когда окно сообщений закрыто, то MapBasic откроет его автоматически. Оператор **Print** похож на **оператор Note**, и любой из них можно использовать, для вывода служебной или отладочной информации. Однако **оператор Note** сообщение в диалоговом окне, и пока пользователь не нажмет кнопку **OK**, выполнение программы будет приостановлено. Выполнение программы после оператора **Print** продолжается, а каждый следующий оператор **Print** будет печатать текст с новой строки в уже открытом окне "Сообщение". Если все окно сообщений будет заполнено или строка будет длиннее ширины окна, то пользователь может прокручивать окно в горизонтальном и вертикальном направлениях.

Чтобы очистить окно "Сообщение" перед выводом сообщения, используйте символ прогона листа (ASCII-код 12):

```
Print Chr$(12) 'This statement clears the Message window
```

Для начала новой строки в тексте сообщения используйте символ возврата каретки (ASCII-код 10). Следующий оператор **Print** выводит сообщение в две строки:

```
Print "Map Layers:" + Chr$(10) + " World, Capitals"
```

Оператор **Print** преобразует табуляцию (ASCII-код 09) в символ пробела (ASCII-код 32).

Пример

Откроем окно "Сообщение", назначим шрифт Arial Cyr, жирный, размером в 10 пунктов, синего цвета. Назначим также размер окна, 3 на 1 дюйм, расположение ниже и правее на четверть дюйма от правого верхнего угла основного окна MapInfo. Теперь можно печатать:

```
Include "MAPBASIC.DEF" ' needed for color name 'BLUE'
Open Window Message ' open Message window
Set Window Message
  Font ("Arial", 1, 10, BLUE) ' Arial bold...
  Position (0.25, 0.25) ' place in upper left
  Width 3.0 ' make window 3" wide
  Height 1.0 ' make window 1" high
Print "MapBasic Dispatcher now on line"
```

Примечание: Объем памяти, выделяемый для окна "Сообщение", был удвоен и теперь можно выводить 8191 символ.

См. также:

[функция Ask\(\)](#), [оператор Close Window](#), [оператор Note](#), [оператор Open Window](#), [оператор Set Window](#)

Оператор Print

Назначение

Записывает данные в файл, открытый в режиме последовательного доступа (в режимах **Output** или **Append**).

Синтаксис

```
Print # file_num [ , expr ]
```

file_num – номер файла, который открыл [оператор Open File](#).

expr – выражение для записи в файл.

Описание

Оператор **Print #** выводит данные в открытый файл. Файл должен быть открыт в режиме последовательного доступа, разрешающем запись (**Output** или **Append**).

Параметр *file_num* соответствует номеру из предложения **As**, который при открытии файла задает **оператор Open File**.

MapInfo Pro записывает выражение *expr* в одну строку файла. Для записи выражений списком через запятую, каждое в отдельную строку файла, используется **оператор Write #** вместо **Print #**.

См. также:

оператор Line Input, оператор Open File, оператор Write #

Оператор PrintWin

Назначение

Печатает содержимое окна. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PrintWin [ Window window_id ] [ Interactive ] [ File output_filename ]  
[ Overwrite ]
```

window_id – идентификатор окна, целое число.

имя_файла – строка, в которой указано имя выходного файла. Если файл уже существует и не указано ключевое слово **Overwrite**, произойдёт ошибка.

Описание

Оператор **PrintWin** используется для вывода содержимого окна на печать.

Если используется предложение **Window**, то MapBasic будет печатать заданное окно или рамку отчёта; если окно не задано, то напечатается содержимое активного окна или рамки отчёта.

Параметр *window_id* должен быть идентификатором окна, смотрите разделы: **функция FrontWindow()** и **функция WindowInfo()**, в которых получение идентификатора описано подробнее.

Для того чтобы получить идентификатор рамки с картой в окне **Отчёт**, используйте функцию **LayoutItemInfo()** с атрибутом LAYOUT_ITEM_INFO_WIN.

Если в команде PrintWin использовать ключевое слово **Interactive**, MapInfo Pro выведет диалог **Печать**. Без этого ключевого слова печать будет производиться автоматически, без диалога с пользователем, используя текущие установки печати.

Примеры

Пример 1

```
Dim win_id As Integer
Open Table "world"
Map From world
win_id = FrontWindow()
'
' knowing the ID of the Map window,
' the program could now print the map by
' issuing the statement:
'
PrintWin Window win_id Interactive
```

Пример 2

```
PrintWin Window FrontWindow() File "c:\output\file.plt"
```

См. также:

функция [FrontWindow\(\)](#), оператор [Run Menu Command](#), функция [WindowInfo\(\)](#)

Функция PrismMapInfo()

Назначение

Возвращает настройки окна карты-призмы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
PrismMapInfo( window_id, attribute )
```

window_id – идентификатор окна.

attribute – целочисленный код, указывающий, какой тип информации должен быть возвращен.

Возвращаемая величина

Вещественное (Float), целое (Integer), или строка символов (String), в зависимости от параметра атрибута.

Описание

Функция **PrismMapInfo()** возвращает информацию об окне Карты-призмы.

Параметр *window_id* определяет, какое окно Карты-призмы обрабатывается функцией. Для того чтобы получить идентификатор окна, используйте [функцию FrontWindow\(\)](#), которая вызывается сразу после открытия окна, а также [функцию WindowID\(\)](#), которую можно вызывать в любой момент после создания окна.

Существует несколько числовых атрибутов, которые **PrismMapInfo()** может вернуть для каждого окна карты-призмы. Параметр *attribute* сообщает функции **PrismMapInfo()**, какие данные об окне карты возвращаются. Коды *attribute* определены в файле `MAPBASIC.DEF`.

attribute-код	ID	Возвращаемая величина
PRISMMAP_INFO_SCALE	1	Вещественное (Float), масштабный фактор карты-призмы.
PRISMMAP_INFO_BACKGROUND	4	Целое (Integer), цвет фона (см. раздел Функция RGB()).
PRISMMAP_INFO_LIGHT_X	6	Вещественное (Float), координата X источника освещения.
PRISMMAP_INFO_LIGHT_Y	7	Вещественное (Float), координата Y источника света.
PRISMMAP_INFO_LIGHT_Z	8	Вещественное (Float), координата Z источника света.
PRISMMAP_INFO_LIGHT_COLOR	9	Целое (Integer), цвет источника, см. раздел Функция RGB() .
PRISMMAP_INFO_CAMERA_X	10	Вещественное (Float), координата X камеры.
PRISMMAP_INFO_CAMERA_Y	11	Вещественное (Float), координата Y камеры.
PRISMMAP_INFO_CAMERA_Z	12	Вещественное (Float), координата Z камеры.
PRISMMAP_INFO_CAMERA_FOCAL_X	13	Вещественное (Float), координата X фокальной точки камеры.
PRISMMAP_INFO_CAMERA_FOCAL_Y	14	Вещественное (Float), координата Y фокальной точки камеры.
PRISMMAP_INFO_CAMERA_FOCAL_Z	15	Вещественное (Float), координата Z фокальной точки камеры.
PRISMMAP_INFO_CAMERA_VU_1	16	Вещественное (Float), первое значение нормального вектора точки просмотра.

attribute-код	ID	Возвращаемая величина
PRISMMAP_INFO_CAMERA_VU_2	17	Вещественное (Float), второе значение нормального вектора точки просмотра.
PRISMMAP_INFO_CAMERA_VU_3	18	Вещественное (Float), третье значение нормального вектора точки просмотра.
PRISMMAP_INFO_CAMERA_VPN_1	19	Вещественное (Float), первое значение параметра нормали плоскости наблюдения.
PRISMMAP_INFO_CAMERA_VPN_2	20	Вещественное (Float), второе значение нормального вектора плоскости просмотра.
PRISMMAP_INFO_CAMERA_VPN_3	21	Вещественное (Float), третье значение нормального вектора плоскости просмотра.
PRISMMAP_INFO_CAMERA_CLIP_NEAR	22	Вещественное (Float), приближение камеры.
PRISMMAP_INFO_CAMERA_CLIP_FAR	23	Вещественное (Float), удаление камеры.
PRISMMAP_INFO_INFOTIP_EXPR	24	Строка (String) для всплывающей подсказки. Ранее не была документирована.

Пример

Распечатываем все стандартные переменные, определенные для окна карты-призмы:

```
include "Mapbasic.def"
Print "PRISMMAP_INFO_SCALE: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_SCALE)
Print "PRISMMAP_INFO_BACKGROUND: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_BACKGROUND)
Print "PRISMMAP_INFO_UNITS: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_UNITS)
Print "PRISMMAP_INFO_LIGHT_X : " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_LIGHT_X )
Print "PRISMMAP_INFO_LIGHT_Y : " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_LIGHT_Y )
Print "PRISMMAP_INFO_LIGHT_Z: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_LIGHT_Z)
Print "PRISMMAP_INFO_LIGHT_COLOR: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_LIGHT_COLOR)
Print "PRISMMAP_INFO_CAMERA_X: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_X)
```

```

Print "PRISMMAP_INFO_CAMERA_Y : " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_Y )
Print "PRISMMAP_INFO_CAMERA_Z : " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_Z )
Print "PRISMMAP_INFO_CAMERA_FOCAL_X: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_FOCAL_X)
Print "PRISMMAP_INFO_CAMERA_FOCAL_Y: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_FOCAL_Y)
Print "PRISMMAP_INFO_CAMERA_FOCAL_Z: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_FOCAL_Z)
Print "PRISMMAP_INFO_CAMERA_VU_1: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VU_1)
Print "PRISMMAP_INFO_CAMERA_VU_2: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VU_2)
Print "PRISMMAP_INFO_CAMERA_VU_3: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VU_3)
Print "PRISMMAP_INFO_CAMERA_VPN_1: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VPN_1)
Print "PRISMMAP_INFO_CAMERA_VPN_2: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VPN_2)
Print "PRISMMAP_INFO_CAMERA_VPN_3: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_VPN_3)
Print "PRISMMAP_INFO_CAMERA_CLIP_NEAR: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_CLIP_NEAR)
Print "PRISMMAP_INFO_CAMERA_CLIP_FAR: " + PrismMapInfo(FrontWindow(),
PRISMMAP_INFO_CAMERA_CLIP_FAR)

```

См. также:

оператор Create PrismMap, оператор Set PrismMap

Функция ProgramDirectory()

Назначение

Возвращает маршрут, по которому установлена рабочая версия MapInfo Pro. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ProgramDirectory$( )
```

Возвращаемая величина

Строка

Описание

Функция **ProgramDirectory\$()** возвращает в виде строки дисковый маршрут (носитель+каталог), по которому установлена рабочая версия MapInfo Pro.

Пример

```
Dim s_prog_dir As String
s_prog_dir = ProgramDirectory$()
```

См. также:

[функция HomeDirectory\(\)](#), [функция SystemInfo\(\)](#)

Оператор ProgressBar

Назначение

Показывает диалог "Минуточку!" с кнопкой **Отмена** и горизонтальным индикатором состояния.

Синтаксис

```
ProgressBar status_message  
Calling handler  
[ Range n ]
```

status_message – строка, которая будет выводиться в диалоге.

handler – имя подпрограммы;

n – число, при котором выполнение программы завершится.

Предупреждение!

Нельзя использовать оператор **ProgressBar** в окне **MapBasic**.

Описание

Оператор **ProgressBar** используется для создания диалогового окна-индикатора выполнения процесса. Такой диалог снабжен процентной шкалой и кнопкой **Отмена**. Диалог сопровождает определенные действия и показывает, насколько они выполнены. Выполнение можно прервать, нажав кнопку **Отмена**. Информацию о том, как завершился процесс – самостоятельно или

был прерван, можно узнать при помощи функции `CommandInfo (CMD_INFO_DLG_OK)`, вызвав ее после оператора **ProgressBar**, где `CMD_INFO_DLG_OK (1)`.

Строковый параметр *status_message* может задавать поясняющий текст, который будет отображен внутри диалога процесса над шкалой. Например, это может быть: "Идет процесс копирования..."

Параметр *handler* является именем процедуры-обработчика, выполняющей те действия, которые сопровождаются диалогом процесса. Как будет описано ниже, подпрограмма должна выполнить определенные действия, чтобы взаимодействовать с оператором **ProgressBar**.

Параметр *n* задает число, которое будет стоять справа у шкалы в окне диалога. Например, процедура-обработчик проверяет 7000 строк таблицы. В операторе **ProgressBar** параметр *n* задается равным семи тысячам. Теперь шкала отображает информацию о построчном выполнении процедуры. Если предложение `Range n` отсутствует, то по умолчанию параметр *n* устанавливается равным 100.

Когда программа выполняет оператор **ProgressBar**, MapBasic вызывает подпрограмму *handler*. Работа, выполняемая этой процедурой, должна быть разбита на небольшие по длительности отрезки, не более нескольких секунд каждый. Это делается для того, чтобы пользователь мог нажать на кнопку **Отмена**, после чего MapBasic удалит с экрана диалог, и выполнение программы передается следующему после **ProgressBar** оператору. Если же пользователь не нажал на кнопку **Отмена**, MapBasic продолжит выполнение процедуры-обработчика *handler*. Если пользователь ни разу не нажал кнопку **Отмена**, то оператор **ProgressBar** повторно вызывает процедуру-обработчик до победного конца.

Поэтому в тексте процедуры-обработчика *handler* должны быть предусмотрены средства для разбиения процесса на небольшие отрезки, а также средства слежения за возобновлением процесса. Пока оператор **ProgressBar** работает, MapBasic периодически обращается к процедуре-обработчику *handler* до тех пор, пока пользователь не нажмет кнопку **Отмена**, либо *handler* не закончит работу. Для контроля за этими событиями поддерживается специальная переменная, имеющая имя *ProgressBar*.

Если обработчик *handler* присвоит этой переменной значение "минус единица" (`ProgressBar = -1`), то MapBasic прерывает процесс и убирает диалог с экрана. И, наоборот, любое положительное значение *handler* переменной *ProgressBar*, например, (`ProgressBar = 50`) используется MapBasic для отображения "процента выполнения" в диалоге. MapBasic вычисляет "процент выполнения" делением текущего значения переменной значение переменной `Range`. MapBasic вычисляет "процент выполнения" делением текущего значения переменной *ProgressBar* на значение переменной `Range`, *n*. Например, если через оператор **ProgressBar** переменной **Range** присвоено значение `Range 400`, то, если значение *ProgressBar* равно 100, "процент выполнения" будет равен 25% и это будет отображено в диалоге.

В выражениях, следующих за оператором **ProgressBar**, Вы можете определить причину окончания процесса, описанного в *ProgressBar*: либо нормальное завершение, либо нажатие на кнопку **Отмена**. Для этого оператора **ProgressBar** используется функция `CommandInfo (CMD_INFO_DLG_OK)` возвращающая значение TRUE, если процесс завершился нормально, и FALSE, если процесс был прерван пользователем. где `CMD_INFO_DLG_OK (1)`.

Пример

Этот пример демонстрирует, как должна быть написана процедура, вызываемая оператором **ProgressBar**. Процесс в этом примере состоит из 600 итераций, это может быть однотипная обработка 600 строк таблицы. Оператор **ProgressBar** вызывает из основной процедуры sub-процедуру "write_out". Sub-процедура "write_out" обрабатывает записи в течение 2 секунд. Затем она возвращает управление, и MapBasic проверяет, не была ли нажата кнопка **Отмена**. Если пользователь не нажимал на кнопку **отмены**, то MapBasic возвращает управление в процедуру-обработчик. Такая последовательность действий повторяется до тех пор, пока не будет выполнена вся задача.

```

Include "mapbasic.def"
Declare Sub Main
Declare Sub write_out

Global next_row As Integer

Sub Main
  next_row = 1
  ProgressBar "Writing data..." Calling write_out Range 600
  If CommandInfo(CMD_INFO_STATUS) Then
    Note "Operation complete! Thanks for waiting."
  Else
    Note "Operation interrupted!"
  End If
End Sub

Sub write_out
  Dim start_time As Float
  start_time = Timer( )
  ' process records until either (a) the job is done,
  ' or (b) more than 2 seconds elapse within this call
  Do While next_row <= 600 And Timer( ) - start_time < 2
    .....
    ''' Here, we would do the actual work '''
    ''' of processing the file.           '''
    .....
    next_row = next_row + 1
  Loop

  ' Now figure out why the Do loop terminated: was it
  ' because the job is done, or because more than 2
  ' seconds have elapsed within this iteration?
  If next_row > 600 Then
    ProgressBar = -1 'tell caller "All Done!"
  Else
    ProgressBar = next_row 'tell caller "Partly done"
  End If
End Sub

```

См. также:

функция CommandInfo(), оператор Note, оператор Print

Функция Proper\$ ()

Назначение

Возвращает строку, преобразуя все первые буквы слов в прописные, а остальные в строчные. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Proper$( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **Proper\$()** возвращает строку, полученную из строки, представленной выражением *string_expr*, преобразованием всех первых букв слов в прописные и остальных в строчные. Такое преобразование имеет смысл для собственных имен и названий.

Пример

```
Dim name, propername As String

name = "ed bergen"
propername = Proper$(name)
' propername now contains the string "Ed Bergen"

name = "ABC 123"
propername = Proper$(name)
' propername now contains the string "Abc 123"

name = "a b c d"
propername = Proper$(name)
' propername now contains the string "A B C D"
```

См. также:

[функция LCase\(\)](#), [функция UCase\(\)](#)

Функция ProportionOverlap()

Назначение

Вычисляет процент перекрытия одного объекта другим. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ProportionOverlap( object1, object2 )
```

object1 – нижний объект, он должен быть замкнутым.

object2 – верхний объект, он должен быть замкнутым.

Возвращаемая величина

Float, вещественное число равно **AreaOverlap**(*object1*, *object2*) / **Area**(*object1*).

Предупреждение!

Функция ProportionOverlap() работает только с замкнутыми объектами. Если один или оба объекта незамкнуты (например, точки и линии), показывается сообщение об ошибке. Замкнутые объекты – это площадные объекты, такие как области.

См. также:

функция **AreaOverlap()**

Оператор Put

Назначение

Записывает в открытый файл содержимое переменной.

Синтаксис

```
Put [ # ] filenum, [ position, ] var_name
```

filenum – целое число, присвоенное файлу **оператором Open File**.

position – позиция файла для записи (не для последовательного доступа).

var_name – имя переменной, значение которой будет использовано как данные для записи.

Описание

Оператор **Put** позволяет записывать значение переменной в открытый файл.

Примечание: Если **оператор Open File** открывает файл в режиме последовательного доступа (**Output** или **Append**), то для записи в файл используйте либо **оператор Print #**, либо **оператор Write #** вместо оператора **Put**.

Если **оператор Open File** открыл файл в режиме произвольного доступа **Random**, то параметр **Position** оператора **Put** задает номер записи, в которую будет записано новое значение. Сразу после открытия файла позиция для записи – это первая запись файла. Если **оператор Open File** открыл файл в режиме бинарного доступа (Binary), то за один раз можно записывать только одну переменную. Порядок, по которому байты записываются в файл, зависит от вычислительной платформы (см. также предложение **ByteOrder** в разделе: **Оператор Open File**). Количество записанных байтов зависит от типа переменной, см. следующую таблицу:

Тип переменной	Хранение в файле
Логическое – Logical.	Однобайтовое значение: либо 0, либо другое ненулевое число.
Короткое целое – SmallInt	Целое число, представленное двумя байтами.
Целое – Integer	Целое число, представленное четырьмя байтами.
Вещественное – Float	Вещественное число, представленное восьми байтами в формате IEEE.
Строка – String	Строка плюс один байт для нулевого значения, обозначающего конец строки.
Дата – Date	Дата, представленное 4 байтами: короткое целое число для года, байт для месяца и байт для дня.
Другие типы	Не записываются оператором Put.

Параметр **Position** задает смещение указателя записи в файле в байтах. Сразу после открытия файла позиция для записи – это первый байт файла (начало файла). При работе оператора **Put** автоматически наращивает смещение по байтам. Если параметр **Position** не задан, то оператор **Put** начнет запись с текущей позиции. В файл, открытый в режиме Binary оператор **Put** не может записывать строки неопределенной длины; все переменные типа String должны

иметь в операторе **Put** фиксированную длину. В файл, открытый в режиме Random оператор **Put** не может записывать строку, которая длиннее длины записи в файле.

См. также:

функция EOF(), оператор Get, оператор Open File, оператор Print #, оператор Write #

Оператор Randomize

Назначение

Инициализирует функцию случайных чисел в MapBasic. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Randomize [ With seed ]
```

seed – целочисленное выражение.

Описание

Оператор **Randomize** включает генератор случайных чисел, результат которого используется, когда будет вызываться функция **Rnd()**. Если оператор Randomize не выполнен, то функция **Rnd()** будет образовывать хотя и случайную, но всегда одинаковую последовательность. Это происходит потому, что без оператора **Randomize** каждое новое значение, которое создает функция **Rnd()**, зависит от предыдущего значения случайной последовательности.

Оператор **Randomize** изменяет начальное значение для функции **Rnd()**, и потому все последующие вызовы этой функции **Rnd()** будут образовывать другую случайную последовательность.

Если оператор Randomize используется с предложением **With**, то для создания начального случайного значения используется псевдослучайный генератор с входным значением *seed*. Если предложение **With** отсутствует, для построения начального случайного числа используется таймер Вашего компьютера. Используйте параметр **With**, если требуется повторяющийся в тестах набор псевдослучайных исходных данных, программа будет создавать "случайные" числа, последовательность которых будет постоянно повторяться.

Пример

```
Randomize
```

См. также:

функция Rnd()

Функция RangeThemeInfo()

Назначение

Возвращает информацию о слое в окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
RangeThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id `map_window_id` – идентификатор окна карты;

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический слой диапазонов карты.

attribute `SmallInt` – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	RangeThemeInfo() возвращает
RANGETHHEME_INFO_ITEMS_COUNT	1	Короткий целочисленный результат (<code>SmallInt</code>), в котором представляется счётчик количества тематических диапазонов (за исключением диапазона со "всеми остальными" значениями – All others).

код атрибута	ID	RangeThemeInfo() возвращает
RANGETHHEME_INFO_VARY_STYLE	2	Короткий целочисленный результат (<code>SmallInt</code>), в котором представляется, как применяются стили. Возможны следующие значения. RANGETHHEME_INFO_VARY_BY_ALL(0) – используются все атрибуты стилей. RANGETHHEME_INFO_VARY_BY_COLOR(1) – используются только стили, связанные с цветным оформлением. RANGETHHEME_INFO_VARY_BY_SIZE(2) – используются только стили, связанные с размерами.
RANGETHHEME_INFO_REPLACE_STYLE	3	Логический результат (<code>Logical</code> – Т или F), определяющий заменяет ли стиль оформления тематических диапазонов стили подлежащих объектов или нет.
RANGETHHEME_OTHERS_INFO_LINE	4	Предложение стиля линии, используемое для отображения объектов, принадлежащих исключительному диапазону, состоящему из всех остальных объектов – All others. Если тематический слой построен не на линейной тематике, возвращается стандартный стиль линии с цветом основы преобладающих геометрических примитивов (областей или точек) стиля исключительного элементов (All others).
RANGETHHEME_OTHERS_INFO_PEN	5	Предложение Pen, описывающее стиль линии границы объектов, которое используется для отображения объектов, принадлежащих диапазону остальных объектов – All others. Если тематический слой построен не на тематике областей, возвращается стандартный стиль линии: Pen(1, 2, 0).
RANGETHHEME_OTHERS_INFO_BRUSH	6	Предложение стиля штриховки области, используемое для отображения объектов, принадлежащих исключительному диапазону, состоящему из всех остальных объектов – All others. Если тематический слой построен не на тематике областей, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (точек или линий) стиля исключительного диапазона All others.
RANGETHHEME_OTHERS_INFO_SYMBOL	7	Предложение стиля символа, используемое для отображения объектов, принадлежащих исключительному диапазону, состоящему из всех остальных объектов – All others. Если тематический слой построен не на точечной тематике, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (областей или линий) стиля исключительного диапазона All others.

код атрибута	ID	RangeThemeInfo() возвращает
RANGETHHEME_INFO_BASE_LINE	8	Предложение Line , описывающее стиль оформления линий, используется в качестве базового стиля. Если тематический слой построен не на основе линий, то появляется сообщение об ошибке.
RANGETHHEME_INFO_BASE_BRUSH	9	Предложение Brush , описывающее стиль оформления площадных объектов, используется в качестве базового стиля. Если тематический слой построен не на основе областей, то появляется сообщение об ошибке.
RANGETHHEME_INFO_BASE_SYMBOL	10	Предложение Symbol , описывающее стиль оформления точечных условных знаков, используется в качестве базового стиля. Если тематический слой построен не на основе точек, то появляется сообщение об ошибке.

См. также:

Функция [RangeThemeItemInfo\(\)](#) на стр. 746

Функция RangeThemeItemInfo()

Назначение

Возвращает информацию об определённой тематическом диапазоне карты на тематическом слое диапазонов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
RangeThemeItemInfo ( map_window_id, thematic_layer_id, range_item_id,  
attribute )
```

map_window_id `map_window_id` – идентификатор окна Карты;

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический слой диапазонов карты.

range_item_id Целое (`Integer`), которое идентифицирует тематическое выражение.

attribute Код – короткое целое (`SmallInt`), см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	RangeThemeItemInfo() возвращает
RANGETHHEME_ITEM_INFO_LINE	1	Предложение стиля линии, использующееся для отображения объектов, принадлежащих заданному по идентификатору диапазону. Если тематический слой построен не на линейной тематике, возвращается стандартный стиль линии: Pen (1, 2, color), где параметр цвета color относится к цвету преобладающих геометрических примитивов (областей или точек) стиля диапазонов.
RANGETHHEME_ITEM_INFO_PEN	2	Предложение Pen , описывающее стиль линии, которое используется для отображения объектов, принадлежащих заданному по идентификатору диапазону. Если тематический слой построен не на тематике областей, возвращается стандартный стиль линии Pen(1, 2, 0).
RANGETHHEME_ITEM_INFO_BRUSH	3	Предложение Brush , описывающее стиль штриховки, которое используется для отображения объектов, принадлежащих заданному по идентификатору диапазону. Если тематический слой построен не на тематике областей, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (точек или линий) стиля диапазона.
RANGETHHEME_ITEM_INFO_SYMBOL	4	Предложение стиля символа, которое используется для отображения объектов, принадлежащих заданному по идентификатору диапазону. Если тематический слой построен не на точечной тематике, возвращается стандартный стиль с цветом преобладающих геометрических примитивов (областей или линий) стиля диапазона .
RANGETHHEME_ITEM_INFO_FROM	5	Возвращается значение (Small Integer, или Integer, или Float, или Time, или Date, или Datetime и т.п.), представляющее минимальное значение заданного по идентификатору диапазона.

код атрибута	ID	RangeThemeItemInfo() возвращает
RANGETHHEME_ITEM_INFO_TO	6	Возвращается значение (Small Integer, или Integer, или Float, или Time, или Date, или Datetime и т.п.), представляющее максимальное значение заданного по идентификатору диапазона.

См. также:

[Функция RangeThemeInfo\(\)](#) на стр. 744

Функция RasterTableInfo()

Назначение:

Возвращает информацию о растровой таблице, таблице поверхностей (Grid) или о таблице WMS. (Таблицы WMS, Tile Server и Seamless Raster не поддерживаются).

Синтаксис

```
RasterTableInfo( table_id, attribute )
```

table_id – либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0. Таблицы должна быть растром или поверхностью.

attribute – целочисленный код, определяющий тему запроса.

Возвращаемая величина

Величина типа String, Integer, SmallInt или Logical, в зависимости от значения параметра attribute.

Параметр attribute определяет вид возвращаемой функцией информации. Коды в левом столбце (например, RASTER_TAB_INFO_IMAGE_NAME) определены в файле стандартных определений MAPBASIC.DEF.

код атрибута	ID	RasterTableInfo() возвращает
RASTER_TAB_INFO_IMAGE_NAME	1	Строка (String), представляющая название файла, под которым изображение зарегистрировано в данной растровой таблице.

код атрибута	ID	RasterTableInfo() возвращает
RASTER_TAB_INFO_WIDTH	2	Целое число (Integer), представляющее ширину изображения в пикселах.
RASTER_TAB_INFO_HEIGHT	3	Целое число (Integer), представляющее высоту изображения в пикселах.
RASTER_TAB_INFO_IMAGE_TYPE	4	Короткое целое (SmallInt), представляющее тип изображения: • IMAGE_TYPE_RASTER (0) для растров • IMAGE_TYPE_GRID (1) для поверхностей
RASTER_TAB_INFO_BITS_PER_PIXEL	5	Короткое целое (SmallInt), представляющее количество бит/пикселей, составляющих изображение.
RASTER_TAB_INFO_IMAGE_CLASS	6	Короткое целое (SmallInt), представляющее класс изображения: • IMAGE_CLASS_PALETTE (2) для изображений на основе палитры • IMAGE_CLASS_GREYSCALE (1) для изображений в оттенках серого • IMAGE_CLASS_RGB (3) для RGB-изображений • IMAGE_CLASS_BILEVEL (0) для двухцветных (черно-белых) изображений
RASTER_TAB_INFO_NUM_CONTROL_POINTS	7	Короткое целое (SmallInt), представляющее количество контрольных точек. Используйте функцию ControlPointInfo() , чтобы получить определённые контрольные точки.
RASTER_TAB_INFO_BRIGHTNESS	8	Короткое целое (SmallInt), представляющее яркость в процентах (в диапазоне 0-100%).
RASTER_TAB_INFO_CONTRAST	9	Короткое целое (SmallInt), представляющее контрастность в процентах (в диапазоне 0-100%).

код атрибута	ID	RasterTableInfo() возвращает
RASTER_TAB_INFO_GREYSCALE	10	Логическое (Logical) значение, обозначающее использование тонов серого вместо стандартных цветов изображения.
RASTER_TAB_INFO_DISPLAY_TRANSPARENT	11	Логическое (Logical) значение, наличие прозрачного цвета. Если возвращается TRUE, то такой цвет обозначает RASTER_TAB_INFO_TRANSPARENT_COLOR.
RASTER_TAB_INFO_TRANSPARENT_COLOR	12	Целое число (Integer), представляющее цвет, назначенный прозрачным в RGB.
RASTER_TAB_INFO_ALPHA	13	Короткое целое (SmallInt), представляющее альфа-фактор для прозрачности растра (в диапазоне 0-255).
RASTER_TAB_INFO_HANDLERNAME	14	Строковый результат, представляющий имя растра и/или заголовка обработчика регулярной сети. Этот только имя заголовка-обработчика без маршрута адреса.
RASTER_TAB_INFO_LEGEND	15	Логический результат, представляющий состояние – может ли изображение поддерживать легенду.
RASTER_TAB_INFO_LEGEND_XML	16	Строковый результат, представляющий XML-строку растровой легенды. Это пустая строка, если не имеется соответствующего XML.
RASTER_TAB_INFO_HANDLERFILLPATH	17	Строковый результат, представляющий полный маршрут адреса растра и/или заголовка обработчика регулярной сети.

Примеры

В этом примере возвращается имя растра и/или заголовка-обработчика регулярной сети.

```
print RasterTableInfo("tablename", 14)
```

В этом примере возвращается значение TRUE, если растровое изображение поддерживает легенду.

```
print RasterTableInfo("tablename", 15)
```

См. также:

оператор **Create Grid**, функция **GetGridCellValue()**, функция **GridTableInfo()**, функция **IsGridCellNull()**, функция **OverlayNodes()**

Функция RegionInfo()

Назначение

Эта функция предназначена для определения направления обхода узлов полигона – по-часовой или против часовой. Единственный атрибут, который возвращает эта функция – "направление обхода" узлов заданного полигона. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис,

```
RegionInfo( object, REGION_INFO_IS_CLOCKWISE, polygon_num )
```

где:

```
REGION_INFO_IS_CLOCKWISE      1
```

object – выражение, задающее объект;

REGION_INFO_IS_CLOCKWISE – параметр, задающий ориентацию объекта (по часовой стрелке или против). Значение 1 означает ориентацию по часовой стрелке.

polygon_num – число представляющее номер компоненты полигона, состоящего из нескольких частей.

Пример:

Если на карте Соединенных Штатов Америки "States.TAB" выбрать штат Юта (Utah) и выполнить в окне **MapBasic** эту команду, то результатом будет **F** или **False**, поскольку узлы единственного полигона штата Юта имеют направление обхода против часовой. Узлы полигона штата Колорадо (Colorado) имеют направление обхода по-часовой и, в этом случае, будет возвращено значение **T** или **True**.

```
print RegionInfo(selection.obj,1,1)
```

Функция ReadControlValue()

Назначение

Считывает состояние элемента управления активного диалога.

Синтаксис

```
ReadControlValue( id_num )
```

id_num – целочисленный идентификатор элемента диалога.

Возвращаемая величина

Integer, Logical, String, Pen, Brush, Symbol или Font, в зависимости от типа элемента управления.

Описание

Функция **ReadControlValue()** возвращает текущее значение одного из элементов активного диалога. Так как функция **ReadControlValue()** может считывать значения только из открытого диалога, то функцию **ReadControlValue()** можно употреблять только в процедурах-обработчиках элементов диалога.

Параметр *id_num* должен задавать идентификатор элемента, значение которого прочтает функция. Если значение *id_num* равно -1 (минус единица) функция **ReadControlValue()** вернет значение элемента, который на текущий момент был изменен последним. Чтобы получить информацию о заданном элементе диалога, целочисленное значение идентификатора диалога необходимо передать функции **ReadControlValue()**.

Примечание: Диалог не будет иметь уникального идентификатора, если не включить параметр **ID оператора Dialog** в предложении **Control**. Некоторые управляющие элементы диалога не имеют атрибутов, которые можно было бы считать (например, статические надписи).

В таблице ниже перечислены типы возвращаемых значений разных элементов диалога. Обратите внимание, что для элемента MultiListBox существует специальная процедура чтения текущего значения. Так как пользователь может одновременно выбрать сразу несколько строк в списке элемента MultiListBox, то чтение значений этого элемента требует многократного вызова функции **ReadControlValue()**.

Элемент диалога	Возвращаемое значение ReadControlValue()
EditText – окошко ввода текста	String, строка длиной до 32767 байт, содержащая текст из окошка; если окошко элемента многострочное, то текст может содержать символы конца строки Chr\$(10), что означает, что пользователь ввел перевод строки (например, в Windows, нажал Ctrl+Enter).
CheckBox – флажки	Логическое: TRUE, если флажок установлен, и FALSE, если сброшен.
DocumentWindow	Целое число идентификатора HWND элемента управления окна. Этот идентификатор должен передаваться в качестве маркера родительского окна в оператор Set Next Document .
RadioGroup – группа кнопок-переключателей	SmallInt, короткое целое число, номер выбранной кнопки (начиная с 1).
PopupMenu – "всплывающее" меню	SmallInt, короткое целое число, номер выбранной строки в списке меню (начиная с 1).
ListBox – список	SmallInt, короткое целое число, номер выбранной строки в списке (1 – первая строка, 0 – ни одной).
BrushPicker	Величина типа Brush (стиль штриховки).
FontPicker	Величина типа Font (шрифт).
PenPicker	Величина типа Pen (стиль линии).
SymbolPicker	Величина типа Symbol (символ).
MultiListBox – список множественного выбора	Integer, целое число, номер выбранной строки в списке. Пользователь может выбрать одну или сразу несколько строк в списке MultiListBox. Так как пользователь может одновременно выбрать несколько строк в списке элемента MultiListBox, то чтение значений этого элемента требует многократного вызова функции ReadControlValue(). Первый вызов функции ReadControlValue() дает номер первой строки, выбранной в списке; следующий – второй, и т. д. Когда все значения будут прочитаны функция ReadControlValue() вернет ноль. Следующий за этим вызов ReadControlValue() начнет все сначала, с вершины списка. Если ноль будет получен при первом вызове функции ReadControlValue() то, следовательно, в списке ничего не выбрано.

Ошибки

ERR_FCN_ARG_RANGE (644), если неправильно задано значение аргумента.

ERR_INVALID_READ_CONTROL (842), если функция **ReadControlValue()** вызвана не при активном диалоге.

Пример

Создадим диалог, предлагающий пользователю ввести свое имя. Если после этого пользователь нажмет кнопку **ОК**, программа вызовет функцию **ReadControlValue()** считает это значение.

```
Declare Sub Main
Declare Sub okhandler
Sub Main
  Dialog
    Title "Sign in, Please"
    Control OKButton
      Position 135, 120 Width 50
      Title "OK"
      Calling okhandler
    Control CancelButton
      Position 135, 100 Width 50
      Title "Cancel"
    Control StaticText
      Position 5, 10
      Title "Please enter your name:"
    Control EditText
      Position 55, 10 Width 160
      Value "(your name here)"
      Id 23 'arbitrary ID number
  End Sub
Sub okhandler
  ' this sub is called when/if the user
  ' clicks the OK control
  Note "Welcome aboard, " + ReadControlValue(23) + "!"
End Sub
```

См. также:

[оператор Alter Control](#), [оператор Dialog](#), [оператор Dialog Preserve](#), [оператор Dialog Remove](#)

Оператор ReDim

Назначение

Изменяет размерность массива.

Синтаксис

```
ReDim var_name ( newsize ) [ , ... ]
```

var_name – имя массива локальных или глобальных переменных.

newsize – целое число, задающее новую размерность. Максимум – 32 767.

Описание

Оператор **ReDim** изменяет размерность, (то есть количество элементов) одного или более уже объявленных массивов. Имя массива переменных, задано параметром *var_name*, для объявления которого необходимо предварительно использовать либо **оператор Dim**, либо **оператор Global**.

Оператор **ReDim** может либо увеличивать, либо уменьшать размер существующего массива. Если программе более не требуется массив заданного размера оператор **ReDim** может уменьшить размер массива до нуля (этим можно минимизировать объем памяти, необходимый для хранения переменных).

В отличие от других BASIC-языков, MapBasic не позволяет задавать произвольный номер первого элемента массива. Другими словами, первый элемент массива в MapBasic всегда имеет номер 1 (единица).

Если переменные хранились в массиве, а затем размер массива был увеличен оператором **ReDim**, хранившиеся значения не меняются.

Пример

```
Dim names_list(10) As String, cur_size As Integer
' The following statements determine the current
' size of the array, and then ReDim the array to
' a size 10 elements larger

cur_size = UBound(names_list)
ReDim names_list(cur_size + 10)

' The following statement ReDims the array to a
' size of zero elements. Presumably, this array
' is no longer needed, and it is resized to zero
' for the sake of saving memory.

ReDim names_list(0)
```

Ниже показывается, как оператор **ReDim** может применяться к массивам переменных сложного типа, составленным при помощи оператора **Type**, и к массивам, которые являются элементами переменных сложного типа.

```
Type customer
  name As String
  serial_nums(0) As Integer
End Type

Dim new_customers(1) As customer

' First, redimension the "new_customers" array,
' making it five items deep:

ReDim new_customers(5)

' Now, redimension the "serial_nums" array element
' of the first item in the "new_customers" array:

ReDim new_customers(1).serial_nums(10)
```

См. также:

[оператор Dim](#), [оператор Global](#), [функция UBound\(\)](#)

Оператор Refresh Designer Frame

Назначение

MapBasic-программы могут обновлять рамки с изображениями в окне **Отчёт**, когда изменяется исходный файл с изображением. Этот оператор также может использоваться с текстовыми рамками. Когда он применяется к текстовым рамкам, происходит повторное выполнение любых выражений в динамической тексте, встречающиеся в тексте. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Refresh Designer Frame
[ Window window_id ]
ID frame_id [, frame_id ...]
```

window_id — идентификатор окна **Отчёт**, целое число; Если окно не задано, то команда будет применяться к самому верхнему окну **Отчёт**.

frame_id — число, указывающее какую рамку в окне **Отчёт** требуется опросить. Разделы пронумерованы от 1 до n, где n это номер рамки в отчёте.

См. также:

оператор **Add Designer Frame**, оператор **Alter Designer Frame**, оператор **Create Designer Legend**, оператор **Set Designer Legend**

Оператор Register Table

Назначение

Оператор **Register Table** преобразует электронные таблицы, таблицы баз данных, текстовые файлы, табличные геоданные GeoPackage, растровые снимки и файлы регулярных поверхностей в таблицы MapInfo Pro. Оператор проверяет неродной файл, например, dBASE-файл и создает TAB-файл. Только после этого MapInfo Pro может получить доступ к данным в этом файле, как таблице.

Оператор **Register Table** не меняет данные в неродном файле-источнике. Он определяет тип данных колонок файла и создает TAB-файл. Для того чтобы открыть новую таблицу, требуется использовать оператор **Open Table**. Регистрировать каждый файл необходимо только один раз.

Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Register Table source_file {
  Type NATIVE |
  Type DBF [ Charset char_set ] |
  Type ASCII [ Delimiter delim_char ] [ Titles ]
    [ CharSet char_set ] |
  Type WKS [ Titles ] [ Range range_name ] |
  Type WMS Coordsys... |
  Type WFS [ Charset char_set ] Coordsys... [ Symbol... ]
    [ Linestyle Pen(...) ] [ Regionstyle Pen(...) Brush(...) ]
    [ Editable ] |
  Type XLS [ Charset char_set ] [ Titles ] [ Range range_name ] [
Interactive ] |
  Type ACCESS Table table_name
    [ Password pwd ] [ CharSet char_set ] |
  Type ODBC
    Connection { Handle connection_number | connection_string }
    Toolkit toolkit_name
    [ Charset char_set ]
    [ Cache { ON | OFF } [ Type { NATIVE | NATIVEX } ] ]
    [ Autokey { ON | OFF } ]
  Table SQLQuery
    [ Versioned { ON | OFF } ]
```

```

[ Workspace Workspace_name ]
[ ParentWorkspace ParentWorkspace_name ]
[ Symbol... ] [ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ] |
Type GRID | Type RASTER
[ ControlPoints ( MapX1, MapY1 ) ( RasterX1, RasterY1 ),
  ( MapX2, MapY2 ) ( RasterX2, RasterY2 ),
  ( MapX3, MapY3 ) ( RasterX3, RasterY3 )
  [ , ... ] ]
[ CoordSys... ]
[ Charset char_set ] |
Type FME [ Charset char_set ]
CoordSys...
Format format_type
Schema feature_type
[ Use Color ]
[ Database ]
[ SingleFile ]
[ Symbol... ]
[ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
[ Font... ]
Settings string1 [ , string2 .. ] |
Type SHAPEFILE [ Charset char_set ] CoordSys Auto
[ PersistentCache { ON | OFF } [ Type { NATIVE | NATIVEX } ] ]
[ Symbol... ] [ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
Type GPKG
[ Table GeoPackageTableName ]
[ Symbol... ] [ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
[ Font... ]
[ Coordsys... ]
[ into destination_file ]
}
[ ReadOnly ]

```

source_file -- строка, определяющая имя существующей базы данных, электронной таблицы, текстового файла, раstra или файла поверхности. Если требуется зарегистрировать таблицу Access, этот аргумент должен идентифицировать доступную базу данных Access.

char_set – имя набора символов; см. раздел [Предложение CharSet](#). Если он не задан, используется системный набор символов.

delim_char -- знак разделителя между значениями полей таблицы (только для текстовых ASCII-файлов). Если в файле используется табулятор (задаваемый клавишей TAB) в качестве разделителя, укажите 9. Если используется запятая, укажите 44.

Range_name – строка, указывающая именованный диапазон (для WKS-файлов) или диапазон ячеек (для XLS-файлов). Например, Excel-диапазон может быть задан как "Sheet1!R1C1:R9C6" или "Sheet1!A1:F9".

table_name -- строка, определяющая имя таблицы Access.

pwd – пароль на уровне базы данных, определяемый при включении защиты базы данных.

connection_number – целое значение, которое определяет существующее соединение с базой данных ODBC.

connection_string – строка, используемая для подключения к серверу базы данных. Смотрите: [функция Server_ConnectInfo\(\)](#).

toolkit_name – это “ODBC” или “ORAINET”.

Workspace_name – это имя текущей Рабочей области, в которой обрабатывается таблица. Имя зависит от регистра.

ParentWorkspaceName – это имя порождающей Рабочей области.

Format_type – это строка, которая используется FME для того, чтобы определить формат.

feature_type определяет тип объектов (по существу, имя схемы).

string1 [, *string2* ..] – параметры Safe Software FME, которые могут меняться в зависимости от выбранного формата и необходимых преобразований.

GeoPackageTableName – имя таблицы внутри файла пакета геоданных GeoPackage. Файл пакета геоданных GeoPackage может содержать более одной таблицы.

destination_file -- имя будущей таблицы MapInfo (TAB-файл). Строка может включать в себя маршрут, если же он не указан, то файл будет строиться в той же директории, где и исходный файл.

Описание

Перед тем, как использовать в MapInfo файлы “неродных” форматов (например, dBASE файл), Вы должны их зарегистрировать. Оператор **Register Table** понуждает MapInfo Pro проверить, может ли использоваться заданный формат для создания таблицы. Далее MapInfo создает для него файлы-компоненты таблицы (*filename*.TAB, и др.) После того как оператор **Register Table** создаст файл для таблицы, можно обращаться с ним как с таблицей MapInfo.

Оператор **Register Table** не копирует и не изменяет оригинал файла данных. Вместо этого, он проверяет данные, определяет типы данных в колонках и создает отдельную таблицу. Таблица не открывается автоматически. Для открытия таблицы нужно использовать [оператор Open Table](#).

Примечание: Каждый файл данных должен быть зарегистрирован только раз. Так как при регистрации создаются файлы-компоненты, то при следующем сеансе работы MapInfo Pro может просто открывать зарегистрированную ранее таблицу оператором **Open**, а не регистрировать файл заново оператором **Register Table**.

Предложение **Type** задает формат, в котором был создан файл данных. Это включает ключевое слово **Type**, за которым следует одна из следующих символьных констант: **NATIVE**, **NATIVEX**, **DBF**, **ASCII**, **WKS**, **WMS**, **WFS**, **XLS**, **ACCESS**, **ODBC**, **GRID**, **RASTER**, **FME**, **SHAPEFILE** или

GPKG. Для определенных типов таблиц требуется дополнительная информация. Если тип регистрируемой таблицы это **GRID**, то информация о системе координат будет считана из файла поверхности (grid) и будет создан TAB-файл MapInfo. Если регистрируется растровый файл, TAB-файл будет создан в тот момент, когда пользователь выберет "Показать" при открытии растрового изображения в диалоге **Открыть таблицу**.

С помощью предложения **Type** после предложения **Cache** можно задавать тип кэша таблицы ODBC: либо традиционный формат MapInfo TAB (**NATIVE**), либо расширенный формат MapInfo Extended (**NATIVEX**). Если этот параметр не задан, то используется тип **NATIVE**.

Если тип регистрируемого файла **XLS**, набор символов уточняет кодировку, используемую при записи TAB-файла. Для того чтобы разрешить использование Unicode-кодировки для адресов и имён excel-файлов и для того чтобы разрешить Unicode-символы в именах листов или диапазонов, следует задавать набор символов UTF-8.

Если тип регистрируемой таблицы это **GRID**, то информация о системе координат будет считана из файла поверхности (grid) и будет создан TAB-файл MapInfo. Если регистрируется растровый снимок земной поверхности, то создаваемый TAB-файл зависит от информации о геопривязке, которая может храниться в файле снимка или в связанном с ним World-файле. Если тип файла **RASTER**, набор символов уточняет кодировку, используемую при записи TAB-файла. Для того чтобы разрешить использование Unicode-кодировки для адреса и имени растрового файла, следует задавать набор символов UTF-8.

Предложение **CharSet** определяет набор символов. Параметр *char_set* должен быть строковой константой, такой как "MacRoman" или "WindowsLatin1". Если опустить предложение **CharSet**, MapInfo Pro будет использовать системный набор символов. Подробнее см. [Предложение CharSet](#).

Предложение **Delimiter** задает знак разделителя между значениями полей для текстовых файлов. По умолчанию принимается символ табуляции. Предложение **Titles** определяет первую строку данных как названия полей в таблице MapInfo. Предложение **Range** позволяет задавать рабочий лист XLS-файла, который требуется зарегистрировать, или именованный диапазон, заданный в файле WKS-типа. Предложение **Info** задает в виде строки имя файла TAB и расположение на диске. Если предложения в операторе нет, имя файла таблицы будет таким же, как имя файла-источника и располагаться он должен в том же каталоге. В случае, если для диска, на котором расположен файл-источник, разрешено только чтение (например, CD-ROM), то файл таблицы будет сохранен где-то в другом месте, где запись разрешена.

ControlPoints – дополнительное предложение, которое может быть определено, если тип данных поверхность (Grid) или растровое изображение (Raster). Если задано ключевое слово **ControlPoints**, то требуется задать как минимум 3 пары точек координат Map или Raster, которые используются для регистрации изображения. Если заданы контрольные точки **ControlPoints**, то они будут использоваться вместо любых контрольных точек, которые уже ассоциировались с изображением или ассоциировались с регистрационным файлом World.

Interactive является дополнительной опцией для xls-таблиц, растров или файлов поверхности. При задании типов "Grid" или "Raster" нужно указывать все контрольные точки и информацию о проекции. Если их не задать, то TAB-файл будет открыт в режиме только для чтения, как

если бы он был открыт в режиме "Показать" в диалоге команды **Открыть**. **Interactive** не применяется при регистрации шейпфайлов, имеющих расширение SHP.

Примечание: Если задать ключевое слово **Interactive** для XLS-файлов, то это выведет диалоговое окно Настройки полей (Set Field Properties) при импорте файлов Excel.

Предложение **CoordSys** требуется задавать для таблиц WMS и шейпфайлов, иначе при компиляции появится сообщение об ошибке. Для других типов файлов предложение **CoordSys** не обязательно. Если **CoordSys** определено, то им будут переписаны любые системы координат, ассоциированные с изображением. Это полезно когда регистрируется растр, имеющий ассоциированный регистрационный файл World. Дополнительная информация содержится в разделе **Предложение CoordSys**. Предложенный **CoordSys** должен использоваться в исходном SHP-файле. Оператор Register Table не будет менять проекцию исходного SHP-файл на другую координатную систему. Если исходный SHP-файл, например, в универсальной поперечной проекции Меркатора для 18 зоны – UTM Zone 18 (NAD83), то при выполнении оператора Register Table или в диалоге Открыть, открывая SHP-файл необходимо правильно указать ту же самую координатную систему, чтобы корректно открыть SHP-файл.

Используйте токен *auto*, когда в набор данных шейп-файла входит описывающий географические данные PRJ-файл, шейп-файл будет показан в заданной координатной системе. Если PRJ-файл не задан или не найден, либо, если координатная система не преобразована в координатную систему MapInfo оператор не будет выполнен и будет показано сообщение об ошибке.

Symbol – стиль символа для точечного объекта, созданного для шейп-файла, см. раздел **Предложение Symbol**.

Параметр **LineStyle** задаёт стиль оформления линейных объектов.

Параметр **Pen** задаёт стиль линии для линейного объекта, созданного для шейп-файла, см. **раздел Pen**.

Параметр **Regionstyle** задает стили оформления границ и заливок площадных объектов, создаваемых для шейп-файла, смотрите разделы: **Pen** и **Brush**.

Если параметр **Autokey** задан в состоянии **ON**, то таблица будет зарегистрирована с автоматическим приращением ключа. Если параметр **Autokey** установлен в положение **OFF** или этот параметр пропущен, то таблица будет зарегистрирована без автоматического приращения ключа.

SQLQuery -- SQL-запрос, использованный для создания таблицы MapInfo.

Versioned – показывает, включен ли в открываемой таблице контроль версий (ON) или нет (OFF).

Применяйте параметры **Use** и **Color**, для того чтобы пользоваться информацией о цвете из набора данных.

Параметр **Database** можно использовать, для того чтобы задавать опорный источник данных из базы данных.

С помощью параметра **SingleFile** можно задать условие, что опорный источник данных состоит из единственного файла.

Если для параметра **PersistentCache** задать состояние **ON**, то MAP и ID файлы, открытых шейп-файлов, будут сохранены после закрытия таблицы. Если же для параметра **PersistentCache** задать состояние **OFF**, то MAP и ID файлы будут удалены после закрытия таблицы (они всегда создаются при открытии таблицы).

Можно задавать кэш **PersistentCache** для шейп-файлов (предложение **SHAPEFILE**) либо в виде таблицы традиционного типа MapInfo (**NATIVE**), либо в виде расширенной таблицы (**NATIVEX**). Если этот параметр не задан, то по умолчанию используется **NATIVE**. Таблицы, которые не являются TAB-файлами традиционного типа MapInfo или расширенного типа MapInfo Extended, приводят к появлению сообщения об ошибке. Если набор символов для шейп-файла задан UTF-8, то можно использовать только кэш типа **NATIVEX**. Предложение о типе кэша зависит от порядка использования и должно следовать за предложением **PersistentCache {ON|OFF}**.

Параметр **ReadOnly** означает, что таблицу нельзя править.

Регистрация таблиц Access

Когда регистрируете таблицу Access, MapInfo проверяет колонку с уникальными индексами. Если уже существует такая колонка, MapInfo Pro регистрирует ее в TAB-файле. Эта колонка только для чтения.

Если в таблице Access нет такой колонки, MapInfo Pro модифицирует таблицу Access добавляя к ней колонку с именем MAPINFO_ID с типом данных counter. В этом случае, такая колонка тоже не отображается в MapInfo.

Примечание: Не пытайтесь ни каким способом изменять колонку счетчика. Это автоматически выполняется только MapInfo Pro.

Тип данных баз данных Access транслируется в закрытый тип данных MapInfo. Специальные типы данных Access, такие, как OLE объекты и бинарные поля, в MapInfo Pro не редактируются.

Регистрация ODBC таблиц

Прежде чем установить прямой доступ к удаленной базе данных, рекомендуется предварительно открыть картографическую таблицу для этой базы данных. В случае если Вы предварительно не откроете карту, вся таблица базы данных будет загружаться целиком, что может занять значительное время.

Откройте таблицу карты и измените масштаб таким образом, что будет показан район, относящийся к набору строк таблицы удаленной базы, которые Вы хотели бы увидеть. Например, если требуется извлечь из базы данных строки, относящиеся к Калужской области, установите такой масштаб карты, чтобы в окне отображалась Калужская область. В результате, когда откроете таблицу базы данных, загрузятся только те строки, которые попадают в МОП

карты (минимальный описывающий прямоугольник), в данном случае строки, относящиеся к Калужской области.

Ниже приводится список известных ограничений, которые могут встретиться при работе с прямым доступом:

- Каждая таблица должна иметь единственный уникальный ключ.
- Режим быстрого редактирования (FastEdit) не поддерживается.
- При работе MS ACCESS, в случаях, когда ключевое поле символьное, не будут отображаться строки, значения которых в ключевой колонке меньше чем полная ширина колонки. То есть, если колонка определена как `char(5)`, значение 'aaaa' будет выглядеть как удаленная строка.
- При прямом доступе, флажок **Только чтение** в диалоге сохранения таблицы будет недоступен.
- Изменения, сделанные другим пользователем, не будут видны до тех пор, пока окно не будет прокручено или обновлено. Строки, добавленные другим пользователем, не будут видны до тех пор, пока: 1). поиск по минимальному описывающему прямоугольнику (МОП) не возвратит запись или 2); не применена команда Упаковать. Кроме того, при включенном режиме Хранить в памяти изменения, сделанные другим пользователем могут не показываться в списке до тех пор, пока не будет осуществлено масштабирование карты или сдвиг карты в окне.
- Возникнут проблемы, если приложение-клиент в одном запросе (командой **SQL-Запрос** или с помощью MapBasic) обратится к двум или более таблицам SPATIALWARE, хранящимся в разных системах координат. (Эффективнее осуществлять такие выборки с помощью SQL-запросов на сервере). Эта проблема относится к текущей версии.
- При редактировании таблиц Oracle 7, индексированных по десятичному полю размером более 8 байт, возможно аварийное завершение работы MapInfo Pro.
- Если используется сервер Oracle, то индикатор **Autokey** оповещает о состоянии новой возможности --автоматическом приращении ключа.
- Если оператор **Cache** имеет значение OFF перед соединением, то будет сгенерировано сообщение об ошибке при компиляции.

Регистрация шейп-файлов

Когда регистрируются шейп-файлы, то они могут открываться в MapInfo Pro исключительно в режиме "только для чтения". Поскольку шейп-файл не содержит в себе информации о системе координат, надо определить предложение **CoorSys**. Также возможно определить стили оформления объектов для отображения в MapInfo Pro. Информация о проекции и стилях хранится как метаданные в TAB-файле.

Примечание: Параметр **Interactive** не используется при регистрации SHP-файлов.

Пример: DBF

```
Register Table "c:\mapinfo\data\rpt23.dbf"
Type DBF
Into "Report23"

Open Table "c:\mapinfo\data\Report23"
```

Пример: ODBC

```
Open Table "C:\Data\CANADA\Canada.tab" Interactive
Map From Canada
set map redraw off
Set Map Zoom 1000 Units "mi"
set map redraw on
Register Table "odbc_cancaps"
TYPE ODBC
TABLE "Select * From schemaname.can_caps"
CONNECTION
DSN=dsnname;UID=username;PWD=password;DATABASE=dbname
SERVER=servername
Into
"D:\MI\odbc_cancaps.TAB"
Open Table "D:\MI\odbc_cancaps.TAB" Interactive
Map From odbc_cancaps
```

Пример: RASTER

Регистрация полностью привязанного снимка (обработчик растров может вернуть не менее трех контрольных точек и информацию о проекции).

```
Register Table "GeoRef.tif" type RASTER into "GeoRef.TAB"
```

Регистрация снимка со связанным World-файлом, в котором заданы контрольные точки, но нет сведений о проекции.

```
Register Table "RasterWithWorld.tif" type RASTER coordsys earth projection
9, 62, "m", -96, 23, 29.5, 45.5, 0, 0 into "RasterWithWorld.TAB"
```

Регистрация растрового снимка без контрольных точек и сведений о проекции.

```
Register Table "NoRegistration.BMP" type RASTER controlpoints (1000,2000)
(1,2), (2000,3000) (2, 3), (5000,6000) (5,6) coordsys earth projection
9,
62, "m", -96, 23, 29.5, 45.5, 0, 0 into "NoRegistration.tab"
```

Пример: SHAPEFILE

Регистрация шейп-файла.

```
Register Table "C:\Shapefiles\CNTYLN.SHP" TYPE SHAPEFILE Charset
"WindowsLatin1" CoordSys Earth Projection 1, 33 PersistentCache Off
linestyle Pen (2,26,16711935) Into "C:\Temp\CNTYLN.TAB"
Open Table "C:\Temp\CNTYLN.TAB" Interactive
Map From CNTYLN
```

Пример: ODBC

В следующем примере будет создан и открыт TAB-файл.

```
Register Table "SMALLINTEGER" TYPE ODBC
TABLE "Select * From ""MIPRO"".""SMALLINTEGER""
CONNECTION "SRVR=scout;UID=mipro;PWD=mipro "
toolkit "ORAINET"
Autokey ON
Into
"C:\projects\data\testscripts\english\remote\SmallIntEGER.TAB"
Open Table "C:\Projects\Data\TestScripts\English\remote\SmallIntEGER.TAB"

Interactive
Map From SMALLINTEGER
```

В следующем примере будет создан и открыт TAB-файл. В примере используется рабочий набор.

```
Register Table "Gwmusa" TYPE ODBC
TABLE "Select * From ""MIUSER"".""GWMUSA""
CONNECTION "SRVR=troyny;UID=miuser;PWD=miuser"
toolkit "ORAINET"
Versioned On
Workspace "MIUSER"
ParentWorkspace "LIVE"
Into "C:\projects\data\testscripts\english\remote\Gwmusa.tab"
Open Table "C:\Projects\Data\TestScripts\English\remote\Gwmusa.TAB"
Interactive Map From Gwmusa
```

Пример: FME (внешние данные)

```
Register Table "D:\MUT\DWG\Data\afrika_miller.DWG" Type FME
CoordSys Earth Projection 11, 104, "m", 0 Format "ACAD"
Schema "afrika_miller" Use Color SingleFile Symbol (35,0,16)
Linestyle Pen (1,2,0) RegionStyle Pen (1,2,0)
Brush (2,16777215,16777215) Font ("Arial",0,9,0)
Settings "RUNTIME_MACROS",
"METAFILE,acad,_EXPAND_BLOCKS,yes,ACAD_IN_USE_BLOCK_HEAD
```

```

ER_LAYER,yes,ACAD_IN_RESOLVE_ENTITY_COLOR,yes,_EXPAND_VISIBLE,yes,_BULGES
_AS_ARCS,no,_STORE_BULGE_INFO,no,_READ_PAPER_SPACE,no,ACAD_IN_READ_GROUPS
,no,_IGNORE_UCS,no,_ACADPreserveComplexHatches,no,_MERGE_SCHEMAS,YES",
"META_MACROS",
"Source_EXPAND_BLOCKS,yes,SourceACAD_IN_USE_BLOCK_HEADER_LA
YER,yes,SourceACAD_IN_RESOLVE_ENTITY_COLOR,yes,Source_EXPAND_VISIBLE,yes,
Source_BULGES_AS_ARCS,no,Source_STORE_BULGE_INFO,no,Source_READ_PAPER_SPA
CE,no,SourceACAD_IN_READ_GROUPS,no,Source_IGNORE_UCS,no,Source_ACADPreser
veComplexHatches,no", "METAFILE", "acad", "COORDSYS", "", "IDLIST", ""
Into "C:\Temp\afrika_miller.tab"
Open table "C:\Temp\afrika_miller.tab"
Map From "afrika_miller"

```

Пример: GPKG (GeoPackage)

```

Register Table "D:\Maps\gpkg\asia.gpkg" Type GPKG
Table "asia_regions"
Symbol (35,0,16) Linestyle Pen (1,2,0)
RegionStyle Pen (1,2,0) Brush (2,16777215,16777215)
Font ("Arial",0,9,0)
Into "D:\Maps\gpkg\asia.tab"

```

Поддержка транзакций WFS-слоев

Синтаксис

```

Register Table source_file
{ Type NATIVE |
Type DBF [ Charset char_set ] |
Type ASCII [ Delimiter delim_char ][ Titles ][ CharSet char_set ] |
Type WKS [ Titles ][ Range range_name ] |
Type WMS Coordsys...
Type WFS [ Charset char_set ] Coordsys... [ Symbol... ]
[ Linestyle Pen(...) ][ Regionstyle Pen(...) Brush(...) ]
[Editable]

```

где:

Editable -- представляет режим Разрешить редактирование.

См. также:

оператор **Open Table**, оператор **Create Table**, оператор **Server Create Workspace**, оператор **Server Link Table**

Оператор Relief Shade

Назначение

Отмывка рельефа открытой таблицы поверхности. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Relief Shade  
Grid tablename  
Horizontal xy_plane_angle  
Vertical incident_angle  
Scale z_scale_factor
```

tablename -- псевдоним поверхности, для которой рассчитывается отмывка.

xy_plane_angle -- дирекционный угол, в градусах, от источника света в горизонтальной или XY-плоскости. Значение *xy_plane_angle* равное нулю, представляет источник света, светящий с Востока. Положительные значения угла отсчитываются против часовой стрелки, так, например источник с Северо-Запада имеет значение *xy_plane_angle* равное 135.

incident_angle -- угол от источника света над горизонтом или плоскостью XY. Значение *Incident_angle* равное нулю представляет источник, лежащий на линии горизонта. Значение *incident_angle* равное 90 помещает источник света в зенит.

z_scale_factor -- масштабный фактор, применяемый к параметру z для каждой ячейки поверхности. Увеличение *z_scale_factor* улучшает эффект отмывки. Это может быть использовано для выявления мелких деталей поверхности.

Пример

```
Relief Shade  
Grid Lumens  
Horizontal 135  
Vertical 45  
Scale 30
```

Оператор Reload Symbols

Назначение

Открывает и перезагружает файл символов MapInfo и тем самым меняет набор символов, используемый MapInfo и показываемый в диалоге **Стиль символа**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис (вариант 1 – версия для символов MapInfo 3.0):

```
Reload Symbols
```

Синтаксис 2 – символы из растровых файлов

```
Reload Custom Symbols From directory
```

directory -- строка с маршрутом к каталогу.

Описание

Оператор используется в утилите SYMBOL.MBX, которая дает пользователю возможность создавать свои символы.

Примечание: Символы MapInfo 3.0 были введены в MapInfo Pro для Windows 3.0 и поддерживаются во всех последующих версиях MapInfo Pro.

См. также:

[Оператор Alter Object](#)

Процедура RemoteMapGenHandler

Назначение

Имя специальной процедуры, вызываемой тогда, когда клиент OLE Automation вызывает метод MapGenHandler Automation.

Синтаксис

```

Declare Sub RemoteMapGenHandler

Sub RemoteMapGenHandler
    statement_list
End Sub

```

statement_list -- список операторов MapBasic, выполняемых когда клиент OLE Automation вызывает метод MapGenHandler.

Описание

RemoteMapGenHandler -- это имя процедуры MapBasic специального назначения, которая вызывается через OLE Automation. Если вы используете OLE Automation для управления MapInfo Pro, и вызываете метод MapGenHandler, MapInfo Pro вызывает процедуру **RemoteMapGenHandler** любого работающего приложения MapBasic. Метод MapGenHandler – это компонента механизма MapGen Automation, представленного в MapInfo Pro 4.1.

Метод MapGenHandler Automation использует один аргумент: строку. Внутри процедуры RemoteMapGenHandler можно восстановить строковый аргумент с помощью вызова следующей функции CommandInfo (CMD_INFO_MSG) и присвоить результат строковой переменной.

Пример

Примером использования **RemoteMapGenHandler**, является программа MAPSRVR.MB.

Процедура RemoteMsgHandler

Назначение

Процедура, автоматически выполняющаяся при получении командного сообщения из другой программы.

Синтаксис

```

Declare Sub RemoteMsgHandler

Sub RemoteMsgHandler
    statement_list
End Sub

```

statement_list -- группа операторов, которую нужно выполнить.

Описание

RemoteMsgHandler -- зарезервированное имя процедуры MapBasic, предназначенной для обслуживания связи между прикладными программами и другими программами. Если выполняется MapBasic-программа, в которой имеется процедура **RemoteMsgHandler**, а другая программа (например, электронная таблица или база данных) передает исполнимую команду "execute", MapInfo Pro автоматически вызывает процедуру **RemoteMsgHandler**. После этого MapBasic-процедурой вызывается **функция CommandInfo()**, с помощью которой строка с исполняемой командой воспринимается программой.

Для того чтобы остановить выполнение процедуры **RemoteMsgHandler**, если она больше не нужна, может быть использован **оператор End Program**. Однако **оператор End Program** должен использоваться очень осторожно, чтобы случайно не остановить выполнение процедуры **RemoteMsgHandler**.

DDE-связь между программами в среде Windows

Если программа Windows поддерживает DDE-связь (Dynamic Data Exchange – динамический обмен данными), то такая программа может инициировать обмен данными с MapInfo Pro. В таком обмене данными внешняя программа выступает в роли клиента (активная сторона обмена), а заданная MapBasic-программа – сервера (пассивная сторона обмена).

Всякий раз, когда DDE-клиент передает исполняемую команду, MapInfo Pro вызывает серверную процедуру **RemoteMsgHandler**. Внутри процедуры **RemoteMsgHandler** можно использовать вызов функции,

```
CommandInfo(CMD_INFO_MSG)
```

где:

```
CMD_INFO_MSG (1000)
```

для прочтения сообщения. В динамическом обмене данных имя программы, выступающей в роли сервера (например, "C:\MAPBASIC\DISPATCH.MBX") должно использоваться как объект (topic) обмена с помощью процедуры **RemoteMsgHandler**.

См. также:

оператор DDEExecute, **функция DDEInitiate()**, **процедура SelChangedHandler**, **процедура ToolHandler**, **процедура WinChangedHandler**, **процедура WinClosedHandler**

Функция RemoteQueryHandler()

Назначение

Процедура, автоматически выполняющаяся при получении командного сообщения из другой программы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Declare Function RemoteQueryHandler() As String

Function RemoteQueryHandler() As String
    statement_list
End Function
```

statement_list -- группа операторов, которую нужно выполнить при получении “считывающего” (peek) запроса.

Описание

Функция **RemoteQueryHandler()** работает с механизмом DDE (Dynamic Data Exchange). Более подробная информации о DDE содержится в *Руководстве пользователя MapBasic*. Внешняя программа может начать сеанс DDE-связи с Вашей MapBasic-программой. При инициации связи внешняя программа использует значение “MapInfo” как имя DDE-программы и имя Вашей MapBasic-программы как DDE-тему (topic). Как только связь установится, клиент (внешняя программа) может посылать считывающие запросы к MapBasic-программе (серверу).

Для обработки таких запросов в Вашу MapBasic-программу нужно включать обработчик **RemoteQueryHandler()**. Как только от клиента поступает считывающий запрос, MapInfo Pro автоматически вызывает функцию **RemoteQueryHandler()**. Следующие считывающие запросы ожидают своей очереди, т.е. выполняются после того, как **RemoteQueryHandler()** вернет значение.

Примечание: DDE-клиент может считывать значения глобальных переменных MapBasic-программы без помощи **RemoteQueryHandler()**. Если клиент выдаст считывающий запрос к глобальной переменной MapBasic, MapInfo Pro автоматически возвратит требуемое значение вместо того, чтобы обратиться к **RemoteQueryHandler()**. Другими словами, если данные, которые требуется передать, хранятся в глобальных переменных, то в **RemoteQueryHandler()** не нужна.

Пример

В следующем примере вызывается **функция CommandInfo()**, чтобы определить, какой объект требует DDE-клиент. Если объект называется "code1", то действия продолжаются по одному сценарию, если нет, то по другому.

```
Function RemoteQueryHandler() As String
    Dim s_item_name As String

    s_item_name = CommandInfo(CMD_INFO_MSG)

    If s_item_name = "code1" Then
        RemoteQueryHandler = custom_function_1()
    Else
        RemoteQueryHandler = custom_function_2()
    End If

End Function
```

См. также:

функция DDEInitiate(), **процедура RemoteMsgHandler**

Оператор Remove Cartographic Frame

Назначение

Оператор позволяет удалять разделы из существующей картографической легенды, которую создает **оператор Create Cartographic Legend**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Remove Cartographic Frame
[ Window legend_window_id ]
Id frame_id, frame_id, frame_id, ...
```

legend_window_id – целочисленный идентификатор окна, который возвращают **функция FrontWindow()** и **функция WindowID()**.

frame_id – ID-номер раздела легенды. Нельзя использовать имя слоя. Например, три раздела легенды могут иметь идентификаторы ID 1, 2 и 3.

См. также:

оператор Add Cartographic Frame, **оператор Alter Cartographic Frame**, **оператор Create Cartographic Legend**, **оператор Set Cartographic Legend**

Оператор Remove Designer Frame

Назначение

Позволяет удалять либо разделы легенды существующего окна **Легенда**, созданного оператором **Create Designer Legend**, либо рамки в окне **Отчёт**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Remove Designer Frame
[ Window legend_or_layout_window_id ]
Id frame_id, frame_id, frame_id, ...
```

Legend_or_layout_window_id – целочисленный идентификатор окна конструктора легенды или окна **Отчёт**, который можно получить с помощью [функции FrontWindow\(\)](#) и [функции WindowID\(\)](#).

frame_id – ID-номер раздела легенды. Нельзя использовать имя слоя. Например, три раздела легенды могут иметь идентификаторы ID 1, 2 и 3.

Если опустить параметр **Window**, то оператор будет обрабатывать либо самое верхнее окно, либо самое верхнее окно **Легенда**, либо самое верхнее окно **Отчёт**, в зависимости от ситуации.

См. также:

[оператор Add Designer Frame](#), [оператор Alter Designer Frame](#), [оператор Create Designer Legend](#), [оператор Set Designer Legend](#)

Оператор Remove Designer Page

Этот оператор удаляет страницу из окна **Отчёт**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Remove Designer Page
[ Window layout_win ] ID page_number [ , page_number2 . . . ]
```

window_id – идентификатор окна **Отчёт**, целое число;

Page_number – номер страницы от первой (1) до количества страниц окна **Отчёт**. Можно задавать больше одного номера страниц, разделяя номера запятыми.

Описание

Оператор **Remove Designer Page** удаляет одну или несколько страниц окна **Отчёт**. Если в отчёте только единственная страница, то нельзя удалять эту страницу с помощью этого оператора; единственный способ удалить последнюю страницу отчёта это закрыть окно **Отчёт** (смотрите описание оператора **Close Window**).

Сообщения об ошибке появляются в следующих условиях:

- Сообщение об ошибке 8221 появляется, если значение pageNum меньше 1 или больше чем количество страниц.
- Сообщение об ошибке 8220 появляется, если оператор содержит один и тот же номер страницы несколько раз.

Пример

```
Remove Designer Page Window FrontWindow( ) ID 3, 4
```

См. также:

[оператор Close Window](#), [оператор Create Designer Legend](#), [оператор Add Designer Frame](#), [оператор Add Designer Page](#), [оператор Add Designer Text](#), [оператор Add Image Frame](#)

Оператор Remove Designer Text

Оператор **Remove Designer Text** удаляет текстовые разделы из окна **Легенда**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Remove Designer Text
[ Window legend_window_id ]
[ ID textframe_id [, textframe_id] . . .]
```

legend_window_id – это целочисленный идентификатор окна, который возвращают [функция FrontWindow\(\)](#) и [функция WindowID\(\)](#).

textframe_id – уникальный номер текстового раздела (т.е. не раздела легенды) в окне **Легенда**. Если номеров несколько, они отделяются запятой.

Описание

ID задает уникальный номер (номера) удаляемых текстовых разделов.

Пример

```
Remove Designer Text Window frontwindow()
    ID 1, 2, 4
```

См. также:

[оператор Create Designer Legend](#), [оператор Add Designer Text](#), [оператор Alter Designer Text](#)

Оператор Remove Map

Назначение

Удаляет один или несколько слоев карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Remove Map [ Window window_id ]
    Layer map_layer [ , map_layer ... ] |
    GroupLayer group_id [ , group_id ... ]
    [ Interactive ]
```

window_id – идентификатор окна карты, целое число, для того чтобы его получить, используются [функция FrontWindow\(\)](#) или [функция WindowID\(\)](#).

map_layer -- задает слой карты, который требуется удалить; смотрите примеры ниже.

Описание

Оператор **Remove Map** удаляет один или несколько слоев или группы слоев из окна карты. Если оператор не содержит идентификатора *window_id* оператор будет работать с тем окном карты и лежит выше остальных.

Число *group_id* может быть либо положительным числом большим нуля, номером определенной группы слоев карты, либо именем группы слоев. Если это имя группы, то будет удалена первая группа с таким именем сверху списка слоев. Так как число *map_layer* является уникальным, что оно может обозначать слой внутри группы. Чтобы удалить группу вместе со всеми вложенными группами, используйте предложение **GroupLayer**. Параметр *map_layer* является

либо положительным числом большим нуля, либо строкой, содержащей имя таблицы, либо словом **Animate**, как показано в следующей таблице.

Примеры	Описание
<code>Remove Map Layer 1</code>	Если задать в <i>map_layer</i> "1" (единицу), то будет удален самый верхний некосметический слой. Если в параметре задается "1, 2", то будут удалены два верхних некосметических слоя. Пример: <pre>Remove Map GroupLayer 1</pre> Удаляет первую группа слоев из списка.
<code>Remove Map Layer "Zones"</code>	Оператор удаляет слой Zones (то есть тот, который в списке слоев обозначен именем "Zones").
<code>Remove Map Layer "Zones (1) "</code>	Оператор удаляет первый тематический слой, построенный на основе слоя Zones.
<code>Remove Map Layer Animate</code>	Оператор удаляет слой анимации. Информацию об анимационных слоях см. в разделе: Оператор Add Map .

Если используется ключевое слово **Interactive** и если удаление слоя ведет к потере подписей или тематических объектов, то MapInfo Pro выведет на экран диалог, который позволит сохранить пользователю рабочий Набор перед удалением слоя. Если ключевое слово в операторе **Interactive** опущено, то пользователь не предупреждается о потере.

Удаляя слой, который отображает данные некоторой открытой таблицы оператор **Remove Map** не закрывает ее. Если оператор **Remove Map** удаляет последний не косметический слой окна карты, MapInfo Pro закроет это окно автоматически.

См. также:

[оператор Create Map](#), [оператор Map](#), [оператор Set Map](#)

Оператор Rename File

Назначение

Изменяет имя открытого файла. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Rename File old_filespec As new_filespec
```

old_filespec -- строка с именем файла (и, если необходимо, маршрут); файл не должен быть открыт.

new_filespec -- строка с новым именем файла (и, если необходимо, маршрутом).

Описание

Оператор **Rename File** переименовывает файл.

Параметр *new_filespec* задает новое имя файла. Если маршрут *new_filespec* отличается от оригинального маршрута, то MapInfo Pro перенесет файл в заданный каталог.

Пример

```
Rename File "startup.wor" As "startup.bak"
```

См. также:

[оператор Rename File](#), [оператор Save File](#)

Оператор Rename Table

Назначение

Изменяет имя (и расположение на диске) файла таблицы MapInfo. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Rename Table table As newtablespect
```

table – имя открытой таблицы;

newtablespect -- строка с новым именем файла таблицы (и, если необходимо, маршрутом).

Описание

Оператор **Rename Table** позволяет изменить имя таблицы MapInfo.

Параметр *newtablespect* задает новое имя файла таблицы. Если в параметре *newtablespect* задано имя каталога, то MapBasic, помимо переименования, переместит таблицу в заданный каталог. Оператор **Rename Table** также переименовывает все файлы, которые являются компонентами таблицы. Так как оператор **Rename Table** работает с файлами таблицы непосредственно на диске, то результат его работы отменить нельзя.

Примечание: Операция может повлиять на рабочие наборы. Если до переименования были сохранены файлы рабочих наборов, переименования его составляющих могут повлиять на их загрузку, так как рабочие наборы могут искать файлы таблиц там, где их нет.

Не следует с помощью оператора **Rename Table** назначать имена для строки заголовка таблицы. Этого можно добиться, применяя [оператор Open Table](#) с дополнительным параметром **As**.

Оператор **Rename Table** нельзя применить к таблице, которая имеет в своем определении слово "View". Например, нельзя переименовывать таблицы улиц (стандарта StreetInfo), такие, как SF_STRTS, состоящие фактически из двух связанных таблиц, (в нашем случае SF_STRT1 и SF_STRT2). Также нельзя переименовывать временные таблицы (например, QUERY1). Нельзя также переименовывать таблицы с несохраненными изменениями. Перед выполнением оператора **Rename Table** в этом случае надо сохранить или отменить удаление (операторами Commit и Rollback).

Пример

Переименуем таблицу CASANFRA в SF_HIWAY, используя оператор **Rename Table**.

```
Open Table "C:\DATA\CASANFRA.TAB"
Rename Table CASANFRA As "SF_HIWAY.TAB"
```

В этом примере таблица переименовывается из CASANFRA в SF_HIWAY и перемещается на другой каталог.

```
Open Table "C:\DATA\CASANFRA.TAB"
Rename Table CASANFRA As "c:\MAPINFO\SF_HIWAY"
```

См. также:

[оператор Close Table](#), [оператор Drop Table](#)

Оператор Reproject

Назначение

Позволяет определить, какая колонка таблицы появится следующей при открытии окна списка. Этот оператор устарел и не рекомендован к использованию.

Оператор Resume

Назначение

Выход из процедуры-обработчика ошибок, назначенного оператором **OnError**.

Синтаксис

```
Resume { 0 | Next | label }
```

label -- строка с меткой в тексте некоторой подпрограммы или функции.

Ограничения!

Нельзя использовать оператор **Resume** в окне **MapBasic**.

Описание

Оператор **Resume** позволяет выйти из процедуры обработки ошибок.

Группа операторов после метки, которую задает [оператор OnError](#), выполняется, если при работе программы произошла ошибка. Обычно, эта процедура-обработчик ошибок может содержать один или несколько операторов **Resume**. Оператор **Resume** выводит программу MapBasic из процедуры обработки ошибок.

Несколько форм оператора **Resume** позволяют переходить к определенному оператору MapBasic после выхода из процедуры обработки ошибок:

Оператор **Resume 0** возвращает управление программой туда, откуда был вызван обработчик ошибок с повторением оператора, в котором произошла ошибка.

Оператор **Resume Next** осуществляет перевод управления программой к оператору, следующему за оператором, где произошла ошибка.

Оператор **Resume label** передает управление программой строке с меткой, заданной параметром label. Метка label должна находиться в теле какой-нибудь процедуры.

Пример

```
...
OnError GoTo no_states
Open Table "states"
Map From states
after_mapfrom:
...
End Program
no_states:
Note "Could not open States; no Map used."
Resume after_mapfrom
```

См. также:

[функция Err\(\)](#), [оператор Error](#), [функция Error\\$\(\)](#), [оператор OnError](#)

Функция RGB()

Назначение

Возвращает значение цвета в системе RGB, вычисляя из установок красного, зеленого и синего цветов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
RGB( red, green, blue )
```

red -- числовое выражение в диапазоне от 0 до 255, определяющее концентрацию красного.

green -- числовое выражение в диапазоне от 0 до 255, определяющее концентрацию зеленого.

blue -- числовое выражение в диапазоне от 0 до 255, определяющее концентрацию синего.

Возвращаемая величина

Целое – Integer

Описание

MapBasic использует цвет в операторах как часть установки стиля линии, штриха или символа (например, см [Оператор Create Point](#)). Каждый цвет в стилях линии и штриха задается целым числом, понимаемым как RGB-величина. Функция **RGB()** позволяет получить это число.

В таком представлении цвета используется сочетание трех компонент – красного, зеленого и синего цветов. Соответственно, в функции **RGB()** присутствуют три параметра – red, green и blue. Каждая цветовая компонента должна иметь целочисленное значение в диапазоне от 0 до 255, включительно.

RGB-величина вычисляется по формуле:

$$(red * 65536) + (green * 256) + blue$$

Замечание: Файл стандартных определений MAPBASIC.DEF с помощью оператора **Define** назначает имена для основных 8 цветов: BLACK -- черный; WHITE -- белый; RED -- красный; GREEN -- зеленый; BLUE -- синий; CYAN -- голубой; MAGENTA -- розовый; YELLOW -- желтый. Если требуется задать красный цвет, то можно использовать имя RED вместо вызова функции **RGB(255,0,0)**.

Пример

В следующем примере RGB-значение, хранящееся в переменной color, будет представлять чистый, насыщенный красный.

```
Dim red,green,blue,color As Integer
red = 255
green = 0
blue = 0
color = RGB(red, green, blue)
```

См. также:

[предложение Brush](#), [предложение Font](#), [предложение Pen](#), [предложение Symbol](#)

Функция Right\$() f

Назначение

Извлекает из правой части строки определенное количество символов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Right$( string_expr, num_expr )
```

string_expr – строковое выражение.

num_expr – числовое выражение

Возвращаемая величина

Строка – String

Описание

Функция **Right\$()** возвращает строку, составленную из *num_expr* правых символов строки *string_expr*.

В результате вычисления выражения *num_expr* должно получаться целое положительное число, не равное нулю. Если величина *num_expr* дробная, MapBasic округляет ее до ближайшего целого. Если параметр *num_expr* равен нулю, то функция **Right\$()** возвращает строку нулевой длины. Если значение параметра *num_expr* больше, чем длина строки *string_expr*, то результатом выполнения функции **Right\$()** будет полная строка *string_expr*.

Пример

```
Dim whole, partial As String
whole = "Afghanistan"
partial = Right$(whole, 4)

' at this point, partial contains the string: "stan"
```

См. также:

[функция Left\\$\(\)](#), [функция Mid\\$\(\)](#)

Функция Rnd()

Назначение

Генератор случайных чисел. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Rnd( list_type )
```

list_type -- целое число, задающее режим случайной последовательности.

Возвращаемая величина

Float, вещественное число в диапазоне от 0 до 1 (не включая).

Описание

Функция **Rnd()** возвращает случайное десятичное число, больше нуля, но меньше единицы.

Обычно используется в форме **Rnd(1)**, возвращающее случайное число. Последовательность случайных чисел всегда получается одна и та же, кроме случая, когда в программе используется **оператор Randomize**. Любой положительный аргумент *list_type* приводит к тем же результатам.

Реже используется другой вариант, когда этот оператор используется в форме: **Rnd(0)**; в этом случае возвращается предыдущее случайное число, которое было порождено функцией **Rnd()**. Такая возможность использования этой функции предусмотрена, в основном для отладки программ.

Еще реже используется вариант, когда в качестве аргумента *list_type* задается отрицательное значение, например, **Rnd(-1)**. При заданном отрицательном значении функция **Rnd()** всегда возвращает одно и то же число, независимо от того использовался или нет **оператор Randomize**. Такая возможность использования этой функции предусмотрена, в основном для отладки программ.

Пример

```
Chknum = 10 * Rnd(1)
```

См. также:

оператор Randomize

Оператор Rollback

Назначение

Отменяет все изменения в таблице, которые еще не были сохранены на диске. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Rollback Table tablename
```

tablename – имя открытой таблицы.

Описание

Если в таблице были сделаны изменения, но эти изменения не были сохранены оператор **Rollback** их отменяет. Оператор выполняет такие же действия, как команда **Восстановить** в MapInfo Pro.

Примечание: Когда оператор Rollback применяется к таблице запроса, MapInfo Pro автоматически отменяет все изменения в постоянной таблице, на основе которой была сделана временная таблица (кроме тех случаев, когда в процессе запроса использовалось объединение или когда запрос возвратил обобщенные значения, т.е. использовался **оператор Select** с параметром **Group By**).

Таблица запросов (например, "QUERY1") обычно отражает текущее содержание какой-то постоянной таблицы (например, "World"). Если применить оператор Rollback к таблице "QUERY1", MapInfo Pro отменит любые изменения, сделанные в исходной таблице "World".

Оператор **Rollback**, использованный по отношению к связанной таблице, отменит все несохраненные изменения такой таблицы и вернет её в состояние, в котором она была до изменений.

Пример

```
If keep_changes Then
  Table towns
Else
  Rollback Table towns
End If
```

См. также:

Оператор Commit Table

Функция Rotate()

Назначение

Позволяет вращать объект (не текстовый) вокруг центроида. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Rotate( object, angle )
```

object -- объект, который может вращаться. Этот объект не может быть текстовым.

angle -- вещественная величина, определяющая угол (в градусах), на который поворачивается объект.

Возвращаемая величина

Повёрнутый объект.

Описание

Функция **Rotate()** поворачивает все типы объектов, кроме текстовых, при этом исходный объект не изменяется.

Для поворота текстовых объектов используйте оператор **Alter Object OBJ_GEO_TEXTANGLE**.

Если поворачивается дуга, эллипс, прямоугольник и скругленный прямоугольник, то результирующий объект будет преобразован в полилинию/полигон для того, чтобы узлы могли быть повернуты.

Пример

```
dim RotateObject as object
Open Table "C:\MapInfo_data\TUT_USA\USA\STATES.TAB"
map from states
select * from States where state = "IN"
RotateObject = rotate(selection.obj, 45)
insert into states (obj) values (RotateObject)
```

См. также:

функция **RotateAtPoint()**

Функция RotateAtPoint()

Назначение

Позволяет вращать объект (не текстовый) вокруг заданной точки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
RotateAtPoint( object, angle, anchor_point_object )
```

object -- объект, который может вращаться. Этот объект не может быть текстовым.

angle -- вещественная величина, определяющая угол (в градусах), на который поворачивается объект.

anchor_point_object -- объект-точка, вокруг которой поворачиваются узлы вращаемого объекта.

Возвращаемая величина

Повёрнутый объект.

Описание

Функция **RotateAtPoint()** поворачивает все типы объектов, кроме текстовых, при этом исходный объект не изменяется.

Для поворота текстовых объектов используйте оператор **Alter Object OBJ_GEO_TEXTANGLE**.

Если поворачивается дуга, эллипс, прямоугольник и скругленный прямоугольник, то результирующий объект будет преобразован в полилинию/полигон для того, чтобы узлы могли быть повернуты.

Пример

```
dim RotateAtPointObject as object
dim obj1 as object
dim obj2 as object
Open Table "C:\MapInfo_data\TUT_USA\USA\STATES.TAB" ]
map from states
select * from States where state = "CA"
obj1 = selection.obj
select * from States where state = "NV"
obj2 = selection.obj
oRotateAtPointObject = RotateAtPoint(obj1 , 65, centroid(obj2))
insert into states (obj) values (RotateAtPointObject )
```

См. также:

Функция Rotate()

Функция Round()

Назначение

Округляет число с заданной точностью. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Round( num_expr, round_to )
```

num_expr – числовое выражение

round_to -- число, определяющее точность округления переменной *num_expr*.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Round()** возвращает округленное значение от числа, полученного в результате вычисления выражения *num_expr*.

Число *round_to* задает точность округления. Функция **Round()** округляет переменную *num_expr* до ближайшего, кратного *round_to*. Например, если параметр *round_to* равен 0.01, то число *num_expr* будет округлено до сотых. Если *round_to* равен 5, то MapInfo Pro вернет значение, ближайшее к *num_expr* и кратное пяти.

Пример

```
Dim x, y As Float
x = 12345.6789

y = Round(x, 100)
' y now has the value 12300

y = Round(x, 1)
' y now has the value 12346

y = Round(x, 0.01)
' y now has the value 12345.68
```

См. также:

[функция Fix\(\)](#), [функция Format\\$\(\)](#), [функция Int\(\)](#)

Функция RTrim\$()

Назначение

Удаляет пробелы в конце строки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
RTrim$( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **RTrim\$ ()** возвращает строку, заданную параметром *string_expr*, но без пробелов в конце строки.

Пример

```
Dim s_name As String  
s_name = RTrim$("Mary Smith ")
```

s_name теперь содержит строку "Mary Smith" (без замыкающего пробела).

См. также:

[Функция LTrim\\$\(\)](#)

оператор Run Application

Назначение

Загружает прикладную программу или рабочий Набор в MapInfo, окна которого будут добавлены к уже открытым. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Run Application [ NoMRU ] file [ Mode default | Current ]
```

file -- имя файла прикладной программы или рабочего набора.

Mode default – открывает новый проект, при этом не предупреждая о необходимости закрыть или сохранить уже открытые рабочие наборы.

Mode default – открывает новый проект, предупреждая о необходимости закрыть или сохранить уже открытые рабочие наборы.

Если в операторе задано предложение **NoMRU**, то название программы или рабочего набора не будет добавляться в списке последних использованных файлов.

Описание

Оператор **Run Application** запускает программу MapBasic или загружает рабочий набор. Выполняя оператор **Run Application**, одна прикладная программа, написанная на MapBasic, может запустить на выполнение другую программу, определенную параметром *file*. Оператор **Run Application** не может запускать файлы с расширением MB, то есть файлы, содержащие тексты программ. Для остановки программы, запущенной оператором **Run Application** другой используется [оператор Terminate Application](#).

Пример

Следующий оператор загружает программу REPORT.MBX:

```
Run Application "C:\MAPBASIC\APP\REPORT.MBX"
```

Следующий оператор загружает рабочий набор PARCELS.WOR:

```
Run Application "Parcels.wor"
```

См. также:

[оператор Run Command](#), [оператор Run Menu Command](#), [оператор Run Program](#), [оператор Terminate Application](#)

Оператор Run Command

Назначение

Выполняет оператор MapBasic, заданный строкой. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Run Command command
```

command -- строка символов, представляющая оператор MapBasic.

Описание

Оператор **Run Command** интерпретирует строку *command* в оператор MapBasic и выполняет его.

Оператор **Run Command** имеет некоторые ограничения, обусловленные тем, что строка *command* интерпретируется сразу, а не выполняется из скомпилированной программы. Нельзя использовать оператор **Run Command**, для того чтобы применить **оператор Dialog**. К тому же, в строке *command* не могут быть использованы имена переменных; то есть, переменные не могут появляться в кавычках. Например, следующая группа операторов не будет работать, так как указание на переменные *x* и *y* попадает в строку, ограниченную кавычками:

```
' this example WON'T work
Dim cmd_string As String
Dim x, y As Float

cmd_string = " x = Abs(y) "
Run Command cmd_string
```

Однако, в строке *command* можно использовать имена переменных.

В следующем примере, строка *command* построена из выражений, в которых используются строковые переменные.

```
'this example WILL work
Dim cmd_string As String
Dim map_it, browse_it As Logical

Open Table "world"
If map_it Then
  cmd_string = "Map From "
  Run Command cmd_string + "world"
```

```

End If
If browse_it Then
  cmd_string = "Browse * From "
  Run Command cmd_string + "world"
End If

```

Пример

Оператор **Run Command** представляет гибкий путь выполнения операторов, аргументы которых представляются списками переменной длины. Например, **оператор Map From** может включать в себя имя одной таблицы или список имен таблиц через запятую. Программа может интерпретировать оператор во время выполнения, чтобы определить (используя данные, введенные пользователем), сколько таблиц включить в карту. Оператор Run Command позволяет на ходу сконструировать и выполнить **оператор Map From**. Один способ выполнения этого заключается в следующем: создать строку текста при выполнении программы и передать команду оператору **Run Command**.

```

Dim cmd_text As String
Dim cities_wanted, counties_wanted As Logical

Open Table "states"
Open Table "cities"
Open Table "counties"

cmd_text = "states" ' always include STATES layer

If counties_wanted Then
  cmd_text = "counties, " + cmd_text
End If

If cities_wanted Then
  cmd_text = "cities, " + cmd_text
End If

Run Command "Map From " + cmd_text

```

Следующий пример показывает как можно дублировать окно карты. **Функция WindowInfo ()** возвращает строку с операторами MapBasic, а оператор **Run Command** выполняет эту строку.

```

Dim i_map_id As Integer

' First, get the ID of an existing Map window
' (assuming the Map window is the active window):
i_map_id = FrontWindow()

' Now clone the active map window:
Run Command WindowInfo(i_map_id, WIN_INFO_CLONEWINDOW)

```

См. также:

оператор **Run Application**, оператор **Run Command**, оператор **Run Menu Command**

Оператор Run Menu Command

Назначение

Позволяет вызвать стандартную команду MapInfo Pro так, как если бы вы указали на ее имя в списке меню. Может быть использован для выбора кнопки панели инструментов. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Run Menu Command { command_code | ID command_ID }
```

command_code — целочисленный код из файла `MENU.DEF`, соответствующий коду команды из действующей системы меню MapInfo или кнопке.

command_ID — число, представляющее созданный элемент меню или кнопку.

Описание

Чтобы выполнить стандартную команду меню MapInfo Pro, ее код должен быть задан в параметре *command_code*. Значение этого параметра должно соответствовать одному из кодов меню, перечисленных в файле `MENU.DEF`. Например, следующий MapBasic-оператор выполняет команду меню MapInfo Pro **Файл > Новая**:

```
Run Menu Command M_FILE_NEW
```

Чтобы "нажать" одну из стандартных кнопок MapInfo, нужно указать её код из `MENU.DEF`). Например, для "нажатия" кнопки "Поиск-в-Круге", применяется следующий оператор:

```
Run Menu Command M_TOOLS_SEARCH_RADIUS
```

Для выбора созданных пользователем кнопки или команды в меню (то есть кнопка или элемент меню, созданные и обрабатываемые программой MapBasic), используется предложение **ID**.

Например, если программа создаёт кнопку инструмента при помощи следующего оператора...

```
Alter ButtonPad ID 1 Add
  ToolButton
    Calling sub_procedure_name
    ID 23
    Icon MI_ICON_CROSSHAIR
```


... то кнопка будет иметь идентификатор ID равный 23. Следующий оператор выбирает кнопку.

```
Run Menu Command ID 23
```

Следующий оператор **Run Menu Command** позволяет открыть диалог **Справка > Учебник MapInfo Pro**.

```
Run Menu Command M_HELP_MAPINFO_WWW_TUTORIAL
```

Можно выполнить команду меню **Запрос > Обратить выборку**, используя следующую MapBasic-команду:

```
Run Menu Command M_QUERY_INVERTSELECT.
```

Доступ к настройкам страницы в диалоге **Настройки > Режимы > Принтер** осуществляется с использованием следующего синтаксиса:

```
RUN MENU COMMAND M_EDIT_PREFERENCES_PRINTER
```

Для того чтобы открыть веб-страницу с репозиторием MapBasic-приложений, написанных пользователями MapInfo Pro, используйте параметр M_MBTOOL_GET_MB_UTILITIES (или его значение 1811).

```
RUN MENU COMMAND M_MBTOOL_GET_MB_UTILITIES
```

См. также:

оператор Run Application, оператор Run Program

Приложения интегрированной картографии

Приложения интегрированной картографии не могут показывать диалог Управление слоями как окно. Однако, в интегрированной картографии можно показывать одноименный модальный диалог, используя оператор Run Menu Command с аргументом M_MAP_LAYER_CONTROL_DIALOG (801). Например:

```
Run Menu Command M_MAP_LAYER_CONTROL_DIALOG
```

Диалог Режимы

Диалог **Режимы** MapInfo Pro — это специальный случай. Диалог **Режимы** имеет несколько кнопок, которые вызывают другие под-диалоги. Оператор **Run Menu Command** позволяет открывать эти под-диалог. Например, следующий оператор показывает под-диалог Режимы окна Карты:

```
Run Menu Command M_EDIT_PREFERENCES_MAP
```

Перенос окна наверх

Для того чтобы перенести окна, перечисленные в списке открытых окон в меню, используйте `M_WINDOW_FIRSTWIN`.

```
Run Menu Command M_WINDOW_FIRSTWIN
```

Эту команду также можно использовать, когда требуется поднять рамку наверх макета в окне **Отчёт**.

`M_WINDOW_FIRSTWIN` относится только к первому окну в этом списке. Далее приведён образец списка с окном **Карта** (`WORLDCAP, world Карта:1`), окном **Отчёт** (Конструктор отчёта:1), окном **Список** (`WORLDCAP Список`), и классическим окном **Отчёт** (Отчёт:1), открытыми в MapInfo Pro:

```
WORLDCAP, world Карта:1
Конструктор отчета:1
WORLDCAP Список
WORLDCAP, world Карта:2
Конструктор отчета:2
WORLDCAP, world Карта:3
Конструктор отчета:2
Отчет:1
```

Примечание: О фоне классических окон **Отчёт** смотрите раздел: [Работа с классическим окном отчёта](#).

Следующая команда поднимает окно **WORLDCAP, world Map:1** на самый верх.

```
Run Menu Command M_WINDOW_FIRSTWIN
```

Далее команда переносит следующее окно **Layout Designer:1** наверх:

```
Run Menu Command M_WINDOW_FIRSTWIN + 1
```

Любое окно в этом списке, включая окна встроенные в окно **Отчёт**, поднимается наверх с помощью следующей команды, где $(n - 1)$ — номер окна минус единица:

```
Run Menu Command M_WINDOW_FIRSTWIN + (n - 1)
```

Например, если требуется поднять наверх седьмое окно (`Layout Designer:2`):

```
Run Menu Command M_WINDOW_FIRSTWIN + 6
```

Окно и диалог Управление слоями

В версиях MapInfo Pro 10.0 и выше Управление слоями представлено отдельным окном, а не диалогом. Что касается MapBasic 10.0, программы, написанные на нем, могут показывать как окно, так и диалог Управление слоями, для чего выполняется оператор Run Menu command с разными аргументами:

- Для того чтобы показать окно управления слоями, используйте M_MAP_LAYER_CONTROL:

```
Run Menu Command M_MAP_LAYER_CONTROL
```

- Для показа диалогом с кнопками **ОК** и **Отмена**) используйте аргумент M_MAP_LAYER_CONTROL_DIALOG (или число 801):

```
Run Menu Command M_MAP_LAYER_CONTROL_DIALOG
```

Версии MapInfo Pro до 10.0 умеют показывать только диалог Управление слоями, поэтому в программах MapBasic (MBX) написанные на MapBasic 9.5 или ранних версий подразумевается, что это диалог. Для совместимости с предыдущими версиями, при выполнении в MapInfo Pro версии 10.0 и выше программ, написанных на более ранних версиях языка MapBasic, вместо окна Управление слоями показывается диалог.

Следующие средства помогут проконтролировать, в каком виде программа MapBasic (MBX) представит на экране Управление слоями:

- Если перекомпилировать MBX-программу в среде MapBasic 10.0 (или с использованием файла MENU.DEF из MapBasic 10.0), то новая MBX-программа будет показывать окно Управление слоями, а не диалог. Это полезно сделать для большинства задач, поскольку пользователи MapInfo Pro 10.0 ожидают увидеть Управление слоями в виде окна.
- Если перекомпилировать MBX-программу в среде MapBasic 10.0, но при этом требуется видеть на экране диалог Управление слоями, введите следующий оператор Run Menu Command:

```
Run Menu Command M_MAP_LAYER_CONTROL_DIALOG
```

О диалоге Управление слоями

Диалог **Управление слоями** обладает меньшим функционалом по сравнению с окном Управление слоями. Следующие средства не присутствуют в диалоговой версии Управления слоями:

- Кнопки Вверх и Вниз отключены, если на карте есть сгруппированные слои.
- При щелчке правой кнопкой мыши по слою контекстное меню не появляется. Таким образом, большинство операций с группами слоев не доступны.
- Нельзя передвигать тематические слои.

Оператор Run Program

Назначение

Выполняет другие программы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Run Program program_spec
```

program_spec -- командная строка, задающая имя программы и, если необходимо, список аргументов.

Описание

Если программа, заданная строкой *program_spec*, не является программой Windows, то MapBasic сначала создаст копию командного процессора (DOS shell), а потом будет загружена DOS-программа в его среде. Если параметр *program_spec* имеет значение "COMMAND.COM", MapBasic только откроет окно командного процессора. Если из прикладной программы была запущена программа для DOS, то для возвращения в MapInfo надо ввести команду `Exit`. Даже если программу оператором **Run Program**, Windows продолжает управлять компьютером: то есть может параллельно поддерживать другие Windows-программы и даже саму MapBasic-программу в многозадачном режиме. Возникающие в этой ситуации конфликты прикладная программа обычно не может разрешить. Операторы MapBasic, выполняемые после **Run Program**, не смогут никак обращаться к вызванной программы.

Поэтому в программе должны быть предприняты меры предосторожности, чтобы избежать конфликтов многозадачности при выполнении оператора **Run Program**. Один из способов избежать таких конфликтов: поместить оператор **Run Program** в конце программы или группы операторов, которые могут конфликтовать с вызываемой программой. Например, когда требуется создать элемент меню, который вызывает процедуру-обработчик, в которой в самом конце используется оператор **Run Program**.

Пример

Оператор **Run Program** загружает текстовый редактор Windows, который называется "Notepad", и открывает в нем текстовый файл `THINGS.2DO`.

```
Run Program "notepad.exe things.2do"
```

Следующий оператор вызывает DOS-команду.

```
Run Program "command.com /c dir c:\mapinfo\ > C:\temp\dirlist.txt"
```

См. также:

[оператор Run Application](#), [оператор Run Command](#), [оператор Run Menu Command](#)

Оператор Save File

Назначение

Копирует файл. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Save File old_filespec As new_filespec [ Append ]
```

old_filespec – строка с именем (и, если необходимо, с маршрутом) существующего на диске файла, который не должен быть открыт;

new_filespec – строка с именем (и, если необходимо, с маршрутом), под которым будет создана копия; при этом файл с этим именем не должен быть открыт.

Описание

Оператор **Save File** копирует файл. В это время операции ввода/вывода в этот файл должны быть запрещены.

Если в операторе используется ключевое слово **Append** и параметр *new_filespec* задает имя и маршрут уже существующего файла, то содержимое файла *old_filespec* будет дописано в конец файла *new_filespec*.

Не надо использовать оператор **Save File** для копирования файлов, являющихся компонентами таблицы (таких как *filename.tab*, *filename.map* и т. п.). Для копирования таблиц правильно будет использовать [оператор Commit Table...As](#).

Оператор **Save File** не может копировать файл сам в себя.

Пример

```
Save File "settings.txt" As "settings.bak"
```

См. также:

[оператор Kill](#), [оператор Rename File](#)

Функция SaveLayoutItemCustomFrameFiles()

Назначение

Такая функция сохраняет файлы рамок (*.mlcf и *.xps), которые присутствуют в окне **Отчёт**, по заданному адресу. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SaveLayoutItemCustomFrameFiles( window_id, page_num frame_id, path)
```

window_id — идентификатор окна **Отчёт**, целое число;

Page_num — номер страницы, на которой требуется выполнить запрос идентификатора рамки. Используйте номера в диапазоне от 1 до n, где n это номер страницы отчёта.

frame_id — число, указывающее какую рамку в окне **Отчёт** требуется опросить. Разделы пронумерованы от 1 до n, где n это номер в Легенде.

Path — строка, которая назначает адрес, где будут сохраняться файлы, описывающие рамки.

Возвращаемая величина

Логическое

При удачном выполнении эта функция возвращает TRUE. Иначе, она возвращает FALSE.

Описание

Эта функция позволяет сохранять *.mlcf и *.xps файлы, описывающие рамки, в директории по заданному адресу.

См. также:

[оператор Create Frame](#), [оператор Alter Designer Frame](#), [функция LayoutItemMetadataValue\(\)](#)

Оператор Save MWS

Назначение

Этот оператор позволяет сохранять текущий рабочий набор в виде файла MWS в формате XML для использования в приложениях MapXtreme. К этим MWS-файлам можно организовывать совместный доступ с разных платформ способами, недоступными при работе с рабочими наборами. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Save MWS Window ( window_id [ , window_id ... ] )
Default default_window_id As filespec
```

window_id – идентификатор окна, целое число;

default_window_id – целочисленный идентификатор окна Карты, записываемый в MWS в качестве Карты по умолчанию.

Описание

MapInfo Pro позволяет сохранять карты в рабочих наборах в виде XML-файлов для использования в приложениях MapXtreme. При сохранении рабочих наборов в формате MWS, сохраняются только окна карты или легенды. Все остальные окна отбрасываются, поскольку приложения MapXtreme не могут прочитать эту информацию. Сохраненный в этом формате рабочий набор может быть открыт при помощи программы Менеджер рабочих наборов (Workspace Manager), которая устанавливается вместе с MapXtreme, или при помощи приложения, созданного с использованием MapXtreme. Файл является обычным XML-файлом, так что Вы можете работать с ним, используя любой редактор или средство просмотра XML файлов. MWS-файлы, созданные MapInfo Pro 7.8 или более поздних версий, могут быть проверены по схемам, поставляемым с MapXtreme.

Примечание: Нельзя прочитать MWS-файлы в MapInfo Pro версии 7.8 и или более ранних.

В MapInfo Pro можно настроить модификатор тематической карты таким образом, что тематическая карта будет показана, даже если отключить показ слоя, на основе которого построена эта тематическая карта. В MapXtreme модифицированные тематические карты (плотности точек, диапазонов, отдельных значений) будут показаны, если слой, на основе которого построены такие карты, видим. Для того чтобы гарантировать, что модификаторы видимости, отмеченные в MapInfo Pro, в программах типа "Менеджера рабочих наборов" будут показаны правильно, мы принудительно включаем показ слоя, на основе которого построены тематические карты.

Важно помнить, что многие операторы и функции MapBasic не переводятся в формат MWS. Ниже показано, какие элементы карты могут быть сохранены в MWS-файле, а какие нет. Подробнее о совместимости MapBasic и MySQL смотрите в *Руководстве пользователя MapInfo Pro*.

Что хранится в файлах MWS

В файлах рабочих наборов MWS содержится следующая информация:

- имя таблицы и синоним
- информация о системе координат
- центр карты и масштаб
- список слоев в заданном порядке
- размер карты в пикселях (ширина и высота)
- метод изменения границ рамки карты
- принудительное изменение стилей оформления объектов
- принудительное изменение стилей оформления растровых снимков
- автоматические подписи
- подписи, созданные пользователем
- запросы с соответствующими окнами карт
- тематические картодиаграммы отдельных значений
- тематические картодиаграммы плотности точек
- тематические картодиаграммы значков
- тематические столбчатые картодиаграммы
- тематические картодиаграммы диапазонов
- тематические круговые картодиаграммы
- тематические картодиаграммы поверхностей в виде слоев поверхностей MapXtreme с принудительно заданным стилем оформления
- тематические выражения и выражения для создания подписей по колонке атрибутов.
- описания единообразных оформлений, соответствующие масштабным диапазонам.

Что не хранится в файлах MWS

В файлах рабочих наборов MWS не содержится следующая информация:

- любые окна, кроме карт (списки, схемы, окна районирования, трехмерные карты, карты-призмы)
- единицы расстояния, площади, координаты XY и единицы армейской системы координат
- настройки режима совмещения, автопрокрутки и плавной прокрутки
- настройки принтера
- таблицы, основанные на запросах, которые не представлены отдельными окнами
- любая тема, которая основана на вычисляемых колонках или на выражениях, которые не могут быть переведены из синтаксиса MapBasic в синтаксис MI SQL

- подписи, составленные из выражений, которые не могут быть переведены из синтаксиса MapBasic в синтаксис MI SQL
- запросы с операторами “подзапроса”
- слои, основанные на запросах, содержащих операторы "подзапроса"

Примечание: Оператор “подзапроса” – это любой оператор **Select**, вложенные внутрь другого оператора **Select**.

- настройки экспорта
- геолинки
- группы слоев
- режимы показа узлов объекта, центроидов и направлений линий.

См. также:

[оператор Save Workspace](#)

Оператор Save Window

Назначение

Позволяет сохранять изображение окна в файл; соответствует команде **Сохранить**. Этот оператор используется для сохранения окна карты в растровом и векторно-растровом форматах. MapBasic поддерживает режим полупрозрачности растров. Версии MapBasic10.0 и выше поддерживают полупрозрачность для векторных изображений и форматы EMF+ и EMF+Dual.

Поддерживаются следующие векторные форматы: WMF, EMF, EMF+ и EMF+Dual. WMF и EMF основаны на тех же самых старых технологиях, которые применялись к окнам без визуальных эффектов. Они отображают прозрачные векторные карты, однако на экране изображение выглядит размытым, не как действительно прозрачное изображение. EMF+, который использует технологии расширенной визуализации, выводит карты на экран в очень качественном виде. EMF+Dual - это файл, который содержит как EMF, так и EMF+ изображение.

Многие более ранние приложения не могут прочитать EMF+. Приложение пытается открыть его как EMF (потому что используется расширение EMF) и, в итоге, терпит неудачу. Формат EMF+Dual является компромиссом; ранние версии приложений могут открывать файлы с расширением EMF, а более новые приложения - с расширением EMF+. Например, приложения Office 2000 могут читать EMF, но не EMF+. Office 2007 читает EMF+. Сохраняя окна в формате EMF+Dual, оба приложения могут прочитать изображение.

MapInfo Pro читает все поддерживаемые форматы изображений кроме EMF+. Все изображения открываются в растровом виде (включая WMF и EMF).

Синтаксис

```

Save Window window_id
As filespec
Type filetype
[ Width image_width [ Units paper_units ] ]
[ Height image_height [ Units paper_units ] ]
[ Resolution output_dpi ]
[ Copyright notice [ Font... ] ]

```

window_id – целое число идентификатора окна **Карта**, **Список**, **Отчёт**, **График**, **Статистика**, **Информация** или **Линейка**; определить идентификатор окна поможет [функция FrontWindow\(\)](#) или [функция WindowID\(\)](#).

filespec – строка с именем создаваемого файла.

filetype – строка, определяющая формат файла. Файлы, типы которых отмечены звездочкой (*), не поддерживаются при сохранении окна **Легенда**.

- “BMP” определяет растровый формат Bitmap
- “WMF” определяет формат Windows Metafile *
- “JPEG” определяет формат JPEG
- “JP2” определяет формат JPEG 2000 *
- “PNG” определяет формат Portable Network Graphics
- “TIFF” определяет формат TIFF
- “TIFFCMYK” определяет формат TIFF CMYK
- “TIFFG4” определяет формат TIFFG4
- “TIFFLZW” определяет формат TIFFLZW
- “GEOTIFF” определяет формат TIFF с геопривязкой *
- “GIF” определяет формат GIF
- “PSD” определяет формат Photoshop 3.0 *
- “EMF” определяет формат Windows Enhanced Metafile *
- “EMF+” определяет формат Windows Enhanced Metafile + *
- “EMF+DUAL” определяет формат, при котором информация типов EMF и EMF+ хранится в одном файле *

image_width – число, определяющее нужную ширину снимка.

image_height – число, определяющее нужную высоту снимка.

paper_units – строка, представляющая “бумажные” единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также циперо.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки

- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

output_dpi – число, определяющее нужное разрешение снимка в точках на дюйм (DPI).

notice – строка с сообщением об авторских правах; это сообщение появится в нижней части снимка.

Параметром этого оператора **Font** можно задать стиль оформления текста.

Описание

Оператор **Save Window** сохраняет изображение окна в файл. Действие оператора аналогично действию команды **Сохранить**, с тем исключением, что оператор **Save Window** не выводит диалог на экран. Размер изображения, полученного из окна **Карта**, **Отчёт** или **График**, по умолчанию будет равен размеру самого окна. Размер изображения, полученного из окна легенды, информации, статистики или линейки, по умолчанию будет устанавливаться таким, чтобы показать в окне все данные. Можно задать свои размеры экспортируемых изображений в предложениях **Width** и **Height**. **Resolution** позволяет задать разрешение изображения в dpi при экспорте изображения в растровый формат. **Предложение Font** определяет стиль текста об авторских правах.

Примечание: Окно **Отчёт** нельзя экспортировать в метафайл (EMF/WMF).

Чтобы включить текст авторских прав в нижнюю часть изображения, используйте дополнительное предложение **Copyright**. Смотрите пример ниже. Чтобы стереть текст авторских прав, задайте предложение **Copyright** с пустой строкой ("").

В случае нехватки места на диске при экспорте окна, может быть сгенерирована ошибка под номером 408. Имейте это в виду, если попытаетесь создать слишком большое изображение.

Примеры

В этом примере создается метафайл Windows:

```
Save Window i_mapper_ID As "riskmap.wmf" Type "WMF"
```

В этом примере показано, как оформить сообщение об авторских правах. **Функция Chr\$()** используется, для того чтобы вставить значок копирайта.

```
Save Window i_mapper_ID As "riskmap.bmp"
Type "BMP"
Copyright "Copyright " + Chr$(169) + "2014, Precisely "
```

См. также:

оператор Export

Оператор Save Workspace

Назначение

Создает файл рабочего набора с текущим состоянием рабочего окна MapInfo Pro. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Save Workspace As filespec [ Mode ( Default | Current ) ] [ CharSet  
char_set ]
```

filespec – строка с именем создаваемого файла.

Char_set – строковая константа, например, "WindowsLatin1".

Описание

Оператор **Save Workspace** создает рабочий набор по текущему состоянию программы MapInfo Pro. Действие оператора аналогично действию команды **Сохранить Рабочий набор** на вкладке **ГЛАВНАЯ**, с тем исключением, что оператор **Save Workspace** не выводит диалог на экран.

Когда в предложении **Mode** задан режим **Default**, сохраняется рабочий набор без обновления заголовка текущего рабочего набора в строке названий MapInfo Pro. Когда задан режим **Current**, сохраняется рабочий набор и обновляется заголовок текущего рабочего набора в строке названий MapInfo Pro.

Предложение **CharSet** определяет набор символов. Параметр *char_set* должен быть строковой константой, такой как "MacRoman" или "WindowsLatin1". Если предложение **CharSet** не задано, MapBasic использует системный набор символов, который используется в текущем сеансе. Подробнее см. **Предложение CharSet**. Использование "UTF-8" в качестве набора символов рабочего набора допускает использование символов любого языка.

Для загрузки существующего рабочего набора используйте **оператор Run Application**.

Пример

```
Save Workspace As "market.wor"
```

См. также:

оператор Run Application

Функция SearchInfo()

Назначение

Возвращает информацию о результатах поиска, сделанного функцией SearchPoint() или SearchRect(). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SearchInfo( sequence_number, attribute )
```

sequence_number - целое число от 1 до количества найденных объектов;

attribute – короткое целое число, код результата функции.

Возвращаемая величина

Строка или целое число. Тип зависит от значения параметра *attribute*.

Описание

После вызова функции SearchRect() или SearchPoint(), осуществляющих поиск объектов на карте, можно применить функцию SearchInfo() для обработки результатов поиска.

Параметр *sequence_number* должен принимать значения от 1 и более. Максимальное значение для этого параметра равняется результату функции SearchPoint() или SearchRect().

Параметр *attribute* должен принимать значение одного из кодов, имена для которых заданы в файле стандартных определений MAPBASIC.DEF):

код атрибута	ID	SearchInfo() возвращает:
SEARCH_INFO_TABLE	1	Строка, имя таблицы, содержащей этот объект. Если объект принадлежит Косметическому слою, то строка будет равна "CosmeticN" (где N – число от 1 и более).
SEARCH_INFO_ROW	2	Целое число типа Integer, номер записи. Этот номер может впоследствии использовать оператор Fetch или в предложении Where оператора Select.

Результаты поиска хранятся в памяти до тех пор, пока выполняется программа или до следующего поиска. Результаты поиска не будут удалены из памяти, если пользователь

закроет окно или таблицу, в которых осуществлялся поиск; поэтому не надо затягивать с обработкой результатов поиска. Чтобы форсировать удаление результатов поиска из памяти, проведите поиск, который обязательно ничего не найдет (например, поиск по координатам 0, 0).

MapInfo Pro поддерживает отдельные наборы значений для результатов поиска для каждой действующей MapBasic-программы, а также набор для результатов поиска самой программы MapInfo Pro (для команд, введенных из окна **MapBasic**).

Ошибки

В результате выполнения функции может генерироваться код ошибки ERR_FCN_ARG_RANGE (644), если значение параметра *sequence_number* больше числа найденных объектов.

Пример

Следующая программа создает кнопки для двух инструментов. Если пользователь пользуется точечным инструментом, то вызывается **функция SearchPoint()**; если пользователь рисует рамки, то вызывается **функция SearchRect()**. В каждом случае программа использует функцию **SearchInfo()** для определения того, какие объекты попались пользователю.

```
Include "mapbasic.def"
Include "icons.def"
Declare Sub Main
Declare Sub tool_sub

Sub Main
  Create ButtonPad "Searcher" As
    ToolButton Calling tool_sub ID 1
    Icon MI_ICON_ARROW
    Cursor MI_CURSOR_ARROW
    DrawMode DM_CUSTOM_POINT
    HelpMsg "Click on a map location\nClick a location"
  Separator
    ToolButton Calling tool_sub ID 2
    Icon MI_ICON_SEARCH_RECT
    Cursor MI_CURSOR_FINGER_LEFT
    DrawMode DM_CUSTOM_RECT
    HelpMsg "Drag a rectangle in a map\nDrag a rectangle"
  Width 3

  Print "Searcher program now running."
  Print "Choose a tool from the Searcher toolbar"
  Print "and click on a map."
End Sub

Sub tool_sub
  ' This procedure is called whenever the user uses
  ' one of the custom buttons on the Searcher toolbar.
  Dim x, y, x2, y2 As Float,
    i, i_found, i_row_id, i_win_id As Integer,
```

```

s_table As Alias
i_win_id = FrontWindow()
If WindowInfo(i_win_id, WIN_INFO_TYPE) <> WIN_MAPPER Then
    Note "This tool only works on Map windows."
    Exit Sub
End If
' Determine the starting point where the user clicked.
x = CommandInfo(CMD_INFO_X)
y = CommandInfo(CMD_INFO_Y)
If CommandInfo(CMD_INFO_TOOLBTN) = 1 Then
    ' Then the user is using the point-mode tool.
    ' determine how many objects are at the chosen point.
    i_found = SearchPoint(i_win_id, x, y)
Else
    ' The user is using the rectangle-mode tool.
    ' Determine what objects are within the rectangle.
    x2 = CommandInfo(CMD_INFO_X2)
    y2 = CommandInfo(CMD_INFO_Y2)
    i_found = SearchRect(i_win_id, x, y, x2, y2)
End If

If i_found = 0 Then
    Beep ' No objects found where the user clicked.
Else
    Print Chr$(12)
    If CommandInfo(CMD_INFO_TOOLBTN) = 2 Then
        Print "Rectangle: x1= " + x + ", y1= " + y
        Print "x2= " + x2 + ", y2= " + y2
    Else
        Print "Point: x=" + x + ", y= " + y
    End If

    ' Process the search results.
    For i = 1 to i_found
        ' Get the name of the table containing a "hit".
        s_table = SearchInfo(i, SEARCH_INFO_TABLE)

        ' Get the row ID number of the object that was a hit.
        i_row_id = SearchInfo(i, SEARCH_INFO_ROW)

        If Left$(s_table, 8) = "Cosmetic" Then
            Print "Object in Cosmetic layer"
        Else
            ' Fetch the row of the object the user clicked on.
            Fetch rec i_row_id From s_table
            s_table = s_table + ".coll"
            Print s_table
        End If
    Next
End If
End Sub

```

См. также:

функция **SearchPoint()**, функция **SearchRect()**

Функция SearchPoint()

Назначение

Ищет объекты в заданной точке карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SearchPoint( map_window_id, x, y )
```

map_window_id – идентификатор окна Карты;

x – координата по оси X (например, долгота);

y – координата по оси Y (например, широта).

Возвращаемая величина

Целое число найденных объектов.

Описание

Функция **SearchPoint()** осуществляет поиск объектов в заданной точке карты. Поиск проводится по всем доступным слоям окна карты, включая косметический слой, если для него установлен режим доступности. Полученное значение указывает количество найденных объектов.

Функция не выбирает найденные объекты и не отменяет текущего выбора. Вместо этого функция помещает список выбранных объектов в оперативную память. Для чтения этого списка используется **функция SearchInfo()**, вызываемая после выполнения **функции SearchPoint()**.

Поиск имеет небольшой допуск, аналогичный допуску при действии инструмента Информация. Точки и линейные объекты, расположенные слишком близко к месту указания мышкой, считаются найденными, даже если координаты нажатия кнопки мышки не совпадают с точкой.

Для того, чтобы пользователь мог осуществить выбор точки на карте при помощи мышки, можно использовать **оператор Create ButtonPad** или **оператор Alter ButtonPad** для создания нового инструмента. Используйте код DM_CUSTOM_POINT для рисования “точечным” инструментом. Для определения координат точки, которую пользователь задал новым инструментом, в обработчике инструмента используется **функция CommandInfo()**.

Пример

Смотрите пример в разделе: [Функция SearchInfo\(\)](#).

См. также:

[функция SearchInfo\(\)](#), [функция SearchRect\(\)](#)

Функция SearchRect()

Назначение

Ищет объекты на карте в заданном прямоугольнике (рамке). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SearchRect( map_window_id, x1, y1, x2, y2 )
```

map_window_id – идентификатор окна Карты;

x1, y1 – координаты, задающие один из углов прямоугольника;

x2, y2 – координаты, задающие противоположный по диагонали угол прямоугольника.

Возвращаемая величина

Целое число найденных объектов.

Описание

Функция **SearchRect()** осуществляет поиск объектов в заданном прямоугольнике в окне Карты и возвращает количество найденных объектов. Поиск проводится по всем доступным слоям окна карты, включая косметический слой, если для него установлен режим доступности. Полученное значение указывает количество найденных объектов.

Примечание: Функция не выбирает найденные объекты и не отменяет текущего выбора. Вместо этого функция помещает список выбранных объектов в оперативную память. Для чтения этого списка используется функция **SearchRect()** вызываемая после выполнения [функции SearchInfo\(\)](#).

Механизм поиска работает аналогично механизму инструмента MapInfo Pro Выбор-в-Рамке: если центроид объекта попадает в прямоугольник, то объект включается в список найденных.

Для того, чтобы пользователь мог задавать прямоугольник на карте при помощи мышки, можно использовать **оператор Create ButtonPad** или **оператор Alter Button** для создания нового инструмента. Используйте код DM_CUSTOM_RECT для рисования “рамочным” инструментом. Для определения координат прямоугольника, который пользователь задал новым инструментом, в обработчике инструмента используется **функция CommandInfo()**.

Пример

Смотрите пример в разделе: **Функция SearchInfo()**.

См. также:

функция SearchInfo(), **функция SearchPoint()**

Функция Second()

Назначение

Извлекает секунды из значения типа Time в виде вещественного числа типа Float (0-59.999). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Second( Time )
```

Возвращаемая величина

Число

Пример

Скопируйте текст этого примера в окно **MapBasic** и проверьте работу этой функции.

```
dim X as time
dim fSec as Float
X = CurDateTime()
fSec = Second(X)
Print fSec
```

См. также:

функция Hour(), **функция Minute()**

Функция Seek()

Назначение

Возвращает текущую позицию в файле для операций ввода/вывода.

Синтаксис

```
Seek( filenum )
```

filenum – номер файла, который был присвоен ему при открытии.

Возвращаемая величина

Большое целочисленное значение (LargeInt)

Описание

Функция **Seek()** возвращает текущую позицию в открытом файле для операций ввода/вывода.

Значение параметра *filenum* должно быть номером файла, под которым он был задан в предложении **As**, когда выполняется **оператор Open File**.

Результатом функции **Seek()** будет целое число. Если файл был открыт в режиме прямого доступа, **Seek()** вернет номер записи (запись, которая будет прочитана или записана). Если файл открыт в двоичном режиме функция **Seek()** вернет номер байта, который будет прочитан или записан следующей операцией ввода/вывода.

Ошибки

ERR_FILEMGR_NOTOPEN (366), если файл не был открыт.

См. также:

оператор Get, **оператор Open File**, **оператор Put**, **оператор Seek**

Оператор Seek

Назначение

Назначает в определенном файле текущую позицию для дальнейших операций ввода/вывода.

Синтаксис

```
Seek [ # ] filenum, position
```

Filenum – целочисленное значение LargeInt, обозначающее номер открытого файла.

Position – значение LargeInt, обозначающее нужное положение файла, из которого требуется прочитать.

Описание

Оператор **Seek** устанавливает доступ к содержимому открытого файла, начиная с позиции *position*. Операции ввода/вывода, следующие за оператором **Seek**, будут применяться записи, заданной в **Seek**.

Если файл был открыт в режиме прямого доступа (RANDOM), то значение *position* задает номер записи.

Если файл был открыт в режиме последовательного доступа, то оператор устанавливает доступ к содержимому файла, начиная с байта, номер которого задается параметром *position*; единица соответствует началу файла.

См. также:

оператор **Get**, оператор **Input #**, оператор **Open File**, оператор **Print #**, оператор **Put**, функция **Seek()**, оператор **Write #**

Процедура SelChangedHandler

Назначение

Процедура, которая автоматически вызывается при изменении выбора строк в таблицах.

Синтаксис

```
Declare Sub SelChangedHandler  
  
Sub SelChangedHandler  
    statement_list  
End Sub
```

statement_list – список операторов процедуры-обработчика.

Описание

SelChangedHandler – зарезервированное имя для процедуры MapBasic. Когда пользователь запускает программу, имеющую процедуру **SelChangedHandler**, программа не завершается после выполнения всех операторов процедуры Main и других процедур, вызванных из нее. Программа пребывает в режиме ожидания до тех пор, пока не будет выполнен **оператор End Program**. Пока программа остается в памяти, MapInfo Pro автоматически вызывает **SelChangedHandler** в случае изменения в составе выбранных строк любой таблицы.

Информацию об изменениях выборки слоя **SelChangedHandler** предоставляет применяемая в теле этой процедуры **функция CommandInfo()** со следующими кодами:

код атрибута	ID	Результат CommandInfo (attribute):
CMD_INFO_SELTYPE	1	1, если строка была добавлена в выборку; 2, если строка была исключена из предыдущей выборки; 3, если все строки в таблице выбраны; 4, если все строки таблицы были исключены из предыдущей выборки.
CMD_INFO_ROWID	2	Целое число (Integer) – номер строки, которая была добавлена в выбор или исключена из выбора (работает только, если одна строка была выбрана или исключена из выбора).
CMD_INFO_INTERRUPT	3	Логическое значение: TRUE, если пользователь прервал выбор, нажав клавишу Esc; FALSE – во всех остальных случаях.

Когда процедура закончит свои действия (**оператор End Program**), программа полностью прекращает работу и освобождается занятая ею память. Поэтому используйте **оператор End Program** для удаления процедуры **SelChangedHandler**, когда она больше не нужна. И наоборот, не используйте **оператор End Program** если процедура **SelChangedHandler** может еще понадобиться.

Одновременно в состоянии ожидания могут находиться несколько программ. При изменении выбора автоматически выполняются все процедуры **SelChangedHandler** одна за другой.

Процедура **SelChangedHandler** не может активизировать выполнять такие операции, как управление фокусом или перерисовкой активного окна. Другими словами, в процедуре **SelChangedHandler** не могут быть выполнены такие операторы, как **оператор Note**, **оператор Print** или **оператор Dialog**.

См. также:

функция CommandInfo(), **функция SelectionInfo()**

Оператор Select

Назначение

Выбирает отдельные строки и колонки из одной или более таблиц и составляет из них временные таблицы запросов. Этот оператор часто используется для сортировки и вычисления промежуточных сумм. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Select expression_list
From table_name [ , ... ] [ Where expression_group ]
[ Into results_table [ Noselect ] [ Hide ] ]
[ Group By column_list ]
[ Order By column_list ] [Limit n]
```

expression_list — список выражений через запятую, задающих содержимое колонок временной таблицы "Selection".

expression_group — одно выражение или список выражений, разделенных словами AND или OR.

table_name — имя открытой таблицы.

results_table — имя таблицы, в которую будут помещены результаты выбора.

column_list — имя колонки или список имен, разделенных запятыми.

N — количество записей, возвращаемое оператором выбора Select.

Описание

Оператор выбора — **Select** предоставляет программистам MapBasic возможности диалогового окна **SQL -запрос**.

Оператор **Select** построен по образцу одноимённого оператора в языке запросов SQL (Structured Query Language). В случае, если вы пользуетесь SQL-ориентированным программным обеспечением для работы с базами данными, оператор Select скорее всего вам знаком. Однако, вариант оператора **Select**, применяемый в MapBasic, позволяет использовать уникальные географические возможности MapInfo, которые не имеют многие базы данных, использующие язык SQL.

Имена колонок, задаваемые выражениями типа *имятаблицы.имяколонки* в операторе **Select** могут использовать имена только тех таблиц, которые перечислены в предложении **From**. Например, оператор **Select** выбирает значения в колонке STATES.OBJ, если таблица STATES включена в список предложения **From** в операторе Select.

Операторы **Select** могут выполнять множество различных задач. Один оператор **Select** может перестроить таблицу и отфильтровать (т.е. выбрать по критерию) записи. Другой оператор **Select** может вычислить промежуточные и итоговые суммы для всей таблицы. С помощью оператора **Select** также можно: отсортировать записи в таблице, создать и заполнить новую колонку из других и скомбинировать колонки из разных таблиц и т. д.

Общее назначение оператора **Select** состоит в том, что из одной или нескольких таблиц выбираются строки, которые помещаются во временную таблицу под именем "Selection" или под именем, заданным параметром `results_table`. Эта временная таблица оператора **Select** в дальнейшем интерпретируется MapInfo Pro, как и любая другая.

После того, как оператор **Select** образует выборку, **функция SelectionInfo()** дает информацию о составе выборки.

Синтаксис оператора **Select** допускает несколько предложений, большинство которых не обязательны. Природа и функция оператора **Select** зависит от того, какие предложения входят в состав оператора. Например: если требуется с помощью оператора **Select** отфильтровать записи по критерию, то нужно включить предложение **Where**, если нужно просуммировать значения, то включите в состав оператора **Select** предложение **Group By**; для сортировки оператора **Select** включите предложение **Order By**. Обратите внимание, что каждое из этих предложений не является взаимоисключающим. Оператор **Select** может содержать все возможные предложения сразу.

Предложение Select

Это предложение задает колонки, которые будут включены в результирующую таблицу. Список колонок располагается сразу за первым словом **Select** в операторе и является обязательным. Самое простое значение для параметра *expression_list* – звездочка (*). Это значит, что в результирующую таблицу будут включены все колонки. Например, оператор:

```
Select * From world
```

говорит MapBasic включить все колонки во временную таблицу "Selection". Другим возможным значением параметра *expression_list* является список выражений через запятую. Каждое из выражений представляет колонку для результирующей таблицы. Обычно такое выражение включает в себя имена одной или нескольких колонок таблицы, из которой производится выбор. При составлении списка колонок можно использовать функции и/или операторы MapBasic.

Например, в следующем операторе **Select** список выражений *expression_list* состоит из двух выражений:

```
Select country, Round(population,1000000)
From world
```

Список *expression_list* состоит из двух выражений: первое – имя колонки `country` ("страна"), второе – вызов функции округления (**Round()**) значений колонки `"population"` ("Население") до целых миллионов.

После выполнения этого оператора **Select** первая колонка в результирующей таблице будет содержать величины из колонки "country" с названиями стран в таблице WORLD, а вторая — величины из колонки "population" с численностью населения, округленной до миллионов.

Выражение из списка *expression_list* может также назначить синоним, которым будет именоваться соответствующая колонка, если, например, показывать эту результирующую таблицу в окне Списка. Следующий оператор назначает для второй колонки синоним "Миллионы":

```
Select country, Round(population, 1000000) "Millions" From world
```

Кроме того, можно также использовать ключевое слово **AS**, для того чтобы присвоить имя псевдонима для столбца.

```
Select country, Round(population, 1000000) AS "Millions" From world
```

Примечание: Псевдонимами, синонимами можно пользоваться, не обращая внимания на регистр, но в них не должны присутствовать пробелы.

Таблицы, которые могут иметь присоединенные графические объекты, имеют специальную колонку, которая имеет имя "object" (или "obj"). Если в список колонок включить слово "obj", то в результирующую таблицу будет добавлена строка, содержащая тип *expression_list* графического объекта, присоединенного к данной записи или ничего, если такого объекта нет.

Значение параметра *expression_list* может быть либо списком выражений, либо звездочкой, но не одновременно. Если звездочка используется после ключевого слова **Select**, то звездочка должна присутствовать единственным элементом списка выражений *expression_list*. Следующий оператор НЕ РАБОТАЕТ:

```
Select *, object From world ' this won't work!
```

Можно также выбирать все колонки заданной таблицы без перечисления всех имён колонок таблицы в списке колонок *expression_list*. Для этого используйте вариант: "*tablename.**" в списке колонок *expression_list*, для того чтобы извлечь все колонки в результат запроса.

```
Select usa_caps.Capital, a.* , usa_caps.Capital from states as "a" ,
usa_caps where states.state =
    usa_caps.state
```

Предложение From

Обязательное предложение **From**, в котором должно указываться имя открытой таблицы, из которой производится выбор. Если требуется создать таблицу для совместного доступа, выбранные таблицы должны быть базовыми, а не результатами предыдущего запроса.

Каждая таблица может быть явно вызвана с помощью псевдонима после имени таблицы, используя ключевое слово **AS**. Следующий оператор назначает псевдоним "b" для таблицы *usa_caps* и псевдоним "a" для таблицы *states*.

```
Select a.State, b.Capital from usa_caps AS "b", states AS "a" where
a.state = b.state
```

Предложение Where

Одна из функций предложения **Where**, основная, заключается в задании критерия выбора строк в таблице *table_name*. Здесь могут использоваться любые выражения (смотрите раздел "Выражения" ниже). Но обратите внимание, что два или более выражения разделяются словами **And** или **Or**, а не запятыми. Например, предложение **Where**, состоящее из двух выражений должно выглядеть следующим образом:

```
Where Income > 15000 And Income < 25000
```

Оператор **And** играет роль логического "И", то есть оба условия должны выполняться, а оператор **Or** играет роль логического "ИЛИ", то есть MapBasic выберет запись, если она удовлетворяет любому из условий.

В соответствии с записью специальной колонки, предложение **Where** может контролировать географическую информацию, содержащуюся в каждой строке таблицы. Можно также применять выражение "Not object" для выбора строк, к которым не присоединены графические объекты.

Например, следующий оператор **Where** выбирает строки, лишённые присоединенного графического объекта.

```
Where Not Object
```

Если в операторе **Select** используются две или более таблицы, то предложение **Where** должно присутствовать обязательно, и, более того, в этом предложении должны быть заданы условия объединения. Обычно такой объединяющий оператор выглядит примерно так: **Where таблица1.поле = таблица2.поле**, где две колонки в разных таблицах сопоставляются друг другу. В следующем примере показано, как можно объединить таблицы *STATES* и *CITY_1K*, если в колонках "City_1k.state" и "States.state" содержатся двухбуквенные сокращения названий штатов США:

```
Where States.state = City_1k.state
```

Можно произвести объединение, используя также географический оператор:

```
Where states.obj Contains City_1k.obj
```

Предложение **Where** можно также применять для более изощренного выбора, для чего используются специальные операторы **Any** и **All**. Оператор **Any**, применяемый к некоему множеству значений, позволяет проверить, выполняется ли выражение в предложении **Where**

для какого-либо из этих значений. Оператор **All**, наоборот, требует, чтобы выражение в предложении **Where** выполнялось для всех значений, объединяемых этим оператором.

Следующий запрос выбирает любую запись о клиентах, содержащую в колонке аббревиатур штатов значения "NY", "MA" или "PA". Функция Any() работает так же, как и оператор "IN" в обычном языке SQL.

```
Select * From customers
Where state = Any ("NY", "MA", "PA")
```

В предложение **Where** может быть включен самостоятельный оператор **Select**, называемый "подзапросом". В следующем примере используются две таблицы: PRODUCTS, содержащая записи о продаваемых товарах, и ORDERS — с данными о заказах на товары. Оплаченные товары могут в данный момент не находиться на складе. Задача состоит в том, чтобы выяснить номенклатуру заказанных товаров, находящихся на складе. Другими словами, нужно выбрать все заказы, которые не находятся в списке распроданных товаров.

```
Select * From orders
Where partnum <>
All(Select partnum from products
where not instock)
```

Во второй строке запроса слово **Select** появляется второй раз, уже как подзапрос. Этот подзапрос выбирает все товары, которых в данный момент **НЕТ** на складе. Главный запрос **Where** выполняет поставленную задачу с помощью функции All() и результатов подзапроса. (Слово "instock" означает "на складе"; это логическая переменная).

В примере, приведенном выше, подзапрос создает набор значений, а предложение **Where** главного запроса проверяет условие запроса для них. В других случаях подзапрос может использовать функции обобщения, вычисляющие одно значение.

В следующем примере используется функция Avg() для вычисления среднего значения в поле Population ("Население") таблицы STATES.

В результате выполнения оператора **Select** выбираются записи о населении выше среднего.

```
Select * From states
Where population >
(Select Avg(population) From states)
```

MapInfo Pro также поддерживает принятое в SQL ключевое слово In. В операторе **Select** можно использовать ключевое слово In в операторе вместо = Any. Другими словами, предложение **Where** с ключевым словом **Any**:

```
Where state = Any ("NY", "MA", "PA")
```

эквивалентно предложению **Where** с ключевым словом **In**:

```
Where state In ("NY", "MA", "PA")
```

а фраза **Not** в запросе эквивалентна фразе: <> All.

Примечание: Простой оператор **Select** может не содержать множественных невложенных подзапросов. Оператор **Select** не поддерживает в MapBasic "синхронные" или "коррелирующие" подзапросы, то есть такие, которые ссылаются на внешние таблицы. Оператор **Select** не поддерживает в MapBasic "синхронные" или "коррелирующие" подзапросы, то есть такие, которые ссылаются на внешние таблицы. Внутренние запросы перезапрашиваются для каждой строки внешней таблицы. Таким образом достигается корреляция. Пример:

```
' Note: the following statement, which illustrates
' correlated subqueries, will NOT work in MapBasic

Select * from leads
Where lead.name =
  (Select var.name From vars
   Where lead.name = customer.name)
```

Последнее замечание адресовано профессионалам, привыкшим составлять сложные SQL-запросы к другим базам данных.

Предложение Into

Задаёт имя для результирующей таблицы. Если предложения **Into** нет в операторе, то таблица с результатами выбора будет названа именем "Selection". Если последующие запросы обращаются к таблице "Selection", то результаты запросов будут называться именами QUERY*n* (например QUERY1).

Если в предложении используется ключевое слово **Noselect**, то оператор формирует запрос, не меняя предыдущую таблицу "Selection". То есть те записи, которые уже были выбраны после оператора с ключом **Noselect**, останутся выбранными.

Если в операторе использовалось ключевое слово **Noselect**, то Процедура **SelChangedHandler** не реагирует на изменение выборки и не запускается.

Если включить ключевое слово **Hide**, то таблица запроса будет создаваться скрытой. Это полезно, когда создаётся запрос, в котором используется запрограммированная обработка, но не требуется, чтобы результаты отображались в интерфейсе программы, например, список таблиц. Использование ключевого слова **Hide** поэтому может иногда существенно уменьшить лишние накладные расходы, если программа выполняет несколько операторов `Select ... Into... statements.`

Предложение Group By

Предложение **Group By** определяет порядок группировки строк при выполнении обобщающих действий (вычислений промежуточных сумм и других значений). В предложении **Group By** обычно задается имя колонки (или несколько имен колонок), на основании значений из которых группируются промежуточные результаты. Например, если нужно обобщить какие-либо данные

для областей России, то в предложении **Group By** должна быть задана колонка, содержащая имена областей России. Предложение **Group By** может не ссылаться на функцию, возвращающую переменную, такую как [функция ObjectInfo\(\)](#).

Функции обобщения **Sum()**, **Min()**, **Max()**, **Count(*)**, **Avg()**, and **WtAvg()** позволяют вычислять обобщённые результаты.

Дополнительно можно пользоваться пространственными обобщениями, например, **AggregateCombine()**, **AggregateBuffer()**, **AggregateMBR()** и **AggregateConvexHull()** в списке колонок *expression_list*, подобно обычным функциям обобщения.

Примечание: В списке выражений столбцов допускается только одна пространственная агрегатная функция.

Примечание: Пространственные агрегатные функции не могут использоваться как подвыражения.

Обычно функции обобщения и пространственного обобщения не используются в предложении **Group By**. Они помещаются в список *expression_list* оператора **Select**, а предложение **Group By** только задаёт группирующую колонку.

В следующем примере таблица Q4SALES содержит информацию о продажах за четвертый квартал. Каждая запись содержит информацию о сумме каждой сделки в колонке "amount". В колонке "territory" задана территория, на которой произошла сделка. В запросе вычисляется, сколько сделок произошло на каждой территории и суммарный объем этих сделок.

```
Select territory, Count(*), Sum(amount)
From q4sales
Group By territory
```

Предложение **Group By** заставляет MapBasic сначала сгруппировать записи по "территориям", а затем сосчитать сумму сделок на каждой "территории". Запрос **Select** создает три колонки: имя территории, количество записей в таблице Q4SALES, относящихся к данной территории, и сумму сделок на этой территории.

Примечание: Функция **Sum()** принимает в качестве параметра имя колонки. Функция **Count()** с параметром-звездочкой просто пересчитывает количество записей в каждой из групп. Функция **Count()**, в отличие от других функций обобщения, не нуждается в аргументах-именах колонок.

В следующей таблице приведено описание функций обобщения данных:

Функция	Описание	Тип
Avg(column)	Возвращает среднее значение из значений в колонке.	вещественное – float

Функция	Описание	Тип
Count (*)	Возвращает количество записей в группе. Вместо имен колонок введите ** (звездочку).	целое – integer
Max (column)	Возвращает наибольшее значение из значений в колонке column для всех записей группы.	вещественное – float
Min (column)	Возвращает наименьшее значение из значений в колонке column для всех записей группы.	вещественное – float
Sum (column)	Возвращает сумму значений в колонке column для всех записей группы.	вещественное – float
WtAvg (column , weight_column)	Возвращает среднее взвешенное значение из значений в колонке column для всех записей группы. Примеры см. ниже.	вещественное – float

В следующей таблице приведено описание функций обобщения данных MapInfo Pro:

Функция	Описание	Тип
AggregateBuffer (geometry, resolution, width, unit_name)	Возвращает объект типа "Область", представляющий собой буферную зону вокруг выбранного объекта (область, граница которой отстоит от границы объекта на заданное расстояние). <i>Geometry</i> — выражение, которое возвращает геометрические результаты. <i>resolution</i> – число узлов многоугольника, принимаемого как окружность (число типа SmallInt); <i>Width</i> — вещественное значение – Float, представляющее радиус буферной зоны. <i>unit_name</i> – имя единицы измерения расстояний (например, "mi" – миля, "km" – километр).	объект

Функция	Описание	Тип
AggregateCombine (<i>geometry</i>)	Возвращает геометрический объект, который представляет совмещение(объединение) агрегированного набора геометрических объектов. <i>Geometry</i> — выражение, которое возвращает геометрические результаты.	объект
AggregateConvexHull (<i>geometry</i>)	Возвращает геометрический объект из нескольких полигонов, который представляет собой выпуклую оболочку вокруг набора точек, вложенных в агрегированное, объединённое множество геометрических объектов. <i>Geometry</i> — выражение, которое возвращает геометрические результаты.	объект
AggregateMBR (<i>geometry</i>)	Возвращает геометрический объект типа Контейнер (Envelope), который представляет собой минимальный ограничивающий прямоугольник (MBR) агрегированного, объединённого набора геометрических объектов. <i>Geometry</i> — выражение, которое возвращает геометрические результаты.	объект

Примеры пространственных обобщений:

```
Select Aggregatecombine(obj) from usa_caps, states where usa_caps.state = states.state //combine of all objects from usa_caps table.
```

```
Select Aggregatebuffer(obj, 20, 10, "mi") from usa_caps //Buffer and Combine of all the objects from usa_caps table.
```

```
Select AggregateCombine(world.obj) from World group by World.Continent Into Continents
```

Примечание: Ни одна функция MapBasic не возвращает десятичное значение. Десятичные поля являются только формой представления данных. Все вычисления проводятся с вещественными числами.

Вычисление средних взвешенных

Функция вычисления взвешенной средней величины **WtAvg()** использует значения из дополнительной колонки в качестве коэффициентов значений из основной колонки. Например,

следующий оператор использует функцию **Wtavg()** для вычисления уровня грамотности на каждом континенте:

```
Select continent, Sum(pop_1994), WtAvg(literacy, Pop_1994)
From World
Group By continent
Into Lit_query
```

Предложение **Group By** задаёт группировку строк таблицы. MapInfo Pro группирует записи по значениям из колонки "Континент". Все записи, в которых в колонке "Континент" встречается "Северная Америка", будут считаться одной группой; все записи, в которых в колонке "Континент" встречается "Азия", будут считаться другой группой и т.п.. Для каждой группы записей, другими словами для каждого континента — MapInfo Pro вычисляет средневзвешенные значения уровней грамотности.

Простое среднее (функция **Avg()**) вычисляется как сумма, деленная на количество слагаемых. Взвешенное среднее (функция **WtAvg()**) более сложна; при вычислении взвешенного среднего одни значения могут иметь больший вес и больше влиять на результат. В нашем примере среднее значение грамотности вычисляется с учетом населения (колонка "Нас_1994"); другими словами, страны с большим населением вносят больший вклад в результат.

Выражения, задающие колонки, в предложении **Group By**

В предложении **Group By** колонки могут задаваться именем, например, как в приведенном выше примере группировки по территориям. Альтернативным способом задания колонки **Group By** является использование ее порядкового номера в таблице, используя выражение вида *col#*. В таком выражении вместо знака # должно стоять целое число (единица и более), указывающее одну из колонок в предложении **Select**. Таким образом, упомянутый выше оператор **Select** может использовать *Group By col1* или даже *Group By 1*, а не **Group By**.

Альтернативный синтаксис в обозначениях колонок бывает необходим в тех случаях, когда нужно обозначить колонку, имя которой является результатом вычислений. В следующем примере функция **Month()** используется для вычисления группирующих значений, но название колонки из выражения получить нельзя. Поэтому нужно применить альтернативный синтаксис:

```
Select Month(sick_date), Count(*)
From sickdays
Group By 1
```

В этом примере каждая запись таблицы SICKDAYS ("Болезни") должна содержать дату болезни. В результате запроса образуется таблица из 12 строк (по количеству месяцев); во второй колонке будет содержаться количество дней, пропущенных в этом месяце по болезни.

Группирование нескольких колонок

В некоторых случаях Вам надо будет задать более одного имени колонки в предложении **Group By**. Это нужно, когда информации в одной колонке недостаточно для точного задания условия группировки. Пусть имеется карта районного деления России; известно, что названия некоторых районов, принадлежащих разным областям, совпадают. Например, Первомайский район есть в нескольких областях. Поэтому, определив в предложении **Group By** только

название района, в результате запроса можно получить недостоверные данные – обобщенные по всем районам с именем "Первомайский", независимо от того, в какой области они находятся.

Чтобы решить эту проблему (или аналогичную), задайте колонку имен областей, как дополнительную в предложении **Group By**. Например, это предложение может выглядеть следующим образом:

```
Group By county, state
```

В этом случае для каждой уникальной пары *район, область* будет произведена отдельная группировка. Суммарные значения для пары "Первомайский-Нижегородская" будут различаться со значениями для пары "Первомайский-Дагестан".

Предложение Order By

Предложение Order By задает колонку или колонки, по значениям которых должна происходить сортировка в результирующей таблице. Также как и в предложении **Group By**, колонки задаются списком имен полей или списком номеров этих полей. Элементы списка разделяются запятыми.

Стандартный порядок сортировки **Order By** — по возрастанию, то есть от "А" до "Z", от "А" до "Я" и от 0 до 9. Сортировка по убыванию задается ключевым словом **Desc** как показано в следующем примере:

```
Select * From cities
Order By state, population Desc
```

Этот запрос выполняет двухуровневую сортировку таблицы CITIES. Сначала MapBasic сортирует по возрастанию названий штатов, затем внутри каждой группы, сортирует по убыванию населения городов. Слово **Desc** отделяется пробелом, а не запятой.

Предложение **Order By** не может содержать вызов функции, возвращающей переменное значение. Например, так как функция **ObjectInfo()** возвращает переменное значение, она не может быть использована для сортировки в операторе Select.

Предложение Limit

Предложение Limit позволяет ограничивать количество записей, возвращаемых оператором выбора Select. Этим ограничением можно пользоваться для получения образцов результатов или, для того чтобы проверить запрос без долгого ожидания результатов выполнения запроса. Оно также полезно для получения заданного количества *n* самых старших или *n* самых младших данных таблицы. Значение *n* должно быть больше 0. Если *n* — отрицательное, то появится сообщение об ошибке. Код MapBasic для этой ошибки — 1055.

Пример использования предложения LIMIT.

```
Select * From Countries Where Pop_1994 > 2000000 And Pop_1994 < 3000000
Limit 5
```


В приведённом выше примере запрос останавливается после того, как найдены первые пять результатов с условием *WHERE*.

```
Select * From Countries Order By Pop_1994 Desc Limit 5
```

В приведённом примере результатов перечислены пять государств с наибольшей численностью населения в 1994 году, начиная от самого многочисленного до самого малочисленного. Поскольку используется предложение *Order By*, запрос будет обрабатывать все записи, прежде чем мы узнаем пять лучших результатов. Это означает, что потребуются долгое ожидание, прежде чем появятся ограниченные результаты.

```
Select * From Countries Where Pop_1994 > 2000000 And Pop_1994 < 3000000
Order By Pop_1994 Desc
Limit 5
```

В примере, приведённом выше, первые пять результатов ищутся с помощью условия *WHERE*, а затем полученные результаты сортируются по колонке "Pop_1994" от младшего к старшему значению, и полученные пять лучших результатов выдаются в заданном порядке.

Географические операторы

MapBasic поддерживает несколько географических операторов: Contains, Contains Part, Contains Entire, Within, Partly Within, Entirely Within, и Intersects. Они используются в любых выражениях и особенно полезны в предложении **Where** для задания критерия выбора на основании взаимного расположения объектов на Карте. Все графические операторы работают только со значениями объектного типа и в результате дают логическую величину. Операторы перечислены в следующей таблице.

Применение	Возвращает TRUE, если:
objectA Contains objectB	центроид второго объекта находится внутри первого объекта
objectA Contains Part objectB	часть второго объекта содержится внутри первого объекта
objectA Contains Entire objectB	весь второй объект содержится внутри первого объекта
objectA Within objectB	центроид первого объекта внутри второго объекта
objectA Partly Within objectB	часть первого объекта помещается внутри второго
objectA Entirely Within objectB	первый объект полностью помещается внутри второго

Применение	Возвращает TRUE, если:
objectA Intersects objectB	Два объекта пересекаются хотя бы в одной точке

Ускоренный выбор

Некоторые варианты оператора **Select** выполняют операцию выбора быстрее, чем другие. Скорость выбора зависит от содержания предложения **Where**, задающего критерий выбора.

Если после слова **Where** стоит одно выражение в форме:

```
columnname = constant_expression
```

(где columnname — имя колонки, а constant_expression — выражение из постоянных строковых величин) или выражений такой формы несколько и они разделены операционными словами And, то такой оператор **Select** будет выполняться быстрее, так как в этом случае максимально используются преимущества индексации. Если подобные выражения объединены оператором Or, то преимущества индексации не используются.

Также MapInfo Pro сравнительно быстрее выполняет выбор с предложениями *Where* вида:

```
[ tablename. ] obj geographic_operator object_expression
```

(где tablename — имя таблицы, geographic_operator — географический оператор, object_expression — выражение из постоянных строковых величин) и вида:

```
RowID = constant_expression
```

RowID — это имя специальной колонки, содержащей номер записи в таблице и обычно скрытой от пользователя. Каждое значение RowID является номером строки соответствующей таблицы; другими словами, первая строка в таблице имеет RowID равным единице.

Перекрёстные объединения

Можно объединять две таблицы и фильтровать по условиям полученные результаты. В дополнение к использованию операции "=" и пространственных операторов можно объединять таблицы с помощью операторов сравнения "<", ">", "<=", ">=" и "<>", что приводит к перекрёстному объединению двух таблиц. Можно использовать выражения с переменными, константами или колонками с любой стороны операнда.

Примечание: Два условия могут быть объединены только с помощью оператора AND.

Например, для того чтобы найти объекты рядом друг с другом:

```
Select A.ID, B.NAME From SomeTable1 as "A", SomeTable2 as "B" Where
ObjectDistance(A.OBJ, B.OBJ, "m") < 100
```

Задание таблицы с объектами при объединении

Можно задавать таблицу, объекты которой будут использоваться, и добавлять напрямую на карту с результатами запроса объекты, полученные с помощью выражений. Требуется задать ключевое слово `object`, для того чтобы выбрать таблицу или выражение, объекты которых будут добавлены на карту с результатами запроса.

Примечание: В качестве поля с объектами можно задавать единственную колонку. В случае, когда задаётся пространственная функция обобщения, поле с объектами не может быть повторно задано.

```
Select usa_caps.state, states.obj object from usa_caps,states where
usa_Caps.state = states.state //In case of join the object used to come
from first table by default, but in this query we
    have specified the object keyword to states.obj therefore now the
query will have object from
    states table.
```

Если выражение для объектов присутствует в списке колонок, например, `"combine(states.obj, usa_caps.obj) as "Combine" object"`, то другие графические колонки будут обрабатываться как колонки символьного типа `Char`. Например,

```
Open Table "D:\Maps\us\usa_caps.TAB" Interactive
Open Table "D:\Maps\Imp_data\STATES.TAB" Interactive
select usa_caps.state, states.obj as "StateObj", Buffer(OBJ, 20, 10,
"mi") "BufferObj" Object from usa_caps, states where usa_Caps.state =
states.state into query
Map From query
set map window frontwindow() layer 0 Editable On
Set Map Window frontwindow() Layer 1 Display Off
Dim aOBJ as Object
fetch first from query
aOBJ = query.StateObj ---> Here you will get cannot convert data error
because we have a object expression in query.
Insert Into WindowInfo(frontwindow(), 10) (OBJ) Values(aOBJ)

select usa_caps.state, states.obj as "StateObj" object, Buffer(OBJ, 20,
10, "mi") "BufferObj" from usa_caps, states where usa_Caps.state =
states.state into query
Map From query
set map window frontwindow() layer 0 Editable On
Set Map Window frontwindow() Layer 1 Display Off
Dim aOBJ as Object
```

```
fetch first from query
aOBJ = query.BufferObj --> This will work, because we don't have object
expression in query. We just have specified which table the object
should come from.
Insert Into WindowInfo(frontwindow(), 10) (OBJ) Values(aOBJ)
```

Примеры

В этом примере выбираются все клиенты в Нью-Йорке, Коннектикуте или штате Массачусетс. Каждая запись о клиенте не обязательно содержит имя области; запрос вместо этого опирается на географическое расположение клиента.

```
Select * From customers
Where obj Within Any(Select obj From states
Where state = "NY" or state = "CT" or state = "MA")
```

В этом примере демонстрируется действие подзапроса. Мы желаем выбрать все территории сбыта, в которых проживают клиенты, имеющие признак "Federal". Подзапрос сначала выбирает подмножество записей о клиентах с признаком "Federal", после чего основной запрос выбирает территории по значениям из подмножества.

```
Select * From territories
Where obj Contains Any (Select obj From customers
Where customers.source = "Federal")
```

Следующий запрос выбирает все земельные участки, пересекающиеся с участком номер 120059.

```
Select * From parcels
Where obj Intersects (Select obj From parcels
Where parcel_id = 120059)
```

См. также:

[Оператор Open Table](#)

Функция SelectionInfo()

Назначение

Возвращает информацию о текущем выборе в таблицах (информацию о временной таблице "Selection"). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Примечание: Выбранные подписи не образуют выборку, так как сами являются атрибутами других объектов и не являются отдельными объектами.

Синтаксис

```
SelectionInfo ( attribute )
```

attribute — целочисленный код, см. таблицу ниже.

Возвращаемая величина

Целое число (Integer) или строка (String) в зависимости от значения *attribute*.

Описание

В файле стандартных определений MapBasic (MAPBASIC.DEF) определены имена для кодов, которые можно использовать в функции SelectionInfo.

Параметры	ID	Возвращаемое значение SelectionInfo()
SEL_INFO_TABLENAME	1	Строка, величина типа String: Имя таблицы, на базе которой создана таблица "Selection". Если ничего не выбрано, будет возвращена ошибка.
SEL_INFO_SELNAME	2	Строка, величина типа String: Имя временной таблицы (например, "Query1" или "Запрос1"). Если ничего не выбрано, будет возвращена ошибка.
SEL_INFO_NROWS	3	Целое число типа Integer: Количество выбранных строк в таблице. Если ничего не выбрано, будет возвращен 0 (ноль).
SEL_INFO_ISNEW	4	Логическое: возвращает Т (true), если текущая выборка является новой.

Примечание: Если записи выбраны в таблице, которая является объединением двух или более таблиц, то функция **SelectionInfo(SEL_INFO_NROWS)** возвращает количество строк в базовой таблице, которое может быть меньшим, чем количество строк в таблице "Selection". Пример смотрите ниже.

Условия появления сообщения об ошибке

ERR_FCN_ARG_RANGE (644) — сообщение об ошибке создаётся, если неправильно задано значение аргумента.

Пример

Оператор Select используется для объединения. После выполнения запроса переменная *i* будет равна 40 (числу строк, выбранных в базовой таблице STATES), а переменная *j* будет равна 125 (числу строк таблицы запроса).

```
Dim i, j As Integer
Select * From States, City_125
  Where States.obj Contains City_125.obj Into QResults
i = SelectionInfo(SEL_INFO_NROWS)
j = TableInfo(QResults, TAB_INFO_NROWS)
```

См. также:

оператор Select, функция TableInfo()

Оператор Server Begin Transaction

Назначение

Посылает уведомление на удаленный сервер о начале нового сеанса работы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Begin Transaction
```

ConnectionNumber — целое число, идентификатор соединения.

Описание

Оператор **Server Begin Transaction** используется для обозначения начала сеанса обработки транзакций. Результаты последующих операторов языка SQL "Insert", "Delete" и "Update" (внести, удалить и обновить), которые выполняет **функция Server_Execute()**, не сохраняются в базе данных до тех пор, пока не будет выполнен **оператор Server Commit**. **Оператор Server Rollback** используется для отмены изменений.

Пример

```
Dim hdbc As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
Server hdbc Begin Transaction
```

```
' ... other server statements ...
Server hdbc Commit
```

См. также:

[оператор Server Commit](#), [оператор Server Rollback](#)

Оператор Server Bind Column

Назначение

Назначает локальную область хранения для удаленного сервера базы данных. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server StatementNumber Bind Column n To Variable, StatusVariable
```

StatementNumber — целое значение, номер SQL-оператора.

n — номер столбца в результирующем наборе, связываемого с переменной.

Variable — MapBasic-переменная для хранения значения столбца после выборки.

StatusVariable — переменная состояния, в которую записывается код, указывающий статус значения: пустое, усеченное или целое положительное значение.

Описание

Оператор Bind Column назначает переменную приложения для сохранения значения столбца в результирующем наборе, который задает удаленный SQL-[оператор Select](#). Когда последующий [оператор Server Fetch](#) выбирает строку данных из базы, значение связанного столбца присваивается переменной, которую задает [оператор Server Bind Column](#). Статус результата сохраняется в переменной состояния, указанной параметром StatusVariable.

Значение StatusVariable	ID	Условие
SRV_NULL_DATA	-1	Возвращается, если столбец не имеет данных в полученной строке (пустое значение).
SRV_TRUNCATED_DATA	-2	Возвращается, если столбец содержит больше данных, чем может быть сохранено в указанной MapBasic-переменной.

Значение StatusVariable	ID	Условие
Целое положительное значение		Число байт, возвращенное сервером данных.

Пример

Далее — приложение "печатающее" подписи с адресами. Предполагается, что существует реляционная таблица ADDR с 6 столбцами

```
Dim hdbc, hstmt As Integer
Dim first_name, last_name, street, city, state, zip As String
Dim fn_stat, ln_stat, str_stat, ct_stat, st_stat, zip_stat As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "select * from ADDR")
Server hstmt Bind Column 1 To first_name, fn_stat
Server hstmt Bind Column 2 To last_name, ln_stat
Server hstmt Bind Column 3 To street, str_stat
Server hstmt Bind Column 4 To city, ct_stat
Server hstmt Bind Column 5 To state, st_stat
Server hstmt Bind Column 6 To zip, zip_stat
Server hstmt Fetch NEXT
While Not Server_Eot(hstmt)
  Print first_name + " " + last_name
  Print street
  Print city + ", " + state + " " + zip
  Server hstmt Fetch NEXT
Wend
Server hstmt Close
Server hdbc Disconnect
```

См. также:

[функция Server_ColumnInfo\(\)](#)

Оператор Server Close

Назначение

Освобождает ресурсы, занятые удаленным SQL-оператором доступа к данным. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server StatementNumber Close
```

StatementNumber — целое значение, номер SQL-оператора.

Описание

Оператор **Server Close** используется для оповещения сервера о завершении обработки текущего удаленного SQL-оператора. Все ресурсы, ассоциированные с исполнением этого оператора, возвращаются в распоряжение системы. Не забывайте вызывать **оператор Server Close** сразу же после того, как отработает **функция Server_Execute()** для любого не выполняющего запрос данных SQL-оператора, обработка которого закончена в приложении.

Пример

```
' Fetch the 5th record then close the statement
hstmt = Server_Execute(hdbc, "Select * from Massive_Database")
Server hstmt Fetch Rec 5
Server hstmt Close
```

См. также:

функция Server_Execute ()

Функция Server_ColumnInfo()

Назначение

Возвращает информацию о столбцах результирующего набора. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_ColumnInfo( StatementNumber, ColumnNo, Attr )
```

StatementNumber — целое значение, номер SQL-оператора.

ColumnNo — номер столбца в наборе; нумерация слева направо, начиная с 1.

Attr — код, указывающий на возвращаемую колонку.

Возвращаемая величина

Возвращаемое значение зависит от значения атрибута (параметр *Attr*).

Описание

Функция **Server_ColumnInfo** возвращает информацию о текущем выбранном столбце результирующего набора, который перед этим был определил SQL-оператор **Select**. Параметр *StatementNumber* задает номер-указатель (handle) SQL-оператора, ассоциированный с данным соединением с сервером данных. Параметр *ColumnNo* указывает столбец, информацию о котором Вы хотите получить. Параметр *Attr* выбирает тип возвращаемой информации.

В следующей таблице перечислены возможные атрибуты параметра *Attr*. Эти типы определены в MAPBASIC.DEF.

Attr-код	ID	Возвращает Server_ColumnInfo():
SRV_COL_INFO_NAME	1	Строковый результат с именем колонки.
SRV_COL_INFO_TYPE	2	Целый результат; код типа столбца: • SRV_COL_TYPE_NONE (0) • SRV_COL_TYPE_CHAR (1) • SRV_COL_TYPE_DECIMAL (2) • SRV_COL_TYPE_INTEGER (3) • SRV_COL_TYPE_SMALLINT (4) • SRV_COL_TYPE_DATE (5) • SRV_COL_TYPE_LOGICAL (6) • SRV_COL_TYPE_FLOAT (8) • SRV_COL_TYPE_FIXED_LEN_STRING (16) • SRV_COL_TYPE_BIN_STRING (17) Информацию об интерпретации типов данных приложением MapInfo Вы можете найти в Руководстве пользователя MapBasic.
SRV_COL_INFO_WIDTH	3	Целый результат, указывающий максимальное число разрядов для столбцов типа SRV_COL_TYPE_CHAR (1) или SRV_COL_TYPE_FIXED_LEN_STRING (16). При использовании модуля ODBC пустой терминатор не учитывается. Возвращаемое значение совпадает с шириной столбца таблицы базы данных.
SRV_COL_INFO_PRECISION	4	Целый результат, указывающий общее число разрядов для столбца типа SRV_COL_TYPE_DECIMAL (2), или -1 для столбца любого другого типа.

Attr-код	ID	Возвращает Server_ColumnInfo():
SRV_COL_INFO_SCALE	5	Целый результат, указывающий число разрядов справа от десятичной точки для столбца типа SRV_COL_TYPE_DECIMAL (2), или -1 для столбца любого другого типа.
SRV_COL_INFO_VALUE	6	Тип результата варьируется. Возвращается текущее значение данных в столбце для текущей выбранной записи. Длинные строковые значения столбца, превышающие 32766 символов, усекаются. Двоичные (неструктурированные) значения столбца возвращаются в виде шестнадцатеричных символьных строк двойной длины.
SRV_COL_INFO_STATUS	7	Целый результат; статус значения столбца: • SRV_NULL_DATA (-1) Возвращается, если столбец не имеет данных в полученной строке (пустое значение). • SRV_TRUNCATED_DATA (-2) Возвращается, если столбец содержит больше данных, чем может быть сохранено в указанной MapBasic-переменной. • Целое положительное значение Число байт, возвращенное сервером данных.
SRV_COL_INFO_ALIAS	8	Строковый результат; псевдоним столбца, если в запросе данных использовался псевдоним.

Пример

```

Dim hdbc, Stmt As Integer
Dim Col As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
Stmt = Server_Execute(hdbc, "Select * from emp")
Server Stmt Fetch NEXT
For Col = 1 To Server_NumCols(Stmt)
  Print Server_ColumnInfo(Stmt, Col, SRV_COL_INFO_NAME) +
    " = " +
    Server_ColumnInfo(Stmt, Col, SRV_COL_INFO_VALUE)
Next

```

См. также:

оператор Server Bind Column, оператор Server Fetch, функция Server_NumCols()

Оператор Server Commit

Назначение

Вызывает фиксацию транзакции в удаленной базе данных. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Commit
```

ConnectionNumber — целое число, идентификатор соединения.

Описание

Оператор **Server Commit** фиксирует транзакцию, т.е. сохраняет в базе данных изменения, произведенные всеми удаленными SQL-операторами, которые в данном сеансе соединения открывал **оператор Server Begin Transaction**. Оператор **Server Commit** выполняется только при наличии открытой транзакции, которые открывал **оператор Server Begin Transaction**. Для запуска новой транзакции необходимо передать серверу новый **оператор Server Begin Transaction**, за которым в дальнейшем должен быть исполнен оператор **Server Commit**.

Пример

```
hdbc = Server_Connect("ODBC", "DLG=1")  
Server hdbc Begin Transaction  
hstmt = Server_Execute(hdbc, "Update Emp Set salary = salary * 1.5")  
Server hdbc Commit
```

См. также:

оператор Server Begin Transaction, оператор Server Rollback

функция Server_Connect()

Назначение

Устанавливает соединение с удаленным сервером данных. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_Connect( toolkit, connect_string )
```

toolkit – указывает модуль интерфейса удаленного доступа к MapInfo, например, ORAINET™ и FDO. Значения *toolkit* обычно возвращает [функция Server_DriverInfo\(\)](#).

connect_string – параметр, который предоставляет интерфейсному модулю дополнительную информацию, необходимую для подключения к серверу данных.

Возвращаемая величина

Целое – Integer

Описание

Функция **Server_Connect()** выполняет соединение с базой данных и возвращает номер – указатель соединения, который должен быть передан всем операторам удаленного доступа (как параметр ConnectionNumber), которые Вы хотите выполнить в данном сеансе соединения с сервером данных.

Параметр *toolkit* определяет модуль интерфейса удаленного доступа MapInfo Pro (динамически загружаемую библиотеку), через который будет осуществляться соединение с сервером данных. Информацию о возможных значениях параметра могут вернуть [функция Server_NumDrivers\(\)](#) и [функция Server_DriverInfo\(\)](#).

Параметр *connect_string* передает модулю toolkit дополнительную информацию, необходимую для подключения к серверу данных. Значение строки подключения определяется требованиями удаленного сервера данных, к которому осуществляется доступ.

Строка подключения, задаваемая в функции **Server_Connect()**, имеет формат:

```
attribute=value[;attribute=value...]
```

Примечание: В строке подключения должны отсутствовать пробелы.

Прохождение соединения с DLG=1 обеспечивает удобный диалог соединения с активными кнопками справочной системы.

Смотрите один из следующих разделов о типах баз данных, с которыми выполняется соединение:

- [Соединение с Microsoft ACCESS](#)
- [Соединение с SQLite](#)
- [Соединение с SQL SERVER](#)
- [Соединение с Oracle Spatial](#)
- [Соединение с POSTgreSQL](#)

Соединение с Microsoft ACCESS

Пример строки подключения для Microsoft ACCESS:

```
"DSN=MI ACCESS;UID=ADMIN;PWD=SECRET"
```

В следующей таблице перечислены атрибуты, используемые СУБД ACCESS:

attribute-код	Описание
DSN	Имя ODBC-источника для Microsoft ACCESS.
UID	Регистрационный идентификатор пользователя (ID).
PWD	Пароль пользователя.
SCROLL	По умолчанию значение равно NO. Если SCROLL=NO, то библиотека ODBC не используется для этого соединения, дающего возможность вызывать первую, последнюю, предыдущую или произвольную запись в базе данных.

В этом примере описывается соединение с файлом MS ACCESS, который называется atstest.mdb:

```
Server_connect("ODBC", "DRIVER=Microsoft Access Driver (*.mdb, *.accdb);  
MDB;DBQ=atstest.mdb;DefaultDir=C:\database\odbc\;DriverId=25;FIL=MS  
Access;MaxBufferSize=2048;PageTimeout=5;SCROLL=No")
```

Соединение с SQLite

Необходимо указать "FDO" в качестве типа *toolkit*, чтобы соединиться с базой данных SQLite. Примеры соединения с базами данных SQLite смотрите в разделе *Доступ/Обновление удаленных баз данных при помощи связанных таблиц* в *Руководстве пользователя MapBasic*.

Примечание: Вместе с MapInfo Pro устанавливается модуль доступа данных к базам данных SQLite – Feature Data Object (FDO). Заботиться о том, как получить FDO, не требуется.

Пример строки подключения для СУБД SQLite:

```
"File=C:\Data\MIPRO71252.sqlite;PROVIDER=OSGeo.SQLite;SCROLL=NO"
```

В следующей таблице перечислены атрибуты, используемые СУБД SQLite:

attribute-код	Описание
Файл	Полный маршрут к файлу базы данных SQLite.
PROVIDER	Формат данных, например, Open Source Geospatial Foundation (OSGeo).
SCROLL	По умолчанию значение равно NO. Если SCROLL=NO, то библиотека FDO не используется для этого соединения, дающего возможность вызывать первую, последнюю, предыдущую или произвольную запись в базе данных.

В следующем примере открывается связанная таблица базы данных SQLite (FDO). Пример:

1. Устанавливается соединение с базой данных SQLite, которая называется MIPRO71252.sqlite.
2. Открывается таблица, которая называется EN_STATE, в родном, нативном формате NativeX (расширенном MapInfo) с использованием набора символов UTF-8 таким образом, что открытая таблица оказывается связанной с базой данных. В этом примере username – учётная запись пользователя Windows.
3. Открывается окно **Карта**, в котором отображаются данные EN_STATE.

```
dim i as integer
i = Server_Connect("FDO",
"File=C:\Data\MIPRO71252.sqlite;PROVIDER=OSGeo.SQLite;SCROLL=NO")

Server i Link Table "select * from EN_STATE" Toolkit "FDO" INTO "EN_STATE"
FILE "C:\Users\username\Documents\EN_STATE.tab" TYPE NATIVEX Charset
"UTF-8"

map from EN_STATE
```

В следующем примере выполняется соединение с базой данных SQLite и открывается таблица прямого доступа с включённым кэшированием. Пример:

1. Устанавливается соединение с базой данных SQLite, которая называется MIPRO71252.sqlite.
2. Регистрируется таблица с именем EN_STATE, определяется, что в ней используется набор символов UTF-8, и открывается расширенном формате MapInfo (NativeX). В этом примере username – учётная запись пользователя Windows.
3. Таблица EN_STATE открывается в интерактивном режиме, чтобы вносить изменения напрямую в таблицу базы данных.
4. Открывается окно **Карта**, в котором отображаются данные EN_STATE.

```
dim i as integer
i = Server_Connect("FDO",
"File=C:\Data\MIPRO71252.sqlite;PROVIDER=OSGeo.SQLite;SCROLL=NO")
```

```
Register Table "EN_STATE" Type ODBC Charset "UTF-8" Table "select * from
EN_STATE" Connection Handle i Toolkit "FDO" Cache ON TYPE NATIVEX Into
"C:\Users\username\Documents\EN_STATE.tab"
```

```
Open Table "C:\Users\username\Documents\EN_STATE.tab" Interactive
```

```
Map From EN_STATE
```

Пропуск предложения `Charset` и `TYPE` при регистрации таблицы приводит к тому, что она будет открыта в традиционном нативном формате MapInfo с использованием стандартного системного набора символов операционной системы Windows.

Замена предложения `Handle` маршрутом к базе данных SQLite открывает таблицу без предварительного соединения с базой данных (поэтому не требуется оператор `dim . . . Server_Connect`).

```
Register Table "EN_STATE" Type ODBC Charset "UTF-8" Table "select * from
""EN_STATE"" Connection
"File=C:\Data\MIPRO71252.sqlite;PROVIDER=OSGeo.SQLite;SCROLL=NO" Toolkit
"FDO" Cache ON TYPE NATIVEX Into
"C:\Users\username\Documents\EN_STATE.tab"
```

Примечание: Вместе с MapInfo Pro устанавливается модуль доступа данных к базам данных SQLite – Feature Data Object (FDO). Заботиться о том, как получить FDO, не требуется.

Соединение с SQL SERVER

Если приложение требует передачи строки параметров для обеспечения подключения к источнику данных, необходимо указать имя источника данных. Это имя, источника данных уже настроенного для использования в системе, который будет вызван драйвером, для определения стандартных параметров, обеспечивающих подключение. Кроме того, можно явно указать пары “*параметр=значение* параметра”, которые могут быть использованы для подключения к базе данных без применения уже зарегистрированного в системе источника данных. В этом случае значения параметров не будут записаны в качестве системных.

Строка параметров при этом выглядит следующим образом:

```
DSN=data_source_name[;attribute=value[;attribute=value]...]
```

Пример строки соединения, в котором используется JOHN в качестве идентификатора пользователя (UID) и XYZZY в качестве пароля (PWD) для SQL Server:

```
DSN=Accounting;UID=JOHN;PWD=XYZZY
```

Ниже перечислены полные имена параметров, их сокращенные формы, возможные используемые значения и краткие описания. Отдельно отмечены стандартные значения для каждого параметра, используемые при отсутствии этого параметра в строке подключения, как

при явном указании имени параметра, так и при использовании системного имени подключения. При настройке параметра в подключении, которому в Вашей системе было задано имя, такие настройки будут использованы как стандартные.

Address: сетевой адрес сервера, на котором выполняется SQL Server. Этот параметр следует использовать только в том случае, если в параметре Server явно не указано имя сервера на котором работает сервер базы данных SQL Server. Адресом может служить имя сервера в сети, а также другие способы адресации к этой вычислительной машине, например, имя или номер канала, TCP/IP адрес и номер порта, адрес программного интерфейса (socket). Например, для TCP/IP: 199.199.199.5, 1433 или MYSVR, 1433.

AnsiNPW: AnsiNPW={yes | no}. Определяет использование правил ANSI. В случае, если установлено значение, равное логической единице, драйвером используются правила ANSI (американского национального института стандартов), в которых определяются способы обработки сравнений значений с пустыми элементами (NULL), последовательности символов и конкатенации значений с пустыми (NULL) элементами. В случае, если установлено значение, равное логическому нулю, правила ANSI не применяются.

APP: определяет имя приложения, вызывающего команду SQLDriverConnect (может использоваться для дополнительной проверки условий выполнения запроса). Если этот параметр используется, то значение указанное при его вызове будет сохранено в столбце master.dbo.sysprocesses. При вызове функций sp_who и Transact-SQL APP_NAME возвращается значение program_name.

DATABASE: имя базы данных SQL Server, используемой в устанавливаемом подключении по умолчанию. Если этот параметр не задан, используется база данных определенная, как используемая по умолчанию для пользователя при входе в систему. Имя базы данных, определенное как используемое по умолчанию, в источнике данных ODBC, будет использовано вместо имени базы данных, определенной по умолчанию для пользователя при входе в систему, в случае если подключение устанавливается по имени источника данных ODBC. База данных должна существовать, кроме случая, когда применяется параметр AttachDBFileName. При совместном применении с параметром AttachDBFileName, используется основной (primary) файл, на который указывает значение параметра AttachDBFileName, а базе данных присваивается имя, определенное значением параметра DATABASE.

LANGUAGE: может использоваться как дополнительный, необязательный параметр. SQL-сервер может хранить системные сообщения на разных языках. При подключении к серверу SQL Server, для которого предусмотрена возможность выдавать системные сообщения на разных языках, можно указать язык, на котором следует создавать такие сообщения.

Network: имя сетевой библиотеки. Для этого имени не нужно указывать полного пути и расширения имени DLL-файла, например, Network=dbnmpntw.

PWD: пароль, указываемый пользователем при входе в SQL-сервер, который определен в переменной UID. PWD не нужно определять, если для входа в систему установлен пустой пароль или в случае использования авторизации пользователей средствами Windows NT (Trusted_Connection=yes).

QueryLogFile: полное имя файла, который будет использоваться для создания журнала обработки данных при продолжительных запросах.

QueryLog_On: QueryLog_On={yes | no}. Включает процесс создания журнала обработки продолжительных запросов. Если значение параметра установлено равным логической единице (Yes), то журнал обработки продолжительных запросов для этого подключения ведется. Если значение параметра установлено равным логическому нулю (No), то журнал обработки продолжительных запросов не будет вестись.

QueryLogTime: последовательность цифр, определяющих интервал (в миллисекундах) записи журнала продолжительных запросов. Если отклик на запрос не пришел в течении заданного интервала, то этот запрос будет добавлен в журнал.

QuotedID: QuotedID={yes | no}. Включает или выключает обработку идентификаторов в кавычках. Если значение параметра установлено равным логической единице (Yes), то параметр QUOTED_IDENTIFIERS включен. SQL- Server в этом случае использует правила SQL-92, определяющие использование кавычек в предложениях SQL. Если значение параметра установлено равным логическому нулю (No), то параметр QUOTED_IDENTIFIERS выключен. В этом случае SQL Server использует правила Transact-SQL, определяющие использование кавычек в предложениях SQL.

Regional: Regional={yes | no}. Включает автоматическое преобразование формы представления символов валюты, даты и времени. Если значение параметра установлено равным логической единице (Yes), то драйвер SQL Server Wire Protocol использует настройки машины клиента при преобразовании символов валюты, даты и времени. Это преобразование одностороннее; драйвер не распознает форматы символов валют или даты, не определенные стандартом ODBC. Если значение параметра установлено равным логическому нулю (No), то драйвер использует стандартные в ODBC форматы представления валюты, даты и времени и в соответствии с ними преобразует такие данные в последовательности символов.

SAVEFILE: имя файла источника данных ODBC, в который записываются атрибуты соединения при успешном подключении.

SERVER: имя сервера в сети, на котором работает SQL-сервер. В качестве такого имени можно использовать либо имя существующего в сети компьютера, либо имя описанное как существующее в SQL Server Client Network Utility. В случае использования копии SQL Server на том же самом компьютере, с которого выполняется подключение, можно ввести стандартное имя "(local)", как имя сервера Windows NT.

StatsLogFile: полное имя файла, включая маршрут, который будет использоваться для ведения журнала операций драйвера SQL Server Wire Protocol.

StatsLog_On: StatsLog_On={yes | no}. Включает и выключает сбор статистики операций, выполняемых драйвером SQL Server Wire Protocol. Если значение параметра установлено равным логической единице (Yes), то сведения об операциях, выполненных драйвером SQL Server Wire Protocol заносятся в журнал. Если значение параметра установлено равным логическому нулю (No), то сведения об операциях драйвера SQL Server Wire Protocol, выполненных с использованием этого подключения, не сохраняются.

Trusted_Connection: Trusted_Connection={yes | no}. Определяет сведения, которые будут использоваться драйвером SQL Server Wire Protocol, для подтверждения полномочий пользователя при входе в систему. Если значение параметра установлено равным логической единице (Yes), то драйвер SQL Server Wire Protocol будет работать в режиме проверки прав

пользователей при входе в систему Windows NT Authentication Mode. Дополнительно можно задать параметры UID и PWD. Если значение параметра установлено равным логическому нулю (No), то для подтверждения прав доступа к данным пользователем драйвер SQL Server Wire Protocol будет использовать имя и пароль пользователя базы данных SQL Server. Параметры UID и PWD нужно задавать обязательно.

UID: полноценная учетная запись входа в систему SQL-сервер. Пользуясь штатными средствами подтверждения входа в систему Windows NT, параметр UID можно не задавать.

WSID: идентификатор рабочей станции. Обычно, сетевое имя компьютера, на котором выполняется приложение. Использовать этот параметр не обязательно. Если этот параметр задается, то его значение будет добавлено в столбец "hostname" таблицы master.dbo.sysprocesses. Это значение будет возвращаться по вызову функций sp_who и Transact-SQL HOST_NAME.

В следующем примере выполняется соединение с базой данных SQL Server, которая называется MIPRO-SQL2014:

```
Server_connect("ODBC", "DRIVER={SQL Server Native Client 11.0};  
DATABASE=QADB;Server=MIPRO-SQL2014;UID=myusername;PWD=mypassword;QuotedID=YES;Trusted_Connection=No;  
Network=DBMSSOCN;Address=MIPRO-SQL2014,1433")
```

Соединение с Oracle Spatial

Oracle Spatial — это новый вариант программного обеспечения для баз данных с пространственными объектами от корпорации Oracle. Хотя присутствует определенное сходство с предыдущим ПО Oracle SDO, но различия существенны. Oracle Spatial обрабатывает данные Oracle SDO с помощью реляционной схемы. Однако, MapInfo Pro не поддерживает реляционную схему Oracle SDO через OCI. MapInfo Pro не поддерживает одновременные соединения с Oracle через OCI и с другими базами данных через ODBC. MapInfo Pro не поддерживает загрузку геометрических таблиц Oracle Spatial через ODBC, используя текущий драйвер ODBC фирмы Intersolv. Отсутствует компонент DSN.

Пример строки соединения для доступа к серверу Oracle Spatial с именем OraMIPRO, когда используется TCP/IP:

```
"SRVR=OraMIPRO;UID=myusername;PWD=mypassword"
```

attribute-код	Описание
LogonID (UID)	Имя пользователя (logon ID) которое приложение использует для связи с Вашей базой данных Oracle. Этот идентификатор требуется только в том случае, если включены средства обеспечения безопасности и защиты базы данных. Для получения необходимого идентификатора следует обратиться к администратору базы данных.
Пароль (PWD)	Ваш пароль. Его тоже выдает системный администратор.

attribute-код	Описание
ИмяСѐлвера (SRVR)	Имя сервера Oracle.

В следующем примере выполняется соединение с сѐрвером базы данных Oracle Spatial, который называется MiProPostGIS21:

```
server_connect ("ORAINET", "SRVR=OraMIPRO;UID=myusername;PWD=mypassword")
```

Соединение с POSTgreSQL

Пример строки подключения для PostgreSQL:

```
DSN=AccountingDepartment;UID=myusername;PWD=mypassword
```

Следующие параметры могут использоваться для драйвера PostgreSQL:

Определение	Ключевое слово	Сокращение
описание источника данных	Описание	не задано
имя сервера	Servename	не задано
проверяемый постмастером порт	Port	не задано
Имя пользователя	Username	не задано
Пароль	Пароль	не задано
флаг отладки	Debug	B2
максимум счетчика выборки	Fetch	A7
размер буфера сокета	Socket	A8
защита от записи базы данных	ReadOnly	A0
передача записей журнала в архив	CommLog	B3
протокол архива PostgreSQL	Protocol	A1

Определение	Ключевое слово	Сокращение
оптимизатор архива	Optimizer	B4
оптимизация ключей запросов	Ksqo	B5
передать в архив при соединении	ConnSettings	A6
использовать уникальную индексацию	UniqueIndex	не задано
размеры произвольных результатов	UnknownSizes	A9
отмена по FreeStmt	CancelAsFreeStmt	C1
использовать курсоры Declare/Fetch	UseDeclareFetch	B6
текст LongVarchar	TextAsLongVarchar	B7
произвольные данные LongVarchar	UnknownsAsLongVarchar	B8
логические в виде Char	BoolsAsChar	B9
максимальный размер Varchar	MaxVarcharSize	B0
максимальный размер LongVarchar	MaxLongVarcharSize	B1
замена уникального индекса по OID	FakeOidIndex	A2
включить OID в SQLColumns	ShowOidColumn	A3
версии записей	RowVersioning	A4
показать системные таблицы	ShowSystemTables	A5
выражения для разбора	Parse	C0
префиксы системных таблиц	ExtraSysTablePrefixes	C2
запретить предсказанные переходы	DisallowPremature	C3

Определение	Ключевое слово	Сокращение
обновляемые курсоры	UpdatableCursors	C4
преобразования LF <-> CR/LF	LFConversion	C5
логическая единица как минус единица (-1)	TruelsMinus1	C6
тип данных отчета о колонках int8	BI	не задано
байт как as LongVarBinary	ByteaAsLongVarBinary	C7
использовать подготовку на сервере	UseServerSidePrepare	C8
идентификатор нижнего регистра	LowerCaseIdentifier	C9
SSL-режим	SSLmode	CA
дополнительно	AB	не задано
сократить (простой вариант рекомендованных значений)	CX	не задано

В следующем примере выполняется соединение с сервером Postgres с PostGIS, который называется MiProPostGIS21:

```
Server_connect("ODBC", "DRIVER={PostgreSQL UNICODE (x64)};DATABASE=QADB;  
Server=MiProPostGIS21;UID=myusername;PWD=mypassword;Port=5432")
```

См. также:

[оператор Server Disconnect](#)

Функция Server_ConnectInfo()

Назначение

Этот оператор присваивает геоинформацию таблице MapInfo, связанной с таблицей в удаленной базе данных. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_ConnectInfo( ConnectionNo, Attr )
```

ConnectionNo — целое число, номер соединения с сервером данных, который возвращает функция **Server_Connect()**.

Attr — код, определяющий характер возвращаемой информации.

Возвращаемая величина

Строка

Описание

Функция **Server_ConnectInfo** возвращает информацию о соединении с базой данных. Первый параметр указывает номер соединения (начиная с 1). Второй параметр выбирает тип возвращаемой информации, как показано в следующей таблице:

Attr-код	ID	Server_ConnectInfo() возвращает:
SRV_CONNECT_INFO_DRIVER_NAME	1	Строку, показывающую имя драйвера, соответствующего данному соединению.
SRV_CONNECT_INFO_DB_NAME	2	Строковый результат, возвращающий тип базы данных, с которой устанавливается соединение. Возможные возвращаемые значения: • OSGeo.SQLite • Microsoft SQL Server • PostgreSQL PGIS • Oracle_Spatial
SRV_CONNECT_INFO_SQL_USER_ID	3	Строку, возвращающую идентификатор пользователя SQL.
SRV_CONNECT_INFO_DS_NAME	4	Строку, возвращающую имя источника данных.
SRV_CONNECT_INFO_QUOTE_CHAR	5	Строку, возвращающую кавычки.
SRV_CONNECT_INFO_DATABASE_NAME	6	Строковый результат, возвращающий имя базы данных, с которой устанавливается соединение.

Attr-код	ID	Server_ConnectInfo() возвращает:
SRV_CONNECT_INFO_SERVER_NAME	7	Строковый результат, в котором возвращается имя сервера, обеспечивающего работу базы данных, с которой устанавливается соединение.

Пример

```
Dim dbname as String
Dim hdbc As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
dbname=Server_ConnectInfo(hdbc, SRV_CONNECT_INFO_DB_NAME)
Print dbname
```

См. также:

[функция Server_Connect\(\)](#)

оператор Server Create Map

Назначение

Этот оператор присваивает геоинформацию таблице MapInfo, связанной с таблицей в удаленной базе данных. Модификация таблицы (например, путем добавления столбцов с пространственной информацией в таблицу) при этом не осуществляется. В данной версии мы добавили возможность размещать текст примечаний (annotation) Oracle 11g на карты MapInfo. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Оператор Server <Connection Number> Create Map используется для регистрации в каталоге Карт MAP_CATALOG средствами диалога **Присоединить геоинформацию**. Для создания текста подписей в связанных таблицах (linked_tables), вводится новый тип объекта, Text. Теперь оператор выглядит так:

Синтаксис

```
Server ConnectionNumber Create Map
For linked_table
Type { MICODE columnname | XYINDEX (xcolumnname, ycolumnname) | {
  SPATIALWARE | ORA_SP | SQLSERVERSPATIAL {
    GEOMETRY | GEOGRAPHY } | POSTGIS }
columnname}
[ CoordSys... ]
```



```
[ MapBounds { Data | Coordsys | Values ( x1, y1 ) ( x2, y2 ) } ]
[ ObjectType { Point | Line | Region | Text | ALL } ]
[ Symbol(...) ]
[ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
[ Style Type style_number [ Column column_name ] ]
```

Text — этот параметр позволяет превращать текст Oracle Spatial annotation в текстовые объекты. Режим ALL не включает этот текстовый объект.

connectionNumber — целое значение, номер соединения с базой данных;

linked_table — имя открытой связанной ODBC-таблицы

columnname — имя столбца, содержащего координаты специфицированного типа

xcolumnname — имя колонки X со значениями долгот координат

ycolumnname — имя колонки Y со значениями широт координат

x1, y1, x2, y2 — определяют границы системы координат.

CoordSys — предложение, задающее координатную систему и проекцию.

MapBounds — предложение, позволяющее задавать границы таблицы в каталоге карт MapCatalog. По умолчанию используется значение **Data**, вычисляющее все границы слоя. (Для программ, скомпилированных в версиях ранее 7.5, по умолчанию используется **Coordsys**).

Coordsys задает границы координатной системы. Его нежелательно использовать, поскольку слой может показаться пустым, если границы Coordsys значительно больше границ существующих данных. Большинство пользователей отдаляют карту слишком быстро и не успевают заметить свои данные.

Values — позволяет задавать собственные границы для MapCatalog.

ObjectType — предложение, задающее тип объекта в таблице: точки, линии, регионы или все объекты. Если предложение **ObjectType** не задано, по умолчанию используется тип **Точка**. Режим *Text* позволяет помещать аннотаций Oracle Spatial annotation в текстовый объект. Режим *ALL* не включает текст.

Symbol — корректное [предложение Symbol](#), задающее символ для точечного объекта.

Linestyle Pen — корректное [предложение Pen](#), задающее стиль линии для линейных объектов.

Regionstyle Pen — корректное [предложение Pen](#), а **Brush** — корректное [предложение Brush](#), которое задают стиль линии и штриховки для отрисовки площадного объекта.

StyleType — устанавливает символы для отдельных строк. *Style_number* — значение либо 0, либо 1. Ключевое слово **Column** и аргумент должны присутствовать, если параметр *style_number* задан 1 (единица). Когда значение *style_number* равно нулю, то **Column** игнорируется и создаются колонки соответствия в каталоге карт MapCatalog.

Описание

Оператор **Server Create Map** присваивает геоинформацию таблице MapInfo, связанной с таблицей в удаленной базе данных. Присоединяя геоинформацию к таблице SpatialWare, Oracle Spatial, SQL Server Spatial или PostGIS, можно использовать точки, линии и полигоны. Для всех других таблиц можно отображать на карте только точки. Любая таблица MapInfo Pro может быть отображена в окне списка, но только таблица с геоинформацией может иметь присоединенные графические объекты, и только такие таблицы могут быть отображены в окнах карты MapInfo.

Примечание: Если сервер это Oracle9i и система координат определена как Долгота/Широта без определения датума, то по умолчанию будет использоваться стандартный датум World Geodetic System 1984(WGS 84). Такое поведение позволяет использовать **оператор Server Create Table** и программу Easyloader.

Типы атрибутов	Описание
ORA_SP <i>columnname</i>	OracleSpatial
SPATIALWARE	SpatialWare для SQL Server
MICODE	Индекс XYINDEX
SQLSERVERSPATIAL GEOMETRY	Геометрия SQL Server Spatial
SQLSERVERSPATIAL GEOGRAPHY	География SQL Server Spatial
POSTGIS	PostGIS для PostgreSQL

Примеры

```
Sub Main
  Dim ConnNum As Integer
  ConnNum = Server_Connect("ODBC",
    "DSN=SQLServer;DB=QADB;UID=mipro;PWD=mipro")
  Server ConnNum Create Map For "Cities"
  Type SPATIALWARE
  CoordSys Earth Projection 1, 0
  ObjectType All
  ObjectType Point
  Symbol (35,0,12)
  Server ConnNum Disconnect
End Sub
```

В следующем примере этот оператор применяется к таблице ANNOTEXT_TABLE:

```
Server 1 Create Map For ""MIPRO"". "ANNOTEXT_TABLE"
Type ORA_SP "TEXTOBJ"
CoordSys Earth Projection 12, 62, "m", 0 Bounds
(-34012036.7393, -8625248.51472) (34012036.7393, 8625248.51472)
mapbounds data
ObjectType Text
```

См. также:

[оператор Server Link Table](#), [оператор Unlink](#)

Server Create Style

Для таблицы с картой можно назначить индивидуальные стили оформления каждого объекта. В этом варианте синтаксиса возвращается состояние обработки – успешное выполнение или ошибка. Следующий синтаксис используется и в [операторе Set Map](#).

```
Server ConnectionNumber Create Map linked_table...
[ Style Type style_number [ Column column_name ] ]
```

ConnectionNumber – целое значение, номер соединения с базой данных;

linked_table – имя открытой связанной ODBC-таблицы

columnname – имя столбца, содержащего координаты специфицированного типа

StyleType – устанавливает символы для отдельных строк. *style_number* – значение равно 0 или 1. Когда Type равно 1 (единице), то ключевое слово **Column** и его аргумент должны быть представлены. Когда значение *style_number* равно нулю, то **Column** игнорируется и создаются колонки соответствия в MapCatalog.

Описание

Чтобы оператор сработал правильно, каталог карт должен иметь структуру, поддерживающую стили. Он должен содержать колонки RENDITIONTYPE, RENDITIONCOLUMN и RENDITIONTABLE. Команда не будет успешно выполнена, если колонки стилей не являются текстовыми. Оператор SQL сам выдаст ошибку, если попытается установить строковую величину в колонку с различными типами данных.

Пример

```
Server 2 Create Map For "qadb:sample.arc"
Type MICODE "mi_sql_micode" ("mi_sql_x", "mi_sql_y")
CoordSys Earth Projection 1, 0 ObjectType Point Symbol (35,0,12) Style
Type 1 Column "mi_style"
```

См. также:

функция **Server_Connect()**, оператор **Set Map**.

Оператор Server Create Table

Назначение

Создает новую таблицу в указанной удаленной базе данных. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Create Table TableName
( ColumnName ColumnType [ , ... ] )
[ KeyColumn ColumnName ]
[ ObjectColumn ColumnName [ Type SQLServerSpatial
{ Geometry | Geography } ] ]
[ StyleColumn ColumnName ]
[ CoordSys... ]
```

ConnectionNumber – целое значение, номер соединения с базой данных;

Tablename – строковая переменная, определяющая имя, которое нужно присвоить таблице базы данных;

ColumnName – имя создаваемой колонки. Имя колонки может быть длиной до 31 символа, может содержать буквы, числа и символ подчеркивания(_). Имя колонки не может начинаться с цифры.

ColumnType – тип данных, ассоциированных с колонкой.

Описание

Оператор **Server Create Table** создает новую пустую таблицу в базе данных с числом колонок до 250.

Tablename – строковая переменная имени таблицы базы данных; имя таблицы может включать в себя имя схемы, которой принадлежит таблица. Если имени схемы не задано, то таблица принадлежит стандартной схеме. Пользователь должен задать правильное имя схемы, должен знать имя учетной записи и обладать правами доступа к указанной схеме. Это касается только SQL Server 2005.

Длина имени таблицы *TableName* изменяется в зависимости от типа баз данных. Мы рекомендуем использовать 14 или менее символов для имени таблицы, чтобы быть застрахованным от ошибок при работе с любой базой данных. Таким образом, пусть максимальная длина имени таблицы будет 14 символов.

ColumnType использует тот же тип данных, что и **оператор Create Table**. Некоторые типы данных могут быть конвертированы в те, которые поддерживаются используемой базой данных.

KeyColumn – предложение, определяющее ключевую колонку таблицы. Если предложение определено, то по этой колонке будет создан уникальный индекс. Мы рекомендуем использовать это предложение, так как оно позволяет MapInfo Pro открывать таблицу при прямом доступе к базе данных.

Дополнительное предложение **ObjectColumn** позволит создать таблицу с колонкой пространственной геометрии/объектами. Если предложение определено, то пространственный индекс также будет создан для этой колонки. Таким образом, если сервер не имеет возможности обработать пространственную геометрию/объекты, то таблицы создана не будет. Если сервер это SQL Server со SpatialWare, то таблица будет настроена на пространственную геометрию/объекты с момента создания. Если сервер это Oracle Spatial, то пространственные метаданные обновятся в момент создания таблицы.

Если используется **Server Create Table** и предложение **ObjectColumn** пропущено в операторе, также требуется использовать **оператор Server Create Map** для того, чтобы открыть таблицу в MapInfo Pro.

StyleColumn – предложение, определяющее колонку Per Row Style, которая позволяет использовать различные стили объектов для каждой записи таблицы.

Необязательное **предложение CoordSys** задаёт систему координат и проекцию. Это предложение требуется использовать только тогда, когда создаётся таблица с пространственными объектами/геометрией в базе данных Oracle Spatial (Oracle9i или более новая версия с пространственной компонентой). Если сервер это Oracle9i и система координат определена как Долгота/Широта без определения датума, то по умолчанию будет использоваться стандартный датум World Geodetic System 1984 (WGS 84). Система координат должна быть такой же, как и система, которую задает **оператор Server Create Map**. Для других баз данных это предложение не влияет на создание таблицы.

Поддерживаются следующие базы данных: Oracle, SQL Server, PostGIS и Microsoft Access. Таким образом, для создания таблицы с колонкой пространственной геометрии/объектами, SpatialWare требуется для SQL Server, а для Oracle требуются пространственные настройки.

Замечания о типах данных Дата\Время, Время и Дата

Специальных синтаксических изменений при использовании нового типа данных нет. Существует следующие ограничения некоторых типов данных:

Типы данных **Время** и **Дата/Время** очень полезны, но перед их использованием требуется продумать структуру базы данных. В большинство баз данных не включены соответствующие типы. Раньше мы обеспечивали работу только с типами данных Date (Дата). Даже тип Date преобразовывался в тип данных сервера, если сервер не поддерживал тип данных Date. В MapBasic версии 9.0 и выше, этот оператор поддерживает только те типы, которые поддерживает сервер. Поэтому, тип Time запрещен в этом операторе для Oracle, SQL Server и Access, а тип Date запрещен для SQL Server и Access, а тип Date – для SQL Server и Access.

Если необходимо хранить в таблице базы данных сведения о времени в отдельной колонке, то эти "неподдерживаемые" типы следует заменять типом данных DateTime.

Примечание: Для Microsoft SQL Server и Access и версий MapInfo Pro старше 9.0 преобразование проводится в фоновом режиме. В версии 9.0 пользователи должны выбирать DATETIME вместо DATE, иначе операция не сработает.

Примеры

Следующие примеры показывают, как создать таблицу с именем ALLTYPES, которая содержит семь колонок, охватывающих каждый из типов данных, поддерживаемых MapInfo Pro, плюс три колонки Key, SpatialObject и Style. Всего колонок должно быть десять.

Для SQL Server с SpatialWare:

```
dim hodbс as integer
hodbс = server_connect("ODBC", "dlg=1")
Server hodbс Create Table ALLTYPES( Field1 char(10),Field2 integer,Field3
SmallInt,Field4 float,Field5 decimal(10,4),Field6 date,Field7 logical)
KeyColumn SW_MEMBER
ObjectColumn SW_GEOMETRY
StyleColumn MI_STYLE
```

Для Oracle Spatial:

```
dim hodbс as integer
hodbс = server_connect("ORAINET", "SRVR=cygnus;UID=mipro;PWD=mipro")
Server hodbс Create Table ALLTYPES( Field1 char(10),Field2 integer,Field3
SmallInt,Field4 float,Field5 decimal(10,4),Field6 date,Field7logical)
KeyColumn MI_PRINX
ObjectColumn GEOLOC
StyleColumn MI_STYLE
Coordsys Earth Projection 1, 0
```

См. также:

[оператор Create Map](#), [оператор Server Create Map](#), [оператор Server Link Table](#) , [оператор Unlink](#)

Оператор Server Create Workspace

Назначение

Создаёт новую рабочую область в базе данных (Oracle 9i или более новой). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Create
Workspace WorkspaceName
[ Description Description ]
[ Parent ParentWorkspaceName ]
```

ConnectionNumber – целое число, идентификатор соединения.

WorkspaceName – имя рабочей области. Имя чувствительно к регистру и должно быть уникальным. Имя рабочей области не должно превышать 30 символов.

Description – строка, описывающая рабочий набор.

ParentWorkspaceName – имя рабочего набора, являющегося порождающим по отношению к новому рабочему набору *WorkspaceName*. По умолчанию, когда создаётся рабочий набор, то он создаётся из самого последнего или рабочего набора базы данных прямого (LIVE) доступа.

Описание

Этот оператор применяется только к Oracle9i или старше. Новый рабочий набор *WorkspaceName* является порожденным по отношению к порождающему рабочему набору *ParentWorkspaceName* или LIVE, если порождающий рабочий набор не указан.

Более подробную информацию смотрите в *Oracle9i Application Developer's Guide - Workspace Manager*.

Примеры

В следующем примере создается рабочий набор, названный MIUSER.

```
Dim hdbc As Integer
hdbc = Server_Connect("ORAINET", "SRVR=TROINY;UID=MIUSER;PWD=MIUSER")
Server hdbc Create
Workspace "MIUSER"
Description "MIUser private workspace"
```

В следующем примере создаётся порождённый рабочий набор из порождающего рабочего набора MIUSER в базе данных.

```
Dim hdbc As Integer
hdbc = Server_Connect("ORAINET", "SRVR=TROINY;UID=MIUSER;PWD=MIUSER")
Server hdbc Create Workspace "MBPROG" Description "MapBasic project"
Parent "MIUSER"
```

См. также:

оператор [Server Remove Workspace](#), оператор [Server Versioning](#)

Оператор Server Disconnect

Назначение

Прекращает связь, которую установила с удаленным сервером данных [функция Server_Connect\(\)](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Disconnect
```

ConnectionNumber – целое число, идентификатор соединения.

Описание

Оператор **Server Disconnect** отключает приложение от базы данных. Все ресурсы, выделенные для указанного соединения, возвращаются в распоряжение системы.

Пример

```
Dim hdbc As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
Server hdbc Disconnect
```

См. также:

[функция Server_Connect\(\)](#)

функция Server_DriverInfo()

Назначение

Выдает информацию об установленном программном обеспечении и источниках данных. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_DriverInfo( DriverNo, Attr )
```


DriverNo – целое значение, назначенное приложением MapInfo Pro модулю интерфейса удаленного доступа при запуске MapInfo Pro.

Attr – код, определяющий характер возвращаемой информации.

Возвращаемая величина

Строка – String

Описание

Функция **Server_DriverInfo()** возвращает информацию об источниках данных. Первый параметр выбирает модуль интерфейса удаленного доступа (начиная с 1). Общее число установленных интерфейсных модулей может вернуть **функция Server_NumDrivers()**. Вторым параметром выбирается тип возвращаемой информации, как показано в следующей таблице:

Attr-код	ID	Функция Server_DriverInfo() возвращает:
SRV_DRV_INFO_NAME	1	Строку, показывающую имя модуля. ODBC указывает источник данных ODBC. ORAINET указывает соединение Oracle Spatial.
SRV_DRV_INFO_NAME_LIST	2	Строку – список имен всех установленных интерфейсных модулей, разделенных точками с запятой. А именно – ODBC, ORAINET. Параметр <i>DriverNo</i> игнорируется.
SRV_DRV_DATA_SOURCE	3	Строку – имена источников данных, поддерживаемых указанным модулем интерфейса. Последовательные вызовы функции последовательно выбирают имена источников. После выборки последнего имени для данного модуля функция возвратит пустую строку. Следующий вызов функции для того же модуля установит список на начало и возвратит первое имя в списке.

Пример

```
Dim dlg_string, source As String
dlg_string = Server_DriverInfo(0, SRV_DRV_INFO_NAME_LIST)
source = Server_DriverInfo(1, SRV_DRV_DATA_SOURCE)
While source <> ""
    Print "Available sources on toolkit " +
    Server_DriverInfo(1, SRV_DRV_INFO_NAME) + ": " +
    source
    source = Server_DriverInfo(1,
    SRV_DRV_DATA_SOURCE)
Wend
```

См. также:

функция [Server_NumDrivers\(\)](#)

Функция [Server_EOT\(\)](#)

Назначение

Определяет, был ли достигнут конец результирующего набора в процессе последовательной выборки записей, которую проводит [оператор Server Fetch](#). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_EOT( StatementNumber )
```

StatementNumber — целое число, номер проверяемого оператора [Server Fetch](#).

Возвращаемая величина

Логическое

Описание

Функция **Server_EOT()** возвращает TRUE или FALSE, в зависимости от результатов предыдущего оператора [Server Fetch](#). Значение TRUE возвращается, как при попытке выбрать предыдущую запись сразу же после выборки первой записи набора, так и в случае выборки следующей записи после последней записи набора.

Пример

```
Dim hdbc, hstmt As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "Select * from ADDR")
Server hstmt Fetch FIRST
While Not Server_EOT(hstmt)
  ' Processing for each row of data ...
  Server hstmt Fetch Next
Wend
```

См. также:

[оператор Server Fetch](#)

Функция `Server_Execute ( )`

Назначение

Посылает SQL-строку для исполнения на удаленный сервер данных. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_Execute( ConnectionNumber, server_string )
```

ConnectionNumber — целое число, идентификатор соединения.

server_string — строка, представляющая любой корректный SQL-оператор, поддерживаемый сервером, с которым установлено соединение. Информацию о корректных SQL-операторах смотрите в руководстве по языку SQL для СУБД на сервере.

Возвращаемая величина:

Целое – Integer

Описание

Функция **Server_Execute ()** пересылает SQL-строку, заданную параметром *server_string* и представляющую SQL-оператор, через соединение с сервером, указанное параметром *ConnectionNumber*. Любой корректный SQL-оператор, поддерживаемый активным сервером, является допустимым значением параметра *server_string*. Информацию о корректных SQL-операторах смотрите в руководстве по языку SQL для СУБД на сервере.

Эта функция возвращает номер-указатель (handle) оператора, используемый для ассоциации (через параметр StatementNumber) последующих SQL-обращений (таких как **оператор Server Fetch** и **оператор Server Close** с конкретным SQL-оператором.

Необходимо выполнить оператор Server Close для каждого вызова функции **Server_Execute ()**. Для операторов типа Select — после выборки требуемых данных. При этом на удаленном сервере данных будет закрыт курсор и освобожден результирующий набор. В противном случае Вы можете превысить лимит на число открытых курсоров, и дальнейшие обращения к базе данных исполняться не будут. Не все серверы баз данных поддерживают курсоры с прямой и обратной последовательностью выполнения. Для других SQL-операторов используйте оператор Server Close сразу после вызова функции **Server_Execute ()**.

```
Dim hdbc, hstmt As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
```

```
hstmt = Server_Execute(hdbc, "Select * from ADDR")
Server hstmt Close
```

Пример

```
Dim hdbc, hstmt As Integer
hdbc = Server_Connect("ODBC", DSN=ORACLE7;DLG=1")
hstmt = Server_Execute (hdbc,
  "CREATE TABLE NAME_TABLE (NAME CHAR (20))")
Server hstmt Close
hstmt = Server_Execute (hdbc,
  "INSERT INTO NAME_TABLE VALUES ('Steve')")
Server hstmt Close
hstmt = Server_Execute ( hdbc,
  "UPDATE NAME_TABLE SET name = 'Tim' ")
Server hstmt Close
Server hdbc Disconnect
```

См. также:

[оператор Server Close](#), [оператор Server Fetch](#)

Оператор Server Fetch

Назначение

Осуществляет выборку записей результирующего набора с удаленного сервера данных. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server StatementNumber Fetch [ NEXT | PREV | FIRST | LAST | [ REC ] recno
]
```

или

```
Server StatementNumber Fetch INTO Table [ FILE path ]
```

StatementNumber — целое значение, номер SQL-оператора.

recno — целое число, номер выбираемой записи.

path — маршрут по диску к существующей таблице.

Описание

Оператор **Server Fetch** извлекает записи результирующего набора (заданного значением *StatementNumber* SQL-оператора, создавшего набор) из сервера данных. Для построчной выборки данных они помещаются в локальную область хранения, и могут быть связаны с переменными, если выполнить **оператор Server Bind Column**. Для выборки данных по столбцам применяется функция `Server_ColumnInfo(SRV_COL_INFO_VALUE)`. Можно также в одной операции выбрать полный результирующий набор в таблицу MapInfo, используя предложение **INTO Table**.

Выполнение операторов **Server Fetch** и **Server Fetch Into** прерывается с установкой кода ошибки `ERR() = ERR_SRV_ESC` при нажатии пользователем клавиши ESC, что позволяет MapBasic-приложению использовать команды **Server Fetch** для обработки этого события.

По исполнении оператора **Server Fetch Into**, таблица MapInfo фиксируется так, что для нее не остается незавершенных транзакций. Все символьные поля, превышающие 254 байта, усекаются; все двоичные (неструктурированные) поля загружаются в таблицу как шестнадцатеричные символьные строки двойной длины. Имена столбцов в загруженной таблице будут использовать псевдонимы столбцов, если в запросе задавались псевдонимы.

Обработка null-значений

Если был исполнен **оператор Select** с последующей выборкой записи, включающей столбец таблицы, который содержит пустое (null) значение, то происходит следующее. Поскольку в MapInfo не поддерживается концепция пустых (null) значений в таблице или переменной, используется значение по умолчанию в рамках домена для соответствующего типа данных — значение MapBasic-переменной, декларированной в инструкции Dim, но не инициализированной. Однако при этом обеспечивается индикация возврата пустого значения.

Для связанных переменных (см. **оператор Server Bind Column**) могут быть заданы переменные состояния, значения которых будут указывать на возврат пустого значения при выборке. Для столбцов, не связанных с переменными, функция `Server_ColumnInfo()` с *атрибутом* `SRV_COL_INFO_STATUS` будет возвращать статус столбца, информирующий о возможно пустом значении.

См. в *Руководстве пользователя MapBasic* информацию о том, как MapInfo Pro интерпретирует типы данных.

Ошибки

Команда **Server n Fetch Into table** будет генерировать ошибку для любых неудачных попыток вставки записей в локальную таблицу MapInfo. Команды **Server n Fetch [Next | Prev | recno]** вызывают сообщения об ошибках, если необходимая запись не доступна.

Пример 1

```
' An example of Server Fetch downloading into a MapInfo table
Dim hdbc, hstmt As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "Select * from emp")
Server hstmt Fetch Into "MyEmp"
Server hstmt Close
```

Пример 2

```
' An example of Server Fetch using bound variables
Dim hdbc, hstmt As Integer
dim NameVar, AddrVar as String
dim NameStatus, AddrStatus as Integer
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "Select Name, Addr from emp")
Server hstmt Bind Column 1 to NameVar, NameStatus
Server hstmt Bind Column 2 to AddrVar, AddrStatus
Server hstmt Fetch Next
While Not Server_Eot(hstmt)
  Print "Name = " + NameVar + "; Address = " + AddrVar
  Server hstmt Fetch Next
Wend
```

См. также:

[функция Server_ColumnInfo\(\)](#)

Функция Server_GetODBCConn()

Назначение

Возвращает целое значение, содержащее указатель связи ODBC, ассоциированной со связью с сервером данных. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_GetODBCConn( ConnectionNumber )
```

ConnectionNumber – целое число, номер соединения с сервером данных, который возвращает [функция Server_Connect\(\)](#).

Описание

Эта функция возвращает целое значение, содержащее указатель связи ODBC, ассоциированной со связью с сервером данных. Это позволяет Вам любую функцию в ODBC DLL, чтобы расширить функциональные возможности, посредством использования MapBasic операторов типа **Server**.

Пример

```

'* Find the identity of the Connected database
DECLARE FUNCTION SQLGetInfo LIB "ODBC32.DLL" (BYVAL odbchdbc AS INTEGER,
BYVAL infoflag AS INTEGER, val AS STRING, BYVAL len AS INTEGER, outlen
AS
INTEGER) AS INTEGER
Dim rc, outlen, hdbc, odbchdbc AS INTEGER
Dim DBName AS STRING
' Connect to a database
hdbc = Server_Connect("ODBC", "DLG=1")
odbchdbc = Server_GetodbcHConn(hdbc) ' get ODBC connection handle
' Get database name from ODBC
DBName = STRING$(33, "0") ' Initialize output buffer
rc = SQLGetInfo(odbchdbc, 17 , DBName, 40, outlen) ' get ODBC Database
Name
' Display results (database name)
if rc <> 0 THEN
  Note "SQLGetInfo Error rc=" + rc + ", outlen=" + outlen
else
  Note "Connected to Database: " + DBName
end if

```

См. также:

[Функция Server_GetODBCStmt\(\)](#)

Функция Server_GetODBCStmt()

Назначение

Возвращает указатель оператора ODBC, ассоциированный с MapBasic-оператором типа **Server**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_GetODBCStmt( StatementNumber )
```

StatementNumber — целое число, номер результирующего набора, выполненного SQL-оператором, который возвращает [функция Server_Execute\(\)](#).

Описание

Эта функция возвращает указатель оператора ODBC, ассоциированный с MapBasic-оператором типа **Server**. Это позволяет вызывать любую функцию ODBC, для расширения функциональных возможностей посредством использования MapBasic-операторов типа **Server**.

Пример

```
' Find the Number of rows affected by an Update
Dim rc, outlen, hdbc, hstmt, odbchstmt AS INTEGER
Dim RowsUpdated AS INTEGER
' Find the Number of rows affected by an Update
DECLARE FUNCTION SQLRowCount LIB "ODBC32.DLL" (BYVAL odbchstmt AS INTEGER,
rowcnt AS INTEGER) AS INTEGER
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "UPDATE TIML.CUSTOMER SET STATE='NY' WHERE
STATE='NY'")
odbchstmt = Server_GetodbcHStmt(hstmt)
rc = SQLRowCount(odbchstmt, RowsUpdated)
Note "Updated " + RowsUpdated + " New customers to Tier 1"
```

См. также:

[функция Server_GetODBCConn\(\)](#)

Оператор Server Link Table

Назначение

Создаёт связанную таблицу MapInfo (ТАВ-файл). Этот оператор устанавливает соединение с сервером базы данных и связанной таблицей. Связанная таблица определяет обновляемые на сервере данные, метаданные сохраняются в ТАВ-файле. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Server Link Table
  SQLQuery
Using ConnectionString
```



```
[ Symbol... ] [ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
Into TableName
Toolkit Toolkitname
[ File filespec ]
[ Charset char_set ]
[ Type { NATIVE | NATIVEX } ]
[ ReadOnly ]
[ Autokey { Off | On } ]
```

Синтаксис 2

```
Server ConnectionNumber Link Table
SQLQuery
Toolkit toolkitname
[ Symbol... ] [ Linestyle Pen(...) ]
[ Regionstyle Pen(...) Brush(...) ]
Into TableName
[ File filespec ]
[ Charset char_set ]
[ Type { NATIVE | NATIVEX } ]
[ ReadOnly ]
[ Autokey { Off | On } ]
```

ConnectionNumber – целое значение, номер существующего соединения;

SQLQuery – SQL-оператор запроса (на языке SQL с добавлением объектных ключевых слов), который генерирует результирующий набор. Таблица MapInfo связывается именно с этим результирующим набором.

ConnectionString – строка, используемая для подключения к серверу базы данных. См. [функция Server_Connect\(\)](#).

TableName – псевдоним создаваемой таблицы MapInfo.

fileSpec – имя табличного файла. Если этот параметр отсутствует, имя файла генерируется в текущем каталоге диска на базе псевдонима таблицы. Если параметр *fileSpec* задан, а табличный файл с указанным именем уже существует, то генерируется ошибка.

char_set – имя набора символов; см. раздел [Предложение CharSet](#).

Toolkitname – строка, указывающая тип соединения, ODBC или ORAINET.

Описание

Этот оператор создает связанную таблицу MapInfo на диске. Эта таблица открывается и к ней обращается запрос. Связанная таблица обрабатывается как обычная таблица MapInfo в большинстве случаев, с одним ограничением: MapBasic-[оператор Alter Table](#) не выполняется для связанных таблиц. Связанные таблицы не могут быть упакованы. В список диалога

Упаковка эти таблицы не включаются. Синтаксис **Server Link Table** используется для установки связи сервера базы данных и связанной таблицы. **Server ConnectionNumber Link Table** используются для связи таблицы с подключенным сервером базы данных. Связанные таблицы содержат информацию, позволяющую восстановить соединения и идентифицировать удаленные данные, в которые нужно внести изменения. Эта информация сохранена как метаданные в файле таблицы.

Отсутствие ключевого слова **ReadOnly** не означает возможности редактирования таблицы. Связанная таблица может быть запрещена для записи при следующих обстоятельствах: 1) результирующий набор разрешен только для чтения; 2) результирующий набор не содержит первичного ключа; 3) в результирующем наборе отсутствуют редактируемые столбцы; 4) указан режим **ReadOnly**. Если сервер – Oracle, **Autokey** задает режим использования авто-инкремента.

Если режим **Autokey** включен (**On**), таблица будет открыта в режиме автоинкремента. Если режим **Autokey** выключен (**Off**) или отсутствует, таблица будет открыта без применения режима автоинкремента.

Параметр **Symbol** задает стиль оформления точечных объектов;

Параметр **Brush** задает стиль заливки графических объектов.

Параметр **LineStyle** задает стиль оформления линейных объектов.

Параметр **Regionstyle** задает стили оформления границ и заливок площадных объектов.

Дополнительное предложение **CharSet** позволяет выполнять сохранение в определенном наборе символов. Если не используется, подразумевается набор символов исходной таблицы. Использование несовместимого набора символов может привести к тому, что символьные данные целевой таблицы будут преобразованы в символы подчёркивания (например, WindowsCyrillic в WindowsLatin1). Предпочтительнее преобразовывать в наборы символов UTF-8 или UTF-16. Используйте предложение **Charset** внутри предложения **Type**.

Дополнительное, необязательное предложение **Type** обозначает формат локальной таблицы: либо родной, нативный формат MapInfo (**NATIVE**), либо расширенный вариант формата MapInfo Extended (**NATIVEX**). Формат **NATIVEX** поддерживает таблицы, размер которых превышает 2GB, и содержащих наборы символов UTF-16 и UTF-16. Когда предложение **Type** не задано, по умолчанию используется набор символов *system_charset* (по умолчанию **NATIVE**). Когда задан расширенный формат **NATIVEX**, записанный TAB-файл будет иметь тип **LINKEDX**.

ReadOnly – задает использование таблицы только для чтения.

Синтаксис запросов SQL

Ключевое слово MapInfo **OBJECT** может быть использовано для установления связи между пространственными колонками и запросом SQL. MapInfo Pro преобразует ключевое слово **OBJECT** в соответствующие пространственные колонки. Таблица с именем SELECT*FROM

(*tablename*) всегда включает эти пространственные колонки, но если В хотите указать подмножество колонок, используйте ключевое слово **OBJECT**. Например:

```
SELECT col1, col2, OBJECT
FROM tablename
```

будут загружены две колонки и пространственный объект. Этот синтаксис будет работать в любой базе данных, поддерживаемой MapInfo Pro.

Пространственные запросы MapInfo Pro

MapInfo Pro поддерживает ключевое слово **WITHIN**, используемое в пространственных запросах. Оно используется для выбора пространственных объектов в таблице. Следующие два ключевых слова могут использоваться вместе с ключевым словом **WITHIN**:

- **CURRENT MAPPER**: вся прямоугольная область, отображаемая в текущем окне Карты.
- **SELECTION**: выбранная область в текущем окне Карты.

Синтаксис поиска всех строк, содержащих пространственные объекты, существующих в текущем окне карты, будет выглядеть следующим образом:

```
SELECT col1, col2, OBJECT FROM tablename
WHERE OBJECT WITHIN CURRENT_MAPPER
```

Этот синтаксис будет работать в любой базе данных, поддерживаемой MapInfo Pro. MapInfo Pro будет также выполнять пространственные запросы SQL, написанных на языке SQL для пространственных баз данных. Корректные значения для *toolkitname* смотрите в разделе о функции [Server_DriverInfo\(\)](#).

Примеры

```
Declare Sub Main
Sub Main
  Open table "C:\mapinfo\data\states.tab"
  Server Link Table "Select * from Statecap" Using
    "DSN=MS Access;DBQ=C:\MSOFFICE\ACCESS\DB1.mdb"
  Into test File "C:\tmp\test"
  Map From Test,States
End Sub 'Main
Declare Sub Main
Sub Main
  Dim ConnNum As Integer
  ConnNum = Server_Connect("ODBC", "DSN=SQS;PWD=sysmal;SRVR=seneca")
  Server ConnNum Link Table
    "Select * from CITY_1"
  Into temp
  Map From temp
```

```
Server ConnNum Disconnect
End Sub
```

В следующем примере показано создание связанных таблиц.

```
Dim hdbc As Integer
hdbc = Server_Connect("ORAINET", "SRVR=ONTARIO;UID=MIPRO;PWD=MIPRO")
Server hdbc link table
  "Select * From ""MIPRO"". ""SMALLINTEGER""
  Toolkit "ORAINET"
  Into SMALLINTEGER
  Autokey ON
  Map From SMALLINTEGER
```

См. также:

[оператор Close Table](#), [оператор Commit Table](#), [оператор Drop Table](#), [оператор Rollback](#), [оператор Save File](#), [оператор Server Refresh](#), [оператор Unlink](#)

Функция Server_NumCols()

Назначение

Возвращает число столбцов в результирующем наборе. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_NumCols( StatementNumber )
```

StatementNumber — целое значение, номер SQL-оператора.

Возвращаемая величина

Целое – Integer

Описание

Функция **Server_NumCols()** возвращает число столбцов в результирующем наборе, ссылка на который осуществляется по указателю *StatementNumber*.

Пример

```
Dim hdbc, hstmt As Integer
hdbc = Server_Connect("ODBC", "DLG=1")
hstmt = Server_Execute(hdbc, "Select Name, Addr from emp")
Print "Number of columns = " + Server_NumCols(hstmt)
```

См. также:[функция Server_ColumnInfo\(\)](#)

Функция Server_NumDrivers()

Назначение

Возвращает число интерфейсных модулей удаленного доступа к базам данных, установленных в данный момент для доступа из MapInfo. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server_NumDrivers()
```

Возвращаемая величина

Целое – Integer

Описание

Функция **Server_NumDrivers()** возвращает число модулей интерфейса удаленного доступа, через которые может осуществляться соединение с удаленным сервером данных, установленных для использования приложением MapInfo Pro.

Пример

```
Print "Number of drivers = " + Server_NumDrivers()
```

См. также:[функция Server_DriverInfo\(\)](#)

Оператор Server Refresh

Назначение

Осуществляет синхронизацию связанной таблицы MapInfo с данными в удаленной базе данных. Эта команда может выполняться только в отсутствие ожидающих запросов на редактирование связанной таблицы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server Refresh TableName Rec RecNumber
```

TableName — имя открытой связанной таблицы MapInfo.

RecNumber — номер записи единственной строки в таблице реляционной базы данных (РБД).

Описание

Если соединение с базой данных открыто, произойдет обновление данных. Если соединение с базой данных отсутствует, оно будет создано. Если потребуется какая-либо информация (например, пароль), она будет запрошена автоматически.

Синхронизация связанной таблицы включает следующие этапы:

1. Удаление всех записей и объектов из связанной таблицы (если она содержит записи) методом удаления и воссоздания табличного файла (не используя **оператор Delete**).
2. Если обработчик, указатель (handle) соединения сохранён вместе со структурой TABLE, используется этот указатель. В противном случае соединение с сервером базы данных производится с использованием строки подключения, сохраненной в метаданных связанной таблицы.
3. Выполняется преобразование SQL-запроса, сохраненного в метаданных, в запрос, специфичный для удаленной СУБД.
4. Этот SQL-запрос исполняется на удаленном сервере базы данных.
5. Таблица заполняется строками, выбранными курсором из удаленной СУБД. Эта операция может сопровождаться индикатором выполнения MapInfo Pro.
6. Курсор в удаленной СУБД закрывается.

Rec *RecNumber* обновляется запись в единственной строке таблицы реляционной базы данных (РБД). Таблица не может содержать ожидающих подтверждения исправлений в строке, записи, которая будет обновлена. Эта функция применяется как к связанным, так и к таблицам прямого доступа.

Пример

```
Server Refresh "City_1k"
```

См. также:

[оператор Commit Table](#), [оператор Server Link Table](#), [оператор Unlink](#)

Оператор Server Remove Workspace

Назначение

Сбрасывает все версии строк, связанные с рабочей областью, и удаляет рабочую область в базе данных (Oracle 9i или более новой). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Remove  
Workspace WorkspaceName
```

ConnectionNumber – целое число, идентификатор соединения.

WorkspaceName – имя Рабочей области. Имя зависит от регистра.

Описание

Этот оператор применяется только к Oracle9i или старше. Эта операция может быть применена только для тех рабочих областей, которые занимают самое нижнее место в иерархии. Не должно быть никаких других пользователей удаляемой рабочей области.

Примеры

Этот пример удаляет рабочую область MIUSER базы данных.

```
Dim hdbc As Integer  
hdbc = Server_Connect("ORAINET", "SRVR=TROJNY;UID=MIUSER;PWD=MIUSER")  
Server hdbc Remove Workspace "MIUSER"
```

См. также:

[оператор Server Create Workspace](#)

Оператор Server Rollback

Назначение

Выполняет откат транзакции на удаленном сервере данных. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Rollback
```

ConnectionNumber – целое число, идентификатор соединения.

Описание

Оператор **Server Rollback** восстанавливает исходное состояние базы данных, т.е. ликвидирует все изменения, внесенные в базу данных SQL-операторами с того момента, как был выполнен последний **оператор Server Begin Transaction** в данном соединении с сервером. Чтобы этот оператор сработал, необходимо иметь открытую транзакцию, которую инициирует **оператор Server Begin Transaction**.

Пример

```
hdbc = Server_Connect("ODBC", "DLG=1")
Server hdbc Begin Transaction

...

' All changes since begin_transaction are about ' to be discarded
Server hdbc Rollback
```

См. также:

оператор Server Begin Transaction, оператор Server Commit

Оператор Server Set Map

Назначение

Этот оператор позволяет менять стили объектов для изображаемой в виде карты таблицы ODBC. Это действие отображается в MapCatalog. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Set Map linked_table
[ ObjectType { Point | Line | Region | Text | ALL } ]
[ Symbol (...) ]
[ Linestyle Pen (...) ]
[ Regionstyle Pen (...) Brush (...) ]
```

ConnectionNumber – целое число, идентификатор соединения.

linked_Name – имя открытой связанной таблицы СУБД.

ObjectType – указывает тип объектов в таблице и позволяет определять объекты как регионы, линии или объекты. Подробнее смотрите в разделе об [операторе Server Create Map](#).

Symbol – корректное [предложение Symbol](#), задающее символ для точечного объекта.

Linestyle Pen – предложение, определяющее стиль линии, используемый для объекта типа линия.

Regionstyle Pen (...) Brush(...) – предложение, задающее стиль линии и заливки фона, используемый для объекта типа область.

Описание

Оператор **Server Set Map** изменяет стили объектов на Карте открытой таблицы ODBC. Таблица ODBC становится отображаемой в виде карты, если выполнить [оператор Server Create Map](#).

Пример

```
Declare Sub Main
Sub Main
  Dim ConnNum As Integer
  ConnNum = Server_Connect("ODBC", "DSN=SQS;PWD=sys;SRVR=seneca")
  Server ConnNum Set Map "Cities"
  ObjectType Point
  Symbol (35,0,12)
```

```
Server ConnNum Disconnect
End Sub
```

См. также:

[оператор Server Create Map](#)

Оператор Server Versioning

Назначение

Включает или выключает контроль версий для таблиц Oracle версии 9i или более поздних, которые создают или удаляют все необходимые структуры для поддержки истории строк, что позволяет воспользоваться преимуществами Oracle Workspace Manager. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server ConnectionNumber Versioning {
  ON [ History HistoryValue ] |
  OFF [ Force { OFF | ON } ]
} Table ServerTableName
```

ON | OFF – указывает, включен (**ON**) или выключен (**OFF**) контроль версий для таблицы.

ConnectionNumber – целое число, идентификатор соединения.

ServerTableName – имя таблицы на сервере Oracle, для которой включается/выключается режим контроля версий. Длина имени таблицы не должна превышать 25 символов. Есть разница между строчными и заглавными буквами в имени таблицы.

History – не обязательный параметр при включенном контроле версий таблицы (**ON**).

Предложение **History** определяет метод контроля изменения таблицы *ServerTableName*, то есть, позволяет отслеживать изменения, сделанные во всех строках в таблицах, для которых включен контроль версий, и сохранять копии всех изменений или только самых последних изменений каждой строки. *HistoryValue* должно принимать одно из следующих значений:

- **SRV_WM_HIST_NONE** (0): изменения таблицы не отслеживаются. (Применяется по умолчанию)
- **SRV_WM_HIST_OVERWRITE** (1): с возможностью перезаписи (**W_OVERWRITE**). Создаётся вид *ServerTableName_HIST*, содержащий информацию об истории, но в нём показаны только самые последние изменения той же самой версии таблицы. История модификаций версии не составляется; то есть, последующие изменения в строке, в той же самой версии, переписывают более ранние изменения. (Колонка **CREATETIME** файла *TableName_HIST* содержит только время самого последнего обновления).

- **SRV_WM_HIST_NO_OVERWRITE (2)**: без возможности перезаписи (WO_OVERWRITE). Создаётся вид `ServerTableName_HIST`, содержащий информацию об истории, в нём будут показаны все изменения для одной и той же версии таблицы. Поддерживается история модификаций версии; то есть, последовательные изменения записи той же самой версии не переписывают более ранние изменения.

Однако, на таблицы, которые могут использовать этот режим, накладывается много ограничений. Более подробную информацию смотрите в руководстве *Oracle9i Application Developer's Guide - Workspace Manager*.

Force - необязательный параметр, используемый при контроле версий (**OFF**).

Если режим **Force** включен (**ON**), то все данные рабочих наборов, за исключением LIVE, будут удалены перед отключением контроля версий. **OFF** (по умолчанию) предупреждает отключение контроля версий в случае, если *ServerTableName* была изменена в каком-либо из рабочих наборов (кроме LIVE) и если рабочий набор, изменивший *ServerTableName*, еще существует.

Описание

Этот оператор применяется только к Oracle9i или старше. Таблица *ServerTableName*, к которой применяется контроль версий, должна иметь первичный ключ. Только владелец таблицы или пользователь с правами WM_ADMIN могут отключать или включать возможность проверки версии таблицы. Таблицы, для которых включена поддержка версий и владельцы таких таблиц не могут быть удалены. Для удаления нужно сначала отключить механизм поддержки версий. Таблицы, владельцем которых является SYS, версионность не поддерживают. Более подробную информацию смотрите в *Oracle9i Application Developer's Guide - Workspace Manager*.

Примеры

В этом примере включается контроль версий таблицы MIUUSA3

```
Dim hdbc As Integer
hdbc = Server_Connect("ORAINET", "SRVR=TROINY;UID=MIUSER;PWD=MIUSER")
Server hdbc Versioning ON Table "MIUUSA3"
```

or

```
Server hdbc Versioning ON History 1 Table "MIUUSA3"
```

В этом примере отключается контроль версий таблицы MIUUSA3.

```
Dim hdbc As Integer
hdbc = Server_Connect("ORAINET", "SRVR=TROINY;UID=MIUSER;PWD=MIUSER")
Server hdbc Versioning OFF Force ON Table "MIUUSA3"
```

См. также:

оператор **Server Create Workspace**Оператор **Server Workspace Merge***Назначение*

Применяет изменения в таблице (все строки, или те, которые описаны в предложении *Where*) рабочей области, по отношению к порождающей (*parent*) рабочей области базы данных (Oracle 9i или более новой). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server Workspace Merge
Table TableName
[ Where WhereClause ]
[ RemoveData { OFF | ON } ]
[ { Interactive | Automatic merge_keyword } ]
```

TableName – имя (псевдоним) открытой таблицы MapInfo с сервера Oracle9i или более новой версии. Таблица содержит строки, которые будут перенесены в порождающую рабочую область.

WhereClause – строка, указывающая записи для слияния с порождающей Рабочей областью.

merge_keyword – ключевое слово, ограничивающее слияние в автоматическом режиме **Automatic**.

Описание

Этот оператор применяется только к Oracle9i или старше. Все данные, которые удовлетворяют предложению *WhereClause* в таблице *TableName* применяются к порождающей Рабочей области. Снимаются любые блокировки между сливающимися строками. Если есть конфликты между объединяемой и порождающей рабочими областями, эта операция предлагает выбор вариантов того, как решить конфликт. Объединение выполняется только после разрешения всех конфликтов. Таблица не может быть объединена с рабочей областью прямого доступа LIVE (такая рабочая область не имеет порождающей рабочей области). Таблица в открытой базе данных не может быть подвергнута объединению или обновлению, если для этой таблицей не завершены транзакции.

Более подробную информацию смотрите в руководстве *Oracle9i Application Developer's Guide - Workspace Manager*.

WhereClause определяет строки, которые могут быть объединены с основной Рабочей областью. Само предложение не должно содержать ключевое слово **Where**. например, 'MI_PRINX = 20'. В предложении **Where** могут быть указаны только колонки с первичным

ключом. Предложение **Where** не может содержать подзапрос. Если *WhereClause* не определено, то все строки в таблице *TableName* будут объединены.

Если режим **RemoveData** включен (**ON**), данные в таблице (в соответствии с *WhereClause*) порожденной Рабочей области будут удалены. Этот флажок установлен, если рабочая область не имеет порожденных (т.е. не имеет нисходящих связей). **OFF** (используется по умолчанию) означает, что данные не будут удалены.

Если есть конфликты между рабочей областью, в которой произошло слияние и породившей её рабочей областью, пользователь должен сначала разрешить конфликты, чтобы слияние прошло успешно. MapInfo Pro позволяет пользователю сначала разрешить конфликты, а затем осуществлять слияния, прерывая процесса. С помощью параметров **Interactive** и **Automatic** можно управлять конфликтами. Эти предложения не действуют, если нет конфликтов между рабочей областью, в которой проводится объединение и породившей её рабочей областью.

Если выбран параметр **Interactive**, MapInfo Pro откроет диалог **Conflict Resolution**. Конфликты будут разрешены один за одним или все сразу, в зависимости от того, что выберет пользователь. После разрешения всех конфликтов, таблица будет объединена с породившей её таблицей, с учётом настроек, сделанных пользователем.

Примечание: Из-за ограничений системы, эта настройка недоступна, если используется сервер Oracle9i.

В следующей таблице указаны возможные значения для *merge_keyword*, используемые в режиме **Automatic**.

Значение <i>merge_keyword</i>	Описание
StopOnConflict	Если возник конфликт, MapInfo Pro остановится. (Этот режим используется по умолчанию, то есть в случае, если не используются параметры Interactive или Automatic .)
RevertToBase	Если произошёл конфликт, MapInfo Pro будет использовать исходные (base) значения. (исходные записи будут скопированы в порожденную рабочую область, но не в порождающую; Однако, конфликт считается разрешенным; когда выполняется объединение порожденного набора, базовые строки также копируются в порождающий набор. Обратите внимание, что BASE игнорируется для конфликтов типа вставка-вставка, если базовая строка не существует; в этом случае предложение Automatic должно содержать UseParent или UseCurrent.
UseCurrent	Если произошёл конфликт, MapInfo Pro использует значения из порождённой рабочей области.
UseParent	Если произошёл конфликт, MapInfo Pro использует значения из порождающей рабочей области.

Примеры

В следующем примере объединяются изменения в таблице GWMUSA2 где MI_PRINX=60 в MIUSER в ее порождающий рабочий набор.

```
Server Workspace Merge
Table "GWMUSA2"
Where "MI_PRINX = 60"
Automatic UseCurrent
```

См. также:

[оператор Server Workspace Refresh](#)

Оператор Server Workspace Refresh

Назначение

Синхронизирует все изменения таблицы (все строки или указанные предложением **Where**) в порождающем рабочем области с рабочей областью в базе данных (Oracle 9i или более новой). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Server Workspace Refresh
Table TableName
[ Where WhereClause ]
[ { Interactive | Automatic merge_keyword } ]
```

TableName – имя (псевдоним) открытой таблицы MapInfo с сервера Oracle9i или более новой версии. Таблица содержит строки, которые будут перенесены из порождающей (parent) рабочей области.

WhereClause определяет строки, которые могут быть перенесены из основной рабочей области. Само предложение не должно содержать ключевое слово **Where**.

merge_keyword – строка с ключевыми словами, ограничивающими обновление в автоматическом режиме (**Automatic**).

Описание

Этот оператор применяется только к Oracle9i или старше. Используются все изменения данных, которые удовлетворяют предложению *WhereClause* в таблице в порождающей Рабочей области на момент создания Рабочей области или последнего её обновления. Если есть конфликты между обновляемой и порождающей рабочими областями, эта операция предлагает выбор

вариантов разрешения конфликтов. Операция обновления может быть выполнена только после разрешения всех конфликтов. Таблица в рабочей области прямого доступа (LIVE) не может обновлена (такая рабочая область не имеет порождающей рабочей области). Таблица в открытой базе данных не может быть подвергнута объединению или обновлению, если для этой таблицей не завершены транзакции.

Более подробную информацию смотрите в *Oracle9i Application Developer's Guide - Workspace Manager*.

WhereClause определяет строки, которые могут быть перенесены из основной рабочей области. Само предложение не должно содержать ключевое слово **Where**. Например: `MI_PRINX = 20` В предложении **Where** могут быть указаны только колонки с первичным ключом. Предложение **Where** не может содержать подзапрос. Если *WhereClause* не определено, то будут обновлены все строки в таблице *TableName*.

Если есть конфликты между обновляемой и породившей её рабочими областями, пользователь должен сначала разрешить конфликты, чтобы обновление прошло успешно. MapInfo Pro позволяет пользователю сначала разрешить конфликты, а затем осуществлять обновление, не прерывая процесса. С помощью параметров **Interactive** и **Automatic** можно управлять конфликтами. Эти предложения не действуют, если нет конфликтов между обновляемой и породившей её рабочими областями.

Если выбран параметр **Interactive**, MapInfo Pro откроет диалог **Разрешение конфликтов**. Конфликты будут разрешены один за одним или все сразу, в зависимости от того, что выберет пользователь. После того, как все конфликты будут разрешены, операция обновления будет выполнена.

Примечание: Из-за ограничений системы, эта настройка недоступна, если используется сервер Oracle9i.

В следующей таблице указаны возможные значения для *merge_keyword*, используемые в режиме **Automatic**.

Значение <i>merge_keyword</i>	Описание
StopOnConflict	Если возник конфликт, MapInfo Pro остановится. (Этот режим используется по умолчанию, то есть в случае, если не используются параметры Interactive или Automatic .)
RevertToBase	Если произошёл конфликт, MapInfo Pro будет использовать исходные (base) значения. (исходные записи будут скопированы в порожденную рабочую область, но не в порождающую; Конфликт считается разрешенным. Когда выполняется объединение порожденного набора, базовые строки также копируются в порождающий набор. Обратите внимание, что BASE игнорируется для конфликтов типа вставка-вставка, если базовая строка не существует; в этом случае предложение Automatic должно содержать UseParent или UseCurrent.

Значение merge_keyword	Описание
UseCurrent	Если произошёл конфликт, MapInfo Pro использует значения из порождённой рабочей области.
UseParent	Если произошёл конфликт, MapInfo Pro использует значения из порождающей рабочей области.

Примеры

В следующем примере показано как обновить MIUSER, применяя изменения GWMUSA2 where MI_PRINX=60 в порождающем рабочем наборе.

```
Server Workspace Refresh
Table "GWMUSA2"
Where "MI_PRINX = 60"
Automatic UseParent
```

См. также:

[оператор Server Workspace Merge](#)

Функция SessionInfo()

Назначение

Возвращает различные блоки информации о сеансе работы MapInfo Pro. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SessionInfo( attribute )
```

attribute — целочисленный код, определяющий тему запроса.

Возвращаемая величина

Строка

Описание

Функция **SystemInfo()** возвращает информацию о текущем сеансе программе MapInfo Pro. Вид информации задает параметр attribute. Коды определены в MAPBASIC.DEF.

Код атрибута	ID	Возвращаемая величина
SESSION_INFO_COORDSYS_CLAUSE	1	Строка, возвращающая единицы измерения отчёта.
SESSION_INFO_DISTANCE_UNITS	2	Строка, возвращающая единицы расстояния.
SESSION_INFO_AREA_UNITS	3	Строка, возвращающая единицы площади.
SESSION_INFO_PAPER_UNITS	4	Строка, возвращающая единицы измерения отчёта. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
SESSION_INFO_DEFAULT_NATIVE_BLOCKSIZE	5	Строковый результат с возвращаемым размером блока, который будет использоваться при создании карт в традиционном, классическом формате MapInfo Tab-файла (Native).
SESSION_INFO_DEFAULT_NATIVE_BLOCKSIZE	6	Строковый результат с возвращаемым размером блока, который будет использоваться при создании карт в расширенном формате MapInfo Tab-файла (NativeX).
SESSION_INFO_TEMP_PATH	7	Строковый результат с именем и адресом временной папки, используемой в сеансе работы MapInfoPro.

Папка для временных файлов MapInfoPro используется во время сеанса работы для хранения служебных файлов, например, файлов транзакций (*.tma, *.tda, *.dmp файлы и т.п)

Это выделяемая системой временная папка TEMP, в которой создана ветка вложенных папок "MIPRO\MapInfo.Application.x64_", в названии которой присутствует идентификатор процесса MapInfoPro.exe. Например,

```
C:\temp\MIPRO\MapInfo.Application.x64_4216\
```

Если требуется получить идентификатор процесса MapInfoPro из MapBasic, то можно вызвать метод GetCurrentProcessId() Windows API с помощью такой функции:

```
Declare Function GetCurrentProcessId Lib "kernel32" () As Integer
```

Условия появления сообщения об ошибке

ERR_FCN_ARG_RANGE (644), если значение аргумента выходит за пределы, заданные при его определении.

Пример

```
Include "mapbasic.def"
print SessionInfo(SESSION_INFO_COORDSYS_CLAUSE)
```

Оператор Set Adornment

Назначение

Изменяет элемент оформления, который создает **оператор Create Adornment**. Можно изменять расположение элемента оформления, размер и стиль его рамки. Для масштабной линейки также можно изменять ее представление, единицы измерения, размеры и представление масштаба. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Adornment
Window window_id
[ Type adornment_type ]
[ Position {
  [ Fixed [ ( x, y ) [ Units paper_units ] ] ] |
  [ win_position [ Offset ( x, y ) ] [ Units paper_units ] ]
} ]
[ Layout Fixed Position { Frame | Geographic } ]
[ Size [ Width win_width ] [ Height win_height ] [ Units paper_units ] ]
[ Background [ Brush ... ] [ Pen ... ] ]
[ < SCALEBAR_CLAUSE > ]
```

Где *SCALEBAR_CLAUSE*:

```
[ BarType type ]
[ Ground Units distance_units ]
[ Display Units paper_units ]
[ BarLength paper_length ]
[ BarHeight paper_height ]
[ BarStyle [ Pen .... ] [ Brush ... ] [ Font ... ] ]
```

```
[ Scale [ { On | Off } ] ]
[ Auto [ { On | Off } ] ]
```

Параметр *adornment_type* может принимать значение **scale bar** (масштабной линейки)

Значения (*x*, *y*) в предложении **Fixed** задают смещение элемента оформления от левого верхнего угла карты, для которого смещения равно (0, 0). При этом способе разметки элементов оформления, элемент оформления сохранит свое положение на карте при изменении размера карты. Например, положение в точке (3, 3) дюйма на карте может быть в нижнем правом углу карты крупного масштаба или в центре карты мелкого масштаба. При изменении размера карты, положение элемента оформления будет стремиться сохраниться внутри выведенной на экран карты.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цитеро. Обычно, по умолчанию, значение содержит "бумажные" единицы MapBasic, смотрите: **Оператор Set Paper Units**.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

win_position – задаёт один из кодов положения элемента оформления, которые определены в файле MAPBASIC.DEF и перечислены в следующем *Описании*.

Смещение (*x*, *y*) в предложении **Offset** отсчитывается от выбранной позиции.

Переменные *win_width* и *win_height* определяют размер элемента оформления. В случае масштабной линейки MapInfo Pro игнорирует эти параметры, так как размер масштабной линейки задается переменной *BarLength*.

type – задаёт один из кодов положения масштабной линейки, которые определены в файле MAPBASIC.DEF и перечислены в следующем *Описании*.

distance_units – единица измерения, которые представляет масштабная линейка:

distance value	Представляет
"ch"	чейны
"cm"	сантиметры
"ft"	футы (иногда называется "международным футом"; один "международный фут" равен точно 30,48 cm)
"in"	дюймы

distance value	Представляет
"km"	километры
"li"	линки
"m"	метры
"mi"	мили
"mm"	миллиметры
"nmi"	морские мили (1 морская миля = 1852 метрам)
"rd"	роды
"survey ft"	геодезический фут США геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам) геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам)
"yd"	ярды

Параметр *paper_length* в переменной *paper_units* задает длину линейки в отчете. Можно задавать максимальную длину линейки в 34 дюйма или 86.3 см на распечатке.

Параметр *paper_height* в переменной *paper_units* высоту линейки в отчете. Установите высоту элемента оформления, максимум 44 дюйма или 111.76 см на печатной карте.

Описание

Масштабная линейка имеет длину *paper_length* в единицах *paper_units*.

Предложение **Position** может либо принимать значение **Fixed** и тогда элемент оформления будет помещен в левый верхний угол окна карты, либо значение будет отсчитываться относительно некой точки в окне карты, заданной переменной *win_position*.

Offset – расстояние, на которое элемент оформления будет смещен на карте в одном из заданных положений *win_position*. Например, если значение параметра *win_position* совпадает с ADORNMENT_INFO_MAP_POS_TL (вверху, слева), то x-значение параметра **Offset** будет отсчитываться вправо от от верхнего левого угла, а y-значение – вниз от верхнего левого угла.

Код атрибута	ID	Описание положения элемента оформления
ADORNMENT_INFO_MAP_POS_TL	0	В верхнем левом углу. Значение x параметра Offset отсчитывается вправо, а значение y – вниз.
ADORNMENT_INFO_MAP_POS_TC	1	В центре от верхнего края. Значение x параметра Offset игнорируется.
ADORNMENT_INFO_MAP_POS_TR	2	В верхнем правом углу. Значение x параметра Offset отсчитывается влево, а значение y – вниз.
ADORNMENT_INFO_MAP_POS_CL	3	В центре от левого края. Значение координаты y параметра Offset игнорируется.
ADORNMENT_INFO_MAP_POS_CC	4	Строго по центру. Оба значения координат x и y параметра Offset игнорируются.
ADORNMENT_INFO_MAP_POS_CR	5	В центре от правого края. Значение координаты y параметра Offset игнорируется.
ADORNMENT_INFO_MAP_POS_BL	6	В нижнем левом углу. Значение x параметра Offset отсчитывается вправо, а значение y – вверх.
ADORNMENT_INFO_MAP_POS_BC	6	В центре от нижнего края. Значение x параметра Offset игнорируется.
ADORNMENT_INFO_MAP_POS_BR	6	В нижнем правом углу. Значение x параметра Offset отсчитывается влево, а значение y – вверх.

Предложение **Layout Fixed Position** задает **фиксированное** положение элемента оформления на листе отчета. Если это параметр **Geographic**, то элемент оформления будет помещен в то же положение на листе отчета, которое этот элемент оформления занимает на карте, выведенной на экран. Если размеры рамки отчета будут изменены, то элемент оформления изменит свое положение относительно границ рамки, сохраняя свои географические координаты. Если значение этого параметра **Frame**, то элемент оформления останется в заданном относительно рамки положении, подобно описанному выше в разделе о параметре **Position**. Если в предложении **Position** элемент оформления помещен в точку с координатами (1.0, 1.0) дюймов, то этот элемент оформления будет помещен в точку на дюйм правее и ниже верхнего левого угла рамки. Изменение размера рамки не приводит к изменению положения элемента оформления относительно рамки. По умолчанию используется значение параметра **Geographic**.

Предложение **Background** с параметром **Brush** задает фоновую штриховку, используемую при создании или изменении масштабной линейки.. Параметр **Pen** в этом же предложении задает стиль контура при прорисовке фона.

Brush – корректное **предложение Brush**. Разрешены только непрозрачные штриховки. Несмотря на то, что прозрачные и прочие штриховки не вызовут сообщения об ошибках, будет всё равно подставлен непрозрачный аналог. Это предложение создает фон для элемента оформления.

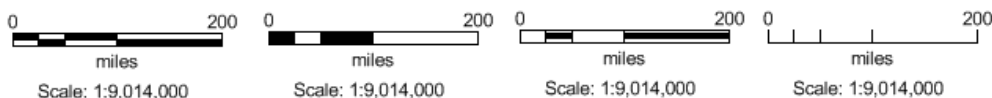
Pen – корректное **предложение Pen**. Из-за особенностей механизма прорисовки окна элемента оформления в окне карты, линии шириной иной, чем 1 пиксель могут быть показаны некорректно. Кроме того, стили линий, которые на являются непрозрачными, могут показываться некорректно. Это предложение задает стиль рамки элемента украшения.

Font – корректное **предложение Font**.

Если в предложении **Auto** задано значение **On**, то значения, показываемые в масштабной линейке, будут автоматически округляться. Если задано **Off**, то эти значения округляться не будут, как это было в прошлых версиях. Стандартное значение – **Auto Off**.

Можно задать режим **Scale On** для показа дробного представления масштаба. В MapInfo Pro, масштабная линейка обычно не показывает числового представления масштаба, такого как 1:63,360 или 1:1,000,000, называемого **картографическим масштабом**

Когда требуется определить тип масштабной линейки **BarType**, значение параметра *type* представляет собой код масштабной линейки, которая будет показана: **0** Классический, **1** Сплошной, **2** Линией или **3** Чёрточками.



Коды определены в файле MAPBASIC.DEF и:

Код атрибута	ID	Описания масштабных линеек
SCALEBAR_INFO_BARTYPE_CHECKEDBAR	0	Классический
SCALEBAR_INFO_BARTYPE_SOLIDBAR	1	Сплошной
SCALEBAR_INFO_BARTYPE_LINEBAR	2	Линией
SCALEBAR_INFO_BARTYPE_TICKBAR	3	Чёрточками

Пример

```

set adornment
window 261727232
type scalebar
position 6
background Brush (2,16777215,16777215) Pen (1,2,0)
bartype 0 ground units "km" display units "cm"
barlength 0.978478 barheight 0.078740
barstyle Pen (1,2,0) Brush (2,0,16777215) Font ("Arial",0,8,0)
scale on

```

См. также:

Оператор Create Adornment

Оператор Set Application Window

Назначение

Устанавливает, какое окно будет порождающим для всех новых диалогов и окон. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Application Window HWND
```

HWND – целое число типа Integer, уникальный системный номер окна.

Описание

Этот оператор объявляет, какое окно будет окном приложения. Для всех последующих окон диалогов MapInfo Pro будет считаться, что они порождены этим другим окном. Этот прием используется в интегрированной картографии, когда окна MapInfo Pro показываются из других приложений, написанных, например, на Visual Basic.

Обычно программа, написанная на Visual Basic, сначала создает объект MapInfo Object и затем посылает MapInfo Pro **оператор Set Application Window**, после чего приложение на Visual Basic становится порождающим окном для диалогов MapInfo Pro. Если оператор **Set Application Window**, не был выполнен, то становится очень трудно координировать передачу фокуса между MapInfo Pro и Visual Basic.

Использование команды `Set Application Window 0` возвратит MapInfo Pro к ее исходному состоянию. Этот оператор переподчиняет окна диалогов. Для переподчинения документального окна, такого как окна карты, используйте **оператор Set Next Document**.

Примечание: Если задается параметр *HWND* как шестнадцатеричное значение, то необходимо использовать приставку *&H* с *шестнадцатеричным числом*. Иначе MapInfo Pro попытается интерпретировать параметр как десятичное значение. (Это бывает, когда программа на Visual Basic создает командную строку, содержащую **оператор Set Application Window**.)

Для получения другой информации об интегрированной картографии смотрите *Руководство Пользователя MapBasic*.

См. также:

оператор Set Next Document

Оператор Set Area Units

Назначение

Устанавливает единицы измерения площади для использования в операторах и функциях MapBasic по умолчанию. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Area Units area_name
```

area_name - строка, определяющая единицы измерения площади (к примеру, "акр").

Описание

Оператор **Set Area Units** устанавливает единицы измерения площади. Установки единиц измерения площади используются в диалоге **SQL-запрос** в MapInfo Pro. По умолчанию, MapBasic использует квадратные мили ("sq mi"), т. е. если в Вашей программе нет оператора **Set Area Units**, то единицами измерения площади будут квадратные мили. Параметр *area_name* должен иметь строковое значение, список которых приведен в таблице:

Имя единицы	Представляет
"acre"	акры
"hectare"	гектары

Имя единицы	Представляет
"perch"	перчи
"rood"	роды
"sq ch"	квадратные чейны
"sq cm"	квадратные сантиметры
"sq ft"	квадратные футы
"sq in"	квадратные дюймы
"sq km"	квадратные километры
"sq li"	квадратные линки
"sq m"	квадратные метры
"sq mi"	квадратные мили
"sq mm"	квадратные миллиметры
"sq rd"	квадратные роды
"sq survey ft"	квадратные геодезические футы
"sq yd"	квадратные ярды

Пример

```
Set Area Units "acre"
```

См. также:

функция Area(), оператор Set Distance Units

Оператор Set Blocksize

Назначение

Позволяет задавать размеры блоков текущем сеансе работы MapInfo Pro.

Синтаксис

```
Set Blocksize Native { nBytes | default }
```

или

```
Set Blocksize NativeX { nBytes | default }
```

nBytes – значение, которое лежат в диапазоне от нуля (0) до 32767. Если заданное значение меньше 512, то оно будет округлено до 512. Минимальный размер блока может 512, иначе оно будет округлено до значения кратного 512.

Стандартное значение, используемое по умолчанию, – 16834.

Описание

Эта команда может использоваться, для того чтобы переназначить размер блока, который задаётся в файле `Preferences.xml`, для использования в текущем сеансе работы MapInfo Pro. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Выполнение команды не обновляет файл **Preferences.xml**. Таким образом, настройка размера блока, заданная с помощью этой команды, не будет использоваться в следующих сеансах работы MapInfo Pro.

Пример

В следующем примере в текущем сеансе работы MapInfo Pro задаётся размер блока 30720 для классических, нативных таблиц:

```
Set Blocksize Native 30720
```

В следующем примере в текущем сеансе работы MapInfo Pro задаётся размер блока 16834 для расширенных таблиц NativeX :

```
Set Blocksize NativeX default
```

Оператор Set Browse

Назначение

Изменяет существующее окно списка. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Browse
[ Window window_id ]
[ Grid { On | Off } ]
[ Row row_num ]
[ Column column_num ]
[ Columns Resize ]
[ Order By sortColumn [ Desc ] [ , sortColumn2 ... ] ]
[ Order None ]
[ Filter Where
  (filterCondition [ And | Or filterCondition ] )
[ And (filterCondition [ And | Or filterCondition ] ) ... ] ]
[ Filter None ]
[ SortFilter { On | Off } ]
[ Reapply ]
```

window_id – целочисленный идентификатор окна списка или районирования.

row_num – целое число типа SmallInt от одного и более, где 1 представляет первую строку таблицы;

column_num – целое число типа SmallInt от нуля и более, где 0 представляет первую колонку таблицы.

sortColumn определяет колонку(и) для сортировки, используя синтаксис как у SQL Select.

Desc - необязательный параметр и задаёт сортировку в обратном порядке. До четырёх колонок может использоваться при сортировке, например:

```
Set Browse Order By Country, City, Income Desc
```

filterCondition – простое логическое условие (например, Population > 12345), определяющее показ строки в окне списка. Выражения могут включать только простые имена колонок, операторы и константы (например, числовую константу 12345.67 и символьную строку типа "Kerry"). Не поддерживаются вызовы функций, за исключением оператора Like в сочетании с замещающими символами, примененного к символьной колонке. К одной колонке можно применять до двух условных предложений, заключенных в скобки.

Описание

Оператор **Set Browse** управляет представлением окна Списка. Если параметр *window_id* не задан, то действия оператора распространяются на самое верхнее из открытых окон Списка.

С помощью дополнительного, необязательного параметра **Window** можно выбирать, какое окно документа следует использовать. Если идентификатор окна *window_id* не задан, то будет найдено последнее использованное окно списка или районирования. Если ни окно районирования, ни окно списка не является самым верхним, но среди открытых окон есть оба этих типа, то Set Browse открывает одно из окон списка. Для того чтобы задать тип окна следует либо включить в функцию Set Browse параметр идентификатора окна *window_id*, либо до вызова функции Set Browse сделать окно районирования самым верхним окном.

Дополнительное предложение **Grid** позволяет показать (включает) или скрыть (отключает) линии сетки в окне списка.

Предложения **Row** и **Column** позволяют назначить строку, которая будет видна первой, и колонку, которая будет самой левой в окне Списка.

С помощью дополнительного, необязательного параметра **Columns** можно изменить ширину колонки по заголовку колонки и содержимому видимых ячеек. Открываясь в первый раз, окно списка автоматически подстраивает ширину колонок так, чтобы все значения, попавшие на экран, были видимы. Позже, когда окно списка прокручивается вертикально, ширина колонок уже не подстраивается под ширину показываемых значений. Чтобы такая подстройка происходила, задайте предложение Columns. После повторного вычисления ширины колонки, эта ширина не меняется при прокрутке, то есть ширина колонок уже не зависит от новых показываемых данных. Если пользователь установит ширину колонки вручную, то эта ширина больше не меняется.

Order By – если был использован диалог **Внести поля в список** для изменения имён колонок оператор Set Browse должен использовать эти новые имена колонок. Если имя колонки содержит пробелы ("Customer Name"), то в операторе Set Browse должен использоваться номер колонки, например **Order By Col/2** или **Order By 2**.

Предложение **Filter Where** применяет условия фильтрации к записям в окне списка. Синтаксически предложение **Where** устроено проще, чем его аналог в операторе SQL Select. Для каждой колонки должно быть задано не более двух условий, и в них не должны применяться вызовы функций и сложные выражения. Можно, однако, применять оператор Like в сочетании с замещающими символами, например:

```
Set Browse Window FrontWindow() Filter Where (age > 50 And age <= 70)
Set Browse Window FrontWindow() Filter Where
  (continent Like "%America") And (startdate > "19991231")
```

Создав условие сортировки один раз, Вы можете затем применять его или отменять предложением **SortFilter**:

```
Set Browse Window FrontWindow() SortFilter Off
```

Если к содержимому окна применена сортировка и/или фильтрация, то после изменения исходной таблицы содержимое окна не изменится автоматически. Чтобы обновить окно, применяйте предложение **Reapply**:

```
Set Browse Window FrontWindow() Reapply
```

Чтобы отменить фильтр, примените предложение **Filter None**.

Условие фильтра ассоциируются с окном списка, а не с таблицей. Чтобы запомнить условия фильтрации, сохраните файл рабочего набора. Чтобы создать таблицу, которая будет содержать отфильтрованные записи из окна списка, применяйте **оператор Create Query**.

Для того, чтобы изменить высоту, ширину и расположение окна, используйте **оператор Set Window**.

Ошибки

Во время исполнения программы порождается ошибка, если в предложении Order By заданы несуществующие в окне списка колонки, или если предложение Order By пытается провести сортировку с участием зарезервированных имен колонок "OBJECT" или "ROWID".

Примеры

```
Dim i_browser_id As Integer
Open Table "world"
Browse * From world
i_browser_id = FrontWindow()
Set Browse Window i_browser_id Row 47
```

Оператором **Set Browse** можно пользоваться для сортировки до пяти колонок с помощью параметра **Order By**.

```
Set Browse Order By Country, State, City, ZipCode, IncomeGroup
```

См. также:

оператор Browse, **оператор Set Window**

Оператор Set Buffer Version

Назначение

Задаёт режим, в котором MapInfo Pro выполняет операции семейства Buffer... с использованием старого алгоритма из версий MapInfo Pro 9.5.1 и более ранних.

Синтаксис

```
Set Buffer Version version_num
```

version_num – число 950 или 951. В версиях 951 и выше будет сгенерирована ошибка, а в версиях 950 и более ранних используется старый алгоритм.

Описание

В MapInfo Pro 9.5.1 используется новый алгоритм обработки данных для самопересекающихся полигонов. Однако, Oracle не полагает такие полигоны корректными, и не может возвращать такие объекты в MapInfo Pro или обрабатывать их. Чтобы полноценно обрабатывать данные, содержащие самопересекающиеся полигоны в Oracle, Вы должны либо найти и удалить их, либо выполнять операции Buffer на основе старого алгоритма (поддерживаемого версиями MapInfo Pro до 9.5.1).

Чтобы узнать, содержатся ли в данных самопересекающиеся полигоны, выполните команду **Проверка полигонов** из меню **Объекты**. Для удаления нежелательных самопересекающихся полигонов примените к таблице команду **Коррекция топологии**. Имейте в виду, что операции Проверка полигонов и Коррекция топологии займут некоторое время для обработки больших таблиц.

Чтобы явно задать, какой алгоритм будет обрабатывать команды семейства Buffer..., добавьте инструкцию MapBasic в рабочий набор, такой как `startup.wor`, или выполните соответствующую команду в окне **MapBasic**. Нельзя добавлять эти команды в приложение MapBasic напрямую, поскольку команда не будет откомпилирована. Однако, она сработает, будучи включена в состав оператора Run Command в программе MapBasic.

Пример

Операции семейства Buffer из MapInfo Pro 9.5 или более ранних, будут работать при выполнении команды MapBasic:

```
Set Buffer Version 950
```

Операции семейства Buffer из MapInfo Pro версий 9.5.1 и более поздних, будут работать при выполнении команды MapBasic:

```
Set Buffer Version 951
```

См. также:

[оператор Set Combine Version](#)

Оператор Set Cartographic Legend

Назначение

Оператор Set Cartographic Legend позволяет Вам включать или выключать режим перерисовки, обновления, устанавливать книжную или альбомную ориентацию легенды, или изменять порядок следования разделов в картографической легенде, которую создает **оператор Create Cartographic Legend**. (Для изменения размера, позиции или заголовка окна легенды, используйте **оператор Set Window**.) Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Cartographic Legend
[ Window legend_window_id ]
Redraw { On | Off }
```

или

```
Set Cartographic Legend
[ Window legend_window_id ]
[ Refresh ]
[ Portrait [ Columns number_of_columns ] |
  Landscape [ Lines number_of_lines ] ]
[ Align ]
[ Style Size { Small | Large } ]
[ Frame Order { frame_id, frame_id, frame_id, ... } ]
```

legend_window_id – целочисленный идентификатор окна, который возвращают **функция FrontWindow()** и **функция WindowID()**.

frame_id – ID-номер раздела легенды. Нельзя использовать имя слоя. Например, три раздела легенды могут иметь идентификаторы 1, 2 и 3.

number_of_columns – указывает ширину легенды.

number_of_lines – определяет высоту легенды.

Описание

Оператор **Set Cartographic Legend** позволяет включать или выключать режим перерисовки, обновления, устанавливать книжную или альбомную ориентацию легенды, или изменять порядок следования разделов в картографической легенде, которую создает **оператор Create Cartographic Legend**.

Если предложение **Window** не определено, MapInfo Pro будет использовать самое верхнее окно легенды.

Нельзя использовать другие предложения, если используется **Redraw**.

Ключевое слово **Refresh** приводит к обновлению окна легенды. Таблицы, для которых будут обновляться разделы легенды, будут проверены на предмет используемых в них стилей.

Ключевые слова **Portrait** и **Landscape** определяют ориентацию окна легенды.

Align – выравнивает стиль и текст в рамках независимо от того, в каком слое открыто окно легенды.

Предложение **Frame Order** изменяет порядок следования разделов в легенде.

Пример

Если был использован [оператор Create Cartographic Legend](#) для выбора легенды большого размера, следующий пример продемонстрирует обновление текущего окна легенды:

```
Set Cartographic Legend Window WindowID(0) Refresh Portrait Align Style
Size Large
```

См. также:

[оператор Add Cartographic Frame](#), [оператор Alter Cartographic Frame](#), [оператор Create Cartographic Legend](#), [оператор Remove Cartographic Frame](#)

Оператор Set Combine Version

Назначение

Задаёт режим, в котором MapInfo Pro выполняет операции семейства Combine с использованием старого алгоритма из версий MapInfo Pro 9.5.1 и более ранних.

Синтаксис

```
Set Combine Version version_num
```

version_num – число 950 или 951. В версиях 951 и выше будет сгенерирована ошибка, а в версиях 950 и более ранних используется старый алгоритм.

Описание

В MapInfo Pro 9.5.1 используется новый алгоритм обработки данных для самопересекающихся полигонов. Однако, Oracle не полагает такие полигоны корректными, и не может возвращать такие объекты в MapInfo Pro или обрабатывать их. Чтобы полноценно обрабатывать данные, содержащие самопересекающиеся полигоны в Oracle, необходимо либо найти и удалить их, либо выполнять операции Combine на основе старого алгоритма (поддерживаемого версиями MapInfo Pro до 9.5.1).

Чтобы узнать, содержатся ли в данных самопересекающиеся полигоны, выполните команду **Проверка полигонов** из меню **Объекты**. Для удаления нежелательных самопересекающихся полигонов примените к таблице команду **Коррекция топологии**. Имейте в виду, что операции Проверка полигонов и Коррекция топологии займут некоторое время для обработки больших таблиц.

Чтобы явно задать, какой алгоритм будет обрабатывать команды семейства Combine..., добавьте инструкцию MapBasic в Рабочий набор, такой как `startup.wor`, или выполните соответствующую команду в окне **MapBasic**. Нельзя добавлять эти команды в приложение MapBasic напрямую, поскольку команда не будет откомпилирована. Однако, она сработает, будучи включена в состав оператора Run Command в программе MapBasic.

Пример

Операции семейства Combine из MapInfo Pro 9.5 или более ранних, будут работать при выполнении команды MapBasic:

```
Set Combine Version 950
```

Операции семейства Combine из MapInfo Pro 9.5.1 или более поздних, будут работать при выполнении команды MapBasic:

```
Set Combine Version 951
```

См. также:

[оператор Set Buffer Version](#)

Оператор Set Command Info

Назначение

Помещает в память значения, которые может прочитать [функция CommandInfo\(\)](#), из другой процедуры. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Command Info attribute To new_value
```

attribute – код, который использует **функция CommandInfo()**, например, CMD_INFO_ROWID (2).

new_value – новая величина, тип которой должен соответствовать значению кода *attribute* (например, если используется код CMD_INFO_ROWID, то параметр *new_value* должен быть положительным целым числом типа Integer).

Описание

Обычно функция CommandInfo() возвращает значения, описывающие системные события. Оператор **Set Command Info** помещает значения в память, так что следующий за этим вызов функции CommandInfo() возвращает заданные значения вместо системных.

Пример

Допустим, что в программе есть **процедура SelChangedHandler**. Из ее тела следующая функция определяет идентификатор строки, которая была выбрана или исключена из выбора:

```
CommandInfo(CMD_INFO_ROWID)
```

Когда **процедура SelChangedHandler** вызывается автоматически, то устанавливаются данные, которая должна считать функция CommandInfo(). Теперь представим, что **процедура SelChangedHandler** вызывается явно, через оператор Call — например, для нужд отладки. Тогда перед **оператором Call** должен быть размещен оператор, значение которого использует **функция CommandInfo()**:

```
Set Command Info CMD_INFO_ROWID To 1
```

См. также:

функция CommandInfo(), оператор Set Handler

Оператор Set Connection Geocode

Назначение

Настраивает соединение с удаленной службой геокодирования. Соединение уже должно быть создано с использованием **оператора Open Connection**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```

Set Connection connection_number Geocode
[ Batch Size batch_size ]
[ ResultCode MarkMultiple [ On | Off ] ]
[ MixedCase [ On | Off ] ]
[ Match
  [ StreetName [ On | Off ] [ , ] ]
  [ StreetNumber [ On | Off ] [ , ] ]
  [ Municipality [ On | Off ] [ , ] ]
  [ CountrySubdivision [ On | Off ] [ , ] ]
  [ CountrySecondarySubdivision [ On | Off ] [ , ] ]
  [ PostalCode [ On | Off ] [ , ] ]
  [ MunicipalitySubdivision [ On | Off ] [ , ] ]
  | All [ On | Off ] [ , ] ] ]
[ Fallback
  [ Geographic [ On | Off ] [ , ] ]
  [ PostalCode [ On | Off ] [ , ] ] ]
[ Dictionary
  [ All | Address | User | Prefer ( Address | User ) ]
  [ Offset [ [ Center offset_num_expr Units distance_unit_name ]
    [ End offset_num_expr Units distance_unit_name ]
  [ PassThrough name value, name value, ... ] ] ]

```

connection_number – целое число, указатель на соединение, которое устанавливает оператор **Open Connection**.

batch_size – целое число, максимальное число записей отправляемых на сервер за один раз.

offset_num_expr – численное выражение, указывающее сдвиг от концов или центра улицы. Это значение есть просто сдвиг точки, возвращаемой сервером от центра или концов улицы соответственно.

distance_unit_name – строка, содержащая единицы в которых вычисляется *offset_num_expr*.

name – имя, одна из частей пары параметров, передаваемых на сервер геокодирования.

value – значение, одна из частей пары параметров, передаваемых на сервер геокодирования.

Описание

Оператор **Set Connection** используется для установки предпочтительных параметров геокодирования, для того, чтобы не задавать их каждый раз в запросе. Оператор **Set Connection** использует шесть параметров. Это **Batch**, **Match**, **Fallback**, **Dictionary**, **Offset** и **PassThrough**.

Предложение Batch

Предложение **Batch** определяет максимальное количество записей передаваемых серверу за один раз. Это позволяет оптимизировать обработку записей таким образом, чтобы сохранить баланс времени затрачиваемого локальным компьютером и удалённой службой. Если это

число велико, то потребуется больше времени на ожидание обработки данных на сервере. Если число мало, то пользователь имеет больше возможностей для отмены запроса. После того как запрос отправлен на сервер, прервать его невозможно. Если прервать выполнение команды, оставшиеся запросы не обрабатываются

Предложение *Match*

Если параметр близкого соответствия **Match** установлен, то сервер геокодирования будет учитывать при обработке только передаваемые названия, полностью совпадающие с названиями, по которым выполняется геокодирование. Например, если параметр **Match StreetName** равен **On**, сервер геокодирования не будет учитывать названия улиц, которые не полностью соответствуют геокодируемому названию улицы, в качестве близкого варианта.

Для индивидуальных настроек, по умолчанию используется параметр **On** оператора **Match**. Если конкретный параметр упомянут, то это равносильно тому, что для него установлен режим **On**. Например, команда `Set Connection connectionHandle Geocode Match Municipality` эквивалентна `Set Connection connectionHandle Geocode Match Municipality On`.

Match StreetName указывает, используется ли наименование улицы при поиске соответствия.

Match StreetNumber указывает, используется ли номер улицы при поиске соответствия.

Match Municipality указывает, используется ли название городского района при поиске соответствия.

Match CountrySubdivision указывает, используется ли местное административное деление (обычно область или район) при поиске соответствия.

Match CountrySecondarySubdivision указывает, используется ли вторичное административное деление при поиске соответствия.

Match PostalCode указывает, используется ли почтовый индекс при поиске соответствия.

Match MunicipalitySubdivision указывает, используется ли городской район при поиске соответствия.

Match All устанавливает все режимы поиска соответствий **On** или **Off**. Этот режим может использоваться в комбинации с любыми другими. Например, команда `Match All On`, `Match PostalCode Off` включает все параметры совпадения и отключает выравнивание всех кодов почтовых индексов.

Предложение *FallBack*

Fallback Geographic указывает, использовать ли геокодирование введённого адреса по географическому центроиду, если геокодирование на уровне улиц не может быть осуществлено. Это значение имеет смысл, если введённый вами адрес содержит улицу. Если запись не содержит адреса улицы, это значение игнорируется.

Fallback Postal указывает, использовать ли геокодирование введённого адреса, используя центр оид почтовой зоны, если геокодирование на уровне улиц не может быть осуществлено.

Предложение Dictionary

Dictionary задает комбинацию использования словарей, адресного словаря MapMarker и пользовательского словаря, используемых в процессе геокодирования. Существует пять возможных вариантов для предложения **Dictionary**:

- **Dictionary All** – использовать оба словаря и пользовательский и адресный
- **Dictionary Address** – использовать только адресный словарь
- **Dictionary User** – использовать только пользовательский словарь
- **Dictionary Prefer Address** – использовать оба словаря с приоритетом адресного
- **Dictionary Prefer Address** – использовать оба словаря с приоритетом пользовательского

Предложение Offset

Offset End – указывает расстояние, на которое точка сдвигается от перекрёстка.

Offset End – указывает расстояние, на которое точка сдвигается от центральной линии улицы.

Units – строка, описывающая единицы, в которых исчисляется **Offset Center** и **Offset End**. Список доступных единиц измерения смотрите в разделе **оператор Set Distance Units**.

Предложение PassThrough

PassThrough – это набор пар "имя/значение", пересылаемых на сервер геокодирования. Эти пары специфичны для серверов геокодирования и документируются для каждого отдельного сервера.

Пример

Следующий пример устанавливает соединение с сервером геокодирования с некоторыми режимами поиска соответствия.

```
set connection MapMarkerHandle1 geocode match streetname, streetnumber,
municipality, municipalitysubdivision, postalcode, countrysubdivision
```

Следующий пример добавляет предложение **PassThrough** к оператору **Set Connection**. В этом примере отбираются только сертифицированные CASS-результаты запроса для данных по США.

```
PassThrough "KEY_CASS_RULES" "true"
```

См. также:

оператор **Geocode**, оператор **Open Connection**

Оператор Set Connection Isogram

Назначение

Настройка операций построения зон транспортной доступности. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Connection connection_handle Isogram
[ Banding [ On | Off ] ]
[ MajorRoadsOnly [ On | Off ] ]
[ MaxOffRoadDistance distance_value Units distance_units ]
[ ReturnHoles [ On | Off ] ] [ MajorPolygonOnly [ On | Off ] ]
[ SimplificationFactor simplification ]
[ PointsOnly [ On | Off ] ]
[ DefaultAmbientSpeed ambient_speed
Units distance_units Per time_units ]
[ DefaultPropagationFactor propagation_factor ]
[ Batch Size batch_size ]
```

connection_handle – целое, определяющее число соединений, которое возвращает оператор **Open Connection**.

distance_value – вещественное значение, указывающее максимальное расстояние передвижения по просёлочным дорогам сети.

distance_units – строка, содержащая единицы в которых вычисляется *distance_value*. Полный список допустимых единиц измерения расстояний приводится в разделе об операторе **Set Distance Units**.

simplification – вещественное, контролирует число узлов в результирующем полигоне, выраженное в процентах. Допустимые значения от 0 до 1 включительно.

ambient_speed – численное значение, определяющее рекомендуемую по умолчанию скорость движения. Число, используемое в параметрах *distance_units* и *time_units*.

time_units – строка, указывающая единицы времени. Допустимые значения “hr”, “min” и “sec”.

propagation_factor – вещественное значение, указывающее значение по умолчанию для параметра движения. Допустимые значения от 0 до 1 включительно.

batch_size – целое выражение, указывающее размер каждого запроса, передаваемого на сервер. По умолчанию принимается значение 2, максимальное - 50.

Описание

Оператор **Set Connection Isogram** конфигурирует соединение используемое для создания зон транспортной доступности (используя **оператор Create Object**).

Banding – используется только, когда создаётся несколько зон доступности по времени или расстоянию в операции построения зон транспортной доступности. Если задано значение **On**, зоны создаваемые для движения из одной точки будут неперекрывающимися. Маленькие территории вырезаются из результата. Таким образом получаются регионы без "брешей", представляющих редкие исключения на практике. Например, если создаются зоны транспортной доступности в 10, 20 и 30 минут, то 20-ти минутная изолиния будет очерчивать территорию доступную в интервале времени от 10 до 20 минут, а 30-ти минутная изолиния территорию доступную в течении от 20 до 30 минут движения. Если задан параметр **Off**, то построенные зоны будут изображать площадь, доступную от 0 до заданного значения времени или расстояния.

MajorRoadsOnly указывает, использовать ли при создании зон транспортной доступности только основные дороги. В режиме **MajorRoadsOnly** изолинии создаются значительно быстрее.

MaxOffRoadDistance указывает максимально допустимое расстояние для движения по просёлкам.

ReturnHoles определяет наличие или отсутствие брешей в полученных результатах.

MajorPolygonOnly определяет, что полученная зона будет иметь только одну внешнюю границу.

SimplificationFactor определяет параметр упрощения формы полигона. Параметр упрощения выражается в процентах от количества исходных узлов той части, по которым будет построен полигон-результат. Полигон или группа точек может содержать много точек-узлов. Параметр упрощения – это вещественное число между 0.01 и 1.0 (1 - 100% и 0.01 - 1%). Чем меньше это значение, тем меньше точек в полигоне и тем быстрее осуществляется обмен через Internet-соединение. По умолчанию её значение равно 0.05.

PointsOnly определяет, как обрабатывать не точечные объекты – пропускать или нет.

DefaultAmbientSpeed – используется только когда указано время. Пример синтаксиса:

```
DefaultAmbientSpeed 12 "mi" Per "hr"
```

DefaultPropagationFactor определяет процентное отношение проселочных дорог в оценке остатка (расстояний) допустимое при вычислении полос дальностей. Дороги, не включенные в сеть дорог, могут быть в том числе и проездами или подъездными путями. Параметр движения – процентное отношение оценки при вычислении дальности от начальной точки до заданной на заданном расстоянии. **DefaultPropagationFactor** используется только для построения зон по расстояниям.

По умолчанию этот параметр имеет значение 0.16.

Допустимый диапазон значений от 0.01 до 1.

Batch Size – число записей передаваемых серверу на обработку за один раз. Этот параметр может повлиять на производительность и время ответа на запрос. Чем больше отправляемый запрос, тем больше времени потребуется на создание изолиний и получение их и тем больше времени потребуется на то, чтобы прервать запрос в MapInfo Pro и обновить **индикатор**. Малое значение улучшает отклик на команду и уменьшает вероятность превышения времени ожидания. По умолчанию значение равно 2.

Пример

Следующий пример показывает использование оператора Set Connection Isogram.

```
Set Connection iConnect Isogram
Banding On MajorRoadsOnly On MaxOffRoadDistance 2 Units "mi"
ReturnHoles On MajorPolygonOnly On SimplificationFactor .05
DefaultAmbientSpeed 50 Units "mi" Per "hr" DefaultPropagationFactor .2
Batch Size 2 Point On
```

См. также:

оператор Create Object, оператор Open Connection

Оператор Set CoordSys

Назначение

Назначает координатную систему, в дальнейшем используемую прикладной программой MapBasic. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set CoordSys...
```

CoordSys – слово, с которого начинается стандартное предложение оператора для определения координатной системы.

Описание

Оператор **Set CoordSys** устанавливает координатную систему, которая в дальнейшем будет использоваться прикладной программой MapBasic. По умолчанию, MapBasic использует долготу и широту. Данная установка влияет на возвращаемые значения географическими функциями, такими как **функция CentroidX()** и **функция ObjectNodeX()**. MapBasic-программа может выполнить собственный оператор **Set CoordSys**, вследствие чего значения,

возвращаемые географическими функциями, будут автоматически отражать новую координатную систему. При этом координатная система в MapInfo может оставаться прежней.

Оператор **Set CoordSys** не действует на окно карты. Чтобы задать проекцию или систему координат окна карты, следует использовать оператор **Set Map...CoordSys**.

В параметре CoordSys может использовать предложения **Table** или **Window** для установки системы координат, такой же, как в таблице, или такой же, как в окне. Подробнее см. **предложение CoordSys**.

Пример

Установим с помощью оператор **Set CoordSys** не топоцентрическую координатную систему:

```
Set CoordSys Earth
```

Следующий оператор **Set CoordSys** устанавливает в качестве координатной системы равноплощадную проекцию Алберса.

```
Set CoordSys Earth  
Projection 9,7,"m",-96.0,23.0,20.0, 60.0, 0.0, 0.0
```

В окне отчета координатная система определяется оператором **Set CoordSys**. Перед работой с объектами отчета необходимо определить координатную систему на странице.

```
Set CoordSys Layout Units "in"
```

Примечание: Выполнив оператор **Set CoordSys Layout**, MapBasic продолжает использовать координатную систему отчета до тех пор, пока не переопределите ее. Аналогично, если требуется работать с объектами на проецированной карте, выполните оператор **Set CoordSys Earth**.

См. также:

предложение CoordSys, **оператор Set Area Units**, **оператор Set Distance Units**, **оператор Set Paper Units**

Оператор Set Cursor

Назначение

Переключает между 1-разрядным и 32-разрядным цветом пикселей курсора: Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

При работе в среде Citrix XenApp могут наблюдаться задержки в MapInfo Pro при рендеринге 32-разрядных курсоров, что приводит к уменьшению производительности. Переключение к 1-разрядному отображению курсоров исправляет проблему.

Синтаксис

```
Set Cursor Truecolor ( On | Off )
```

Описание

Параметр **Truecolor** включает переключает 32-разрядный режим отображения курсоров. Когда он включен значением **On**, MapInfo Pro использует 32-разрядные цвета пикселей курсоров. Когда он включен значением **Off**, MapInfo Pro отображает черно-белые курсоры.

Примечание: Изменения, заданные командами MapBasic, не являются постоянными, они используются в текущем сеансе.

Стиль курсоров задаётся в диалоге **Системные настройки** MapInfo Pro.

Пример:

В следующем примере включается режим отображения курсоров в 1-битном цвете каждого пикселя.

```
Set Cursor Truecolor Off
```

Оператор Set Date Window

Назначение

Отображает окно даты, которое переводит двузначный год в четырехзначный. Кроме того можно настроить стандартный вариант преобразования даты, наиболее удобным способом. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Date Window { nYear | Off }
```

nYear – это короткое целое, от 0 до 99, которое определяет число, значения превышающие это число будут отнесены к 20-му столетию (19xx), а числа меньшие заданного будут отнесены к 21-му столетию (20xx). (Например, если *nYear* равно 70, двузначный год от 70-ти и выше

будет соответствовать годам с 1970 по 1999, а двузначный год от 69-ти и ниже – с 2000 по 2069.)

Off отключает окно даты. Двузначные годы будут конвертироваться в текущее столетие (по системному календарю и времени).

Описание

Запуск команды Set Date Window из окна **MapBasic** изменит поведение дат, но не изменит системные настройки даты в MapInfo Pro. При запуске команды **Set Date Window** из программы **MapBasic**, поведение дат изменится только локально, настройки дат в системе не изменятся.

Команда **Set Date Window**, запущенная в окне **MapBasic** любого рабочего набора, включая STARTUP.WOR, или любого картографического приложения, выполняющего эту команду посредством интерфейса приложения MapInfo, изменит настройки текущего сеанса.

Когда команда **Set Date Window** запускается из программы MapBasic (также как **оператор Run Command**), новые настройки распространяются только на локальное содержимое программы. Настройки текущего сеанса и системные настройки не будут изменены. Локальный контекст программы визуализируется на основе настроек сеанса. Нумерация и датирование работает аналогично. Они действуют для программы, если программа запущена, в противном случае, они изменяют общие настройки.

Введите число от 0 до 99. Ниже будет указано, в какой четырехзначный формат будет переведена дата.

Например, если вы вводите число 50, то ниже будет указано:

Годы 00-49 станут 2000-2049.

Годы 50-99 станут 1950-1999.

Пример

В следующем примере переменная Date1 = 19890120, Date2 = 20101203 и MyYear = 1990.

```
DIM Date1, Date2 as Date
DIM MyYear As Integer
Set Format Date "US"
Set Date Window 75
Date1 = StringToDate("1/20/89")
Date2 = StringToDate("12/3/10")
MyYear = Year("12/30/90")
```

См. также:

функция DateWindow()

Оператор Set Datum Transform Version

Назначение

С помощью этого оператора можно выбирать старый или новый вариант преобразования топоцентрической системы координат (датума). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Datum Transform Version version_number
```

version_number – целое число. Если *version_number* равно или больше 800, будут использоваться новые алгоритмы преобразования датумов. Иначе, используются старые алгоритмы.

Описание

По умолчанию, MapInfo Pro использует новые алгоритмы преобразования топоцентрических систем координат (датумов). Эти алгоритмы лучше согласуются с используемыми в других программах и не следует отказываться от их применения. В предыдущих версиях MapInfo Pro использовались алгоритмы, оптимизированные по скорости выполнения, и результаты слегка отличались от полученных в других программах

Оператор Set Designer Frame

Назначение

Преобразует либо обычную рамку отчёта в пустую рамку, либо пустую рамку в рамку с картой или со списком, выбирая существующее окно **Карта** или **Список**. Смотрите [Создание пустых рамок](#) на стр. 267 о подробностях создания пустых рамок.

Синтаксис

```
Set Designer Frame  
[ Window layout_window_id ]  
Id { empty_frame_id }
```

```
[ From Window map_or_browser_window_id [ Legends On | Off ] ]
[ Clear ]
```

window_id – идентификатор окна **Отчёт**, целое число;

frame_id – число, указывающее какую рамку в окне **Конструктор отчёта** требуется опросить.

Map_or_browser_window_id – целочисленный идентификатор окна **Карта** или **Список**, которое требуется вставить в пустую рамку.

Описание

Если вслед за идентификатором **Id**, который указывает на пустую рамку, используется параметр **From Window**, то пустая рамка будет заполнена копией окна **Карта** или **Список** с идентификатором *map_or_browser_window_id*. Содержимое карт будет подогнано таким образом, чтобы содержимое копии в рамке поместилось в рамке. Содержимое списка будет обрезано, чтобы оно поместилось в рамке.

Параметр **From Window** может содержать дополнительный параметр **Legends**. Укажите параметр **Legends On**, если карта, копия которой вставляется в рамку, содержит легенду, чтобы в карте окна **Отчёт** присутствовала легенда.

Параметр **Clear** очищает рамку отчёта, превращая её в пустую рамку. После выполнения такого оператора остаётся пустая рамка того же размера и в том же месте, где была заполненная рамка, присвоенное которой имя сохраняется. Всё содержимое рамки (списки, карты, легенды, фигуры, изображения и текст) может быть удалено с помощью этой команды, которую можно применять к уже пустой рамке. После очистки активной рамки с картой или списком (верхнее окно), родительское окно **Отчёт** становится верхним окном. Свойства рамки, например, стиль оформления границы, штриховки или параметр *z-order* остаются без изменения, кроме рамок, содержащих фигуры. Стили оформления линий и штриховок фигур применяются самим к фигурам, поэтому они не применяются к рамкам, содержащим фигуры. Поэтому, когда удаляются фигуры, параметры со стилями оформления линий и штриховок для фигур теряются. Сообщение об ошибке, когда рамка не может быть очищена: 1699 "Не удалось очистить рамку отчёта ^0."

Оператор Set Designer Legend

Назначение

Обновляет окно **Легенда**, с возможностью перестройки порядка разделов. (Чтобы изменить размер, положение или заголовок окна **Легенда**, выполните **оператор Set Window**.) Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Set Designer Legend
[ Window legend_window_id ]
[ Refresh ]
[ Portrait | Landscape ]
```

legend_window_id – целочисленный идентификатор окна, который возвращают **функция** **FrontWindow()** и **функция** **WindowID()**.

Синтаксис 2

```
Set Designer Legend
[ Antialias { On | Off } ]
```

Описание

Оператор **Set DesignerLegend** позволяет перерисовывать окно конструктора легенды, а также выстраивать разделы в нем горизонтально или вертикально.

Если предложение **Window** не определено, MapInfo Pro будет использовать самое верхнее окно **Легенда**.

Слово **Refresh** обновляет окно **Легенда**.

Слово **Portrait** выстраивает разделы легенды вдоль левого края окна **Легенда**, а **Landscape** – вдоль нижнего.

Параметр **Antialias** включает и выключает на карте сглаживание растровых символов. Символы теперь отображаются как вектор. Стили линий и полигонов по-прежнему отображаются как растр, но теперь уже без сглаживания, поэтому не выглядят размытыми. По умолчанию параметр сглаживания в операторе Set Designer Legend выключен, находится в состоянии off.

Примеры

Оператор обновляет окно **Легенда**.

```
Set Designer Legend Window 123432 Refresh
```

Оператор выстраивает разделы легенды вдоль ее верхнего края.

```
Set Designer Legend Window 123432 Landscape
```

Оператор выстраивает разделы легенды вдоль ее левого края.

```
Set Designer Legend Window 123432 Portrait
```

Оператор обновляет окно легенды и выстраивает ее разделы вдоль левого края.

```
Set Designer Legend Window 123432 Refresh Portrait
```

Этот оператор позволяет включить режим сглаживания во всех уже существующих и вновь создаваемых разделах окна **Легенда**

```
Set Designer Legend Antialias On
```

Этот оператор выключает сглаживание.

```
Set Designer Legend Antialias Off
```

Примечание: Для того чтобы изменить стандартное состояние параметра обработки легенд, либо внесите MapBasic-команду в рабочий набор `startup.wor`, либо выполните эту команду в окне **MapBasic**.

См. также:

[оператор Add Designer Frame](#), [оператор Alter Designer Frame](#), [оператор Create Designer Legend](#), [оператор Remove Designer Frame](#)

Оператор Set Digitizer

Назначение

Назначает координаты для оцифровки изображения с бумажной карты. Также включает и выключает режим дигитайзера. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Set Digitizer
( mapx1, mapy1 ) ( tabletx1, tablety1 ) [ Label name ] ,
( mapx2, mapy2 ) ( tabletx2, tablety2 ) [ Label name ]
[ , ... ]
CoordSys...
[ Units... ]
[ Width tabletwidth ]
[ Height tabletheight ]
[ Resolution xresolution, yresolution ]
[ Button click_button_num, double_click_button_num ]
[ Mode { On | Off } ]
```

Синтаксис 2

```
Set Digitizer Mode { On | Off }
```

mapx – расположение относительно Запада и Востока на бумажной Карте;

mapy – расположение относительно Севера и Юга на бумажной Карте;

tablet_x – X-координата на планшете, соответствующая *mapx*;

tablet_y – Y-координата на планшете, соответствующая *mapy*;

name – имя контрольной точки;

Параметр **CoordSys** определяет систему координат, используемую на карте.

click_button_num – номер кнопки, которая симулирует один щелчок мыши;

double_click_button_num – номер кнопки, которая симулирует двойной щелчок мыши.

Описание

Оператор **Set Digitizer** используется для настройки планшета дигитайзера. Параметры оператора **Set Digitizer** соответствуют режимам и данным, которые пользователь MapInfo Pro может задать при помощи диалогового окна команды Карта > Настройка дигитайзера. Все измерения проводятся в заданных пользователем единицах измерения бумажной карты. Оператор **Set Digitizer** не настраивает другие (системные) режимы работы с дигитайзером, такие как порты и скорость связи. Пользователь должен их настроить вне MapBasic и MapInfo.

Оператор **Set Digitizer** позволяет задать MapInfo Pro координатную систему, используемую на бумажной карте, а также две или более контрольных точек. Каждая контрольная точка определяет соответствие координатной пары карты (долгота и широта) координатной паре на планшете. Каждая пара координат дигитайзера (планшета) соответствует положению на планшете заданной пары координат карты. Координаты на планшете представляются в единицах измерения дигитайзера, измеряемых от левого верхнего угла планшета.

Параметр **CoordSys** определяет систему координат, используемую на бумажной карте. Подробнее смотрите описание параметра **CoordSys**.

Примечание: Оператор **Set Digitizer** игнорирует предложение **Bounds** параметра **CoordSys**.

Предложения **Width**, **Height** и **Resolution** предназначены только внутреннего использования в MapInfo Pro. MapInfo Pro может сохранять настройку дигитайзера для следующих сеансов работы в файле рабочего набора. Программы MapBasic не требуют указывать эти предложения.

Режим оцифровки

Если дигитайзер настроен, пользователь может включать и выключать режим оцифровки, нажимая на клавишу “D”. Этот режим может переключать и программа MapBasic оператором

```
Set Digitizer Mode On
```

или

```
Set Digitizer Mode Off
```

Для определения включения режима оцифровки используется функция **SystemInfo**(`(SYS_INFO_DIG_MODE)`), которая возвращает “Да” (TRUE), если режим установлен.

При включенном режиме оцифровки в активном окне карты помимо указателя мышки появляется также курсор дигитайзера в виде большого перекрестия.

Если отключен режим оцифровки или окно карты неактивно, курсор дигитайзера не показывается и панель дигитайзера начинает работать как мышь (если дигитайзер поддерживает режим эмуляции мыши).

См. также:

[предложение CoordSys](#), [функция SystemInfo\(\)](#)

Оператор Set Distance Units

Назначение

Устанавливает единицы измерения расстояний, используемые в географических операторах, таких как [оператор Create Object](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Distance Units unit_name
```

unit_name – наименование единиц расстояния (например, “m” для метров).

Описание

Оператор **Set Distance Units** устанавливает единицы линейных измерений. По умолчанию MapBasic использует мили (“mi”), то есть, если в Вашей программе нет оператора **Set Distance Units**, единицами измерения расстояния будут мили. В следующих операторах и функциях будет использоваться установленная единица измерения расстояний, если единицы измерения

специально не определяются в самих операторах или функциях. Например оператор Create Object с параметром **Width** может либо задавать единицы измерения расстояний, либо нет. Если в параметре **Width** не указаны единицы расстояний оператор использует текущую установку (мили или любые другие, используемые последним оператором **Set Distance Units**).

Параметр *unit_name* должен являться одним из кодов, показанных в следующей таблице.

Значение unit_name	Представляет
"ch"	чейны
"cm"	сантиметры
"ft"	футы (иногда называется "международным футом"; один "международный фут" равен точно 30,48 см)
"in"	дюймы
"km"	километры
"li"	линки
"m"	метры
"mi"	мили
"mm"	миллиметры
"nmi"	морские мили (1 морская миля = 1852 метрам)
"rd"	роды
"survey ft"	геодезический фут США геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам) геодезический фут равен 12/39.37 метрам или, приблизительно, 30,48006 сантиметрам)
"yd"	ярды

Пример

```
Set Distance Units "km"
```

См. также:

функция `Distance( )`, функция `ObjectLen( )`, оператор `Set Area Units`, оператор `Set Paper Units`

Оператор Set Drag Threshold

Назначение

Назначает временную задержку, необходимую для фиксации мыши на перемещаемом объекте. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Drag Threshold pause
```

pause – вещественное число, задающее задержку в секундах (по умолчанию – 1.0).

Описание

Когда пользователь указывает на объект, оставляя клавишу мышки нажатой, MapInfo Pro требует выдержать небольшую паузу. Эта задержка предотвращает случайное перемещение объекта. Оператор **Set Drag Threshold** назначает продолжительность этой задержки.

Пример

```
Set Drag Threshold 0.25
```

Оператор Set Event Processing

Назначение

Позволяет временно отключить реакцию на системные события и избегать лишних перерисовок экрана. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Event Processing { On | Off }
```

Описание

Оператор **Set Event Processing** позволяет временно отключать реакцию на системные события и тем самым избежать ненужной перерисовки содержимого экрана. Тот же оператор позволяет затем снова включить реакцию на системные события.

Несколько последовательных команд могут изменять окно, что сопровождается перерисовкой его содержимого. Такие множественные перерисовки окна нежелательны, поскольку вынуждают пользователя тратить время на ожидание. Чтобы сэкономить время, пользователь может отключить перерисовку окон, применив оператор:

```
Set Event Processing Off
```

А после того, как все операторы, изменяющие окно (например **оператор Set Map**), отработают, включить перерисовку снова:

```
Set Event Processing On
```

Каждый оператор **Set Event Processing Off** должен иметь парный **Set Event Processing On**. Если, работая в многозадачных системах (например, Windows или System 7), Вы забудете включить обработку событий обратно, это может повлиять на работу других программ.

Также можно подавлять перерисовку изображения в окне оператором **Set Map...Redraw Off**, действие которого похоже на действие оператора **Set Event Processing Off**. Однако **оператор Set Map** управляет перерисовкой одного окна карты, а действие оператора **Set Event Processing** распространяется на все окна MapInfo.

Оператор Set FastPointRendering

Назначение

Позволяет MapBasic-разработчикам включать и выключать режим быстрого рендеринга. Эта команда переопределяет параметр **Быстрый рендеринг символов** в настройках MapInfo Pro. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set FastPointRendering { On | Off }
```

Описание

Используйте оператор **Set FastPointRendering**, чтобы переопределить параметр **Быстрый рендеринг символов** в настройках MapInfo Pro. Пользователи MapInfo Pro могут включать

или выключать этот режим на вкладке **PRO** в разделе **Настройки**, выполнив команду **Параметры вывода**. Этот флажок системного параметра называется **Быстрый рендеринг символов**. Оператор **Set FastPointRendering** не меняет значение системного параметра настройки MapInfo Pro.

Оператор Set File Timeout

Назначение

Предписывает MapInfo Pro повторять попытку доступа к файлу после сетевого конфликта. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set File Timeout n
```

n – число от нуля и больше, задающее ожидание в секундах.

Описание

Обычно, если операция не может быть продолжена из-за конфликта в сети, MapInfo Pro показывает диалог типа **Повторить/Отменить**. Если программа MapBasic выполнит оператор **Set File Timeout**, то MapInfo Pro вместо вывода диалога будет автоматически повторять попытки открыть файл, доступ к которому в сети запрещен.

Если число *n* больше нуля, то через каждые *n* секунд MapInfo Pro делает очередную попытку открыть файл. В любой момент, когда пользователь попытается прочитать таблицу, доступа к которой нет (например, таблица сохраняется другим пользователем), MapInfo Pro повторяет попытки получить доступ к таблице. Если через *n* секунд таблица все еще не доступна, MapInfo Pro показывает диалог **Повторить/Отмена** с соответствующим сообщением. Обратите внимание, что диалог **Повторить/Отмена** нельзя перехватить и обработать средствами обработки ошибок MapBasic.

Если *n*=0, повторные попытки обработки отключены. MapInfo Pro демонстрирует диалог **Повторить/Отмена** немедленно, как только обнаруживает, что таблица недоступна.

Не используйте одновременно оператор **Set File Timeout** и обработчик ошибок OnError. Там, где обработка ошибок разрешена, значение задержки должно быть равно нулю.

Когда значение задержки обращения в файлу не равно нулю, обработка ошибок должна быть заблокирована. Для получения другой информации об интегрированной Более подробно возможные конфликты в сети описаны в *Руководстве пользователя MapBasic*.

Пример

```
Set File Timeout 100
```

Оператор Set Format

Назначение

Определяет способ, которым MapBasic составляет строки из численных значений и значений даты и времени. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1

```
Set Format Date { "US" | "Local" }
```

Синтаксис 2

```
Set Format Number { "9,999.9" | "Local" }
```

Описание

Пользователь может установить разные форматы для даты и чисел в своем компьютере. Например, пользователь Windows может изменить формат даты, используя панель инструментов Windows.

Некоторые функции MapBasic, такие как [функция Str\\$\(\)](#), используют эти системные установки. Другими словами, некоторые функции могут иметь разные результаты, так как выполнялись в компьютерах с разными системными конфигурациями.

Оператор **Set Format** заставляет программу MapBasic игнорировать внешние установки так, чтобы функции (такие как [функция Str\\$\(\)](#)) вели себя предсказуемо.

Оператор	Эффект в программе MapBasic
<code>Set Format Date "US"</code>	MapBasic использует форму Месяц/День/Год для представления даты вместо внешней установки компьютера пользователя.
<code>Set Format Date "Local"</code>	MapBasic использует форму для представления даты согласно установке компьютера пользователя.
<code>Set Format Number "9,999.9"</code>	Функция Format\$() использует установку для форматирования числа, принятую в США. (десятичной точкой является точка, а разделителем тысяч – запятая) вместо внешней установки компьютера пользователя.
<code>Set Format Number "Local"</code>	Функция Format\$() использует форму для представления чисел согласно установке компьютера пользователя.

Первый вариант синтаксиса (**Set Format Date**) предназначен для следующих случаев: вызывается [функция StringToDate\(\)](#); дата передаётся [функции Str\\$\(\)](#); выполнение операций в MapBasic, ведущих к автоматическому конвертированию строковых значений и значений дат (например, [оператор Print](#) для печати даты или при присваивании переменной строкового типа величины типа Date).

Второй вариант синтаксиса (**Set Format Number**) используется, когда применяются функции [Format\\$\(\)](#) и [FormatNumber\\$\(\)](#). Программы, откомпилированные MapBasic 3.0 и более ранней версии, по умолчанию используют стандарт США. Программы, откомпилированные в MapBasic версии 4.0 и в более поздних версиях по умолчанию используют местные параметры форматирования. Для определения установленного режима форматирования, используется [функция SystemInfo\(\)](#). Установка оператора **Set Format**, сделанная в одной программе MapBasic, не влияют на другие выполняющие программы.

Пример

Пусть переменная типа даты `date_var` содержит "June 11, 1995". Функция:

```
Str$( date_var )
```

может возвращать как "11.06.95", так и "95.11.06", в зависимости от того, какой формат использует пользователь на своем компьютере. Если вставить в текст программы оператор

Set Format Date "US" перед местом, где используется **функция Str\$()**, то **функция Str\$()** будет использовать формат M/D/YY, что сделает результат предсказуемым.

См. также:

функция Format\$(), **функция FormatNumber\$()**, **функция Str\$()**, **функция StringToDate()**, **функция SystemInfo()**

Оператор Set Handler

Назначение

Включает и выключает вызов обработчика системных событий, вызов такой процедуры как **процедура SelChangedHandler**.

Предупреждение!

Этот оператор нельзя запускать из окна **MapBasic**.

Синтаксис

```
Set Handler handler_name { On | Off }
```

handler_name – имя процедуры обработчика, такое как **процедура SelChangedHandler**.

Описание

Обычно, если программа имеет в своем теле процедуры-обработчики, MapInfo Pro вызывает их автоматически, как только происходит соответствующее событие. Например, если в программе содержится **процедура SelChangedHandler**, MapInfo Pro вызовет ее автоматически, как только произойдет изменение в выборке.

Оператор **Set Handler** может отключить автоматический вызов определенного обработчика в MapBasic-программе и включить его снова.

Оператор **Set Handler...Off** имеет эффект только для автоматических вызовов; на вызовы, которые делает **оператор Call**), он не влияет.

Пример

Следующий пример показывает, как обойти лишние запуски обработчика изменения выбора при помощи оператора **Set Handler**.

```
Sub SelChangedHandler
  Set Handler SelChangedHandler Off

  ' Issuing a Select statement here
  ' will not cause an infinite loop.

  Set Handler SelChangedHandler On
End Sub
```

См. также:

процедура **SelChangedHandler**, процедура **ToolHandler**

Оператор Set Layout

Назначение

Изменяет существующее окно отчета. Этот оператор можно использовать и с классическим окном отчёта, и с окном **Отчёт**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Layout
[ Window window_id ]
[ Page page_number ]
[ Center ( center_x, center_y ) ]
[ Extents { To Fit | ( pages_across , pages_down ) } ]
[ Pagebreaks { On | Off } ]
[ Frame Contents { Active | On | Off } ]
[ Ruler { On | Off } ]
[ Zoom { To Fit | zoom_percent } ]
[ { Objects Alpha alpha_value } |
  { Objects Translucency translucency_percent } ]
[ Selection { All | Clear | [ Add | Remove ] Frame ID frame_id, frame_id,
frame_id,... } ]
[ Grid [ Visibility { On | Off } ]
[ Snap { On | Off } ]
[ Size distance { Units paper_unit } ] ]
[ Smart Guides { On | Off } ]
```

window_id — идентификатор окна отчета, целое число.

Page_number — номер страницы, на которую требуется перейти (1 или больше).

center_x — горизонтальная координата центра окна Отчета.

Center_y — вертикальная координата центра окна отчёта.

pages_across — число страниц (одна или более), расположенных подряд по горизонтали.

Pages_down — число страниц (одна или более), расположенных подряд по вертикали.

zoom_percent — процентное соотношение размера отображения Отчета к реальному размеру.

alpha_value — целое число, представляющее значение альфа-канала для полупрозрачности. Значения изменяются в диапазоне от 0 (полная прозрачность) до 255 (полная непрозрачность). Промежуточные значения представляют градации полупрозрачности.

translucency_percent — целое число, представляющее значение полупрозрачности для объекта Отчёта. Значения в диапазоне между 0-100. 0 — полностью непрозрачные. 100 — это полная прозрачность.

Примечание: Задавайте либо режим **Alpha**, либо **Translucency**, но не оба, так как это разные способы достижения одного результата. Если указано несколько таких параметров, будет использоваться последний.

frame_id — идентификатор раздела легенды. Например, три рамки в отчёте будут иметь последовательные идентификаторы — 1, 2 и 3. Если идентификатор выходит за пределы диапазона, то появляется сообщение об ошибке "Неверный id рамки n — Invalid frame id n", где число n — идентификатор ID переданный в команде. Если идентификатор является дубликатом, появится сообщение об ошибке: "Повторяющийся id рамки n — Duplicate frame id n", где n — номер идентификатора, переданный командой. Если рамка уже выбрана, то сообщение об ошибке не показывается.

Distance — размер интервала между линиями сетки в окне отчёта.

paper_unit — дополнительное, необязательное значение, которое определяет единицы измерения, в которых задаётся расстояние *distance*. В качестве значения параметра *paper_unit* используется завышенная аббревиатура принятых в MapInfo Pro стандартных "бумажных" единиц измерения (например, "in"). Если значение не задано, то, по умолчанию, используются текущие "бумажные" единицы измерения. Если заданная единица измерения не является "бумажной", то выполнение оператора приводит к ошибке.

Описание

Оператор **Set Layout** управляет настройкой отображения содержимого окна Отчета. Он работает с обоими типами окон: и с классическими окнами отчёта, и с окнами **Отчёт**. Однако, в окнах **Отчёт** пропускаются все параметры кроме: **Center**, **Zoom** *zoom_percent* и **Selection**. Параметр **Zoom To Fit** используется только с классическими окнами **Отчёт**. О фоне классических окон **Отчёт** смотрите раздел: [Работа с классическим окном отчёта](#).

Если параметр *window_id* не задан, то действия оператора распространяются на самое верхнее из открытых окон Отчета. Этот оператор позволяет программе управлять отображением данных в окне отчёта так же, как это делает пользователь при помощи команд из меню Отчет.

Используйте предложение **Page**, чтобы задать текущую страницу в многостраничном окне **Отчёт**. Некорректный номер страницы вызывает сообщение об ошибке ERR_FCN_ARG_RANGE (644). Корректными являются номера от 1 и выше, при условии что страница существует.

Предложение **Center** задает точку, которая будет центральной в окне отчёта.

Предложение **Extents** управляет количеством страниц отчёта. Следующий вариант оператора:

```
Set Layout Extents To Fit
```

определяет, на скольких страницах установленного формата для принтера, который на данный момент подключен к системе, смогут поместиться все существующие объекты отчёта. После слова **Extents** можно также назначать и определенное количество печатных страниц. Например, следующий оператор задает шесть страниц, по три в два ряда:

```
Set Layout Extents (3, 2)
```

Если отчёт состоит более чем из одной страницы, то переключатель **Pagebreaks** позволяет управлять отображением линий, разделяющих изображение на страницы. По умолчанию используется режим On, когда MapInfo Pro показывает границы страниц линиями из точек.

Предложение **Frame Contents** управляет режимом обновления содержимого рамок. Отчёт может содержать от одного или более таких объектов, отображающих изображения из других открытых окон в MapInfo Pro (карт, графиков, легенд и т. п.). При изменении данных в окне, на базе которого был создан отчет, можно выбирать следует ли MapInfo Pro тратить время на перерисовывание окна отчёта. Некоторые пользователи предпочитают всегда видеть текущее состояние данных окна. Но поскольку эта операция требует времени, некоторые пользователи могут хотеть, чтобы окно отчёта перерисовывалось только когда оно активно.

Следующий оператор задает программе MapInfo Pro синхронную перерисовку в окне отчёта:

```
Set Layout Frame Contents On
```

Перерисовка может происходить только тогда, когда окно отчёта становится активным. Для этого используется следующий режим:

```
Set Layout Frame Contents Active
```

Для того, чтобы программа MapInfo Pro вообще не обновляла изображение в окне отчёта, используйте следующий вариант:

```
Set Layout Frame Contents Off
```

В случае, когда параметр **Frame Contents** установлен в режим **Off**, все объекты типа "рамка" показываются в окне отчета в виде закрашенных прямоугольников с именами соответствующих окон (например, "РОССИЯ Карта").

Предложение **Ruler** включает режим отображения линеек сверху и слева в окне Отчета. По умолчанию, для параметра **Ruler** задан режим **On**.

Предложение **Zoom** определяет отношение реальных размеров отчета и представленного на экране макета. Например, следующее предложение задает пятидесятипроцентный масштаб макета:

```
Set Layout Zoom 50.0
```

Другими словами, размеры изображения на экране будут ровно в два раза меньше реальных размеров отчёта. Помните, что страница отчета может иметь очень большие размеры в целях обеспечения высокой точности работы. Соответственно, окно отчёта, отображаемое на 200 процентов, покажет увеличенную страницу. Предложение **Zoom** позволяет задавать коэффициент масштаба в любых пределах от 6.25% до 800% для классических, традиционных окон отчётов или от 5% до 400% для окон конструктора отчёта. В предложении **Zoom** можно не только задавать определённое значение процентов, а потребовать, чтобы все страницы поместились на экране:

```
Set Layout Zoom To Fit
```

С помощью параметра **Selection** можно управлять выбором рамок только в окне отчёта. Если *window_id* указывает на классическое окно отчёта, то параметр **Selection** будет скомпилирован, но при выполнении появится следующее сообщение об ошибке: "Error: Expecting Layout Window".

- **All** — все рамки отчёта. Сообщения об ошибках не появятся, если все рамки уже выбраны или в отчёте ещё нет рамок.

```
Set Layout Selection All
```

- **Clear** — отменяет выбор всех рамок отчёта. Сообщения об ошибках не появятся, если ни одна рамка не выбрана или в отчёте нет рамок.

```
Set Layout Selection Clear
```

- **Add** — является дополнительным, необязательным параметром-токеном, который употребляется перед списком идентификаторов рамок. Если используется параметр **Add**, то все перечисленные рамки будут выбраны и добавлены в список выбранных рамок. Без этого параметра сначала отменяется выбор всех прежде выбранных рамок, а список рамок, перечисленных в этом операторе, становится новой набором выбранных рамок. Сообщение об ошибке не появляется, если рамка, идентификатор на которую есть в списке, уже выбрана.

```
Set Layout Selection Add Frame ID 4 // adds frame with ID 4 to current selection
```

- **Remove** — дополнительный, необязательный параметр-токен перед списком идентификаторов рамок. Идентификаторы, перечисленные в списке параметра, удаляются из текущей выборки. Сообщение об ошибке не появляется, если рамка, идентификатор на

которую есть в списке, уже выбрана. Этот параметр нельзя использовать вместе с оператором **Add**.

```
Set Layout Selection Remove Frame ID 3 // frame with ID 3 is removed
from the current selection
```

С помощью параметра **Grid** можно управлять линиями сетки в окне отчёта.

- Ключевое слово **Visibility** позволяет включать и выключать видимость линий сетки в окне отчёта. Если ключевое слово не задано, то, по умолчанию, используется включенное состояние видимости.
- Ключевое слово **Snap** включает и выключает возможность привязки, совмещения объектов по линиям сетки. Если ключевое слово не задано, то, по умолчанию, используется выключенное состояние привязки.
- Ключевое слово **Size** позволяет управлять размером интервала между линиями сетки в окне отчёта.

Предложение **Smart Guides** по умолчанию выключено. Предложение "Smart Guides" позволяет управлять включением и выключением режима динамических направляющих. Если токен Smart Guides отсутствует, то это будет синтаксической ошибкой. Если пропущены токены **On** или **Off**, то по умолчанию используется вариант **On**. Скрытые, динамические направляющие используются, для того чтобы выравнивать кромку перемещаемой рамки относительно либо кромок других невыбранных на странице рамок, либо относительно границ страницы.

Примечание: Если в окне выбран объект – "рамка", то в MapBasic-программе можно использовать **оператор Run Menu Command** для выполнения команд Достать наверх или Подложить вниз. Также, поскольку объекты рамок (в некотором смысле) являются обычными графическими объектами MapInfo Pro, то в приложении на MapBasic **оператор Alter Object** позволяет изменять стили линий и штриховок, порядок наложения объектов рамок.

Для изменения таких параметров окна Отчёта, как ширина окна, высота и расположение его на экране, используйте **оператор Set Window**.

Примеры

Следующее делает первую страницу (1) текущей в многостраничном окне **Отчёт**.

```
Set Layout Window FrontWindow( ) Page 1
```

В следующем примере показывается, как добиться того, что все объекты отчёта будут напечатаны. Также прячутся линейки классического окна отчёта.

```
Set Layout
Zoom To Fit Extents To Fit
Ruler Off
Frame Contents On
```

Следующие операторы устанавливают уровень приближения окна **Отчёт** в 200% и меняют центр макета отчёта.

```
Set CoordSys Layout Units "in"
Set Layout Window FrontWindow() Center (4.25, 5.5) Zoom 200
```

См. также:

[оператор Alter Object](#), [оператор Create Frame](#), [оператор Layout](#), [оператор Run Menu Command](#), [оператор Set Window](#)

Оператор Set Legend

Назначение

Изменяет окно легенды тематической карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Legend
[ Window window_id ]
[ Layer { layer_id | layer_name | Prev }
  [ Display { On | Off } ]
  [ Shades { On | Off } ]
  [ Symbols { On | Off } ]
  [ Lines { On | Off } ]
  [ Count { On | Off } ]
  [ Title { Auto | layer_title [ Font... ] } ]
  [ SubTitle { Auto | layer_subtitle [ Font... ] } ]
  [ Region [ Height region_height [ Units paper_units ] ] ]
  [ Region [ Width region_width [ Units paper_units ] ] ]
  [ Line [ Width line_width [ Units paper_units ] ] ]
  [ Auto Font Size { On | Off } ]
  [ Style Size { Large | Small } ]
  [ Columns number_of_columns ]
  [ Ascending { On | Off } | Order { Ascending | Descending |
    Custom } ]
  [ Ranges { Auto | [ Font... ]
    [ Range { range_identifier | default } ]
    range_title [ Display { On | Off } ] ]
  [ , ... ] ] ]
[ , ... ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя Карты, короткое целое число (SmallInt);

layer_name – строка, определяющая слой Карты.

layer_title, *layer_subtitle* – строки заголовка и подзаголовка легенды тематической карты;

number_of_columns – значение, задающее ширину колонки.

region_height – новая высота образца для картографического раздела окна **Легенда**. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров. Если значение не задано, то используется значение 32 пункта (может быть задано как стандартное).

region_width – новая ширина образца для картографического раздела окна **Легенда**. Можно задать от 8 до 144 пунктов, от 0.666667 до 12 пик, от 0.111111 до 2 дюймов, от 2.82222 до 50.8 миллиметров или от 0.282222 до 5.08 сантиметров. Если значение не задано, то используется значение 32 пункта (может быть задано как стандартное).

line_width – новая ширина линий для картографического раздела окна **Легенда**. Можно задать от 12 до 144 пунктов, от 1 до 12 пик, от 0.666667 до 2 дюймов, от 4.23333 до 50.8 миллиметров или от 4.23333 до 50.8 сантиметров. Если значение не задано, то используется значение 36 пункта (может быть задано как стандартное).

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также цистеро.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

range_title – строка, описывающая выделенный диапазон значений слоя.

Описание

Оператор **Set Legend** задает настройки для окна легенды. Этот оператор позволяет программе управлять отображением условных обозначений. Для изменения таких параметров окна легенды, как ширина окна, высота и расположение его на экране [оператор Set Window](#).

Оператор **Set Legend** может использоваться в файле рабочего набора. Для примера оператора **Set Legend** можно открыть окно карты, создать условное выделение, открыть окно легенды и сохранить рабочий набор (например, под именем `LEGEND.WOR`). Теперь откройте файл рабочего набора в любом текстовом редакторе и вы увидите оператор **Set Legend**, задающий те настройки, которые были использованы в открытом ранее окне легенды.

Независимо от количества открытых окон карт, на экране может присутствовать только одно окно легенды, относящееся к одному окну карты. Окно легенды отображает информацию о карте. В операторе **Set Legend** параметр *window_id* задает окно карты, а не легенды. Если параметр *window_id* не задан, то оператор использует самое верхнее окно карты.

Предложение **Layer** задает изменения в описании слоя в легенде. Слой идентифицируется значением параметра **Layer** либо своим порядковым номером в окне карты, либо именем, либо задается словами **Layer Prev**. Предложение **Layer Prev** идентифицирует слой, который был создан или изменен последним, для чего используются оператор **Set Shade** или оператор **Shade**.

Если карта содержит два или более тематических слоев, то оператор **Set Legend** может содержать столько же предложений **Layer**, по каждому на один тематический слой.

Оставшиеся ключевые слова в операторе **Set Legend** составляют предложение **Layer**, то есть описываемые ниже предложения являются ключевыми словами в предложении **Layer**.

Предложение **Count** определяет показ в скобках количества записей, принадлежащих к данному диапазону. Предложения **Shades**, **Symbols** и **Lines** задают показ в строке легенды элементов оформления, соответствующих диапазонам. Если оператор включает предложение **Shades On**, то в окне легенды будут показаны образцы штриховок, используемых Картой. Если оператор включает предложение **Symbols On**, то в окне легенды будут показаны образцы символов точечных объектов. Если оператор включает предложение **Lines On**, то в окне легенды будут показаны образцы линий и контуров.

Предложения **Title**, если оно задано, и **Subtitle** управляют показом заголовка и подзаголовка соответственно. Каждая из этих строк не должна быть длиннее 32 символов. В окне легенды строка заголовка всегда располагается сверху, строка подзаголовка ниже, а затем идут описания диапазонов. Если используется предложение **Auto**, текст генерируется автоматически.

Параметром этого оператора **Font** можно задать стиль оформления текста.

Предложение **Columns** позволяет задавать ширину легенды.

Предложение **Region Height** определяет высоту раздела с образцами в окне **Легенда**.

Предложение **Region Width** определяет ширину раздела с образцами в окне **Легенда**.

Предложение **Line Width** определяет ширину образца линии в разделе образцов в окне **Легенда**.

Auto Font Size контролирует режим подстройки размера шрифтов в окне образцов при изменении размера этого окна.

Если не задать предложение **Ascending On**, то строки легенды, описывающие диапазоны, будут располагаться в стандартном, возрастающем порядке. Иначе они расположатся в убывающем порядке.

Предложение **Ranges** задает текстовое описание диапазона (*range_title*), скомбинированное с режимом, задаваемым предложением **Display**. Необходимо задать параметр *range_title* для каждого диапазона, а управлять его представлением можно предложениями *Display On* или *Display Off*. Предложение **Ranges** должно включать предложение *range_title Display* для каждого диапазона тематической Карты, даже для диапазонов, которые не будут показаны.

Если на некий слой условное выделение производилось методом размерных символов, то понадобится две комбинации *range_title Display*. Если условное выделение производилось методом задания плотности точек, то понадобится задать одну комбинацию *range_title Display*.

В остальных случаях надо будет дополнительно к описаниям диапазонов задать еще одну строку для диапазона "остальные", в который попадают все объекты, не охваченные диапазонами.

Предложения **Order** и **Range** изменяют версию рабочего набора на последнюю. Старые рабочие наборы будут анализироваться корректно, поскольку поддерживаются предложением **Ascending**. Если порядок нетипичный, Mapinfo Pro запишет исходное предложение **Ascending** и не будет обновлять версию рабочего набора.

Предложение **Order** – это еще один способ указать порядок записей (по возрастанию или по убыванию). Исходное предложение **Ascending** поддерживает обратную совместимость. Можно использовать предложение **Order** или **Ascending**, но не оба одновременно. Оба предложения не могут быть одновременно включены в один и тот же оператор MapBasic, поскольку это приведет к ошибке синтаксиса.

Вариант **Custom** для предложения **Order** разрешен только для Индивидуальных значений. Получится ошибка, если применить пользовательский порядок к другим типам тематических карт. Сообщение об ошибке: "Custom legend label order is only allowed for Individual Value themes."

Если в параметре **Order** используется вариант **Custom**, каждый диапазон в предложении **Ranges** должен содержать идентификатор. В противном случае появится ошибка синтаксиса. Идентификатор диапазонов должен идти перед заголовком диапазона и предложением **Display**. Идентификатор диапазонов является той же строкой или значением, которые оператор **Shade** использует в предложении **Values**. Для категории "все остальные" используется идентификатор "по умолчанию".

Должны быть учтены все категории, включая категорию "по умолчанию", в противном случае появится ошибка. Сообщение об ошибке: "Incorrect number of ranges specified for custom order."

Порядок размещения категорий "по умолчанию" и "все остальные" также может быть изменен, независимо от того, что наиболее удачное место расположения этих аргументов в начале или в конце предложения **Ranges**.

Если идентификатор диапазона не относится к правильной категории, произойдет ошибка. Сообщение об ошибке: "Invalid range value for custom order."

Предложение **Style Size** изменяет размер.

Пример

В следующем примере высота образца условного знака меняется на значение в 40 пунктов.

```
Set Legend
Window FrontWindow()
Layer 1 Region Height 40 Units "pt"
```

См. также:

оператор Map, оператор Open Window, оператор Set Map, оператор Set Window, оператор Shade

Настройка порядка перечисления условных знаков легенды

Оператор **Set Legend** позволяет настраивать только тематические легенды, но тематические легенды могут отображаться либо в традиционных окнах **Тематическая легенда**, либо в окнах **Конструктор легенды** или **Конструктор отчёта**. Предложение **Order** в операторе **Set Legend** позволяет менять порядок перечисления строк легенды любой тематической легенды. Можно задавать либо восходящий, либо нисходящий порядок перечисления. Только для тематических легенд индивидуальных значений поддерживается произвольный порядок перечисления.

В качестве примера, на тематическом слое индивидуальных значений (первый слой карты) имеется четыре диапазона, каждая из четырёх строк в легенде: "Высокий (High)", "Умеренный (Moderate)", "Низкий (Low)" и "Очень высокий (Very High)". Для того чтобы расположить диапазоны легенды в нисходящем алфавитном порядке перечисления, MapBasic-оператор должен быть таким:

```
Set Legend Layer 1 Order Descending
```

Для того чтобы настроить порядок диапазонов таким образом, чтобы они следовали от "Очень высокого (Very High)" к "Низкому (Low)", необходимо специально перечислить все диапазоны:

```
Set Legend Layer 1 Order Custom Ranges Range "Very High" auto, Range  
"High" auto, Range "Moderate" auto, Range "Low" auto, Range default auto
```

Последняя часть `Range default auto` попадает в диапазон «Все остальные (All Others)». Порядок перечисления в диапазоне «Все остальные (All Others)» тоже можно менять: он просто будет отображаться в конце.

Оператор Set LibraryServiceInfo

Назначение

Восстанавливает режимы работы библиотечной службы (Library Service). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set LibraryServiceInfo  
{ URL url }
```

url – корректный URL-адрес библиотечного сервиса.

Описание

URL – корректное **предложение URL**, задающее URL-адрес библиотечной службы (Library Service).

Пример

```
Include "mapbasic.def"
declare sub main
Set LibraryServiceInfo URL
"http://localhost:8080/LibraryService/LibraryService"
end sub
```

См. также:

функция LibraryServiceInfo(), **предложение URL**

Оператор Set Map

Назначение

Изменяет существующее окно карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro. Предложения оператора Set Map описываются в отдельных разделах ниже.

Синтаксис

```
Set Map
[ Window window_id ]
[ MAP_BEHAVIOR_CLAUSE ]
[ VIEW_CLAUSE ]
[ LAYER_PROPERTY_CLAUSE ]
[ LABEL_CLAUSE ]
[ STYLE_OVERRIDE_CLAUSE ]
[ LABEL_OVERRIDE_CLAUSE ]
[ GROUPLAYER_PROPERTY_CLAUSE ]
[ ORDER_LAYERS_CLAUSE ]
[ COORDSYS_CLAUSE ]
[ IMAGE_CLAUSE ]
[ LAYER_ACTIVATE_CLAUSE ]
[ LAYER_OVERRIDE_CLAUSE ]
```

window_id – целочисленный идентификатор окна Карты.

MAP_BEHAVIOR_CLAUSE см. **Управление поведением карты**

VIEW_CLAUSE см. **Изменение изображения в окне карты**

LAYER_PROPERTY_CLAUSE см. [Управление представлением и поведением слоев](#)

LABEL_CLAUSE см. [Управление настройками подписей на слое](#)

STYLE_OVERRIDE_CLAUSE см. [Добавление определений единообразного оформления для слоя.](#)

LABEL_OVERRIDE_CLAUSE см. [Добавление единообразного оформления подписей.](#)

GROUPLAYER_PROPERTY_CLAUSE см. [Управление группировкой слоев](#)

ORDER_LAYERS_CLAUSE см. [Задание порядка слоев](#)

COORDSYS_CLAUSE см. [Управление координатной системой карты](#)

IMAGE_CLAUSE см. [Управление растровыми изображениями](#)

LAYER_ACTIVATE_CLAUSE см. [Управление геолинками.](#)

LAYER_OVERRIDE_CLAUSE см. [Добавление определений единообразного оформления для слоя.](#)

Описание

Оператор **Set Map** задает настройки для окна карты. Если параметр *window_id* не задан, то действия оператора распространяются на самое верхнее из открытых окон Карты. Этот оператор позволяет программе управлять отображением слоев карты так же, как это может делать пользователь при помощи команд MapInfo Pro [Управление слоями](#), [Показать по-другому](#) и [Режимы](#). Например оператор **Set Map** позволяет задать слой Карты, который будет доступен для редактирования, и установить масштаб или степень увеличения (zoom) Карты.

Примечание: Заметим, что оператор **Set Map** управляет настройками в окне Карты. Для изменения таких атрибутов, как размер окна и его расположение на экране, используется [оператор Set Window](#).

Оператор **Set Map** может использоваться в файле рабочего набора, если в нем есть хотя бы одно окно карты. Чтобы увидеть, как выглядит оператор **Set Map**, создайте Карту, сохраните Рабочий набор (например, MAPPER.WOR) и изучите его текст в текстовом редакторе, таком как Notepad.

Все предложения оператора **Set Map** не являются обязательными, но хотя бы одно должно присутствовать. При этом важен порядок расположения предложений, несоблюдение порядка может привести к синтаксической ошибке в программе.

См. также:

[оператор Add Map](#), [оператор Map](#), [функция MapperInfo\(\)](#), [оператор Remove Map](#), [оператор Set Window](#), [функция LayerInfo\(\)](#), [функция LayerListInfo\(\)](#), [функция LayerStyleInfo\(\)](#), [функция StyleOverrideInfo\(\)](#), [функция LabelOverrideInfo\(\)](#)

Управление поведением карты

Следующие предложения управляют такими компонентами карты, как единицы измерения, режимы врезки и перерисовки окна.

Синтаксис

```
Set Map
[ Window window_id ]
[ Clipping [ Object clipper ] [ { Off | On } ] ]
[ Using { Display { PolyObj | All } | Overlay } ] ]
[ Preserve { Scale | Zoom } ]
[ Area Units area_unit ]
[ Distance Units dist_unit ]
[ Display { Scale [ Cartographic ] | Position | Zoom } ]
[ Redraw { On | Off | Suspended } ]
[ Move Nodes { value | Default } ]
```

window_id – целочисленный идентификатор окна Карты.

clipper – выражение Object; будет показана только часть Карты, в пределах заданного объекта. Смотрите описание предложения Clipping.

area_unit – строка, представляющая единицы измерения, в которых производятся вычисления на карте (например, “sq mi” для квадратных миль, “sq km” для квадратных километров; см. описание единиц измерения в разделе [Оператор Set Area Units](#)). Например:

```
Set Map Area Units "sq km"
```

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: [Оператор Set Distance Units](#)). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

value – может быть 0 или 1. Если значение равно 0, дублирующиеся узлы не удаляются. Если значение равно 1, все дублирующиеся узлы с одного слоя удаляются.

Описание

Clipping задаёт вырезающий объект для окна карты; соответствует команде MapInfo Pro **Создать врезку** (на вкладке **ОБЪЕКТЫ**). После назначения области врезки можно управлять показом фрагмента-врезки заданием режимов **Clipping On** или **Clipping Off**.

```
Set Map Clipping Object obj_variable_name
```

Для создания врезки используются три метода. Использование режима **Overlay** приводит к применению метода удаления всего, что находится вне объекта (аналогично команде

ОбъектыУдалить снаружи) для создания врезки. Полилинии и регионы будут обрезаться по границе обрезающего полигона. Точки и подписи будут показаны на врезке полностью только в том случае, если они лежат внутри обрезающего полигона. Текст всегда отображается и никогда не обрезается. Стили для всех объектов никогда не обрезаются. Использование метода **Display All**, приводит к созданию врезки системными средствами Windows. Все объекты (включая точки, подписи и текст) будут обрезаться по границе полигона, ограничивающего врезку. Все стили также обрезаются по границе полигона. Это стандартный режим.

Использование метода **Display PolyObj** приводит к созданию врезки системными средствами Windows, причем обрезаются только полилинии и регионы. Стили для полилиний и регионов обрезаются по границе полигона врезки. Точки и подписи будут показаны на врезке полностью только в том случае, если они лежат внутри обрезающего полигона. Текст всегда отображается и никогда не обрезается. Стили для точек, подписи и текст никогда не обрезаются.

В большинстве случаев системные средства Windows из методов **Display All** и **Display PolyObj** обеспечивают лучший результат чем функциональность Overlay. Например:

```
Set Map Clipping Object obj_variable_name Using Display All
```

Display диктует, какой тип информации должен отображаться в строке состояния, когда активно окно карты: **Display Scale** – показывает текущий масштаб в единицах измерения расстояний, **Display Scale Cartographic** – показывает текущий масштаб в "бумажных" единицах измерения (значение отношения 1 к значению масштаба, например, 1:10000), **Display Position** – показывает положение курсора (например, десятичные градусы долготы/широты), а **Display Zoom** – показывает текущий охват карты (ширина показанной области). "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#).

```
Set Map Display Position
```

Предложение **Preserve** определяет поведение Карты при изменении пользователем размеров окна. Предложение **Preserve Zoom** задает показ Карты всегда в одном масштабе, независимо от изменений размеров окна. И, наоборот, **Preserve Scale** задает увеличение или уменьшение масштаба карты в зависимости от увеличения или уменьшения окна, сохраняя в окне карты фрагмент постоянного размера. Эта опция соответствует параметрам, задаваемым в диалоге MapInfo Pro **Параметры окна карты** (на вкладке **PRO** нажмите сначала **Настройки**, а затем выполните команду **Окно карты**).

Предложение **Redraw** управляет автоматической перерисовкой карты в окне. Если программа выполнила оператор **Set Map Redraw Off**, то последующие операторы, изменяющие карту (такие как **Set Map**, **Add Map Layer**, **Remove Map Layer**), будут выполняться без автоматического обновления изображения в окне карты. После внесения всех нужных изменений в окно карты, выполните оператор **Set Map Redraw On**, восстанавливающий режим автоматической перерисовки окна.

Примечание: В окне отобразятся все изменения, которые были выполнены "вслепую". Если оператор **Redraw Off** не предотвращает перерисовку окна, то можно использовать оператор [Set Event Processing Off](#).

Для **Redraw** можно использовать три значения – On, Off и Suspended. При использовании специального значения параметра **Suspended** на карте появится надпись о состоянии окна. Пользователь может отключить автоматическую перерисовку окна, нажав на кнопку внизу диалога Управление слоями.

```
Set Map Redraw Suspended
```

Move Nodes – может принимать значения 0 или 1. Если значение равно 0, дублирующиеся узлы не удаляются. Если значение равно 1, все дублирующиеся узлы с одного слоя удаляются. Если **величина Move Node** задана, то будет использоваться заданное значение. Возвратиться в стандартный режим можно, задав **Move Nodes Default**.

Если в операторе **Set Map Move Nodes** задано нестандартное значение, то для Карты используется пользовательская настройка. Если для окна карты заданы собственные настройки, в частности режим совмещения, то общие настройки не действуют. Общие настройки (стандартные) действуют на все новые окна. Установки для существующих окон могут быть сделаны с помощью оператора **Set Map Move Nodes** языка MapBasic.

Пример

Программа открывает две таблицы и показывает их в окне карты. Затем оператор **Set Map** изменяет окно.

```
Open Table "world"
Open Table "cust1993" As customers
Map From customers, world

Set Map
Center (-100, 40) 'center map over mid-USA
Zoom 4000 Units "mi" 'show entire USA
Preserve Zoom 'preserve zoom when resizing
Display Position 'show lat/long on status bar
```

Изменение изображения в окне карты

Следующие предложения изменяют показ карты в окне, т.е. центральную точку и размеры показываемой в окне области.

Синтаксис

```
Set Map
[ Window window_id ]
[ Center ( longitude, latitude ) [ Smart Redraw ] ]
[ Zoom {
  zoom_distance [ Units dist_unit ] | Entire [ Layer layer_id | Selection ]
| Tileservers Layer layer_id } ]
[ Pan pan_distance [ Units dist_unit ]
```

```
{ North | South | East | West } [ Smart Redraw ] ]
[ Scale screen_dist [ Units dist_unit ] For map_dist
[ Units dist_unit ] ]
```

window_id – целочисленный идентификатор окна Карты.

longitude, latitude – координаты центра Карты;

zoom_distance – числовое выражение, описывающая ширину области показа;

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: **Оператор Set Distance Units**). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

pan_distance – расстояние, на которое нужно сместить Карту;

screen_dist и *map_dist* – задают масштаб Карты (например, screen_dist = 1 дюйм, map_dist = 1 мили).

Описание

Предложение **Center** задает центр карты в окне. Например, город Нью-Йорк расположен приблизительно на 74 долготы и 41 широты. Следующий оператор **Set Map** помещает Нью-Йорк в центр окна Карты. При этом широта и долгота должны задаваться в десятичных единицах, а не в градусах, минутах и секундах.

```
Set Map Center (-74.0, 41.0)
```

Выполнение **Set Map...Center** ведет к полной перерисовке окна карты, если в оператор не включено предложение **Smart Redraw**. Детали о предложении **Smart Redraw** смотрите ниже.

Предложение **Pan** перемещает окно Карты в заданном направлении. Например, следующий оператор перемещает карту на 100 километров к северу:

```
Set Map Pan 100 Units "km" North
```

Обычно, после выполнения оператора **Set Map...Pan** изображение в окне карты перерисовывается полностью. Если в оператор включено предложение **Smart Redraw**, то MapInfo Pro будет обновлять только ту часть изображения, которая этого требует (обновление

изображения в окне Карты будет происходить так, как, если бы пользователь использовал в окне инструмент Сдвиг).

```
Set Map Pan 100 Units "km" North Smart Redraw
```

Осторожно: Если используется предложение Smart Redraw, окно карты передвигается шагами, кратными восьми пикселям. Из-за этого карта может показываться не совсем так, как ожидалось. Например, передвигая карту на север на 100 км, вы добьетесь только передвижения на 80 километров, из-за того, что 20 километров на экране занимают более восьми пикселей.

Предложение **Scale** изменяет масштаб показа карты в окне. Например, следующий оператор изменяет увеличение карты так, чтобы в 1 дюйме изображения было показано 10 миль карты:

```
Set Map Scale 1 Units "in" For 10 Units "mi"
```

Предложение **Zoom** определяет размер в ширину фрагмента карты, показанного в окне. Например, следующий оператор Set Map показывает участок шириной в 100 километров (если текущими единицами расстояний в MapBasic сейчас являются "km"):

```
Set Map Zoom 100 Units "km"
```

Если предложение **Zoom** включает ключевое слово **Entire**, то MapInfo Pro изменяет масштаб показа таким образом, чтобы показать все объекты на слое карты, все выбранные на слое карты объекты или все объекты на всех объектах карты:

В следующем примере будут показаны все объекты на слое 2:

```
Set Map Zoom Entire Layer 2
```

В следующем примере будут показаны все выбранные на слое карты объекты:

```
Set Map Zoom Entire Selection
```

В следующем примере будет показана вся карта:

```
Set Map Zoom Entire
```

Если указать ключевое слово **Selection** в предложении **Zoom**, когда нет выбранных объектов появится сообщение “Нет выбранных объектов.” Когда выбираемый слой не виден, появится диалог с сообщением “Выбираемый слой должен быть видимым.” (Ключевое слово **Selection** также работает со сшитыми таблицами.)

Токен **Tileserver** требует использовать предложение **Layer**.

Примечание: Эта команда

```
Set Map Window FrontWindow() Zoom Tileserver Layer
```

неправильна, поскольку отсутствует *layer_id*.

```
Set Map Window FrontWindow() Zoom Tileserver Layer 1
```

Управление представлением и поведением слоев

Следующие режимы управляют показом и поведением слоев. **Оператор Set Map** необязательно должен содержать эти режимы.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ LAYER_ACTIVATE_CLAUSES ]
  [ Editable { On | Off } ]
  [ Selectable { On | Off } ]
  [ Zoom ( min_zoom, max_zoom )
    [ Units dist_unit ] [ { On | Off } ] ]
  [ Arrows { On | Off } ]
  [ Centroids { On | Off } ]
  [ Default Zoom ]
  [ Nodes { On | Off } ]
  [ Inflect num_inflections [ by percent ] at
    color:value [ , color:value ]
    [ Round rounding_factor ] ]
  [ Contrast contrast_value ]
  [ Brightness brightness_value ]
  [ {Alpha alpha_value } | { Translucency translucency_percent } ]
  [ FriendlyName layer_name ]
]
[ Transparency { Off | On } ]
[ Color transparent_color_value ]
[ GrayScale { On | Off } ]
[ Relief { On | Off } ]
[ LABEL_CLAUSES ]
[ LAYER_OVERRIDE_CLAUSES ]
[ Display { Off | Graphic | Global } ]
[ Global Line... ] [ , Line... ] ...
[ Global Pen... ] [ , Pen... ] ...
[ Global Brush... ] [ , Brush... ] ...
[ Global Symbol... ] [ , Symbol... ] ...
[ Global Font... ] [ , Font... ] ...
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая

косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

LAYER_ACTIVATE_CLAUSES представляет собой обозначение группы режимов, а не слово из синтаксиса MapBasic. См. ниже описание предложения Layer Activate в разделе [Управление геолинками](#).

min_zoom – минимальное значение для масштабного эффекта слоя;

max_zoom – максимальное значение для масштабного эффекта слоя.

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: [Оператор Set Distance Units](#)). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

num_inflections – численное выражение, задающее пары точек перелома *color:value* в теме для поверхности.

color – выражение для цвета, его задает [функция RGB\(\)](#).

value – точка перелома для тематической расцветки двумя цветами.

rounding_factor – это числовое выражение, задающее величину округления при вычислении переломного значения.

contrast_value – значение контраста, от 0 до 100. Оно соответствует позиции бегунка в диалоге **Представление поверхности**, который можно открыть из диалога **Изменить тематическую Карту**.

brightness_value – значение яркости, от 0 до 100. Оно соответствует позиции бегунка в диалоге **Представление поверхности**, который можно открыть из диалога **Изменить тематическую Карту**.

Значения *contrast_value* и *brightness_value* не были описаны ранее. Это числа от 0 до 100, представляющие контраст и яркость; они соответствуют позиции бегунка в диалоге **Представление поверхности**, который можно открыть из диалога **Изменить тематическую Карту**.

alpha_value – целое число, представляющее значение альфа-канала для полупрозрачности. Значение изменяется в пределах 0-255. 0 – это полная прозрачность. 255 – это полная непрозрачность. Значения между 0-255 делают изображение слоя полупрозрачным.

Значение *translucency_percent* – это целое число, представляющее полупрозрачность либо векторного, либо растрового слоя, либо слоя поверхности. Значение изменяется в пределах 0-100. 0 – это полная непрозрачность. 100 – это полная прозрачность.

Примечание: Задавайте либо режим **Alpha**, либо **Translucency**, но не оба, так как это разные способы достижения одного результата. Если указано несколько таких параметров, будет использоваться последний.

layer_name – удобное (понятное) имя, которое будет использоваться для слоя карты. Это имя отображается в MapInfo Pro вместо имени таблицы слоя. Длина строки ограничена максимальным значением в 255 символов и не может быть пустым. Использование строки со значением "Default" очищает понятное имя.

transparent_color_value – задает значение цвета. В режиме **Transparency** растровое изображение показывается во всех цветах, за исключением одного (*transparent_color_value*).

LABEL_CLAUSES представляет собой обозначение группы режимов, а не слово из синтаксиса MapBasic. См. [Управление свойствами отдельных подписей](#).

LAYER_OVERRIDE_CLAUSES представляет собой обозначение группы режимов, а не слово из синтаксиса MapBasic. См. [Добавление определений единообразного оформления для слоя](#).

Описание

Предложение **Editable** устанавливает атрибут изменяемости для соответствующего Слая. В предложении Layer предложение **Editable** может установить режим изменяемости только для одного слоя. Если слой становится **изменяемым**, то он автоматически становится и **доступным**. Следующий оператор **Set Map** делает **изменяемым** только верхний некосметический слой:

```
Set Map
  Layer 1 Editable On
```

Предложение **Selectable** устанавливает для данного слоя режим доступности, т.е. разрешает выбирать на нем объекты такими инструментами, как Стрелка. **Доступными** могут быть несколько слоев карты или все слои. Следующий оператор **Set Map** делает **доступным** верхний некосметический слой, а следующие два – **недоступными**:

```
Set Map
  Layer 1 Selectable On
  Layer 2 Selectable Off
  Layer 3 Selectable Off
```

Предложение **Zoom** задает пределы для масштабного эффекта, то есть режима показа слоя только в определенных пределах увеличения. У каждого слоя могут быть пределы масштабного эффекта. Когда эти пределы активированы, MapInfo Pro отображает слой карты только когда ее масштаб находится в указанном диапазоне. Например, улицы города можно показывать только тогда, когда размер карты уменьшится до 10 км.

```
Set Map
  Layer 1 Zoom (0, 10) Units "km" On
```

Для включения масштабного эффект на слое, используйте слово **On**. Для выключения – **Off**.

Arrows – предложение управляет показом стрелок.

Centroids – предложение управляет показом центроидов объектов.

Inflect – переопределяет пары цвет:значения, хранящиеся в файле поверхностей (MIG).

Nodes – предложение управляет показом узлов на объектах.

Relief – включает и отключает отмывку рельефа в гридах. Грид должен иметь рассчитанные значения необходимые для создания отмывки, чтобы это предложение имело эффект. Информацию об отмывке рельефа для грида возвращает **оператор Relief Shade**.

Display задает режим показа слоя. Предложение **Display Off** отменяет показ слоя; **Display Graphic** показывает объекты слоя в собственном (сохраненным в таблице) оформлении; **Display Global** позволяет настраивать Глобальные параметры стиля оформления можно задавать с помощью дополнительных, необязательных предложений **Global**. Следующий оператор устанавливает режим показа объектов первого слоя тонкими зелеными линиями, контурами и зеленой заливкой:

```
Set Map
  Layer 1 Display Global
  Global Line (1, 2, GREEN)
  Global Pen (1, 2, GREEN)
  Global Brush (2, GREEN, WHITE)
```

Global Line задает стиль представления линий и полилиний. Предложение **Line** используется также как **предложение Pen**, за исключением ключевого слова **Line** вместо **Pen**.

Global Pen – корректное **предложение Pen**, задающее стиль линий и границ площадных объектов.

Global Symbol – корректное **предложение Symbol**, задающее стиль показа точечных объектов.

Global Brush – корректное **предложение Brush**, задающее стиль штриховки площадных объектов.

Global Font – корректное **предложение Font**, задающее шрифт текстовых объектов.

Предложения Global поддерживают комбинированные стили, которые задаются списком через запятую. Например, так можно задавать глобальный комбинированный стиль символа:

```
Set Map Layer 1 Display Global Global Symbol (32,16777136,24),
Symbol (36,255,14)
```

Так можно задавать глобальный комбинированный стиль линии:

```
Set Map Layer 1 Display Global
Zoom (0, 10000) Units "mi"
Global Line (4, 193, 16711680), Line (2, 193, 16711680)
```

Настройки отдельного слоя, имеющие постоянный эффект

Предложение **Default Zoom** воздействует на таблицу, а не на карту. Оно используется для установки значений стандартного увеличения и центральной точки для таблицы такими, какие они сейчас в окне карты.

Каждая таблица, к которой присоединена геоинформация, имеет стандартное увеличение и центральную точку. Эти значения применяются для представления открываемой карты.

Если в операторе **Set Map...Layer** содержится предложение **Default Zoom**, MapInfo Pro помещает в таблицу текущие значения увеличения и координат центральной точки. Например, следующий оператор изменяет размер и центр окна для таблицы, изображенной на первом слое:

```
Set Map Layer 1 Default Zoom
```

Default Zoom срабатывает сразу, не дожидаясь операции сохранения таблицы.

Примеры

Следующий пример включает режим показа стрелок, центроидов и узлов объектов первого некосметического слоя карты:

```
Set Map  
Layer 1 Arrows On Centroids On Nodes On
```

Следующий оператор разрешает показ объектов первого слоя в собственном стиле:

```
Set Map  
Layer 1 Display Graphic
```

Управление настройками подписей на слое

Следующие параметры и предложения управляют поведением подписей. Они воздействуют на весь слой. О предложениях, не описанные в этом разделе, можно прочитать в разделе:

[Управление представлением и поведением слоев..](#)

Синтаксис

```
Set Map  
[ Window window_id ]  
[ Layer layer_id  
  [ Label  
    [ Line { Simple | Arrow | None } ]  
    [ Position [ Center ] [ { Above | Below } ] [ { Left | Right } ] ]  
    [ Auto Retry { On | Off } ]  
    [ Font... ] [ Pen... ]  
    [ With label_expr ] [ Parallel { On | Off } ]  
    [ Follow Path [ BestPosition { On | Off } ] [ Percent Over percent ]  
  [ Fallback { On | Off } ] ]  
  [ Visibility { On | Off | Zoom ( min_vis, max_vis )  
    [ Units dist_unit ] } ]  
  [ Auto { On | Off } ] ]
```

```

[ Overlap { On | Off } ]
[ PartialSegments { On | Off } ]
[ Duplicates { On | Off } ]
[ Max [ number_of_labels ] ]
[ Offset offset_amount ]
[ Default ]
[ LabelAlpha alpha_value ]
[ AutoPosition { On | Off } ]
[ AutoSizes { number_font_sizes | Default } ]
[ AutoSizeStep percentage_value ]
[ SuppressIfNoFit { On | Off } ]
[ AutoCallout { On | Off } ]
[ Abbreviation { On | Off } Abbreviate with { field_expression } ]
[ LABEL_OVERRIDE_CLAUSE ] ]
[ Object bd
[ Table alias ]
[ Visibility { On | Off } ]
[ Anchor ( anchor_x, anchor_y ) ]
[ Text text_string ]
[ Position [ Center ] [ { Above | Below } ] [ { Left | Right } ] ]
[ Font... ] [ Pen... ]
[ Line { Simple | Arrow | None } ]
[ Angle text_angle ] [ Follow Path ]
[ Offset offset_amount ]
[ Callout ( callout_x, callout_y ) ]
[ , Object... ] ] ]

```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

label_expr – выражение, используемое для подписывания объекта;

min_vis, max_vis – минимальное и максимальное значения для масштабного эффекта подписи;

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: [Оператор Set Distance Units](#)). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

number_of_labels – целое число, представляющее максимальное количество подписей, которое MapInfo Pro может показать на слое. (по умолчанию лимита нет);

offset_amount – целое число от 0 до 200, расстояние в точках от подписи до точки привязки на подписываемом объекте;

alpha_value – короткое целое (SmallInt), представляющее значение альфа-фактора для подписей на данном слое. Значение изменяется в диапазоне 0-255, где 0 представляет полную

прозрачность, а 255 – полную непрозрачность. Между ними располагаются разные степени полупрозрачности.

number_font_sizes – целое число от 1 до 10, которое описывает шаг изменения размера шрифта, для того чтобы подписи помещались внутри областей. Если этот параметр не задан, то умолчанию используется значение равное 4.

percentage_value – целое число от 1 до 99, которое описывает минимальный процент уменьшения размера шрифта подписей при изменении размера карты. По умолчанию значение равно 50.

field_expression – имя колонки таблицы с аббревиатурами.

LABEL_OVERRIDE_CLAUSE представляет собой обозначение группы режимов, а не слово из синтаксиса MapBasic. См. [Добавление единообразного оформления подписей](#).

ID – целочисленный идентификатор изменяемой подписи (это предложение генерируется автоматически при сохранении Рабочего набора); Идентификатор подписи (*ID*) равен идентификатору строки, к которой присоединен подписываемый объект

alias – псевдоним таблицы-компонента сшитой карты. Предложение **Table alias** порождает ошибку, если слой не является компонентом сшитой карты.

anchor_x, *anchor_y* – координаты, задающие закрепленное положение подписи;

text_string – строка с текстом подписи;

text_angle – угол поворота подписи в градусах;

callout_x, *callout_y* – координаты, определяют конец указки при подписи.

Описание

Предложение **Label** задает режим подстановки и поведения подписей на слое. Предложение **Label** имеет следующие подпредложения:

Line – определяет тип указки, которая сопровождает подпись при ее перемещении, или ее отсутствие. Можно задать **Line Simple**, **Line Arrow** или **Line None**. Например:

```
Set Map Layer 1 Label Line Arrow
```

Position – управляет расположением подписи относительно центра объекта, к которому она прикреплена. Например, следующий оператор располагает подписи сверху и справа от центра:

```
Set Map Layer 1 Label Position Above Right
```

Auto Retry – параметр, позволяющий применять такой алгоритм размещения подписей, чтобы они не накладывались друг на друга.

- При создании рабочего набора, если режим Auto Retry включен (On), то предложение Auto Retry On записывается в рабочий набор после предложения Position (хотя порядок записи

не важен), а версия рабочего набора повышается до 9.5 или большего значения. Если этот режим отключен (Off), то не записывается ничего и версия рабочего набора не меняется. Рабочий набор версии 9.5 или большего значения может содержать запись Auto Retry Off, но автоматически она не вносится, чтобы избежать осложнений с номерами версий.

- *При чтении рабочего набора*: если в тексте обнаруживаются записи Auto Retry On или Auto Retry Off, то проверяется версия рабочего набора; она должна быть равна 9.5 или большего значения, иначе будет порождена ошибка. При задании Auto Retry On, предпринимаются попытки оптимально разместить подпись. При задании Auto Retry Off, попытки разместить подпись оптимально не предпринимаются (стандартный режим). Для того, чтобы режим Auto Retry действовал, режим Overlap должен быть установлен на Off. Если режим Overlap включен (On) то любая установка режима Auto Retry в одном и том же предложении LABELCLAUSE, то механизм оптимального размещения подписей не действует, значит, разрешено наложение подписей.

Font – корректное **предложение Font**, задающее стиль подписи.

Pen – корректное **предложение Pen**, задающее стиль линии для указки. Стиль линии указки имеет смысл для режимов **Line Simple** и **Line Arrow**, а также для случаев, когда пользователь сдвигает подпись с заданного положения:

```
Set Map Layer 1 Label Line Arrow Pen( 2, 1, 255)
```

With – задает выражение, образующее текст надписи. В следующем примере **функция Proper\$()** применяется к значениям из колонки с именами больших городов:

```
Set Map Layer 1 Label With Proper$(Cityname)
```

Parallel –определяет, будет ли строка подписи линейного объекта параллельна подписываемой линии. Задавайте значение **Off**, когда требуется, чтобы подписи не поворачивались вместе с сегментом линии, а оставались горизонтальными, и задавайте значение **On**, чтобы подписи поворачивались по сегменту линии.

```
Set Map Layer 1 Label Parallel On
```

Follow path – задает режим подписывания по кривой. **Path** – автоматически вычисляется и хранится до тех пор, пока положение подписи по кривой не будет изменено.

С помощью параметра **BestPosition** можно попытаться найти более подходящее место для подписи вдоль полилинии, когда полилиния сильно изогнута. Определяется три лучших положения. Если подпись не может нанесена ни в одном из этих положений, то она будет возвращена в стандартное положение для повернутой подписи, когда включен параметр **Fallback**. По умолчанию этот режим выключен.

```
Set Map Window 145919584 Layer 1 Label Follow Path BestPosition On
```

Параметр **Percent Over** применяется только к подписям по-кривой. Он применяется, если длина подписи по-кривой превышает геометрический размер объекта, тогда можно задать допустимую долю превышения (в процентах). Например:

```
Set Map Layer 1 Label Follow Path Percent Over 40
```

параметр **Fallback** применяется только к подписям по-кривой. Когда полилиния сильно изломана (не гладкая или не плавно изогнутая), трудно наносить подписи по-кривой. Переводите параметр **Fallback** в состояние **On**, для того чтобы создать прямую подпись, когда подпись по-кривой не может быть нанесена. Он поворачивает прямую подпись вдоль сегмента около того места, где должна быть нанесена подпись по-кривой. MapInfo Pro допускает пересечение полилинии подписью. С помощью этого параметра можно наносить на карту больше подписей. По умолчанию это параметр выключен, находится в состоянии **Off**. Когда MapInfo Pro не может нанести подпись по-кривой, то строится прямая подпись, при этом параметр, задающий процент свешивания игнорируется.

Visibility – управляет показом подписей для одного слоя. Предложение **Visibility Off** выключает показ как автоматических подписей, так и созданных вручную. Предложение **Visibility Zoom...** устанавливает показ подписей только, когда карта находится в определенном масштабном диапазоне. Следующий пример разрешает подписывание тогда, когда размер карты будет равен 2 километрам и менее.

```
Set Map Layer 1 Label Visibility Zoom (0, 2) Units "km"
```

Auto – управляет автоматическим подписыванием. Предложение **Auto Off** отключает автоматические подписи, но остаются созданные вручную.

Overlap – управляет режимом, разрешающим или запрещающим MapInfo Pro рисовать пересекающиеся подписи. Чтобы избежать наложение подписей установите **Overlap Off**.

PartialSegments – управляет способностью MapInfo Pro подписывать объекты, когда центроид вне зоны видимости окна карты. Если определен **PartialSegments On** (это соответствует установке флажка **Подписывание сегментов линий**), то MapInfo Pro подпишет видимые части объектов. Если задан параметр **PartialSegments Off**, объекты будут подписаны только если их центроид находится в окне карты.

Duplicates – управляет режимом, разрешающим или запрещающим MapInfo Pro дублировать подписи. Чтобы избежать дублирования подписей укажите **Duplicates Off**.

Max – устанавливает максимальное число подписей, которое MapInfo Pro может показать на этом слое. По умолчанию, если опустить аргумент *number_of_labels*, ограничений на количество подписей нет.

Offset – задает отступ подписи от центроида. Параметр *offset_amount* может принимать значения от 0 до 50 шрифтовых пунктов. Если задать **Offset 0**, то подпись будет примыкать к центроиду. Если задать **Offset 10**, то подпись будет располагаться на 10 пунктов в сторону. Отступ будет проигнорирован, если расположение подписи будет задано в центре объекта

(**Position Center**). Следующий оператор задает расположение подписей справа и на 10 точек в сторону от центроида:

```
Set Map Layer 1 Label Overlap On Position Right Offset 10
```

AutoPosition – включает или выключает автоматическое подписывание. Параметры **AutoSizes**, **AutoSizeStep** и **SuppressIfNoFit** активны, когда параметр **AutoPosition** включен с помощью аргумента оператора **On**. По умолчанию этот параметр выключен – стандартное состояние **Off**.

Параметр **AutoSizes** задает количество размеров шрифта, которые будут использованы при попытке поместить подпись в область в режиме единообразного подписывания. Значения могут быть от 1 до 10, по умолчанию значение задаётся равным 4. При значении параметра равном 1 используется текущий размер шрифта. Если задать значение параметра больше 1, то размер шрифта будет уменьшен до минимального значения размера шрифта, вычисленного с помощью **AutoSizeStep**. Задание параметра **Default** заставляет MapInfo Pro определить количество размеров шрифтов, которые будут использоваться (от 1 до 10).

AutoSizeStep определяет процент, на который будут меняться размеры шрифтов, чтобы подписи могли поместиться при изменении размеров карты. Значение может меняться от 1 до 99. Например, если исходный размер шрифта в 24 пункта, а параметр *number_font_sizes* – 66, то минимальный размер шрифта будет на 66 процентов меньше 24 пунктов, то есть минимальный шрифт будет размером 8 пунктов. По умолчанию значение равно 50.

Когда включен параметр **SuppressIfNoFit**, т. е. он находится в состоянии **On**, то будут показаны только те подписи, которые помещаются на карте после изменения размера карты. Если подпись не помещается внутри области, то она нанесена на карту не будет. Этот параметр по умолчанию включен, находится в состоянии **On**.

Параметр **AutoCallout** включает и выключает нанесение стрелок-выносок при дополнительном оформлении подписей. По умолчанию этот режим выключен. Оба параметра **AutoPosition** и **SuppressIfNoFit** должны быть в состоянии **On**, чтобы пользоваться параметром **AutoCallout**.

Параметр **Abbreviation** применяется только для подписей. Когда он включен и находится в состоянии **On**, то поле-колонка *field_expression* с аббревиатурами используется для подписей, которые не могут быть нанесены, поскольку перекрывают другие подписи или не могут быть вписаны внутри области. Если этот параметр включен, находится в состоянии **On**, то необходимо задавать параметр **Abbreviate with**.

Параметр **Abbreviate with** используется совместно с параметром **Abbreviation**. Этот параметр задаёт выражение с колонкой аббревиатур *field_expression*, которое будет использоваться для слоя таблицы.

```
Set Map Window 145919584 Layer 1 Label Abbreviation On
```

```
Set Map Window 145919584 Layer 1 Label Abbreviate with ShortName
```

Default – восстанавливает все подписи на слое и удаляет все внесенные вручную подписи и изменения в автоматических. Следующим оператором удаляются все отредактированные подписи с верхнего слоя окна карты, восстанавливая стандартные подписи:

```
Set Map Layer 1 Label Default
```

Предложение **Object** позволяет задать отдельную подпись для индивидуального объекта. Например, если была изменена одна подпись в MapInfo Pro, и сохранён рабочий набор, то он будет содержать оператор **Set Map** со столькоими предложениями **Object**, сколько подписей было индивидуально изменено.

Пример синтаксиса предложения **Object** можно увидеть, если откроете такой рабочий набор в любом текстовом редакторе.

Добавление определений единообразного оформления для слоя

Назначение

Предложение **Override Add** создает определение единообразного оформления объектов слоя и добавляет его в список единообразных оформлений для слоя, если такой список уже есть. Определения единообразного оформления позволяют показывать карту в разных стилях в зависимости от её масштаба.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Zoom ( min_zoom, max_zoom ) [ Units unit_dist ] ]
  [ Display { Off | Graphic | Global } ]
  [ Global Pen...[ , Pen... ]...]
  [ Global Line...[ , Line... ]...]
  [ Global Symbol...[ , Symbol... ]...]
  [ Global Brush...[ , Brush... ]...]
  [ Global Font...]
  [ { Alpha alpha_value } | { Translucency translucency_percent } ]
  [ STYLEOVERRIDE_CLAUSE ] ... ]
```

где **STYLEOVERRIDE_CLAUSE**:

```
[ [ Style ] Override Add [override_name] {
  [ Using [ Window window_id ] Layer layer_id {
    All | Override { override_index | override_name } }
  ] |
  Zoom ( min_zoom, max_zoom )
  [ Units dist_unit ]
  [ { Alpha alpha_value } | { Translucency translucency_percent } ]
```

```
[ Enable { On | Off } ]
[ Arrows { On | Off } ]
[ Centroids { On | Off } ]
[ Nodes { On | Off } ]
[ Line... ] [ , Line... ]
[ Pen... ] [ , Pen... ] ...
[ Symbol... ] [ , Symbol... ] ...
[ Brush... ] [ , Brush... ] ...
[ Font... ] [ , Font... ] ... }
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

override_index – целочисленный номер (отсчет от 1) описания единообразного оформления стилей на данном слое. Каждое такое определение соответствует некоему масштабному диапазону и порядок задается от самого меньшего значения масштаба в диапазонах (номер 1) к большим значениям.

override_name – имя определения единообразного оформления.

min_zoom – числовое выражение, задающее минимальный масштаб, с которого начинает применяться единообразное оформление.

max_zoom – числовое выражение, задающее максимальный масштаб, с которого начинает применяться единообразное оформление.

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: **Оператор Set Distance Units**). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

alpha_value – целое число, представляющее значение альфа-канала для полупрозрачности. Значение изменяется в пределах 0-255. 0 – это полная прозрачность. 255 – это полная непрозрачность. Значения между 0-255 делают изображение слоя полупрозрачным.

Значение *translucency_percent* – это целое число, представляющее полупрозрачность либо векторного, либо растрового слоя, либо слоя поверхности. Значение изменяется в пределах 0-100. 0 – это полная непрозрачность. 100 – это полная прозрачность.

Примечание: Задавайте либо режим **Alpha**, либо **Translucency**, но не оба, так как это разные способы достижения одного результата. Если указано несколько таких параметров, будет использоваться последний.

Описание

Определение единообразного оформления задает набор инструкцию стилизации, действующих в определенном диапазоне масштабов. К одному слою может быть приписано несколько определений единообразного оформления. Можно выбрать несколько масштабных диапазонов и в каждом из них задать разные стили показа объектов.

Вне заданных масштабных диапазонов соответствующие единообразные оформления отменяются. Для каждого слоя может быть определен только один такой набор определений единообразного оформления.

Arrows – предложение управляет показом стрелок.

Centroids – предложение управляет показом центроидов объектов.

Nodes – предложение управляет показом узлов на объектах.

Line – управляет стилем линий и полилиний. Предложение **Line** используется также как **предложение Pen**, за исключением ключевого слова **Line** вместо **Pen**.

Pen – корректное **предложение Pen**, задающее стиль линий и границ площадных объектов.

Symbol – корректное **предложение Symbol**, задающее стиль показа точечных объектов.

Brush – корректное **предложение Brush**, задающее стиль штриховки площадных объектов.

Font – корректное **предложение Font**, задающее шрифт текстовых объектов.

Using – предложение, позволяющее скопировать описание единообразного оформления с одного слоя на другой. Такое копирование действует только в рамках одного оператора.

Для каждого векторного слоя может поддерживаться несколько описаний единообразного оформления стилей и несколько описаний единообразного оформления подписей. Для оформления стилей задается масштабный диапазон, при этом диапазоны для разных описаний не должны пересекаться. Для оформления подписей также задается масштабный диапазон, при этом диапазоны для разных описаний не должны пересекаться. Однако, множества масштабных диапазонов для стилей и для подписей не должны совпадать.

Если для показа карты выбран масштаб, который содержится в одном из описаний единообразного оформления некоего слоя, то объекты на этом слое показываются в соответствии с этим описанием, а не так, как предписывают инструкции оформления слоя.

Если выбран масштаб, для которого нет описания единообразного оформления, то стиливое оформление слоя задается обычными инструкциями его оформления. Аналогично, если выбран масштаб, для которого нет описания единообразного оформления, то оформление подписей слоя задается обычными инструкциями его оформления.

Более подробно об описаниях единообразного оформления см. разделы: **Изменение определений единообразного оформления для слоя** и **Включение, выключение и отмена единообразных описаний стилей**. Смотрите также: **функция LayerStyleInfo()** и **функция StyleOverrideInfo()**.

Примеры

Следующий оператор добавляет описание единообразного оформления для слоя:

```
Set Map Layer 1
Style Override Add Zoom (0, 10000) Units "mi" Line (2, 193, 16711680)
```

Следующий оператор добавляет несколько описаний единообразного оформления для слоя:

```
Set Map Layer 1 Display Global
Zoom (1, 10000) Units "mi"
Global Line (1, 193, 16711680)
Style Override Add Zoom (1, 1000) Units "mi" Line (4, 193, 16711680),

Line (2, 193, 16711680)
Style Override Add Zoom (1000, 10000) Units "mi"
Line (2, 193, 16711680)
```

Пример: копирование стилей из одной карты в другую

Чтобы скопировать стилизацию слоя с одной карты на слой другой:

```
Set Map Layer 1 Style Override Add Using Window 81132792 Layer 1 All
```

Примеры: копирование стилей из одного слоя в другой

Следующий оператор добавляет к слою 2 описание оформления стиля под названием *layer1_style2* со слоя 1:

```
Set Map Layer 2
Style Override Add Using Layer 1 layer1_style1
```

Следующий оператор добавляет к слою 2 описание оформления стиля 3 со слоя 1:

```
Set Map Layer 2
Style Override Add layer2_style2 Using Layer 1 Override 3
```

Следующий оператор копирует все описания единообразного оформления стилей со слоя 2 на слой 3:

```
Set Map Layer 3 Display Global
Global Line (1, 193, 16711680)
Style Override Add Using Layer 2 All
```


Изменение определений единообразного оформления для слоя

Предложение `Override` без слова `Add` изменяет свойства существующих описаний единообразного оформления стилей, задаваемых либо номером (начиная с 1), либо именем описания.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ MODIFYSTYLEOVERRIDE_CLAUSE ]
  [ MODIFYSTYLEOVERRIDE_CLAUSE ] ... ]
```

где `MODIFYSTYLEOVERRIDE_CLAUSE`:

```
[ [ Style ] Override { override_index | override_name } {
[ Zoom ( min_zoom, max_zoom ) ]
[ Units dist_unit ]
[ { Alpha alpha_value } | { Translucency translucency_percent } ]
[ Enable { On | Off } ]
[ Arrows { On | Off } ]
[ Centroids { On | Off } ]
[ Nodes { On | Off } ]
[ Line... ] [ , Line... ]
[ Pen... ] [ , Pen... ] ...
[ Symbol... ] [ , Symbol... ] ...
[ Brush... ] [ , Brush... ] ...
[ Font... ] } ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа `SmallInt` (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

override_index – целочисленный номер (отсчет от 1) описания единообразного оформления стилей на данном слое. Каждое такое определение соответствует некоему масштабному диапазону и порядок задается от самого меньшего значения масштаба в диапазонах (номер 1) к большим значениям.

override_name – имя определения единообразного оформления.

min_zoom – числовое выражение, задающее минимальный масштаб, с которого начинает применяться единообразное оформление.

max_zoom – числовое выражение, задающее максимальный масштаб, с которого начинает применяться единообразное оформление.

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: **Оператор Set Distance Units**). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

alpha_value – целое число, представляющее значение альфа-канала для полупрозрачности. Значение изменяется в пределах 0-255. 0 – это полная прозрачность. 255 – это полная непрозрачность. Значения между 0-255 делают изображение слоя полупрозрачным.

Значение *translucency_percent* – это целое число, представляющее полупрозрачность либо векторного, либо растрового слоя, либо слоя поверхности. Значение изменяется в пределах 0-100. 0 – это полная непрозрачность. 100 – это полная прозрачность.

Примечание: Задавайте либо режим **Alpha**, либо **Translucency**, но не оба, так как это разные способы достижения одного результата. Если указано несколько таких параметров, будет использоваться последний.

Описание

Arrows – предложение управляет показом стрелок.

Centroids – предложение управляет показом центроидов объектов.

Nodes – предложение управляет показом узлов на объектах.

Line – управляет стилем линий и полилиний. Предложение **Line** используется также как **предложение Pen**, за исключением ключевого слова **Line** вместо **Pen**.

Pen – корректное **предложение Pen**, задающее стиль линий и границ площадных объектов.

Symbol – корректное **предложение Symbol**, задающее стиль показа точечных объектов.

Brush – корректное **предложение Brush**, задающее стиль штриховки площадных объектов.

Font – корректное **предложение Font**, задающее шрифт текстовых объектов.

Более подробно об описаниях единообразного оформления см. разделы: **Добавление определений единообразного оформления для слоя** и **Включение, выключение и отмена единообразных описаний стилей**. Смотрите также: **Функция LayerStyleInfo()** и **Функция StyleOverrideInfo()**.

Пример

```
Set Map Layer 1 Style Override 1 Alpha 119
```

Включение, выключение и отмена единообразных описаний стилей

Если для слоя определены описания единообразных оформлений, то они включены по умолчанию.

Чтобы включить или выключить описание, пользуйтесь ключевыми словами On и Off в предложении Override.

Чтобы удалить описание, пользуйтесь ключевым словом Remove в предложении Override.

Синтаксис: включение и выключение описаний

```
Set Map
[ Window window_id ]
[ Layer layer_id [ [ Style ] Override { On | Off } ] ]
```

Синтаксис: удаление назначенных стилей

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ [ Style ] Override Remove {
    All | override_index [ , override_index, ] } ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

override_index – целочисленный номер (отсчет от 1) описания единообразного оформления стилей на данном слое. Каждое такое определение соответствует некоему масштабному диапазону и порядок задается от самого меньшего значения масштаба в диапазонах (номер 1) к большим значениям.

Описание

Если для слоя определены описания единообразных оформлений, то они включены по умолчанию. Чтобы включить или выключить описание (но не удалить), пользуйтесь соответственно ключевыми словами On и Off в предложении Override.

Более подробно об описаниях единообразного оформления см. разделы [Добавление определений единообразного оформления для слоя](#) и [Изменение единообразного стиля оформления слоя](#).

Пример

```
Set Map Layer 1 Style Override Remove 3, 2
Set Map Layer 1 Style Override Remove All
```

Добавление единообразного оформления подписей

Предложение Label Override добавляет масштабный диапазон к существующему описанию единообразного оформления подписей или создает новое описание.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Label
    [ LABEL_OVERRIDE_CLAUSE ]
    [ LABEL_OVERRIDE_CLAUSE ] ... ]
```

где LABEL_OVERRIDE_CLAUSE:

```
[ [ Label ] Override Add [ labeloverride_name ] {
[ Using [ Window window_id ] Layer layer_id {
  All | Override {labeloverride_index | labeloverride_name } } ] |
  Zoom ( min_vis, max_vis )
[ Enable { On | Off } ]
[ Units dist_unit ]
[ Line { Simple | Arrow | None } ]
[ Position [ Center ] [ { Above | Below } ] [ { Left | Right } ] ]
[ Auto Retry { On | Off } ]
[ Font... ] [ Pen... ]
[ With label_expr ] [ Parallel { On | Off } ]
[ Follow Path [ BestPosition { On | Off } ] [ Percent Over percent ] [
Fallback { On | Off } ] ]
[ Auto { On | Off } ]
[ Overlap { On | Off } ]
[ PartialSegments { On | Off } ]
[ Duplicates { On | Off } ]
[ Max [ number_of_labels ] ]
[ Offset offset_amount ]
[ LabelAlpha alpha_value ]
[ AutoPosition { On | Off } ]
[ AutoSizes { number_font_sizes | Default } ]
[ AutoSizeStep percentage_value ]
[ SuppressIfNoFit { On | Off } ]
[ AutoCallout { On | Off } ]
[ Abbreviation { On | Off } Abbreviate with { field_expression } ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

labeloverride_index – целочисленный индекс (начинающийся с единицы 1) описания режима единообразного подписывания на данном слое. Каждое описание режима единообразного подписывания привязано к масштабному коэффициенту и упорядочено так, что единообразный режим показа для минимального масштаба находится сверху (индекс равен 1).

labeloverride_name – заданное пользователем имя определения.

min_vis, *max_vis* – числа, задающие максимальный и минимальный масштаб, между которыми применяется единообразное оформление.

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, "mi" для миль, "m" для метров; см. описание единиц измерения в разделе: **Оператор Set Distance Units**). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

label_expr – выражение, используемое для подписывания объекта;

Percent Over – если длина подписи по кривой превышает геометрический размер объекта, то можно задать допустимую долю превышения (в процентах).

number_of_labels – целое число, представляющее максимальное количество подписей, которое MapInfo Pro может показать на слое. (по умолчанию лимита нет);

offset_amount – целое число от 0 до 200, расстояние в точках от подписи до точки привязки на подписываемом объекте;

alpha_value – целое число, представляющее значение альфа-канала для полупрозрачности. Значение изменяется в пределах 0-255. 0 – это полная прозрачность. 255 – это полная непрозрачность. Значения между 0 и 255 представляют градации полупрозрачности.

number_font_sizes – целое число от 1 до 10, которое описывает шаг изменения размера шрифта, для того чтобы подписи помещались внутри областей. Если этот параметр не задан, то по умолчанию используется значение равное 4.

percentage_value – целое число от 1 до 99, которое описывает минимальный процент уменьшения размера шрифта подписей при изменении размера карты. По умолчанию значение равно 50.

field_expression – имя колонки таблицы с аббревиатурами.

Описание

Можно задавать масштабный диапазон, внутри которого к подписям будут применяться особые правила форматирования. Таких диапазонов для слоя можно задать несколько. С их помощью

можно, например, изменять текст подписи в зависимости от масштаба или добиваться эффекта увеличения подписей при увеличении масштаба.

Вне заданных масштабных диапазонов соответствующие единообразные оформления отменяются. Для каждого слоя может быть определен только один такой набор определений единообразного оформления.

Line – определяет тип указки, которая сопровождает подпись при ее перемещении, или ее отсутствие. Можно задать **Line Simple**, **Line Arrow** или **Line None**. Например:

Position – управляет расположением подписи относительно центра объекта, к которому она прикреплена. Например, следующий оператор располагает подписи сверху и справа от центра:

Auto Retry – параметр, позволяющий применять такой алгоритм размещения подписей, чтобы они не накладывались друг на друга.

- *При создании рабочего набора*, если режим Auto Retry включен (On), то предложение Auto Retry On записывается в рабочий набор после предложения Position (хотя порядок записи не важен), а версия рабочего набора повышается до 9.5 или большего значения. Если этот режим отключен (Off), то не записывается ничего и версия рабочего набора не меняется. Рабочий набор версии 9.5 или большего значения может содержать запись Auto Retry Off, но автоматически она не вносится, чтобы избежать осложнений с номерами версий.
- *При чтении рабочего набора*: если в тексте обнаруживаются записи Auto Retry On или Auto Retry Off, то проверяется версия рабочего набора; она должна быть равна 9.5 или большего значения, иначе будет порождена ошибка. При задании Auto Retry On, предпринимаются попытки оптимально разместить подпись. При задании Auto Retry Off, попытки разместить подпись оптимально не предпринимаются (стандартный режим). Для того, чтобы режим Auto Retry действовал, режим Overlap должен быть установлен на Off. Если режим Overlap включен (On) то любая установка режима Auto Retry в одном и том же предложении LABELCLAUSE, то механизм оптимального размещения подписей не действует, значит, разрешено наложение подписей.

Font – корректное **предложение Font**, задающее стиль подписи.

Pen – корректное **предложение Pen**, задающее стиль линии для указки. Стиль линии указки имеет смысл для режимов **Line Simple** и **Line Arrow**, а также для случаев, когда пользователь сдвигает подпись с заданного положения:

```
Set Map Layer 1 Label Line Arrow Pen( 2, 1, 255)
```

With – задает выражение, образующее текст надписи. В следующем примере **функция Proper\$()** применяется к значениям из колонки с именами больших городов:

```
Set Map Layer 1 Label With Proper$(Cityname)
```

Parallel – определяет, будет ли строка подписи линейного объекта параллельна подписываемой линии. Задавайте значение **Off**, когда требуется, чтобы подписи не поворачивались вместе с

сегментом линии, а оставались горизонтальными, и задавайте значение **On**, чтобы подписи поворачивались по сегменту линии.

```
Set Map Layer 1 Label Parallel On
```

Follow path – задает режим подписывания по кривой. **Path** – автоматически вычисляется и хранится до тех пор, пока положение подписи по кривой не будет изменено.

С помощью параметра **BestPosition** можно попытаться найти более подходящее место для подписи вдоль полилинии, когда полилиния сильно изогнута. Определяется три лучших положения. Если подпись не может нанесена ни в одном из этих положений, то она будет возвращена в стандартное положение для поворнутой подписи, когда включен параметр **Fallback**. По умолчанию этот режим выключен.

```
Set Map Window 145919584 Layer 1 Label Override 1 Follow Path BestPosition On
```

Параметр **Percent Over** применяется только к подписям по-кривой. Он применяется, если длина подписи по-кривой превышает геометрический размер объекта, тогда можно задать допустимую долю превышения (в процентах). Например:

```
Set Map Layer 1 Label Override Follow Path Percent Over 40
```

параметр **Fallback** применяется только к подписям по-кривой. Когда полилиния сильно изломана (не гладкая или не плавно изогнутая), трудно наносить подписи по-кривой. Переводите параметр **Fallback** в состояние **On**, для того чтобы создать прямую подпись, когда подпись по-кривой не может быть нанесена. Он поворачивает прямую подпись вдоль сегмента около того места, где должна быть нанесена подпись по-кривой. MapInfo Pro допускает пересечение полилинии подписью. С помощью этого параметра можно наносить на карту больше подписей. По умолчанию это параметр выключен, находится в состоянии **Off**. Когда MapInfo Pro не может нанести подпись по-кривой, то строится прямая подпись, при этом параметр, задающий процент свешивания игнорируется.

Auto – управляет автоматическим подписыванием. Предложение **Auto Off** отключает автоматические подписи, но остаются созданные вручную.

Overlap – управляет режимом, разрешающим или запрещающим MapInfo Pro рисовать пересекающиеся подписи. Чтобы избежать наложение подписей установите **Overlap Off**.

PartialSegments – управляет способностью MapInfo Pro подписывать объекты, когда центроид вне зоны видимости окна карты. Если определён **PartialSegments On** (это соответствует установке флажка **Подписывание сегментов линий**), то MapInfo Pro подпишет видимые части объектов. Если задан параметр **PartialSegments Off**, объекты будут подписаны только если их центроид находится в окне карты.

Duplicates – управляет режимом, разрешающим или запрещающим MapInfo Pro дублировать подписи. Чтобы избежать дублирования подписей укажите **Duplicates Off**.

Max number_of_labels – устанавливает максимальное число подписей, которое MapInfo Pro может показать на этом слое. По умолчанию, если опустить аргумент *number_of_labels*, ограничений на количество подписей нет.

Offset offset_amount – задает отступ подписи от центроида. Параметр *offset_amount* может принимать значения от 0 до 50 шрифтовых пунктов. Если задать **Offset 0**, то подпись будет примыкать к центроиду. Если задать **Offset 10**, то подпись будет располагаться на 10 пунктов в сторону. Отступ будет проигнорирован, если расположение подписи будет задано в центре объекта (**Position Center**).

AutoPosition – включает или выключает автоматическое подписывание. Параметры **AutoSizes**, **AutoSizeStep** и **SuppressIfNoFit** активны, когда параметр **AutoPosition** включен с помощью аргумента оператора **On**. По умолчанию этот параметр выключен – стандартное состояние **Off**.

Параметр **AutoSizes** задает количество размеров шрифта, которые будут использованы при попытке поместить подпись в область в режиме единообразного подписывания. Значения могут быть от 1 до 10, по умолчанию значение задается равным 4. При значении параметра равном 1 используется текущий размер шрифта. Если задать значение параметра больше 1, то размер шрифта будет уменьшен до минимального значения размера шрифта, вычисленного с помощью **AutoSizeStep**. Задание параметра **Default** заставляет MapInfo Pro определить количество размеров шрифтов, которые будут использоваться (от 1 до 10).

AutoSizeStep определяет процент, на который будут меняться размеры шрифтов, чтобы подписи могли поместиться при изменении размеров карты. Значение может меняться от 1 до 99. Например, если исходный размер шрифта в 24 пункта, а параметр *number_font_sizes* – 66, то минимальный размер шрифта будет на 66 процентов меньше 24 пунктов, то есть минимальный шрифт будет размером 8 пунктов. По умолчанию значение равно 50.

Когда включен параметр **SuppressIfNoFit**, т. е. он находится в состоянии **On**, то будут показаны только те подписи, которые помещаются на карте после изменения размера карты. Если подпись не помещается внутри области, то она нанесена на карту не будет. Этот параметр по умолчанию включен, находится в состоянии **On**.

Параметр **AutoCallout** включает и выключает нанесение стрелок-выносок при дополнительном оформлении подписей. По умолчанию этот режим выключен. Оба параметра **AutoPosition** и **SuppressIfNoFit** должны быть в состоянии **On**, чтобы пользоваться параметром **AutoCallout**.

Параметр **Abbreviation** применяется только для подписей. Когда он включен и находится в состоянии **On**, то поле-колонка *field_expression* с аббревиатурами используется для подписей, которые не могут быть нанесены, поскольку перекрывают другие подписи или не могут быть вписаны внутри области. Если этот параметр включен, находится в состоянии **On**, то необходимо задавать параметр **Abbreviate with**.

Параметр **Abbreviate** with используется совместно с параметром **Abbreviation**. Этот параметр задаёт выражение с колонкой аббревиатур *field_expression*, которое будет использоваться для слоя таблицы.

```
Set Map Window 145919584 Layer 1 Label Override 1 Abbreviation On
```

```
Set Map Window 145919584 Layer 1 Label Override 1 Abbreviate with ShortName
```

Более подробно об описаниях единообразного оформления см. разделы [Изменение определений единообразного оформления для слоя](#) и [Включение, выключение и отмена единообразных описаний стилей](#). См. также: [функция LabelOverrideInfo\(\)](#).

Примеры

В следующем примере задается единообразное оформление подписей:

```
Set Map Layer 1 Label Zoom (1, 100000) with State
Override Add Zoom (1, 1000) with State_Name
Override Add Zoom (1000, 10000) with State
```

В следующем примере добавляется новый стиль оформления подписей:

```
Set Map Layer 1 Label Override Add Zoom (10000, 100000)
with State Overlap Off
```

Изменение описаний единообразного оформления подписей

Предложение **Override** без слова **Add** изменяет свойства существующих описаний единообразного оформления подписей, задаваемых либо номером (начиная с 1), либо именем описания.

```
Set Map [Layer layer_id
[Label...
[ MODIFYLABEL_OVERRIDE_CLAUSE ]
[ MODIFYLABEL_OVERRIDE_CLAUSE ] ... ] ]
```

где **MODIFYLABEL_OVERRIDE_CLAUSE**:

```
[ Override { labeloverride_index | labeloverride_name } {
[ Zoom ( min_vis, max_vis ) [ Units dist_unit ] ]
[ Enable { On | Off } ]
[ Line { Simple | Arrow | None } ]
[ Position [ Center ] [ { Above | Below } ] [ { Left | Right } ] ]
[ Auto Retry { On | Off } ]
[ Font... ] [ Pen... ]
[ With label_expr ] [ Parallel { On | Off } ]
```



```
[ Follow Path [ BestPosition { On | Off } ] [ Percent Over percent ] [
Fallback { On | Off } ] ]
[ Auto { On | Off } ]
[ Overlap { On | Off } ]
[ PartialSegments { On | Off } ]
[ Duplicates { On | Off } ]
[ Max [ number_of_labels ] ]
[ Offset offset_amount ]
[ LabelAlpha alpha_value ] }
[ AutoPosition { On | Off } ]
[ AutoSizes { number_font_sizes | Default } ]
[ AutoSizeStep percentage_value ]
[ SuppressIfNoFit { On | Off } ]
[ AutoCallout { On | Off } ]
[ Abbreviation { On | Off } Abbreviate with { field_expression } ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

labeloverride_index – целочисленный индекс (начинающийся с единицы 1) описания режима единообразного подписывания на данном слое. Каждое описание режима единообразного подписывания привязано к масштабному коэффициенту и упорядочено так, что единообразный режим показа для минимального масштаба находится сверху (индекс равен 1).

labeloverride_name – заданное пользователем имя определения.

min_vis, *max_vis* – числа, задающие максимальный и минимальный масштаб, между которыми применяется единообразное оформление.

dist_unit – строка, представляющая единицы измерения, в которых производятся вычисления расстояний на карте (например, “mi” для миль, “m” для метров; см. описание единиц измерения в разделе: [Оператор Set Distance Units](#)). Задавать этот параметр необязательно. Если эта величина не задана, то используются единицы координатной системы из обрабатываемой таблицы.

label_expr – выражение, используемое для подписывания объекта;

Percent Over – если длина подписи по кривой превышает геометрический размер объекта, то можно задать допустимую долю превышения (в процентах).

number_of_labels – целое число, представляющее максимальное количество подписей, которое MapInfo Pro может показать на слое. (по умолчанию лимита нет);

offset_amount – целое число от 0 до 200, расстояние в точках от подписи до точки привязки на подписываемом объекте;

alpha_value – целое число, представляющее значение альфа-канала для полупрозрачности. Значение изменяется в пределах 0-255. 0 – это полная прозрачность. 255 – это полная непрозрачность. Значения между 0 и 255 представляют градации полупрозрачности.

number_font_sizes – целое число от 1 до 10, которое описывает шаг изменения размера шрифта, для того чтобы подписи помещались внутри областей. Если этот параметр не задан, то умолчанию используется значение равное 4.

percentage_value – целое число от 1 до 99, которое описывает минимальный процент уменьшения размера шрифта подписей при изменении размера карты. По умолчанию значение равно 50.

field_expression – имя колонки таблицы с аббревиатурами.

Описание

Line – определяет тип указки, которая сопровождает подпись при ее перемещении, или ее отсутствие. Можно задать **Line Simple**, **Line Arrow** или **Line None**. Например:

Position – управляет расположением подписи относительно центра объекта, к которому она прикреплена. Например, следующий оператор располагает подписи сверху и справа от центра:

Auto Retry – параметр, позволяющий применять такой алгоритм размещения подписей, чтобы они не накладывались друг на друга.

- *При создании рабочего набора*, если режим Auto Retry включен (On), то предложение Auto Retry On записывается в рабочий набор после предложения Position (хотя порядок записи не важен), а версия рабочего набора повышается до 9.5 или большего значения. Если этот режим отключен (Off), то не записывается ничего и версия рабочего набора не меняется. Рабочий набор версии 9.5 или большего значения может содержать запись Auto Retry Off, но автоматически она не вносится, чтобы избежать осложнений с номерами версий.
- *При чтении рабочего набора*: если в тексте обнаруживаются записи Auto Retry On или Auto Retry Off, то проверяется версия рабочего набора; она должна быть равна 9.5 или большего значения, иначе будет порождена ошибка. При задании Auto Retry On, предпринимаются попытки оптимально разместить подпись. При задании Auto Retry Off, попытки разместить подпись оптимально не предпринимаются (стандартный режим). Для того, чтобы режим Auto Retry действовал, режим Overlap должен быть установлен на Off. Если режим Overlap включен (On) то любая установка режима Auto Retry в одном и том же предложении LABELCLAUSE, то механизм оптимального размещения подписей не действует, значит, разрешено наложение подписей.

Font – корректное **предложение Font**, задающее стиль подписи.

Pen – корректное **предложение Pen**, задающее стиль линии для указки. Стиль линии указки имеет смысл для режимов **Line Simple** и **Line Arrow**, а также для случаев, когда пользователь сдвигает подпись с заданного положения:

```
Set Map Layer 1 Label Line Arrow Pen( 2, 1, 255)
```

With – задает выражение, образующее текст надписи. В следующем примере **функция Proper\$()** применяется к значениям из колонки с именами больших городов:

```
Set Map Layer 1 Label With Proper$(Cityname)
```

Parallel –определяет, будет ли строка подписи линейного объекта параллельна подписываемой линии. Задавайте значение **Off**, когда требуется, чтобы подписи не поворачивались вместе с сегментом линии, а оставались горизонтальными, и задавайте значение **On**, чтобы подписи поворачивались по сегменту линии.

```
Set Map Layer 1 Label Parallel On
```

Follow path – задает режим подписывания по кривой. **Path** – автоматически вычисляется и хранится до тех пор, пока положение подписи по кривой не будет изменено.

С помощью параметра **BestPosition** можно попытаться найти более подходящее место для подписи вдоль полилинии, когда полилиния сильно изогнута. Определяется три лучших положения. Если подпись не может нанесена ни в одном из этих положений, то она будет возвращена в стандартное положение для повернутой подписи, когда включен параметр **Fallback**. По умолчанию этот режим выключен.

```
Set Map Window 145919584 Layer 1 Label Override 1 Follow Path BestPosition On
```

Параметр **Percent Over** применяется только к подписям по-кривой. Он применяется, если длина подписи по-кривой превышает геометрический размер объекта, тогда можно задать допустимую долю превышения (в процентах). Например:

```
Set Map Layer 1 Label Override Follow Path Percent Over 40
```

параметр **Fallback** применяется только к подписям по-кривой. Когда полилиния сильно изломана (не гладкая или не плавно изогнутая), трудно наносить подписи по-кривой. Переводите параметр **Fallback** в состояние **On**, для того чтобы создать прямую подпись, когда подпись по-кривой не может быть нанесена. Он поворачивает прямую подпись вдоль сегмента около того места, где должна быть нанесена подпись по-кривой. MapInfo Pro допускает пересечение полилинии подписью. С помощью этого параметра можно наносить на карту больше подписей. По умолчанию это параметр выключен, находится в состоянии **Off**. Когда MapInfo Pro не может нанести подпись по-кривой, то строится прямая подпись, при этом параметр, задающий процент свешивания игнорируется.

Auto – управляет автоматическим подписыванием. Предложение **Auto Off** отключает автоматические подписи, но остаются созданные вручную.

Overlap – управляет режимом, разрешающим или запрещающим MapInfo Pro рисовать пересекающиеся подписи. Чтобы избежать наложение подписей установите **Overlap Off**.

PartialSegments –управляет способностью MapInfo Pro подписывать объекты, когда центроид вне зоны видимости окна карты. Если определён **PartialSegments On** (это соответствует установке флажка **Подписывание сегментов линий**), то MapInfo Pro подпишет видимые

части объектов. Если задан параметр **PartialSegments Off**, объекты будут подписаны только если их центроид находится в окне карты.

Duplicates – управляет режимом, разрешающим или запрещающим MapInfo Pro дублировать подписи. Чтобы избежать дублирования подписей укажите **Duplicates Off**.

Max number_of_labels – устанавливает максимальное число подписей, которое MapInfo Pro может показать на этом слое. По умолчанию, если опустить аргумент *number_of_labels*, ограничений на количество подписей нет.

Offset *offset_amount* – задает отступ подписи от центроида. Параметр *offset_amount* может принимать значения от 0 до 50 шрифтовых пунктов. Если задать **Offset 0**, то подпись будет примыкать к центроиду. Если задать **Offset 10**, то подпись будет располагаться на 10 пунктов в сторону. Отступ будет проигнорирован, если расположение подписи будет задано в центре объекта (**Position Center**).

AutoPosition – включает или выключает автоматическое подписывание. Параметры **AutoSizes**, **AutoSizeStep** и **SuppressIfNoFit** активны, когда параметр **AutoPosition** включен с помощью аргумента оператора **On**. По умолчанию этот параметр выключен – стандартное состояние **Off**.

Параметр **AutoSizes** задает количество размеров шрифта, которые будут использованы при попытке поместить подпись в область в режиме единообразного подписывания. Значения могут быть от 1 до 10, по умолчанию значение задаётся равным 4. При значении параметра равном 1 используется текущий размер шрифта. Если задать значение параметра больше 1, то размер шрифта будет уменьшен до минимального значения размера шрифта, вычисленного с помощью **AutoSizeStep**. Задание параметра **Default** заставляет MapInfo Pro определить количество размеров шрифтов, которые будут использоваться (от 1 до 10).

AutoSizeStep определяет процент, на который будут меняться размеры шрифтов, чтобы подписи могли поместиться при изменении размеров карты. Значение может меняться от 1 до 99. Например, если исходный размер шрифта в 24 пункта, а параметр *number_font_sizes* – 66, то минимальный размер шрифта будет на 66 процентов меньше 24 пунктов, то есть минимальный шрифт будет размером 8 пунктов. По умолчанию значение равно 50.

Когда включен параметр **SuppressIfNoFit**, т. е. он находится в состоянии **On**, то будут показаны только те подписи, которые помещаются на карте после изменения размера карты. Если подпись не помещается внутри области, то она нанесена на карту не будет. Этот параметр по умолчанию включен, находится в состоянии **On**.

Параметр **AutoCallout** включает и выключает нанесение стрелок-выносок при дополнительном оформлении подписей. По умолчанию этот режим выключен. Оба параметра **AutoPosition** и **SuppressIfNoFit** должны быть в состоянии **On**, чтобы пользоваться параметром **AutoCallout**.

Параметр **Abbreviation** применяется только для подписей. Когда он включен и находится в состоянии **On**, то поле-колонка *field_expression* с аббревиатурами используется для подписей, которые не могут быть нанесены, поскольку перекрывают другие подписи или не могут быть вписаны внутри области. Если этот параметр включен, находится в состоянии **On**, то необходимо задавать параметр **Abbreviate with**.

Параметр **Abbreviate** with используется совместно с параметром **Abbreviation**. Этот параметр задаёт выражение с колонкой аббревиатур *field_expression*, которое будет использоваться для слоя таблицы.

```
Set Map Window 145919584 Layer 1 Label Override 1 Abbreviation On
```

```
Set Map Window 145919584 Layer 1 Label Override 1 Abbreviate with  
ShortName
```

Более подробно о единообразном оформлении см. разделы: [Добавление определений единообразного оформления для слоя](#) и [Включение, выключение и отмена единообразных описаний стилей](#). См. также: [функция LabelOverrideInfo\(\)](#).

Пример

В следующем примере изменяются стили подписей:

```
Set Map Layer 1 Label Override 3 Zoom (1000, 10000)  
Line Arrow Pen (2, 1, 255)
```

Включение, выключение и отмена единообразных описаний подписей

Если для слоя определены описания единообразных оформлений, то они включены по умолчанию.

Чтобы включить или выключить описание, пользуйтесь ключевыми словами On и Off в предложении **Label Override**.

Чтобы удалить описание, пользуйтесь ключевым словом Remove в предложении Override.

Синтаксис: включение и выключение описаний

```
Set Map  
[ Window window_id ]  
[ Layer layer_id  
  [ Label [ Overrides { On | Off } ] ] ]
```

Синтаксис: удаление назначенных стилей

```
Set Map  
[ Window window_id ]  
[ Layer layer_id  
  [ Label [ Override Remove { All | labeloverride_index  
    [ , labeloverride_index ... ] } ] ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

labeloverride_index – целочисленный номер (отсчет от 1) определения единообразного оформления подписи. Каждое описание режима единообразного подписывания привязано к масштабному коэффициенту и упорядочено так, что единообразный режим показа для минимального масштаба находится сверху (индекс равен 1).

Более подробно об описаниях единообразного оформления см. разделы [Добавление единообразного оформления подписей](#) и [Изменение единообразного оформления подписей](#). См. также: [функция LabelOverrideInfo\(\)](#).

Пример

В следующем примере удаляются стили:

```
Set Map Layer 1 Label Override Remove 3
Set Map Layer 1 Label Override Remove All
```

Управление группировкой слоев

Следующие предложения описывают приемы группировки слоев. **Оператор Set Map** задает свойства групп аналогично свойствам слоев.

Примечание: О предложениях, не описанные в этом разделе, можно прочитать в разделе: [Управление представлением и поведением слоев](#).

Синтаксис 1 (группировка)

```
Set Map
[ Window window_id ]
[ GroupLayer group_id [ Display { On | Off } ] ]
[ Title "new_friendly_name" ] ]
```

Синтаксис 2 (ропуск группы)

```
Set Map
[ Window window_id ]
[ GroupLayer group_id [ Ungroup [ All ] ] ]
```

window_id – целочисленный идентификатор окна Карты.

group_id – может быть либо номером, либо именем группы. Имя будет ссылаться на первую группу в списке с этим именем.

Описание

Ungroup распускает группу слоев, но сохраняет все слои группы в основном списке. Специальное значение параметра *All* используется для удаления вложенных групп и переноса всех слоев из вложенной группы в родительскую. Порядок наложения слоев сохранится.

Примеры

GroupLayer "Tropics" (group 1)

Tropic_Of_Capricorn (layer 1)

Tropic_Of_Cancer (layer 2)

Wgrid15 (layer 3)

GroupLayer "World Places" (group 2)

WorldPlaces (layer 4)

WorldPlacesMajor (layer 5)

WorldPlaces_Capitals (layer 6)

Airports (layer 7)

Set Map GroupLayer 1 Ungroup создает следующий набор слоев (с пересчитанными номерами групп и слоёв):

Tropic_Of_Capricorn (layer 1)

Tropic_Of_Cancer (layer 2)

Wgrid15 (layer 3)

GroupLayer "World Places" (group 2)

WorldPlaces (layer 4)

WorldPlacesMajor (layer 5)

WorldPlaces_Capitals (layer 6)

Airports (layer 7)

в то время как

Set Map GroupLayer 1 Ungroup All создает такой список:

```
Tropic_Of_Capricorn (layer 1)
Tropic_Of_Cancer (layer 2)
Wgrid15 (layer 3)
WorldPlaces (layer 4)
WorldPlacesMajor (layer 5)
WorldPlaces_Capitals (layer 6)
Airports (layer 7)
```

Заголовок будет переименован на имя группы слоев, заданное строкой в *grouplayer_id_string*. В следующем примере группа слоев 2 получает название "Миру - Мир":

```
Set Map GroupLayer 2 Title "Hello World"
```

Задание порядка слоев

Следующие предложения позволяют перемещать слои и группы слоев по общему списку слоев.

Синтаксис

```
Set Map
[ Window window_id ]
[ Order layer_id, [ , layer_id ... ] ]
[
  [ GroupLayers group_layer_id [ , group_layer_id... ] ]
  [ Layers layer_id [ , layer_id ] ] . . .
  [ DestGroupLayer group_layer_id [ Position position ] ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – число, обозначающее слой, который будет изменен, отсчитываемое в структуре слоев от 1, которая обозначает самый верхний слой.

group_layer_id – число, обозначающее группу слоев, которая будет изменена, отсчитываемое сверху вниз в структуре слоев.

position в списке определяется по формуле: 1- базисный индекс внутри группы, в которую добавляется список перемещаемых слоев. Стандартное значение *position* – первое место в группе (*position* = 1).

Описание

Косметический слой является специальным слоем и имеет номер слоя, равный 0 (нулю). Он рисуется всегда последним и его не нужно задавать в предложении **Order**. В следующем примере первый (верхний) слой Карты и второй (находящийся под ним) меняются местами. Косметический слой остается самым верхним.

```
Set Map Order 2, 1, 3, 4
```

Следующий оператор **Set Map Order** снова перетасовывает слои Карты. Он помещает слои 3, 2, и 1 наверх, при этом они извлекаются из групп, которым, возможно, принадлежали.

```
Set Map Order Layers 3, 2, 1
```


Сочетание предложений `GroupLayers` и `Layers` позволяет Вам более гибко перемещать слои и группы слоев.

Необязательным параметром **DestGroupLayer** задается группа, в которую можно включить один или несколько слоев или групп, и их позиция. Для задания точного положения в списке добавляемых слоев этим предложением можно пользоваться и в старом варианте синтаксиса. Если он пропущен, это означает, что группы и/или слои будут добавлены в начало списка (так как было предусмотрено старым вариантом синтаксиса). Однако начало списка можно задавать с помощью идентификатора группы равным 0.

Положение `position` в списке определяется по формуле: 1- базисный индекс внутри группы, в которую добавляется список перемещаемых слоев. Если параметр `position` пропущен, то используется первый номер группы (`position = 1`).

Если заданный номер `position` превышает количество слоев в группе, то новые слои, и/или группа слоев будет вставлены в конец группы назначения.

Идентификатор слоя или группы слоев может быть либо число или имя. Диапазон номеров идентификатора групп может быть от 0 до количества групп в списке слоев.

После изменения порядка перечисления элементов списка все идентификаторы последовательно перенумеровываются в порядке от начала списка до его конца.

Тематические и их базовые слои должны перечисляться строго последовательно, поэтому оператор `Set Map Order` не позволяет добавлять слои между слоями тематического набора. Если по заданному адресу положения `position` добавляемые слои попадают внутрь тематического набора (базовый слой и все тематические карты, построенные на основе его данных), то перемещаемые слои будут вставлены в список выше или ниже тематического набора, в зависимости от того, чей адрес меньше отличается от адреса `position`.

Управление подписями

Если карта содержит автоматические подписи для двух или более слоёв, то не все объекты будут подписаны. Это происходит вследствие того, что подписи разных слоёв соревнуются за ограниченное пространство карты, поэтому город может быть не подписан, так как на этом месте уже подписана, например, дорога. Кроме того иногда возникает необходимость управлять возможностью выбора подписей. Если подписи нельзя выбирать, то нельзя будет по ошибке выбрать подпись вместо объектов, которые располагаются рядом или перекрываются подписью. Следующее предложение задаёт, что слой имеет высокий приоритет показа подписей, перемещая идентификатор слоя наверх списка, и позволяет переключать возможность выбора подписей.

Синтаксис

```
Set Map
[ Window window_id ]
[ Label
```

```
[ Selection { On | Off } ]
[ Priority { Default | layer_id [ , layer_id ... ] } ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – число, обозначающее слой, который будет изменен, отсчитываемое в структуре слоев от 1, которая обозначает самый верхний слой.

Описание

Косметический слой является специальным слоем и имеет номер слоя, равный 0 (нулю). Он рисуется всегда последним; поскольку нуль не должен появляться в предложении **LabelPriority** в качестве значения *layer_id*.

Если выбран некорректный идентификатор слоя, то появится сообщение об ошибке 600:

Предложение **Selection** включает и выключает выбор подписей на карте. По умолчанию можно выбирать подписи в окне карты. После того, как эта возможность отключена, нельзя:

- выбирать любые подписи на карте независимо от того, какому слою они принадлежат;
- с помощью двойного щелчка открывать диалог для подписей (это касается и объектов карты).

Когда выбор подписей карты отключен,

```
Set Map Label Selection Off
```

оператор сохраняется в рабочем наборе и номер версии рабочего набора увеличивается до значения 1200.

Предложение **Priority** управляет приоритетом показа подписей на карте; первым слоем, перечисленным в списке в этом операторе, является слой с наивысшим приоритетом подписей. Например, для карты с четырьмя слоями, не считая косметического, стандартный приоритет подписей такой: 4, 3, 2, 1 – слой 4 обладает старшим приоритетом.

В этом операторе могут быть перечислены идентификаторы любых слоёв карты, но перечислять их все необязательно. Идентификатор слоя, с которым выполняется работа, перемещается наверх списка приоритетов подписей. Порядок перечисления идентификаторов остальных слоёв остаётся неизменным. (Это полезно при перегруппировке слоёв, например,

```
Set Map Order 3
```

для того чтобы переместить 3-ий слой наверх списка.)

Порядок перечисления слоёв (перемещение слоя и группировка слоя в определенном месте списка слоёв) влияет на приоритет подписей; слой выше в списке имеет более высокий приоритет подписей. Однако, настройка приоритета подписей переопределяет порядок слоёв. После настройки приоритета подписей, изменение порядка перечисления слоёв не оказывает влияния на приоритет подписей.

Set Map LabelPriority Default – восстанавливает список приоритетов подписей к стандартному варианту от самого нижнего слоя к самому верхнему, который совпадает со списком слоёв в окне управления слоями.

Подписи появляются в том порядке, как идут записи в таблице. Чтобы изменить порядок показа подписей, сохраните копию таблицы, отсортируйте подписи – важно, какая подпись окажется первой, – и используйте эту новую таблицу для подписывания вместо исходной таблицы. В таблице, отсортированной в алфавитном порядке по улицам (подобно слою StreetPro Display) может оказаться, что улицы с названиями, начинающимися на буквы А, В или С перекроют названия улиц, начинающихся с последних букв. Таким образом “Aberdeen Street” перекроет “State Highway 177”.

Приоритет подписей имеет обратный порядок относительно списка слоёв, поэтому самый нижний слой имеет самый старший приоритет подписей. При добавлении линий сетки, изначально это самый верхний слой, а подписи на нём принудительно приобретают старший приоритет подписей. Чтобы помочь запомнить это, можно считать, что если имеется слой линий сетки со стандартным приоритетом подписей, то приоритет подписей линий сетки принудительно становится старшим.

Пример:

В следующем примере возможность выбора подписей на всей карте включена и порядок подписей задан так, что слой с идентификатором 2 имеет наивысший приоритет.

```
Set Map Label Selection On Priority 2, 4, 3, 1
```

Управление координатной системой карты

Следующие предложения позволяют управлять координатной системой карты и способом измерений.

Синтаксис

```
Set Map
[ Window window_id ]
[ CoordSys... ]
[ Distance Type { Spherical | Cartesian } ]
[ XY Units xy_unit [
  { Display Decimal {On | Off} |
    Display Grid [ { MGRS | USNG [ Datum datumid ] } ]
  ]
] ]
```

window_id – целочисленный идентификатор окна Карты.

xy_unit – строка, задающая единицу измерения X/Y-координат (например, "m" – метр или "degree" – градус); Если координаты **XY Units** - градусы, предложение **Display Decimal**

определяет использование десятичных градусов. Поставьте **On**, чтобы отображать десятичные градусы или **Off**, чтобы отображать координаты в градусах, минутах или секундах. Используйте **Display Grid**, чтобы использовать единицы армейской системы координат.

datumid – число, представляющее референц-эллипсоид. Результатом его выполнения должно быть одно из следующих значений.

```
DATUMID_NAD27 (62)
DATUMID_NAD83 (74)
DATUMID_WGS84 (104)
```

Примечание: DATUMID_* определены в MapBasic.def. Значения WGS84 и NAD83 трактуются как одинаковые.

Описание

Параметр **CoordSys** задает в окне карты систему координат и проекцию. Синтаксис см. в разделе: **Предложение CoordSys**.

Систему координат MapBasic должен явно задавать **оператор Set CoordSys**, а информацию о ней возвращает **функция SessionInfo()**.

Примечание: Если оператор **Set Map** используется с параметром CoordSys, то координатная система, установленная для программы MapBasic, автоматически приводится в соответствие с системой координат Карты.

Этот пример только изменяет систему координат и единицы карты; система координат MapBasic не затронута:

```
Set Map XY Units "m" CoordSys Earth Projection 8,
33, "m", -55.5, 0, 0.9999, 304800, 0
```

Distance Type принимает значения **Spherical** или **Cartesian**. Все расстояния, длины, периметры и площади для объекта в окне карты могут рассчитываться по одному из двух методов. Обратите внимание, что если система координат окна карты является план-схемой, то вычисления будут по декартовому методу (Cartesian), независимо от настроек, и если система координат окна карты это Широта/Долгота, то вычисления будут осуществляться по сферическому методу (Spherical), независимо от настроек.

Предложение **XY Units** задает координатные единицы положения курсора, показываемые в левом нижнем углу окна Карты. Единицами могут быть градусы ("degree") для измерения широты и долготы или единицы измерения расстояния, например, метры ("m").

Если **XY Units** представлены в градусах, предложение **Display Decimal** укажет следует ли использовать десятичные градусы (**On**) или градусы, минуты, секунды (**Off**). **Display Grid** -

координаты будут отображаться в единицах армейской системы координат, независимо от того как определены **XY Units**.

```
Set Map XY Units "m" Display Grid
Set Map XY Units "degree" Display Grid
Set Map XY Units "degree" Display Decimal On
Set Map XY Units "degree" Display Decimal Off
```

Следующий оператор указывает в качестве единиц системы координат метры:

```
Set Map XY Units "m"
```

Управление растровыми изображениями

Следующие предложения управляет перепроецированием и прорисовкой растрового изображения.

Синтаксис

```
Set Map
[ Window window_id ]
[ Image
  [ Reprojection { None | Always | Auto } ]
  [ Resampling { CubicConvolution | NearestNeighbor } ] ]
```

window_id – целочисленный идентификатор окна Карты.

Описание

Параметр *image* имеет два значения: **Reprojection** и **Resampling**. Любой из них или оба можно использовать в предложении с параметром **image**.

Image Reprojection имеет три варианта использования: **Always**, **Auto** (в этом случае повторная дискретизация выполняется автоматически) и **None**.

- *None* означает, что MapInfo Pro обрабатывает растровые слои, как в версиях до 8.5, приспособлявая векторные слои к растровому слою.
- *Always* означает, что перепроецирование осуществляется всегда; координаты пересчитываются используя точные формулы и пиксели сглаживаются используя методы кубической свертки (cubic convolution) или ближайшего соседства (nearest neighbor).
- *Auto* – определяет, что перепроецирование выполняется на основе сведений о внешнем виде полученного изображения после трансформации к области занятой исходным изображением. Если это был “строгий” прямоугольник, т.е. две стороны были параллельны оси X, а две другие – оси Y, то работает старый код MapInfo Pro, например, для растягивания изображения используются стандартные функции Windows. Это самый быстрый способ отобразить результирующий растр. Если используется другой метод (сжатие не эффективно

из-за нелинейности и/или искажений на полученном изображении), работает новый код перепроецирования.

Image Resampling – представляет два варианта повторной дискретизации: **кубическая свертка (cubic convolution)** и **ближайшее соседство (nearest neighbor)**.

- *CubicConvolution* – кубическая свёртка, метод сглаживания раstra обеспечивает наилучшее “восстановление” значений пикселей, отсутствующих на исходном изображении (из-за дискретности). При этом значение пикселя на полученном изображении вычисляется по окну размером 4x4 вокруг “базового” пикселя на исходном изображении. Координаты (обычно, вещественные) исходного пикселя вычисляются для каждого пикселя получаемого изображения по специальной оптимизированной процедуре. Пиксели внутри окна, описанного выше, определенным образом взвешиваются по мантиссам координат исходного пикселя.
- *NearestNeighbor* – ближайшее соседство, метод сглаживания раstra, при котором значение исходного пикселя просто переносится на новое место.

Настройка слоя карты сетки линий

В этом разделе описывается, как можно настраивать сетку линий. Сетка линий визуально разделяет столбцы и строки в окне **Отчёт**. Можно использовать сетку линий для выравнивания положения рамок в окне **Отчёт**.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
Gridline
[ Spacing
[ [ Horizontal [ x_map ] [ Min x_paper Units x_paper_units ] ]
[ Vertical [ y_map ] [ Min y_paper Units y_paper_units ] ] ] |
[ Auto [ Horizontal x_paper Units x_paper_units ]
[ Vertical y_paper Units y_paper_units ] ] ]
[ Offset value Units paper_units ]
[ Label [ Left | Right | Top | Bottom | None ] [ Directionality [
Compass | Plusminus ] ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – число, обозначающее слой, который будет изменен, отсчитываемое в структуре слоев от 1, которая обозначает самый верхний слой.

X_map – значение в "бумажных" единицах, которое предназначено для задания точного горизонтального шага линий сетки, которое не меняется при изменении масштаба показа.

X_paper – значение в "бумажных" единицах шага горизонтальных линий сетки.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (точки), а также пики.

y_map – значение в "бумажных" единицах, которое предназначено для задания точного вертикального шага линий сетки, которое не меняется при изменении масштаба показа.

y_paper – значение в "бумажных" единицах шага вертикальных линий сетки.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (точки), а также пикси.

Value – значение параметра в "бумажных" единицах, по умолчанию равное 5 пунктам.

paper_units – строка, представляющая "бумажные" единицы измерения: cm (сантиметры), mm (миллиметры), in (дюймы), pt (пункты), а также циперо.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Описание

Предложения **Horizontal** и **Vertical** для шага на карте используется независимо от состояния предложения **Gridline** – задан или не задан автоматический режим выравнивания шага (**Auto**). Если в параметре **Gridline** задан автоматический шаг, то он получит точное значение шага. Он пересчитывает шаг попиксельно, основываясь на актуальном шаге **линий сетки** карты и будет использовать полученное значение при будущих изменениях масштаба показа. Если в параметре **Gridline** не задан автоматический шаг, то горизонтальные и вертикальные значения, заданные параметрами **Horizontal** и **Vertical**, становятся новыми используемыми на карте значениями.

Если в параметре **Min** задаются минимальные значения, то автоматический шаг линий сетки отключается. Если задан параметр **Auto**, то включается режим автоматического шага сетки линий.

Предложение **Auto** задаёт режим автоматического шага. Используйте подклассы **Horizontal** и **Vertical**, чтобы задавать значения шагов в "бумажных" единицах. Функция использует параметры настройки измерения листа и карты при текущем масштабе, чтобы определить шаг линий на карте. Предложение **Auto** поддерживает шаг линий сетки даже при изменении масштаба показа.

Для точного размещения линий сетки по заданному шагу, которое не меняется при изменении масштаба показа, следует использовать значения *x_map* и *y_map* вместо предложения **Auto**. Однако, при этом могут получиться близко расположенные линии сетки, например, когда шаг сетки настроен для работы на уровне города, при изменении масштаба для показа всей страны, и замедляется производительность. Для того чтобы предотвратить получение карты, которой нельзя пользоваться, потому что шаг линий слишком мал при уменьшенном масштабе показа, линии сетки автоматически настраиваются, чтобы дальше отстоять друг от друга, учитывая минимальное значение "бумажных" единиц измерения.

Значение смещения **Offset** – специальное смещение подписей линий сетки от края карты (внутрь). Значение задаётся в "бумажных" единицах и по умолчанию составляет пять (5) типографских пунктов. Это отличается от существующего смещения подписей (Label Offset), которое тоже используется для линий сетки, чтобы смещать подписи относительно самих линий (вверх или вниз от горизонтальных либо вправо или влево от вертикальных линий сетки).

Значение **Label** позволяет задавать, где будут отображаться на карте подписи линий при использовании предложения, управляющим местоположением:

- **Left** – размещает подписи горизонтальных линий сетки в левой части карты.
- **Right** – размещает подписи горизонтальных линий сетки в правой части карты.
- **Top** – размещает подписи вертикальных линий сетки в верхней части карты.
- **Bottom** – размещает подписи вертикальных линий сетки в нижней части карты.
- **None** – гарантирует, что никаких подписей линий сетки не будет отображаться.

Label Directionality применяется к линиям сетки, когда используется координатная система долгота/широта. Используйте предложение **Compass** для подписей направлений компаса, Север (N), Юг (S), Восток (E) или Запад (W), дающих ориентацию к ним, чтобы обозначить полушарие Земли, в котором они находятся. Используйте предложение **Plusminus**, чтобы предварительно добавить знак минуса к подписи, отмечающий полушарие, в котором находится объект карты.

Настройка слоя карты сетки линий

В этом разделе описываются возможности оператора **Set Map Layer**, когда обрабатывается слой сетки линий. Все нижеперечисленные подклассы имеют префиксы:

```
Set Map [Window window_id] Layer layer_id
```

Слой

При работе с линиями сетки:

- `Editable { On | Off }`

Не разрешается и приводит к ошибке.

- `Selectable { On | Off }`

Не разрешается и приводит к ошибке. Все линии сетки имеют собственные стили, которые изменяются единым блоком.

- `Zoom ( min_zoom, max_zoom ) [ Units dist_units ] [ { On | Off } ]`

Разрешается и полезно при не-автоматическом шаге линий сетки.

- `Arrows { On | Off }`

Не разрешается и вызывает ошибку.

- `Centroids { On | Off }`

Не разрешается и вызывает ошибку.

- `Default Zoom`

Не разрешается и вызывает ошибку.

- `Nodes { On | Off }`

Не разрешается и вызывает ошибку.

- `Infect num_inflections [ By Percent ] At color:value [ , color:value ]`

Не разрешается, это работает только с определёнными типами слоёв (например, поверхностями, растрами, WMS, сервера тайлов), а не с линиями сетки. Либо ничего не происходит, либо приводит к ошибке.

- `Contrast contrast_value`

Не разрешается, это работает только с определёнными типами слоёв не с линиями сетки. Либо ничего не происходит, либо приводит к ошибке.

- `Brightness brightness_value`

Не разрешается, это работает только с определёнными типами слоёв не с линиями сетки. Либо ничего не происходит, либо приводит к ошибке.

- `Alpha alpha_value`

```
Translucency translucency_value
```

Разрешено.

Подпись слоя

При работе с линиями сетки:

- `Line { Simple | Arrow | None }`

Не разрешено, по умолчанию задаётся состояние **None** и при попытке задать этот параметр вызывается ошибка.

- `Position [ Center ] [ { Above | Below } ] [ { Left | Right } ]`

Разрешено и работает как и с любой другой подписью.

- `Auto Retry { On | Off }`

Не разрешается и приводит к ошибке.

- `Font ...`

Разрешено.

- `Pen ...`

Разрешено, но поскольку выносок нет – ничего не происходит.

- `With label_expr`

Не разрешается и приводит к ошибке.

- `Parallel { On | Off }`

Разрешено. Подпись вертикальной линии сетки поворачивается на 90 градусов, параллельно линиям сетки.

- `Follow Path [ BestPosition { On | Off } ] [ Percent Over percent ] [ Fallback { On | Off } ]`

Разрешено.

- `Visibility { On | Off | Zoom ( min_vis, max_vis ) [Units dist_unit ] }`

Разрешено и используется для включения или выключения заказных подписей.

- `Auto { On | Off }`

Разрешено. Только включает и выключает автоматические подписи.

- `Overlap { On | Off }`

Разрешено. Перекрытие подписей линий сетки запрещено по умолчанию.

- `PartialSegments { On | Off }`

Не разрешается и приводит к ошибке. Подписи частичных сегментов включены по умолчанию. Подписи размещаются в начале линий, а не в середине.

- `Duplicates { On | Off }`

Не разрешается и приводит к ошибке. Этот параметр включен по умолчанию (разрешая дублирование). В зависимости от координатной системы, местоположения на поверхности Земли и форматирования подписей, подписи горизонтальных линий сетки могут быть такими же как подписи горизонтальных линий сетки. Также можно разрешать подписи на обоих концах линий сетки (левых и правых концах, верхних и нижних концах), что всегда приводит к дублированию. Нельзя выключать дублирование для линий сетки.

- `Max [ number_of_labels ]`

Разрешено.

- `Offset offset_amount`

Разрешено и по умолчанию принято значение равное 2 пунктам (единицы всегда в пунктах). Это смещение подписей от линий сетки; вверх или вниз от горизонтальных либо вправо или влево от вертикальных линий сетки.

- `Default`

Разрешено, но ничего не делает для линий сетки. Операция удаляет заказные подписи и восстанавливает стандартные автоматические подписи, но мы не разрешаем заказные подписи для линий сетки.

- `LabelAlpha alpha_value`

Разрешено.

- `AutoPosition { On | Off }`

`AutoSizes { number_font_sizes | Default }`

`SuppressIfNoFit { On | Off }`

`AutoCallout { On | Off }`

Разрешено для полигонов. Эта опция не влияет на слои линий сетки, которые являются полилиниями.

- `Abbreviation { On | Off } Abbreviate with { field_expression }`

Не разрешается и приводит к ошибке.

- Object

(м все последующие)

Разрешено и обозначает заказные подписи. Нельзя работать с заказными подписями и линиями сетки, такие действия вызывают ошибку.

Управление геолинками

Предложение Layer Activate, которое содержит **оператор Set Map**, позволяет поддерживать несколько определений геолинков. Можно добавлять новые элементы, изменять атрибуты существующих, удалять их и изменять порядок обращения к элементам. О том, как MapInfo Pro поддерживает старый синтаксис, см. раздел **Исключения, обеспечивающие совместимость с предыдущими версиями**.

Назначение

С помощью параметра Activate можно определять новые геолинки. Геолинки используются для того чтобы открывать файлы или новые адреса URL в Интернет-браузере прямо из окна карты.

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Activate LAYER_ACTIVATE_CLAUSES ] ]
```

где LAYER_ACTIVATE_CLAUSES:

```
Using launch_expr [ On { Labels | Objects | Labels Objects } ]
[ Relative Path { On | Off } ]
[ Enable { On | Off } ]
[ Alias expression ]
[ , LAYER_ACTIVATE_CLAUSES ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

launch_expr – это выражение, которое обеспечивает запуск файла, когда активизируется ассоциированный с ним объект.

expression – имя файла или URL.

Описание

Relative Path – относительный маршрут к файлу. например: если таблица

C:\DATA\STATES.TAB содержит геолинки к файлам рабочих наборов, хранящихся в каталоге C:\data. Файл рабочего набора для Нью-Йорка, NEWYORK.WOR хранится в каталоге C:\data\ny и геолинк связан с "NY\NEWYORK.WOR". Установка относительного пути в состояние **ON** сообщает MapInfo Pro, что надо добавить вторую часть к маршрута по которому находится TAB-файл, чтобы получился полный маршрут, и в результате получится строка "C:\DATA\NY\NEWYORK.WOR".

Примечание: Геолинки, определенные как адреса URL, не изменяются перед запуском, независимо от установки относительного пути. Чтобы определить, является ли геолинк адресом URL, используется функция ShellAPI.

Предложение *Enable* задает режимы **On** и **Off**. В режиме **On** геолинк делается активным, а в режиме *Off* геолинк неактивен.

Таким образом, предложение *Enable* позволяет пользователю "отключать" отдельные геолинки, оставляя их определения нетронутыми. (В версиях до 10.0 для отключения геолинка требовалось задать пустое выражение "", тем самым теряя исходное.)

Активный объект – это такой объект в окне карты, который имеет адрес URL или с которым связан ассоциированный файл. Выделение такого объекта инструментом геолинк позволяет получить доступ к связанному с ним URL или файлу. Например, если строка `http://www.boston.com` ассоциирована с точкой на карте, то щелкнув на таком точечном объекте или его подписи, мы открывшем Интернет-браузер, а в нем страницу `http://www.boston.com`. Начиная с версии 10.0, добавлен новый вариант использования предложения "Alias" для геолинков. В предложении Alias задается выражение, которое будет служить текстовым элементом значение которого заменяется реальным именем, задаваемым выражением File Name. В текущей реализации геолинка FileName может быть URL-адресом или именем файла. Выяснилось, что URL могут быть очень длинным, что вызывает сложности при выборе одного из нескольких адресов в предназначенном для этого окошке. Для решения этой проблемы, в GUI добавлено специальное выражение Alias. Специальный параметр Alias MapBasic предназначен для его использования. Если для активного объекта задано несколько возможных адресов геолинков, то объявив в выражении Alias ссылку на правильные адреса, в всплывающем окошке появится псевдоним, а не длинные URL. Если для активного объекта задано несколько возможных адресов геолинков, то объявив в выражении Alias ссылку на правильные адреса, в всплывающем окошке появится псевдоним, а не длинные URL.

Примечание: Значение, заданное выражением Alias, появится во всплывающем окошке только в том случае, когда в нем задано любое отличное от стандартного "нет" значение. Таким образом геолинком может быть задан либо псевдонимом, либо нет. Любой геолинк, созданный программой на MapBasic без использования специального параметра Alias,

в графическом интерфейсе будет выглядеть как геолинк без псевдонима (т.е. значение будет "None").

Эта версия команды уничтожит существующее определение, но создаст одно или несколько новых. Следует обязательно использовать параметр **Using**, не допуская пустых строк в *launch_expr* (например, ""). Если используется параметр **Enable**, а его значение установлено в состояние Off, то определение геолинка будет выключено.

Предложения **On**, **Relative**, **Enable** и **Alias** являются необязательными.

Более подробно о геолинках см. разделы [Как добавить новые определения геолинка](#), [Изменение существующего определения геолинка](#), [Как удалять определения геолинка](#) и [Как изменить последовательность определений геолинка](#).

Исключения, обеспечивающие совместимость с предыдущими версиями

Параметр Using можно опустить, но только для первого определения геолинка. Связанное с параметром Using выражение может быть пустым (""), но только для первого определения геолинка.

Без параметра Using

В обеих следующих командах опущен параметр *Using*, что в случае использования версии 850 можно использовать для обновления свойств единственного определения геолинка, даже если параметр *Using* никогда не задавался. В версии 900 такие команды могут вызвать проблемы, поскольку слои карт создаются вообще без определений геолинка.

- Activate On Objects
- Activate Relative Path On

В качестве решения проблемы, MapInfo Pro позволяет использовать пустое выражение, но только в первом определении геолинка. Как и в версиях предшествующих 900, геолинк с пустым выражением будет отключен. Если опустить предложение *Using*, будет порождена ошибка, если только в программе или Рабочем наборе не будет заранее объявлена версия до 900.

Пустой параметр Using

Следующая команда создает выражение с пустой строкой, что по существу отключает геолинк на слое. Если выражение, включенное в геолинк, является пустой строкой, то определение геолинка не используется, до тех пор пока оно не будет заменено непустой строкой. В версии 850 так можно было включить/выключить геолинк. В версии 900 используется включение/выключение геолинка в явном виде командой Set Map Layer Activate Enable On/Off, поэтому использования пустых строк следует избегать.

```
Set Map Layer 1 Activate Using ""
```

Допускается использование пустого выражения, но только в первом определении геолинка. Как и в версиях, предшествующих 9.0, геолинк с пустым выражением будет отключен.

Если в команде Set Map Layer Activate нет параметра Using или параметр Using пустой, то результат зависит от состояния геолинков слоев.

В таблице ниже перечислены примеры использования разных сценариев.

Действие	Количество геолинков	Результат
Activate Using " "	Один или несколько	Первое определение геолинка состоит из пустой строки. Определение отключено до тех пор, пока ему не будет присвоено какое-либо не пустое значение.
Activate Using " "	Ни одного	Будет создан новый геолинк с пустым определением. Определение отключено до тех пор, пока ему не будет присвоено какое-либо не пустое значение.
Activate On Objects	Один или несколько	Если выполняется в приложении или рабочем наборе, созданном ранее версии 9.0: • обновляется первое определение геолинка значением, задаваемым выполняемой командой. Выражение не будет изменено. Если выполняется в приложении или рабочем наборе, созданном в версии 9.0 (или более новой): • появится сообщение о синтаксической ошибке "Пропущен обязательный параметр Using"
Activate On Object	Ни одного	Если выполняется в приложении или рабочем наборе, созданном ранее версии 9.0: • Будет создан новый геолинк с первым пустым определением и остальными значениями, задаваемыми командой. Определение будет отключено до тех пор, пока ему не присвоят какое-либо непустое значение. Если выполняется в приложении или рабочем наборе, созданном в версии 9.0 (или более новой): • появится сообщение о синтаксической ошибке "Пропущен обязательный параметр Using"

Обратите внимание, что те же действия будут выполняться при чтении метаданных таблицы.

Пример

```
Set Map Layer 1 Activate Using Url1 On Objects Relative Path Off Enable
On, Using Url2 On Objects Relative Path On Enable On
```

Как добавить новые определения геолинка

Следующее предложение добавляет определение геолинка к карте. Описание предложения Activate см. в разделе [Управление геолинками](#).

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Activate Add [First] LAYER_ACTIVATE_CLAUSES ] ]
```

где **LAYER_ACTIVATE_CLAUSES**:

```
Using launch_expr [ On { Labels | Objects | Labels Objects } ]
[ Relative Path { On | Off } ] [ Enable { On | Off } ]
[ Alias expression ] [ ,LAYER_ACTIVATE_CLAUSES ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

launch_expr – не может быть пустой строкой (например, "").

expression – имя файла или URL.

Описание

First – необязательное предложение, помещающее новые объекты в начало списка.

Предложение *Enable* задает режимы **On** и **Off**. В режиме **On** геолинк делается активным, а в режиме **Off** геолинк неактивен.

Примеры

```
Set Map Layer 1 Activate Add Using URL1 On Objects Relative Path On Alias
```

```
URL, Using URL2 On Objects Enabled Off
Set Map Layer 1 Activate Add First Using URL1 On Objects
```

Изменение существующего определения геолинка

Следующее предложение изменяет определение геолинка на карте. Описание предложения Activate см. в разделе: [Управление геолинками](#).

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Activate Modify MODIFY_CLAUSES ] ]
```

где **MODIFY_CLAUSES**:

```
hotlink_id {
  [ Using launch_expr ]
  [ On { Labels | Objects | Labels Objects } ]
  [ Relative Path { On | Off } ] [ Enable { On | Off } ]
  [ Alias expression ] }
[ , MODIFYCLAUSE ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

hotlink_id – целочисленный указатель, с помощью которого можно задать определение геолинка. Необходимо задать хотя бы один *hotlink_id*.

launch_expr – не может быть пустой строкой (например, "").

expression – имя файла или URL.

Описание

Предложение *Enable* задает режимы **On** и **Off**. В режиме **On** геолинк делается активным, а в режиме **Off** геолинк неактивен.

Примеры

```
Set Map Layer 1 Activate Modify 1 Using URL1 On Objects Alias URL, 2
Relative Path Off
```

```
Set Map Layer 1 Activate Modify 2 On Objects, 4 On Labels
Set Map Layer 1 Activate Modify 3 Relative Path On Enable Off
Set Map Layer 1 Activate Modify 2 Enable Off, 3 Enable On
```

Как удалять определения геолинка

Следующее предложение добавляет определение геолинка к карте. Описание предложения Activate см. в разделе [Управление геолинками](#).

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Activate Remove {
    All | hotlink_id [ , hotlink_id, hotlink_id, ... ] } ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

hotlink_id – целочисленный указатель, с помощью которого можно задать определение геолинка. Необходимо задать хотя бы один *hotlink_id*.

Описание

Режим *All* задает определения геолинков, которые будут удалены.

Примеры

```
Set Map Layer 1 Activate Remove 2, 4
Set Map Layer 1 Activate Remove All
```

Как изменить последовательность определений геолинка

Следующее предложение изменяют последовательность определений геолинков на карте. Описание предложения Activate см. в разделе [Управление геолинками](#).

Синтаксис

```
Set Map
[ Window window_id ]
[ Layer layer_id
  [ Activate Order hotlink_id [ , hotlink_id, hotlink_id, ... ] ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – определяет слой Карты; задаётся либо числом типа SmallInt (коротким целочисленным; например, используйте 1 для определения самого верхнего слоя, не считая косметического), либо это строковая переменная, соответствующая имени таблицы, отображенной на Карте.

hotlink_id – целочисленный указатель, с помощью которого можно задать определение геолинка. Необходимо задать хотя бы один *hotlink_id*.

Пример

```
Set Map Layer 1 Activate Order 2, 3, 1
```

Оператор Set Map3D

Назначение

Изменяет настройки существующего окна 3D-карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Map3D
[ Window window_id ]
[ Camera [ Zoom factor | Pitch angle | Roll angle | Yaw angle |
  Elevation angle | Position (x,y,z) | FocalPoint (x,y,z) ] ]
[ Orientation ( vu_1, vu_2, vu_3, vpn_1, vpn_2, vpn_3,
  clip_near, clip_far ) ] ]
[ Light [ Position ( x, y, z | Color lightcolor ] ]
[ Resolution ( res_x, res_y ) ]
[ Scale grid_scale ]
[ Background backgroundcolor ]
[ Refresh ]
```

mapper_creation_string - указывает командную строку, которая создает изображение по гриду.

factor – коэффициент увеличения.

angle – угол, измеряемый в градусах. Горизонтальный угол в диалоге может изменяться от 0 до 360 и вращает карту вокруг центральной точки грида. Вертикальный угол в диалоге изменяется от 0 до 90 и измеряет вращение в вертикальной плоскости от стартовой точки прямо над картой.

res_x, res_y – количество образцов по осям X и Y. Эти значения могут быть увеличены до максимально возможного разрешения для данной поверхности. Разрешение можно увеличивать до тех пор, пока не будет достигнуто максимальное значение размера поверхности по X и Y. Если размер поверхности 200x200, то значения разрешения могут лежать в пределах 200x200. Вы не можете увеличивать разрешение грида, можно изменить только разрешение его изображения.

grid_scale – коэффициент масштабирования грида в направлении оси Z. Значение, большее 1, увеличит топологию в направлении оси Z, а значение, меньшее 1, уменьшит.

backgroundcolor – это цвет, используемый для фона; его задает [функция RGB\(\)](#).

Описание

Оператор **Set Map3D** изменяет настройки уже созданной 3D-Карты. Если исходная таблица, из которой была создана 3D-Карта, была изменена добавлением подписей или изменением геометрии, **Refresh** отражает изменения и пересоздает 3D-Карту с учетом этих изменений.

Camera – определяет позицию и ориентацию камеры.

Pitch – регулирует поворот камеры вокруг оси X

Roll – регулирует поворот камеры вокруг оси Z

Yaw – регулирует поворот камеры вокруг оси Y

Elevation – регулирует поворот камеры вокруг оси X относительно фокальной точки камеры

Position – регулирует позицию камеры/источника света

FocalPoint – регулирует позицию фокуса камеры/источника света

Orientation – регулирует позицию камеры ViewUp (*vu_1, vu_2, vu_3*), ViewPlane Normal (*vpn_1, vpn_2, vpn_3*) и Clipping Range (*clip_near, clip_far*). (Используется для повышения устойчивости точки наблюдения.)

Resolution – это разрешение в направлении X и Y. Эти значения могут быть увеличены до максимально возможного разрешения для данной поверхности. Разрешение можно увеличивать до тех пор, пока не будет достигнуто максимальное значение размера поверхности по X и Y. Если размер поверхности 200x200, то значения разрешения могут лежать в пределах 200x200. Вы не можете увеличивать разрешение грида, можно изменить только разрешение его изображения.

Units – определяет единицы измерения грида. Не надо указывать единицы для поверхностей-графиков, например, для трехмерного представления температурного поля или плотности. Эту настройку надо делать к моменту создания грида. Если со значениями грида

связаны единицы измерения, их следует указывать при создании 3D-карты. Нельзя изменить единицы измерения позднее, используя оператор **Set Map3D** или диалог **Настройки**.

Refresh – создает новую текстуру из исходной таблицы.

Пример

Здесь мы изменим разрешение окна карты-призмы по осям X и Y, масштаб по оси Z (< 1) и изменим цвет фона на желтый.

```
Dim win3D as Integer
Create Map3D Resolution(75,75) Resolution(100,100) Scale 2 Background
RGB(255,0,0)
win3D = FrontWindow()
Set Map3D Window win3D Resolution(150,100) Scale 0.75 Background
RGB(255,255,0)
```

См. также:

[оператор Create Map3D](#), [функция Map3DInfo\(\)](#)

Оператор Set Multitouch

Назначение

MapInfo Professional поддерживает управление жестами, что позволяет двумя пальцами изменять увеличение карты на экране тачскрина. Этот оператор позволяет *включать* или *выключать* режим сенсорного экрана. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Multitouch [ On | Off ]
```

Параметр **On** включает на сенсорном экране распознавание жестов, с помощью которых можно выполнять зуммирование или навигацию по экрану.

Off – отключает жесты на сенсорном экране.

См. также:

[оператор Set Searchpoint Tolerance Touch](#)

Оператор Set Next Document

Назначение

Переподчиняет документальное окно в MapInfo Pro (например, окно карты становится подчиненным или порожденным окном приложения на Visual Basic). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Next Document
{ Parent HWND | Style style_flag | Parent HWND Style style_flag }
```

HWND – целое число, уникальный номер порождающего окна;

style_flag – целочисленный код (смотрите таблицу ниже), задающий стиль окна.

Описание

Этот оператор используется в приложениях интегрированной картографии. Более подробная информации об интегрированной картографии содержится в *Руководстве пользователя MapBasic*.

Для того чтобы создать окно MapInfo Pro, используйте оператор **Set Next Document**, а затем один из операторов: **оператор Map**, **оператор Browse**, **оператор Layout** или **оператор Create Legend**.

Предложение **Parent** используется для задания существующего окна, которое будет считаться порождающим по отношению к окну MapInfo Pro, которое Вы собираетесь создать. Предложение **Style** определяет стиль окна. Если создается окно с документом (такое как карту), то включите оба предложения.

Значение параметра *style_flag* должно быть равно коду из следующей таблицы; коды определены в MAPBASIC.DEF.

Значение style_flag	ID	Воздействие на следующее окно документа:
WIN_STYLE_STANDARD	0	Этот код восстанавливает стиль окна до стандартного вида. Если выполнить оператор Set Next Document Style 1 , то восстановить стиль окна можно оператором Set Next Document Style 0 .
WIN_STYLE_CHILD	1	Следующее окно создается подчиненным.

Значение style_flag	ID	Воздействие на следующее окно документа:
WIN_STYLE_POPUP_FULLCAPTION	2	Следующее создаваемое окно имеет стиль рорир (новое окно) и создается с обычной строкой заголовка.
WIN_STYLE_POPUP	3	Следующее создаваемое окно имеет стиль рорир и строка заголовка показывается в половину обычной ее высоты.

Установка стиля и подчиненности окна срабатывает только для следующего создаваемого окна. После того, как оно создается и к нему будут применены стили и подчиненность, MapInfo Pro восстанавливает стандартные режимы подчиненности и стиля. То есть каждое новое переподчиняемое окно требует предварительного срабатывания оператора **Set Next Document**.

Примечание: Оператор **Create ButtonPad** переустанавливает режимы подчиненности и стиля, однако новые инструментальные панели не переподчиняются.

Этот оператор переподчиняет окна документов. Чтобы переподчинить окна диалогов, используйте оператор **оператор Set Application Window**. Чтобы переподчинить специальные окна типа “Информация”, используйте **оператор Set Window**.

Пример

Программа LEGENDS.MB использует следующие операторы для создания окна тематической легенды в окне карты.

```
Dim win As Integer
win = FrontWindow()
...
Set Next Document
  Parent WindowInfo(win, WIN_INFO_WND)
  Style 1
Create Legend From Window win
```

См. также:

оператор Set Application Window, оператор Set Window

Оператор Set Paper Units

Назначение

Устанавливает единицы измерения, описывающие размеры и положения окон на экране. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Paper Units unit
```

unit – строка с именем единицы линейных измерений (например, "cm" – сантиметры).

Описание

Оператор **Set Paper Units** назначает так называемые "бумажные" единицы.

Это единицы линейных измерений на экране (например, "mm" – миллиметры). Если оператор **Set Paper Units** не участвовал в программе, то по умолчанию используются дюймы ("in").

Некоторые операторы MapBasic (такие, как **Set Window**) включают предложения **Position**, **Width** и **Height**, с помощью которых устанавливают положение, ширину и высоту окон. Если эти предложения не содержат предложение Units, то численные параметры задают размеры в единицах, объявленных ранее оператором Set Paper Units, или в дюймах. Например, **оператор Set Window** `Set Window Width 5` преустанавливает ширину окна. Новая ширина окна зависит от используемых бумажных единиц. Если в MapBasic заданы дюймы в качестве бумажных единиц измерения, то **оператор Set Window** задаст карту шириной в пять дюймов.

Если же единицами измерения выбраны сантиметры, то **оператор Set Map** задаст карту шириной в пять сантиметров.

Заметим, что "бумажные" единицы, устанавливаемые MapBasic, являются внутренним атрибутом и не доступны пользователю MapInfo Pro.

В следующей таблице представлены значения параметра **Units**, которые могут использоваться в операторе:

Название единицы измерения листа
Единица измерения
Сравнение единиц измерения

"cm"	Сантиметры	1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
"in"	Дюймы	1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки
"mm"	Миллиметры	1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки
"pt"	Точки	1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
"pica"	Пика (цицero)	1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек

См. также:

[оператор Set Area Units](#), [оператор Set Distance Units](#)

Оператор Set Path

Назначение

Позволяет пользователю программно изменить маршрут к специальному каталогу MapInfo Pro, заданному в диалоге Настройки. Данный каталог содержит специальные файлы MapInfo. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Path current_path_id path
```

current_path_id принимает следующие значения:

```
PREFERENCE_PATH_TABLE (0)
PREFERENCE_PATH_WORKSPACE (1)
PREFERENCE_PATH_MBX (2)
PREFERENCE_PATH_IMPORT (3)
PREFERENCE_PATH_SQLQUERY (4)
PREFERENCE_PATH_THEMETHEMPLATE (5)
PREFERENCE_PATH_MIQUERY (6)
PREFERENCE_PATH_NEWGRID (7)
PREFERENCE_PATH_CRYSTAL (8)
PREFERENCE_PATH_GRAPHSSUPPORT (9)
PREFERENCE_PATH_REMOTETABLE (10)
PREFERENCE_PATH_SHAPEFILE (11)
PREFERENCE_PATH_WFSTABLE (12)
PREFERENCE_PATH_WMSTABLE (13)
```

path – строка, представляющая каталог, который должен содержать файлы определенного типа.

Описание

Оператор **Set Path** позволяет задавать каталог хранения вспомогательных файлов MapInfo определенного типа по заданному ID-номеру. Ниже приводится пример поиска маршрута к каталогу, в который MapInfo помещает новые таблицы.

Пример

```
include "mapbasic.def"
declare sub main
sub main
Set Path PREFERENCE_PATH_WORKSPACE "C:\Temp\"
Print GetCurrentPath$(PREFERENCE_PATH_WORKSPACE)
end sub
```

См. также:

функция [GetPreferencePath\\$\(\)](#)

Оператор Set PrismMap

Назначение

Изменяет настройки существующего окна карты-призмы. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set PrismMap
[Window window_id ]
[ Camera [ Zoom factor | Pitch angle | Roll angle | Yaw angle |
```

Elevation *angle* **Position** (*x,y,z*) |

FocalPoint (*x,y,z*)]]

[**Orientation** (*vu_1, vu_2, vu_3, vpn_1, vpn_2, vpn_3,*

clip_near, clip_far)]]

[**Light** [**Position** (*x,y,z*) | **Color** *lightcolor*]]

[**Scale** *grid_scale*]

[**Background** *backgroundcolor*]

[**Label With** *infotips_expr*]

[**Refresh**]

window_id - это идентификатор окна Карты, которое содержит слой поверхности. Если такой слой не найден, появится сообщение об ошибке.

mapper_creation_string - указывает командную строку, которая создает изображение по гриду.

Camera – определяет позицию и ориентацию камеры.

angle – угол, измеряемый в градусах. Горизонтальный угол в диалоге может изменяться от 0 до 360 и вращает карту вокруг центральной точки грида. Вертикальный угол в диалоге изменяется от 0 до 90 и измеряет вращение в вертикальной плоскости от стартовой точки прямо над картой.

Pitch – регулирует поворот камеры вокруг оси X.

Roll – регулирует поворот камеры вокруг оси Z.

Yaw – регулирует поворот камеры вокруг оси Y.

Elevation – регулирует поворот камеры вокруг оси X относительно фокальной точки камеры.

Position – регулирует позицию камеры/источника света.

FocalPoint – регулирует позицию фокуса камеры/источника света.

Orientation – регулирует позицию камеры ViewUp (*vu_1*, *vu_2*, *vu_3*), ViewPlane Normal (*vpn_1*, *vpn_2*, *vpn_3*) и Clipping Range (*clip_near*, *clip_far*). (Используется для повышения устойчивости точки наблюдения.)

backgroundcolor – это цвет, используемый для фона; его задает **функция RGB()**.

infotips_expr – выражение, используемое для всплывающей подсказки InfoTips.

Refresh – создает новую текстуру из исходной таблицы.

Описание

Оператор **Set PrismMap** изменяет настройки уже созданной карты-призмы.

Пример

Здесь мы изменим разрешение окна карты-призмы по осям X и Y, масштаб по оси Z (< 1) и изменим цвет фона на желтый.

```
Dim win3D as Integer
Create PrismMap Resolution(75,75) Resolution(100,100) Scale 2 Background
RGB(255,0,0)
win3D = FrontWindow()
Set PrismMap Window win3D Resolution(150,100) Scale 0.75 Background
RGB(255,255,0)
```

См. также:

оператор Create PrismMap, **функция PrismMapInfo()**

Оператор Set ProgressBars

Назначение

Показывает или скрывает диалог-индикатор выполнения процесса (диалог Минуточку). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set ProgressBars { On | Off }
```

Описание

Некоторые операторы MapBasic, такие как [Оператор Create Object](#) на стр. 298 оператор `Create Object As Buffer` автоматически выводят диалог, показывающий процент выполнения и кнопку **Отмена**. Оператор **Set ProgressBars Off** используется для подавления диалога-индикатора выполнения процесса. Если диалог не выводится на экран, то пользователь лишается возможности отменить выполнение процесса кнопкой **Отмена**. Оператор **Set ProgressBars On** возобновляет вывод диалогов-индикаторов на экран.

Если оператор **Set ProgressBars Off** выполняется из MapBasic-программы (MBX-файла), то отключается только порожденный MBX-файлом диалог-индикатор. Те диалоги-индикаторы, которые иллюстрируют действия пользователя, не отключаются. Кроме того, [оператор Run Menu Command](#) может игнорировать запрещение показа диалогов-индикаторов, потому что [оператор Run Menu Command](#) симулирует выполнение команд меню пользователем.

Чтобы отключить показ диалога-индикатора, который сопровождает [оператор Run Menu Command](#), выполните оператор **Set ProgressBars Off** в окне **MapBasic** (или пошлите эту команду в MapInfo Pro через механизмы OLE Automation или DDE).

Если приложение свертывает MapInfo Pro в иконку (например оператором `Set Window MapInfo Min`), то диалог-индикатор выполнения нужно отключать, так как в этом случае диалог-индикатор “зависает” до тех пор, пока окно MapInfo Pro не раскроется снова. Если скрыть диалог Минуточку, работа по-прежнему будет выполняться, даже если окно MapInfo Pro минимизировано.

См. также:

[оператор ProgressBar](#), [оператор Run Menu Command](#)

Оператор Set Redistricter

Назначение

Изменяет состав и характеристики районов. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис 1 (Изменение)

```
Set Redistricter districts_table
[ Change district_name
  [ To new_district_name ] [ Pen... ] [ Brush... ] [ Symbol... ] ]
[ Add new_district_name [ Pen... ] [ Brush... ] [ Symbol... ] ]
[ Remove district_name ]
```

Синтаксис 2 (Порядок)

```
Set Redistricter districts_table
Order { "Alpha" | "MRU" | "Unordered" }
```

Синтаксис 3 (Проценты)

```
Set Redistricter districts_table
Percentage from { column | row }
```

Синтаксис 4 (Изменяемый район)

```
Set Redistricter districts_table
Target district_name
```

Синтаксис 5 (Выборка)

```
Set Redistricter districts_table
Selection { As | To } Target
```

districts_table – имя таблицы районов (например, Districts);

district_name – строка с именем открытого окна районирования;

new_district_name – строка с новым именем для открытого окна районирования (используется при добавлении или переименовании района).

Pen – корректное [предложение Pen](#), задающее стиль линии. Например, **Pen MakePen** (*width*, *pattern*, *color*).

Brush – корректное [предложение Brush](#), задающее штриховку. Например, **Brush MakeBrush** (*pattern*, *forecolor*, *backcolor*).

Symbol – корректное [предложение Symbol](#), задающее символ для точечного объекта. Например, **Symbol MakeSymbol** (*shape*, *color*, *size*).

Описание

Оператор **Set Redistricter** изменяет состав районов. Процедуру районирования начинает оператор **Create Redistricter**. Правила работы с районами подробно описаны в документации MapInfo Pro.

Первый вариант синтаксиса используется для добавления, роспуска и изменения района. Предложение **Change** изменяет название и/или стиль оформления районов. Предложение **Add** добавляет новый район, а предложение **Remove** распускает существующий, при этом освободившиеся объекты переходят в район "остальные".

Параметры *district_name* и *new_district_name* должны быть строковыми константами или выражениями, даже если колонка районов численная. Например, если район представлен числом 33, то параметр должен задаваться строкой "33".

Для сортировки строк в окне районирования используется второй вариант синтаксиса оператора. Ключевое слово "Alpha" задает сортировку в алфавитном порядке. Ключевое слово "MRU" используется, если Вы хотите, чтобы последняя группа, с которой Вы работали, автоматически становилась первой в списке районов. В режиме "Unordered" все новые районы добавляются в конец списка.

Для задания изменяемого района используется вариант синтаксиса 4. С помощью пустой строки " " можно назначить изменяемым пустой район.

Чтобы сделать изменяемым район, содержащий выбранный объект, используйте Синтаксис 5 с ключевым словом **As**. Это сработает только тогда, когда выбран единственный объект, и этот объект принадлежит таблице-источнику. Это все равно, что выполнить команду Выбрать изменяемый район на карте из меню Районирование, за исключением того, что команда применяется к выборке.

Чтобы включить выборку в изменяемый район, используйте Синтаксис 5 с ключевым словом **To**. Это сработает только если выбранные объекты принадлежат таблице-источнику. Это все равно, что выполнить команду Добавить выборку к району из меню Районирование, за исключением того, что команда применяется к выборке.

Примеры

Следующий оператор создает новый район:

```
Set Redistricter Districts
Add "NorthWest" Brush MakeBrush(2, 255, 0)
```

Следующий оператор переименовывает район "NE" в с в ходе процедуры районирования. Изначально любые строки из района "NE" содержат "NE" в соответствующей колонке После применения оператора **Set Redistricter... Change**, запись в каждой из тех строк меняется на "NorthWest".

```
Set Redistricter Districts
Change "NE" To "NorthEast"
```

Следующий оператор удаляет район "NorthWest" из таблицы DISTRICTS:

```
Set Redistricter Districts
Remove "NorthWest"
```

Следующий оператор задает упорядочивание строк в списке районов по дате их использования:

```
Set Redistricter Districts
Order "MRU"
```

Следующие операторы делают район "NorthEast" изменяемым:

```
Set Redistricter Districts
Target "NorthEast"
```

Следующие операторы делают изменяемым тот район, в котором содержится провинция "Alberta":

```
Select * from CANADA Where Province_Name = "Alberta"
Set Redistricter Districts
Selection As Target
```

Следующие операторы включают в изменяемый район те провинции, население которых в 1994 году не достигало 100,000 человек:

```
Select * from CANADA Where Pop_1994 < 100000
Set Redistricter Districts
Selection To Target
```

См. также:

[оператор Create Redistricter](#)

Оператор Set Resolution

Назначение

Устанавливает параметр графического разрешения для операций изменения типа объекта. Эта характеристика влияет на количество узлов в объекте, полученном преобразованием типа объекта. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Resolution node_limit
```


node_limit – целое число типа SmallInt от 2 до 32 762 включительно; по умолчанию 100.

Описание

По умолчанию MapInfo Pro создает 100 узлов на окружности или дуге при преобразовании их в область и полилинию. Оператор **Set Resolution** устанавливает число узлов для преобразования окружности в область. Прирост значения разрешения приводит к более гладким результатам.

Оператор **Set Resolution** влияет на результаты таких команд, как **Превратить в области** и **Превратить в полилинии**. Значение разрешения влияет также на результаты некоторых операторов и функций MapBasic, таких как [функция ConvertToRegion\(\)](#) и [функция ConvertToPline\(\)](#). Кроме этого, от значения разрешения зависят результаты операций, в которых конвертирование производится автоматически (например, Objects Split, Combine).

Установка оператора **Set Resolution** не влияет на создание буферной области. **Оператор Create Object As Buffer** и [функция Buffer\(\)](#) имеют обязательный параметр, явно задающий разрешение для создания области.

См. также:

[функция ConvertToPline\(\)](#), [функция ConvertToRegion\(\)](#)

Оператор Set Shade

Назначение

Изменяет тематический слой карты. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Shade
[ Window window_id ] { map_layer_id | "table ( theme_layer_id )" }
[ Style Replace { On | Off } ]
...
```

window_id – идентификатор окна.

map_layer_id – короткое целое число (SmallInt), задающее номер тематического слоя.

table – имя таблицы, на которой основывается тематический слой.

theme_layer_id – короткое целое число (SmallInt) от 1 и больше, задающее номер изменяемого тематического слоя (например, 1 используется для первого созданного слоя).

Описание

После того как **оператор Shade** создаст тематический слой карты, с помощью оператора **Set Shade** можете изменять этот слой. Применение оператора **Set Shade** аналогично выполнению команды **Настройка тематики** (на вкладке **ЛЕГЕНДА**). Синтаксис операторов **Set Shade** и **оператора Shade** почти одинаковы, за исключением первых ключевых слов **оператора Set Shade** и параметров, задающих слой тематического выделения. Оператор **Set Shade** может идентифицировать слой по его номеру, как в следующем примере...

```
Set Shade
Window i_map_winid
2
With Num_Hh_90
Graduated 0.0:0 11000000:24 Vary Size By "SQRT"
```

...или по имени таблицы, на данных которой основано тематическое выделение (в скобках указывается номер созданного тематического слоя):

```
Set Shade
Window i_map_winid
"States(1)"
With Num_Hh_90
Graduated 0.0:0 11000000:24 Vary Size By "SQRT"
```

"STATES(1)" означает первый слой из тематических, основанный на данных таблицы STATES

Style Replace On запрещает отображение слоёв под тематическим слоем.

Style Replace Off предписывает отображать слои под тематическим слоем, что позволяет создавать многовариантные прозрачные тематические слои.

Style Replace On – это режим по умолчанию, обеспечивающий обратную совместимость с существующими настройками, так что по умолчанию нижележащие слои не отображаются.

См. также:

оператор Shade

Оператор Set Searchpoint Tolerance Touch

Назначение

Задаёт уровень чувствительности, порог срабатывания сенсорного экрана. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Searchpoint Tolerance Touch { val }
```

Val – задаёт значение уровня чувствительности сенсорного экрана. Диапазон от 3 до 50

Описание

Используйте оператор **Set Searchpoint Tolerance Touch**, чтобы задать значение уровня чувствительности сенсорного экрана. Этот оператор действует только на порог чувствительности, который используется [функцией SearchPoint\(\)](#), и меняет область, чувствительную к касаниям, при реализации интерактивных запросов на сенсорном экране с помощью [функции SearchPoint\(\)](#). Стандартным значением этого параметра чувствительности является 15. Этот оператор можно использовать, чтобы менять значение параметра чувствительности в диапазоне от 3 до 50.

Оператор Set Style

Назначение

Изменяет текущие стили объектов типа Pen, Brush, Symbol или Font. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Style
{ Brush... | Font... | Pen... |
  BorderPen | LinePen | Symbol... }
```

Font – корректное [предложение Font](#), задающее стиль текста.

Pen – корректное [предложение Pen](#), задающее стиль линии.

Brush – корректное [предложение Brush](#), задающее штриховку.

Symbol – корректное [предложение Symbol](#), задающее символ для точечного объекта.

BorderPen начинает [предложение Pen](#), определяющее стиль границы региона.

LinePen начинает [предложение Pen](#), определяющее стиль линии.

Описание

Оператор **Set Style** изменяет текущие стили **Pen**, **Brush**, **Symbol** или **Font**.

Предложение Pen устанавливает оба стиля – для линий и для границ региона. Чтобы установить их отдельно, используйте предложение **LinePen** для установки стиля линий и предложение **BorderPen** для установки стиля границы региона. Когда пользователь рисует новый графический объект в окне карты или отчёта, MapInfo Pro создает объект используя текущие стили **Font**, **Pen**, **Brush** и/или **Symbol**.

Пример

Пример изменения стилей **Brush**, **Symbol**, и **Font**:

```
Include "mapbasic.def"
Set Style Brush MakeBrush(64, CYAN, BLUE)
Set Style Symbol MakeSymbol( 9, BLUE, 14)
Set Style Font MakeFont("Arial", 1, 14, BLACK,WHITE)
```

Пример изменения стиля **Pen**:

В этом примере линия и граница региона красного цвета.

```
Include "mapbasic.def"
Set Style Pen MakePen(3, 9, RED)
```

Пример изменения стиля **LinePen** и **BorderPen**:

В этом примере линия красная а граница региона зеленая.

```
Include "mapbasic.def"
Set Style LinePen MakePen(6, 77, RED)
Set Style BorderPen MakePen(6, 77, GREEN)
```

См. также:

[функция CurrentBrush\(\)](#), [функция CurrentFont\(\)](#), [функция CurrentPen\(\)](#), [функция CurrentSymbol\(\)](#), [функция MakeBrush\(\)](#), [функция MakeFont\(\)](#), [функция MakePen\(\)](#), [функция MakeSymbol\(\)](#), [функция RGB\(\)](#)

Оператор Set Table

Назначение

Изменяет некоторые режимы работы с открытой таблицей. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Table tablename
[ FastEdit { On | Off } ]
[ Undo { On | Off } ]
[ ReadOnly ]
[ Seamless { On | Off } [ Preserve ] ]
[ UserMap { On | Off } ]
[ UserBrowse { On | Off } ]
[ UserClose { On | Off } ]
[ UserEdit { On | Off } ]
[ UserRemoveMap { On | Off } ]
[ UserDisplayMap { On | Off } ]
[ Persist { On | Off } ]
[ Datum datum_number ]
[ Description layer_name ]
[ InUse { ReadAccess | WriteAccess | Off } unique_token ]
[ Redraw { On | Off } ]
[ Adopt Transaction ]
```

table — строка, указывающая имя таблицы.

Layer_name — понятное имя слоя.

Unique_token — строка, которая будет повторно использоваться, когда разметка страницы не будет использоваться.

Adopt Transaction позволяет принимать файлы транзакции, созданные MapXtreme.NET для таблицы, открытой в MapInfoPro.

Описание

Оператор **Set Table** устанавливает, можно ли изменять данные таблицы и как это сделать. Оператор **Set Table** может задать открытой таблице режим "только чтение", и пользователь не сможет внести в эту таблицу никакие изменения. Оператор **Set Table** может активизировать и выключать специальный режим редактирования, который выключает механизм защиты исходных данных таблицы ради ускорения выполнения действий правки.

datum — записывает информацию о топоцентрической проекции (датуре) в MAP-файл, включая номер эллипсоида, 3 параметра сдвига, 3 параметра поворота и коэффициент масштабирования. Так как некоторые топоцентрические проекции (датуры) совпадают, то этот оператор хранит номер топоцентрической проекции (датуры) вместе с параметрами в памяти, а затем записывает их в MAP-файл.

Предложение **Persist Off** указывает, что таблица не будет сохраняться в рабочем наборе при сохранении рабочего набора. Предложение **Persist** или **Persist On** указывает, что таблица, которая раньше была указана с предложением **Persist Off**, теперь будет записываться в рабочий набор при сохранении рабочего набора.

Description — присваивает слою стандартное «понятное» имя, которое задаётся из параметра **Description** (входит в раздел **Definition**) TAB-файла, который открыт в MapInfo Pro. Для того чтобы удалить параметр **Description** TAB-файла, можно воспользоваться пустой строкой:

```
Set table tablename description ""
```

С помощью параметра **InUse** можно защитить таблицу, чтобы её нельзя было закрыть или изменить во время работы с ней. Это полезно, когда над таблицами выполняются фоновые задачи, чтобы нельзя было случайно внести в них изменения, до тех пор пока они не будут помечены как неиспользуемые. Параметры-токены **ReadAccess** и **WriteAccess** позволяют задавать тип операции доступа к используемой таблице. Параметр-токен **Off** помечает таблицу как неиспользуемую. Для того чтобы пометить таблицу как используемую, необходимо использовать параметр с уникальной строкой-токеном — *unique_token*, такая же строка требуется, чтобы пометить таблицу как неиспользуемую. Для того чтобы отметить таблицу как используемую, можно воспользоваться атрибутом TAB_INFO_INUSE **оператора Set Table**. Следует принять во внимание следующее:

- Таблицу можно пометить как используемую с правом записи только один раз. Право записи является эксклюзивным, таблицу можно повторно пометить как используемую, только после того как она была помечена как неиспользуемая.
- Помечать таблицу как используемую с правом чтения можно несколько раз для разных заданий, используя разные строки-токены.
- Когда в нескольких заданиях таблица помечена как используемая, то, чтобы эта таблица стала рассматриваться как действительно неиспользуемая, необходимо во всех таких заданиях пометить таблицу как больше неиспользуемую, .
- Таблицу, в которой имеются несохраненные изменения, нельзя пометить как используемую. Необходимо сначала либо сохранить изменения, либо отказаться от них и удалить.
- Таблицы, предназначенные только для чтения и защищённые от записи нельзя пометить как используемые с правом записи.
- Таблицы, находящиеся в режиме быстрого правки, нельзя помечать как используемые с правом записи.
- Сшитые и растровые таблицы нельзя пометить как используемые.
- Если запрос помечен как используемый, то базовая таблица, на основе которой он построен, считается используемой, при этом все зависимые таблицы (запросы) нельзя изменять, но можно закрыть.
- Если объединение таблиц помечено как используемое, то все базовые таблицы считаются используемыми, при этом все зависимые таблицы (запросы) нельзя изменять, но можно закрыть.

Режим быстрой правки *FastEdit*

Обычно, всякий раз, когда таблица редактируется (все равно, кем — пользователем или прикладной программой), то MapInfo Pro не записывает изменения сразу в таблицу. Вместо этого MapInfo Pro помещает информацию об изменениях во временный файл, который называется файлом транзакций. Записывая изменения в файл транзакций вместо того, чтобы

вносить их непосредственно в таблицу, MapInfo Pro дает возможность пользователю удалить эти изменения (например, командой **Восстановить**).

Если выполнить оператор **Set Table** с предложением **FastEdit On**, то MapInfo Pro будет записывать изменения непосредственно в таблицу, минуя файл транзакций. Операции правки таблицы **FastEdit** в таком режиме будут производиться быстрее.

Пока включен режим **FastEdit**, таблица изменяется немедленно и не требуется выполнять **оператор Commit Table** для фиксации изменений на диске. Следует осторожно пользоваться режимом быстрой правки **FastEdit**, нельзя отказаться от изменений, внесённых в этом режиме, командами **Закреть** или **Восстановить таблицу**.

Режим **FastEdit** можно включать только для нормальных, базовых таблиц; нельзя включать режим **FastEdit** для временных таблиц и таблиц-запросов, например, Query1. Нельзя включать режим **FastEdit** для таблицы, в которой уже существуют несохранённые изменения. Также нельзя устанавливать режим редактирования **FastEdit** для связанных таблиц.

Осторожно: Пока открытая таблица редактируется в режиме FastEdit, другие пользователи сети не могут ее открыть. После того, как выполнены все изменения в режиме FastEdit, выполните **оператор Commit Table** или **оператор Rollback** для установки состояния таблицы, чтобы она была доступна другим пользователям сети.

Ключевое слово **ReadOnly** в операторе включает режим "только чтение" для таблицы, так что пользователь не сможет изменять таблицу вплоть до конца сеанса в MapInfo Pro. Оператор **Set Table** не может выключить режим "только чтение". Режим **ReadOnly** также можно установить для таблицы при ее открытии, поместив его в **оператор Open Table**.

Предложение **Undo On** устанавливает режим работы с таблицей, при котором MapInfo Pro запоминает всю информацию об изменениях, позволяя пользователю применять команду **Отменить**. Если было использовано в операторе **Set Table** предложение **Undo Off**, то MapInfo Pro не запоминает информацию о последних изменениях в таблице. В последнем режиме экономятся ресурсы компьютера, а операции редактирования таблицы выполняются существенно быстрее.

Управление сшитыми таблицами

В сшитой таблице группируются несколько таблиц в единое целое. Концепция сшитых таблиц подробно описана в документации MapInfo Pro.

Предложение **Seamless** устанавливает или отменяет атрибут сшитости для таблицы. Режим **Seamless Off** открывает таблицу, входящую в группу сшитых, для редактирования. Режим **Seamless On** восстанавливает атрибут сшитости. Ключевое слово **Preserve** сохраняет режим; то есть MapInfo Pro записывает режим в таблицу. Без слова **Preserve** смена режима действует только до конца сеанса работы.

Предложения **User...** позволяют выборочно запрещать пользователю применять определенные операции к таблице. Это полезно, если требуется запретить пользователю открывать, закрывать или изменять определенные таблицы или окна.

Предложения **UserMap Off** ограждают таблицы только от действий пользователя, но не от операторов программ MapBasic.

Примечание: Также, эти предложения не действуют на косметический слой.

Пример	Эффект
UserMap Off	Таблица не появится в окне диалоге Новое окно Карты и Добавить слой .
UserBrowse Off	Таблица не появится в окне диалоге Новое окно Списка .
UserClose Off	Таблица не появится в окне диалоге Заккрыть таблицу .
UserEdit Off	Пользователь не может редактировать таблицу: окна списка и информации не редактируются и соответствующий слой карты невозможно сделать изменяемым.
UserRemoveMap Off	Когда эта таблица появляется в окне Управление слоями, кнопка Удалить для нее неактивна.
UserDisplayMap Off	Когда эта таблица появляется в окне Управление слоями, кнопка Видимый Вкл/Выкл для нее неактивна.

Временное отключение перерисовки карты

Предложение **ReDraw** позволяет отключать перерисовку карты, когда в таблицу вносятся изменения, например, при выполнении операций вставки, обновления или удаления. Это полезно, когда обрабатываются несколько таблиц, поскольку это эффективнее, чем отложенное обновление окна **Карта** до завершения обновления таблиц. Это может существенно увеличить общую производительности при обработке в зависимости от сложности карты и действиях по обновлении таблиц.

Предложение **ReDraw** позволяет сворачивать несколько обновлений таблицы, как показано в этом примере:

```
Set Table Test REDRAW OFF
For I = 1 to 1000
  ' Insert | Update | Delete statement against Test
Next
Set Table Test REDRAW ON
```


Примеры

Следующий пример не допускает таблицу World в диалог **Заккрыть** таблицы.

```
Set Table World UserClose Off
```

Пометка таблицы как используемой

Следующий оператор помечает таблицу с именем CN21 как используемую. Используется уникальная строка-токен "cn21abc".

```
Set table CN21 inuse readaccess "cn21abc"
```

Таблица CN21 помечается как используемая с правом записи. Уникальная строка-токен "cn21abc» продолжает использоваться.

```
Set table CN21 inuse writeaccess "cn21abc"
```

После выполнения задания над таблицей CN21 в фоновом режиме, она освобождается и становится доступной для других процессов и пользователей.

```
Set table CN21 inuse off "cn21abc"
```

См. также:

[оператор Set Datum Transform Version](#), [оператор Set Table](#), [функция TableInfo\(\)](#)

Оператор Set Target

Назначение

Назначает изменяемый объект или объекты или отменяет назначение. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Target { On | Off }
```

Описание

Используйте оператор **Set Target**, чтобы задавать или очищать список изменяемых объектов; это соответствует командам MapInfo Pro **Выбрать изменяемый объект** и **Освободить изменяемый объект** на вкладке **ОБЪЕКТЫ**. Некоторые операции в MapInfo Pro предваряются выбором объектов, которые надо изменить. Например, необходимо сначала назначить

изменяемым объект перед тем, как выполнить **оператор Objects Split**. Правила назначения изменяемого объекта описаны в документации MapInfo Pro.

Оператор **Set Target On** соответствует команде **Объекты > Выбрать изменяемый объект**. После выполнения оператора или команды объекты, выбранные на данный момент, становятся изменяемыми. Если не выбрано ни одного объекта оператор генерирует ошибку.

Оператор **Set Target Off** соответствует команде **Объекты > Освободить изменяемый объект**.

См. также:

оператор Objects Combine, оператор Objects Erase, оператор Objects Intersect, оператор Objects Overlay, оператор Objects Split, оператор Set Combine Version

Оператор Set Undo

Назначение

Оператор **Set Undo** позволяет включать или выключать в окне **Конструктор отчёта** возможность отменять операции. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Undo [ Window window_id ] { Enable | Disable }
```

window_id – идентификатор окна **Отчёт**, целое число; Если пропустить дополнительное, необязательное предложение **Window**, то этот оператор применяется к самому верхнему окну **Конструктор отчёта**.

Описание

Оператор **Set Undo** позволяет включать или выключать в окне **Конструктор отчёта** возможность отменять операции. Когда MapBasic-скрипт выполняет большое количество операций изменения отчёта, отключение возможности отмены позволяет сократить использование памяти или увеличить скорость обработки.

По умолчанию, в каждом окне **Конструктор отчёта** можно отменять последние операции дополнения и правки в отчёте. Для того чтобы отключить возможность выполнения отмен операций, требуется выполнить оператор **Set Undo** с ключевым словом **Disable**. Для того чтобы повторно включить возможность выполнения отмен операций, требуется выполнить оператор **Set Undo** с ключевым словом **Enable**.

Примечание: Оператор **Set Undo** не оказывает влияния на операции правки таблицы. О том как отключить отмену операций с таблицами, смотрите описание в разделе:

Оператор Set Table.

Каждое окно **Конструктор отчёта** имеет отдельную историю отмен. Оператор **Set Undo** оказывает влияние только на заданное окно. Для того чтобы отключить возможность отменять в нескольких отчётах, требуется применить несколько операторов **Set Undo**.

Отключение отмен в окне **Конструктор отчёта** очищает историю операций в этом окне. Когда включён режим отмены операций, начинает записываться история операций, которые можно впоследствии отменять.

См. также:

функция UndoInfo() **оператор Set Table**

Оператор Set Window

Назначение

Изменяет состояние, размер и положение окна на экране, управляет принтером, размером бумаги и полями. Этот оператор обновлён — теперь можно использовать улучшенный метод рендеринга, для того чтобы иметь возможность управлять параметрами сглаживания в окнах конструктора отчётов. Это означает, что оператор MapBasic, который управляет настройками сглаживания в окне карты теперь применяется также и в окнах конструктора отчёта. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Window window_id
[ Position ( x, y ) [ Units paper_units ] ]
[ Width win_width [ Units paper_units ] ]
[ Height win_height [ Units paper_units ] ]
[ Font... ]
[ Enhanced { On | Off } ]
[ Smooth [ Vector { None | Antialias } ] [ Text { None | Antialias } ]

[ Image { None | Low | High } ] ]
[ Min | Max | Restore | Floating | Docked | Tabbed | AutoHidden ]
[ Front ]
[ Title { new_title | Default } ]
[ Help [ { File help_file | File Default | Off } [ Permanent ] ]
[ Contents ] [ ID context_ID ] { Show | Hide } ]
[ Printer { Default | Name printer_name }
[ Orientation { Portrait | Landscape } ]
```

```

[ Copies number ]
[ Papersize number ]
[ Border { On | Off } ]
[ TrueColor { On | Off } ]
[ Dither { Halftone | ErrorDiffusion } ]
[ Method { Device | Emf | PrintOsbm } ]
[ Transparency
  [ Raster { Device | ROP } ]
  [ Vector { Device | Internal } ] ]
[ Margins
  [ Left d1 ] [ Right d2 ] [ Top d3 ] [ Bottom d4 ]
  [ Units paper_units ] ] ]
[ Export { Default |
  [ Border { On | Off } ]
  [ TrueColor { On | Off } ]
  [ Dither { Halftone | ErrorDiffusion } ]
  [ Transparency
    [ Raster { Device | ROP } ]
    [ Vector { Device | Internal } ] ]
]
[ Scale Patterns { On | Off } ]
[ Antialiasing { On | Off } ]
[ Threshold threshold_value ]
[ MaskSize size_value ]
[ Filter filter_value ] ] ]
[ ScrollBars { On | Off } ]
[ Autoscroll { On | Off } ]
[ Parent HWND ]
[ ReadOnly | Default Access ]
[ Table table_name Rec record_number ]
[ Show | Hide ]
[ Smart Pan { On | Off } ]
[ SysMenuClose { On | Off } ]
[ Snap [ Mode { On | Off } ] ]
[ Threshold { pixel_tolerance | Default } ]
[ Toolbar { On | Off } ]
[ Topmost { On | Off } ]

```

window_id — целочисленный идентификатор окна или имя специального окна (например, Statistics).

x — расстояние от левого края рабочего поля в окне MapInfo Pro до левого края перемещаемого окна.

y — расстояние от верхнего края рабочего поля окна MapInfo Pro до верхнего края перемещаемого окна.

paper_units — строка, представляющая "бумажные" единицы измерения: "cm" для сантиметров, "mm" для миллиметров, "in" — дюймов, "pt" — пунктов.

- 1 дюйм (in) = 2.54 сантиметра, 254 миллиметра, 6 пик, 72 точки

- 1 точка (pt) = 0.01389 дюйма, 0.03528 сантиметра, 0.35278 миллиметра, 0.08333 пик
- 1 пика = 0.16667 дюйма, 0.42333 сантиметра, 4.23333 миллиметра, 12 точек
- 1 сантиметр (cm) = 0.39370 дюйма, 10 миллиметров, 2.36220 пик, 28.34646 точки
- 1 миллиметр (mm) = 0.1 сантиметра, 0.03937 дюйма, 0.23622 пик, 2.83465 точки

Параметром этого оператора **Font** можно задать стиль оформления текста.

win_width — ширина окна.

win_height — высота окна.

new_title — строка, задающая новый заголовок окна.

help_file — имя файла Справки (например, в Windows "FILENAME.HLP").

context_ID — целочисленный идентификатор контекста Справочника для задания раздела.

printer_name — имя принтера. Принтер может быть локальным или сетевым.

number — количество копий, которое будет передано на печать.

HWND — целочисленный номер окна. Окно с номером *HWND* станет порождающим окном по отношению к окну легенды, Статистики, Информации, Линейки или Сообщений.

table_name — имя открытой таблицы для показа в окне сообщений.

record_number — целое число: значение от 1 и больше для показа определенной записи в окне Информации или 0 для показа сообщения "Нет записей".

Enhanced — задает улучшенный режим прорисовки и печати графики.

При использовании параметра *On*, включается улучшенный режим рендеринга, также как с помощью флажка **Включить визуальные &эфффекты** MapInfo Pro. При выборе параметра *Off* используются старые технологии показа, которые соответствуют выбору иного, чем **Включить визуальные &эфффекты** режима сглаживания MapInfo Pro.

Smooth задает режим прорисовки векторных, текстовых объектов, подписей и растров, при котором сглаживаются края.

Примечание: Режимы сглаживания (*Vector*, *Text* и *Image*) требуют установки режима *Enhanced* в положение *On*. MapBasic сгенерирует ошибку, если наряду с заданием режима *Enhanced Off* будет задан режим сглаживания, отличный от *None*.

Vector — задает режимы сглаживания для векторных объектов.

None — указывает, что сглаживание к векторным объектам, линиям и границам областей не применяется.

Antialias — означает, что применяется сглаживание к векторным объектам. Этот режим требует установки режима *Enhanced* в положение *On*.

Text — задает сглаживание текста подписей, подписей по кривой и текстовых объектов.

Параметр *None* указывает, что сглаживание к повернутым и горизонтальным подписям не применяется.

Параметр **Antialias** означает, что применяется сглаживание к повернутым и горизонтальным подписам и текстовым объектам. Этот режим требует установки режима **Enhanced** в положение **On**.

Image — задает режимы сглаживания для растров.

None — указывает, что сглаживание к растрам не применяется.

Low — означает, что включен режим сглаживания для растровых объектов и задействован алгоритм билинейной интерполяции. Этот режим показывает растры в лучшем качестве, чем в режиме **None**, но хуже, чем в режиме **High**. В режиме **Low** растр отрисовывается медленнее, чем в режиме **None**, но быстрее, чем в режиме **High**. Этот режим требует установки режима **Enhanced** в положение **On**.

High — означает, что включен режим сглаживания для растровых объектов и задействован алгоритм бикубической интерполяции. Этот режим показывает растры в лучшем качестве, чем в режиме **Low**, но работает медленнее. Этот режим требует установки режима **Enhanced** в положение **On**.

Printer — будет указывать окно, предназначенное для печати.

Export — будет указывать окно, предназначенное для экспорта.

Default — будут использоваться стандартные настройки печати/экспорта.

Name printer_name — определяет имя используемого принтера.

Orientation Portrait — (книжная) располагает бумагу для печати в книжной ориентации.

Orientation Landscape — (альбомная) располагает бумагу для печати в альбомной ориентации.

Copies number — указывает число копий для печати.

Papersize number — информация о размере бумаги для данного окна. Эти числа (номера) универсальные для всех принтеров под Windows. Например, 1 соответствует размеру Letter, 5 соответствует размеру Legal. Число можно найти в служебном MapBasic-файле `PAPERSIZE.DEF`. Драйверы некоторых принтеров (например, плоттеры больших форматов) могут использовать другой порядок присвоения чисел для определения размеров листов бумаги, отличающийся от того, который задан в MapBasic-файле определителей "PaperSize.def". По этой причине, пользователи, использующие разные драйверы принтера, возможно, не смогут правильно идентифицировать информацию о размере бумаги, хранящуюся в рабочем наборе. В этом случае будет установлен стандартный для этого принтера размер бумаги.

Border — определяет, будет ли отображаться черная рамка вокруг окна при печати и экспорте.

Truecolor — определяет, будет ли создан 24-битный вывод с цветом True Color, если это вообще возможно. Если **Truecolor** отключен, вывод будет с 256 цветами.

Dither — определяет, какой метод растеризации будет использован, если требуется конвертировать 24-разрядное изображение в 256 цветов. Эта настройка используется при печати растров и файлов поверхностей. Растеризация произойдет, если режим **Truecolor** отключен или выводное устройство не может поддерживать 24-разрядные цвета.

Method — это новое ключевое слово определяет, будет ли печать проведена непосредственной передачей файла на принтер, или MapInfo Pro создаст промежуточный Windows Enhanced Metafile и только потом пошлёт его на принтер. Ранние версии MapInfo Pro всегда посылали файл прямо на устройство. Новый метод позволяет существенно сократить размер задания на печать.

Transparency Raster Internal — удалено в версии 7.0; таким образом, можно работать с данными предыдущих версий без сообщения об ошибке.

Transparency Raster — определяет, как будут отображаться полупрозрачные пиксели. Выберите **Device** или **ROP** в зависимости от драйвера принтера или формата экспортного файла. Сначала попробуйте оба способа, сравните и сделайте выбор.

Transparency Raster ROP соответствует опции **Использовать ROP-метод для отображения прозрачного раstra** в интерфейсе MapInfo Pro (диалог **Параметры вывода** и кнопка **Дополнительно** в диалогах **Печать** и **Экспорт окна в файл**). Если выбран **ROP**, то полупрозрачное изображение прорисовывается с использованием растровой операции (ROP), обрабатывающей полупрозрачные пиксели. Этот метод используется для обработки полупрозрачного изображения на экране; однако он не всегда хорошо работает при печати. Проведите эксперимент и посмотрите, как драйвер принтера обрабатывает методом ROP. Если требуется экспортировать изображение, используя команду **Экспорт окна**, то лучше выбрать формат метафайла (EMF или WMF). Метод ROP позволяет обрабатывать данные с подложки в их изначальной форме.

Transparency Raster Device — MapInfo Pro не будет осуществлять специальные преобразования при выводе растров и гридов, содержащих прозрачность. Изображение генерируется тем же методом, что применяется для отображения на экране. При этом не исключены некоторые проблемы при генерации вывода.

Transparency Vector Internal — MapInfo Pro будет делать специальную обработку при выводе прозрачных шаблонов заливки или прозрачных растровых символов.

Transparency Vector Device — MapInfo Pro будет делать специальную обработку при выводе прозрачных шаблонов заливки или прозрачных растровых символов. При этом не исключены некоторые проблемы при генерации вывода.

Margins User — пользователь может установить поля для принтера в виде вещественных значений в требуемых единицах. Эти значения могут быть увеличены драйвером принтера если они меньше, чем физически допустимые поля для данного принтера.

Antialiasing — определяет, используется ли сглаживание при экспорте раstra. Режим **Antialiasing** игнорируется, если растр экспортируется в форматы EMF или WMF, а также при экспорте содержимого окна Списка.

Threshold — указывает значение, определяющее, какие пиксели будут сглаживаться. Применение фильтра сглаживания к изображению связывает значения с каждым пикселем. Сглаживаются только пиксели со значением больше *threshold_value*. Если значение параметра *threshold_value* задано равным нулю, то все пиксели сглаживаются. *threshold_value* должен иметь значение от 0 до 255.

MaskSize — число, определяющее размер маски сглаживания. Например, значение равное трём означает маску сглаживания 3x3. Если значение параметра маски *size_value* слишком большое, то изображение становится размытым.

Filter — указывает, какой сглаживающий фильтр будет применён. В настоящее время MapInfo Pro поддерживает 6 различных фильтров, приведенных в таблице ниже.

Фильтр	ID	Описание
FILTER_VERTICALLY_AND_HORIZONTALLY	0	Сглаживание раstra по вертикали и горизонтали.
FILTER_ALL_DIRECTIONS_1	1	Сглаживание раstra во всех направлениях
FILTER_ALL_DIRECTIONS_2	2	Сглаживание во всех направлениях. Этот фильтр отличается от фильтра FILTER_ALL_DIRECTIONS_1 и даёт лучшие результаты при сглаживании текста.
FILTER_DIAGONALLY	3	Диагональное сглаживание раstra.
FILTER_HORIZONTALLY	4	Горизонтальное сглаживание раstra
FILTER_VERTICALLY	5	Вертикальное сглаживание раstra

Toolbar регулирует показ или скрытие инструментальной панели в верхней части окна. Это относится только к окнам **Список**. Например:

```
Set Window windowId Toolbar Off
```

Topmost — позволяет задавать самое верхнее положение окна Проводник.

Описание

Оператор **Set Window** используется для изменения размеров и положения окна, шрифта в окне, заданном параметром *window_id*.

Этот оператор можно использовать, для того чтобы скрывать панель инструментов вверху окна **Легенда**.

Целочисленное значение параметра идентификатора окна *window_id* возвращают **функция FrontWindow()** или **функция WindowInfo()**. Для использования оператора **Set Window** по отношению к специальным окнам, таким как "Статистика", можно использовать имена окон (например, Statistics) или имена кодов (например, WIN_STATISTICS), определенных в файле стандартных определений MAPBASIC.DEF.

Следующая таблица содержит имена окон и кодов окон, которые можно использовать в качестве параметра *window_id*. Идентификаторы окон **Список таблиц** (WIN_TABLE_LIST, 2001), **Управление слоями** (WIN_LAYER_CONTROL, 2002) и **Переместить** (WIN_MOVE_MAP_TO 2003) игнорируются.

значение	Описание окна
Help	Окно Справочной системы (WIN_HELP).
Info	Окно Информация (WIN_INFO).
Legend	Окно Легенда (WIN_LEGEND).
MapBasic	Окно MapBasic (WIN_MAPBASIC).
Message	Окно Сообщение , которое использует оператор Print (WIN_MESSAGE).
Ruler	Окно инструмента Линейка (WIN_RULER).
Statistics	Окно Статистика (WIN_STATISTICS).

Дополнительное предложение **Position** задает расположение окна на экране в Рабочем наборе MapInfo Pro. Верхний левый угол рабочего набора расположен в точке с координатами 0, 0. Дополнительные, необязательные параметры ширины и высоты **Width** и **Height** позволяют управлять размером окна. В параметре "Position" используются "бумажные" единицы измерения, такие, как "in" (дюймы) или "cm" (сантиметры). MapBasic имеет по умолчанию установку в дюймах; программа MapBasic может поменять единицы, используя **оператор Set Paper Units**. Оператор **Set Window** может изменить текущие единицы измерения, для этого надо включить дополнительное подпредложение **Units** в предложения **Position**, **Width** и/или **Height**.

Если оператор включает в себя дополнительное слово **Max**, то окно будет максимизировано (это действует на весь Рабочий набор). Если оператор включает в себя дополнительное ключевое слово **Min**, то окно будет минимизировано (окно исчезнет, останется только иконка в нижней части экрана). Если окно уже минимизировано или максимизировано, и если оператор включает добавочное ключевое слово **Restore**, то окно будет восстановлено в прежнем виде. Также можно добавлять следующие дополнительные состояния:

- **Floating** — соответствует плавающему окну.
- **Docked** — соответствует закреплённому в стандартном положении окну.
- **Tabbed** — соответствует окнам, оформленным в виде вкладок, которые также, иногда, вызываться как документы.
- **AutoHidden** — соответствует автоматически спрятанному окну.

Примечание: Все четыре выше перечисленных состояния окна относятся только к 64-разрядной версии MapInfo Pro.

Если оператор включает в себя дополнительное ключевое слово **Front**, то MapBasic сделает окно активным; это также называется установкой фокуса окна. Окно станет активным сразу после щелчка мыши на заголовке окна. Если *window_id* является встроенной в окно **Отчёт** картой, то рамка будет активирована. Активной может быть единственная рамка, даже если существует другой отчёт или другое окно.

Оператор может всегда указывать предложение **Position** или предложение **Front**, независимо от типа окна. Таким образом, некоторые предложения в операторе **Set Window** применяются только к определенным типам окон. Например, окно Линейка не может изменять свой размер, максимизироваться или минимизироваться.

Для смены заголовка окна включите дополнительное предложение **Title**. Заголовок окна приложения (главный заголовок “MapInfo”) не может быть изменен иначе, чем при запуске Runtime-версии MapInfo Pro.

Предложения **Show** и **Hide** применяются к окнам документов, например, **Карта**, **Список**, **Районирование**, **Отчёт**, **3D-карта**, **Легенда**. Далее приведен пример того, как показать или спрятать окно. Параметр *id* — идентификатор окна.

```
Set Window id Show
```

```
Set Window id Hide
```

Параметры **Show** и **Hide** могут быть применены ко всем окнам инструментов таким, как окна: Список соединений, Диспетчер задач, Менеджер программ, Список окон и Проводник рабочих наборов. Далее показано как показать или спрятать окно списка соединений:

```
Set Window connectionlist Show
```

```
Set Window connectionlist Hide
```

Используйте параметры *connectionlist*, *taskmanager*, *toolmanager*, *windowlist* или *workspaceexplorer*, чтобы показать или спрятать окно инструмента.

Для окон документов, когда известен идентификатор окна, состояние параметра *show/hide* может быть определён при помощи следующей команды, где *id* – идентификатор окна:

```
print WindowInfo(id, WIN_INFO_OPEN)
```

Для того чтобы определять идентификатор окна инструментов, вместо *WIN_INFO_OPEN* используйте один из перечисленных параметров.

Тип окна	ID	Описание окна
WIN_WORKSPACE_EXPLORER	2004	Окно Проводник рабочих наборов
WIN_WINDOW_LIST	2005	Окно Список окон
WIN_TOOL_MANAGER	2006	Окно Менеджер программ
WIN_TASK_MANAGER	2007	Окно Диспетчер задач
WIN_CONNECTION_LIST	2008	Окно Список соединений
WIN_NOTIFICATIONS	2009	Предупреждения используются MapInfo Pro или дополнительными программами, для того чтобы сообщить о необходимости обратить внимание на что-то. Предупреждения появляются в окне сообщений.
WIN_QUERY_BUILDER	2010	Улучшенный диалог SQL-запрос

Method задает, будет ли печать направляться прямо на принтер, или MapInfo Pro сначала сгенерирует Windows Enhanced Metafile и пошлет его на принтер, или же MapInfo самостоятельно проведет растеризацию. Ранние версии MapInfo Pro всегда посылали файл прямо на устройство. Новый метод позволяет печатать карты с растрами, чего не было раньше. Собственная растеризация в MapInfo Pro включается в зависимости от режимов прозрачности слоев карты и перерисовки экрана. Задав режим OSBM явно, мы форсируем собственную растеризацию, вне зависимости от режимов прозрачности и/или сглаживания.

Предложение **SysMenuClose** позволяет отменить команду Заккрыть (Close) в системном меню (меню, появляющемся при щелчке в верхнем левом углу окна). Отключение команды Заккрыть действует только на пользовательский интерфейс; программы MapBasic могут закрыть окно, применив **оператор Close Window**. Следующий пример отключения команды Заккрыть для активного окна:

```
Set Window FrontWindow() SysMenuClose Off
```

Примечание: В версиях до 10.5 можно было сделать кнопку закрытия Инструментальной панели активной или неактивной вне зависимости от того, закреплена она или нет. Начиная с версии 10.5, нельзя сделать кнопку закрытия активной или неактивной, если Инструментальная панель закреплена. Можно изменить ее состояние, только если панель плавающая или плавающая и скрытая.

Улучшенные параметры рендеринга в окнах конструктора отчётов

Расширенные параметры рендеринга с визуальными эффектами, с помощью которых можно управлять сглаживанием стилей оформления, могут использоваться для окон конструктора отчёта. Это означает, что оператор MapBasic, который управляет настройками сглаживания в окне карты, теперь применяется также и в окнах конструктора отчёта.

```
Set Window layout_window_id [ Enhanced { On | Off } [ Smooth [ Vector {
None | Antialias } ] [ Image { None | Low | High } ] ]
```

Для того чтобы применить параметры сглаживания в окне конструктора отчёта, убедитесь что включены визуальные эффекты, например:

```
Set Window layout_window_id Enhanced On
```

Когда режим Enhanced выключен, сглаживание не применяется. Можно пользоваться придаточным предложением Vector, для того чтобы переключать режим сглаживания при рендеринге стилей оформления линий, а придаточным предложением Image можно пользоваться, для того чтобы переключать в отчёте сглаживание растровых символов и рамок с изображениями. В одном операторе можно комбинировать разные параметры сглаживания:

```
Set Window layout_window_id Enhanced On Smooth Vector Antialias Image
High
```

Подчинённое предложение Vector

Используйте подчинённое предложение Vector, для того чтобы переключать режим сглаживания линий контуров рамок, фигур, линий или полилиний. Например, когда включён режим визуальных эффектов — Enhanced, с помощью оператора `Set Window layout_window_id Smooth Vector Antialias` сглаживаются линии контуров всех рамок и фигур на каждой странице окна отчёта, которое имеет идентификатор `layout_window_id`, а с помощью оператора `Set Window layout_window_id Smooth Vector None` линии всех стилей отображаются без сглаживания.

Подчинённое предложение Image

Используйте подчинённое предложение Image, для того чтобы переключать режим сглаживания растровых символов и рамок с изображениями. Например, когда включён режим визуальных эффектов — оператор

`Set Window layout_window_id Smooth Image High` позволяет сильно сглаживать все рамки с изображениями и растровые символы, а оператор `Set Window layout_window_id Smooth Image None` отключает сглаживание всех изображений и растровых символов в окне отчёта.

Подчинённое предложение Text

Существует специальное предложение, которое позволяет управлять сглаживанием текста в окне карты: `[ Text { None | Antialias } ]`, однако, оно не применяется в конструкторах отчётов. Весь текст в окнах конструкторов отчётов всегда отображается со сглаживанием.

Синтаксис предложения Help

Для управления окном **Справки** задайте ключевое слово **Help** вместо целочисленного аргумента *window_id*. Например, следующий оператор покажет раздел 23 из пользовательского файла Справки:

```
Set Window Help File "custom.chm" ID 23
```

Предложение **File help_file** показывает, какой файл справки активен. В Windows, это действие автоматически показывает окно **справки** (если Вы не включили слово **Hide**). Если задать **File Default**, то MapInfo Pro будет использовать стандартную *справку MapInfo Pro*, но не прорисует на экране файл справки. MapInfo Pro имеет только один установленный файл справки, который применяется ко всем запущенным приложениям MapBasic. Если одно предложение устанавливает текущий файл справки, то и другие приложения могут им пользоваться.

Предложение **Off** отключает справку MapInfo Pro, так что при нажатии **F1** в MapInfo Pro эффекта не будет. Используйте предложение **Off** если вы интегрируете функциональность MapInfo Pro в другое приложение (например, в программу Visual Basic), если требуется, чтобы пользователь не видел справки MapInfo Pro. (Справка MapInfo Pro содержит ссылки к командам меню MapInfo Pro, которые могут не иметь действия в программе Visual Basic.)

Предложение **Permanent** заставляет MapInfo Pro всегда пользоваться Справочником *help_file*, даже если пользователь нажал **F1** в диалоге MapInfo Pro. Режим **Permanent** не меняется до конца работы с MapInfo Pro, если только вы не измените его оператором **Set Window Help File....**

Чтобы сразу открыть **Справочник** на нужном месте, задавайте предложение **ID** (для показа нужной информации).

В комплект поставки языка MapBasic не включен компилятор справочных файлов. Более подробно Справочная система описана в *Руководстве пользователя MapBasic*.

Синтаксис для окон карт и отчётов

Предложение **ScrollBars** применимо только к окнам карт. Предложение **ScrollBars** управляет показом строки (полосы) прокрутки в окне карты.

Предложение **Autoscroll** применимо окнам карт и отчётов. По умолчанию, режим автоматической прокрутки действует в картах и отчётах, т.е. при выполнении операции с нажатой кнопкой мыши в окне карты и отчёта, содержимое окна автоматически сдвигается вслед за мышью при приближении ее указателя к краю окна. Чтобы отключить автоматическую прокрутку, задайте **Autoscroll Off**. **Функция WindowInfo()** поможет определить, в каких окнах действует режим автоматической прокрутки. .

Предложения **ScrollBars**, **Autoscroll** и **Smart Pan** в окне **Отчёт** не выполняются (эти предложения игнорируются).

Smart Pan изменяет статус операции сдвига в окне. Когда **Smart Pan** включен для окна Карты или Отчета, то при сдвиге и прокрутке используется отключение растровой прорисовки для уменьшения бликов. Стандартное состояние для **Smart Pan** – отключенное.

Когда **Smart Pan** активизирован для окна Отчета, то перерисовка действует только при использовании инструмента Сдвиг.

Когда **Smart Pan** активизирован для окна Карты, то эффект будет зависеть от метода перемещения Карты. Инструмент Сдвиг автоматически прорисовывает область окна при использовании самого инструмента. Карта будет перемещаться медленнее, чем при отключенном режиме **Smart Pan**. Чем сложнее карта, тем медленнее она будет перемещаться. Прокрутка и автопрокрутка действует похоже на инструмент Сдвиг, но скорость прокрутки не влияет на точность сдвига. Когда команда **Set Map** используется для центрирования или сдвига изображения со включенной командой **Smart Redraw**, изменения окна Карты при прорисовке произойдут без бликования.

Примечание: Если растровый режим прорисовки изображения на экране отключен, то предложение **Smart Pan** в окне Карты ведет себя также как и в окне Отчета.

Синтаксис окна отчёта

Не все предложения-конструкции **Set Window** работают с окном **Отчёт**.

Предложения **ScrollBars**, **Autoscroll** и **Smart Pan** в окне **Отчёт** не выполняются (эти предложения игнорируются).

Применение следующих предложений-конструкций к окну конструктора отчёта вызывает сообщение об ошибке о том, что эти предложения не могут быть выполнены по отношению к окну.

- Предложение **Enhanced** (для применения эффектов оформления)
- Предложение **Smooth**
- Предложение **Export**
- Предложение **Parent**
- Предложения **ReadOnly**, **Default Access**
- Предложение **Table**
- Предложение **Show/Hide**
- Конструкции из предложений **Snap Mode/Threshold**

Синтаксис для вспомогательных окон (Легенда, Линейка и другие)

Предложение **Parent**, позволяющее задать новое порождающее окно для окон Легенды, Статистики, Информации, Линейки или Сообщений, действует только в Windows. Окно с номером *window_id* становится родителем, подчиненным окну с номером-указателем *HWND*.

Примечание: Переподчинение окна таким способом изменяет значение ID для этого окна. Чтобы снова подчинить окно первоначальному “родителю”, MapInfo Pro, задайте ноль в качестве *HWND*.

Предложения **ReadOnly** / **Default Access** применяются только к окну Информации и управляют возможностью изменения данных в нем. и управляют возможностью изменения данных в нем. **ReadOnly** запрещает редактирование данных. Предложение **Default Access** снимает контроль со стороны MapBasic, и тогда уже действуют запреты или разрешения для самой таблицы. Это работает для главной легенды и картографических легенд, которые создает **оператор Create Legend** или **оператор Create Cartographic Legend**.

Предложение **Table** позволяет выбрать данные для показа в окне Информации (и только для него). Это предложение форсирует показ окна Информации.

Предложения **Show** и **Hide** управляют показом или скрыванием окон, для которых эта операция существенна (например, для Линейки), но может применяться и для окна MapInfo Pro.

Управление принтером

По умолчанию, окна распечатываются используя настройки системного принтера. Это может быть принтер Windows или настроенный в MapInfo Pro принтер, в зависимости от установок пользователя. Использование предложения **Name** в приложении, рабочем наборе или окне **MapBasic** может изменить принтер для отдельного документа. Некоторые настройки для принтера также могут задаваться некоторыми другими командными предложениями. Таким образом, когда настройки принтера изменяются через пользовательский интерфейс, соответственно генерируются команды MapBasic. Подобные изменения настроек хранятся в рабочих наборах, а отменить их можно командой **Set Window Printer Default**.

Если **Scale Patterns** установлено **On**, штриховки масштабируются на основе соотношений разрешения устройства вывода и разрешения экрана.

Коды WIN_INFO_PRINTER_NAME (21), WIN_INFO_PRINTER_ORIENT (22) или WIN_INFO_PRINTER_COPIES (23), также возвращает **функция WindowInfo()**.

Пример

```
Set Window frontwindow( )
Printer Name "\\Discovery\HP 2500CP"
Orientation Portrait
Copies 10
```

Примечание: Для того чтобы определить принтер, связанный с окном, запустите MapInfo Pro и выполните команду **Настройка печати** на вкладке **ГЛАВНАЯ**. Нажмите кнопку **Принтер**. Используйте имя принтера из этого диалога.

Управление радиусом совмещения

Можно настроить радиус совмещения, измеряемый в пикселах для данного окна, вернуть стандартный размер радиуса совмещения для этого окна или получить информацию о текущем радиусе совмещения. Можно также включать/отключать режим совмещения для данного окна или запрашивать включен или нет режим совмещения для данного окна.

Настройки радиуса совмещения для обычного окна может запрашивать с использованием новых атрибутов параметров [функция WindowInfo\(\)](#). Теперь режим совмещения может настраиваться для каждого окна карты или отчёта. Эти настройки сохраняются в рабочем наборе для каждого окна.

Пример

```
Dim win_id As Integer
Open Table "world"
Map From world
win_id = FrontWindow( )
Set Window win_id Width 5 Height 3
```

Сохранение WOR-файла так, чтобы его можно было открыть в локализованных и нелокализованных версиях

До версии MapInfo Pro/MapBasic 9.0.2, если был создан рабочий набор с отчётом в одной языковой среде, а затем передавали его пользователю, работающему в другой языковой среде, то при открытии этого рабочего набора происходила ошибка. Это происходило из-за конфликтов имен карт в разных языках.

Для обхода этой проблемы, мы создали параметр в реестре, который позволяет пользователям в странах с разными локалями обмениваться рабочими наборами. Этот параметр необходимо ввести вручную.

Чтобы избежать конфликта имен карт при переходе в другую локаль:

1. Введите `regedit` в командной строке.
2. В окне программы **Редактор реестр** перейдите в раздел `My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Mapinfo\Mapinfo\Common`.
3. Щелкните правой кнопкой мыши и выберите **Создать > Параметр DWORD** для создания нового параметра реестра типа DWORD.
4. Введите **WriteWindowTitle** и нажмите **Enter**.
Откроется окошко **правки параметра DWORD**.
5. Введите **1** в поле **значения параметра** и нажмите **ОК**, чтобы сохранить параметр.
6. Закройте MapInfo Pro и запустите снова.

Теперь имя таблицы будет явно записываться в отчёт. Это позволяет избежать переименования, если изменяются языковые параметры файла.

Ссылки, связанные с данной

[Оператор Browse](#) на стр. 133

[Оператор Layout](#) на стр. 575

[Оператор Map](#) на стр. 609

[Оператор Set Paper Units](#) на стр. 992

Оператор Set Workspace Path

Назначение

Позволяет настраивать маршрут к заданному файлу, не изменяя его. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Workspace Path filespec
```

filespec – строка, представляющая имя файла и, если необходимо, маршрут в дисковом пространстве.

Описание

Этот оператор указывает MapInfo Pro на заданный файл, не изменяя его.

Пример

```
Set Workspace Path "C:\market.wor"
```

Оператор Set Workspace Warning

Назначение

Позволяет включать или выключать сообщение с предупреждением о сохранении рабочего набора, которое появляется при попытке закрыть MapInfo Pro. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Set Workspace Warning { On [ Auto | Named ] | Off }
```

Описание

Если **Auto** добавлено к ключевому слову **On**, MapInfo Pro предложит сохранить работу, когда закрываются все таблицы или выполняется выход из программы, по следующим сценариям:

- Открыт рабочий набор, в котором имеются несохранённые изменения.
- Открыт не рабочий набор, а только таблицы, в которых имеются несохранённые изменения. В этом случае будет предложено сохранить рабочий набор с именем "untitled.wor".
- Если открыто несколько рабочих наборов, в которых имеются несохранённые изменения. В этом случае будет предложено сохранить рабочий набор с именем "untitled.wor".

Если добавлено слово **Named** к ключевому слову **On**, MapInfo Pro предлагает сохранить работу, когда закрываются все таблицы или при выходе из приложения, только, если в открытом рабочем наборе имеются несохранённые изменения.

Стандартным, используемым по умолчанию, является вариант, который выполняет MapInfo, когда слово **Auto** добавлено к ключевому слову **On**. Оператор **Set Workspace Warning** можно использовать, чтобы изменить стандартное поведение.

Если этот оператор выполняется с ключевым словом **Off**, MapInfo Pro не будет предлагать сохранить рабочий набор при закрытии сессии.

Изменения, внесённые выполнением оператора **Set Workspace Warning**, сохраняются только пока продолжается сеанс работы MapInfo Pro.

Функция Sgn()

Назначение

Распознает знак числа. Возвращает -1, 0, или 1, указывая является ли число отрицательным, положительным или равным нулю. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Sgn ( num_expr )
```

num_expr – числовое выражение

Возвращаемая величина

Вещественное (-1, 0 или 1)

Описание

Функция **Sgn()** возвращает -1 (минус единица), если число, заданное выражением *num_expr*, отрицательно, 0, если число равно нулю, и 1 (единица), если число больше нуля.

Пример

```
Dim x As Integer
x = Sgn(-0.5)

' x now has a value of -1
```

См. также:

[функция Abs](#)

Оператор Shade

Назначение

Создает тематический слой и добавляет его к существующей карте. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

См. следующие разделы:

- [Диапазоны значений](#)
- [Индивидуальные значения](#)
- [Плотность точек](#)
- [Размерные символы](#)
- [Круговые диаграммы](#)
- [Столбчатые диаграммы](#)

Описание

Оператор **Shade** используется для создания тематического слоя в окне карты. Оператор **Shade** соответствует команде MapInfo Pro **Добавить тематику**. Введение в тематическое картографирование и описание команды **Добавить тематику** содержится в *Справочной системе* MapInfo Pro.

MapInfo Pro может запоминать тип тематической Карты, помещая оператор **Shade** в файл рабочего набора, если в нем есть хотя бы одно окно карты с тематическим слоем. Для примера

работы оператора **Shade** можно открыть окно карты, выполнить команду **Добавить тематику** и сохранить рабочий набор (например, под именем `THEME.WOR`) и проверить его в текстовом окне MapBasic. Теперь откройте файл рабочего набора в любом текстовом редакторе и вы увидите оператор **Shade**, задающий те настройки слоя, которые были заданы в открытом ранее окне карты. Можно копировать этот оператор из рабочего набора в свою программу. Оператор **Shade** будет запротоколирован, если при работе MapInfo Pro было открыто окно **MapBasic** перед выбором команд **Добавить тематику**).

Диапазоны значений

Синтаксис

```
Shade [ Window window_id ]
{ layer_id | layer_name }
With Metadata
With expr
[ Ignore value_to_ignore ]
Ranges
[ Apply { Color | Size | All } ]
[ Use { Color | Size | All } [ Line... ] [ Brush... ]
[ Symbol... ]
]
{ [ From Variable float_array Style Variable style_array ] |
  minimum : maximum [ Pen... ] [ Line... ] [ Brush... ]
[ Symbol... ] [ Hide | Show ]
[ , minimum : maximum [ Pen... ] [ Line... ] [ Brush... ]
[ Symbol... ] [ Hide | Show ] ... ]
}
[ Style Replace { On | Off } ]
[ Default [ Pen... ] [ Line... ] [ Brush... ] [ Symbol... ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

value_to_ignore – величина, которая должна быть проигнорирована, т.е. не будет создано тематического объекта для строки, значение которой равно параметру *value_to_ignore*; обычно это 0, если выражение численное, или пустая строка, если выражение строковое.

float_array – численный массив типа Float, который возвращает **оператор Create Ranges**.

style_array – массив величин типа **Pen**, **Brush** или **Symbol**, который возвращает **оператор Create Styles**.

minimum – минимальная величина для диапазона.

maximum – максимальная величина для диапазона/

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т. д. Кроме того, слой может задаваться именем (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Если указать предложение **With Metadata**, будет отображена тематическая Карта, построенная на основе тем, сохранённых в метаданных.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Ranges** задаёт тематическую раскраску объектов карты по диапазонам значений.

Предложение Pen начинает стандартное предложение контура для замкнутых объектов. Например, **MakePen**(*width, pattern, color*).

Line начинает стандартное предложение оператора для определения линии для объектов типа "Линия", "Полилиния" и "Дуга". Предложение **Line** и Предложение **Pen** имеют одинаковый синтаксис, за исключением начального ключевого слова **Line** вместо **Pen**.

Предложение Brush оператора для определения стиля штриха. Например, **MakeBrush**(*pattern, forecolor, backcolor*).

Предложение Symbol начинает стандартное предложение оператора для определения стиля символа точечного объекта. Например, **MakeSymbol**(*shape, color, size*).

Особенности синтаксиса карты диапазонов смотрите в разделе: **Синтаксис Диапазоны значений**.

В первом варианте синтаксиса предложения **From Variable** и **Style Variable** используются для чтения заранее вычисленных значений, определяющих границы диапазонов и стили для их оформления. Значения для массивов переменных, имена которых задаются в этих предложениях, должны быть вычислены до выполнения оператора **Shade**, для чего применяются **оператор Create Ranges** и **оператор Create Styles**. Пример использования массивов в операторе **Shade** приводится в разделе: **Оператор Create Ranges**.

Если оператор **Shade** строит тематический слой методом диапазонов (предложение **Ranges**) или отдельных значений (ключевое слово **Values**; смотрите описание ниже), то он может использовать предложение **Default**, задающее единый стиль оформления объектов, которые попадают в группу "остальные", т. е. для них не выполняется условие тематического выделения. Таким образом можно задать единый стиль оформления для оставшихся объектов. Всего

может быть задано от одного до шестнадцати диапазонов. Каждое описание диапазона состоит из пары чисел (`minimum` и `maximum`), разделенных двоеточием и определяющих верхний и нижний предел диапазона, и следующими за ними описаниями стилей оформления объектов, принадлежащих этому диапазону. Если выражение `expr` для записи больше или равно `minimum` и меньше `maximum`, то эта запись принадлежит этому диапазону. Описания диапазонов в предложении `Ranges` разделяются друг от друга запятыми.

```
Open Table "states"
Map From states
Shade states With Pop_1990 Ranges
  4827000:29280000 Brush (2,0,201326591) ,
  1783000: 4827000 Brush (8,0,16777215) ,
  449000: 1783000 Brush (5,0,16777215)
```

Стиль выделения областей при создании карты диапазонов задает **предложение Brush**. Стиль раскраски и выделения точек определяет **предложение Symbol**. Линейные объекты раскрашиваются, используя предложение **Line**, а не **предложение Pen**, синтаксис которых одинаков, кроме замены ключевого слова **Pen** ключевым словом **Line**. (В операторе **Shade** предложение `Pen` управляет стилем контура замкнутых объектов.)

Задавая ключевое слово **Hide**, можно прятать тематический слой диапазонов в окне карты. Альтернативно, если задается ключевое слово **Show**, то тематический слой диапазонов будет видим в окне карты. Использование этих ключевых слов является необязательным и MapInfo Pro стандартно отображает тематические слои диапазонов, если ключевое слово **Hide** не задано.

Style Replace On запрещает отображение слоёв под тематическим слоем.

Style Replace Off предписывает отображать слои под тематическим слоем, что позволяет создавать многовариантные прозрачные тематические слои.

Style Replace On – это режим по умолчанию, обеспечивающий обратную совместимость с существующими настройками, так что по умолчанию нижележащие слои не отображаются.

С помощью предложения **Apply** можно выбрать способ, которым будут выделены графические объекты на тематическом слое.

Предложение Apply Эффект

Apply Color	Выделение производится только цветом, а размеры и другие атрибуты, такие как размер символа и тип линии, будут такими же, как на карте. Исходные форма и размер точечных объектов не меняются. Исходные тип и толщина линейных объектов не меняются. Раскрашенные объекты визуализируются в своем исходном цвете, но тематическое выделение управляет цветом тона.
Apply Size	Выделение производится только изменением размеров символов и толщиной линии. Точечные объекты при этом не меняют свой рисунок и раскраску, а линейные объекты – тип линий и цвет. Исходные тип и толщина линейных объектов не меняются.

Предложение Apply Эффект

Apply All	В тематическом выделении участвуют все атрибуты объекта: размер и форма символа, тип линий, толщина и цвет.

По умолчанию используется режим, который устанавливается предложением **Apply All**.

Предложение **Use** позволяет управлять тем, в каком объеме MapInfo Pro применяет стилизацию объектов при раскраске диапазонов. В следующем примере показан смысл этого предложения. Пусть таблица WorldCap, содержащая точки, раскрашена без участия предложения **Use**.

```
Shade WorldCap With Cap_Pop Ranges
Apply All
0 : 300000 Symbol(35,YELLOW,9) ,
300000 : 900000 Symbol(35,GREEN,18) ,
900000 : 20000000 Symbol(35,BLUE,27)
```

В результате тематическая карта будет раскрашена как указано в **предложении Symbol** звездочками (код 35) так, как определяет : наименьшие объекты будут показаны желтыми звездочками в 9 пунктов, объекты среднего объема будут показаны зелеными звездочками в 18 пунктов и большие объекты будут показаны голубыми звездочками в 27 пунктов.

В следующем примере добавлено предложение **Use Size**.

```
Shade WorldCap With Cap_Pop Ranges
Apply All

Use Size Symbol(34, RED, 24) ' <<<<< Note!

0 : 300000 Symbol(35,YELLOW,9) ,
300000 : 900000 Symbol(35,GREEN,18) ,
900000 : 20000000 Symbol(35,BLUE,27)
```

Примечание: Обратите внимание на то, что **Use Size** вносит свой стиль символа (код 34 представляет кружок).

В связи с предложением **Use Size**, из всех режимов, которые задает **предложение Symbol**, применяются только размеры (9, 18, 27 точки). MapInfo Pro игнорирует остальные атрибуты (например, YELLOW, GREEN, BLUE). Тематическая карта отобразит красные кружки, поскольку **предложение Symbol** задает в **Use Size** красные кружки. Конечный результат: наименьшие объекты показываются красными кружками в 9 пунктов, средние объекты показываются красными кружками в 18 пунктов, крупные объекты показываются красными кружками в 27 пунктов.

Если задать **Use Color** вместо **Use Size**, MapInfo Pro будет использовать только цвета, которые зададо последнее **предложение Symbol**. На карте будут показаны желтые, зеленые и голубые кружочки размером в 24 точки.

Предложение **Use All** равносильно отсутствию предложения **Use** вообще.

Предложение **Use** действует только в паре с **Apply All** (либо **Apply** отсутствует вообще).

Индивидуальные значения

Синтаксис

```
Shade [ Window window_id ]
{ layer_id | layer_name }
With Metadata
With expr
[ Ignore value_to_ignore ]
Values const [ Pen... ] [ Line... ] [ Brush... ] [ Symbol... ]
[ , const [ Pen... ] [ Line... ] [ Brush... ] [ Symbol... ] ... ]
[ Vary { Color | All } ]
[ Style Replace { On | Off } ]
[ Default [ Pen... ] [ Brush... ] [ Symbol... ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

value_to_ignore – величина, которая должна быть проигнорирована, т.е. не будет создано тематического объекта для строки, значение которой равно параметру *value_to_ignore*; обычно это 0, если выражение численное, или пустая строка, если выражение строковое.

const – численная или строковая константа или выражение, в котором не используется переменных.

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т.д. Кроме того оператор **Shade** может определять слой карты по имени (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Если указать предложение **With Metadata**, будет отображена тематическая Карта, построенная на основе тем, сохранённых в метаданных.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Values** порождает карту, на которой каждый объект выделяется уникальным образом.

Предложение Pen начинает стандартное предложение контура для замкнутых объектов. Например, **MakePen**(*width, pattern, color*).

Line начинает стандартное предложение оператора для определения линии для объектов типа "Линия", "Полилиния" и "Дуга". Предложение **Line** и Предложение **Pen** имеют одинаковый синтаксис, за исключением начального ключевого слова **Line** вместо **Pen**.

Предложение Brush оператора для определения стиля штриха. Например, **MakeBrush**(*pattern, forecolor, backcolor*).

Предложение Symbol начинает стандартное предложение оператора для определения стиля символа точечного объекта. Например, **MakeSymbol**(*shape, color, size*).

Особенности синтаксиса карты диапазонов смотрите в разделе **Диапазоны значений**.

За ключевым словом **Values** может следовать от одного до 255 описаний. Каждое состоит из уникального значения (строка или число) и описаний стилей оформления объектов, принадлежащих значению. Если выражение *expr* для записи равно одному из уникальных значений, определенных в операторе **Shade**, то объект, который присоединен к этой записи, будет оформлен соответственно описанию для этого значения. Список описаний разделяется запятыми.

Если оператор **Shade** имеет предложение **Ranges** или **Values**, то он может иметь и предложение **Default**, задающее единый стиль оформления объектов, которые попадают в группу "остальные", т. е. для них не выполняется условие тематического выделения. Таким образом можно задать единый стиль оформления для оставшихся объектов. Предложение **Vary** позволяет указать, как будет меняться вид объектов. По умолчанию применяется стиль **Vary All**. Если применяется стиль **Vary All**, к тематическому слою применяются все инструменты отображения объектов в каждом диапазоне размеров объектов. Если задать **Vary Color**, то для каждого диапазона будет использован заданный цвет.

Style Replace On запрещает отображение слоёв под тематическим слоем.

Style Replace Off предписывает отображать слои под тематическим слоем, что позволяет создавать многовариантные прозрачные тематические слои. Это позволяет отображать прозрачные штриховки в том же слое.

Style Replace On – это режим по умолчанию, обеспечивающий обратную совместимость с существующими настройками, так что по умолчанию нижележащие слои не отображаются.

В следующем примере таблица территорий Великобритании UK_SALES содержит колонку "Sales_Rep", заполненную фамилиями торговых представителей, действующих на территории Великобритании. Оператор **Shade** закрашивает каждый округ в зависимости от того, какой

торговый представитель его курирует. Так, территории, за которые отвечает Боб, будут закрашены одним цветом, а округа, в которых торгует Джон, другим и т.д.

```
Open Table "uk_sales"
Map From uk_sales

Shade 1 With Proper$(Sales_Rep)
Ignore ""
Values
    "Alan" ,
    "Amanda" ,
    "Bob" ,
    "Jan"
```

Плотность точек

Синтаксис

```
Shade [ Window window_id ]
{ layer_id | layer_name }
With expr
Density dot_value { Circle | Square }
Width dot_size
[ Color color ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

dot_value – число, соответствующее одной точке при использовании метода плотности точек.

dot_size – размер одной точки в пикселах при использовании метода плотности точек.

color – значение RGB для цвета точек при использовании метода плотности точек.

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т. д. Кроме того, слой может задаваться именем (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Density** порождает Карту плотности точек.

Синтаксис для карты плотности точек описан в разделе **Синтаксис Плотность точек**.

За ключевым словом **Density** должно следовать только одно описание *dot_value:dot_size*. Можно выбирать между стилями Circle (кружок) и Square (квадратик). Раскрашиваемая карта должна содержать области, которые будут заполняться точками. Например, каждая точка соответствует одной тысяче домов.

Для такой карты численное выражение *expr* вычисляет значение для каждой области слоя карты и делит его на *dot_value*. MapInfo Pro определяет, сколько точек отображать для полигона, путём деления значения *expr* этого полигона на значение *dot_value*. Например, если для области *значение* выражения равно 100, а в операторе **Shade** *dot_value* было задано равным 5, то MapBasic нарисует 20 точек в этой области.

За ключевым словом **Width** следует параметр *dot_size*. Параметр *dot_size* определяет размер иллюстративной точки в пикселах. Для круглых точек параметр *dot_size* может изменяться от 2 до 25 пикселей. Для квадратных точек параметр *dot_size* может изменяться от 1 до 25 пикселей. Предложение **Color** можно использовать для указания цвета точек.

В следующем примере в таблице штатов STATES строится тематическая карта плотности точек по количеству домовладений в каждом штате в 1990 году по колонке "Pop_1990". Каждая точка имеет размер в 4 пиксела и представляет 60 000 домовладений.

```
Open Table "states"
Map From states
shade window 176942288 7
with Pop_1990
density 600000 circle width 4
color 255
```

Примечание: В целях обратной совместимости, поддерживается синтаксис MapBasic версий 7.5 и более ранних.

Размерные символы

Синтаксис

```
Shade [ Window window_id ]
{ layer_id | layer_name }
With expr
```

```

Graduated min_value : symbol_size max_value : symbol_size
Symbol...
[ Inflect Symbol... ]
[ Vary Size By { "LOG" | "SQRT" | "CONST" } ]

```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

max_value – максимальный размер символа.

min_value – минимальный размер символа.

symbol_size – размер символа в пунктах, используемого в методе размерных символов.

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т. д. Кроме того, слой может задаваться именем (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Graduated** порождает карту размерных символов.

Синтаксис для карт размерных символов описан в разделе: **Синтаксис Размерные символы**.

За ключевым словом **Graduated** следует пара операторов *value* : *symbol_size*. Параметр *value* из первого описания соответствует минимальному значению, представляемому минимальным размером символа *symbol_size*. Параметр *value* из второго описания соответствует максимальному значению, представляемому максимальным размером символа *symbol_size*. Размер символов промежуточных значений MapInfo Pro подбирает автоматически.

Предложение **Symbol** диктует, какой тип символа выбирается для выделения (кружок, звездочка и т.п.). Если оператор использует предложение **Inflect**, задающее второй вариант типа символа, то MapInfo Pro обозначает этими символами объекты, имеющие отрицательное значение.

Следующий оператор создает тематический слой, отображающий доходы и убытки. Пункты, приносящие доход, обозначаются зелеными треугольниками, направленными вверх. Пункты, терпящие убытки, обозначаются красными треугольниками, направленными вниз.

```
Shade stores With Net_Profit
  Graduated
  0.0:0 15000:24
  Symbol(36, GREEN, 24)
  Inflect Symbol(37, RED, 24)
  Vary Size By "SQRT"
```

Предложение **Vary Size By** управляет методом, которым размерные символы отражают разницу между значениями, которые они представляют. Если предложение **Vary Size By** не задано, то MapInfo Pro использует метод "SQRT" (квадратный корень), который задает размер символа пропорциональным квадратному корню из выделяемого значения. При действии этого метода, если в одной записи значение в два раза больше соответствующего значения в другой записи, то и площадь, занятая на экране символом для первого значения будет в два раза больше площади символа, представляющего второе значение.

Примечание: Увеличение площади в два раза не идентично увеличению размера символа в точках. Увеличение размера символа в точках увеличивает и длину, и ширину; следовательно, площадь увеличивается в квадрате.

Круговые диаграммы

Синтаксис

```
Shade [ Window window_id ]
  { layer_id | layer_name | Selection }
  With expr [ , expr... ]
  [ Half ] Pie [ Angle angle ] [ Counter ]
  [ Fixed ] [ Max Size chart_size [ Units unitname ]
  [ At Value max_value [ Vary Size By {"LOG" | "SQRT" | "CONST" } ] ] ]

  [ Border Pen... ]
  [ Position [ { Left | Right | Center } ] [ { Above | Below | Center } ] ]

  [ Style Brush... [ , Brush... ] ]
```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

angle – начальный угол для круговой диаграммы.

chart_size – число типа Float, представляющее максимальную высоту для круговых диаграмм или столбчатых диаграмм.

unitname – "бумажная" единица измерения (например, "in" – дюйм, "cm" – сантиметр).

max_value – число, соответствующее точке в методе размерных символов. Для каждой записи, если сумма выражений в колонке равна *max_value*, значение высоты соответствующей круговой или столбчатой диаграммы будет равно *chart_size*; меньшие диаграммы – для последовательностей с меньшими суммами.

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т. д. Кроме того, слой может задаваться именем (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Pie** порождает Карту с круговыми диаграммами.

Предложение Pen начинает стандартное предложение контура для замкнутых объектов. Например, **MakePen**(*width*, *pattern*, *color*).

Предложение Brush оператора для определения стиля штриха. Например, **MakeBrush**(*pattern*, *forecolor*, *backcolor*).

Синтаксис карт с гистограммами приведен в разделе **Синтаксис Столбчатые гистограммы**.

Ключевое слово **Pie** задает метод построения тематического выделения объектов путем создания для каждого объекта маленького графика типа круговая диаграмма. При этом в предложении **With** через запятую должны быть заданы два или более выражений, значения которых для каждого объекта будут показаны круговой диаграммой.

Диаграммы могут иметь форму полного круга, и могут иметь форму половины окружности, если перед словом **Pie** поставить слово **Half**.

Стартовый угол для диаграммы Вы можете задать в предложении **Angle**. По умолчанию он равен 180 градусам.

Если используется ключевое слово **Counter**, то сектора в диаграмме располагаются от стартового угла против часовой стрелки.

Предложение **Max Size** определяет максимальный размер круговой диаграммы в "бумажных" единицах. "Бумажные" единицы измерения описаны в разделе: [Оператор Set Paper Units](#).
Предложение **Fixed** задает рисование круговых диаграмм в кругах одинакового размера.

Следующий оператор задает выделение диаграммами одинакового размера, в четверть дюйма:

```
Shade sales_95 With phone_sales, retail_sales
  Pie Fixed
  Max Size 0.25 Units "in"
```

Чтобы круговая диаграмма могла устанавливать размер самостоятельно, применяйте вместо слова **Fixed** предложение **At Value**. Например, следующий оператор создает тематическую карту с изменяющимся размером круговой диаграммы. Если сумма значений в записи равна 85 000, то круговая диаграмма будет иметь радиус 2 см; записи с меньшими суммами породят меньшие диаграммы.

```
Shade sales_95 With phone_sales, retail_sales
  Pie
  Max Size 0.25 Units "in" At Value 85000
```

Предложение **Vary Size By** можно включить для выбора метода градуировки размера круговых диаграмм. Это предложение описано выше в разделе [Размерные символы](#).

Предложение **Position** задает место относительно объекта, где помещается диаграмма. По умолчанию диаграммы помещаются в центр объекта.

В предложении **Style** можно задать раскраску для секторов диаграммы. Каждый стиль Brush должен соответствовать выражению из списка в предложении **With**. Если стили не заданы, то MapInfo Pro использует значение типа Brush, сохраненное пользователем.

Следующий оператор создает тематический слой, в котором круговые диаграммы для двух значений помещаются выше центра объекта.

```
Shade sales_95 With phone_sales, retail_sales
  Pie Angle 180
  Max Size 0.5 Units "in" At Value 85000
  Vary Size By "SQRT"
  Border Pen (1, 2, 0)
  Position Center Above
  Style Brush(2, RED, 0), Brush(2, BLUE, 0)
```

Столбчатые диаграммы

Синтаксис

```
Shade [ Window window_id ]
  { layer_id | layer_name | Selection }
```



```

With expr [ , expr... ]
{ Bar [ Normalized ] | Stacked Bar [ Fixed ] }
[ Max Size chart_size [ Units unitname ]
  [ At Value max_value [ Vary Size By { "LOG" | "SQRT" | "CONST" } ] ]
[ Border Pen... ]
[ Frame Brush... ]
[ Width value [ Units unitname ] ]
[ Position [ { Left | Right | Center } ] [ { Above | Below | Center } ] ]

[ Style Brush... [ , Brush... ] ]

```

window_id – целочисленный идентификатор окна Карты.

layer_id – идентификатор слоя в окне карты, от единицы и больше.

layer_name – имя слоя в окне карты.

expr выражение, задающее условие выделения и раскраски, такое как имя колонки.

chart_size – число типа Float, представляющее максимальную высоту для круговых диаграмм или столбчатых диаграмм.

max_value – число, соответствующее точке в методе размерных символов. Для каждой записи, если сумма выражений в колонке равна *max_value*, значение высоты соответствующей круговой или столбчатой диаграммы будет равно *chart_size*; меньшие диаграммы – для последовательностей с меньшими суммами.

unitname – "бумажная" единица измерения (например, "in" – дюйм, "cm" – сантиметр).

значение

Описание

Параметр *window_id*, задающий окно Карты, является необязательным. Если он опущен, то MapBasic создаст тематический слой в самом верхнем окне Карты.

Оператор **Shade** должен явно задать слой, на котором будет производиться условное выделение, даже если это единственный слой на карте. Слой может задаваться номером (*layer_id*), причем первый номер соответствует самому верхнему некосметическому слою, слой под первым будет вторым, и т. д. Кроме того, слой может задаваться именем (например, "world").

Способ тематического оформления оператором **Shade** зависит от того, какое предложение следует за выражением *expr*. MapInfo Pro определяет эти выражения для каждого выделенного объекта таблицы; в соответствии с оператором **Shade** MapInfo Pro выбирает стиль отображения каждого объекта, основанный на значении *expr*. Выражения обычно содержат имена одной или более колонок из выбранной таблицы.

Ключевые слова, следующие за предложением *expr*, указывают тип выделения. Ключевое слово **Bar** порождает карту с гистограммами.

Предложение Pen начинает стандартное предложение контура для замкнутых объектов. Например, **MakePen**(*width, pattern, color*).

Предложение Brush оператора для определения стиля штриха. Например, **MakeBrush**(*pattern, forecolor, backcolor*).

Синтаксис карт с гистограммами приведен в разделе: **Синтаксис круговых диаграмм**.

Ключевое слово **Bar** задает метод построения тематического выделения объектов путем создания для каждого объекта маленькой столбчатой диаграммы. При этом в предложении **With** через запятую должны быть заданы два или более выражений, значения которых для каждого объекта будут показаны диаграммы.

Если использовалось ключевое слово **Stacked** перед словом **Bar**, то MapInfo Pro размещает столбики стопкой, иначе столбики располагаются в ряд. Если слово **Stacked** отсутствует, то можно задать слово **Normalized** и тогда столбики могут иметь независимые шкалы.

Если задано ключевое слово **Stacked**, то его можно дополнить словом **Fixed**, определяющим, что все стопки будут иметь одинаковый размер. Если слово **Fixed** не задано, то MapInfo Pro задает размер каждой стопки пропорционально сумме отображаемых значений.

Предложение **Frame**, входящее в **предложение Brush** задает стиль раскраски фона графика.

Предложение **Position** управляет как ориентацией столбчатых диаграмм, так и ориентацией диаграммы по отношению к центру объекта. Если в предложении **Position** заданы слова **Left** или **Right**, то столбцы горизонтальны, в других случаях они вертикальны.

Предложение **Style** содержит перечисленные через запятую стили штриховки (Brush). Каждый из стилей штриховки соответствует выражению из предложения **With**.

В следующем примере создается тематический слой на карте, и выделение производится столбчатыми диаграммами, расположенными над центроидами объектов.

```
Shade sales_93
  With phone_sales, retail_sales
  Bar
  Max Size 0.4 Units "in" At Value 1245000
  Vary Size By "CONST"
  Border Pen (1, 2, 0)
  Position Center Above
  Style Brush(2, RED, 0), Brush(2, BLUE, 0)
```

См. также:

оператор Create Ranges, оператор Create Styles, оператор Map, оператор Set Legend, оператор Set Map, оператор Set Shade

Функция Sin()

Назначение

Вычисляет синус. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Sin( num_expr )
```

num_expr – численное выражение угла в радианах.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Sin()** возвращает синус числа, полученного в результате вычисления выражения *num_expr*. Угол должен задаваться в радианах. Диапазон возвращаемого **Sin()** значения находится между единицей и минус единицей включительно. Для перевода градусов в радианы число необходимо умножить на число DEG_2_RAD. Для обратного конвертирования используется коэффициент RAD_2_DEG. Для того, чтобы программа могла использовать эти коэффициенты конвертирования, она должна содержать ссылку на файл MAPBASIC.DEF.

Пример

```
Include "mapbasic.def"  
Dim x, y As Float  
x = 30 * DEG_2_RAD  
y = Sin(x)  
' y will now be equal to 0.5  
' since the sine of 30 degrees is 0.5
```

См. также:

[функция Acos\(\)](#), [функция Asin\(\)](#), [функция Atn\(\)](#), [функция Cos\(\)](#), [функция Tan\(\)](#)

Функция Space\$()

Назначение

Возвращает строку, состоящую из пробелов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Space$( num_expr )
```

num_expr – целочисленное выражение.

Возвращаемая величина

Строка

Описание

Функция **Space\$()** возвращает строку из пробелов. Количество пробелов определяется выражением *num_expr*. Если *num_expr* принимает значение меньше или равное нулю функция **Space\$()** возвратит пустую строку.

Пример

```
Dim filler As String
filler = Space$(7)
' filler is now equal to the string "      "
' (7 spaces)
Note "Hello" + filler + "world!"
'this displays the message "Hello      world!"
```

См. также:

[функция String\\$\(\)](#)

Функция SphericalArea()

Назначение

Возвращает площадь, вычисленную на сфере в координатах "Широта/Долгота" по дугам больших кругов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalArea( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – строка, определяющая единицы измерения площади (к примеру, "кв. км.").

Возвращаемая величина

Вещественное – Float

Описание

Функция **SphericalArea()** возвращает площадь географического объекта, обозначенного выражением *obj_expr*. Функция возвращает площадь области в единицах измерения, указанных в параметре *unit_name*; к примеру, чтобы получить площадь в акрах, укажите "акры" в качестве значения параметра *unit_name*. Список доступных единиц измерения смотрите в разделе: **Оператор Set Area Units**.

Функция **SphericalArea()** всегда возвращает площадь на сфере, вычисленную в координатах "Широта/Долгота". Для данных в плановой системе координат возвращается значение -1, поскольку эти данные не могут быть преобразованы в мировую систему координат ("Широта/Долгота").

Только полигоны, эллипсы, прямоугольники и прямоугольники со скругленными углами имеют площадь. Для точки, дуги, текста или полилинии значение функции **SphericalArea()** по определению равно нулю. Для прямоугольников со скругленными углами функция **SphericalArea()** возвращает приблизительные результаты. MapBasic вычисляет площадь прямоугольников со скругленными углами, как если бы они были обычными прямоугольниками.

Примеры

Следующий пример демонстрирует вычисление площади одного географического объекта при помощи функции **SphericalArea()**. Обратите внимание, что выражение *tablename.obj*

(аналогично `states.obj`) представляет географический объект из текущей строки указанной таблицы.

```
Dim f_sq_miles As Float
Open Table "states"
Fetch First From states
f_sq_miles = Area(states.obj, "sq mi")
```

Также можно совмещать функцию **SphericalArea()** и **оператор Select**, как показано в следующем примере.

```
Select state, SphericalArea(obj, "sq km")
From states Into results
```

См. также:

функция CartesianArea(), функция SphericalArea()

Функция SphericalConnectObjects()

Назначение

Возвращает объект, представляющий кратчайшее или самое большое расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalConnectObjects( object1, object2, min )
```

object1 и *object2* — выражения, указывающие на объекты.

min — логическое выражение. Если это выражение принимает значение "Да" (TRUE), функция вычисляет минимальное расстояние между объектами, а если выражение принимает значение "Нет" (FALSE), вычисляется максимальное расстояние.

Возвращаемая величина:

Оператор возвращает объект типа "Полилиния", состоящий из одного сегмента и двух точек, и представляющий собой кратчайшее расстояние (`min == TRUE`) или самое большое расстояние (`min == FALSE`) между объектами *object1* и *object2*.

Описание

Одна точка из полученного в результате объекта-полилинии принадлежит объекту — *object1*, а другая точка принадлежит объекту — *object2*. Обратите внимание, что расстояние между двумя исходными объектами можно вычислить с помощью функции **ObjectLen()**. Если существует несколько вариантов кратчайших или наибольших расстояний (например, полученный сегмент представляет одно расстояние, а существуют еще и другие сегменты равной длины), эти функции возвращают лишь один из вариантов. Не существует способа определить, является ли полученный объект полилинии единственным кратчайшим расстоянием.

SphericalConnectObjects() возвращает объект типа "полилиния", соединяющий *object1* и *object2* кратчайшим (`min == TRUE`) или наибольшим (`min == FALSE`) способом, полученным методом сферических расчетов. Если вычисления не могут быть произведены методом сферических расстояний (например, если используется плановая система координат), вызов этой функции приведет к ошибке.

См. также:

функция **ConnectObjects()**, функция **CartesianConnectObjects()**, оператор **Nearest**, оператор **Farthest**, функция **CartesianObjectDistance()**, функция **ObjectDistance()**, функция **SphericalObjectDistance()**

Функция SphericalDistance()

Назначение

Возвращает расстояние между двумя точками, заданными координатами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalDistance( x1, y1, x2, y2, unit_name )
```

x1 и *x2* – координаты по оси X (например, долгота).

y1 и *y2* – координаты по оси Y (например, широта).

unit_name – строка, определяющая единицы измерения расстояния (к примеру, "км.").

Возвращаемая величина

Вещественное – Float

Описание

Функция **SphericalDistance()** вычисляет расстояние между двумя точками.

Она возвращает значение в указанных единицах *unit_name*; например, чтобы получить расстояние в милях, задайте "mi" в качестве *unit_name*. Список доступных единиц измерения смотрите в разделе **оператор Set Distance Units**.

Координаты по осям X и Y должны быть заданы в текущей системе координат MapBasic. По умолчанию MapInfo Pro предполагает, что координаты заданы в мировой системе координат ("Широта/Долгота"). Можно установить систему координат по умолчанию, выполнив **оператор Set CoordSys**.

Функция **SphericalDistance()** всегда возвращает площадь на сфере, вычисленную в координатах "Широта/Долгота" по дугам больших кругов. Для данных в плановой системе координат возвращается значение -1, поскольку эти данные не могут быть преобразованы в мировую систему координат ("Широта/Долгота").

Пример

```
Dim dist, start_x, start_y, end_x, end_y As Float
Open Table "cities"
Fetch First From cities
start_x = CentroidX(cities.obj)
start_y = CentroidY(cities.obj)
Fetch Next From cities
end_x = CentroidX(cities.obj)
end_y = CentroidY(cities.obj)
dist = SphericalDistance(start_x, start_y, end_x, end_y, "mi")
```

См. также:

функция CartesianDistance(), функция Distance()

Функция SphericalObjectDistance()

Назначение

Возвращает расстояние между двумя объектами. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalObjectDistance( object1, object2, unit_name )
```

object1 и *object2* — выражения, указывающие на объекты.

unit_name — строка, определяющая единицы измерения расстояния.

Возвращаемая величина

Вещественное – Float

Описание

SphericalObjectDistance() возвращает минимальное расстояние между объектами *object1* и *object2*, вычисленное по методу сферических расчетов в единицах *unit_name*. Если вычисления не могут быть произведены методом сферических расстояний (например, если используется плановая система координат), вызов этой функции приведет к ошибке.

См. также:

функция **ObjectDistance()**, функция **CartesianObjectDistance()**, оператор **Nearest**, оператор **Farthest**, функция **SphericalConnectObjects()**, функция **ConnectObjects()**, функция **CartesianConnectObjects()**

Функция SphericalObjectLen()

Назначение

Возвращает географическую протяженность объекта линии или полилинии. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalObjectLen( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – строка, определяющая единицы измерения расстояния (к примеру, “км.”).

Возвращаемая величина

Вещественное – Float

Описание

Функция **SphericalObjectLen()** возвращает длину объекта. Помните, что ненулевые длины имеют только объекты линий и полилиний, для измерения периметра прямоугольника, эллипса или полигона, используется [функция Perimeter\(\)](#).

Функция **SphericalObjectLen()** всегда возвращает площадь на сфере, вычисленную в координатах "Широта/Долгота". Для данных в плановой системе координат возвращается значение -1, поскольку эти данные не могут быть преобразованы в мировую систему координат ("Широта/Долгота").

Функция **SphericalObjectLen()** возвращает длину в единицах измерения, указанных в параметре *unit_name*. Например, чтобы получить длину в милях, укажите "миль" в качестве значения параметра *unit_name*. Список допустимых единиц измерения смотрите в разделе: [Оператор Set Distance Units](#).

Пример

```
Dim geogr_length As Float
Open Table "streets"
Fetch First From streets
geogr_length = SphericalObjectLen(streets.obj, "mi")
```

geogr_length теперь представляет длину сегмента улицы в милях.

См. также:

[функция CartesianObjectLen\(\)](#), [функция SphericalObjectLen\(\)](#)

Функция SphericalOffset()

Назначение

Возвращает копию исходного объекта, сдвинутую на указанное расстояние и угол в сферических координатах. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalOffset( object, angle, distance, units )
```

object – объект, подвергаемый сдвигу.

angle – угол перемещения объекта.

distance – расстояние перемещения объекта.

units – строка, представляющая единицы измерения расстояния *distance*.

Возвращаемая величина

Объект

Описание

Эта функция создает новый объект, являющийся копией исходного и сдвинутый на расстояние *distance* вдоль угла *angle* (в градусах, где 0 градусов соответствует положительному направлению по оси X, а угол увеличивается против часовой стрелки). Строка, задающая единицы измерения, аналогична тем, которые используют [функция ObjectLen\(\)](#) и [функция Perimeter\(\)](#). Тип данных измерения расстояний (DistanceType) – сферический. Если системой координат исходного объекта является план-схема, возникнет ошибка, поскольку сферические координаты для план-схем не могут быть определены. В случае ошибки возвращается объект NULL. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Если переместить объект в мировой системе координат, контуры объекта сохраняются, однако площадь объекта изменяется. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если используется мировая система координат, для преобразования в градусы используется фиксированная точка. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
SphericalOffset(Rect, 45, 100, "mi")
```

См. также:

[функция SphericalOffsetXY\(\)](#)

Функция SphericalOffsetXY()

Назначение

Возвращает копию исходного объекта, передвинутую в точку и на указанное расстояние в сферических координатах. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalOffsetXY( object, xoffset, yoffset, units )
```

object – объект, подвергаемый сдвигу.

xoffset и *yoffset* – расстояния по осям x и y, на которые требуется сдвинуть объект.

units – строка, представляющая единицы измерения расстояния distance.

Возвращаемая величина

Объект

Описание

Функция **SphericalOffsetXY()** создает копию исходного объекта, сдвинутую на *xoffset* по оси X и на *yoffset* по оси Y. Строка, задающая единицы измерения, аналогична тем, которые используют [функция ObjectLen\(\)](#) и [функция Perimeter\(\)](#). Тип данных измерения расстояний (DistanceType) – сферический. Если системой координат исходного объекта является план-схема, возникнет ошибка, поскольку сферические координаты для план-схем не могут быть определены. В случае ошибки возвращается объект NULL. Используется координатная система исходного объекта.

При проведении вычислений в сферической системе координат следует учитывать некоторые особенности, которые не встречаются при применении декартовой системы координат. Если переместить объект в мировой системе координат, контуры объекта сохраняются, однако площадь объекта изменяется. Это происходит потому, что когда сдвига определяется в градусах, реальное измеренное расстояние в различных точках неодинаково.

Для функций сдвига реальная дельта переноса вычисляется в некоторой фиксированной точке объекта (например, в центре описанного прямоугольника), после чего это значение преобразуется из исходных единиц измерения в единицы измерения системы координат. Если используется мировая система координат, для преобразования в градусы используется фиксированная точка. Действительное расстояние сдвига может быть неодинаковым для различных точек объекта. Расстояние от исходного объекта до его перемещённой копии будет точным только в фиксированной точке.

Пример

```
SphericalOffsetXY(Rect, 92, -22, "mi")
```

См. также:

[функция SphericalOffset\(\)](#)

Функция SphericalPerimeter()

Назначение

Возвращает периметр графического объекта. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SphericalPerimeter( obj_expr, unit_name )
```

obj_expr – выражение, определяющее объект.

unit_name – строка, определяющая единицы измерения расстояния (к примеру, “км.”).

Возвращаемая величина

Вещественное – Float

Описание

Функция **SphericalPerimeter()** вычисляет периметр объекта *obj_expr*. Функция **SphericalPerimeter()** определена для следующих типов объектов: эллипсы, прямоугольники, прямоугольники со скругленными углами, многоугольники. Другие типы объектов имеют периметр нулевой длины. Функция **SphericalPerimeter()** возвращает длину в единицах измерения, указанных в параметре *unit_name*. Например, чтобы получить длину в милях, укажите “миль” в качестве значения параметра *unit_name*. Список допустимых единиц измерения смотрите в разделе: [Оператор Set Distance Units](#).

Функция **SphericalPerimeter()** всегда возвращает площадь на сфере, вычисленную в координатах "Широта/Долгота". Для данных в плановой системе координат возвращается значение -1, поскольку эти данные не могут быть преобразованы в мировую систему координат ("Широта/Долгота"). Для прямоугольников со скругленными углами функция **SphericalPerimeter()** возвращает приблизительные результаты. MapBasic вычисляет периметр прямоугольников со скругленными углами как если бы они были обычными прямоугольниками.

Пример

В следующем примере демонстрируется применение функции **SphericalPerimeter()** для вычисления периметра конкретного географического объекта.

```
Dim perim As Float
Open Table "world"
Fetch First From world
perim = SphericalPerimeter(world.obj, "km")
```

Переменная `perim` теперь содержит периметр полигона, который соответствует первой записи таблицы `world`.

Функцию **SphericalPerimeter()** также можно использовать, вставив в **оператор Select**. Следующий **оператор Select** извлекает информацию из таблицы `States` и сохраняет результаты во временной таблице `Results`. Поскольку **оператор Select** содержит вызов функции **SphericalPerimeter()**, таблица `Results` будет включать колонку, отражающую периметр каждого из штатов.

```
Open Table "states"
Select state, Perimeter(obj, "mi")
From states
Into results
```

См. также:

функция CartesianPerimeter(), функция Perimeter()

Функция Sqr()

Назначение

Вычисляет квадратный корень. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Sqr ( num_expr )
```

`num_expr` – численное выражение, результатом которого должно быть положительное число.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Sqr()** возвращает квадратный корень от числа, полученного в результате вычисления выражения *num_expr*. Это число должно быть большим или равным нулю.

Извлечение квадратного корня эквивалентно возведению числа в степень 0.5. Соответственно выражение **Sqr(*n*)** эквивалентно $n ^ 0.5$. Функция MapBasic **Sqr()**, однако, быстрее вычисляет квадратный корень, чем возведение в степень 0.5.

Пример

```
Dim n As Float
n = Sqr(25)
```

См. также:

функция **Cos()**, функция **Sin()**, функция **Tan()**

Оператор StatusBar

Назначение

Показывает или прячет строку сообщений в рабочем окне MapInfo или выдает в ней сообщение. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StatusBar { Show | Hide }
[ Message message ]
[ ViewDisplayPopup { On | Off } ]
[ EditLayerPopup { On | Off } ]
```

message – это сообщение, которое будет показано в строке сообщений.

Описание

Оператор **StatusBar** управляет отображением строки сообщений в рабочем окне MapInfo, а также позволяет печатать в ней сообщения из прикладной программы.

Чтобы напечатать сообщение в строке сообщений, используйте предложение **Message**.

```
StatusBar Message "Calculating coordinates..."
```

MapInfo Pro автоматически заменит сообщение другими, когда пользователь будет перемещать указатель мышки по командам меню или кнопкам инструментальных панелей. Поэтому текст сообщения должен быть лаконичен, чтобы пользователь успел его заметить и прочесть за короткий промежуток времени. По этой причине не советуем помещать важные сообщения в строку сообщений.

Для этих целей можно выполнить **оператор Print**, который откроет окно Сообщений.

Используйте параметр **ViewDisplayPopup**, позволяющий изменять вид строки сообщений. Если этот параметр установлен на значение **On**, то пользователь сможет изменять масштаб и положение курсора, устанавливаемых из строки сообщений.

Использование параметра **EditLayerPopup** позволяет пользователю устанавливать редактируемый слой в окне карты из строки сообщений. Если этот параметр установлен на значение **On**, то пользователь сможет выбирать редактируемый слой из строки сообщений.

См. также:

оператор Note, **оператор Print**

Оператор Stop

Назначение

Приостанавливает выполнение прикладной программы для отладки. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Stop
```

Предупреждение!

Оператор **Stop** нельзя выполнить во внутренних функциях программы. Нельзя выполнить оператор **Stop** в обработчиках элемента диалога **оператор Dialog**, так как отладочный процесс невозможен, пока открыто диалоговое окно.

Описание

Оператор **Stop** является отладочным средством. Оператор приостанавливает выполнение программы и передает управление пользователю. В данном случае пользователь – это программист, который отлаживает свою программу.

Если программа была приостановлена при активном окне **MapBasic**, то в нем выводится сообщение с номером строки программы, в которой был выполнен оператор **Stop**.

Если окно MapBasic не было открыто, то оно будет открыто. Далее пользователь может использовать окно **MapBasic** для исследования текущего состояния программы. Если набрать:

```
? Dim
```

в окне **MapBasic**, то MapInfo Pro выведет список всех используемых локальных переменных в программе. Если набрать:

```
? Global
```

в окне **MapBasic**, то MapInfo Pro выведет список всех используемых глобальных переменных в программе.

Задав после знака вопроса имя переменной, пользователь может получить ее текущее значение. Пользователь сам может присвоить значение какой-нибудь переменной. Формат присвоения следующий:

```
variable_name = new_value
```

где *variable_name* – имя локальной или глобальной переменной, и *new_value* – выражение, представляющее новую величину для этой переменной.

Для продолжения выполнения приостановленной программы нужно выполнить команду **Файл > Продолжить** (команда **Продолжить** появится в меню Файл вместо команды **Выполнить**). Выполнив **оператор Continue** в окне **MapBasic**, можно продолжить выполнение программы.

В режиме остановки (после оператора **Stop**) MapInfo Pro продолжает держать открытым файл приложения. Поэтому его нельзя перекомпилировать. Поэтому выйдите из режима **остановки** командой **Файл > Продолжить** программу, если хотите перекомпилировать файл.

См. также:

Оператор Continue

Функция Str\$

Назначение

Возвращает строковое представление числа, объекта и стиля. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Str$( expression )
```

expression – выражение, результатом которого является число или величина типа Date, Pen, Brush, Symbol, Font и Object.

Возвращаемая величина

Строка – String

Описание

Функция **Str\$()** возвращает строку, которая представляет величину, полученную в результате вычисления выражения.

Если параметром *expression* является отрицательное число, то результатом будет строка, начинающаяся с символа знака минус (-). Если число положительно, то строка начинается с пробела.

Длина строки, возвращаемая функцией **Str\$()**, зависит от точности вычисления выражения *expression* и количества цифр справа от запятой. Другими словами функция **Str\$()** возвращает строку, которая представляет собой округленную величину. Чтобы управлять преобразованием, используется [функция Format\\$\(\)](#).

Если величина в *expression* имеет объектный тип (Object) функция **Str\$()** вернет значения одного из следующих типов: Дуга, Эллипс, Рамка, Линия, Точка, Полилиния, Прямоугольник, Область, Скруглённый прямоугольник или Текст.

Если параметр *expression* задает объект (тип Object) выражением в форме *tablename.obj* (где *tablename* – имя открытой таблицы) и если к текущей строке таблицы не присоединен графический объект, то функция **Str\$()** вернет пустую строку.

Примечание: Передача неинициализированной объектной переменной приведет к тому, что функция **Str\$()** вернет ошибку.

Если параметр *expression* задает дату (Date) функция **Str\$()** вернет строку в формате, установленном в вычислительной системе, в которой выполняется эта прикладная программа. Например, следующее выражение:

```
Str$( NumberToDate(19951231) )
```

может быть равно и “31.12.95”, и “1995/12/31” (и т.п.), в зависимости от того, какой формат использует пользователь в своем компьютере. Для управления форматом преобразования функцией **Str\$()** используйте [оператор Set Format](#).

Если параметр *expression* задает число, то функция **Str\$()** будет использовать точку в качестве десятичной точки, даже если в компьютере пользователя используется другой знак. Функция

Str\$() не использует разделитель тысяч при конвертировании числа в строку. Чтобы получить число с разделителями тысяч и десятичной точкой такой, какая задана в данной вычислительной платформе, используется **функция FormatNumber\$()**.

Пример

```
Dim s_spelled_out As String, f_profits As Float
f_profits = 123456
s_spelled_out = "Annual profits: $" + Str$(f_profits)
```

См. также:

функция Format\$(), функция FormatNumber\$(), оператор Set Format, функция Val()

Функция String\$()

Назначение

Строит строку, повторяя символ заданное количество раз. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
String$( num_expr, string_expr )
```

num_expr — положительное целочисленное выражение.

string_expr — строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **String\$()** возвращает строку, состоящую из первых символов строки, заданной параметром *string_expr*. Поэтому параметр *num_expr*, который задает длину будущей строки (в символах), должен быть положительным коротким целым числом (smallint).

Пример

```
Dim filler As String
filler = String$(5, "ABCDEFGH")
' at this point, filler contains the string "AAAAA"
' (5 copies of the 1st character from the string)
```

См. также:

функция [Space\\$\(\)](#)

Функция StringByteLength()

Назначение

Возвращает количество байт в строке заданного набора символов (charset). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringByteLength( string_expr, charset )
```

String_expr — строка символов, для которой требуется вычислить количество байт, используемое строкой.

Charset — строка, представляющая набор символов, используемый в строке *string_expr*. Смотрите раздел: [Предложение Charset](#), где перечислены корректные наборы символов.

Возвращаемая величина:

Целое – Integer

Описание

Эта функция возвращает количество байт в строке заданного набора символов. Иногда полезно знать количество байт, которое необходимо, для того чтобы отобразить строку с многобайтовыми наборами символов (например: UTF-8 или WindowsJapanese). В строке с набором символов UTF-8 каждый из символов может занимать до 4 байт.

Если в переменную *string_expr* или *charset* передаётся несимвольное значение, то выдаётся сообщение об ошибке.

Если в переменную *charset* передаётся некорректный набор символов, то выдаётся сообщение об ошибке.

Если никакого значения не передаётся в параметре *charset*, то передаётся сообщение об ошибке.

Примеры

```
print StringByteLength("滋賀県草津市草津", "UTF-16")
```

В примере, приведённом выше, возвращается число 16, поскольку каждый из восьми японских символов занимает два байта, когда используется UTF-16.

```
print StringByteLength("滋賀県草津市草津", "UTF-8")
```

В примере, приведённом выше, возвращается число 24, поскольку в UTF-8 каждый из восьми японских символов занимает три байта.

```
print StringByteLength("滋賀県草津市草津", "WindowsJapanese")
```

В примере, приведённом выше, возвращается число 16, поскольку в WindowsJapanese каждый из восьми японских символов занимает два байта.

```
Print StringByteLength("قبل البدء في استعمال", "UTF-8")
```

В примере, приведённом выше, возвращается число 37, поскольку в UTF-8 каждый из двадцати арабских символов занимает от одного до трёх байт.

Пример использования:

При сохранении таблицы (например: World.tab) с набором символов WindowsArabic в новую таблицу с набором символов UTF-8, максимальную длину, необходимую для колонки, можно оценить:

```
select Max(StringByteLength(World.Coll, "UTF-8")) from World
```

См. также:

- [функция Len\(\)](#)
- [предложение Charset](#)

Функция StringCompare()

Назначение

Сравнивает две строки, учитывая различия строчных и прописных символов. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringCompare( string1, string2 )
```

string1 и *string2* – строковые выражения.

Возвращаемая величина

Короткое целое число -1, 1 или 0. Величина типа SmallInt.

Описание

Функция **StringCompare()** позволяет сравнивать строки с учетом регистра. MapBasic, сравнивая строки с использованием операции "=", не учитывает разницу между прописными и строчными буквами. Например, в операторе:

```
If "ABC" = "abc" Then
```

сравнение возвращает TRUE (истину).

Функция **StringCompare()** позволяет делать сравнения строк, рассматривая строчные и прописные буквы как разные. Сравнение строк происходит посимвольно. Как только встречаются два разных символа, выполнение функции прекращается.

Возвращаемая величина:	Результат:
-1	Код символа из первой строки меньше, чем код соответствующего символа из второй строки.
0	Две строки равны.
1	Код символа из первой строки больше, чем код соответствующего символа из второй строки.

Пример

Функция `StringCompare("ABC", "abc")` возвращает значение -1 ($n = -1$), поскольку ANSI-код "A" меньше "a".

См. также:

[функция Like\(\)](#), [функция StringCompareIntl\(\)](#)

Функция StringCompareIntl()

Назначение

Сравнивает две строки, учитывая особенности сортировки для разных языков. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringCompareIntl( string1, string2 )
```

string1 и *string2* – строковые выражения.

Возвращаемая величина

Короткое целое число -1, 1 или 0. Величина типа SmallInt.

Описание

Функция **StringCompareIntl()** позволяет сравнивать две строки, учитывая языковые особенности, заданные в Control Panel (Панели управления Windows). В состав сравниваемых строк могут входить символы, используемые в других языках.

Сравнение строк происходит посимвольно. Как только встречаются два разных символа, выполнение функции прекращается.

Возвращаемая величина:	Результат:
-1	Код символа из первой строки меньше, чем код соответствующего символа из второй строки.
0	Две строки равны.
1	Код символа из первой строки больше, чем код соответствующего символа из второй строки.

См. также:

[функция Like\(\)](#), [функция StringCompare\(\)](#)

Функция StringToDate()

Назначение

Переводит строку в величину даты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringToDate ( "datestring" )
```

datestring – строка, представляющая дату. Можно пользоваться последовательностью 17 цифр:

- 4 для года
- По 2 для месяцев, дней, часов, минут и секунд
- 3 для миллисекунд

Последовательность цифр можно задавать либо в виде строки с кавычках, либо в виде числа. Если введено менее 17 цифр, то строка и числительное интерпретируются по-разному. К строке нули добавляются справа, а к числительному – слева, "20150909" будет рассматриваться как строка "201509090000000000", а 20150909 – как 00000000020150909.

Возвращаемая величина

DateTime – специальный тип

Описание

Функция **StringToDate()** создает из строки величину типа Date. MapBasic интерпретирует данные в соответствии с форматами, которые установлены в компьютере, в котором работает программа. Компьютеры часто настроены отображать даты в формате День/Месяц/Год, а качестве символа-разделителя, обычно, используются либо точка (.), либо косая черта (/). Компьютеры в США настроены на формат дат Месяц/День/Год. Для того чтобы заставить функцию **StringToDate()** применять американские правила форматирования, используйте **оператор Set Format**.

Примечание: Независимо от установок на компьютере, используйте **функцию NumberToDate()** вместо **StringToDate()**. **Функция NumberToDate()** не зависит от настроек персонального компьютера.

Если используется формат США, то параметр *datestring* должен содержать компоненты числа, месяца и года, разделенные косой чертой (/). Первые два символа задают месяц, значение

которого должно находиться в диапазоне от 1 до 12. Два символа второй компоненты задают число, значение которого должно находиться в диапазоне от 1 до 31. Год может задаваться двумя или четырьмя символами. Если год не указан, то принимается значение настоящего года. Если задать год двумя цифрами (например, 96), то MapInfo Pro припишет этот год к веку, который задает **оператор Set Date Window()**.

Пример

Данный пример показывает строку типа Date в формате, принятом в США: Месяц/День/Год. Эта программа вызывает **оператор Set Format** без использования функции **StringToDate()**, чтобы убедиться, что строки типа Date в формате США интерпретируются корректно независимо от установок системы.

```
Dim d_start, d_end As Date
Set Format Date "US"
d_start = StringToDate("12/17/92")
d_end = StringToDate("01/02/1995")
Set Format Date "Local"
```

В этом примере переменная Date1 = 19890120, Date2 = 20101203 и MyYear = 1990.

```
DIM Date1, Date2 as Date
DIM MyYear As Integer
Set Format Date "US"
Set Date Window 75
Date1 = StringToDate("1/20/89")
Date2 = StringToDate("12/3/10")
MyYear = Year("12/30/90")
```

Эти результаты предопределяет **оператор Set Date Window()** позволяющий контролировать год в двухзначном формате.

См. также:

функции даты и времени, **функция NumberToDate()**, **функция NumberToDateTime()**, **оператор Set Format**, **функция Str\$()**, **функция StringToDateTime()**, **функция StringToTime()**

Функция StringToDateTime()

Назначение

Возвращает значение типа DateTime (Дата/Время) на основе сведений из строки, в которой перечислены дата и время. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringToDateTime( string )
```

string – строка, представляющая дату и время.

Возвращаемая величина

ДатаВремя

Описание

MapBasic интерпретирует данные типа **DateTime** в соответствии с форматами, заданными в компьютере, где работает программа. Между датой и временем требуется хотя бы один пробел. См. также описания функций **StringToDate() function** и **StringToTime function** в Справочнике MapBasic.

Пример

В этом примере используется формат времени, принятый в США, в котором запятые разделяют часы (hh), минуты (mm), секунды (ss) и десятичные доли секунд (fff) (подробнее смотрите разделы: **Время** или **функция FormatTime\$()**),

```
dim strX as string
dim Z as datetime
strX = "1999:09:25:12:32:45"
Z = StringToDateTime(strX)
Print FormatDate$(Z)
Print FormatTime$(Z, "hh:mm:ss.fff tt")
```

См. также:

функции даты и времени, функция NumberToDateTime(), оператор Set Format, функция Str\$(), функция StringToDate(), функция StringToTime()

Функция StringToTime()

Назначение

Возвращает значение типа Time (Время) на основе сведений из строки с временем. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringToTime ( string )
```

string – строка, представляющая время.

Возвращаемая величина

Время

Описание

MapBasic интерпретирует данные типа Time в соответствии с форматом, заданным в компьютере, где работает программа. При этом допускается либо 12-часовой, либо 24-часовой отсчет времени. Менее важные компоненты времени можно опускать, поэтому требуется задавать только часы, а минуты, секунды и миллисекунды являются дополнительными, необязательными.

Пример

В этом примере используется формат времени, принятый в США, в котором запятые разделяют часы (hh), минуты (mm), секунды (ss) и десятичные доли секунд (fff) (подробнее смотрите разделы: **Время** или **функция FormatTime\$()**),

```
dim strY as string
dim X as time
strY = "12:32:45"
X = StringToTime(strY)
Print FormatTime$ (X, "hh:mm:ss.fff tt")
```

См. также:

функции даты и времени, функция NumberToDateTime(), функция NumberToTime(), оператор Set Format, функция Str\$(), функция StringToTime()

Функция StyleAttr()

Назначение

Возвращает значение одной из компонент стиля оформления объекта: Pen, Brush, Font или Symbol. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StyleAttr( style, attribute )
```

style – величина, выражающая стиль (величина типа Pen, Brush, Font или Symbol).

attribute – целочисленный код, управляющий результатом функции.

Возвращаемая величина

Целое число или строка, в зависимости от значения параметра attribute.

Описание

Функция **StyleAttr()** извлекает из величины типа Pen, Brush, Symbol или Font определенную компоненту.

Все типы в MapBasic являются составными. Например, определение стиля Brush состоит из трех компонентов: pattern, foreground color и background color. При вызове функции **StyleAttr()**, параметр *attribute* контролирует, какой из атрибутов возвращается.

Параметр *attribute* должен являться одним из кодов, показанных в нижеследующей таблице. Коды в левом столбце (например, PEN_WIDTH) определены в файле стандартных определений MAPBASIC.DEF.

Штриховки:

Параметры	ID	StyleAttr() возвращает:
BRUSH_PATTERN	1	Целое число (Integer), атрибут стиля Brush, задающий номер штриха.
BRUSH_FORECOLOR	2	Целое число (Integer), атрибут стиля Brush, задающий RGB-код цвета штриха.
BRUSH_BACKCOLOR	3	Целое число (Integer), атрибут стиля Brush, задающий RGB-код цвета фона штриха или -1, если фон прозрачный.

Шрифты:

Параметры	ID	StyleAttr() возвращает:
FONT_NAME	1	Атрибут стиля Font, задающий строку с именем шрифта.

Параметры	ID	StyleAttr() возвращает:
FONT_STYLE	2	Целое число (Integer) от 0 до 7, атрибут стиля Font (0 = простой, 1 = жирный, и т.д.); см. раздел Предложение Font для более подробной информации.
FONT_POINTSIZE	3	Целое число (Integer), атрибут стиля Font, задающий размер шрифта в точках. Примечание: Если текстовый объект принадлежит таблице, а не Отчету, то размер будет равен 0 и высота букв будет определяться текущим масштабом карты.
FONT_FORECOLOR	4	Целое число (Integer), атрибут стиля Font, задающий RGB-код цвета символов строки.
FONT_BACKCOLOR	5	Целое число (Integer), атрибут стиля Font, задающий RGB-код цвета фона строки или -1, если фон прозрачный. Если стиль шрифта включает кайму, то задает RGB-цвет каймы.

Линии:

Параметры	ID	StyleAttr() возвращает:
PEN_WIDTH	1	Целое число (Integer), атрибут стиля Pen, задающий ширину линии в точках.
PEN_PATTERN	2	Целое число (Integer), атрибут стиля Pen, задающий номер вида линии.
PEN_COLOR	4	Целое число (Integer), атрибут стиля Pen, задающий RGB-код цвета линии.
PEN_INDEX	5	Целое число (Integer), представляющее индекс линии из библиотеки линий.
PEN_INTERLEAVED	6	Логическое: TRUE если стиль пересекающихся линий.

Символы:

Параметры	ID	StyleAttr() возвращает:
SYMBOL_CODE	1	Целое число (Integer), атрибут стиля Symbol, задающий номер символа. Используется для типа символа версии MapInfo 3.0 и TrueType.
SYMBOL_COLOR	2	Целое число (Integer), атрибут стиля Symbol, задающий RGB-код цвета символа.
SYMBOL_POINTSIZE	3	Целое число (Integer) от 1 до 48, атрибут стиля Symbol, задающий размер символа в пунктах.
SYMBOL_ANGLE	4	Действительное число (Float), угол поворота символа TrueType.
SYMBOL_FONT_NAME	5	Строка (String), имя шрифта TrueType, который используется как библиотека символов.
SYMBOL_FONT_STYLE	6	Целое число (Integer), задающее написание символа TrueType (0 = нормальное написание, 1 = жирное, и т.д.). См. раздел Предложение Symbol .
SYMBOL_KIND	7	Целое число (Integer), тип символа: 1 – символ MapInfo версии 3.0; 2 – символ шрифта TrueType; 3 – символ из растрового файла.
SYMBOL_CUSTOM_NAME	8	Строка (String), имя растрового файла.
SYMBOL_CUSTOM_STYLE	9	Целое число (Integer), задающее стиль для растрового символа (0 = нормальное, 1 = показать фон и т.д.). См. раздел Предложение Symbol .

Ошибки

ERR_FCN_ARG_RANGE (644), если неправильно задано значение аргумента.

Пример

Сначала выполняется [функция CurrentPen\(\)](#) для того, что бы узнать, какой установлен сейчас стиль линии в MapInfo, затем с помощью функции **StyleAttr()** узнаем толщину линии в пикселах.

```
Include "mapbasic.def"
Dim cur_width As Integer
cur_width = StyleAttr(CurrentPen(), PEN_WIDTH)
```

См. также:

предложение [Brush](#), предложение [Font](#), предложение [Pen](#), предложение [Symbol](#), функция [MakeBrush\(\)](#), функция [MakeFont\(\)](#), функция [MakePen\(\)](#), функция [MakeSymbol\(\)](#)

Функция [StyleOverrideInfo\(\)](#)

Возвращает информацию об определенном описании единообразного оформления стилей.

Синтаксис

```
StyleOverrideInfo( window_id, layer_number, override_index, attribute )
```

window_id – целочисленный идентификатор окна Карты.

layer_number – номер слоя (1 – самый верхний слой); чтобы определить номер слоя в окне карты, используется [функция MapperInfo\(\)](#).

override_index – целочисленный номер (отсчет от 1) описания единообразного оформления стилей на данном слое.

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Тип величины зависит от значения *attribute*.

Описание

Функция возвращает информацию об определенном описании единообразного оформления стилей на заданном слое в существующем окне карты. Число *layer_number* должно корректно задавать стиль (1 представляет верхний табличный стиль и т.д.). Параметр *attribute* должен принимать одно из следующих значений в таблице; коды определены в `MAPBASIC.DEF`.

Код атрибута	ID	Возвращаемое значение LayerInfo()
STYLE_OVR_INFO_NAME	1	Имя описания единообразного оформления стилей.

Код атрибута	ID	Возвращаемое значение LayerInfo()
STYLE_OVR_INFO_VISIBILITY	2	Короткое целое (Smallint), индикатор видимости единообразного оформления; возвращает следующие значения: • STYLE_OVR_INFO_VIS_OFF (0) отключено и не показывается никогда • STYLE_OVR_INFO_VIS_ON (1) в данный момент применено на карте • STYLE_OVR_INFO_VIS_ZOOM (2) в данный момент не применено, так как задан другой экранный масштаб
STYLE_OVR_INFO_ZOOM_MIN	3	Вещественное (Float), минимальный масштаб, при котором применяется единообразное оформление стиля.
STYLE_OVR_INFO_ZOOM_MAX	4	Вещественное (Float), максимальный масштаб, при котором применяется единообразное оформление стиля.
STYLE_OVR_INFO_ARROWS	5	Логическая величина: TRUE, если в линейных объектах используется стрелка.
STYLE_OVR_INFO_NODES	6	Логическая величина: TRUE, если показываются узлы объектов слоя.
STYLE_OVR_INFO_CENTROIDS	7	Логическая величина: TRUE, если показываются центроиды объектов слоя.
STYLE_OVR_INFO_ALPHA	8	Короткое целое (Smallint), представляющее альфа-фактор для данного слоя. • 0 = полностью прозрачные. • 255 = полностью непрозрачные.
STYLE_OVR_INFO_TRANSLUCENCY	9	Короткое целое (Smallint), представляющее процент полупрозрачности для данного слоя. • 100 = полностью прозрачные. • 0 = полностью непрозрачные.
STYLE_OVR_INFO_LINE	10	Стиль линии, используемый для отображения линейных объектов. Если стилей несколько, возвращается нижнее значение Pen.

Код атрибута	ID	Возвращаемое значение LayerInfo()
STYLE_OVR_INFO_PEN	11	Стиль линии, используемый для отображения контуров объектов, имеющих площадь. Если стилей несколько, возвращается нижнее значение Pen.
STYLE_OVR_INFO_BRUSH	12	Стиль штриха, используемый для отображения объектов, имеющих площадь. Если стилей несколько, возвращается нижнее значение Brush.
STYLE_OVR_INFO_SYMBOL	13	Стиль символа, используемый для отображения точечных объектов. Если стилей несколько, возвращается нижнее значение Symbol.
STYLE_OVR_INFO_FONT	14	Стиль шрифта, используемый для отображения текстовых объектов. Если стилей несколько, возвращается нижнее значение Font.
STYLE_OVR_INFO_SYMBOL_COUNT	15	Короткое целое (SmallInt), число символьных стилей.
STYLE_OVR_INFO_LINE_COUNT	16	Короткое целое (SmallInt), число стилей линий.
STYLE_OVR_INFO_PEN_COUNT	17	Короткое целое (SmallInt), число стилей Pen.
STYLE_OVR_INFO_BRUSH_COUNT	18	Короткое целое (SmallInt), число стилей штриховки.
STYLE_OVR_INFO_FONT_COUNT	19	Короткое целое (SmallInt), число шрифтов.

Пример

```
StyleOverrideInfo(nMID, nLayer, nOverride, STYLE_OVR_INFO_PEN_COUNT)
```

См. также:

[функция LabelOverrideInfo\(\)](#), [функция LayerStyleInfo\(\)](#), [оператор Set Map](#), [функция LayerInfo\(\)](#)

Оператор Sub...End Sub

Назначение

Определяет процедуру, которую может вызвать **оператор Call**.

Синтаксис

```
Sub proc_name [ ( [ ByVal ] parameter As var_type [ , ... ] ) ]  
    statement_list  
End Sub
```

proc_name – имя процедуры.

parameter – имя параметра процедуры.

var_type – стандартный в MapBasic тип переменных (например, Integer) или сложно-определенный тип, заранее объявленный оператором Type.

statement_list список (от 0 и более) операторов процедуры.

Предупреждение!

Оператор **Sub...End Sub** нельзя выполнить в окне **MapBasic**.

Описание

Оператор **Sub...End Sub** описывает процедуру в MapBasic (или "подпрограмму"). Как только Sub-процедура определена, к ней можно обращаться из любой другой части программы **оператор Call**.

Каждую процедуру, задаваемую оператором **Sub...End Sub**, должен быть заранее объявить **оператор Declare Sub**.

Каждая подпрограмма может иметь или не иметь параметры, значения которых передаются из процедуры, вызвавшей эту Sub-процедуру. *Parameter* – имя параметра. Каждое такое определение параметра процедуры должно быть уникально. Параметры для подпрограммы задаются списком через запятую.

По умолчанию, каждый параметр Sub-процедуры определен "ссылкой" ("by reference"). Для каждого такого параметра при вызове sub-процедуры, имя этого "ссылочного" параметра должно стоять в списке параметров вызова. Далее, если Sub-процедура изменяет такой "ссылочный" параметр, то эти изменения передаются и в соответствующую переменную, заданную при вызове Sub-процедуры. Таким образом, параметр, передаваемый ссылкой, позволяет не только доставлять значение из вызывающей процедуры, но и возвращать

значение обратно в этом же параметре. Если при объявлении использовалось ключевое слово **ByVal** то параметр процедуры передается Sub-процедуре значением ("by value"). Когда параметр передается значением, то Sub-процедура получает значение выражения из записи вызова; таким образом, вызывающая программа может передавать в Sub-процедуру выражения и константы Sub-процедура может менять значение параметров **ByVal**, не затрагивая при этом переменных, предназначенных для возврата результата.

Sub-процедура может обрабатывать массивы. При задании оператора **Sub...End Sub** в списке параметров процедуры массив объявляется с помощью пустых скобок, следующих за именем переменной (аналогично объявляется **оператор Declare Sub**). В следующем примере в процедуре "ListProcessor" объявлен в качестве параметра массив целых величин "items":

```
Sub ListProcessor(items() As Integer)
```

При вызове такой процедуры параметром вызова должно быть имя массива переменных целого типа без круглых скобок.

Если в процедуре объявляется локальная переменная с таким же именем, как у существующей уже глобальной переменной, то под данным именем в процедуре используется только локальная переменная.

Sub-процедура завершается, если выполняется **оператор Exit Sub**.

Нельзя передавать процедурам массивы, переменные заданного пользователем типа или переменные-синонимы параметрами **ByVal** (by-value). Однако, все такие данные можно передавать параметрами-ссылками.

Пример

Процедура "Cube" возводит число в куб (в третью степень) и возвращает результат. Процедура имеет два параметра: первый для возводимого степень вещественного числа, второй для результата.

```
Declare Sub Main
Declare Sub Cube(ByVal original As Float, cubed As Float)

Sub Main
  Dim x, result As Float
  Call Cube(2, result)
  ' result now contains the value: 8 (2 x 2 x 2)
  x = 1
  Call Cube(x + 2, result)
  ' result now contains the value: 27 (3 x 3 x 3)
End Sub

Sub Cube (ByVal original As Float, cubed As Float)
  ' Cube the "original" parameter value, and store
  ' the result in the "cubed" parameter.
  cubed = original ^ 3
End Sub
```

См. также:

[оператор Call](#), [оператор Declare Sub](#), [оператор Dim](#), [оператор Exit Sub](#), [оператор Function...End Function](#), [оператор Global](#)

Предложение Symbol

Назначение

Позволяет задавать стиль символа, которым отображаются точечные объекты, например, начертание, цвет и размер. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

```
Symbol ( shape, color, size )
```

Shape – специальный символ, например, кружок или звёздочка, который можно взять в стандартном наборе символа MapInfo Pro, смотрите [Символы MapInfo 3.0](#).

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

Symbol – предложение не является отдельным оператором MapBasic. Оно задаёт параметры, которые определяют внешний точечного объекта: операторы, связанные с объектами, например, [функция CreatePoint\(\)](#), используют предложение **Symbol** для задания стиля оформления объекта.

См. также:

[функция CreatePoint\(\) function](#), [функция RGB\(\)](#), [символы MapInfo 3.0](#), [функция StyleAttr\(\)](#)

Символы MapInfo 3.0

Синтаксис

```
Symbol ( shape, color, size )
```

shape – целое число, 31 или больше, задающее, какой из символов стандартного набора MapInfo Pro следует использовать. 31 или большее, задающее символ из стандартного набора MapInfo Pro (см. таблице ниже). Стандартный набор условных знаков включает символы от 31 до 67, но пользователь может отредактировать любой символ с помощью программы Символ.

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

Описание

Symbol – предложение не является отдельным оператором MapBasic. Оно задаёт параметры, которые определяют внешний точечного объекта: операторы, связанные с объектами, например, **функция CreatePoint()**, используют предложение **Symbol** для задания стиля оформления объекта.

Некоторые операторы MapBasic (такие как **Alter Object...Info OBJ_INFO_SYMBOL**) используют выражение типа **Symbol** как параметр без ключевого слова **Symbol**.

Следующая таблица содержит стандартные символы для версии MapInfo 3.0:

31		41	☆	51	✱	61	🛡
32	■	42	△	52	✈	62	🛡
33	◆	43	▽	53	🚩	63	🚩
34	●	44	■	54	🚩	64	✂
35	☆	45	▲	55	🏠	65	⚓
36	▲	46	●	56	✚	66	🚧
37	▽	47	➡	57	🏠	67	🚧
38	□	48	➡	58	⚓		
39	◇	49	+	59	🔍		
18	○	50	×	60	🏠		

Пример

Оператор Set Map может использовать параметр **Symbol**. **Оператор Set Map** назначает стиль символов для показа точечных объектов первого слоя карты в виде кружочков (символ номер 34), закрашенных красным и размером в 18 пунктов.

```
Include "mapbasic.def"
Set Map
  Layer 1 Display Global
  Global Symbol MakeSymbol(34,RED,18)
```

Шрифт True Type

Синтаксис

```
Symbol( shape, color, size, fontname, fontstyle, rotation )
```

shape – целое число, 32 или большее, задающее символ из шрифта TrueType (значение 32 задает невидимый символ);

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

fontname – строка с именем шрифта TrueType (например, "WingDings");

fontstyle – целочисленный код, величина типа Integer, управляющий написанием шрифта;

rotation – вещественное число, угол поворота в градусах.

Описание

Если задаёте символ из шрифта TrueType, то параметр *fontstyle* управляет написанием символа:

Значение <i>fontstyle</i>	Стиль символа
0	Обычный
1	Жирный
16	Черная кайма
32	Тень
256	Белая кайма

Чтобы скомбинировать два и более эффектов, числа нужно сложить. Например, для того, чтобы получить символ жирного и оттененного написания, параметр *fontstyle* должен быть равен 33. Стили со значениями 16 и 256 взаимно исключают друг друга.

Растровые символы пользователя

Синтаксис

```
Symbol( filename, color, size, customstyle )
```

filename – строка до 31 символа длиной с именем растрового файла (файл должен находиться в каталоге, заданном пользователем);

color – целочисленный код цвета в системе RGB, смотрите раздел: [Функция RGB\(\)](#).

size – целое число, размер символа в точках от 1 до 48;

customstyle – целочисленный код типа Integer, управляющий цветом и фоном символа. Смотрите таблицу ниже.

Описание

Если задать новый символ, то параметр *customstyle* управляет, какими будут цвет и фон символа:

значение *customstyle* Стиль символа

0	Не действуют режимы из группы Эффекты диалога Стиль символа, и символ появляется таким, какой он есть. Все белые пиксели растра прозрачны.
1	Действует режим Добавить фон; все белые пиксели растра непрозрачны.
2	Действует режим Покрасить одним цветом; все небелые точки растра закрашены одним цветом.
3	Установлены оба флажка (действуют оба режима).
4	Действует режим Показать реальный размер; растровый символ визуализируется в свою натуральную величину.
5	Действуют режимы Добавить фон и Показать реальный размер.
7	Действуют режимы: Добавить фон, Покрасить одним цветом и Показать реальный размер.

Выражения, создающие символ

Синтаксис

```
Symbol symbol_expr
```

symbol_expr – выражение, результат которого есть величина типа Symbol, например, переменная типа Symbol или MakeSymbol(shape, color, size).

См. также:

[функция MakeCustomSymbol\(\)](#), [функция MakeFontSymbol\(\)](#), [функция MakeSymbol\(\)](#), [функция StyleAttr\(\)](#)

Функция SystemInfo()

Назначение

Возвращает информацию об активной операционной среде и версии программы MapInfo. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
SystemInfo( attribute )
```

attribute — целочисленный код, определяющий тему запроса.

Примечание: Константа, определенная для этой функции в файле MAPBASIC.DEF, равна 18.

Возвращаемая величина

Величина типа SmallInt, Logical или String

Описание

Функция **SystemInfo()** возвращает информацию об оперативной среде и версии программы MapInfo Pro, в которых вы сейчас работаете. Вид информации задает параметр attribute. Коды определены в MAPBASIC.DEF.

Код атрибута	ID	Возвращаемое значение SystemInfo()
SYS_INFO_PLATFORM	1	Целое число типа Integer: код, определяющий операционную систему. Возвращенное число будет кодом PLATFORM_WIN.
SYS_INFO_APPVERSION	2	Целое число типа Integer: номер версии MapBasic, в котором было скомпилировано приложение, умноженный на 100.
SYS_INFO_MIVERSION	3	Целое число типа Integer: версия программы MapInfo Pro, в которой вы сейчас работаете, умноженная на 100 (сто).
SYS_INFO_RUNTIME	4	Логическое: Возвращает логическое "да" (TRUE) если работает runtime-версия MapInfo Pro, и логическое "нет" (FALSE) в противном случае.
SYS_INFO_CHARSET	5	Строка, величина типа String: Возвращает имя используемой системы кодов.
SYS_INFO_COPYPROTECTED	6	Логическое: "Да" (TRUE), если пользователь запустил защищенную от копирования версию MapInfo Pro.
SYS_INFO_APPLICATIONWND	7	IntPtr*: уникальный номер Windows <i>HWND</i> , который устанавливает оператор Set Application Window (или ноль, если <i>HWND</i> устанавливался).
SYS_INFO_DDESTATUS	8	Целое число типа Integer: количество элементов в DDE-очереди на выполнение. Если очередь пуста, то SystemInfo() вернет ноль (если приходящие команды будут выстраиваться в очередь) или -1 (если приходящие команды будут немедленно выполняться).
SYS_INFO_MAPINFOWND	9	IntPtr*: Целое число, представляющее номер Windows <i>HWND</i> главного окна MapInfo Pro . Функция возвращает 0, если программа выполняется не в Windows.
SYS_INFO_NUMBER_FORMAT	10	Строка, величина типа String: "9,999.9" или "Local," в зависимости от используемого форматирования чисел. См. раздел Оператор Set Format .

Код атрибута	ID	Возвращаемое значение SystemInfo()
SYS_INFO_DATE_FORMAT	11	Строка, величина типа String: "US" или "Local", в зависимости от используемого форматирования чисел. См. раздел Оператор Set Format .
SYS_INFO_DIG_INSTALLED	12	Логическое: "Да" (TRUE), если дигитайзер и совместимый драйвер установлен.
SYS_INFO_DIG_MODE	13	Логическое: "Да" (TRUE), если включен режим оцифровки.
SYS_INFO_MIPLATFORM	14	Целое число типа Integer: тип работающей программы MapInfo Pro. Возвращаемые значения MIPLATFORM_WIN32 (2) для 32-разрядной версии MapInfo Pro или MIPLATFORM_WIN64 (7) для 64-разрядной версии MapInfo Pro.
SYS_INFO_MDICLIENTWND	15	IntPtr*: Целое число, представляющее номер Windows <i>HWND</i> для окна MapInfo Pro MDICLIENT. Функция возвращает 0, если программа выполняется не в Windows.
SYS_INFO_PRODUCTLEVEL	16	Целое число типа Integer: указывающее на уровень запущенной версии продукта: 100 для MapInfo Desktop (продукт в отставке) или 200 для MapInfo Pro.
SYS_INFO_APPIDISPATCH	17	IntPtr*: Целое число, представляющее указатель на интерфейс IDispatch OLE Automation для приложения MapInfo. Пример использования: <pre>dim x as intptr x = Systeminfo(17) print x</pre>
SYS_INFO_MIBUILD_NUMBER	18	Целое число типа Integer: эта функция имеет атрибут, с помощью которого MapBasic может выяснить номер версии MapInfo Pro. Например, SystemInfo(SYS_INFO_MIVERSION) возвращает 850 для MapInfo Pro 8.5 и 8.5.2. Можно и дальше различать версии между 8.5 и 8.5.2 с помощью SystemInfo(SYS_INFO_MIBUILD_NUMBER), которая возвращает 32 для 8.5 и 60 для 8.5.2.

Код атрибута	ID	Возвращаемое значение SystemInfo()
SYS_INFO_MIFULLVERSION	19	Целое число типа Integer: целое число, представляющее полный номер выполняющейся версии MapInfo Pro. Это значение представляет строку с номером версии без десятичных точек. SystemInfo(SYS_INFO_MIFULLVERSION) возвращает 1103 для версии 11.0.3. Это более подробный разбор, чем использование SystemInfo(SYS_INFO_MIVERSION), которая возвращает 1100 для MapInfo Pro 11.0 и 11.0.3.
SYS_INFO_IMAPINFOAPPLICATION	20	Возвращает экземпляр приложения MapInfo Pro, представленный в .NET переменной. Этот тип переменных ссылается на экземпляр IMAPINFOPRO (MapInfo.Types.IMapInfoPro), который является основным интерфейсом доступа к компонентам MapInfo Pro из дополнительных программ. Он должен вызываться из запущенной MBX-программы, но только в 64-разрядной версии MapInfo Pro. Такой подход не работает ни в окне MapBasic, ни при использовании MapInfo Pro COM API.
SYS_INFO_MAPINFO_INTERFACE	21	Целое число типа Integer: возвращается род используемого интерфейса: классические, традиционные интерфейс и панели инструментов MIINTERFACE_CLASSICMENU (0) или ленточный интерфейс MIINTERFACE_RIBBON (1).
SYS_INFO_UNICODE	22	Логическое: TRUE, когда MapInfoPro (64-разрядная версия) выполняется в режиме Unicode, и FALSE, когда используется многобайтный набор символов.
SYS_INFO_CULTURE	23	Строка, величина типа String: возвращается двухбуквенный языковой код, который описывает, какая версия MapInfo Pro используется ("en" = английская, "fr" = французская, "de" = немецкая, "ja" = японская, "ru" = русская). Это значение приходит от установленной версии Pro (mires.dll). Корректные языковые коды: cs (чешский), da (датский), de (немецкий), en (английский), es (испанский), fi (финский), fr (французский), he (иврит), it (итальянский), ja (японский), nl (голландский), pl (польский), pt (португальский), ru (русский), sv (шведский), tr (турецкий), zh (китайский). Подробнее об этих кодах смотрите раздел <i>Коды языков</i> на сайте Microsoft по адресу: https://msdn.microsoft.com/en-us/library/ms533052(v=vs.85).aspx.

Код атрибута	ID	Возвращаемое значение SystemInfo()
SYS_INFO_LOCALE	24	Строка, величина типа String: возвращается строка локали, например, "en-US" или "ja-JP", в которой выполняется MapInfo Pro. Это значение берётся от операционной системы. Список строк, описывающих локали, смотрите раздел <i>Table of Language Culture Names, Codes, and ISO Values Method</i> на сайте Microsoft по адресу: https://msdn.microsoft.com/en-us/library/ee825488(v=cs.20).aspx .
SYS_INFO_MIRIBBONSTATE_VERSION	25	Целое число типа Integer: возвращается текущее состояние версии ленты MapInfo Pro.
SYS_INFO_LICENSE_MODE	26	Короткое целое число – SmallInt: Режим лицензирования, в котором работает MapInfo Pro. Ожидаемые значения: 0: Обыкновенный режим лицензирования FlexLM 1: Режим подписки.
SYS_INFO_SUBSCRIPTION_PLAN	27	Строка, величина типа String: Текущий план подписки, в котором работает MapInfo Pro. Ожидаемые значения: Режим подписки возвращает имя плана (например, Viewer). При работе в режиме лицензирования FlexLM возвращается пустая строка.

Зависящий от платформы тип, используемый для указателя или обработчика. Это помогает писать приложения, которые работают в обеих версиях MapInfo Pro — и 32-х и 64-разрядной. 64-разрядная версия MapInfo Pro рассматривает такие значения, как имеющие тип LargeInt, а 32-разрядная версия MapInfo Pro рассматривает как целое — Integer.

Условия появления сообщения об ошибке

ERR_FCN_ARG_RANGE (644) — сообщение об ошибке создаётся, если неправильно задано значение аргумента.

Пример

В следующем примере функция **SystemInfo** используется, для того чтобы определить, в какой операционной среде выполняется программа. Программа может вызвать DDE-процедуру, если программа использует какую-либо версию MapInfo Pro.

```
Declare Sub DDE_Setup

If SystemInfo(SYS_INFO_PLATFORM) = PLATFORM_WIN Then
  Call DDE_Setup
End If
```

См. также:

[предложение CharSet](#)

Функция TableConnectionInfo()

Назначение

Возвращает специальную информацию о соединении с базой данных открытой таблицы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TableConnectionInfo( tableAlias, connectionIndex, connectionInfoCode )
```

tableAlias – "синоним" названия открытой таблицы.

connectionIndex – индекс соединения с базой данных, о котором требуется получить информацию. Значение индекса может лежать в диапазоне от 1 до полного количества соединений. Смотрите раздел: [Функция TableInfo\(\)](#), в котором описывается способ определения количества соединений с базой данных открытой таблицы.

Если задан некорректный параметр *connectionIndex*, то при выполнении команды появится сообщение об ошибке.

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Короткое целое число – SmallInt.

Описание

Функция **TableConnectionInfo()** возвращает информацию о соединении с базой данных открытой таблицы. Потенциально таблица может иметь более одного соединения с базой данных. Эта команда может быть использована для получения информации о каждом имеющемся для этой таблицы соединении.

Значение *connectionIndex* представляет счётчик соединений открытой таблицы, о которой требуется получить информацию, с базой данных. Пересчитать имеющиеся соединения с базой данных можно с помощью вызова функции **TableInfo()** с атрибутом **TAB_INFO_CONNECTIONS_COUNT**. Если задан некорректный параметр *connectionIndex*, то при выполнении команды появится сообщение об ошибке.

Параметр *attribute* определяет вид возвращаемой функцией информации. Имена для режимных кодов определены в файле `MAPBASIC.DEF`.

Код connectionInfoCode	ID	TableConnectionInfo() возвращает
TABLE_CONNECTION_INFO_NUMBER	1	Короткий целочисленный результат, обозначающий номер соединения. Номер соединения можно использовать для получения дополнительной информации о соединении с помощью функции Server_ConnectInfo().

Примеры

В следующем примере возвращается номер соединения первого соединения с базой данных открытой таблицы.

```
TableConnectionInfo(tableAlias, 1, 1)
```

В следующем примере возвращается номер соединения второго соединения с базой данных открытой таблицы.

```
TableConnectionInfo(tableAlias, 2, 1)
```

См. также:

функция **TableInfo()**, функция **Server_ConnectInfo()**

Функция TableInfo()

Назначение

Возвращает информацию об открытой таблице. Функция может различать таблицы FME (внешних данных). Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TableInfo( table_id, attribute )
```

table_id — либо целочисленный номер таблицы, либо имя таблицы в кавычках, либо 0.

attribute — целочисленный код, управляющий результатом функции, см таблицу ниже. В следующем примере функция возвращает координатную систему и границы таблицы:

```
TableInfo(table_id, TAB_INFO_COORDSYS_CLAUSE)  
TableInfo(table_id, 29)
```

Возвращаемая величина

Величина типа String, Integer, SmallInt или Logical, в зависимости от значения параметра attribute.

Описание

Функция **TableInfo()** используется для получения определенной информации об открытой таблице. Первый параметр функции определяет, из какой таблицы была затребована информация.

Параметр *table_id* может быть строкой с именем открытой таблицы Или *table_id* числом, равным ее номеру. Если значение параметра *table_id* равно 0, то функция **TableInfo()** возвращает информацию о таблице, которая была открыта или создана самой последней. Можно использовать функцию с нулевым первым параметром, если непосредственно перед ней был выполнен **оператор Open Table**, в котором предложение **As** не использовалось. Если в момент вызова функции **TableInfo()** не было открыто ни одной таблицы или все таблицы уже закрыты, то результатом функции будет ошибка.

Параметр *attribute* определяет вид возвращаемой функцией информации. Коды в левом столбце (например, TAB_INFO_NAME) определены в файле стандартных определений MAPBASIC.DEF.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_NAME	1	Строка, имя таблицы.
TAB_INFO_NUM	2	Целое число (тип SmallInt), номер открытой таблицы.
TAB_INFO_TYPE	3	Целое число (тип SmallInt) – тип таблицы. Результатом может быть один из следующих кодов: • TAB_TYPE_BASE (1) если таблица обычная или сшитая. • TAB_TYPE_RESULT (2) если таблица представляет результат запроса. • TAB_TYPE_VIEW (3) если таблица является представлением (view), например, таблица STREETINFO является представлением. • TAB_TYPE_IMAGE (4) если таблица представляет растровое изображение. • TAB_TYPE_LINKED (5) если таблица связанная. • TAB_TYPE_WMS (6) если таблица предоставлена сервисом Web Map Service. • TAB_TYPE_WFS (7) если таблица предоставлена сервисом Web Feature Service. • TAB_TYPE_FME (8) если таблица открыта через FME. • TAB_TYPE_TILESERVER (9) если таблица представляет растровое изображение от сервера тайлов.
TAB_INFO_NCOLS	4	Целое число (тип SmallInt), количество колонок в таблице.
TAB_INFO_MAPPABLE	5	Логическая величина: TRUE, если записям таблицы можно сопоставлять графические объекты.
TAB_INFO_READONLY	6	Логическая величина. TRUE, если таблица открыта в режиме "только чтение".

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_TEMP	7	Логическая величина. TRUE, если таблица является временной (например, ЗАПРОС1).
TAB_INFO_NROWS	8	Целое число (тип Integer), количество строк в таблице.
TAB_INFO_EDITED	9	Логическая величина: TRUE, если таблица имеет несохраненные на диске изменения.
TAB_INFO_FASTEDIT	10	Логическая величина: TRUE, если для редактирования таблицы включен режим FastEdit , и FALSE, если выключен. (См. в разделе Оператор Set Table описание режима FastEdit.)
TAB_INFO_UNDO	11	Логическая величина; TRUE, если для редактирования таблицы включен механизм обратных действий (Undo), и FALSE, если механизм отключен с помощью оператора Set Table .
TAB_INFO_MAPPABLE_TABLE	12	Строка, представляющая базовую таблицу, то есть такую, которая содержит графические объекты. Если таблица представляет собой результат объединения, то возвращает имя той, которая содержит графические объекты.
TAB_INFO_USERMAP	13	Логическая величина. FALSE, если оператор Set Table установил режим UserMap в состояние выключено .
TAB_INFO_USERBROWSE	14	Логическая величина. FALSE, если оператор Set Table установил режим UserBrowse в состояние выключено .
TAB_INFO_USERCLOSE	15	Логическая величина. FALSE, если оператор Set Table установил режим UserClose в состояние выключено .
TAB_INFO_USEREDITABLE	16	Логическая величина. FALSE, если оператор Set Table установил режим UserEdit в состояние выключено .

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_USERREMOVEMAP	17	Логическая величина. FALSE, если оператор Set Table установил режим UserRemoveMap в состояние выключено .
TAB_INFO_USERDISPLAYMAP	18	Логическая величина. FALSE, если оператор Set Table установил режим UserDisplayMap в состояние выключено .
TAB_INFO_TABFILE	19	Строка с полным именем файла таблицы, включая DOS-маршрут. Пустая строка возвращается для таблицы запроса.
TAB_INFO_MINX, TAB_INFO_MINY, TAB_INFO_MAXX, TAB_INFO_MAXY	20 21 22 23	Вещественное число, минимальная и максимальная координаты по оси X и Y, координаты углов минимального прямоугольного покрытия всех объектов таблицы.
TAB_INFO_SEAMLESS	24	Логическая величина. TRUE, если для таблицы включен атрибут сшитости.
TAB_INFO_COORDSYS_MINX, TAB_INFO_COORDSYS_MINY, TAB_INFO_COORDSYS_MAXX, TAB_INFO_COORDSYS_MAXY	25 26 27 28	Вещественные величины, минимальная и максимальная координаты по оси X и Y, которые могут быть сохранены в таблице. Если таблица не может иметь объектов, результатом будет ноль.
TAB_INFO_COORDSYS_CLAUSE	29	Строка результата с предложением CoordSys таблицы, например, "CoordSys Earth Projection 1, 0". Возвращается пустая строка, если таблица не может содержать географические объекты.
TAB_INFO_COORDSYS_NAME	30	Строка с именем проекции, каким она названа в файле MAPINFOW.PRJ (но без суффикса "\p..."). Результатом будет пустая строка, если таблица не может иметь графические объекты или нет соответствия в файле MAPINFOW.PRJ.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_NREFS	31	Целое число (тип SmallInt), указывает количество таблиц, связанных с данной таблицей. Возвращает ноль для большинства таблиц. Другое значение возможно в случае, когда таблица определена как слияние двух других таблиц, например, таблица StreetInfo. Может использоваться только с базовыми таблицами (TAB_TYPE_BASE).
TAB_INFO_SUPPORT_MZ	32	Логическая величина. TRUE, если таблица поддерживает значения M и Z.
TAB_INFO_Z_UNIT_SET	133	Логическая величина. TRUE, если для Z-значений установлены единицы измерения.
TAB_INFO_Z_UNIT	34	Строка: указывающая единицы расстояния для Z-значений. Возвращает пустую строку, если единицы измерения не указаны.
TAB_INFO_BROWSER_LIST	35	Строка: указывающая, какие колонки будут отображаться в окне списка. Эта информация хранится в разделе метаданных таблицы. Если эта информация отсутствует возвращается пустая строка.
TAB_INFO_THEME_METADATA	36	Логическая величина. TRUE, если в метаданных есть описание тематической карты, используемой по умолчанию.
TAB_INFO_COORDSYS_CLAUSE_WITHOUT_BOUNDS	37	Строка (String), представляющая предложение CoordSys для таблицы за исключением границ. <pre>TableInfo (table_id, TAB_INFO_COORDSYS_CLAUSE_WITHOUT_BOUNDS)</pre> возвращает параметры CoordSys без границ или <pre>TableInfo (table_id, TAB_INFO_COORDSYS_CLAUSE)</pre> возвращает параметры CoordSys для таблицы с границами.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_DESCRIPTION	38	Строка: описание таблицы, которое может быть указано в TAB-файле. Если в TAB-файле нет описания, возвращается пустая строка.
TAB_INFO_TABLEID	39	Строка: уникальный идентификатор TAB-файла. Если у TAB-файла нет идентификатора, возвращается пустая строка.
TAB_INFO_PARENTTABLEID	40	Строка: уникальный идентификатор TAB-файла, с которого данная таблица была скопирована. Если таблица не является копией другого TAB-файла, возвращается пустая строка.
TAB_INFO_ISMANAGED	41	Логическая величина: TRUE, если таблица управляется сервисом библиотеки, FALSE, если нет.
TAB_INFO_ADSK_TEXTOBJECT	42	Логическая величина: TRUE, если это текстовая Autodesk-таблица.
TAB_INFO_OVERRIDE_COORDINATE_ORDER	43	Логическая величина: TRUE, если таблица является либо таблицей Web Feature Service (WFS), либо таблицей Web Map Service (WMS) с включенным режимом переключения порядка перечисления координат.
TAB_INFO_PERSIST	44	Логическая величина: TRUE – если таблица будет сохранена в рабочем наборе.
TAB_INFO_PREFER_HTML_FOR_INFO_TOOL	45	Логическая величина: TRUE, когда задан режим HTML.
TAB_INFO_INUSE	46	Логическая величина: TRUE, когда таблица используется, и FALSE, когда не используется.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_DATA_FORMAT	47	Целочисленный результат SmallInt: обозначает формат данных базовой таблицы (которая не является WMF, WFS, растром, FME или запросом). Возвращаемые значения: • TAB_DATA_FORMAT_NONE (0): таблица не базовая • TAB_DATA_FORMAT_DBASE (1): dBase DBF • TAB_DATA_FORMAT_LOTUS (2): Lotus 1-2-3 • TAB_DATA_FORMAT_ASCII (3): ASCII с разделителями • TAB_DATA_FORMAT_EXCEL (4): Microsoft Excel • TAB_DATA_FORMAT_NATIVE (5): MapInfo TAB • TAB_DATA_FORMAT_DAO (8): Microsoft Access • TAB_DATA_FORMAT_ODBC (10): ODBC • TAB_DATA_FORMAT_NATIVEX (14): Расширенный MapInfo Extended TAB • TAB_DATA_FORMAT_GPKG (15): Пакеты геоданных Geopackage • TAB_DATA_FORMAT_SHAPE (16): Набор шейп-файлов
TAB_INFO_CHARSET	48	Строковое значение – String: Возвращает имя используемой кодировки.
TAB_INFO_MAP_BLOCKSIZE	49	Целочисленный результат SmallInt: размер блока файла карты. Если таблица не имеет присоединяемых географических объектов, возвращается -1. (Если таблица без .map-файла, то возвращается -1.)
TAB_INFO_READVERSION	50	Целочисленный результат – Integer: самая ранняя версия, которая может прочитать таблицу.
TAB_INFO_EDITVERSION	51	Целочисленный результат – Integer: самая ранняя версия, которая может изменять таблицу.
TAB_INFO_READALLVERSION	52	Целочисленный результат – Integer: самая ранняя версия, которая может полностью прочитать таблицу.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_LOCALE	53	Строковое значение – String: возвращается строка локали, например, "en-US" или "ja-JP", в которой выполняется MapInfo Pro. Это значение берётся от операционной системы. Список строк, описывающих локали, смотрите раздел <i>Table of Language Culture Names, Codes, and ISO Values Method</i> на сайте Microsoft по адресу: https://msdn.microsoft.com/en-us/library/ee825488(v=cs.20).aspx.
TAB_INFO_CONNECTIONS_COUNT	54	Целочисленный результат SmallInt: Счётчик соединений с базой данных таблицы. • Если таблица родного, нативного формата (NATIVE), возвращается ноль (0). • Если таблица связана с открытой базой данных, возвращается единица (1). Но ноль (0) возвращается, если соединение с базой данных закрыто. • Если таблица напрямую соединена с базой данных, возвращается единица (1) (база данных прямого доступа всегда имеет открытое соединение). • Если таблица имеет объединения между двумя таблицами прямого доступа к разным базам данных, возвращается двойка (2). Но будет возвращена единица (1), если две таблицы созданы на основе единственной базы данных. Возвращённое значение можно использовать, когда выполняется функция TableConnectionInfo().

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_SCHEMAS_COUNT	55	Целочисленный результат SmallInt: Счётчик разных схем баз данных таблицы. • Если таблица родного, нативного формата (NATIVE), возвращается ноль (0). • Если таблица не соединена с базой данных, возвращается ноль (0). • Если таблица имеет соединение с базой данных, возвращается единица (1). Имеются следующие исключения, перечисленные ниже: • Если таблица создана в 'Экспертном режиме' на основе объединения на сервере с помощью метода 'SQL view', и объединение на сервере создано из двух таблиц по двум разным схемам, то будет возвращена двойка (2). • Если таблица является локальным объединением двух таблиц, соединённых с общей базой данных, а в обеих таблицах используется схема с одинаковым именем, то возвращается единица (1). Но возвращается двойка (2), если в таблицах используются схемы с разными именами. • Если таблица является объединением двух таблиц, соединённых с базами данными по разным соединениям, а в обеих таблицах используется схема с одинаковым именем, то возвращается двойка (2). Возвращается единица (1) только, если таблицы, для которых используются схемы с одинаковым именем, созданы на основе общей базы данных. Возвращённое значение можно использовать, когда выполняется функция TableSchemaInfo().
TAB_INFO_QUERY_STRING	56	Строковое значение – String: возвращается строка запроса таблицы-запроса. Возвращает только строку запроса для таблицы-запроса, созданной либо с помощью диалога SQL-запроса, либо запроса SQL в окне MapBasic в MapInfo Pro. Если таблица-запрос является некорректной, возвращается сообщение об ошибке "таблица не является допустимой таблицей запросов". Если таблица корректная, а SQL-оператор не доступен (например, при запросе, выполненном из MBX-файла), возвращается пустая строка.

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_ISHIDDEN	57	Логическая величина: возвращается TRUE, если таблица является скрытой. Скрытые таблицы не видны в MapInfo Pro, поэтому требуется знать синоним скрытой таблицы, для того чтобы выполнять операции с ней.
TAB_INFO_NUM_BASETABLES	58	Целое число типа Integer: возвращается количество базовых таблиц, которые составляют объединение, используемое в таблице-запросе. Например, <pre>Select * from USA, USA_CAPS where USA.State = USA_CAPS.State into MyQuery</pre> TableInfo(MyQuery, TAB_INFO_NUM_BASETABLES) — в результате будет возвращено 2.
TAB_INFO_NUM_BASETABLE_ALIASES	59	Строковое значение – String: возвращается список разделённых запятыми базовых таблиц, которые составляют объединение, используемое в таблице-запросе. Например, <pre>Open Table ApplicationDirectory\$() + "USA.tab" as States Open Table ApplicationDirectory\$() + "USA_CAPS.tab" as State_Caps Select * from States, State_Caps where States.State = State_Caps.State into MyQuery TableInfo(MyQuery, TAB_INFO_NUM_BASETABLE_ALIASES) would return "States,State_Caps" as a result.</pre>

Код атрибута	ID	TableInfo() возвращает
TAB_INFO_DOMINANT_OBJECT_TYPE	60	Возвращает наиболее распространённый тип объекта таблицы. Возвращаемые значения • TAB_UNKNOWN_OBJECTS (0) : используется, если тип объекта неизвестен или не принадлежит к объектам карты • TAB_POINT_OBJECTS (2) : в основном – точки • TAB_LINE_OBJECTS (3) : в основном – линии/полилинии • TAB_AREA_OBJECTS (4) : в основном – замкнутые объекты • TAB_MISC_OBJECTS (5) : в основном – иные (текст, ...) объекты • TAB_RASTER_OBJECTS (7) : растр, регулярная поверхность, wms, wmts или тайл-сервер

Условия появления сообщения об ошибке

ERR_TABLE_NOT_FOUND (405), если неправильно задана таблица или ее нет.

ERR_FCN_ARG_RANGE (644) — сообщение об ошибке создаётся, если неправильно задано значение аргумента.

Примеры

```
Include "mapbasic.def"
Dim i_numcols As SmallInt, L_mappable As Logical
Open Table "world"
i_numcols = TableInfo("world", TAB_INFO_NCOLS)
L_mappable = TableInfo("world", TAB_INFO_MAPPABLE)
TableInfo(table_id, TAB_INFO_COORDSYS_CLAUSE)
TableInfo(table_id, 29) - Returns the coordsys clause with bounds
```

Для того чтобы определить, что в таблице Web Feature Service (WFS) используется низменный порядок перечисления координат, можно воспользоваться атрибутом TAB_INFO_OVERRIDE_COORDINATE_ORDER (43). В этом случае результат TRUE будет возвращен, если параметр переключения порядка перечисления координат WFS-таблицы включен, а во всех остальных случаях будет возвращаться результат FALSE.

```
TableInfo(MyWFSTable, TAB_INFO_OVERRIDE_COORDINATE_ORDER)
```

См. также:

оператор Open Table, оператор Set Table, предложение CharSet

Функция TableListInfo()

Назначение

Возвращает информацию об окне со списком таблиц.

Синтаксис

```
TableListInfo( attribute )
```

attribute – целочисленный код, см. таблицу ниже.

Описание

Функция **TableListInfo()** возвращает информацию об окне Список таблиц.

Значение параметра *attribute* показаны в таблице ниже. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код атрибута	ID	TableInfo() возвращает
TL_INFO_SEL_COUNT	1	Короткое целое (Smallint result), количество выбранных объектов.

Примеры

```
TableListInfo(TL_INFO_SEL_COUNT)
```

В следующем примере функция используется в сочетании с командой быстрого меню.

```
include "mapbasic.def"
include "menu.def"
declare sub main
declare sub ShowTABPaths
'=====
sub main
' Add new item to Table List context menu
  alter menu ID M_SHORTCUT_TLV_TABLES add
    "Show TAB path..." calling ShowTABPaths
end sub
'=====
sub ShowTABPaths()
```

```

' Get the number of selected items
dim selCount as integer
selCount = TableListInfo(TL_INFO_SEL_COUNT)
' Print the table name and TAB file location fo all selected items
dim index as integer
for index = 1 to selCount
' Get the table id
dim tableId as integer
tableId = TableListSelectionInfo(index, TL_SEL_INFO_ID)
' Use the table in call to get the TAB path
dim tablePath as string
tablePath = TableInfo(tableId, TAB_INFO_TABFILE)
' Print the info
print tableName + " - " + tablePath
next
end sub

```

См. также:

[функция TableListSelectionInfo\(\)](#)

функция TableListSelectionInfo()

Назначение

Возвращает информацию о выбранном объекте в окне Список таблиц.

Синтаксис

```
TableListSelectionInfo( selection_index, attribute )
```

selection_id – номер выбранного в окне списка таблиц.

attribute – целочисленный код, см. таблицу ниже.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации. Имена для режимных кодов определены в файле MAPBASIC.DEF.

код атрибута	ID	TableInfo() возвращает
TL_SEL_INFO_NAME	1	Строка, представляющая имя выбранного объекта.

код атрибута	ID	TableInfo() возвращает
TL_SEL_INFO_ID	2	Короткое целое (Smallint), ID_номер таблицы, соответствующей выбранному объекту. Это значение может быть использовано для вызова функции TableInfo().

Пример

```
TableListSelectionInfo(index, TL_SEL_INFO_ID)
```

См. также:

функция [TableListInfo\(\)](#)

Функция TableSchemaInfo()

Назначение

Возвращает специальную информацию о схеме открытой таблицы. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TableSchemaInfo( tableAlias, schemaIndex, schemaInfoCode )
```

tableAlias – "синоним" названия открытой таблицы.

connectionIndex – индекс схемы, о котором требуется получить информацию. Значение индекса может лежать в диапазоне от 1 до полного количества схем. Смотрите раздел: [Функция TableInfo\(\)](#), в котором описывается способ определения количества схем открытой таблицы.

Если задан некорректный параметр *schemaIndex*, то при выполнении команды появится сообщение об ошибке.

attribute – целочисленный код, см. таблицу ниже.

Возвращаемая величина

Строка – String

Описание

Функция **TableSchemaInfo()** используется для получения определенной информации об открытой таблице. Первый параметр функции определяет, из какой таблицы была затребована информация. Потенциально таблица может иметь более одной схемы. Эта команда может быть использована для получения информации о каждой имеющейся для этой таблицы схеме.

Значение *schemaIndex* представляет счётчик схем открытой таблицы, о которой требуется получить информацию. Пересчитать имеющиеся схемы можно с помощью вызова **функции TableInfo()** с атрибутом TAB_INFO_SCHEMAS_COUNT. Если задан некорректный параметр *schemaIndex*, то при выполнении команды появится сообщение об ошибке.

Параметр *schemaInfoCode* может любым значением из таблицы ниже. Имена для кодов режимов определены в файле MAPBASIC.DEF.

Код connectionInfoCode	ID	TableConnectionInfo() возвращает
TABLE_SCHEMA_INFO_NAME	1	Строковый результат, обозначающий имя схемы.

Примеры

В следующем примере возвращается имя первой схемы таблицы.

```
TableSchemaInfo(tableAlias, 1, 1)
```

В следующем примере возвращается имя второй схемы таблицы.

```
TableSchemaInfo(tableAlias, 2, 1)
```

См. также:

Функция TableInfo()

Функция Tan()

Назначение

Вычисляет тангенс. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Tan ( num_expr )
```

num_expr – численное выражение угла в радианах.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Tan()** возвращает тангенс от числа, полученного в результате вычисления *num_expr*. Число *num_expr* является угловой величиной в радианах.

Чтобы преобразовать значение в градусах в радианы, умножьте его на DEG_2_RAD (0.01745329252). Чтобы преобразовать значение в радианах в градусы, умножьте его на RAD_2_DEG (57.29577951). Чтобы ваша программа могла использовать эти коэффициенты конвертирования, она должна содержать файл "MAPBASIC.DEF".

Пример

```
Include "mapbasic.def"

Dim x, y As Float

x = 45 * DEG_2_RAD
y = Tan(x)
```

у теперь будет равно 1, поскольку тангенс 45 градусов равен 1.

См. также:

[функция Acos\(\)](#), [функция Asin\(\)](#), [функция Atn\(\)](#), [функция Cos\(\)](#), [функция Sin\(\)](#)

Функция TempFileName()

Назначение

Возвращает имя, которое можно использовать при создании временного файла. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TempFileName$( dir )
```

dir – строка с маршрутом (диск+каталог), где будет создан временный файл; пустая строка ("") задает системный каталог для создания временных файлов.

Возвращаемая величина

Строка. Величина типа String.

Описание

Функция **TempFileName\$()** используется, когда необходимо создать временный файл, но неизвестно, какое имя ему дать.

Если вызвана функция **TempFileName\$()**, MapBasic возвращает строку с полным уникальным именем файла. Сама функция **TempFileName\$()** не создает временный файл. Создать файл можно, используя **оператор Open File**.

Если параметр *dir* задан пустой строкой (""), то полное имя файла будет включать в себя системный каталог для временных файлов, например, "G:\TEMP\~MAP0023.TMP".

При работе в сети возможна ситуация, когда два пользователя одновременно могут создать временный файл на одном и том же месте, с одинаковым именем. Если вы попытаетесь создать файл с именем и на каталоге, которые получили от функции **TempFileName\$()**, может возникнуть конфликтная ситуация, потому что файл уже существует, так как другой пользователь сети успел создать его после того, как программа вызвала функцию **TempFileName\$()**, и до оператора создания файла. Для снижения вероятности такой конфликтной ситуации выполняйте **оператор Open File** непосредственно сразу после вызова функции **TempFileName\$()**. Для снижения вероятности сетевых конфликтов создайте обработчик ошибок и поместите **оператор OnError** перед тем, как сработает **оператор Open File**.

См. также:

функция FileExists()

Оператор Terminate Application

Назначение

Закрывает другие выполняющиеся программы MapBasic или программы MapBasic, находящиеся в состоянии ожидания. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Terminate Application app_name
```

app_name – строковая величина с именем работающей программы (например, SCALEBAR.MBX)

Описание

Если прикладная программа создает свои меню или инструментальные панели, и сама не завершается, то она остается загруженной в режиме ожидания событий, которые надо обработать (пользователь выполняет команду, нажимает на кнопки, созданные прикладной программой и т. д.). Чтобы закрыть программу, находящуюся в режиме ожидания, в другой программе используется оператор **Terminate Application**. Например, если во время отладочного процесса отладочного процесса Вам необходимо закрыть программу, находящуюся в режиме ожидания, выполните оператор **Terminate Application** в окне **MapBasic**.

Если прикладная программа может запустить другую прикладную программу, используя [оператор Run Application](#), то она же может и завершить его оператором **Terminate Application**.

Примечание: **Terminate Application** используется для завершения только других прикладных программ. Для завершения самой программы используется [оператор End Program](#).

См. также:

[оператор End Program](#), [оператор Run Application](#)

Функция TextSize()

Назначение

Возвращает в пунктах размер текста в окне. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TextSize( window_id, text_obj )
```

window_id – целое, идентификатор, определяющий окно Карты или Отчета. Чтобы получить идентификатор окна, используется [функция FrontWindow\(\)](#) или [функция WindowID\(\)](#).

text_obj текстовый объект.

Примечание: Если текстовый объект выбран в окне карты, идентификатор окна должен быть ID-номером окна карты. Если текстовый объект выбран в окне Отчета, идентификатор окна должен быть ID-номером окна Отчета.

Возвращаемая величина

Вещественное – Float

Описание

Функция **TextSize()** возвращает значение текстового объекта в пунктах исходя из текущего масштаба окна. Эта функция похожа на выбор текстового объекта и вызов команды **Информация** или нажатие на клавишу **F7**.

Пример

Если активное окно это окно карты, а текстовый объект выбран:

```
print TextSize(FrontWindow(), selection.obj)
```

См. также:

[Предложение Font](#)

Функция ThemeExpressionInfo()

Назначение

Возвращает информацию об определённом выражении на тематическом слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ThemeExpressionInfo ( map_window_id, thematic_layer_id,  
theme_expression_id, attribute )
```

map_window_id — идентификатор окна карты;

thematic_layer_id — короткое целое – `SmallInt`, которое идентифицирует тематический слой карты.

theme_expression_id — целое – `Integer`, которое идентифицирует выражение.

attribute — короткое целое — `SmallInt`, которое идентифицирует тип возвращаемой информации; смотрите таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

Код атрибута	ID	ThemeExpressionInfo() возвращает
THEME_EXPRESSION_INFO_STRING	1	Строковый <code>String</code> , представляющий имя колонки или выражение, которое используется для создания тематического слоя.
THEME_EXPRESSION_INFO_TYPE	2	Короткое целое число (<code>Small</code>), представляющее тип выражения для обработки данных: EXPRESSION_TYPE_CHAR (1) EXPRESSION_TYPE_DECIMAL (2) EXPRESSION_TYPE_INTEGER (3) EXPRESSION_TYPE_SMALLINT (4) EXPRESSION_TYPE_DATE (5) EXPRESSION_TYPE_LOGICAL (6) EXPRESSION_TYPE_GRAPHIC (7) EXPRESSION_TYPE_FLOAT (8) EXPRESSION_TYPE_TIME (37) EXPRESSION_TYPE_DATETIME (38)

См. также:

[Функция ThemeInfo\(\)](#) на стр. 1106

[Функция ThemeItemInfo\(\)](#) на стр. 1107

Функция ThemeInfo()

Назначение

Возвращает информацию о слое в окне карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ThemeInfo ( map_window_id, thematic_layer_id, attribute )
```

map_window_id Идентификатор окна карты;

thematic_layer_id Короткое целое `SmallInt`, которое идентифицирует тематический слой карты.

attribute Код – короткое целое `SmallInt`, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	ThemeInfo() возвращает
THEME_INFO_TYPE	1	Короткий целочисленный (<code>SmallInt</code>) результат, представляющий тематический тип: THEME_TYPE_RANGES (0) THEME_TYPE_BARS (1) THEME_TYPE_PIES (2) THEME_TYPE_GRADUATED (3) THEME_TYPE_DOTDENSITY (4) THEME_TYPE_UNIQUE (5) THEME_TYPE_GRID (6)
THEME_INFO_ITEM_COUNT	2	Целочисленный результат, представляющий количество тематических элементов.

код атрибута	ID	ThemeInfo() возвращает
THEME_INFO_TABLE_ALIAS	3	Строка, представляющая синоним названия таблицы для таблицы, использованной при создании тематического слоя.
THEME_INFO_EXPRESSIONS_COUNT	4	Целочисленный результат представляющий количество тематических выражений или колонок.

См. также:

[Функция ThemeItemInfo\(\)](#) на стр. 1107

[Функция ThemeExpressionInfo\(\)](#) на стр. 1104

Функция ThemeItemInfo()

Назначение

Возвращает информацию об определённой тематической характеристике на тематическом слое карты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
ThemeItemInfo ( map_window_id, thematic_layer_id, theme_item_id, attribute )
```

map_window_id map_window_id – идентификатор окна карты;

thematic_layer_id Короткое целое SmallInt, которое идентифицирует тематический слой карты.

theme_item_id Целое, которое идентифицирует тематическую характеристику.

attribute Код – короткое целое SmallInt, см. таблицу ниже.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Параметр *attribute* определяет вид возвращаемой функцией информации.

код атрибута	ID	ThemeItemInfo() возвращает
THEME_ITEM_INFO_VISIBLE	1	Логический (<i>Logical</i>) результат, который обозначает является определённая тематическая характеристика видимой или нет.
THEME_ITEM_INFO_LABEL	2	Строковый (<i>String</i>) результат, представляющий подпись тематической характеристикеи.

См. также:

[Функция ThemeInfo\(\)](#) на стр. 1106

[Функция ThemeExpressionInfo\(\)](#) на стр. 1104

Функция Time()

Назначение

Возвращает текущее системное время в строковом формате. Время может быть в 12-и часовом или 24-часовом формате. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
StringVar = Time( format )
```

StringVar – строковая переменная, которой будет присвоено заданное системное время в формате hh:mm:ss *format*, где hh – час, mm – минуты и ss – секунды.

format – это целочисленное значение, определяющее формат возвращаемой строки. Значение 24 возвращает время в 24-часовом формате, а любое другое значение – 12-часовом.

Описание

Эквивалентом этой функции служит вызов [FormatTime\\$\(CurTime\(\)\)](#). Где результат, возвращаемый [Time\(12\)](#) является таким же, как время заданное функцией [FormatTime\\$\(CurTime\(\), "h:mm:ss"\)](#), а [Time\(24\)](#) – аналогично [FormatTime\\$\(CurTime\(\), "H:mm:ss"\)](#).

См. также:

[функции даты и времени](#), [функция FormatTime\\$\(\)](#), [функция GetTime\(\)](#)

Функция Timer()

Назначение

Возвращает число секунд. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Timer ( )
```

Возвращаемая величина

Целое число типа Integer.

Описание

Функция **Timer()** возвращает число секунд, прошедших с полуночи первого января 1970 года. Можно вызывать функцию **Timer()** до и после какой-либо операции, чтобы узнать, сколько времени в секундах она выполняется.

Пример:

```
Declare Sub Ubi  
  
Dim start, elapsed As Integer  
  
start = Timer()  
Call Ubi  
elapsed = Timer() - start
```

elapsed теперь содержит количество секунд, которое потребовалось, чтобы выполнить процедуру Ubi.

Процедура ToolHandler

Назначение

Специальная процедура для обеспечения работы со специальной инструментальной кнопкой (инструмент MapBasic).

Синтаксис

```
Declare Sub ToolHandler
Sub ToolHandler
    statement_list
End Sub
```

statement_list – список операторов, выполняющихся при выборе пользователем инструмента MapBasic.

Описание

ToolHandler – зарезервированное имя для процедуры MapBasic. Процедура работает совместно с инструментом MapBasic.

Задание процедуры **ToolHandler** является одним из простых способов создания своей кнопки на инструментальной панели Операции MapInfo Pro. Однако, способ представление кнопки, связанной с процедурой **ToolHandler** имеет некоторые ограничения. Вы не можете использовать произвольную картинку для кнопки или назначить свой режим рисования этой кнопке. Для создания кнопок, не имеющих этих ограничений, используются [оператор Alter ButtonPad](#) и [оператор Create ButtonPad](#).

Если пользователь запустил программу, которая имеет процедуру **ToolHandler**, то на инструментальной панели Операции появляется кнопка, на которой изображен знак плюс. Эта кнопка называется инструментом MapBasic. Инструмент может использоваться в активном окне списка, карты или отчета. Как только пользователь указал инструментом MapBasic в область окна Списка, Карты или Отчета, MapBasic автоматически запускает на выполнение процедуру **ToolHandler**.

В процедуре **ToolHandler** может быть использована [функция CommandInfo\(\)](#) для распознавания идентификатора окна, на которое было указано. Если инструмент был использован в окне списка функция может также дать информацию о строке и колонке, на которые указал инструмент. Если инструмент был использован в окне карты, функцию CommandInfo можно использовать для получения координат указанной точки в текущей системе координат (смотрите раздел: [Оператор Set CoordSys](#)).

Если инструмент был использован в окне отчета функция CommandInfo() поможет получить координаты точки на листе (т. е. расстояния от верхнего и левого края листа), в которой находился указатель мышки, когда была нажата кнопка мыши. (см. раздел: [Оператор Set Paper Units](#)).

Функция CommandInfo() также может определить, была ли нажата клавиша **Shift** или **Ctrl** или обе вместе. Это позволит определить по крайней мере четыре варианта действий для разных способов использования инструмента **Shift+щелчок**.

Выбрать инструмент на инструментальной панели может сама прикладная программа, выполнив оператор:

```
Run Menu Command M_TOOLS_MAPBASIC
```

Чтобы воспользоваться процедурой **ToolHandler**, пользователю необходимо запустить приложение. Когда пользователь запускает программу, в которой есть процедура **ToolHandler**, программа не завершается после выполнения всех операторов процедуры Main и других процедур, вызванных из нее.

Процедура Main может быть как объявлена, так и задана неявно. Программа будет пребывать в режиме ожидания, потому что процедура **ToolHandler** все еще находится в памяти, хотя и неактивна. Как только будет выбран инструмент MapBasic и применен в окне Списка, Карты или Отчета (достаточно щелчка мышки), автоматически начнется выполнение процедуры **ToolHandler**.

Когда процедура закончит свои действия (**оператор End Program**), программа полностью прекращает работу и освобождается занятая ею память. После использования **оператора End Program** процедура считается больше ненужной. Таким образом, можно использовать **оператор End Program**, чтобы удалить процедуру **ToolHandler**. И, наоборот, следите, чтобы **оператор End Program** не выполнялся, пока сохраняется нужда в процедуре **ToolHandler**.

В зависимости от обстоятельств, с процедурой **ToolHandler** используется **оператор Set CoordSys**, чтобы узнать координаты щелчка мыши. Если процедура **ToolHandler** вызвана в окне списка, то **оператор Set CoordSys** не нужен. Перед распознаванием координат точки в окне Отчета в процедуре **ToolHandler**, используется оператор **Set CoordSys Layout**. Если пользователь указал на точку карты, и координатная система карты не совпадает с координатной системой, принятой в прикладной программе (потому что был выполнен **оператор Set CoordSys**), то процедура **ToolHandler** должна выполнить **оператор Set CoordSys** перед тем, как определять координаты выбранной точки.

Пример

В следующей программе **ToolHandler** будет выдавать сообщения о координатах каждой выбранной точки в окнах Карт, Списков, Отчетов. В этом примере процедура **ToolHandler** просто показывает координаты щелчка мышки.

```
Include "mapbasic.def"
Declare Sub ToolHandler
Note "Ready to test the MapBasic tool."

Sub ToolHandler
  Note "x:" + Round(CommandInfo(CMD_INFO_X), 0.1) + Chr$(10) +
    " y:" + Round(CommandInfo(CMD_INFO_Y), 0.1)
End Sub
```

См. также:

функция CommandInfo()

Функция **TriggerControl()**

Назначение

Возвращает идентификатор элемента диалога, к которому пользователь обращался последним. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TriggerControl( )
```

Возвращаемая величина

Целое – Integer

Описание

Функция **TriggerControl()** используется только внутри процедуры-обработчика диалога, который создает **оператор Dialog** и возвращает номер элемента диалога, который был присвоен ему при создании диалога параметром из предложения ID.

Каждый элемент диалога может вызывать процедуру-обработчик. Это могут быть разные процедуры, а может быть так, что несколько элементов содержат вызов одного и того же обработчика. В последнем случае, с помощью функции **TriggerControl()**, имеет смысл определить, какой именно элемент диалога был выбран пользователем.

Ошибки

ERR_INVALID_TRIG_CONTROL (843), если функция **TriggerControl()** вызвана не при активном диалоге.

См. также:

оператор Alter Control, **оператор Dialog**, **оператор Dialog Preserve**, **оператор Dialog Remove**, **функция ReadControlValue()**

Функция TrueFileName()

Назначение

Возвращает полное имя файла на основе его неполной спецификации. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
TrueFileName$( file_spec )
```

file_spec – строчная величина с неполной спецификацией файла (например, в Windows "C:parcels.tab")

Описание

Функция TrueFileName\$() возвращает полную спецификацию файла

(полное имя диска, все каталоги или папки и имя файла), используя неполную. Например, в DOS следующий файл специфицирован частично (есть имя диска и имя файла, но нет каталогов, образующих маршрут):

```
"C:parcels.tab"
```

Если текущий каталог на диске C: является \MAPINFO\DATA, то функция

```
TrueFileName$( "C:parcels.tab" )
```

вернет следующую строку:

```
"C:\mapinfo\data\parcels.tab"
```

Функцию **TrueFileName\$()** удобно использовать в программе, которая подсказывает пользователю спецификацию для его файла на диске. Например, в диалоге перед сохранением файла.

Функция **TrueFileName\$()** не подтверждает наличие файла с таким именем, а только составляет полную спецификацию для файла, такую, какую бы он мог иметь. Для определения существования файла на диске используется [функция FileExists\(\)](#).

См. также:

[функция ProgramDirectory\(\)](#)

Оператор Type

Назначение

Создает произвольную структуру типа данных, которую позже используют **оператор Dim** и **оператор Global**.

Синтаксис

```
Type type_name Align[n]
  element_name As var_type
  [ ... ]
End Type
```

type_name – имя нового типа данных.

Align[n] – дополнительно определяет выравнивание с целевой DLL по структуре упаковки. Возможные значения: [1|2|4|8|16].

element_name – имя элемента типа.

var_type – тип данных для элемента.

Ограничения!

Все операторы **Type** должны находиться на "глобальном" уровне в тексте программы, например, за рамками sub-процедур. Нельзя использовать оператор **Type** в окне **MapBasic**. Нельзя использовать переменные сложного типа, созданного оператором **Type** как параметр процедуры или функции для пересылки значением ("by-value"). Также нельзя использовать переменные, созданные оператором **Type**, для записи значений в файл, когда используется **оператор Put**.

Описание

Оператор **Type** используется для создания нового типа данных, состоящего из элементов стандартных типов или других заранее определенных сложных типов. Для обращения к элементу переменной сложного типа надо использовать следующий формат: *variable_name.element_name*. Структура, созданная оператором **Type**, в качестве элемента может содержать другую сложную структуру. Элементом также может быть массив, состоящий из элементов как стандартного типа данных, так и элементов сложного типа данных. Нельзя присваивать значение одной переменной сложного типа, заданного оператором **Type**, другой переменной, созданной оператором **Type**, в форме: *var_name = var_name*.

Можно использовать разные типы значений для передачи в DLL. Дополнительное, необязательное слово *Align[n]* можно добавлять в оператор **Type**, который задаёт выравнивание

структуры упаковки с целевой DLL. Ключевое слово является аналогом опции */Zp[n]* компилятора MSVC, выравнивающей структуру членов, или специфического модификатора Microsoft *align(#)* для определения структуры – *struct*. Эта опция введена, чтобы решить несколько проблем выравнивания структур x64, особенно, со строками.

Если оператор **Type** задан с дополнительным, необязательным ключевым словом *Align[n]*, MapInfo Pro будет выравнивать члены по n-байтным границам, чтобы соответствовать спецификациям упаковки целевой DLL. Если *Align[n]* не задан, то MapInfo Pro будет использовать стандартное выравнивание **struct**, которое используется в качестве стандартного выравнивания, согласования в Visual Studio: 8 байт для 64-разрядных платформ и 4 байт – для 32-разрядных платформ. Не требуется строить специальные DLL с выравниванием по одному байту (*/Zp1*). Это нововведение лучше приспособлено к библиотекам DLL, созданных на x64 платформах, как через стандартные вызовы Windows API, при этом допуская специальную упаковку структур, требующуюся пользователям.

Проблемы, связанные с выравниванием структур, проявляются в передаче некорректных структурных членов данных при обмене вызовами с DLL. MapInfo Pro теперь предполагает стандартное значение выравнивания, которое совпадает со стандартным параметром, принятым в Visual Studio, поэтому, чтобы интегрировать MapBasic-программу с индивидуальной DLL, требуется либо:

- изменить декларацию структуры MapBasic, чтобы она совпадала со значением выравнивания к DLL (например *Align1*); либо
- перестроить DLL, используя стандартное выравнивание структуры по членам (*/Zp*).

Пример

```
Type Person Align8
  fullname As String
  age As Integer
  dateofbirth As Date
End Type

Dim sales_mgr, sales_people(10) As Person

sales_mgr.fullname = "Otto Carto"
sales_people(1).fullname = "Melinda Robertson"
```

См. также:

оператор Dim, оператор Global, оператор ReDim

Функция UBound()

Назначение

Возвращает размерность массива. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UBound( array )
```

array – имя массива переменных.

Возвращаемая величина

Целое – Integer

Описание

Функция **UBound()** возвращает текущую размерность массива переменных.

Каждая переменная массива имеет размерность от нуля и выше. Этот размер указывается **оператором Dim** или **оператором Global**. Размерность массивов может также изменяться по ходу программы **оператором ReDim**. Функция **UBound()** возвращает текущее значение размерности на момент вызова функции. Массивы могут иметь размерность до 32 767 элементов.

Пример

```
Dim matrix(10) As Float
Dim depth As Integer

depth = UBound(matrix)
' depth now has a value of 10

ReDim matrix(20)
depth = UBound(matrix)
' depth now has a value of 20
```

См. также:

оператор Dim, оператор Global, оператор ReDim

Функция UCase\$()

Назначение

Возвращает строку, в которой все буквы будут заглавными. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UCase$( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Строка

Описание

Функция **UCase\$()** возвращает строку, преобразуя все строчные буквы в прописные, в строке, заданной параметром *string_expr*.

Преобразованию подвергаются только буквы латинского и русского алфавитов. Цифры и другие символы остаются такими же, какими они были в строке *string_expr*. Например функция **UCase\$("A#12a")** возвращает строку "A#12A".

Пример

```
Dim regular, upper_case As String  
  
regular = "Los Angeles"  
upper_case = UCase$(regular)  
' upper_case now contains the value "LOS ANGELES"
```

См. также:

[функция LCase\\$\(\)](#), [функция Proper\\$\(\)](#)

Оператор UnDim

Назначение

Отменяет назначенную переменную или распускает массив. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UnDim variable_name
```

variable_name – имя переменной, которая была объявлена в рабочем наборе или окне **MapBasic**.

Предупреждение!

Оператор **UnDim** не может использоваться в компилированных программах MapBasic. Его использование разрешено только в Рабочем наборе или в окне **MapBasic**.

Описание

После того, как **оператор Dim** создаст переменную, можно использовать оператор **UnDim** для освобождения ресурсов, отведенных для этой переменной. Например, если был введен **оператор Dim**, объявляющий целочисленную переменную X, в окне **MapBasic**:

```
Dim X As Integer
```

Но необходимо изменить тип переменной. Следующие операторы переопределяют переменную X:

```
UnDim X  
Dim X As Float
```

См. также:

оператор Dim, оператор ReDim

Функция UndoInfo()

Назначение

Функция **UndoInfo()** возвращает информацию об окне, которое поддерживает стек отмен операций. В настоящее время это поддерживается только в окнах **Конструктор отчёта**. Возвращаемая информация включает количество элементов в стеке и название команды, которая будет выполняться при отмене или повторном выполнении операции. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UndoInfo ( window_id attribute )
```

window_id – идентификатор окна **Отчёт**, целое число;

attribute – целочисленный код типа информации, которая будет возвращена. Значения приведены в таблице ниже.

Возвращается

Смотрите таблицу ниже.

Описание

Функция **UndoInfo()** возвращает информацию об окне, которое поддерживает стек отмен операций. В настоящее время это поддерживается только в окнах **Конструктор отчёта**. Возвращаемая информация включает количество элементов в стеке и название команды, которая будет выполняться при отмене операции. Стек отмен – механизм записи одной или нескольких операций, которые будут отменены или повторно выполнены.

Функция **UndoInfo()** возвращает информацию об окне, которое поддерживает стек отмен операций. В настоящее время это поддерживается только в окнах конструктора отчёта.

Параметр *window_id* задает идентификатор окна карты. Для того чтобы получить идентификатор окна, используйте **функцию FrontWindow()**, которая вызывается сразу после открытия окна, а также **функцию WindowID()**, которую можно вызывать в любой момент после создания окна.

Функция **UndoInfo()** может возвращать различные значения параметров заданного окна. Параметр *attribute* должен являться одним из кодов, перечисленных в нижеследующей таблице. Эти коды задаются в файле `mapbasic.def`.

код атрибута	ID	UndoInfo() возвращает
UNDO_INFO_COUNT	1	Целое значение: возвращает количество элементов стека отмен.
UNDO_INFO_UNDO_ACTION_NAME	2	Строковое значение: возвращается название команды, которая будет выполняться при отмене операции.
UNDO_INFO_REDO_ACTION_NAME	3	Строковое значение: возвращается название команды, которая будет выполняться при повторе операции.
UNDO_INFO_CAN_UNDO	4	Логическое значение: возвращается "истина" (true), если операция не может быть отменена.
UNDO_INFO_CAN_REDO	5	Логическое значение: возвращается "истина" (true), если операция не может быть повторена.
UNDO_INFO_ENABLED	6	Логическое значение: возвращается "истина" (true), когда возможности стека включены (когда выполненные операции записываются в стеке отмен).
UNDO_INFO_CURRENT_POS	7	Целое значение: возвращает текущее положение в стеке отмен. Если ничего не может быть отменено, это значение совпадает с UNDO_INFO_COUNT.

См. также:

оператор Set Undo функция FrontWindow() функция WindowID()

Функция UnitAbbr()

Назначение

Возвращает сокращенное имя стандартной единицы измерения в MapInfo Pro. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UnitAbbr$( unit_name )
```

unit_name – строка, представляющая стандартное имя единицы измерения MapInfo Pro (например, "km").

Возвращаемая величина

Строка, представляющая сокращенное название единицы измерения (например, "km")

Описание

Параметр *unit_name* – это строка, представляющая единицу измерения в MapInfo Pro в англоязычном стандарте. Например, "km" (километр) или "sq km" (квадратный километр).

Функция **UnitAbbr\$()** возвращает сокращенный вариант имени единицы измерения. Возвращаемая строка зависит от языковой версии MapInfo Pro: там, где в английской и некоторых европейских версиях следующая функция возвратит "sq km", в русской будет возвращено "кв. км".

```
UnitAbbr$( "sq km" )
```

Список названий единиц измерения, которые могут быть использованы в качестве аргументов функции **UnitAbbr\$()**. Список стандартных единиц измерения расстояния, принятых в MapInfo Pro см. в разделе: [Оператор Set Distance Units](#). Список единиц измерения площади (например, "sq km") см. в разделе [Оператор Set Area Units](#). Список "бумажных" единиц измерения (например, "in") см. в разделе: [Оператор Set Paper Units](#). Единицы измерения времени включают: секунды ("sec"), минуты ("min"), и часы ("hr").

В качестве параметра *unit_name* можно также задать "degree" (функция **UnitAbbr\$()** возвратит "deg" или "град").

См. также:

[оператор Set Area Units](#), [оператор Set Distance Units](#), [оператор Set Paper Units](#), [функция UnitName\\$\(\)](#)

Функция UnitName()

Назначение

Возвращает полное имя стандартной в MapInfo Pro единицы измерения. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
UnitName$( unit_name )
```

unit_name – строковая величина с именем стандартной в MapInfo Pro единицы измерения (например, "km").

Возвращаемая величина

Строка, представляющая полное название единицы измерения (например, "kilometers").

Описание

Параметр *unit_name* – это строка, представляющая единицу измерения в MapInfo Pro в англоязычном стандарте. Например, "km" (километр) или "sq km" (квадратный километр).

Функция **UnitName\$()** возвращает строку с полной версией имени единицы измерения. Возвращаемая строка зависит от языковой версии MapInfo Pro: там, где в английской и некоторых европейских версиях следующая функция возвратит "square kilometers", в русской будет возвращено "квадратные километры".

```
UnitName$( "sq km" )
```

Список названий единиц измерения расстояний, площади, "бумажные" и времени, которыми могут быть использованы в качестве аргументов функции **UnitName\$()**. Список стандартных единиц измерения расстояния, принятых в MapInfo Pro см. в разделе: [Оператор Set Distance Units](#). Список единиц измерения площади (например, "sq km") см. в разделе [Оператор Set Area Units](#). Список "бумажных" единиц измерения (например, "in") см. в разделе: [Оператор Set Paper Units](#). Единицы измерения времени включают: секунды ("sec"), минуты ("min"), и часы ("hr").

В качестве параметра *unit_name* можно также задать "degree" (функция **UnitName\$()** возвратит "degrees" или "градусы").

См. также:

[оператор Set Area Units](#), [оператор Set Distance Units](#), [оператор Set Paper Units](#), функция [UnitAbbr\\$\(\)](#)

Оператор Unlink

Назначение

Разрыв связи с таблицей, которую загрузил с удаленной базой данных [оператор Server Link Table](#). Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Unlink TableName
```

TableName – имя открытой связанной таблицы MapInfo.

Описание

Отмена связи с таблицей удаляет ссылку к удаленной базе данных. Этот оператор не работает, если в таблице есть изменения (другими словами, пользователь должен сначала сохранить или отменить изменения в таблице). Все метаданные, описывающие инструкции связи, удаляются. отмеченные как запрещенные для записи, становятся доступными, В результате будет получена обыкновенная базовая таблица MapInfo.

Пример

```
Unlink "City_1k"
```

См. также:

[оператор Commit Table](#), [оператор Server Link Table](#)

Оператор Update

Назначение

Изменяет одну или более строк в таблице. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Update table Set column = expr [ , column = expr, ...]  
[ Where [ RowID = id ] | expr ]  
[ DropIndex ( Auto | On | Off ) ]
```

table – имя открытой таблицы.

column – имя колонки в таблице.

expr — одно или несколько выражений, разделенных словами AND или OR.

idnum – номер строки в таблице.

Описание

Оператор **Update** изменяет одну или более колонок в таблице. По умолчанию, оператор **Update** обновляет все строки таблицы. Если в операторе используется предложение **Where RowID**, обновляются только указанные строки. В предложении **Set** определяются сами изменения в полях заданной строки или строк.

Используя имя `Obj` в операторе **Set** для специальной колонки графических объектов, присоединенных к строкам таблицы, можно присоединять новые графические объекты к записям. Смотрите пример ниже.

Примечание: После выполнения оператора **Update** (или других операторов, которые изменяют таблицы) можно получать значения из таблиц, пользуясь выражением, содержащим ссылку на колонку (например, *tablename.columnname*). Однако, прежде чем ссылаться на любое выражение, необходимо выполнить оператор **Fetch**, чтобы задать запись, строку таблицы. Подробнее смотрите главу *Работа с таблицами* в *Руководстве пользователя MapBasic*.

Если задать в предложении **DropIndex** состояние **On**, то обновление индексов транзакций приостанавливается во время выполнения операции и восстанавливается по завершении операции. Иногда это значительно повышает производительность, но не всегда. Для получения наивысшей производительности мы рекомендуем устанавливать этот параметр в состояние **Auto**, чтобы программа MapInfo Pro выбирала следует ли приостанавливать или воссоздавать индексы. MapInfo Pro приостановит и воссоздаст индексы позже, если при выполнении операции удаляются или изменяются записи и таблица содержит более 1000 записей, из которых более 0.1% от общего количества записей обновляется. Когда в предложении **DropIndex** задано состояние **Off**, операция выполняется без приостановки и повторного создания индекса транзакций.

Оператор **Update** поддерживает условные выражения с помощью условия **WHERE**. Если оператор содержит условие **WHERE**, то будут обновлены только записи, которые удовлетворяют условию выражения.

```
Update PopAreas Set Area_sqm = Area(obj, "sq_m") Where Area_sqm = 0
```

Если в условном выражении **WHERE** требуется использовать идентификатор `RowId`, применяйте такую конструкцию:

```
"4 > RowId OR 2 < RowId" instead of "RowId > 4 OR RowId < 2"
```

Примеры

Мы имеем данные о служащих. Каждая запись содержит отдел, в котором работает служащий, и его жалование. Допустим, требуется поднять зарплату на семь процентов всем работникам маркетингового отдела, которые в настоящее время получают менее \$20,000. В примере, приведённом ниже, **оператор Select** используется, для того чтобы выбрать все записи,

касающиеся таких работников, а затем с помощью оператора **Update** изменяется соответствующая колонка с жалованьем.

```
Select * From employees
Where department ="marketing" And salary < 20000
Update Selection
Set salary = salary * 1.07
```

Используя предложение **Where RowID**, можно указать MapBasic применять операцию **Set** только к определенной строке таблицы. Теперь повысим жалование работника, данные которого находятся в десятой записи:

```
Update employees
Set salary = salary * 1.07
Where Rowid = 10
```

Используя предложение с условием **Where**, можно указать MapBasic применять операцию **Set** только к определенным строкам, записям таблицы. В следующем примере в таблице со списком сотрудников обновляется колонка с жалованьем работников, имеющих опыт работы более пяти лет:

```
Update employees
Set salary = salary * 1.07
Where EmpExp > 5
```

Создадим точечный объект и присоединим его к первой записи таблицы:

```
Update sites
Set Obj = CreatePoint(x, y)
Where Rowid = 1
```

См. также:

[оператор Insert](#), [оператор Fetch](#)

Оператор Update Window

Назначение

Форсирует обновление изображения в окне. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Update Window window_id
```

window_id – идентификатор окна.

Описание

Оператор **Update Window** обновляет изображение в одном из окон MapInfo Pro.

В некоторых ситуациях операции в окне не отображаются сразу и увидеть изменения можно только после ближайшего обновления окна. Например, если программа использует **оператор Dialog**, и из обработчика элементов диалога производятся изменения в окне карты, то новое изображение пользователь увидит только после закрытия диалогового окна. Заставить MapInfo Pro обновить изображение можно с помощью оператора **Update Window**.

См. также:

оператор Set Event Processing

Предложение URL

Назначение

Задаёт URL-адрес библиотечной службы. Это предложение можно использовать в **MapBasic** в среде MapInfo Pro.

Синтаксис

```
URL url
```

url – корректный URL-адрес библиотечного сервиса.

Описание

Предложение **URL** задаёт стандартный URL-адрес библиотечной службы. Оно проверяет, корректен ли такой адрес, и, если нет, выдаёт сообщение об ошибке.

Изначально стандартный URL-адрес библиотечной службы – пустая строка "", что означает, что библиотечная служба не задана. После того, как установлен корректный URL-адрес, его можно снова переустановить обратно в пустую строку.

Пример

```
Include "MAPBASIC.DEF"  
Set LibraryServiceInfo URL  
  
http://localhost:8080/LibraryService/LibraryService
```

См. также:

оператор `Set LibraryServiceInfo`, функция `LibraryServiceInfo( )`

Функция `USNGToPoint( )`

Назначение

Преобразует строку, представляющую описание координат USNG (United States National Grid) в точечный объект в текущей координатной системе MapBasic. Эту функцию можно вызывать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
USNGToPoint( string )
```

string – строка, представляющая USNG-описание.

Возвращаемая величина

Объект

Описание

Возвращает точки текущей координатной системе MapBasic, которая по умолчанию есть "Широта/Долгота" (без референц-эллипсоида). Чтобы добиться максимальной точности, задайте координатную систему MapBasic такой же, как и в исходной таблице до вызова **USNGToPoint()**. Иначе MapInfo Pro проведет ненужные преобразования координатных систем, что негативно отразится на точности.

Пример 1

```
dim obj1 as Object
dim s_USNG As String
dim obj2 as Object
obj1 = CreatePoint(-74.669, 43.263)
s_USNG = PointToUSNG$(obj1)
obj2 = USNGToPoint(s_USNG)
```

Пример 2

```

Open Table "C:\Temp\MyTable.TAB" as USNGfile
' When using the PointToUSNG$() or USNGToPoint() functions,
' it is very important to make sure that the current MapBasic
' coordsys matches the coordsys of the table where the
' point object is being stored.
'Set the MapBasic coordsys to that of the table used
Set CoordSys Table USNGfile
'Update a Character column (for example COL2) with USNG strings from
'a table of points
Update USNGfile
  Set Col2 = PointToUSNG$(obj)
'Update two float columns (Col3 & Col4) with
'CentroidX & CentroidY information
'from a character column (Col2) that contains USNG strings.
Update USNGfile
  Set Col3 = CentroidX(USNGToPoint(Col2))
Update USNGtestfile ' USNGfile
  Set Col4 = CentroidY(USNGToPoint(Col2))
Commit Table USNGfile
Close Table USNGfile

```

См. также:

[PointToUSNG\\$\(obj, datumid\)](#)

Функция Val()

Назначение

Возвращает численную величину, извлеченную из строки. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Val( string_expr )
```

string_expr – строковое выражение.

Возвращаемая величина

Вещественное – Float

Описание

Функция **Val()** возвращает число, выделяя его из строки, определенной выражением *string_expr*. При считывании числа функция **Val()** игнорирует пробелы, символы табуляции и новой строки в начале строки *string_expr*. Функция **Val()** прекращает считывание, когда находит символ, не являющийся частью числа. Если первый символ строки не является числом, одним из трех символов, описанных выше, точкой, знаком минус или плюс, амперсандом (&), то функция **Val()** вернет 0. Амперсанд используется для шестнадцатеричных чисел.

Примечание: Если строка включает разделитель целой части числа и десятичной, то этот знак должен быть точкой, независимо от того, какой стандарт форматирования чисел используется в компьютере пользователя. Строка также не должна содержать разделители тысяч. Для удаления разделителей тысяч используется [функция DeformatNumber\\$\(\)](#).

Пример

```
Dim f_num As Float
f_num = Val("12 thousand")
' f_num is now equal to 12

f_num = Val("12,345")
' f_num is now equal to 12

f_num = Val(" 52 - 62 Brunswick Ave")
' f_num is now equal to 52

f_num = Val("Eighteen")
' f_num is now equal to 0 (zero)

f_num = Val("&H1A")
' f_num is now equal to 26 (which equals hexadecimal 1A)
```

См. также:

[функция DeformatNumber\\$\(\)](#), [функция Format\\$\(\)](#), [оператор Set Format](#), [функция Str\\$\(\)](#)

Функция Weekday()

Назначение

Возвращает целое число от 1 до 7, соответствующее дню недели. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Weekday ( date_expr )
```

date_expr – выражение, результат которого есть дата (величина типа Data).

Возвращаемая величина

Короткое целое число от 1 до 7 включительно. Величина типа SmallInt.

Описание

Функция **Weekday** () возвращает номер дня в неделе. Число 1 соответствует воскресенью. Если, например, день, заданный выражением *date_expr*, является вторником, то результатом функции будет 3.

Функция **Weekday** () работает с датами, значения которых – не ранее первого января 100 года. Если выражение *date_expr* принимает значение даты ранее указанного значения функция **Weekday** () вернет ноль.

Пример

```
If Weekday( CurDate() ) = 6 Then
,
' then the date is a Friday
,
End If
```

См. также:

[функция CurDate\(\)](#), [функция Day\(\)](#), [функция Month\(\)](#), [функция Year\(\)](#)

Оператор WFS Refresh Table

Назначение

Обновляет таблицу WFS с сервера. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
WFS Refresh Table alias
[ Using Map [ Window window_id ] ]
[ Override Coordinate Order { On | Off } ]
```

alias – псевдоним, выбранный для зарегистрированной таблицы WFS.

window_id – целочисленный идентификатор окна Карты.

Описание

Если таблица была создана по фильтру строк, включающем все геометрические объекты WFS-таблицы, которые попадают в границы открытой карты (фильтр строк в этом случае будет `ogc:BBOX`, а значение передаваемого параметра `CURRENT_MAPPER`), то в таблице будут только те данные координаты которых ограничены пределами открытой карты. При обновлении таблицы будут использованы границы исходной таблицы, кроме случая, когда использовался дополнительный параметр **Using Map**. Тогда будут использоваться границы открытой карты и все операции изменения масштаба и перемещения по карте будут учтены.

Если идентификатор окна не задан, то будут использованы границы самой верхней карты. Если задан, то будут использованы границы заданной идентификатором *window_id* окна Карты.

Использование фильтра строк по границам открытой карты может ускорить показ WFS-таблицы, поскольку ограничивает объем передаваемых с сервера данных.

Параметр **Override Coordinate Order** используется, когда лишь в некоторых таблицах, полученных с сервера, используется неправильный порядок перечисления координат. Этот параметр переключает порядок перечисления координат отдельных таблиц (а не всех таблиц сервера). Этот параметр можно использовать вместе с параметром **Using Map**. Пример определения того, что включен параметр переключения порядка перечисления координат таблицы Web Feature Service (WFS) приведен в разделе о [функции TableInfo\(\)](#).

Пример

Следующий пример демонстрирует обновление локальной таблицы под названием `watershed`.

```
WFS Refresh Table watershed
```

Если WFS-таблица создана по фильтру строк, в котором используются границы открытой карты, а в этой карте после этого были выполнены какие-либо перемещения, то часть WFS-таблицы может быть не показана. Для того чтобы обновить таблицу и показать все объекты на открытой карте можно использовать команду следующего вида.

```
WFS Refresh Table watershed Using Map
```

См. также:

оператор **Register Table**, функция **TableInfo()**

Функция **WKTToCoordSysString()**

Назначение

Преобразует строки общеизвестных текстовых параметров (Well-Known Text – WKT) в параметры координатных систем MapBasic (CoordSys). Параметр **CoordSys** определяет систему координат, используемую на карте. Подробнее смотрите описание параметра **CoordSys**. Этот оператор можно выполнить в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
WKTToCoordSysString$( wkt_string )
```

wkt_string значение строки общеизвестного текстового параметра (Well-Known Text – WKT).

Возвращаемая величина

Строковое выражение, описывающее координатную систему. Если значение строки не обнаружено, возвращается пустая строка.

Пример

После выполнения следующего оператора:

```
print WKTToCoordSysString$( "GEOGCS[\"+\"\"NAD27  
Latitude/Longitude, Degrees\"\"+\", DATUM[  
\"+\"\"North American Datum 1927\"\"+\", SPHEROID[  
\"+\"\"Clarke - 1866\"\"+\", 6378206.4, 294.9786982139006],  
AUTHORITY[\"+\"\"EPSG\"\"+\", \"+\"\"6267\"\"+\"],  
PRIMEM[\"+\"\"Greenwich\"+\"\", 0], UNIT[\"+\"\"degree\"\"+\", 0.0174532925199433]]\" )
```

Будет возвращена следующая строка:

```
CoordSys Earth Projection 1, 62
```

См. также:

предложение CoordSys, **функция CoordSysStringToWKT\$()**, **оператор Set CoordSys**

Оператор While...Wend

Назначение

Циклически выполняет определенные действия, пока истинно определенное условие.

Синтаксис

```
While condition  
    statement_list  
Wend
```

condition – выражение, управляющее выполнением цикла.

statement_list – группа операторов, выполняющаяся за один проход цикла.

Предупреждение!

Оператор **While...Wend** нельзя выполнить в окне **MapBasic**.

Описание

Оператор **While...Wend** является одной из конструкций цикла. MapBasic проверяет условие, заданное выражением *condition*; если условие истинно, MapBasic выполняет операторы *statement_list*.

Далее снова проверяется условие *condition*, и, если оно истинно, повторяется *statement_list*. Цикл выполняется до тех пор, пока значение *condition* не станет ложным. После этого MapBasic пропустит операторы *statement_list* и передаст выполнение программы следующему после **Wend** оператору.

Заметим, что конструкция:

```
While condition  
    statement_list  
Wend
```

фактически идентична конструкции:

```
Do While condition  
    statement_list  
Loop
```

Синтаксис **While...Wend** предлагается для любителей (некоторые программисты предпочитают синтаксис **While...Wend** синтаксису, который использует [оператор Do...Loop](#)).

Пример

```

Dim psum As Float, i As Integer
Open Table "world"
Fetch First From world
i = 1
While i <= 10
    psum = psum + world.population
    Fetch Next From world
    i = i + 1
Wend

```

См. также:

оператор Do...Loop, оператор For...Next

Процедура WinChangedHandler

Назначение

Процедура, автоматически выполняющаяся при перемещении или увеличении/уменьшении изображения в окне карты, а также при добавлении или удалении слоя.

Синтаксис

```

Declare Sub WinChangedHandler
Sub WinChangedHandler
    statement_list
End Sub

```

statement_list – список операторов процедуры.

Описание

WinChangedHandler – специальное имя Main-процедуры MapBasic. Когда пользователь запускает программу, имеющую процедуру WinChangedHandler, программа не завершается после выполнения всех операторов процедуры Main и других процедур, вызванных из нее. MapBasic обратится к WinChangedHandler, когда произойдет перемещение и масштабирование карты в окне или изменение размеров самого окна карты. В процедуре WinChangedHandler может быть использована **функция CommandInfo()** для распознавания идентификатора окна, в котором произошло изменение.

Одновременно в состоянии ожидания могут находиться несколько программ. Поэтому при изменении в окне карты, автоматически выполняются все процедуры **WinChangedHandler** из этих программ, одна за другой.

В некоторых случаях MapBasic вызывает процедуру **WinChangedHandler** на события, не связанные с изменениями размера окна. Например, рисование нового объекта вызывает процедуру **WinChangedHandler**. Чтобы закрыть обработчик и удалить его из памяти применяйте оператор **End Program**.

Автопрокрутка в окне карты

MapInfo Pro версии 4.0 автоматически сдвигает изображение в окне карты, когда пользователь, используя, например, инструмент рисования прямоугольника, растягивая контур будущего объекта, подводит указатель мышки к краю окна. Если действия пользователя привели к автоматическому сдвигу изображения окна, MapInfo вызывает процедуру **WinChangedHandler** после выполнения или отмены действий инструмента.

Например, если был использован инструмент Линейка и рисование каждого сегмента приводит к автопрокрутке, то MapInfo Pro вызовет **WinChangedHandler** тогда, когда закончите измерения двойным щелчком мышки или нажатием клавиши **ESC**. Если автопрокрутка произошла вследствие применения пользователем инструмента MapBasic, то MapInfo Pro сначала вызовет обработчик инструмента, а затем процедуру **WinChangedHandler**.

MapInfo Pro не будет вызывать процедуру **WinChangedHandler**, если пользователь вернет в предыдущее состояние изображение окна, в котором оно находилось перед автопрокруткой, или нажмет на клавишу **ESC**.

Автоматический сдвиг изображения карты можно отключить, используя оператор **Set Window**.

Пример

Пример использования процедуры **WinChangedHandler** можно найти в тексте программы OVERVIEW, которая входит в стандартную поставку MapBasic.

См. также:

функция **CommandInfo()** , процедура **WinClosedHandler**

Процедура WinClosedHandler

Назначение

Процедура автоматически выполняется при закрытии окна **Карта**, **Список**, **График**, **Отчёт**, **Районирование**, **Легенда** или **MapBasic**.

Синтаксис

```
Declare Sub WinClosedHandler
Sub WinClosedHandler
    statement_list
End Sub
```

statement_list – список операторов процедуры.

Описание

WinClosedHandler – зарезервированное имя для процедуры MapBasic. Когда пользователь запускает программу, имеющую процедуру **WinClosedHandler**, программа не завершается после выполнения всех операторов процедуры Main и других процедур, вызванных из нее. Как только пользователь закроет какое-нибудь окно, MapBasic выполняет процедуру с именем **WinClosedHandler**.

В процедуре **WinClosedHandler** может быть использована функция:

```
CommandInfo( CMD_INFO_WIN )
```

для распознавания идентификатора окна, в котором произошло изменение.

Примечание: Когда процедура закончит свои действия (**оператор End Program**), программа полностью прекращает работу и освобождается занятая ею память. После использования **оператора End Program** процедура **WinClosedHandler** считается больше ненужной. Поэтому, если снова понадобятся функции, выполняемые инструментом MapBasic, не используйте **оператор End Program**, пока процедура **WinClosedHandler** еще необходима.

Одновременно в состоянии ожидания могут находиться несколько программ. Поэтому при изменении в окне карты, автоматически выполняются все процедуры **WinClosedHandler** из этих программ, одна за другой.

См. также:

[функция CommandInfo\(\)](#), [процедура EndHandler](#), [процедура RemoteMsgHandler](#), [процедура SelChangedHandler](#), [процедура ToolHandler](#), [процедура WinChangedHandler](#)

Функция WindowID()

Назначение

Возвращает идентификатор окна на экране. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
WindowID ( window_num )
```

window_num – величина типа SmallInt, номер окна.

Возвращаемая величина

Целое – Integer

Описание

Функция WindowID() возвращает уникальный номер окна. Некоторые операторы MapBasic, такие как **оператор Set Map**), используют идентификатор в качестве параметра.

В следующей таблице приводятся возможные способы задания параметра *window_num*:

Значение window_num	Результат
Положительное целое число (величина типа SmallInt) (1, 2, ... <i>n</i>)	MapInfo Pro возвращает идентификатор документального окна, такого, как карта, список. Например, если задана единица, то MapInfo Pro возвращает идентификатор первого документального окна. Заметим, что значение <i>n</i> — это число открытых документов; функция NumWindows() возвращает значение <i>n</i> .
Отрицательное целое число (величина типа SmallInt (-1,-2, ...- <i>m</i>))	MapInfo Pro возвращает идентификатор как документального окна, так и плавающего окна, например, Информации. Обратите внимание, что <i>m</i> это общее количество окон, принадлежащих MapInfo Pro; вызывайте функцию NumAllWindows() , для того чтобы определить <i>m</i> . С помощью такого синтаксиса можно вызывать WindowID() в цикле, для того чтобы построить список идентификаторов всех открытых окон.
Ноль (0)	MapInfo Pro возвращает идентификатор окна либо последнего из открытых документов, либо легенды, созданной пользователем, либо инструментальной панели; либо ноль, если окна не открывались.
Код окна (например, WIN_RULER)	Если задан код от 1001 до 1013, то MapInfo Pro вернет идентификатор соответствующего специального окна. Коды окон определены в MAPBASIC.DEF. Например, содержит код WIN_RULER (значение которого 1007), которое используется инструментом Линейка MapInfo Pro.

Далее приведён список кодов окон, которые возвращает эта функция. Идентификаторы окон **Список таблиц** (WIN_TABLE_LIST, 2001), **Управление слоями** (WIN_LAYER_CONTROL, 2002) и **Переместить** (WIN_MOVE_MAP_TO 2003) игнорируются.

Тип окна	ID	Описание окна
WIN_MAPPER	1	Окно Карта
WIN_BROWSER	2	Окно Списка
WIN_LAYOUT	3	Окно Отчет
WIN_GRAPH	4	Окно График
WIN_BUTTONPAD	19	Окно панели с кнопками
WIN_TOOLBAR	25	Окно инструментальной панели
WIN_CART_LEGEND	27	Окно картографической Легенды
WIN_3DMAP	28	Окно 3D-карты
WIN_ADORNMENT	32	Элемент оформления
WIN_LEGEND_DESIGNER	35	Окно Легенда
WIN_LAYOUT_DESIGNER	36	Окно Отчёт
WIN_HELP	1001	Окно Справочной системы
WIN_MAPBASIC	1002	Окно MapBasic
WIN_MESSAGE	1003	Окно Сообщение (его открывает Оператор Print)
WIN_RULER	1007	Окно Линейка (вызывается инструментом Линейка)
WIN_INFO	1008	Окно Информация (вызывается инструментом Информация)
WIN_LEGEND	1009	Окно Тематическая легенда

Тип окна	ID	Описание окна
WIN_STATISTICS	1010	Окно Статистика
WIN_MAPINFO	1011	Рабочее окно MapInfo Pro
WIN_WORKSPACE_EXPLORER	2004	Окно Проводник рабочих наборов
WIN_WINDOW_LIST	2005	Окно Список окон
WIN_TOOL_MANAGER	2006	Окно Менеджер программ
WIN_TASK_MANAGER	2007	Окно Диспетчер задач
WIN_CONNECTION_LIST	2008	Окно Список соединений
WIN_NOTIFICATIONS	2009	Предупреждения используются MapInfo Pro или дополнительными программами, для того чтобы сообщить о необходимости обратить внимание на что-то. Предупреждения появляются в окне сообщений.
WIN_QUERY_BUILDER	2010	Улучшенный диалог SQL-запрос

Условия появления сообщения об ошибке

ERR_BAD_WINDOW_NUM (648), если неверно значение аргумента *window_num*.

См. также:

функция [WindowInfo\(\)](#), функция [FrontWindow\(\)](#), функция [NumWindows\(\)](#)

Функция WindowInfo()

Назначение

Возвращает информацию об открытом окне. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
WindowInfo( window_spec, attribute )
```

Window_spec — целочисленный идентификатор окна.

attribute — целое число, код возвращаемой информации.

Возвращаемая величина

Тип результата зависит от значения параметра *attribute*.

Описание

Функция **WindowInfo**() используется для получения определенной информации об открытом окне.

Многие значения, передаваемые в качестве параметров **WindowInfo**(), определены в стандартном файле MapBasic-определителей MAPBASIC.DEF. В программе следует указать "MAPBASIC.DEF", если требуется обращаться к **WindowInfo**().

В следующей таблице приводятся возможные способы задания параметра *window_num*:

Значение <i>window_spec</i>	Описание
Целочисленный идентификатор	Для задания окна, о котором требуется получить информацию, можно использовать его идентификатор, вызвав функцию WindowID () или функцию FrontWindow ().
Положительное целое число (величина типа SmallInt) (1, 2, ... <i>n</i>)	Функция возвращает информацию об окне документа, таком как карта или список. Например, если задана единица, то MapInfo Pro возвращает информацию о первом окне документа. Заметим, что значение <i>n</i> — это число открытых документов; функция NumWindows () возвращает значение <i>n</i> .
Отрицательное целое число (величина типа SmallInt) (-1,-2, ...- <i>m</i>)	Функция возвращает информацию как об окне документа, так и о другом плавающем окне, таком, как Информация. Обратите внимание, что <i>m</i> это общее количество окон принадлежащих MapInfo Pro; вызывайте NumAllWindows () функцию, для того чтобы определить <i>m</i> . С помощью такого синтаксиса можно вызывать WindowInfo () в цикле, для того чтобы опросить каждое открытое окно.
Ноль (0)	Функция опрашивает последнее из открывавшихся окон. Если окна не открыты, возникает ошибка.

Значение window_spec	Описание
Код окна (например, WIN_RULER)	Если задан код от 1001 до 1013, то MapInfo вернет информацию о соответствующем специальном окне. Коды окон определены в MAPBASIC.DEF. Например, MAPBASIC.DEF содержит код WIN_RULER (значение которого 1007), которое используется инструментом Линейка MapInfo Pro.

Теперь рассмотрим зависимость результата от значения параметра attribute. Параметр attribute должен являться одним из кодов, показанных в нижеследующей таблице:

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_NAME	1	Строковое значение – String: с именем окна.
WIN_INFO_TYPE	3	Короткое целое (SmallInt): определяющее тип окна, например, WIN_LAYOUT (3). Смотрите таблицу ниже.
WIN_INFO_WIDTH	4	Вещественное значение – Float: ширина окна в "бумажных" единицах измерения, установленных оператором Set Paper Units. Для 64-разрядной версии MapInfo Pro используйте атрибут WIN_INFO_CLIENTWIDTH. "Бумажные" единицы измерения описаны в разделе: Оператор Set Paper Units .
WIN_INFO_HEIGHT	5	Вещественное значение – Float: ширина окна в "бумажных" единицах измерения, установленных оператором Set Paper Units. Для 64-разрядной версии MapInfo Pro используйте атрибут WIN_INFO_CLIENTHEIGHT.
WIN_INFO_X	6	Вещественное значение – Float: расстояние от верхнего края нашего окна до верхнего края рабочего поля MapInfo (в "бумажных" единицах). Если это окно является встроенной в окно конструктора отчёта рамкой, то значение представляет собой расстояние от левого края макета в окне Отчёт . 64-разрядная версия MapInfo Pro возвращает последнее зафиксированное положение левого края плавающего окна относительно левого края основного окна-экрана.

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_Y	7	Вещественное значение – Float: расстояние от верхнего края нашего окна до верхнего края рабочего поля MapInfo (в "бумажных" единицах). Если это окно является встроенной в окно Отчёт рамкой, то значение представляет собой расстояние от верхнего края макета в окне Отчёт . 64-разрядная версия MapInfo Pro возвращает последнее зафиксированное положение верхнего края плавающего окна относительно верхнего края основного окна-экрана.
WIN_INFO_TOPMOST	8	Логическая величина: Если окно активно, значение будет истинно (TRUE).
WIN_INFO_STATE	9	Короткое целое (SmallInt): Эти три индикатора описывают состояние плавающих окон в 64-разрядной версии MapInfo Pro. • WIN_STATE_NORMAL, если нормального размера • WIN_STATE_MINIMIZED (1), если минимизировано • WIN_STATE_MAXIMIZED (2), если максимизировано Это новые состояния, возвращаемые windowinfo: • WIN_STATE_DOCKED (3), если закреплено • WIN_STATE_TABBED (4), если на вкладке • WIN_STATE_AUTOHIDDEN (5), если автоматически скрывается • WIN_STATE_HIDDEN (6), если скрыто
WIN_INFO_TABLE	10	Строковое значение – String: "CosmeticN", если определено окно карты; "LayoutN", если определено классическое окно отчёта. Для окна конструктора отчёта возвращается пустая строка. Для окон списка или графика — имя открытой в этом окне таблицы.
WIN_INFO_ADORNMENTS_MAP	10	Действует то же значение, что для WIN_INFO_TABLE (10). Целое число типа Integer: если WindowID — элемент оформления, то будет возвращен номер окна карты, в котором создан этот элемент оформления.
WIN_INFO_LEGENDS_MAP	10	Действует то же значение, что для WIN_INFO_TABLE (10). Целое число типа Integer: Если составляется запрос об окне легенды, которое создал оператор Create Legend, то результатом будет идентификатор соответствующего окна карты или графика. Если окно легенды стандартно, результатом будет 0.

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_OPEN	11	Логическая величина: определяющая, открыто ли окно (используется для таких окон как Информация).
WIN_INFO_WND	12	IntPtr* значение. В среде Windows представляет Windows <i>HWND</i> для опрашиваемого окна.
WIN_INFO_WINDOWID	13	Целое число, идентификатор окна. Результат такой же, который возвращает функция WindowID() . Это можно использовать для передачи нуля как <i>window_spec</i> .
WIN_INFO_WORKSPACE	14	Строковое значение — String: строка с операторами MapBasic, с помощью которых, благодаря команде Сохранить рабочий набор , запоминается карта в рабочем наборе. Отличается от результата функции с кодом WIN_INFO_CLONEWINDOW (15) тем, что результат включает в себя Оператор Open Table и другие.
WIN_INFO_CLONEWINDOW	15	Строковое значение — String: строка с операторами MapBasic, которая может использоваться оператором Run Command для дублирования окна.
WIN_INFO_SYSMENUCLOSE	16	Логическая величина: FALSE, если оператор Set Window нейтрализовал команду Close из системного меню окна.
WIN_INFO_AUTOSCROLL	17	Логическая величина: TRUE, если режим автоматического сдвига при приближении указателя мышки к краю экрана карты или отчёта включен. Изменить этот режим может Оператор Set Window .
WIN_INFO_SMARTPAN	18	Логическая величина. TRUE, если режим Smart Pan включен.
WIN_INFO_SNAPMODE	19	Возвращает логическую величину. TRUE — совмещение узлов включено. FALSE — если режим совмещения узлов отключен.
WIN_INFO_SNAPTHRESHOLD	20	Возвращает короткое целое — SmallInt, обозначающее допуск в пикселях.
WIN_INFO_PRINTER_NAME	21	Возвращает строку с идентификатором принтера (например, \\DISCOVERYHP4_DEVEL)
WIN_INFO_PRINTER_ORIENT	22	Возвращает WIN_PRINTER_PORTRAIT (1) или WIN_PRINTER_LANDSCAPE (2)

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_PRINTER_COPIES	23	Возвращает целое число копий.
WIN_INFO_PRINTER_PAPERSIZE	24	Целое число. Смотрите в файле PAPERSIZE.DEF (в директории \MapInfo\MapBasic) объяснения значений возвращаемых величин.
WIN_INFO_PRINTER_LEFTMARGIN	25	Вещественное значение – Float: левое поле области печати в текущих единицах длины.
WIN_INFO_PRINTER_RIGHTMARGIN	26	Вещественное значение – Float: правое поле области печати в текущих единицах длины.
WIN_INFO_PRINTER_TOPMARGIN	27	Вещественное значение – Float: верхнее поле области печати в текущих единицах длины.
WIN_INFO_PRINTER_BOTTOMMARGIN	28	Вещественное значение – Float: нижнее поле области печати в текущих единицах длины.
WIN_INFO_PRINTER_BORDER (29)	29	Строковое значение – String: ON, если рамка вокруг окна изображается при печати, OFF если нет.
WIN_INFO_PRINTER_TRUECOLOR	30	Строковое значение – String: ON, если используется 24-разрядный true color для печати растров и регулярных поверхностей (grid). Это бывает, когда изображение 24-разрядное и принтер поддерживает более 256 цветов, OFF в противном случае.
WIN_INFO_PRINTER_DITHER	31	Строковое значение – String: возвращает метод полутоновой растеризации, который используется если необходимо конвертировать 24-разрядное изображение в 256 цветов. Возвращаемые значения HALFTONE и ERRORDIFFUSION. Используется при печати растров и поверхностей. Растеризация произойдет, если WIN_INFO_PRINTER_TRUECOLOR (30) отключён или если принтер поддерживает только 256 цветов или менее.
WIN_INFO_PRINTER_METHOD	32	Строковое значение – String: возвращает значения DEVICE, EMF или PRINTOSBM.
WIN_INFO_PRINTER_TRANSPRSTER	33	Строковое значение – String: возвращает значения DEVICE и INTERNAL.

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_PRINTER_TRANSPVECTOR	34	Строковое значение – String: возвращает значения DEVICE и INTERNAL.
WIN_INFO_EXPORT_BORDER	35	Строковое значение – String: возвращает значения ON и OFF.
WIN_INFO_EXPORT_TRUECOLOR	36	Строковое значение – String: возвращает значения ON и OFF.
WIN_INFO_EXPORT_DITHER	37	Строковое значение – String: возвращает значения HALFTONE и ERRORDIFFUSION.
WIN_INFO_EXPORT_TRANSPRASTER	38	Строковое значение – String: возвращает значения DEVICE и ROP.
WIN_INFO_EXPORT_TRANSPVECTOR	39	Строковое значение – String: возвращает значения DEVICE и INTERNAL.
WIN_INFO_PRINTER_SCALE_PATTERNS	40	Логическая величина. TRUE – если окно масштабировано для печати. FALSE – если не масштабировано.
WIN_INFO_EXPORT_ANTIALIASING	41	Строковое значение – String: ON, если фильтр сглаживания будет использоваться при экспорте и OFF в противном случае.
WIN_INFO_EXPORT_THRESHOLD	42	Целое значение от 0 до 255, определяющее порог сглаживания.
WIN_INFO_EXPORT_MASKSIZE	43	Целое значение между 0 и 100
WIN_INFO_EXPORT_FILTER	44	Целое значение, возвращающее один из доступных фильтров сглаживания: • FILTER_VERTICALLY_AND_HORIZONTALLY (0) • FILTER_ALL_DIRECTIONS_1 (1) • FILTER_ALL_DIRECTIONS_2 (2) • FILTER_DIAGONALLY (3) • FILTER_HORIZONTALLY (4) • FILTER_VERTICALLY (5)
WIN_INFO_ENHANCED_RENDERING	45	Логическая величина: TRUE, если для окна задан улучшенный режим прорисовки. Для того чтобы включить этот режим, используйте оператор Set Window .

Код атрибута	ID	WindowInfo(attribute) возвращает:
WIN_INFO_SMOOTH_TEXT	46	Строковое значение – String: представляет текущий режим сглаживания текста для данного окна. Чтобы изменить режим сглаживания, используйте оператор Set Window .
WIN_INFO_SMOOTH_IMAGE	47	Строковое значение – String: представляет текущий режим сглаживания текста для данного окна. Чтобы изменить режим сглаживания, используйте оператор Set Window .
WIN_INFO_SMOOTH_VECTOR	48	Строковое значение – String: представляет текущий режим сглаживания векторных объектов для данного окна. Чтобы изменить режим сглаживания, используйте оператор Set Window .
WIN_INFO_PARENT_LAYOUT	49	Целое число типа Integer: возвращает идентификатор окна родительского окна отчёта относительно встроеного в отчёт окна; иначе, возвращает 0.
WIN_INFO_CLIENTWIDTH	50	Целое число типа Integer: Возвращает ширину окна-клиента.
WIN_INFO_CLIENTHEIGHT	51	Целое число типа Integer: Возвращает высоту окна-клиента.
WIN_INFO_PRINTER_RESOLUTION	52	Целое число типа Integer: Возвращает разрешение принтера.
WIN_INFO_TOPMOSTWHENFLOATING	53	Логическая величина: TRUE, если окно является самым верхним.

Зависящий от платформы тип, используемый для указателя или обработчика. Это помогает писать приложения, которые работают в обеих версиях MapInfo Pro — и 32-х и 64-разрядной. 64-разрядная версия MapInfo Pro рассматривает такие значения, как имеющие тип LargeInt, а 32-разрядная версия MapInfo Pro рассматривает как целое — Integer.

Если задать WIN_INFO_TYPE в *attribute* и **WindowInfo()** возвращает одно из следующих значений. Идентификаторы окон **Список таблиц** (WIN_TABLE_LIST, 2001), **Управление слоями** (WIN_LAYER_CONTROL, 2002) и **Переместить** (WIN_MOVE_MAP_TO 2003) игнорируются.

Тип окна	ID	Описание окна
WIN_MAPPER	1	Окно Карты
WIN_BROWSER	2	Окно Списка

Тип окна	ID	Описание окна
WIN_LAYOUT	3	Окно Отчет
WIN_GRAPH	4	Окно Графика
WIN_BUTTONPAD	19	Окно панели с кнопками
WIN_TOOLBAR	25	Окно инструментальной панели
WIN_CART_LEGEND	27	Окно картографической Легенды
WIN_3DMAP	28	Окно 3D-карты
WIN_ADORNMENT	32	Элемент оформления
WIN_LEGEND_DESIGNER	35	Окно Легенда
WIN_LAYOUT_DESIGNER	36	Окно Отчёт
WIN_CUSTOM	37	Тип служебного, пользовательского окна (например, торговой площадки MapInfo Marketplace — окно WPF)
WIN_HELP	1001	Окно Справочной системы
WIN_MAPBASIC	1002	Окно MapBasic
WIN_MESSAGE	1003	Окно Сообщение (его открывает Оператор Print)
WIN_RULER	1007	Линейка (вызывается инструментом Линейка)
WIN_INFO	1008	Окно Информация (вызывается инструментом Информация)
WIN_LEGEND	1009	Окно Тематическая легенда
WIN_STATISTICS	1010	Окно Статистика
WIN_MAPINFO	1011	Рабочее окно MapInfo Pro
WIN_WORKSPACE_EXPLORER	2004	Окно Проводник рабочих наборов

Тип окна	ID	Описание окна
WIN_WINDOW_LIST	2005	Окно Список окон
WIN_TOOL_MANAGER	2006	Окно Менеджер программ
WIN_TASK_MANAGER	2007	Окно Диспетчер задач
WIN_CONNECTION_LIST	2008	Окно Список соединений
WIN_NOTIFICATIONS	2009	Предупреждения используются MapInfo Pro или дополнительными программами, для того чтобы сообщить о необходимости обратить внимание на что-то. Предупреждения появляются в окне сообщений.
WIN_QUERY_BUILDER	2010	Улучшенный диалог SQL-запрос

Каждое окно карты имеет специальную временную таблицу, которая содержит данные для косметического слоя карты. Эти таблицы (имеющие имена "Cosmetic1", "Cosmetic2" и т.п.) пользователь не видит. Имена косметических таблиц можно получить от функции `WindowInfo()`, используя код `WIN_INFO_TABLE (10)`. Аналогично, окна отчётов также поддерживают временные скрытые от пользователя таблицы с именами вида "Layout1", имена которых можно получить при помощи функции **`WindowInfo()`** с кодом `WIN_INFO_TABLE (10)`.

Улучшенные параметры рендеринга и сглаживания в окнах конструктора отчётов

Атрибуты, которые возвращают расширенные параметры рендеринга и сглаживания, теперь могут применяться к окнам конструктора отчётов. Используйте `WindowInfo(win, attr)`, для того чтобы задавать расширенные параметры рендеринга для любого окна конструктора отчёта:

- `WIN_INFO_ENHANCED_RENDERING` возвращает `True`, если включен режим расширенного рендеринга, иначе — `False`.
- `WIN_INFO_SMOOTH_IMAGE` возвращает `None`, `Low` или `High`
- `WIN_INFO_SMOOTH_VECTOR` возвращает `None` или `Antialias`
- `WIN_INFO_SMOOTH_TEXT` всегда возвращает `Antialias`

Поведение для новых окон отчёта

Когда создаётся новое окно отчёта, стандартными, используемыми по умолчанию расширенными параметрами рендеринга с визуальными эффектами являются:

- Визуальные эффекты: `True` — истина
- Сглаживание изображения: `High` — высокое
- Сглаживание векторов: сглаживание — `antialias`

- Сглаживание текста: сглаживание всегда включено

Эти параметры можно менять с помощью операторов **Set Window**. При создании нового окна отчёта, расширенные параметры рендеринга с визуальными эффектами по умолчанию совпадают с настройками стилей, заданными в настройках пользователей (кроме сглаживания текста, которое всегда включено в состоянии *Antialias*, независимо от настроек стилей оформления).

Примечание: MapBasic по прежнему управляет сглаживанием при рендеринге карт и легенд с визуальными эффектами с помощью такого же оператора **Set Window Enhanced**, где идентификатор `window_id` применяется либо к рамке с картой в окне конструктора отчёта, либо к отдельному окну карты. Поэтому оператор `Set Designer Legend Antialias On | Off` становится устаревшим.

Условия появления сообщения об ошибке

ERR_BAD_WINDOW_NUM (590), если неверно значение параметра *window_id*.

ERR_FCN_ARG_RANGE (644) — сообщение об ошибке создаётся, если неправильно задано значение аргумента.

Пример

В следующем примере открывается окно Статистика, если оно ещё не открыто.

```
If Not WindowInfo(WIN_STATISTICS,WIN_INFO_OPEN)
Then Open Window WIN_STATISTICS
End If
```

См. также:

[функция \(\)](#), [оператор Browse](#), [оператор Map](#)

Процедура WinFocusChangedHandler

Назначение

Процедура, автоматически выполняющаяся при изменении фокуса окна.

Синтаксис

```
Declare Sub WinFocusChangedHandler
Sub WinFocusChangedHandler
```

```
statement_list
End Sub
```

Описание

Если загруженная программа имеет процедуру **WinFocusChangedHandler**, то MapInfo Pro автоматически выполняет процедуру, когда меняется фокус окна (фокус имеет активное окно). Это вариант выполнения применяется по отношению всех типов окон MapInfo Pro (окнам: **Карта, Список, График, Отчёт, Районирование, Легенда** или **MapBasic**). Для определения идентификатора окна, ставшего активным, используйте в процедуре **WinFocusChangedHandler** функцию **CommandInfo(CMD_INFO_WIN)**.

В процедуре **WinFocusChangedHandler** не может быть использован **оператор Note**, процедура обработчика также не может закрывать или открывать какие-либо окна. Эти ограничения аналогичны тем, которые имеют и другие обработчики, например **процедура SelChangedHandler**.

Процедура **WinFocusChangedHandler** должна быть как можно короче, чтобы не замедлять работу системы.

Пример

Следующий фрагмент текста программы показывает, как можно делать доступной или недоступной команду меню в зависимости от того, активно ли окно карты или нет.

```
Include "mapbasic.def"
Include "menu.def"
Declare Sub Main
Declare sub WinFocusChangedHandler
Sub Main
  ' At this point, we could create a custom menu item
  ' which should only be enabled if the current window
  ' is a Map window...
End Sub

Sub WinFocusChangedHandler
  Dim i_win_type As SmallInt

  i_win_type=WindowInfo(CommandInfo(CMD_INFO_WIN),WIN_INFO_TYPE)

  If i_win_type = WIN_MAPPER Then
    ' here, we could enable a map-related menu item
  Else
    ' here, we could disable a map-related menu item
  End If
End Sub
```

См. также:

процедура WinChangedHandler

Оператор Write

Назначение

Запись данных в открытый файл.

Синтаксис

```
Write # file_num [ , expr ... ]
```

file_num – номер файла, который был присвоен ему при открытии.

expr – выражение для записи в файл.

Описание

Оператор **Write #** записывает определенные данные в открытый файл. Он должен быть открыт в режиме последовательного доступа (Output или Append).

Параметр *file_num* соответствует номеру из предложения **As**, который при открытии файла задает [оператор Open File](#).

Выражений *expr* может быть несколько, в операторе они должны быть разделены запятыми. В этом случае записанные значения в файле автоматически разделяются запятыми. Если список выражений пуст, то записывается пустая строка.

Оператор **Write #** автоматически заключает строковые значения в кавычки при записи в файл. Чтобы записать текст в файл, не используя кавычек, используйте [оператор Print #](#).

Используйте [оператор Input #](#) для чтения записей из файла, созданного оператором **Write #**.

См. также:

[оператор Input #](#), [оператор Open File](#), [оператор Print #](#)

Функция Year()

Назначение

Извлекает год из значения даты. Эту функцию можно вызвать в окне **MapBasic** в среде MapInfo Pro.

Синтаксис

```
Year( date_expr )
```

date_expr – выражение, результат которого есть дата (величина типа Date).

Возвращаемая величина

Короткое целое – SmallInt

Описание

Если **оператор Set Date Window()** имеет значение Off, год определяется часами системы.

Примеры

Пример демонстрирует, как можно использовать функцию **Year()** для извлечения из даты только компоненты года.

```
Dim sampleDate as Date
Set Date Window Off
sampleDate=StringToDate("10/1/98")
Print Year(sampleDate)
' 2098 (or 1998 if the computer's system date is set in the 1900's)
' because with date windowing off MapInfo uses the current century
Set Date Window 50
' now assume that two-digit dates fall in the period 1950-2049
print Year(sampleDate)
' still 2098, because date variable has already been assigned!
sampleDate=StringToDate("10/1/98")
' re-assign variable now that the date window has changed
print Year(sampleDate) ' 1998
Undim sampleDate
```

Функция **Year()** в качестве аргумента может принимать не только переменную типа Date, но и строку. В таком случае произойдет неявное преобразование в формат даты. В примере это отображается:

```
Set Date Window Off
Print Year("10/1/99") ' prints 2099
Set Date Window 50
Print Year("10/1/99") ' prints 1999
```

Можно также использовать функцию **Year()** в операторе SQL Select, формирующем SQL-запрос. В этом примере выбираются определенные строки из таблицы Orders. Предполагается, что

в таблице Orders есть колонка с датами, которая называется OrderDate. Предложение Where оператора Select предписывает MapInfo Pro выбирать заказы только за декабрь 2013.

```
Open Table "orders"  
Select * From orders  
Where Month(orderdate) = 12 And Year(orderdate) = 2013
```

См. также:

[функция CurDate\(\)](#), [функция Day\(\)](#), [функция DateWindow\(\)](#), [функция Minute\(\)](#), [функция Month\(\)](#), [функция Second\(\)](#), [функция Weekday\(\)](#)

А - Библиотеки HTTP и FTP

В этом приложении подробно описаны библиотеки HTTP и FTP, с помощью которых программисты на языке MapBasic могут использовать веб-технологии.

Примечание: Поскольку используется внешняя библиотека, все функции и процедуры, перечисленные в этом приложении, не могут быть использованы в окне **MapBasic** MapInfo Pro.

В этом разделе:

В этом разделе

О библиотеках функций для работы с HTTP и FTP.....	1156
Процедура MICloseContent().....	1156
Процедура MICloseFtpConnection().....	1157
Процедура MICloseFtpFileFind().....	1157
Процедура MICloseHttpConnection().....	1158
Процедура MICloseHttpFile().....	1159
Процедура MICloseSession().....	1159
Функция MICreateSession().....	1160
Функция MICreateSessionFull().....	1161
Функция MIErrorDlg().....	1163
Функция MIFindFtpFile().....	1165
Функция MIFindNextFtpFile().....	1166
Функция MIGetContent().....	1166
Функция MIGetContentBuffer().....	1167
Функция MIGetContentLen().....	1168
Функция MIGetContentString().....	1169
Функция MIGetContentToFile().....	1169
Функция MIGetContentType().....	1170
Функция MIGetCurrentFtpDirectory().....	1171
Функция MIGetErrorCode().....	1172
Функция MIGetErrorMessage().....	1173
Функция MIGetFileURL().....	1173
Функция MIGetFtpConnection().....	1174
Функция MIGetFtpFile().....	1175

Функция MIGetFtpFileFind().....	1177
Процедура MIGetFtpFileName().....	1178
Функция MIGetHttpConnection().....	1179
Функция MIsFtpDirectory().....	1180
Функция MIsFtpDots().....	1180
Функция MIOpenRequest().....	1181
Функция MIOpenRequestFull().....	1182
Функция MIParseURL().....	1184
Функция MIPutFtpFile().....	1186
Функция MIQueryInfo().....	1187
Функция MIQueryInfoStatusCode().....	1188
Функция MISaveContent().....	1190
Функция MISendRequest().....	1190
Функция MISendSimpleRequest().....	1191
Функция MISetCurrentFtpDirectory().....	1192
Функция MISetSessionTimeout().....	1193

О библиотеках функций для работы с HTTP и FTP

Эти библиотеки позволяют получать доступ к сведениям рассылаемым по RSS и другим источникам информации о местоположении, основанным на веб-технологиях, например, погоде, ситуации на дорогах, координатах транспортных средств и т.п., а также настраивать соединение по FTP, поиск, получение и отправку файлов из программы на языке MapBasic. Эта библиотека использует обычные DEF-файлы: HTTPLib.DEF, HTTPType.DEF и HTTPUtil.DEF, которые установлены по адресу: <Your MapBasic Installation Directory>\Samples\MapBasic\INC. Эти файлы следует включить в состав секции заголовков программ. Все функции, описанные в этом приложении, работают при наличии файла Gm1Xlat.dll, который устанавливается вместе с MapInfo Pro.

Мы предлагаем несколько примеров приложений, в которых используются эти библиотеки. Смотрите <Your MapBasic Installation Directory>\Samples\MapBasic\HTTPLib и <Your MapBasic Installation Directory>\Samples\MapBasic\FTPLib, где размещены характерные примеры.

Все функции и процедуры, описанные в этом приложении, являются интерфейсными оболочками соответствующих классов Microsoft MFC. Классы обёртки промежуточного интерфейса (враппера) включают CInternetSession, CHttpConnection, CFtpConnection, CHttpFile и CFtpFileFind. Более подробная информация об использовании зависимых классов приводится в MSDN в разделе о MFC классах: ([http://msdn2.microsoft.com/en-us/library/bk77x1wx\(en-US,vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/bk77x1wx(en-US,vs.80).aspx)).

Процедура MICloseContent()

Назначение

Закрывает и удаляет идентификатор CString и освобождает память.

Синтаксис

```
MICloseContent( ByVal hContent As CString )
```

hContent – идентификатор объекта CString, который требуется удалить.

Описание

Используйте процедуру **MICloseContent()** для того, чтобы закрыть и освободить идентификатор CString, полученный вызовом, в котором используется **функция MIGetContent()**, если идентификатор недоступен.

См. также:

Функция MIGetContent()

Процедура MICloseFtpConnection()

Назначение

Закрывает и удаляет идентификатор CFtpConnection и освобождает память.

Синтаксис

```
MICloseFtpConnection( ByVal hConnection As CFtpConnection )
```

hConnection – идентификатор объекта CFtpConnection, который требуется удалить.

Описание

Используйте **MICloseFtpConnection()**, для того чтобы закрыть и освободить идентификатор CFtpConnection, полученный вызовом, в котором применяется **функция MIGetFtpConnection()**, если идентификатор недоступен.

См. также:

Функция MIGetFtpConnection()

Процедура MICloseFtpFileFind()

Назначение

Закрывает и удаляет идентификатор CFtpFileFind и освобождает память.

Синтаксис

```
MIcloseFtpFileFind( ByVal hFTPFind As CFtpFileFind )
```

hFTPFind – идентификатор удаляемого объекта CFtpFileFind.

Описание

Используйте процедуру **MIcloseFtpFileFind()**, для того чтобы закрыть и освободить идентификатор CFtpFileFind, полученный вызовом, в котором используется **функция MIGetFtpFileFind()**, если идентификатор недоступен.

См. также:

Функция MIGetFtpFileFind()

Процедура MIcloseHttpConnection()

Назначение

Закрывает и удаляет идентификатор CHttpConnection и освобождает память.

Синтаксис

```
MIcloseHttpConnection( ByVal hConnection As CHttpConnection )
```

hConnection – идентификатор объекта CHttpConnection, который требуется удалить.

Описание

Используйте процедуру **MIcloseHttpConnection()**, для того чтобы закрыть и освободить идентификатор CHttpConnection, полученный вызовом, в котором используется **функция MIGetHttpConnection()**, если идентификатор недоступен.

См. также:

Функция MIGetHttpConnection()

Процедура MICloseHttpFile()

Назначение

Закрывает и удаляет идентификатор CHttpFile и освобождает память.

Синтаксис

```
MICloseHttpFile( ByVal hFile As CHttpFile )
```

hFile – идентификатор объекта CHttpFile, который требуется удалить.

Описание

Используйте процедуру **MICloseHttpFile()**, для того чтобы закрыть и освободить идентификатор CHttpFile, полученный вызовом, в котором применяется [функция MIOpenRequest\(\)](#) или [функция MIOpenRequestFull\(\)](#), если идентификатор недоступен.

См. также:

[функция MIOpenRequest\(\)](#), [функция MIOpenRequestFull\(\)](#)

Процедура MICloseSession()

Назначение

Закрывает и удаляет идентификатор CInternetSession и освобождает память.

Синтаксис

```
MICloseSession( ByVal hSession As CInternetSession )
```

hSession – идентификатор объекта CInternetSession, который требуется удалить.

Описание

Используйте процедуру **MICloseSession()**, для того чтобы закрыть и освободить идентификатор CInternetSession, полученный вызовом, в котором используется **функция MICreateSession()** или **функция MICreateSessionFull()**, если идентификатор недоступен.

См. также:

функция MICreateSession(), **функция MICreateSessionFull()**

Функция MICreateSession()

Назначение

Создает объект CInternetSession и возвращает идентификатор этого объекта.

Синтаксис

```
MICreateSession( ByVal strAgent As String ) As CInternetSession
```

strAgent – строка, определяющая имя приложения или иной службы, вызывающей интернет-функции (например, “MapInfo Pro”). Если строка пустая, то будет использовано имя приложения.

Возвращаемая величина

Идентификатор объекта CInternetSession. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

MICreateSession() – самая первая интернет-функция, вызываемая приложением. Для того чтобы создать и инициировать единственный или несколько одновременно открытых Интернет-сеансов, а также предать параметры соединения прокси-серверу, используйте CInternetSession. Если требуется выполнить действия с файлами на сервере по определенному протоколу связи (например, HTTP, FTP), необходимо установить соответствующее соединение с этим сервером. Для того чтобы открыть соединение определенного типа к заданной службе, используйте подходящую функцию, такие как: **функция MIGetHttpConnection()** или **функция MIGetFtpConnection()**. За подробной информацией, обращайтесь к Microsoft MSDN library.

При вызове требуется добавить идентификатор, для получения которого используется процедура **MICloseSession()**, если такой идентификатор больше не применяется.

См. также:

Процедура MICloseSession()

Функция MICreateSessionFull()

Назначение

Создает объект CInternetSession и возвращает идентификатор этого объекта.

Синтаксис

```
MICreateSessionFull( ByVal strAgent As String, ByVal dwContext As Integer,
    ByVal dwAccessType As Integer, ByVal strProxyName As String,
    ByVal strProxyBypass As String, ByVal dwFlags As Integer
    As CInternetSession
```

strAgent – строка, определяющая имя приложения или иной службы, вызывающей интернет-функции (например, “MapInfo Pro”). Если строка пустая, то будет использовано имя приложения.

dwContext – идентификатор контекста операции.

dwAccessType – требуемый тип доступа. Ниже перечислены все используемые значения, только любое из них можно использовать:

Значение dwAccessType	Определение
INTERNET_OPEN_TYPE_PRECONFIG	Соединение по параметрам восстановленным из реестра. По умолчанию используется этот тип организации доступа. Для подключения через TIS-прокси, задайте параметру <i>dwAccessType</i> это значение; реестр будет изменен.
INTERNET_OPEN_TYPE_DIRECT	Прямое интернет-соединение.
INTERNET_OPEN_TYPE_PROXY	Соединение с CERN-прокси.

strProxyName – имя привилегированного CERN прокси-сервера, если параметру *dwAccessType* присвоено значение INTERNET_OPEN_TYPE_PROXY. По умолчанию эта строка пустая.

strProxyBypass – строка, в которой перечислены адреса запасных серверов. Эти адреса можно пропустить при использовании прокси-сервера. Если задана пустая строка, то список исключений будет прочитан из реестра. Этот параметр имеет смысл только в случае, когда параметру *dwAccessType* присвоено значение INTERNET_OPEN_TYPE_PROXY.

dwFlags – определяет некоторые настройки кэширования. По умолчанию этому параметру присвоено значение 0. Этому параметру можно присваивать следующие значения:

Значение dwFlag	Определение
INTERNET_FLAG_DONT_CACHE	Не кэшировать данные ни локально, ни на серверах-шлюзах.
INTERNET_FLAG_OFFLINE	Операции загрузки выполняются, используя только постоянное кэширование. Если искомый объект не будет найден в кэше, возвращается соответствующий код ошибки.

Возвращаемая величина

Идентификатор объекта CInternetSession. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция [MIGetErrorMessage\(\)](#)**.

Описание

MICreateSessionFull() – самая первая интернет-функция, вызываемая приложением. Для того чтобы создать и инициировать единственный или несколько одновременно открытых интернет-сеансов, а также предать параметры соединения прокси-серверу, используйте CInternetSession. Если требуется выполнить действия с файлами на сервере по определенному протоколу связи (например, HTTP, FTP), необходимо установить соответствующее соединение с этим сервером. Для того чтобы открыть соединение определенного типа к заданной службе, используйте подходящую функцию, такие как: **функция [MIGetHttpConnection\(\)](#)** или **функция [MIGetFtpConnection\(\)](#)**. За подробной информацией, обращайтесь к Microsoft MSDN library.

При вызове требуется добавить идентификатор, для получения которого используется **процедура [MICloseSession\(\)](#)**, если такой идентификатор больше не применяется.

См. также:

функция [MICreateSession\(\)](#), **процедура [MICloseSession](#)**

Функция MLErrorDlg()

Назначение

Появится диалог с сообщением об ошибке.

Появится диалог с ошибкой, переданной в MLErrorDlg, если соответствующий диалог существует. Кроме того, эта функция проверяет скрытые ошибки идентификаторов заданных CHttpFile и, при необходимости, показывает диалог.

Синтаксис

```
MLErrorDlg( ByVal hFile As CHttpFile, ByVal dwError As Integer) As Integer
```

hFile – идентификатор CHttpFile.

dwError – код ошибки, который присвоен сообщению об ошибке.

Возвращаемая величина

Возвращает либо одно из нижеперечисленных значений, либо код ошибки.

Код ошибки	Описание
ERROR_SUCCESS	Функция выполнена полностью. При регистрации пользователя появление этого значения указывает, что пользователь нажал кнопку Отмена .
ERROR_CANCELLED	Выполнение функции было остановлено пользователем.
ERROR_INTERNET_FORCE_RETRY	Указывает, что функция должна повторно выполнить переданный ей запрос. При регистрации пользователя появление этого значения указывает, что пользователь нажал кнопку ОК .

Описание

Эту функцию следует использовать, для того чтобы сообщение об ошибке выводилось в окне диалога, если соответствующее окно диалога существует. Эта функция также проверяет

идентификаторы любых скрытых ошибок и, при необходимости, показывает диалог. За дополнительной информацией, обращайтесь к Microsoft MSDN library. Допустимые коды ошибок:

Код ошибки	Описание
ERROR_INTERNET_HTTP_TO_HTTPS_ON_REDIRECT	Предупреждает пользователя о нулевом пересечении сообщений к и от защищенного сайта.
ERROR_INTERNET_INCORRECT_PASSWORD	Выводит диалог, запрашивающий имя пользователя и пароль.
ERROR_INTERNET_INVALID_CA	Предупреждает пользователя, что функция не опознала организацию, выдавшую сертификат для данного Secure Socket Layer (SSL) сайта.
ERROR_INTERNET_POST_IS_NON_SECURE	Показывает предупреждение о передаче данных на сервер по незащищенному подключению.
ERROR_INTERNET_SEC_CERT_CN_INVALID	Предупреждает, что параметр Common Name (поле имени хоста) сертификата SSL неправильный. Показывает диалог с недействительным параметром SSL Common Name и позволяет пользователю проверить неправильный сертификат. Кроме того позволяет пользователю выбрать сертификат в ответ на запрос сервера.
ERROR_INTERNET_SEC_CERT_DATE_INVALID	Предупреждает пользователя, что срок действия SSL-сертификата истек.

За дополнительной информацией, обращайтесь к Microsoft MSDN library.

См. также:

[функция MIOpenRequest\(\)](#), [функция MISendRequest\(\)](#)

Функция MIFindFtpFile()

Назначение

Ищет FTP-файл по заданному идентификатору CFtpFileFind.

Синтаксис

```
MIFindFtpFile( ByVal hFTPFind As CFtpFileFind, ByVal strName As String
)
    As SmallInt
```

hFTPFind – идентификатор CFtpFileFind.

strDirName – строка с именем файла, который требуется найти. Если строка пустая, будут найдены все файлы по групповому признаку (*).

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

После вызова функции **MIFindFtpFile()**, для того чтобы получить первый FTP-файл, можно повторить поиск других FTP-файлов, в котором используется [функция MIFindNextFtpFile\(\)](#).

См. также:

[функция MIGetFtpFileFind\(\)](#), [функция MIFindNextFtpFile\(\)](#), [функция MIGetFtpFileName\(\)](#), [функция MIsFtpDirectory\(\)](#), [функция MIsFtpDots\(\)](#)

Функция MIFindNextFtpFile()

Назначение

Продолжает поиск, начатый запросом, в котором использовалась [функция MIFindFtpFile\(\)](#) с заданным идентификатором CFtpFileFind.

Синтаксис

```
MIFindNextFtpFile( ByVal hFTPFind As CFtpFileFind ) As SmallInt
```

hFTPFind – идентификатор CFtpFileFind.

Возвращаемая величина

Ненулевое значение, если существуют дополнительные файлы; нуль – если найденный файл последний в каталоге или произошла ошибка.

Описание

Необходимо использовать функцию **MifindNextFtpFile()** хотя бы один раз, прежде чем вызывать любую функцию определения атрибутов, в числе которых: [процедура MIFindFtpFile\(\)](#), [функция MIFindFtpFile\(\)](#), [функция MIFindFtpFile\(\)](#) и [функция MIFindFtpFile\(\)](#).

См. также:

[функция MIFindFtpFile\(\)](#), [функция MIFindFtpFile\(\)](#), [процедура MIFindFtpFile\(\)](#), [функция MIFindFtpFile\(\)](#), [функция MIFindFtpFile\(\)](#)

Функция MIFindFtpFile()

Назначение

Получить оглавление файла.

Синтаксис

```
MIGetContent ( ByVal hFile As CHttpFile ) As CString
```

hFile – идентификатор CHttpFile.

Возвращаемая величина

Идентификатор объекта CString с оглавлением файла. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется функция **MIGetErrorMessage()**.

Описание

MIGetContent() – получает оглавление файла и сохраняет его в объекте CString, в отличие от способа, когда используется функция **MIGetContentToFile()**. Следует учитывать, что функции MIGetContentToFile и MIGetContent взаимно исключают друг друга, можно использовать только одну из них за все время существования объекта CHttpFile object.

При вызове требуется добавить идентификатор, для получения которого используется процедура **MICloseContent()**, если такой идентификатор больше не применяется.

См. также:

функция **MIOpenRequest()**, функция **MISendRequest()**, процедура **MICloseContent()**, функция **MIGetContentToFile()**

Функция MIGetContentBuffer()

Назначение

Получить оглавление в формате строки заданной длины.

Синтаксис

```
MIGetContentBuffer ( ByVal hContent As CString, pBuffer As String,  
ByVal nLen As Integer As SmallInt
```

hContent – идентификатор CString.

pBuffer – ссылка на строку, в которой будет помещено оглавление файла.

nLen – размер строки *pBuffer*, в байтах или символах.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Для того чтобы получить оглавление файла в формате строки заданной длины, используйте функцию **MIGetContentBuffer()**. Требуется выделить память для *pBuffer*, до вызова этой функции.

См. также:

[функция MIGetContent\(\)](#), [функция MIGetContentLen\(\)](#), [функция MIGetContentString\(\)](#).

Функция MIGetContentLen()

Назначение

Получить длину оглавления.

Синтаксис

```
MIGetContentLen( ByVal hContent As CString ) As Integer
```

hContent – идентификатор CString.

Возвращаемая величина

Длина оглавления – число букв или байт. Нуль может означать либо ошибку, либо пустое оглавление. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Для того чтобы получить длину оглавления файла в виде числа букв или байт, используйте функцию **MIGetContentLen()**.

См. также:

[Функция MIGetErrorMessage\(\)](#)

Функция MIGetContentString()

Назначение

Получить оглавление в формате строки.

Синтаксис

```
MIGetContentString( ByVal hContent As CString ) As String
```

hContent – идентификатор CString.

Возвращаемая величина

Строка с оглавлением файла. Для того чтобы определить причину ошибки, если возвращена пустая строка, следует составить вызов, в котором используется [функция MIGetErrorMessage\(\)](#).

Описание

Эту функцию следует использовать, чтобы получить оглавление файла в формате строки.

См. также:

[функция MIGetContent\(\)](#), [функция MIGetContentLen\(\)](#)

Функция MIGetContentToFile()

Назначение

Сохраняет оглавление в заданном файле.

Синтаксис

```
MIGetContentToFile( ByVal hFile As CHttpFile,
  ByVal strFileName As String As SmallInt
```

hFile – идентификатор CHttpFile.

strFileName – строка, определяющая имя локального файла, в котором будет сохранено оглавление *hFile*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Эта функция используется, для того чтобы сохранить оглавление CHttpFile в локальном файле, в отличие от способа, когда используется MIGetContent. Будет создан новый файл с заданным именем. Если такой файл уже существует, он будет обрезан до нулевой длины. Следует учитывать, что функции MIGetContentToFile и MIGetContent взаимно исключают друг друга, можно использовать только одну из них за все время существования объекта CHttpFile object.

См. также:

[функция MIOpenRequest\(\)](#), [функция MIsendRequest\(\)](#), [функция MIGetContent\(\)](#).

Функция MIGetContentType()

Назначение

Получить тип оглавления файла.

Синтаксис

```
MIGetContentType( ByVal hFile As CHttpFile, pBuffer As String,
  pBufferLength As Integer As SmallInt
```

hFile – идентификатор CHttpFile.

pBuffer – ссылка на строку, в которой будет помещен тип оглавления файла.

pBufferLength – ссылка на целую переменную, в которой содержится длина *pBuffer* в виде числа букв или байт. При успешном выполнении этой функции (строка будет записана в *pBuffer*), будет храниться длина строки в буквах минус 1, для ограничения символа NULL.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Функцию **MIGetContentType()** следует использовать для определения типа оглавления файла в виде отформатированной строки. Например, в *pBuffer* можно записать “image/jpeg” или “text/html”.

См. также:

[Функция MIQueryInfo\(\)](#)

Функция MIGetCurrentFtpDirectory()

Назначение

Получить имя текущего каталог на FTP-сервере по заданному идентификатору CFtpConnection.

Синтаксис

```
MIGetCurrentFtpDirectory( Byval hConnection As CFtpConnection,  
    pDirName As String, pLen As Integer As SmallInt
```

hConnection – идентификатор CFtpConnection.

pDirName – ссылка на строку, в которой будет помещено имя каталога.

pLen – ссылка на переменную целого типа, в которой хранится размер буфера, определенного *pDirName* на входе; число букв хранящихся в *pDirName* на выходе.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Параметрами *pDirName* являются полностью или частично определенные имена файлов, относительно текущего каталога. В качестве разделителя имен каталогов и файлов можно использовать как символ обратной косой черты (\), так и символ прямой косой черты (/).

MIGetCurrentFtpDirectory() – транслирует разделители имен каталогов в нужные символы.

См. также:

[функция MIGetFtpConnection\(\)](#), [функция MISetCurrentFtpDirectory\(\)](#)

Функция MIGetErrorCode()

Назначение

Получить последнее сообщение об ошибке, полученное в результате выполнения функции этой библиотеки.

Синтаксис

```
MIGetErrorCode( ) As Integer
```

Возвращаемая величина

Код последней ошибки.

Описание

Функцию **MIGetErrorCode()** следует использовать, если при выполнении какой-либо функции произойдет ошибка, для выяснения кода ошибки. Для того чтобы получить сообщение об ошибке в виде строки, следует использовать вызов, в котором применяется [функция MIGetErrorMessage\(\)](#).

Значение кода ошибки появляется только при выполнении других функций API HTTP и FTP. Эту функцию, в основном, удобно применять для выявления сообщений об ошибках, произошедших в результате выполнения операций, в которых использовалась [функция MISendRequest\(\)](#) или [функция MISendSimpleRequest\(\)](#), а нужное значение можно передать вызову, в котором используется [функция MIErrorDlg\(\)](#).

См. также:

функция **MIGetErrorMessage()**, функция **MISendRequest()**, функция **MISendSimpleRequest()**,
функция **MIErrorDlg()**

Функция **MIGetErrorMessage()**

Назначение

Возвращает текст с сообщением о последней ошибке.

Синтаксис

```
MIGetErrorMessage() As String
```

Возвращаемая величина

Строка с сообщением об ошибке.

Описание

Если возвращенное функцией значение сигнализирует об ошибке и требуется установить её причину, то для того чтобы получить полезную информацию, функцию **MIGetErrorMessage()** следует использовать непосредственно после функции, которая вызвала ошибку. Многие сообщения об ошибках являются системными.

Функция **MIGetFileURL()**

Назначение

Получить имя HTTP-файла в виде URL.

Синтаксис

```
MIGetFileURL( ByVal hFile As CHttpFile, pURL As String,
               ByVal lURLLen As Integer ) As SmallInt
```

hFile – идентификатор CHttpFile.

pURL – ссылка на строковую переменную, в которую будет передано имя HTTP-файла в виде URL.

pURLLen – размер *pURL* в виде числа букв или байт.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Для того чтобы получить имя HTTP-файла в виде URL, используйте **MIGetFileURL()**.

См. также:

[функция MIsendRequest\(\)](#), [функция MIOpenRequest\(\)](#)

Функция MIGetFtpConnection()

Назначение

Установить FTP-соединение и получить идентификатор объекта CFtpConnection.

Синтаксис

```
MIGetFtpConnection( ByVal hSession As CInternetSession,
                     ByVal strServer As String, ByVal strUserName As String,
                     ByVal strPassword As String, ByVal nPort As INTERNET_PORT )
                     As CFtpConnection
```

hSession – идентификатор CInternetSession.

strServer – строка с именем FTP-сервера.

strUserName – строка с именем пользователя, устанавливающего соединение.

strPassword – строка с паролем, который следует использовать при соединении.

nPort – число, определяющее номер порта TCP/IP на сервере.

Возвращаемая величина

Идентификатор объекта CFtpConnection. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

MIGetFtpConnection() устанавливает соединение с FTP-сервером, создает и возвращает идентификатор объекта CFtpConnection. Она не выполняет никаких операций на сервере. Например, если требуется получить или передать файлы на сервер, то придется выполнять эти операции отдельно.

При вызове требуется добавить идентификатор, для получения которого используется **процедура MICloseFtpConnection()**, если такой идентификатор больше не применяется.

См. также:

процедура MICloseFtpConnection(), **функция MIGetHttpConnection()**, **функция MICreateSession()**, **функция MICreateSessionFull()**, **функция MIParseURL()**

Функция MIGetFtpFile()

Назначение

Получить файл с FTP-сервера по заданному идентификатору CFtpConnection и сохранить его локально.

Синтаксис

```
MIGetFtpFile( ByVal hConnection As CFtpConnection,
  ByVal strRemoteFile As String, ByVal strLocalFile As String,
  ByVal bFailIfExists As SmallInt, ByVal dwAttributes As Integer,
  ByVal dwFlags As Integer ) As SmallInt
```

hConnection – идентификатор CFtpConnection.

strRemoteFile – строка с именем файла, который требуется получить с FTP-сервера.

strLocalFile – строка с именем, под которым полученный файл требуется сохранить на локальной системе.

bFailIfExists – указывает на то, что файл с таким именем уже существует. Если локальный файл с таким именем уже существует, а для этого параметра установлено значение TRUE функция **MIGetFtpFile()** не будет выполнена. Иначе, **MIGetFtpFile()** сотрет существующую копию файла.

dwAttributes – определяет атрибуты файла. Возможна любая из ниже перечисленных комбинация флагов **FILE_ATTRIBUTE_***.

Значение dwAttribute	Определение
FILE_ATTRIBUTE_ARCHIVE	Для файла задан атрибут архивирования. Некоторые приложения используют этот атрибут, для того чтобы отметить файл для архивирования или удаления.
FILE_ATTRIBUTE_COMPRESSED	Файл или каталог сжаты. Сжатый файл, в котором все данные сжаты. Сжатый каталог, в котором все новые файлы и подкаталоги будут сжаты.
FILE_ATTRIBUTE_DIRECTORY	Файл является каталогом.
FILE_ATTRIBUTE_NORMAL	У файла нет дополнительных атрибутов. Этот атрибут может использоваться только индивидуально. Если установить любой другой атрибут, то атрибут FILE_ATTRIBUTE_NORMAL будет снят:
FILE_ATTRIBUTE_HIDDEN	Файл является скрытым. Такой файл не будет включен в стандартный список каталога.
FILE_ATTRIBUTE_READONLY	Файл защищен от записи. Приложения могут читать файл, но не могут записывать в него или удалять его.
FILE_ATTRIBUTE_SYSTEM	Файл является частью или используется исключительно операционной системой.
FILE_ATTRIBUTE_TEMPORARY	Файл используется временно. Приложения могут записывать в файл только в случае экстренной необходимости. Основная часть данных в таком файле остается в памяти не перенося её на носитель, поскольку файл будет скоро уничтожен.

dwFlags – определяют условия, при которых происходит перенос данных. Этот параметр может принимать любые из нижеперечисленных значений:

Значение dwFlags	Определение
FTP_TRANSFER_TYPE_ASCII	Перенос файлов будет осуществляться по методу FTP ASCII (Type A). Информация о настройках и форматировании файла будет заменена локальными эквивалентами.
FTP_TRANSFER_TYPE_BINARY	Перенос файлов будет осуществляться бинарным методом FTP (Type I). Файлы будут точно скопированы, без всяких изменений. Этот метод передачи используется по умолчанию.
FTP_TRANSFER_TYPE_UNKNOWN	Будет использоваться вариант FTP_TRANSFER_TYPE_BINARY.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Оба параметра *strRemoteFile* и *strLocalFile* могут быть либо именами определенными относительно текущего каталога, либо быть полностью определенным именем, включая имена каталогов. В качестве разделителя имен каталогов и файлов можно использовать как символ обратной косой черты (\), так и символ прямой косой черты (/).

См. также:

[функция MIGetFtpConnection\(\)](#), [функция MIPutFtpFile\(\)](#)

Функция MIGetFtpFileFind()

Назначение

Получает идентификатор объекта CFtpFileFind.

Синтаксис

```
MIGetFtpFileFind( ByVal hConnection As CFtpConnection ) As CFtpFileFind
```

hConnection – идентификатор CFtpConnection.

Возвращаемая величина

Идентификатор объекта CFtpFileFind. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

Класс CFtpFileFind включает поиск файлов в интернете по FTP-серверам.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется **процедура MICloseFtpFileFind()**, если такой идентификатор больше не применяется.

См. также:

функция MIGetFtpConnection(), **процедура MICloseFtpFileFind**

Процедура MIGetFtpFileName()

Назначение

Получить имя найденного файла, идентификатор CFtpFileFind которого задан.

Синтаксис

```
MIGetFtpFileName( ByVal hFTPFind As CFtpFileFind, pFileName As String,  
ByVal bufferlen As Integer )
```

hFTPFind – идентификатор CFtpFileFind.

pFileName – ссылка на строковую переменную, в которой будет храниться имя найденного файла.

bufferlen – размерность буфера, на который ссылается *pFileName*.

Описание

Прежде, чем применять **функцию MIGetFtpFileName()**, требуется организовать хотя бы один раз вызов, в котором используется **функция MIFindNextFtpFile()**.

См. также:

функция MIGetFtpFileFind(), **функция MIFindNextFtpFile()**, **функция MIFindFtpFile()**

Функция MIGetHttpConnection()

Назначение

Установить HTTP-соединение и получить идентификатор объекта CHttpConnection.

Синтаксис

```
MIGetHttpConnection( ByVal hSession As CInternetSession,  
    ByVal strServer As String, ByVal nPort As INTERNET_PORT  
    As CHttpConnection
```

hSession – идентификатор CInternetSession.

strServer – строка с именем HTTP-сервера.

nPort – число, определяющее номер порта TCP/IP на сервере.

Возвращаемая величина

Идентификатор объекта CHttpConnection. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

MIGetHttpConnection() устанавливает соединение с HTTP-сервером, создает и возвращает идентификатор объекта CHttpConnection. Она не выполняет никаких операций на сервере. Например, если требуется получить идентификатор HTTP, то придется выполнять эти операции отдельно.

При вызове требуется добавить идентификатор, для получения которого используется **процедура MICloseHttpConnection()**, если такой идентификатор больше не применяется.

См. также:

процедура [MICloseHttpConnection\(\)](#), функция [MIGetFtpConnection\(\)](#), функция [MISession\(\)](#), функция [MICreateSessionFull\(\)](#), функция [MIParseURL\(\)](#)

Функция `MIIsFtpDirectory( )`

Назначение

Определить, является ли найденный файл каталогом, заданным идентификатором `CFtpFileFind`.

Синтаксис

```
MIIsFtpDirectory( ByVal hFTPFind As CFtpFileFind ) As SmallInt
```

hFTPFind – идентификатор `CFtpFileFind`.

Возвращаемая величина

Ненулевое значение, если найденный файл является каталогом; иначе 0.

Описание

Требуется по крайней мере единственный вызов, в котором используется [функция `MIFindNextFtpFile\( \)`](#), выполненный прежде, чем вызывать [функцию `MIIsFtpDirectory\( \)`](#).

См. также:

функция [MIGetFtpFileFind\(\)](#), функция [MIFindNextFtpFile\(\)](#), функция [MIFindFtpFile\(\)](#), процедура [MIGetFtpFileName\(\)](#)

Функция `MIIsFtpDots( )`

Назначение

Проверить маркеры текущего и родительского каталога во время итерации по файлам с заданным идентификатором `CFtpFileFind`.

Синтаксис

```
MIIsFtpDots( ByVal hFTPFind As CFtpFileFind ) As SmallInt
```

hFTPFind – идентификатор CFtpFileFind.

Возвращаемая величина

Ненулевое значение, если имя файла “.” или “..”, что говорит о том, что найденный файл это каталог. Иначе 0.

Описание

Требуется по крайней мере единственный вызов, в котором используется **функция MIFindNextFtpFile()**, выполненный прежде, чем вызывать **функцию MIIsFtpDots()**.

См. также:

функция MIGetFtpFileFind(), **функция MIFindNextFtpFile()**, **функция MIFindFtpFile()**, **процедура MIGetFtpFileName()**

Функция MIOpenRequest()

Назначение

Установить HTTP-соединение.

Синтаксис

```
MIOpenRequest( ByVal hConnection As CHttpConnection,  
  ByVal nVerb As Integer, ByVal strObjectName As String  
  ) As CHttpFile
```

hConnection – идентификатор CHttpConnection.

nVerb – число, ассоциированное с типом HTTP-запроса. Может быть одним из нижеперечисленных:

```
HTTP_VERB_POST  
HTTP_VERB_GET  
HTTP_VERB_HEAD
```

```

HTTP_VERB_PUT
HTTP_VERB_LINK
HTTP_VERB_DELETE
HTTP_VERB_UNLINK

```

strObjectName – строка с объектом, над которым требуется выполнить выбранную команду. Обычно это имя файла, исполняемый модуль или описание объектов поиска.

Возвращаемая величина

Идентификатор запрашиваемого объекта CHttpFile. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Эта функция устанавливает HTTP-соединение и возвращает идентификатор объекта CHttpFile, с помощью которого обеспечиваются запросы к службам и чтение файлов на HTTP-сервере. Если во время интернет-сеанса требуется прочитать данные с HTTP-сервера, необходимо получить идентификатор объекта CHttpFile.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется [процедура MICloseHttpFile\(\)](#), если такой идентификатор больше не применяется.

См. также:

[функция MIOpenRequestFull\(\)](#), [функция MIParseURL\(\)](#)

Функция MIOpenRequestFull()

Назначение

Установить HTTP-соединение.

Синтаксис

```

MIOpenRequestFull( ByVal hConnection As CHttpConnection,
  ByVal nVerb As Integer, ByVal strObjectName As String,
  ByVal strReferer As String, ByVal dwContext As Integer,
  ByVal strVersion As String, ByVal dwFlags As Integer )
  As CHttpFile

```

hConnection – идентификатор CHttpConnection.

nVerb – число, ассоциированное с типом HTTP-запроса. Может быть одним из нижеперечисленных:

- HTTP_VERB_POST
- HTTP_VERB_GET
- HTTP_VERB_HEAD
- HTTP_VERB_PUT
- HTTP_VERB_LINK
- HTTP_VERB_DELETE
- HTTP_VERB_UNLINK

strObjectName – строка с объектом, над которым требуется выполнить выбранную команду. Обычно это имя файла, исполняемый модуль или описание объектов поиска.

strReferer – строка с адресом (URL) документа, от которого был получен URL в запросе (*strObjectName*). Если строка пустая, – никакого заголовка HTTP не задано.

dwContext – идентификатор контекста операции **MIOpenRequestFull()**. За подробной информацией, обращайтесь к Microsoft MSDN library.

strVersion – строка, определяющая номер версии HTTP. Если строка пустая, будет использован вариант “HTTP/1.0”.

dwFlags – любая комбинация следующих флагов INTERNET_FLAG_*:

Значение dwFlag	Определение
INTERNET_FLAG_RELOAD	Принудительно загружает выбранный файл, объект или список файлов каталога с сервера, а не из кэша.
INTERNET_FLAG_DONT_CACHE	Не помещает полученное в кэш.
INTERNET_FLAG_MAKE_PERSISTENT	Добавляет полученное в кэш для постоянного хранения. Это значит, что при стандартной очистке кэша, проверке на совместимость или сборе мусора, этот объект не будет удален из кэша.
INTERNET_FLAG_SECURE	При транзакции будет использована защищенная семантика. При этом все сообщения транслируются SSL/PCT, что имеет значение только при HTTP-запросах.
INTERNET_FLAG_NO_AUTO_REDIRECT	Используется только с HTTP, указывает на то, что функция MISendRequest() не должна автоматически применяться для переадресации.

Возвращаемая величина

Идентификатор запрашиваемого объекта CHttpFile. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

Эта функция устанавливает HTTP-соединение и возвращает идентификатор объекта CHttpFile, с помощью которого обеспечиваются запросы к службам и чтение файлов на HTTP-сервере. Если во время интернет-сеанса требуется прочитать данные с HTTP-сервера, необходимо получить идентификатор объекта CHttpFile. Эта функция-надстройка – оболочка над MFC функцией OpenRequest с дополнительными параметрами, определяющими допустимые типы. В этой версии функция-надстройка всегда присваивает этому параметру значение NULL.

При вызове требуется добавить идентификатор, для получения которого используется **процедура MICloseHttpFile()**, если такой идентификатор больше не применяется.

См. также:

функция MIGetHttpConnection(), **функция MIOpenRequest()**, **функция MIParseURL()**

Функция MIParseURL()

Назначение

Разбирает строку с URL и возвращает тип и компоненты службы.

Синтаксис

```
MIParseURL( ByVal strURL As String, pServiceType As Integer,
  pServer As String, ByVal nServerLen As Integer,
  pObject As String, ByVal nObjectLen As Integer,
  pPort As INTERNET_PORT )
  As SmallInt
```

strURL – строка с URL, который требуется разобрать.

pServiceType – ссылка на целую переменную, которой будет передан тип интернет-службы. Допустимы следующие значения:

- INTERNET_SERVICE_FTP
- INTERNET_SERVICE_GOPHER

- INTERNET_SERVICE_HTTP
- AFX_INET_SERVICE_UNK
- AFX_INET_SERVICE_FILE
- AFX_INET_SERVICE_MAILTO
- AFX_INET_SERVICE_MID
- AFX_INET_SERVICE_CID
- AFX_INET_SERVICE_NEWS
- AFX_INET_SERVICE_NNTP
- AFX_INET_SERVICE_PROSPERO
- AFX_INET_SERVICE_TELNET
- AFX_INET_SERVICE_WAIS
- AFX_INET_SERVICE_AFS
- AFX_INET_SERVICE_HTTPS

strsServer – ссылка на строковую переменную, в которой указан первый после типа службы сегмент URL.

nServerLen – размер буфера, на который ссылается *strsServer*.

pObject – ссылка на объект, к которому обращается URL (может быть пустой).

nObjectLen – размер буфера, на который ссылается *pObject*.

pPort – ссылка на целую переменную с заданным номером порта из любого из разделов URL: либо сервера, либо объекта, если такой существует. Номер порта используется для идентификации порта TCP/IP, используемого на сервере.

Возвращаемая величина

Ненулевое значение, если URL был успешно разобран; иначе, 0, если он пустой или не содержит знакомый тип интернет-службы. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Функция **MIParseURL()** разбирает строку с URL и возвращает тип и компоненты службы. Например, при разборе URL, заданного в форме: `ftp://ftp.mysite.org/`, отдельные компоненты будут храниться следующим образом:

```
pServer == "ftp.mysite.org"
pObject == "/"
nPort == #port
pServiceType == INTERNET_SERVICE_FTP
```

Функция MIPutFtpFile()

Назначение

Сохранить файл с заданным идентификатором CFtpConnection на FTP-сервере.

Синтаксис

```
MIPutFtpFile( ByVal hConnection As CFtpConnection,
  ByVal strLocalFile As String, ByVal strRemoteFile As String,
  ByVal dwFlags As Integer )
  As SmallInt
```

hConnection – идентификатор CFtpConnection.

strLocalFile – строка с локальным именем файла.

strRemoteFile – строка с именем, под которым полученный файл требуется сохранить на FTP-сервере.

dwFlags – определяют условия, при которых происходит перенос данных. Этот параметр может принимать любые из нижеперечисленных значений:

Значение dwFlag	Определение
FTP_TRANSFER_TYPE_ASCII	Передача файла будет осуществляться ASCII методом FTP (Type A). Информация о настройках и форматировании будет заменена локальными эквивалентами.
FTP_TRANSFER_TYPE_BINARY	Передача файлов будет осуществляться бинарным методом FTP (Type I). Файлы будут точно скопированы, без всяких изменений. Этот метод передачи используется по умолчанию.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Оба параметра *strRemoteFile* и *strLocalFile* могут быть либо именами определенными относительно текущего каталога, либо быть полностью определенным именем, включая имена каталогов. В качестве разделителя имен каталогов и файлов можно использовать как символ обратной косой черты (\), так и символ прямой косой черты (/).

См. также:

[функция MIGetFtpConnection\(\)](#), [функция MIGetFtpFile\(\)](#)

Функция MIQueryInfo()

Назначение

Возвращает отклик или заголовок HTTP-запроса.

Синтаксис

```
MIQueryInfo( ByVal hFile As CHttpFile, ByVal dwInfoLevel As Integer,
  pBuffer As String, pBufferLength As Integer ) As SmallInt
```

hFile – идентификатор CHttpFile.

dwInfoLevel – комбинация атрибутов запроса и флага модификатора, с помощью которого задается тип запрашиваемой информации: список модификаторов приводится в библиотеке Microsoft MSDN.

pBuffer – ссылка на строку, в которой будет помещена информация. Для атрибута HTTP_QUERY_CUSTOM, в *pBuffer* указывается, какое имя заголовка будет участвовать в запросе.

pBufferLength – ссылка на целую переменную, в которой содержится длина *pBuffer* в виде числа букв или байт. При успешном выполнении этой функции (строка будет записана в *pBuffer*), будет храниться длина строки в буквах минус 1, для ограничения символа NULL.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Эту функцию следует использовать для получения отклика или заголовка из HTTP-запроса. Описание значений атрибутов смотрите в библиотеке Microsoft MSDN в разделе о флагах информации о запросах: ([https://msdn.microsoft.com/en-us/library/windows/desktop/aa385351\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa385351(v=vs.85).aspx)).

См. также:

функция `MIOpenRequest()`, функция `MISendRequest()`

Функция `MIQueryInfoStatusCode( )`

Назначение

Получить код состояния, связанного с HTTP-запросом.

Синтаксис

```
MIQueryInfoStatusCode( ByVal hFile As CHttpFile, pStatusCode As Integer  
s SmallInt
```

hFile – идентификатор `CHttpFile`.

pStatusCode – ссылка на целую переменную для хранения кода состояния. Код состояния сигнализирует об успешном или неудачном выполнении заданного события. Коды состояния HTTP разделены на группы, по успешному или неудачному выполнению запроса. В следующей таблице выделены группы кодов и часто встречающиеся отдельные коды состояния HTTP.

Таблица 6. Группы кодов состояния HTTP

Группа	Значение
200-299	Успешно
300-399	Информация
400-499	Ошибка запроса

Группа	Значение
500-599	Ошибка на сервере

Таблица 7. Часто встречающиеся коды состояния HTTP

Код состояния	Значение
200	URL найден, передача будет продолжена.
400	Непонятный запрос.
404	Запрошенный URL не найден.
405	Сервер не поддерживает запрошенный метод.
500	Неизвестная ошибка на сервере.
503	Достигнута емкость сервера.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется **функция `MIGetErrorMessage()`**.

Описание

Эту функцию следует использовать для получения кода состояния, связанного с HTTP-запросом. За подробной информацией, обращайтесь к Microsoft MSDN library.

См. также:

функция `MIOpenRequest()`, функция `MISendRequest()`

Функция MISaveContent()

Назначение

Сохраняет оглавление в заданном файле.

Синтаксис

```
MISaveContent ( ByVal hContent As CString, ByVal strFileName As String  
                As SmallInt
```

hContent – идентификатор CString.

strFileName – строка, определяющая имя файла, в котором будет сохранено оглавление.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

С помощью этой функции можно сохранить содержимое в файл. Будет создан новый файл с заданным именем. Если такой файл уже существует, он будет обрезан до нулевой длины.

См. также:

[Функция MIGetContent\(\)](#)

Функция MISendRequest()

Назначение

Передать запрос на HTTP-сервер.

Синтаксис

```
MISendRequest ( ByVal hFile As CHttpFile, ByVal strHeaders As String,
    ByVal dwHeadersLen As Integer, ByVal strOptional As String,
    ByVal dwOptionalLen As Integer, ByVal bAuthenticate As SmallInt
    As SmallInt
```

hFile – идентификатор CHttpFile.

strHeaders – строка с именами заголовков, которые требуется передать.

dwHeadersLen – длина заголовков, определенных в *strHeaders*.

strOptional – дополнительные данные, которые требуется передать сразу после заголовков запроса. Обычно используются операции POST и PUT. Может быть пустым, если дополнительной информации передавать не требуется.

dwOptionalLen – длина *strOptional*.

bAuthenticate – определяет, следует ли проверять идентификацию или нет.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Эта функция запрашивает HTTP-сервер.

См. также:

[функция MIOpenRequest\(\)](#), [функция MIOpenRequestFull\(\)](#)

Функция MISendSimpleRequest()

Назначение

Передать запрос на HTTP-сервер.

Синтаксис

```
MISendSimpleRequest( ByVal hFile As CHttpFile,
    ByVal bAuthenticate As SmallInt ) As SmallInt
```

hFile – идентификатор CHttpFile.

bAuthenticate – определяет, следует ли проверять идентификацию или нет.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Эта функция запрашивает HTTP-сервер.

См. также:

[функция MIOpenRequest\(\)](#), [функция MIOpenRequestFull\(\)](#), [функция MISendRequest\(\)](#)

Функция MISetCurrentFtpDirectory()

Назначение

Сменить каталог на FTP-сервере с заданным идентификатором CFtpConnection.

Синтаксис

```
MISetCurrentFtpDirectory( Byval hConnection As CFtpConnection,
    Byval strDirName As String ) As SmallInt
```

hConnection – идентификатор CFtpConnection.

strDirName – строка с именем каталога.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Параметр *pDirName* может быть полностью или частично определенным именем файла, относительно текущего каталога. В качестве разделителя имен каталогов и файлов можно использовать как символ обратной косой черты (\), так и символ прямой косой черты (/).

MISetCurrentFtpDirectory() – транслирует разделители имен каталогов в нужные символы.

См. также:

MIGetFtpConnection() функция, **MIGetCurrentFtpDirectory()** функция

Функция MISetSessionTimeout()

Назначение

Задать интервала ожидания для интернет-сеанса.

Синтаксис

```
MISetSessionTimeout( ByVal hSession As CInternetSession,  
    ByVal Connect As Integer, ByVal Send As Integer,  
    ByVal Receive As Integer ) As SmallInt
```

hSession – идентификатор объекта CInternetSession.

Connect – целая переменная с значением интервала ожидания, которое будет использоваться при запросе на интернет-соединение, в миллисекундах.

Send – целая переменная с интервалом ожидания, который будет использоваться при передаче запроса, в миллисекундах.

Receive – целая переменная с интервалом ожидания, который будет использоваться для отклика на запрос, в миллисекундах.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется **функция MIGetErrorMessage()**.

Описание

Эту функцию следует использовать, для того чтобы задать интервал ожидания для интернет-сеанса. По умолчанию, для каждого из параметров (*Connect*, *Send*, *Receive*) используется 0. За подробной информацией, обращайтесь к Microsoft MSDN library.

См. также:

MICreateSession() функция, **MICreateSessionFull()** функция

В - Библиотека XML

В этом приложении подробно описана библиотека для обработки документов XML, с помощью которых программисты на языке MapBasic могут создавать и разбирать документы XML и использовать другие Web-технологии.

Примечание: Поскольку используется внешняя библиотека, все функции и процедуры, перечисленные в этом приложении, не могут быть использованы в окне **MapBasic** MapInfo Pro.

В этом разделе:

В этом разделе

О библиотеке XML.....	1197
Процедура MIXmlAttributeListDestroy().....	1197
Функция MIXmlDocumentCreate().....	1198
Процедура MIXmlDocumentDestroy().....	1198
Функция MIXmlDocumentGetNamespaces().....	1199
Функция MIXmlDocumentGetRootNode().....	1200
Функция MIXmlDocumentLoad().....	1201
Функция MIXmlDocumentLoadXML().....	1202
Функция MIXmlDocumentLoadXMLString().....	1203
Функция MIXmlDocumentSetProperty().....	1204
Функция MIXmlGetAttributeList().....	1205
Функция MIXmlGetChildList().....	1206
Функция MIXmlGetNextAttribute().....	1207
Функция MIXmlGetNextNode().....	1208
Процедура MIXmlNodeDestroy().....	1209
Функция MIXmlNodeGetAttributeValue().....	1209
Функция MIXmlNodeGetFirstChild().....	1210
Функция MIXmlNodeGetName().....	1211
Функция MIXmlNodeGetParent().....	1212
Функция MIXmlNodeGetText().....	1213
Функция MIXmlNodeGetValue().....	1214
Процедура MIXmlNodeListDestroy().....	1214
Процедура MIXmlSCDestroy().....	1215
Функция MIXmlSCGetLength().....	1216
Функция MIXmlSCGetNamespace().....	1216
Функция MIXmlSelectNodes().....	1217

Функция MIXmlSelectSingleNode().....	1218

О библиотеке XML

Эти библиотеки используют общие DEF-файлы: `XMLLib.DEF` и `XMLTypes.DEF`, которые находятся по адресу: *<Каталог, в котором установлен MapBasic>\Samples\MapBasic\INC*. Эти файлы следует включить в состав секции заголовков программ. Все функции, описанные в этом приложении, работают при наличии файла `GmlXlat.dll`, который устанавливается вместе с MapInfo Pro.

Они являются оболочками соответствующих методов интерфейсов и классов Microsoft XML. Набор интерфейсных классов включает в себя: `IXMLDOMNode`, `IXMLDOMNodeList`, `IXMLDOMNamedNodeMap`, `IXMLDOMSchemaCollection2` и `IXMLDOMDocument2`. Дополнительную информацию об использовании связанных классов и интерфейсов ищите в справочнике MSDN: *XML DOM Objects-Interfaces*.

Процедура `MIXmlAttributeListDestroy()`

Назначение

Удаляет объект `MIXmlNamedNodeMap` и освобождает память.

Синтаксис

```
MIXmlNodeListDestroy( ByVal hXMLNodeList As MIXmlNodeList )
```

hXMLAttributeList идентификатор удаляемого объекта `MIXmlNamedNodeMap`.

Описание

При вызове требуется вызвать эту функцию, для того чтобы освободить идентификатор `MIXmlNamedNodeMap`, для получения которого используется [функция `MIXmlGetAttributeList\(\)`](#), если такой идентификатор больше не используется.

См. также:

[Функция `MIXmlGetAttributeList\(\)`](#)

Функция MIXmlDocumentCreate()

Назначение

Создает объект MIXmlDocument и получает идентификатор этого объекта.

Синтаксис

```
MIXmlDocumentCreate ( ) As MIXmlDocument
```

Возвращаемая величина

Идентификатор объекта MIXmlDocument. Если запрос не будет выполнен, будет возвращено значение Null. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

MIXmlDocumentCreate() – создает и возвращает идентификатор объекта MIXmlDocument. Является самым старшим уровнем исходного кода XML.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlDocumentDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[MIXmlDocumentDestroy\(\)](#), процедура

Процедура MIXmlDocumentDestroy()

Назначение

Удаляет объект MIXmlDocument и освобождает память.

Синтаксис

```
MIXmlDocumentDestroy( ByVal hXMLDocument As MIXmlDocument )
```

hXMLDocument идентификатор MIXmlDocument удаляемого объекта.

Описание

При вызове требуется вызвать эту функцию, для того чтобы закрыть и освободить идентификатор MIXmlDocument, для получения которого используется [функция MIXmlDocumentCreate\(\)](#), если такой идентификатор больше не используется.

См. также:

[Функция MIXmlDocumentCreate\(\)](#)

Функция MIXmlDocumentGetNamespaces()

Назначение

Создает объект MIXMLSchemaCollection и получает идентификатор этого объекта.

Синтаксис

```
MIXmlDocumentGetNamespaces( ByVal hXMLDocument As MIXmlDocument )  
As MIXMLSchemaCollection
```

hXMLDocument – идентификатор объекта MIXmlDocument.

Возвращаемая величина

Идентификатор объекта MIXMLSchemaCollection при успешном выполнении, иначе NULL.

Описание

Этот метод создает объект MIXMLSchemaCollection.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlSCDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[функция MIXmlDocumentCreate\(\)](#), процедура [MIXmlSCDestroy](#)

Функция MIXmlDocumentGetRootNode()

Назначение

Будет получен самый старший элемент документа.

Синтаксис

```
MIXmlDocumentGetRootNode( ByVal hXMLDocument As MIXmlDocument )  
As MIXmlNode
```

hXMLDocument – идентификатор объекта MIXmlDocument.

Возвращаемая величина

Идентификатор объекта MIXmlNode, описывающий самый старший элемент документа, при успешном выполнении, иначе NULL.

Описание

MIXmlDocumentGetRootNode() возвращает идентификатор объекта MIXmlNode, описывающий самый старший элемента дерева документа XML. Будет возвращено значение NULL, если иерархии не существует.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlNodeDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[функция MIXmlDocumentCreate\(\)](#), процедура [MIXmlNodeDestroy](#)

Функция MIXmlDocumentLoad()

Назначение

Загружает XML-документ из указанного места.

Синтаксис

```
MIXmlDocumentLoad( ByVal hXMLDocument As MIXmlDocument,  
    ByVal strPath As String, pbParsingError As SmallInt,  
    ByVal bValidate As SmallInt, ByVal bResolveExternals As SmallInt )  
    As SmallInt
```

hXMLDocument – идентификатор объекта MIXmlDocument.

strPath – строка, содержащая маршрут/URL к XML-файлу.

pbParsingError – указывает на значение типа SmallInt, которое означает TRUE при успешной загрузке; FALSE если загрузка не выполнена.

bValidate – значение типа SmallInt, которое определяет требуется ли проверять синтаксис документа анализатором. Если TRUE (1), документ будет проверен анализатором. Если FALSE (0), будет проанализирован только правильно составленный XML-код.

bResolveExternals – короткая целая переменная (SmallInt), с помощью которой можно задать или отменить разрешение при разборе, независимо от проверки, внешних определителей, допустимости пространства имен, внешних подмножеств определителей типов документа (DTD) и ссылок на внешние источники. Если параметр *bResolveExternals* принимает значение TRUE (1), внешние определители будут исследоваться при разборе. При этом стандартные атрибуты и типы данных можно будет определять по элементам схемы, а для включения файлов использовать DTD. Этот параметр не зависит от выполнения проверки, о чем сигнализирует значение свойства *bValidate*. Если внешние параметры не могут быть разрешены при проверке, появится сообщение об ошибке при проверке. Если параметр *bResolveExternals* принимает значение FALSE (0), внешние параметры не будут исследованы, а проверка не будет выполняться.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Если URL не может быть определен, к нему нет доступа или он не ссылается на XML-документ, этот метод возвращает FALSE. Вызов функции **MIXmlDocumentLoad()** по отношению к существующему документу, моментально освобождает оглавление документа. При загрузке XML-документа из ресурса, загрузка должна выполняться асинхронно, иначе загрузка не будет успешной.

См. также:

MIXmlDocumentCreate() функция, **MIXmlDocumentLoadXML()** функция, **MIXmlDocumentLoadXMLString()** функция

Функция MIXmlDocumentLoadXML()

Назначение

Загрузить XML-документ, используя заданную строку.

Синтаксис

```
MIXmlDocumentLoadXML( ByVal hXMLDocument As MIXmlDocument,
    ByVal hContent As CString, pbParsingError As SmallInt,
    ByVal bValidate As SmallInt, ByVal bResolveExternals As SmallInt )
    As SmallInt
```

hXMLDocument – идентификатор объекта MIXmlDocument.

hContent – идентификатор строковой переменной CString со XML-строкой, которую необходимо загрузить в объект XML-документа. Эта строка может содержать либо весь XML-документ, либо правильно сформатированный фрагмент.

pbParsingError – указывает на значение типа SmallInt, которое означает TRUE (любое ненулевое значение) при успешной загрузке; FALSE (0) если загрузка не выполнена.

bValidate – значение типа SmallInt, которое определяет требуется ли проверять синтаксис документа анализатором. Если TRUE (1), документ будет проверен анализатором. Если FALSE (0), будет проанализирован только правильно составленный XML-код.

bResolveExternals – короткая целая переменная (SmallInt), с помощью которой можно задать или отменить разрешение при разборе, независимо от проверки, внешних определителей, допустимости пространства имен, внешних подмножеств определителей типов документа (DTD) и ссылок на внешние источники. Если параметр *bResolveExternals* принимает значение TRUE (1), внешние определители будут исследоваться при разборе. При этом стандартные

атрибуты и типы данных можно будет определять по элементам схемы, а для включения файлов использовать DTD. Этот параметр не зависит от выполнения проверки, о чем сигнализирует значение свойства *bValidate*. Если внешние параметры не могут быть разрешены при проверке, появится сообщение об ошибке при проверке. Если параметр *bResolveExternals* принимает значение FALSE (0), внешние параметры не будут исследованы, а проверка не будет выполняться.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Вызов функции **MIXmlDocumentLoadXML()** по отношению к существующему документу, моментально освобождает оглавление документа. Выполняется только в кодировках UTF-16 или UCS-2.

См. также:

[MIXmlDocumentCreate\(\)](#) функция, [MIXmlDocumentLoad\(\)](#) функция, [MIXmlDocumentLoadXMLString\(\)](#) функция

Функция MIXmlDocumentLoadXMLString()

Назначение

Загрузить XML-документ, используя заданную строку.

Синтаксис

```
MIXmlDocumentLoadXMLString( ByVal hXMLDocument As MIXmlDocument,
    ByVal strXML As String, pbParsingError As SmallInt,
    ByVal bValidate As SmallInt, ByVal bResolveExternals As SmallInt )
    As SmallInt
```

hXMLDocument – идентификатор объекта MIXmlDocument.

strXML – строковая переменная с XML-строкой, которую требуется загрузить в этот объект XML-документа. Эта строка может содержать либо весь XML-документ, либо правильно сформатированный фрагмент.

pbParsingError – указывает на значение типа SmallInt, которое означает TRUE (любое ненулевое значение) при успешной загрузке; FALSE (0) если загрузка не выполнена.

bValidate – значение типа SmallInt, которое определяет требуется ли проверять синтаксис документа анализатором. Если TRUE (1), документ будет проверен анализатором. Если FALSE (0), будет проанализирован только правильно составленный XML-код.

bResolveExternals – короткая целая переменная (SmallInt), с помощью которой можно задать или отменить разрешение при разборе, независимо от проверки, внешних определителей, допустимости пространства имен, внешних подмножеств определителей типов документа (DTD) и ссылок на внешние источники. Если параметр *bResolveExternals* принимает значение TRUE (1), внешние определители будут исследоваться при разборе. При этом стандартные атрибуты и типы данных можно будет определять по элементам схемы, а для включения файлов использовать DTD. Этот параметр не зависит от выполнения проверки, о чем сигнализирует значение свойства *bValidate*. Если внешние параметры не могут быть разрешены при проверке, появится сообщение об ошибке при проверке. Если параметр *bResolveExternals* принимает значение FALSE (0), внешние параметры не будут исследованы, а проверка не будет выполняться.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

Вызов функции **MIXmlDocumentLoadXMLString()** по отношению, к существующему документу, моментально освобождает оглавление документа. Выполняется только в кодировках UTF-16 или UCS-2.

См. также:

[MIXmlDocumentCreate\(\) функция](#), [MIXmlDocumentLoad\(\) функция](#),
[MIXmlDocumentLoadXML\(\) функция](#)

Функция MIXmlDocumentSetProperty()

Назначение

Присвоить свойства объекту MIXmlDocument.

Синтаксис

```
MIXmlDocumentSetProperty( ByVal hXMLDocument As MIXmlDocument,
    ByVal strPropertyName As String, ByVal strPropertyValue As String )
    As SmallInt
```

hXMLDocument – идентификатор объекта MIXmlDocument.

strPropertyName – строка с именем свойства, которое требуется задать. Список свойств, которые можно задать этим методом, приводится в библиотеке MSDN.

strPropertyValue – строка со значением заданного свойства. Список значений свойств, которые можно задать этим методом, приводится в библиотеке MSDN.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0.

Описание

Этот метод используется для задания свойств объекта MIXmlDocument. Существуют некоторые ограничения в том, какие именно свойства можно задавать, используя этот метод. За подробной информацией, обращайтесь к Microsoft MSDN library.

См. также:

[MIXmlDocumentCreate\(\)](#) функция, [MIXmlDocumentLoad\(\)](#) функция

Функция MIXmlGetAttributeList()

Назначение

Получить объект MIXmlNamedNodeMap с заданным узлом.

Синтаксис

```
MIXmlGetAttributeList( ByVal hXMLNode As MIXmlNode ) As MIXmlNamedNodeMap
```

hXMLNode – идентификатор объекта MIXmlNode.

Возвращаемая величина

Идентификатор объекта MIXmlNamedNodeMap с узлами, от которых можно получить атрибуты. Значение NULL возвращается для всех остальных типов узлов.

Описание

MIXmlGetAttributeList() – создает объект MIXmlNamedNodeMap и возвращает идентификатор объекта. В этом объекте могут присутствовать только узлы, атрибуты которых можно получить (узлы типа Element, Entity и Notation). Для всех остальных – возвращается значение Null. Для узлов, которые отвечают условиям, идентификатор объекта MIXmlNamedNodeMap всегда будет возвращаться; если элемент не атрибутирован, длина списка будет равной нулю. Подробная информация и список узлов, отвечающих условиям, приводится в библиотеке Microsoft MSDN.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlAttributeListDestroy\(\)](#), если возвращенный идентификатор больше не применяется.

См. также:

[функция MIXmlDocumentGetRootNode\(\)](#), [процедура MIXmlAttributeListDestroy](#)

Функция MIXmlGetChildList()

Назначение

Получить объект MIXmlNodeList со списком потомков заданного узла.

Синтаксис

```
MIXmlGetChildList( ByVal hXMLNode As MIXmlNode) As MIXmlNodeList
```

hXMLNode – идентификатор объекта MIXmlNode.

Возвращаемая величина

Идентификатор объекта MIXmlNodeList со списком потомков заданного узла, если функция будет выполнена успешно; иначе NULL.

Описание

MIXmlGetChildList() – используется для получения списка дочерних узлов – потомков заданному. Объект MIXmlNodeList будет возвращен даже, если у узла нет потомков. В этом случае длина списка будет равна 0. Значение зависит от типа узла. За дополнительной информацией, обращайтесь к Microsoft MSDN library.

При вызове требуется добавить идентификатор, для получения которого используется **процедура MIXmlNodeListDestroy()**, если такой идентификатор больше не применяется.

См. также:

MIXmlNodeListDestroy() procedure, **MIXmlSelectNodes()** функция

Функция MIXmlGetNextAttribute()

Назначение

Возвратить следующий узел коллекции.

Синтаксис

```
MIXmlGetNextAttribute( ByVal hXMLAttributeList As MIXmlNamedNodeMap )  
As MIXmlNode
```

hXMLAttributeList – идентификатор объекта MIXmlNamedNodeMap.

Возвращаемая величина

Если функция будет успешно выполнена, идентификатор объекта MIXmlNode, относящегося к следующему узлу коллекции; возвращается NULL, если следующего узла не существует.

Описание

Итератор начинает работать с узла, предшествующего первому в списке, поэтому первый вызов функции **MIXmlGetNextAttribute()** возвращает первый узел в списке. Функция возвращает NULL, если текущий узел последний в списке или в списке ничего нет.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется **процедура MIXmlNodeDestroy()**, если такой идентификатор больше не применяется.

См. также:

[функция MIXmlGetAttributeList\(\)](#), [процедура MIXmlNodeDestroy](#)

Функция MIXmlGetNextNode()

Назначение

Возвратить следующий узел коллекции.

Синтаксис

```
MIXmlGetNextNode( ByVal hXMLNodeList As MIXmlNodeList ) As MIXmlNode
```

hXMLNodeList – идентификатор объекта MIXmlNodeList.

Возвращаемая величина

Если функция будет успешно выполнена, идентификатор объекта *hXMLNodeList*, относящегося к следующему узлу коллекции; возвращается NULL, если следующего узла не существует.

Описание

Итератор начинает работать с узла, предшествующего первому в списке, поэтому первый вызов функции **MIXmlGetNextAttribute()** возвращает первый узел в списке. Функция возвращает NULL, если текущий узел последний в списке или в списке ничего нет.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется [процедура MIXmlNodeDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[MIXmlNodeDestroy\(\) procedure](#), [MIXmlSelectNodes\(\) функция](#), [MIXmlGetChildList\(\) функция](#)

Процедура MIXmlNodeDestroy()

Назначение

Удалить объект MIXmlNode и освободить память.

Синтаксис

```
MIXmlNodeDestroy( ByVal hXMLNode As MIXmlNode )
```

hXMLNode идентификатор MIXmlNode удаляемого объекта.

Описание

При вызове требуется вызвать эту функцию, чтобы освободить идентификатор MIXmlNode, для получения которого используется [функция MIXmlDocumentGetRootNode\(\)](#), если такой идентификатор больше не используется.

См. также:

[MIXmlDocumentDestroy\(\)](#), процедура

Функция MIXmlNodeGetAttributeValue()

Назначение

Получить текст, ассоциированный с заданным именем.

Синтаксис

```
MIXmlNodeGetAttributeValue( ByVal hXMLNode As MIXmlNode,  
    ByVal strAttributeName As String, pValue As String,  
    ByVal nLen As Integer ) As SmallInt
```

hXMLNode – идентификатор объекта MIXmlNode.

strAttributeName – строка с заданным именем атрибута.

pValue – ссылка на строковую переменную, в которую будет помещено значение в узле с заданным атрибутом.

nObjectLen – размер буфера, на который ссылается *pObject*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0.

Описание

MIXmlNodeGetAttributeValue() сначала находит объект MIXmlNamedNodeMap с заданным узлом, *hXMLNode*. Как заявлено в разделе, в котором описана [функция MIXmlGetAttributeList\(\)](#), такой объект содержит только узлы с возвращаемыми атрибутами (узлы типа Element, Entity и Notation). Если существует корректный объект MIXmlNamedNodeMap, заданное имя можно найти в этом объекте, значение в этом узле будет помещено в *pValue*.

См. также:

[MIXmlGetAttributeList\(\)](#) функция, [MIXmlNodeGetValue\(\)](#) функция

Функция MIXmlNodeGetFirstChild()

Назначение

Возвратить первого потомка заданного узла.

Синтаксис

```
MIXmlNodeGetFirstChild( ByVal hXMLNode As MIXmlNode ) As MIXmlNode
```

hXMLNode – идентификатор объекта MIXmlNode.

Возвращаемая величина

Если функция будет выполнена успешно, идентификатор объекта MIXmlNode, который является первым потомком заданного узла *hXMLNode*; иначе NULL.

Описание

MIXmlNodeGetFirstChild() – получает идентификатор объекта MIXmlNode, первого потомка заданного узла. Будет возвращено значение NULL, если потомка не существует.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется **процедура MIXmlNodeDestroy()**, если такой идентификатор больше не применяется.

См. также:

MIXmlNodeDestroy(), **процедура**, **MIXmlNodeGetParent()** **функция**

Функция MIXmlNodeGetName()

Назначение

Получить имя заданного узла.

Синтаксис

```
MIXmlNodeGetName( ByVal hXMLNode As MIXmlNode, pName As String,  
ByVal nLen As Integer ) As SmallInt
```

hXMLNode – идентификатор объекта MIXmlNode.

pName – ссылка на строковую переменную в которой будет храниться имя узла, которое может меняться в зависимости от типа узла.

nObjectLen – размер буфера, на который ссылается *pObject*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0.

Описание

Эта функция используется для того, чтобы получить имя заданного узла. Имя узла, это определенное название элемента, атрибута или ссылки на объект. Значение имени узла меняется в зависимости от типа узла.

См. также:

[MIXmlDocumentGetRootNode\(\)](#) функция, [MIXmlNodeGetText\(\)](#) функция, [MIXmlNodeGetValue\(\)](#) функция

Функция MIXmlNodeGetParent()

Назначение

Возвратить предка заданного узла.

Синтаксис

```
MIXmlNodeGetParent( ByVal hXMLNode As MIXmlNode ) As MIXmlNode
```

hXMLNode – идентификатор объекта MIXmlNode.

Возвращаемая величина

Если функция будет выполнена успешно, идентификатор объекта MIXmlNode, который является предком заданного узла *hXMLNode*; иначе NULL.

Описание

MIXmlNodegetParent() – получает идентификатор объекта MIXmlNode, предка заданного узла. Будет возвращено значение NULL, если предка не существует.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlNodeDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[MIXmlNodeDestroy\(\)](#), процедура

Функция MIXmlNodeGetText()

Назначение

Получить текстовое содержимое заданного узла или сцепленный текст, относящийся к узлу и его потомкам.

Синтаксис

```
MIXmlNodeGetText( ByVal hXMLNode As MIXmlNode, pText As String,  
ByVal nLen As Integer ) As SmallInt
```

hXMLNode – идентификатор объекта MIXmlNode.

pText – ссылка на строковую переменную, в которой будет храниться текстовое содержимое заданного узла и его потомков. Значение меняется в зависимости от типа узла.

nObjectLen – размер буфера, на который ссылается *pObject*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0.

Описание

MIXmlNodegettext() – используется для того, чтобы получить текст в заданном узле. Значение меняется в зависимости от типа узла. Подробности и описание более точной работы с текстом в XML-документе, смотрите в библиотеке Microsoft MSDN.

См. также:

[MIXmlDocumentGetRootNode\(\)](#) функция, [MIXmlNodeGetName\(\)](#) функция,
[MIXmlNodeGetValue\(\)](#) функция

Функция MIXmlNodeGetValue()

Назначение

Получить текст ассоциированный с заданным узлом.

Синтаксис

```
MIXmlNodeGetValue( ByVal hXMLNode As MIXmlNode, pValue As String,  
ByVal nLen As Integer ) As SmallInt
```

hXMLNode – идентификатор объекта MIXmlNode.

pValue – ссылка на строковую переменную в которой будет храниться имя узла, которое может меняться в зависимости от типа узла.

nObjectLen – размер буфера, на который ссылается *pObject*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0.

Описание

Эта функция используется для того, чтобы получить значение в заданном узле. Значение в узле меняется в зависимости от типа узла.

См. также:

[MIXmlDocumentGetRootNode\(\)](#) функция, [MIXmlNodeGetName\(\)](#) функция, [MIXmlNodeGetText\(\)](#) функция

Процедура MIXmlNodeListDestroy()

Назначение

Удаляет объект MIXmlNodeList и освобождает память.

Синтаксис

```
MIXmlNodeListDestroy( ByVal hXMLNodeList As MIXmlNodeList )
```

hXMLNodeList идентификатор MIXmlNodeList удаляемого объекта.

Описание

Используйте функцию **MIXmlNodeListDestroy**(), чтобы освободить идентификатор MIXmlNodeList, полученный, при вызове других функций, например: [функция MIXmlSelectNodes\(\)](#) и [функция MIXmlGetChildList\(\)](#), если идентификатор MIXmlNodeList больше не используется.

См. также:

[MIXmlDocumentDestroy\(\)](#), процедура, [MIXmlGetChildList\(\)](#) функция, [MIXmlSelectNodes\(\)](#) функция

Процедура MIXmlSCDestroy()

Назначение

Удалить объект MIXMLSchemaCollection и освободить память.

Синтаксис

```
MIXmlSCDestroy( ByVal hXMLSchemaCollection As MIXMLSchemaCollection )
```

hXMLSchemaCollection идентификатор MIXMLSchemaCollection удаляемого объекта.

Описание

Используйте функцию **MIXmlSCDestroy**(), чтобы освободить идентификатор MIXMLSchemaCollection, полученный вызовом других функций, например, [функции MIXmlDocumentGetNamespaces\(\)](#), если идентификатор больше не используется.

См. также:

[Функция MIXmlDocumentGetNamespaces\(\)](#)

Функция MIXmlSCGetLength()

Назначение

Получить пространство имен коллекции.

Синтаксис

```
MIXmlSCGetLength( ByVal hXMLSchemaCollection As MIXMLSchemaCollection )  
As Integer
```

hXMLSchemaCollection – идентификатор объекта MIXMLSchemaCollection.

Возвращаемая величина

Число пространств имен коллекции.

Описание

MIXmlSCGetLength() возвращает имена объектов, содержащихся в коллекции.

См. также:

[MIXmlDocumentGetNamespaces\(\)](#) функция, [MIXmlSCGetNamespace\(\)](#) функция

Функция MIXmlSCGetNamespace()

Назначение

Получить пространство имен по заданному индексу.

Синтаксис

```
MIXmlISCGetNamespace ( ByVal hXMLSchemaCollection As MIXMLSchemaCollection,
    ByVal index As Integer, pNamespace As String, ByVal nLen As Integer )
    As SmallInt
```

hXMLSchemaCollection – идентификатор объекта MIXMLSchemaCollection.

index – целая переменная индекса от 0 до значения счетчика минус 1.

pNamespace – ссылка на строку, в которой будет помещено имя множества имен.

nLen – размер буфера, на который ссылается *pNamespace*.

Возвращаемая величина

Ненулевое значение при успешном выполнении, иначе – 0. Для того чтобы определить причину ошибки, обычно используется [функция MIGetErrorMessage\(\)](#).

Описание

MIXmlISCGetNamespace() позволяет итеративно перебрать коллекцию и определить её содержимое.

См. также:

[MIXmlDocumentGetNamespaces\(\)](#) функция, [MIXmlISCGetLength\(\)](#) функция

Функция MIXmlSelectNodes()

Назначение

Сравнивает шаблон с контекстом узла и возвращает список подходящих узлов в объект MIXmlNodeList.

Синтаксис

```
MIXmlSelectNodes ( ByVal hXMLNode As MIXmlNode, ByVal strPattern As String
    ) As MIXmlNodeList
```

hXMLNode – идентификатор объекта MIXmlNode.

strPattern – строковая переменная с выражением XPath.

Возвращаемая величина

Идентификатор объекта MIXmlNodeList. Это коллекция узлов, выбранных операцией сравнения с шаблоном. Если ни одного узла не выбрано, коллекция будет пустой. При ошибке, возвращается NULL.

Описание

MIXmlSelectNodes() – используется для того, чтобы передать коллекцию, совпадающих с шаблоном узлов после выполнения операции сравнения, в объект MIXmlNodeList. Функция **MIXmlSelectNodes()** работает также как [функция MIXmlSelectSingleNode\(\)](#), но возвращает список совпадающих узлов, а не первый совпавший узел.

При вызове требуется добавить возвращенный идентификатор, для получения которого используется [процедура MIXmlNodeListDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[MIXmlNodeListDestroy\(\)](#), [процедура](#), [MIXmlSelectSingleNode\(\)](#) [функция](#), [MIXmlGetChildList\(\)](#) [функция](#)

Функция MIXmlSelectSingleNode()

Назначение

Сравнивает шаблон с контекстом узла и возвращает первый совпавший узел в объект MIXmlNode.

Синтаксис

```
MIXmlSelectSingleNode( ByVal hXMLNode As MIXmlNode,
                        ByVal strPattern As String ) As MIXmlNode
```

hXMLNode – идентификатор объекта MIXmlNode.

strPattern – строковая переменная с выражением XPath.

Возвращаемая величина

Идентификатор объекта MIXmlNode. Возвращает первый совпадающий с шаблоном узел. Если совпадений не найдено, будет возвращено значение NULL.

Описание

MIXmlSelectSingleNode() получает идентификатор объекта MIXmlNode, который является первым узлом, совпавшим с заданным шаблоном. Будет возвращено значение NULL, если потомка не существует. Функция **MIXmlSelectSingleNode()** работает также как [функция MIXmlSelectNodes\(\)](#), но возвращает первый совпавший узел, а не список совпавших с заданным шаблоном узлов.

При вызове требуется добавить идентификатор, для получения которого используется [процедура MIXmlNodeDestroy\(\)](#), если такой идентификатор больше не применяется.

См. также:

[MIXmlNodeDestroy\(\)](#), [процедура MIXmlSelectNodes\(\)](#) [функция](#)

С - Таблица символов

В компьютере каждому символу клавиатуры сопоставлен некоторый числовой код. Например, английскому символу "А" на клавиатуре соответствует код символа 65. Набором символов (или кодировкой символов) называется совокупность символов, доступных в компьютере и сопоставленных им числовых кодов.

В разных странах могут использоваться различные наборы символов. Например, Windows версии для Северной Америки и Западной Европы использует код 176 для представления знака градуса (°), тогда как Windows другой языковой версии может использовать этот код для другого символа. Информацию о работе с международными наборами символов, см. *Международные наборы символов в Руководстве пользователя MapBasic*.

Однако, если файл включает интернациональные символы и требуется его перенести в другую операционную среду, которая использует другую кодировку, не обойтись без предложения CharSet для сохранения целостности данных. Предложение CharSet показывает, какой набор символов использовался в момент создания таблицы.

См. также:
[предложение CharSet](#)

В этом разделе:

В этом разделе

Таблица символов.....	1221

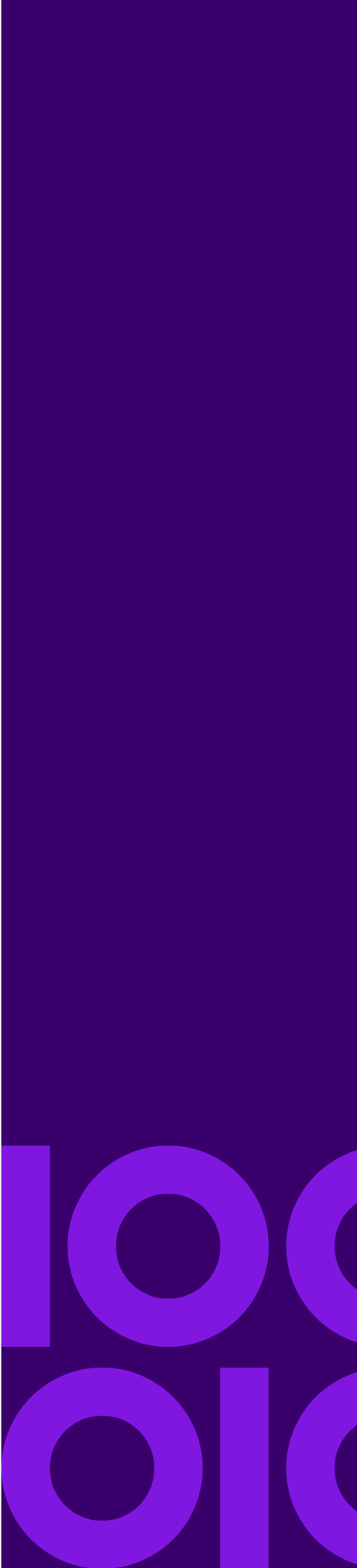


Таблица символов

В следующей таблице собраны символы набора Windows Latin 1. Символы с 32 (пробел) по 126 (тильда) повторяются в других наборах символов. Среди других полезных символов можно упомянуть номер 9 (табуляция), 10 (новая строка), 12 (новая страница) и 13 (перевод каретки).

32		64 @	96 `	128 ■	160	192 À	224 à
33 !	65 A	97 a	129 ■	161 ì	193 Á	225 á	
34 "	66 B	98 b	130 ■	162 é	194 Â	226 â	
35 #	67 C	99 c	131 ■	163 Ê	195 Ã	227 ã	
36 \$	68 D	100 d	132 ■	164 Ë	196 Ä	228 ä	
37 %	69 E	101 e	133 ■	165 ¥	197 Å	229 å	
38 &	70 F	102 f	134 ■	166 Ì	198 Æ	230 æ	
39 '	71 G	103 g	135 ■	167 Š	199 Ç	231 ç	
40 (	72 H	104 h	136 ■	168 ´	200 È	232 è	
41)	73 I	105 i	137 ■	169 ©	201 É	233 é	
42 *	74 J	106 j	138 ■	170 Ñ	202 Ê	234 ê	
43 +	75 K	107 k	139 ■	171 «	203 Ë	235 ë	
44 ,	76 L	108 l	140 ■	172 ¬	204 Ì	236 ì	
45 -	77 M	109 m	141 ■	173 ·	205 Í	237 í	
46 .	78 N	110 n	142 ■	174 ®	206 Î	238 î	
47 /	79 O	111 o	143 ■	175 ¯	207 Ĩ	239 ĩ	
48 0	80 P	112 p	144 ■	176 °	208 Ð	240 ð	
49 1	81 Q	113 q	145 ´	177 ±	209 Ñ	241 ñ	
50 2	82 R	114 r	146 ´	178 º	210 Ò	242 ò	
51 3	83 S	115 s	147 ■	179 »	211 Ó	243 ó	
52 4	84 T	116 t	148 ■	180 ´	212 Ô	244 ô	
53 5	85 U	117 u	149 ■	181 µ	213 Õ	245 õ	
54 6	86 V	118 v	150 ■	182 ¶	214 Ö	246 ö	
55 7	87 W	119 w	151 ■	183 ·	215 ×	247 ÷	
56 8	88 X	120 x	152 ■	184 ¸	216 Ø	248 ø	
57 9	89 Y	121 y	153 ■	185 ´	217 Ù	249 ù	
58 :	90 Z	122 z	154 ■	186 ■	218 Ú	250 ú	
59 ;	91 [	123 {	155 ■	187 >	219 Û	251 û	
60 <	92 \	124	156 ■	188 ¼	220 Ü	252 ü	
61 =	93]	125 }	157 ■	189 ½	221 Ý	253 ý	
62 >	94 ^	126 ~	158 ■	190 ¾	222 Þ	254 þ	
63 ?	95 _	127	159 ■	191 ¸	223 ß	255 ÿ	

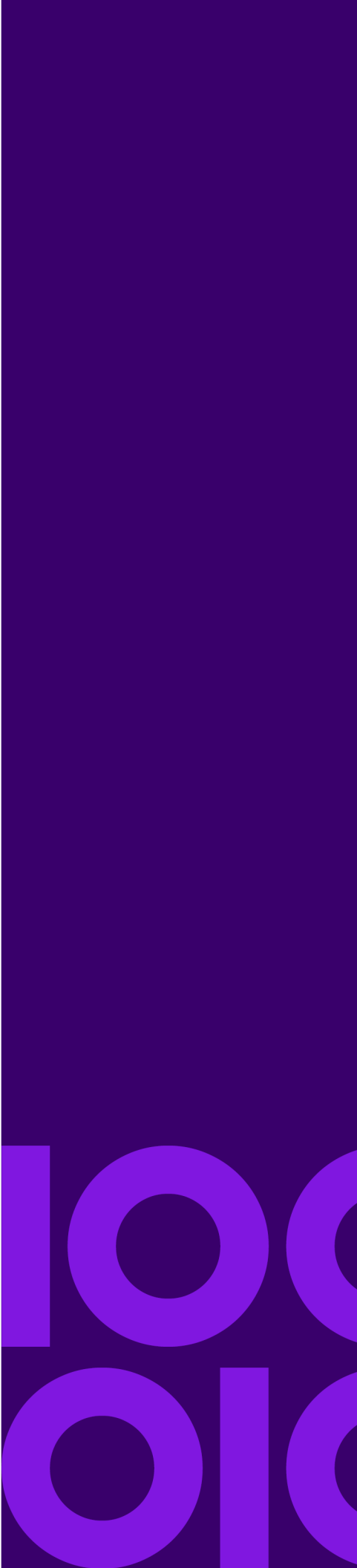
D - Сведения об операторах

Операторы производят некоторое действие над одним или несколькими значениями. Операторы можно классифицировать либо по типу значений, над которыми выполняются операции, либо по типу значения получаемого результата.

В этом разделе:

В этом разделе

Числовые операторы.....	1223
Операторы сравнения.....	1224
Логические операторы.....	1226
Географические операторы.....	1226
Автоматическое преобразование типов.....	1229
Замещающие символы.....	1230



Числовые операторы

Следующие операторы выполняются с двумя числовыми значениями, а в результате должно быть получено числовое значение.

Оператор	Действие	Пример:
+	сложение	<code>a + b</code>
-	вычитание	<code>a - b</code>
*	умножение	<code>a * b</code>
/	деление	<code>a / b</code>
\	целочисленное деление (остаток отброшен)	<code>a \ b</code>
Mod	остаток целочисленного деления	<code>a Mod b</code>
^	Возведение в степень	<code>a ^ b</code>

Два оператора из перечисленных выше могут быть использованы и в другом значении. Знак сложения (+) используется для конкатенации (объединения) двух строк в третью. Знак минус (–) вместе с единственным числом используется в качестве оператора отрицания, результатом выполнения будет числовое значение. Знак амперсанда (&) также используется для конкатенации строк.

Оператор	Действие	Пример:
-	численное отрицание	<code>- a</code>
+	конкатенация строк	<code>a + b</code>
&	конкатенация строк	<code>a & b</code>

Операторы сравнения

С помощью операторов сравнения две величины одного типа сопоставляются друг другу, результатом выполнения оператора будет логическая величина TRUE или FALSE. Хотя операторы сравнения нельзя использовать для сопоставления числовых и нечисловых данных (например, со строковыми выражениями), допустимо сравнение данных целых, коротких целых и вещественных типов. Операторы сравнения часто используются в выражениях с условием, например, `If...Then`.

Оператор	Возвращается TRUE если:	Пример:
=	a равно b	a = b
<>	a не равно b	a <> b
<	a меньше b	a < b
>	a больше b	a > b
<=	a меньше или равно b	a <= b
>=	a больше или равно b	a >= b

Оператор сравнения **Between...And...** позволяет проверять, попадает ли значение в заданный диапазон. Этот оператор является инклюзивным (например, `X >= 500 And X <= 600`). Он работает со строками и данными типа даты, наряду с числовыми данными, и может быть использован внутри параметра **WHERE** в SQL-запросе. В следующем примере в операторе **If...Then** используется сравнение **Between...And...**:

```
If x Between 0 And 100 Then
    Note "Data within range."
Else
    Note "Data out of range."
End If
```

Логические операторы

Логические операторы выполняются над логическими значениями, а результатом их выполнения является логическая величина либо TRUE, либо FALSE:

Оператор	Возвращается TRUE если:	Пример:
And	оба операнда истинны	a And b
Or	значение одного из операндов – истина	a Or b
Not	значение операнда FALSE	Not a

Географические операторы

Географические операторы выполняются над объектами, а результатом их выполнения является логическая величина либо TRUE, либо FALSE:

Оператор	Возвращается TRUE если:	Пример:
Contains	центроид второго объекта находится внутри первого объекта	<code>objectA Contains objectB</code>
Contains Part	Объект A содержит некоторую часть объекта B	<code>objectA Contains Part objectB</code>
Contains Entire	Объект A содержит весь объект B	<code>objectA Contains Entire objectB</code>
Within	центроид первого объекта внутри второго объекта	<code>objectA Within objectB</code>
Partly Within	часть первого объекта помещается внутри второго	<code>objectA Partly Within objectB</code>
Entirely Within	первый объект полностью помещается внутри второго	<code>objectA Entirely Within objectB</code>
Intersects	Два объекта пересекаются хотя бы в одной точке	<code>objectA Intersects objectB</code>

Старшинство выполнения операторов

Специальным типом операторов являются скобки, в которые можно заключать выражения внутри выражений. С помощью скобок можно изменить порядок выполнения операторов. В таблице ниже приведен порядок выполнения операторов MapBasic. Операторы из одной строки имеют одинаковый порядок выполнения. Операторы с более высоким приоритетом

выполняются ранее других. Операторы имеющие одинаковый приоритет выполняются слева направо (кроме операторов возведения в степень, который выполняется справа налево).

(в последнюю очередь)	Оператор MapBasic
(В первую очередь)	скобки
	Возведение в степень
	отрицательный знак
	умножение, деление, Mod, целочисленное деление,
	сложение, вычитание
	Географические операторы
	операторы сравнения операторы Like
	Not
	And
(в последнюю очередь)	или

Например, результатом выполнения выражения $3 + 4 * 2$ будет 11 (умножение выполняется до сложения). Результатом выполнения слегка измененного выражения $(3 + 4) * 2$ будет 14 (сложение в скобках выполняется в первую очередь выполняется в первую очередь). Если есть сомнения, используйте скобки.

Автоматическое преобразование типов

Если создать выражение обрабатывающее данные разных типов, то, для того чтобы получить осмысленный результат, MapInfo Pro будет автоматически преобразовывать выполняемые типы данных. Например, если программа вычитает дату из другой даты, MapBasic вычислит целое значение (определяющее количество дней прошедшее между двумя датами).

В таблице ниже приведен список правил, по которым MapBasic'ом выполняется автоматическое преобразование типов данных. В этой таблице слово "*Целое*" означает целое значение, которое может представлять собой целую переменную, короткую целую переменную или целую константу. Слово "*Число*" обозначает числовое выражение, которое не обязательно является целочисленным.

Оператор	Комбинация операндов	Результат
+	Дата + Число	Дата типа Date
	Число + Дата	Дата типа Date
	Целое + Целое	Целое число типа Integer.
	Число + Число	Вещественное число типа Float.
	Другое + Другое	Строка
-	Дата – Число	Дата типа Date
	Дата – Дата	Целое число типа Integer.
	Целое – Целое	Целое число типа Integer.
	Число – Число	Вещественное число типа Float.

Оператор	Комбинация операндов	Результат
*	Целое * Целое	Целое число типа Integer.
	Число * Число	Вещественное число типа Float.
/	Число / Число	Вещественное число типа Float.
\	Число \ Число	Целое число типа Integer.
MOD	Число MOD Число	Целое число типа Integer.
^	Число ^ Число	Вещественное число типа Float.

Замещающие символы

Оператор LIKE использует символы процента (%) и подчеркивания (_) специальным образом.

Замещающий символ	Действие	Пример
_ (символ подчеркивания)	соответствует одному и только одному символу	<pre>select * from table where Name Like "New Yor_" into Selection</pre>
%	соответствует нескольким символам или ни одному	<pre>select * from table where Name Like "New%" into Selection</pre>

Е - Файл определителей MapBasic

MapBasic включает в себя Файл определителей MapBasic (`mapbasic.def`), в котором перечислены по умолчанию полезные при программировании определения и значения. Для того чтобы операторы и функции MapBasic работали должным образом, перед началом работы добавьте оператор Include в файл `mapbasic.def`.

```
Include "mapbasic.def"
```

Коды, заданные в `mapbasic.def`, нельзя ввести в окно **MapBasic**. Файл `mapbasic.def` содержит множество операторов Define, включая операторы для значений TRUE, FALSE, а также часто используемых цветов (например, BLACK – черный; WHITE – белый; RED – красный; GREEN – зеленый; BLUE – синий; CYAN – голубой; MAGENTA – розовый; YELLOW – желтый.) Каждый оператор Define устанавливает код с определенным значением, например, код ЧЕРНЫЙ имеет числовое значение 0 (ноль). Задавая команды команды в окне **MapBasic**, необходимо использовать актуальные значения, а не имя кода (например, использовать 0 вместо ЧЕРНЫЙ).

Более подробные сведения об анимированных слоях можно найти в *Руководстве пользователя MapBasic*.

В этом разделе:

В этом разделе

Файлы MAPBASIC.DEF.....1232

Файлы MAPBASIC.DEF

Ниже приводится содержимое файла MAPBASIC.DEF, расположенного в каталоге MapBasic.

```
'=====
' MapInfo Pro 2019 - System defines
'-----
' This file contains defines useful when programming in the MapBasic
' language. There are three versions of this file:
'     MAPBASIC.DEF - MapBasic syntax
'     MAPBASIC.BAS - Visual Basic syntax
'     MAPBASIC.H   - C/C++ syntax
'-----
' The defines in this file are organized into the following sections:
'     General Purpose defines:
'         macros, logical constants, angle conversion, colors, string
length
'         BrowserInfo() defines
'         ButtonPadInfo() defines
'         ColumnInfo() and column type defines
'         CommandInfo() and task switch defines
'         DateWindow() defines
'         FileAttr() and file access mode defines
'         GetFolderPath$() defines
'         GetPreference() defines
'         GetPreferencePath$() defines
'         IntersectNodes() parameters
'         LabelInfo() defines
'         GroupLayerInfo() defines
'         LayerListInfo() defines
'         LayerInfo(), display mode, label property, layer type, hotlink
defines
'         LayoutInfo(), LayoutItemInfo() and LayoutPageItemInfo() defines
'         LegendInfo() and legend orientation defines
'         LegendFrameInfo() and frame type defines
'         LegendTextFrameInfo() defines
'         LegendStyleInfo() defines
'         LibraryServiceInfo() defines
'         LocateFile$() defines
'         Map3DInfo() defines
'         MapperInfo(), display mode, calculation type, and clip type defines
'         MenuItemInfoByID() and MenuItemInfoByHandler() defines
'         ObjectGeography() defines
'         ObjectInfo() and object type defines
'         PrismMapInfo() defines
'         SearchInfo() defines
'         SelectionInfo() defines
'         Server statement and function defines
```



```

' SessionInfo() defines
' Set Next Document Style defines
' StringCompare() return values
' StyleAttr() defines
' SystemInfo(), platform, and version defines
' TableInfo() and table type defines
' ThemeInfo() defines
' ThemeItemInfo() defines
' ThemeExpressionInfo() defines
' RangeThemeInfo() defines
' RangeThemeItemInfo() defines
' IndividualThemeInfo() defines
' IndividualThemeItemInfo() defines
' DotDensityThemeInfo() defines
' GraduatedThemeInfo() defines
' BarChartThemeInfo() defines
' BarChartThemeItemInfo() defines
' PieChartThemeInfo() defines
' PieChartThemeItemInfo() defines
' WindowInfo(), window type and state, and print orientation defines
' Abbreviated list of error codes
' Backward Compatibility defines
' =====
' MAPBASIC.DEF is converted into MAPBASIC.H by doing the following:
' - concatenate MAPBASIC.DEF and MENU.DEF into MAPBASIC.H
' - search & replace "'" at begining of a line with "/"
' - search & replace "Define" at begining of a line with "#define"
' - delete the following sections:
'     * General Purpose defines:
'         Macros, Logical Constants, Angle Conversions
'     * Abbreviated list of error codes
'     * Backward Compatibility defines
'     * Menu constants whose names have changed
'     * Obsolete menu items
' =====
' MAPBASIC.DEF is converted into MAPBASIC.BAS by doing the following:
' - concatenate MAPBASIC.DEF and MENU.DEF into MAPBASIC.BAS
' - search & replace "Define <name>" with "Global Const <name> ="
'   e.g. "<Define {[!-z]} + {[!-z]}>" with "Global Const \0 = \1" with
Brief
' - delete the following sections:
'     * General Purpose defines:
'         Macros, Logical Constants, Angle Conversions
'     * Abbreviated list of error codes
'     * Backward Compatibility defines
'     * Menu constants whose names have changed
'     * Obsolete menu items
' =====
' =====
' General Purpose defines
' =====
' -----

```

```

' Macros
'-----
Define CLS                                Print Chr$(12)

'-----
' Logical constants
'-----
Define TRUE                                1
Define FALSE                               0

'-----
' Angle conversion
'-----
Define DEG_2_RAD                           0.01745329252
Define RAD_2_DEG                           57.29577951

'-----
' Time conversion
'-----
Define SECONDS_PER_DAY                     86400

'-----
' Colors
'-----
Define BLACK                                0
Define WHITE                               16777215
Define RED                                 16711680
Define GREEN                               65280
Define BLUE                               255
Define CYAN                               65535
Define MAGENTA                             16711935
Define YELLOW                             16776960

'-----
'Maximum length for character string
'-----
Define MAX_STRING_LENGTH                   32767

'=====
' BrowserInfo() defines
'=====
Define BROWSER_INFO_NROWS                  1
Define BROWSER_INFO_NCOLS                  2
Define BROWSER_INFO_CURRENT_ROW            3
Define BROWSER_INFO_CURRENT_COLUMN         4
Define BROWSER_INFO_CURRENT_CELL_VALUE     5

'=====
' ButtonPadInfo() defines
'=====
Define BTNPAD_INFO_FLOATING                1
Define BTNPAD_INFO_WIDTH                   2
Define BTNPAD_INFO_NBTNS                   3

```

```

Define BTNPAD_INFO_X                4
Define BTNPAD_INFO_Y                5
Define BTNPAD_INFO_WINID            6
Define BTNPAD_INFO_DOCK_POSITION    7

'=====
' New as per MI Pro 10.5.
' Codes returned from ButtonPadInfo() when 'BNTPAD_INFO_DOCK_POSITION'
code
' is used to inquiry about the tool bar position
'=====
Define BBNTPAD_INFO_DOCK_NONE        0
Define BTNPAD_INFO_DOCK_LEFT         1
Define BTNPAD_INFO_DOCK_TOP          2
Define BTNPAD_INFO_DOCK_RIGHT        3
Define BTNPAD_INFO_DOCK_BOTTOM       4

'=====
' ColumnInfo() defines
'=====
Define COL_INFO_NAME                 1
Define COL_INFO_NUM                  2
Define COL_INFO_TYPE                 3
Define COL_INFO_WIDTH                4
Define COL_INFO_DECPLACES            5
Define COL_INFO_INDEXED              6
Define COL_INFO_EDITABLE             7

'-----
' Column type defines, returned by ColumnInfo() for COL_INFO_TYPE
'-----
Define COL_TYPE_CHAR                 1
Define COL_TYPE_DECIMAL              2
Define COL_TYPE_INTEGER              3
Define COL_TYPE_SMALLINT             4
Define COL_TYPE_DATE                 5
Define COL_TYPE_LOGICAL              6
Define COL_TYPE_GRAPHIC              7
Define COL_TYPE_FLOAT               8
Define COL_TYPE_TIME                 37
Define COL_TYPE_DATETIME             38
Define COL_TYPE_LARGEINT             39

'=====
' CommandInfo() defines
'=====
Define CMD_INFO_X                    1
Define CMD_INFO_Y                    2
Define CMD_INFO_SHIFT                3
Define CMD_INFO_CTRL                 4
Define CMD_INFO_X2                   5
Define CMD_INFO_Y2                   6
Define CMD_INFO_TOOLBTN              7

```

```

Define CMD_INFO_MENUITEM      8
Define CMD_INFO_WIN           1
Define CMD_INFO_SELTYPE       1
Define CMD_INFO_ROWID         2
Define CMD_INFO_INTERRUPT     3
Define CMD_INFO_STATUS        1
Define CMD_INFO_MSG           1000
Define CMD_INFO_DLG_OK        1
Define CMD_INFO_DLG_DBL       1
Define CMD_INFO_FIND_RC       3
Define CMD_INFO_FIND_ROWID    4
Define CMD_INFO_XCMD           1
Define CMD_INFO_CUSTOM_OBJ    1
Define CMD_INFO_TASK_SWITCH   1
Define CMD_INFO_EDIT_TABLE    1
Define CMD_INFO_EDIT_STATUS   2
Define CMD_INFO_EDIT_ASK      1
Define CMD_INFO_EDIT_SAVE     2
Define CMD_INFO_EDIT_DISCARD  3
Define CMD_INFO_HL_WINDOW_ID  17
Define CMD_INFO_HL_TABLE_NAME 18
Define CMD_INFO_HL_ROWID      19
Define CMD_INFO_HL_LAYER_ID   20
Define CMD_INFO_HL_FILE_NAME  21

'-----
' Task Switches, returned by CommandInfo() for CMD_INFO_TASK_SWITCH
'-----

Define SWITCHING_OUT_OF_MAPINFO 0
Define SWITCHING_INT0_MAPINFO 1

'=====
' DateWindow() defines
'=====

Define DATE_WIN_SESSION 1
Define DATE_WIN_CURPROG 2

'=====
' FileAttr() defines
'=====

Define FILE_ATTR_MODE 1
Define FILE_ATTR_FILESIZE 2

'-----
' File Access Modes, returned by FileAttr() for FILE_ATTR_MODE
'-----

Define MODE_INPUT 0
Define MODE_OUTPUT 1
Define MODE_APPEND 2
Define MODE_RANDOM 3
Define MODE_BINARY 4

'=====

```

```

' GetFolderPath$() defines
'=====

Define FOLDER_MI_APPDATA -1
Define FOLDER_MI_LOCAL_APPDATA -2
Define FOLDER_MI_PREFERENCE -3
Define FOLDER_MI_COMMON_APPDATA -4
Define FOLDER_APPDATA 26
Define FOLDER_LOCAL_APPDATA 28
Define FOLDER_COMMON_APPDATA 35
Define FOLDER_COMMON_DOCS 46
Define FOLDER_MYDOCS 5
Define FOLDER_MYPICS 39

'=====
' GetPreferencePath$(), GetCurrentPath$(), and Set Path defines
'=====

Define PREFERENCE_PATH_TABLE 0
Define PREFERENCE_PATH_WORKSPACE 1
Define PREFERENCE_PATH_MBX 2
Define PREFERENCE_PATH_IMPORT 3
Define PREFERENCE_PATH_SQLQUERY 4
Define PREFERENCE_PATH_THEMETHEMPLATE 5
Define PREFERENCE_PATH_MIQUERY 6
Define PREFERENCE_PATH_NEWGRID 7
Define PREFERENCE_PATH_CRYSTAL 8
Define PREFERENCE_PATH_GRAPHSSUPPORT 9
Define PREFERENCE_PATH_REMOTETABLE 10
Define PREFERENCE_PATH_SHAPEFILE 11
Define PREFERENCE_PATH_WFSTABLE 12
Define PREFERENCE_PATH_WMSTABLE 13
Define PREFERENCE_PATH_LAYOUTIMAGES 14

'=====
' IntersectNodes() defines
'=====

Define INCL_CROSSINGS 1
Define INCL_COMMON 6
Define INCL_ALL 7

'=====
' LabelInfo() defines
'=====

Define LABEL_INFO_OBJECT 1
Define LABEL_INFO_POSITION 2
Define LABEL_INFO_ANCHORX 3
Define LABEL_INFO_ANCHORY 4
Define LABEL_INFO_OFFSET 5
Define LABEL_INFO_ROWID 6
Define LABEL_INFO_TABLE 7
Define LABEL_INFO_EDIT 8
Define LABEL_INFO_EDIT_VISIBILITY 9

```

```

Define LABEL_INFO_EDIT_ANCHOR                10
Define LABEL_INFO_EDIT_OFFSET                11
Define LABEL_INFO_EDIT_FONT                 12
Define LABEL_INFO_EDIT_PEN                  13
Define LABEL_INFO_EDIT_TEXT                 14
Define LABEL_INFO_EDIT_TEXTARROW            15
Define LABEL_INFO_EDIT_ANGLE                16
Define LABEL_INFO_EDIT_POSITION             17
Define LABEL_INFO_EDIT_TEXTLINE             18
Define LABEL_INFO_SELECT                    19
Define LABEL_INFO_DRAWN                     20
Define LABEL_INFO_ORIENTATION                21

'=====
' Codes passed to the GroupLayerInfo function to get info about a group
' layer.
'=====
Define GROUPLAYER_INFO_NAME                  1
Define GROUPLAYER_INFO_LAYERLIST_ID         2
Define GROUPLAYER_INFO_DISPLAY              3
Define GROUPLAYER_INFO_LAYERS               4
Define GROUPLAYER_INFO_ALL_LAYERS           5
Define GROUPLAYER_INFO_TOPLEVEL_LAYERS      6
Define GROUPLAYER_INFO_PARENT_GROUP_ID      7

'=====
' Values returned by GroupLayerInfo() for GROUPLAYER_INFO_DISPLAY. These
' defines correspond to the MapBasic defines in MAPBASIC.DEF. If you
' alter
' these you must alter MAPBASIC.DEF.
'=====
Define GROUPLAYER_INFO_DISPLAY_OFF          0
Define GROUPLAYER_INFO_DISPLAY_ON           1

'*****
' Codes passed to the LayerListInfo function to help enumerating all
' layers in a Map.
'*****
Define LAYERLIST_INFO_TYPE                  1
Define LAYERLIST_INFO_NAME                  2
Define LAYERLIST_INFO_LAYER_ID              3
Define LAYERLIST_INFO_GROUPLAYER_ID         4

'*****
' Values returned by LayerListInfo() for LAYERLIST_INFO_TYPE. These
' defines correspond to the MapBasic defines in MAPBASIC.DEF. If you
' alter
' these you must alter MAPBASIC.DEF.
'*****
Define LAYERLIST_INFO_TYPE_LAYER            0
Define LAYERLIST_INFO_TYPE_GROUP            1

```

```

'=====
' LayerInfo() defines
'=====
Define LAYER_INFO_NAME 1
Define LAYER_INFO_EDITABLE 2
Define LAYER_INFO_SELECTABLE 3
Define LAYER_INFO_ZOOM_LAYERED 4
Define LAYER_INFO_ZOOM_MIN 5
Define LAYER_INFO_ZOOM_MAX 6
Define LAYER_INFO_COSMETIC 7
Define LAYER_INFO_PATH 8
Define LAYER_INFO_DISPLAY 9
Define LAYER_INFO_OVR_LINE 10
Define LAYER_INFO_OVR_PEN 11
Define LAYER_INFO_OVR_BRUSH 12
Define LAYER_INFO_OVR_SYMBOL 13
Define LAYER_INFO_OVR_FONT 14
Define LAYER_INFO_LBL_EXPR 15
Define LAYER_INFO_LBL_LT 16
Define LAYER_INFO_LBL_CURFONT 17
Define LAYER_INFO_LBL_FONT 18
Define LAYER_INFO_LBL_PARALLEL 19
Define LAYER_INFO_LBL_POS 20
Define LAYER_INFO_ARROWS 21
Define LAYER_INFO_NODES 22
Define LAYER_INFO_CENTROIDS 23
Define LAYER_INFO_TYPE 24
Define LAYER_INFO_LBL_VISIBILITY 25
Define LAYER_INFO_LBL_ZOOM_MIN 26
Define LAYER_INFO_LBL_ZOOM_MAX 27
Define LAYER_INFO_LBL_AUTODISPLAY 28
Define LAYER_INFO_LBL_OVERLAP 29
Define LAYER_INFO_LBL_DUPLICATES 30
Define LAYER_INFO_LBL_OFFSET 31
Define LAYER_INFO_LBL_MAX 32
Define LAYER_INFO_LBL_PARTIALSEGS 33
Define LAYER_INFO_HOTLINK_EXPR 34
Define LAYER_INFO_HOTLINK_MODE 35
Define LAYER_INFO_HOTLINK_RELATIVE 36
Define LAYER_INFO_HOTLINK_COUNT 37
Define LAYER_INFO_LBL_ORIENTATION 38
Define LAYER_INFO_LAYER_ALPHA 39
Define LAYER_INFO_LAYER_TRANSLUCENCY 40
Define LAYER_INFO_LABEL_ALPHA 41
Define LAYER_INFO_LAYERLIST_ID 42
Define LAYER_INFO_PARENT_GROUP_ID 43
'Code 44 - 52 are for override style & label
Define LAYER_INFO_OVR_STYLE_COUNT 44
Define LAYER_INFO_OVR_LBL_COUNT 45
Define LAYER_INFO_OVR_STYLE_CURRENT 46
Define LAYER_INFO_OVR_LBL_CURRENT 47
Define LAYER_INFO_OVR_LINE_COUNT 48
Define LAYER_INFO_OVR_PEN_COUNT 49

```

```

Define LAYER_INFO_OVR_BRUSH_COUNT          50
Define LAYER_INFO_OVR_SYMBOL_COUNT         51
Define LAYER_INFO_OVR_FONT_COUNT           52
Define LAYER_INFO_TILE_SERVER_LEVEL        53
'Advanced labelling options
Define LAYER_INFO_LBL_AUTO_POSITION        54
Define LAYER_INFO_LBL_AUTO_SIZES           55
Define LAYER_INFO_LBL_SUPPRESS_IF_NO_FIT   56
Define LAYER_INFO_LBL_AUTO_SIZE_STEP       57
Define LAYER_INFO_LBL_CURVED_BEST_POSITION  58
Define LAYER_INFO_LBL_CURVED_FALLBACK      59
Define LAYER_INFO_LBL_USE_ABBREVIATION     60
Define LAYER_INFO_LBL_AUTO_CALLOUT         61
Define LAYER_INFO_LBL_ORDER                62
Define LAYER_INFO_FRIENDLYNAME             64
Define LAYER_INFO_GRIDLINE_COORDSYS        65
Define LAYER_INFO_GRIDLINE_SPACE_HORIZONTAL 66
Define LAYER_INFO_GRIDLINE_SPACE_VERTICAL  67
Define LAYER_INFO_GRIDLINE_AUTO            68
Define LAYER_INFO_GRIDLINE_MIN_HORIZONTAL  69
Define LAYER_INFO_GRIDLINE_MIN_VERTICAL    70
Define LAYER_INFO_GRIDLINE_PAPER_HORIZONTAL 71
Define LAYER_INFO_GRIDLINE_PAPER_VERTICAL  72
Define LAYER_INFO_GRIDLINE_OFFSET          73
Define LAYER_INFO_LABEL_POS_RETRY          74
Define LAYER_INFO_GRIDLINE_HORIZONTAL_LABEL_POS 75
Define LAYER_INFO_GRIDLINE_VERTICAL_LABEL_POS 76
Define LAYER_INFO_GRIDLINE_LABEL_DIRECTION 77
Define LAYER_INFO_ALIAS                    78
Define LAYER_INFO_LBL_LINE_PEN             79
Define LAYER_INFO_RASTER_CONTRAST          80
Define LAYER_INFO_RASTER_BRIGHTNESS        81
Define LAYER_INFO_RASTER_ALPHA             82
Define LAYER_INFO_RASTER_GREYSCALE         83
Define LAYER_INFO_LABEL_EXPR_TABLE         84
Define LAYER_INFO_LBL_CURVED_PERCENT_OVER  85
Define LAYER_INFO_LEGEND                   86

'-----
' Values returned by LayerInfo() for LAYER_INFO_LABEL_ORIENTATION and
' LABEL_INFO_ORIENTATION.
'-----
Define LAYER_INFO_LABEL_ORIENT_HORIZONTAL  0
Define LAYER_INFO_LABEL_ORIENT_PARALLEL    1
Define LAYER_INFO_LABEL_ORIENT_CURVED      2

'-----
' Display Modes, returned by LayerInfo() for LAYER_INFO_DISPLAY
'-----
Define LAYER_INFO_DISPLAY_OFF              0
Define LAYER_INFO_DISPLAY_GRAPHIC          1
Define LAYER_INFO_DISPLAY_GLOBAL           2

```



```

Define LAYER_INFO_DISPLAY_VALUE          3

'-----
' Label Linetypes, returned by LayerInfo() for LAYER_INFO_LBL_LT
'-----
Define LAYER_INFO_LBL_LT_NONE            0
Define LAYER_INFO_LBL_LT_SIMPLE          1
Define LAYER_INFO_LBL_LT_ARROW           2

'-----
' Label Positions, returned by LayerInfo() for LAYER_INFO_LBL_POS and
' LabelInfo() for LABEL_INFO_POSITION
'-----
Define LAYER_INFO_LBL_POS_AUTO           -1
Define LAYER_INFO_LBL_POS_CC             0
Define LAYER_INFO_LBL_POS_TL             1
Define LAYER_INFO_LBL_POS_TC             2
Define LAYER_INFO_LBL_POS_TR             3
Define LAYER_INFO_LBL_POS_CL             4
Define LAYER_INFO_LBL_POS_CR             5
Define LAYER_INFO_LBL_POS_BL             6
Define LAYER_INFO_LBL_POS_BC             7
Define LAYER_INFO_LBL_POS_BR             8

'-----
' Layer Types, returned by LayerInfo() for LAYER_INFO_TYPE
'-----
Define LAYER_INFO_TYPE_NORMAL            0
Define LAYER_INFO_TYPE_COSMETIC          1
Define LAYER_INFO_TYPE_IMAGE             2
Define LAYER_INFO_TYPE_THEMATIC          3
Define LAYER_INFO_TYPE_GRID              4
Define LAYER_INFO_TYPE_WMS               5
Define LAYER_INFO_TYPE_TILESERVER        6
Define LAYER_INFO_TYPE_GRIDLINE          7

'-----
' Label visibility modes, from LayerInfo() for LAYER_INFO_LBL_VISIBILITY
'-----
Define LAYER_INFO_LBL_VIS_OFF            1
Define LAYER_INFO_LBL_VIS_ZOOM           2
Define LAYER_INFO_LBL_VIS_ON             3

'-----
' Values Returned by LayerInfo() for
' LAYER_INFO_GRIDLINE_HORIZONTAL_LABEL_POS
' and LAYER_INFO_GRIDLINE_VERTICAL_LABEL_POS.
'-----
Define GRIDLINE_HORIZONTAL_LABEL_NONE    0
Define GRIDLINE_HORIZONTAL_LABEL_LEFT    1
Define GRIDLINE_HORIZONTAL_LABEL_RIGHT   2
Define GRIDLINE_HORIZONTAL_LABEL_BOTH    3

```

```

Define GRIDLINE_VERTICAL_LABEL_NONE      0
Define GRIDLINE_VERTICAL_LABEL_TOP       1
Define GRIDLINE_VERTICAL_LABEL_BOTTOM    2
Define GRIDLINE_VERTICAL_LABEL_BOTH      3

'-----
' Values Returned by LayerInfo() for LAYER_INFO_GRIDLINE_LABEL_DIRECTION.
'-----

Define GRIDLINE_LABEL_DIRECTION_PLUSMINUS 0
Define GRIDLINE_LABEL_DIRECTION_COMPASS   1

'-----
' Code passed to StyleOverrideInfo function to get override style
information
'-----

Define STYLE_OVR_INFO_NAME                1
Define STYLE_OVR_INFO_VISIBILITY          2
Define STYLE_OVR_INFO_ZOOM_MIN            3
Define STYLE_OVR_INFO_ZOOM_MAX            4
Define STYLE_OVR_INFO_ARROWS              5
Define STYLE_OVR_INFO_NODES               6
Define STYLE_OVR_INFO_CENTROIDS           7
Define STYLE_OVR_INFO_ALPHA               8
Define STYLE_OVR_INFO_TRANSLUCENCY        9
Define STYLE_OVR_INFO_LINE               10
Define STYLE_OVR_INFO_PEN                 11
Define STYLE_OVR_INFO_BRUSH               12
Define STYLE_OVR_INFO_SYMBOL              13
Define STYLE_OVR_INFO_FONT                14
Define STYLE_OVR_INFO_SYMBOL_COUNT        15
Define STYLE_OVR_INFO_LINE_COUNT          16
Define STYLE_OVR_INFO_PEN_COUNT           17
Define STYLE_OVR_INFO_BRUSH_COUNT         18
Define STYLE_OVR_INFO_FONT_COUNT          19

'-----
' Possible return value of StyleOverrideInfo for code
STYLE_OVR_INFO_VISIBILITY
'-----

Define STYLE_OVR_INFO_VIS_OFF             0
Define STYLE_OVR_INFO_VIS_ON              1
Define STYLE_OVR_INFO_VIS_OFF_ZOOM        2

'-----
' Code passed to LabelOverrideInfo function to get override label
information
'-----

Define LBL_OVR_INFO_NAME                  1
Define LBL_OVR_INFO_VISIBILITY            2
Define LBL_OVR_INFO_ZOOM_MIN              3
Define LBL_OVR_INFO_ZOOM_MAX              4
Define LBL_OVR_INFO_EXPR                   5

```

```

Define LBL_OVR_INFO_LT 6
Define LBL_OVR_INFO_FONT 7
Define LBL_OVR_INFO_PARALLEL 8
Define LBL_OVR_INFO_POS 9
Define LBL_OVR_INFO_OVERLAP 10
Define LBL_OVR_INFO_DUPLICATES 11
Define LBL_OVR_INFO_OFFSET 12
Define LBL_OVR_INFO_MAX 13
Define LBL_OVR_INFO_PARTIALSEGS 14
Define LBL_OVR_INFO_ORIENTATION 15
Define LBL_OVR_INFO_ALPHA 16
Define LBL_OVR_INFO_AUTODISPLAY 17
Define LBL_OVR_INFO_POS_RETRY 18
Define LBL_OVR_INFO_LINE_PEN 19
Define LBL_OVR_INFO_PERCENT_OVER 20
Define LBL_OVR_INFO_AUTO_POSITION 21
Define LBL_OVR_INFO_AUTO_SIZES 22
Define LBL_OVR_INFO_SUPPRESS_IF_NO_FIT 23
Define LBL_OVR_INFO_AUTO_SIZE_STEP 24
Define LBL_OVR_INFO_CURVED_BEST_POSITION 25
Define LBL_OVR_INFO_CURVED_FALLBACK 26
Define LBL_OVR_INFO_USE_ABBREVIATION 27
Define LBL_OVR_INFO_ABBREVIATION_EXPR 28
Define LBL_OVR_INFO_AUTO_CALLOUT 29
Define LBL_OVR_INFO_EXPR_EDITED 30
Define LBL_OVR_INFO_LT_EDITED 31
Define LBL_OVR_INFO_FONT_EDITED 32
Define LBL_OVR_INFO_PARALLEL_EDITED 33
Define LBL_OVR_INFO_POS_EDITED 34
Define LBL_OVR_INFO_OVERLAP_EDITED 35
Define LBL_OVR_INFO_DUPLICATES_EDITED 36
Define LBL_OVR_INFO_OFFSET_EDITED 37
Define LBL_OVR_INFO_MAX_EDITED 38
Define LBL_OVR_INFO_PARTIALSEGS_EDITED 39
Define LBL_OVR_INFO_ORIENTATION_EDITED 40
Define LBL_OVR_INFO_ALPHA_EDITED 41
Define LBL_OVR_INFO_AUTODISPLAY_EDITED 42
Define LBL_OVR_INFO_POS_RETRY_EDITED 43
Define LBL_OVR_INFO_LINE_PEN_EDITED 44
Define LBL_OVR_INFO_PERCENT_OVER_EDITED 45
Define LBL_OVR_INFO_AUTO_POSITION_EDITED 46
Define LBL_OVR_INFO_AUTO_SIZES_EDITED 47
Define LBL_OVR_INFO_SUPPR_IFNOFIT_EDITED 48
Define LBL_OVR_INFO_AUTO_SIZESTEP_EDITED 49
Define LBL_OVR_INFO_CURV_BEST_POS_EDITED 50
Define LBL_OVR_INFO_CURV_FALLBACK_EDITED 51
Define LBL_OVR_INFO_USE_ABBREV_EDITED 52
Define LBL_OVR_INFO_ABBREV_EXPR_EDITED 53
Define LBL_OVR_INFO_AUTO_CALLOUT_EDITED 54

```

```

'-----
' Possible return value of LabelOverrideInfo for code
LBL_OVR_INFO_VISIBILITY

```

```

'-----
Define LBL_OVR_INFO_VIS_OFF          0
Define LBL_OVR_INFO_VIS_ON           1
Define LBL_OVR_INFO_VIS_OFF_ZOOM     2

'-----
' LayerControlInfo() defines
'-----
Define LC_INFO_SEL_COUNT              1

'-----
' LayerControlSelectionInfo() defines
'-----
Define LC_SEL_INFO_NAME               1
Define LC_SEL_INFO_TYPE              2
Define LC_SEL_INFO_MAPWIN_ID         3
Define LC_SEL_INFO_LAYER_ID          4
Define LC_SEL_INFO_OVR_ID            5
Define LC_SEL_INFO_HAS_COLOR_SAMPLE  6

'-----
' Values returned by LayerControlSelectionInfo() for LC_SEL_INFO_TYPE.
'-----
Define LC_SEL_INFO_TYPE_MAP          0
Define LC_SEL_INFO_TYPE_LAYER        1
Define LC_SEL_INFO_TYPE_GROUPLAYER  2
Define LC_SEL_INFO_TYPE_STYLE_OVR    3
Define LC_SEL_INFO_TYPE_LABEL_OVR    4
Define LC_SEL_INFO_TYPE_THEME_ITEM   5

'-----
' HotlinkInfo()
defines'-----
Define HOTLINK_INFO_EXPR             1
Define HOTLINK_INFO_MODE             2
Define HOTLINK_INFO_RELATIVE         3
Define HOTLINK_INFO_ENABLED          4
Define HOTLINK_INFO_ALIAS            5

'-----
' Hotlink activation modes, from LayerInfo() for LAYER_INFO_HOTLINK_MODE
'-----
Define HOTLINK_MODE_LABEL            0
Define HOTLINK_MODE_OBJ              1
Define HOTLINK_MODE_BOTH             2

'=====
' LegendInfo() defines
'=====
Define LEGEND_INFO_MAP_ID            1
Define LEGEND_INFO_ORIENTATION       2
Define LEGEND_INFO_NUM_FRAMES        3
Define LEGEND_INFO_STYLE_SAMPLE_SIZE 4

```

```

Define LEGEND_INFO_LINE_SAMPLE_WIDTH          5
Define LEGEND_INFO_REGION_SAMPLE_WIDTH        6
Define LEGEND_INFO_REGION_SAMPLE_HEIGHT      7
Define LEGEND_INFO_NUM_TEXTFRAMES            8
Define LEGEND_INFO_DEFAULT_TITLE              9
Define LEGEND_INFO_DEFAULT_TITLE_FONT        10
Define LEGEND_INFO_DEFAULT_SUBTITLE           11
Define LEGEND_INFO_DEFAULT_SUBTITLE_FONT     12
Define LEGEND_INFO_DEFAULT_STYLE             13
Define LEGEND_INFO_DEFAULT_STYLE_FONT        14
Define LEGEND_INFO_DEFAULT_AUTO_FNT_SZ       15

'=====
' Orientation codes, returned by LegendInfo() for LEGEND_INFO_ORIENTATION
'=====
Define ORIENTATION_PORTRAIT                   1
Define ORIENTATION_LANDSCAPE                 2
Define ORIENTATION_CUSTOM                    3

'-----
' Style sample codes, from LegendInfo() for LEGEND_INFO_STYLE_SAMPLE_SIZE
'-----
Define STYLE_SAMPLE_SIZE_SMALL               0
Define STYLE_SAMPLE_SIZE_LARGE               1

'=====
' LegendFrameInfo() defines
'=====
Define FRAME_INFO_TYPE                       1
Define FRAME_INFO_MAP_LAYER_ID              2
Define FRAME_INFO_REFRESHABLE               3
Define FRAME_INFO_POS_X                     4
Define FRAME_INFO_POS_Y                     5
Define FRAME_INFO_WIDTH                     6
Define FRAME_INFO_HEIGHT                    7
Define FRAME_INFO_TITLE                     8
Define FRAME_INFO_TITLE_FONT                9
Define FRAME_INFO_SUBTITLE                  10
Define FRAME_INFO_SUBTITLE_FONT             11
Define FRAME_INFO_BORDER_PEN                12
Define FRAME_INFO_NUM_STYLES                 13
Define FRAME_INFO_VISIBLE                    14
Define FRAME_INFO_COLUMN                    15
Define FRAME_INFO_LABEL                     16
Define FRAME_INFO_COLUMNS                    17
Define FRAME_INFO_NUM_VISIBLE_ROWS          18
Define FRAME_INFO_LINE_SAMPLE_WIDTH         19
Define FRAME_INFO_REGION_SAMPLE_WIDTH       20
Define FRAME_INFO_REGION_SAMPLE_HEIGHT     21
Define FRAME_INFO_AUTO_FONT_SIZE            22
Define FRAME_INFO_SHOW_COUNTS               23
Define FRAME_INFO_DISPLAY_TYPE              24

```

```

Define FRAME_INFO_BAR_THICKNESS          25
Define FRAME_INFO_BAR_LENGTH             26
Define FRAME_INFO_BAR_IS_HORIZONTAL      27

'=====
' LegendTextFrameInfo() defines
'=====
Define TEXTFRAME_INFO_POS_X              1
Define TEXTFRAME_INFO_POS_Y              2
Define TEXTFRAME_INFO_WIDTH              3
Define TEXTFRAME_INFO_HEIGHT             4
Define TEXTFRAME_INFO_TEXT               5
Define TEXTFRAME_INFO_TEXT_FONT          6

'=====
' Frame Types, returned by LegendFrameInfo() for FRAME_INFO_TYPE
'=====
Define FRAME_TYPE_NONE                   0
Define FRAME_TYPE_STYLE                  1
Define FRAME_TYPE_THEME                  2
Define FRAME_TYPE_RASTER                 3

'-----
' Frame display type codes, from LegendFrameInfo() for
FRAME_INFO_DISPLAY_TYPE
'-----
Define FRAME_DISPLAY_TYPE_NOT_SPECIFIED  0
Define FRAME_DISPLAY_TYPE_GAP            1
Define FRAME_DISPLAY_TYPE_PALETTE        2
Define FRAME_DISPLAY_TYPE_BAR            3

'=====
' Geocode Attributes, returned by GeocodeInfo()
'=====
Define GEOCODE_STREET_NAME               1
Define GEOCODE_STREET_NUMBER             2
Define GEOCODE_MUNICIPALITY              3
Define GEOCODE_MUNICIPALITY2             4
Define GEOCODE_COUNTRY_SUBDIVISION       5
Define GEOCODE_COUNTRY_SUBDIVISION2      6
Define GEOCODE_POSTAL_CODE               7

Define GEOCODE_DICTIONARY                 9
Define GEOCODE_BATCH_SIZE                10
Define GEOCODE_FALLBACK_GEOGRAPHIC       11
Define GEOCODE_FALLBACK_POSTAL           12
Define GEOCODE_OFFSET_CENTER             13
Define GEOCODE_OFFSET_CENTER_UNITS       14
Define GEOCODE_OFFSET_END                 15
Define GEOCODE_OFFSET_END_UNITS          16
Define GEOCODE_MIXED_CASE                 17
Define GEOCODE_RESULT_MARK_MULTIPLE      18
Define GEOCODE_COUNT_GEOCODED            19

```

```

Define GEOCODE_COUNT_NOTGEOCODED      20
Define GEOCODE_UNABLE_TO_CONVERT_DATA  21
Define GEOCODE_MAX_BATCH_SIZE          22
Define GEOCODE_PASSTHROUGH             100

```

```

Define DICTIONARY_ALL                  1
Define DICTIONARY_ADDRESS_ONLY        2
Define DICTIONARY_USER_ONLY           3
Define DICTIONARY_PREFER_ADDRESS      4
Define DICTIONARY_PREFER_USER         5

```

```

'=====
' ISOGRAM Attributes, returned by IsogramInfo()
'=====

```

```

Define ISOGRAM_BANDING                  1
Define ISOGRAM_MAJOR_ROADS_ONLY        2
Define ISOGRAM_RETURN_HOLES            3
Define ISOGRAM_MAJOR_POLYGON_ONLY      4
Define ISOGRAM_MAX_OFFROAD_DIST        5
Define ISOGRAM_MAX_OFFROAD_DIST_UNITS  6
Define ISOGRAM_SIMPLIFICATION_FACTOR    7
Define ISOGRAM_DEFAULT_AMBIENT_SPEED    8
Define ISOGRAM_AMBIENT_SPEED_DIST_UNIT  9
Define ISOGRAM_AMBIENT_SPEED_TIME_UNIT 10
Define ISOGRAM_PROPAGATION_FACTOR      11
Define ISOGRAM_BATCH_SIZE              12
Define ISOGRAM_POINTS_ONLY             13
Define ISOGRAM_RECORDS_INSERTED        14
Define ISOGRAM_RECORDS_NOTINSERTED     15
Define ISOGRAM_MAX_BATCH_SIZE          16
Define ISOGRAM_MAX_BANDS               17
Define ISOGRAM_MAX_DISTANCE            18
Define ISOGRAM_MAX_DISTANCE_UNITS      19
Define ISOGRAM_MAX_TIME                20
Define ISOGRAM_MAX_TIME_UNITS          21

```

```

'=====
' LegendStyleInfo() defines
'=====

```

```

Define LEGEND_STYLE_INFO_TEXT          1
Define LEGEND_STYLE_INFO_FONT          2
Define LEGEND_STYLE_INFO_OBJ           3
Define LEGEND_STYLE_INFO_ROW_VISIBLE   4
Define LEGEND_STYLE_INFO_COUNT         5

```

```

'=====
' LayoutInfo() defines
'=====

```

```

Define LAYOUT_INFO_NUM_ITEMS           1
Define LAYOUT_INFO_WIDTH               2
Define LAYOUT_INFO_HEIGHT              3
Define LAYOUT_INFO_LEFT_MARGIN         4
Define LAYOUT_INFO_RIGHT_MARGIN        5

```

```

Define LAYOUT_INFO_TOP_MARGIN 6
Define LAYOUT_INFO_BOTTOM_MARGIN 7
Define LAYOUT_INFO_ZOOM 8
Define LAYOUT_INFO_CENTER_X 9
Define LAYOUT_INFO_CENTER_Y 10
Define LAYOUT_INFO_NUM_PAGES 11
Define LAYOUT_INFO_CUR_PAGE 12
Define LAYOUT_INFO_GRID_VISIBILITY 13
Define LAYOUT_INFO_GRID_SNAP 14
Define LAYOUT_INFO_GRID_SIZE 15
Define LAYOUT_INFO_SMART_GUIDES 16

'=====
' LayoutItemInfo() and LayoutPageItemInfo() defines
'=====
Define LAYOUT_ITEM_INFO_POS_X 1
Define LAYOUT_ITEM_INFO_POS_Y 2
Define LAYOUT_ITEM_INFO_WIDTH 3
Define LAYOUT_ITEM_INFO_HEIGHT 4
Define LAYOUT_ITEM_INFO_WIN 5
Define LAYOUT_ITEM_INFO_SELECTED 6
Define LAYOUT_ITEM_INFO_ACTIVATED 7
Define LAYOUT_ITEM_INFO_EMPTY 8
Define LAYOUT_ITEM_INFO_LEGEND_FRAME_ID 9
Define LAYOUT_ITEM_INFO_LEGEND_DESIGNER_WINDOW 10
Define LAYOUT_ITEM_INFO_TYPE 11
Define LAYOUT_ITEM_INFO_IMAGE_FILE 12
Define LAYOUT_ITEM_INFO_OBJ 13
Define LAYOUT_ITEM_INFO_NAME 14
Define LAYOUT_ITEM_INFO_PEN 15
Define LAYOUT_ITEM_INFO_FILL_BRUSH 16
Define LAYOUT_ITEM_INFO_Z_ORDER 17
Define LAYOUT_ITEM_INFO_ANGLE 18
Define LAYOUT_ITEM_INFO_ADDIN_FILE_NAME 19
Define LAYOUT_ITEM_INFO_SERIALIZED_NAME 20
Define LAYOUT_ITEM_INFO_XPS_VIEW_NAME 21
Define LAYOUT_ITEM_INFO_NUM_ITEMS_ON_PAGE 22
Define LAYOUT_ITEM_INFO_FRAME_UNIQUE_ID 23
Define LAYOUT_ITEM_INFO_TEXTSTRING 24
Define LAYOUT_ITEM_INFO_SMART_TEXTSTRING 25

'=====
' Item types, returned by LayoutItemInfo() and LayoutPageItemInfo() for
' LAYOUT_ITEM_INFO_TYPE
'=====
Define LAYOUT_ITEM_TYPE_ANY -1
Define LAYOUT_ITEM_TYPE_EMPTY 0
Define LAYOUT_ITEM_TYPE_MAPPER 1
Define LAYOUT_ITEM_TYPE_BROWSER 2
Define LAYOUT_ITEM_TYPE_LEGEND 3
Define LAYOUT_ITEM_TYPE_TEXT 4
Define LAYOUT_ITEM_TYPE_SHAPE 5
Define LAYOUT_ITEM_TYPE_IMAGE 6

```



```

Define LAYOUT_ITEM_TYPE_CUSTOM 7

'=====
' CurContext() defines
'=====
Define CONTEXT_WINDOWID 1
Define CONTEXT_PAGENUMBER 2
Define CONTEXT_FRAMENUMBER 3

'=====
' LocateFile$() defines
'=====

Define LOCATE_PREF_FILE 0
Define LOCATE_DEF_WOR 1
Define LOCATE_CLR_FILE 2
Define LOCATE_PEN_FILE 3
Define LOCATE_FNT_FILE 4
Define LOCATE_ABB_FILE 5
Define LOCATE_PRJ_FILE 6
Define LOCATE_MNU_FILE 7
Define LOCATE_CUSTSYMB_DIR 8
Define LOCATE_THMTMPLT_DIR 9
Define LOCATE_GRAPH_DIR 10
Define LOCATE_WMS_SERVERLIST 11
Define LOCATE_WFS_SERVERLIST 12
Define LOCATE_GEOCODE_SERVERLIST 13
Define LOCATE_ROUTING_SERVERLIST 14
Define LOCATE_LAYOUT_TEMPLATE_DIR 15

'=====
' GetLicenseKind() defines
'=====
Define LICENSE_KIND_NOLICENSE 0
Define LICENSE_KIND_REGULAR 1
Define LICENSE_KIND_ADVANCE 2

'=====
' TableConnectionInfo() defines
'=====
Define TABLE_CONNECTION_INFO_NUMBER 1

'=====
' TableSchemaInfo() defines
'=====
Define TABLE_SCHEMA_INFO_NAME 1

'=====
' Map3DInfo() defines
'=====
Define MAP3D_INFO_SCALE 1
Define MAP3D_INFO_RESOLUTION_X 2
Define MAP3D_INFO_RESOLUTION_Y 3

```

```

Define MAP3D_INFO_BACKGROUND 4
Define MAP3D_INFO_UNITS 5
Define MAP3D_INFO_LIGHT_X 6
Define MAP3D_INFO_LIGHT_Y 7
Define MAP3D_INFO_LIGHT_Z 8
Define MAP3D_INFO_LIGHT_COLOR 9
Define MAP3D_INFO_CAMERA_X 10
Define MAP3D_INFO_CAMERA_Y 11
Define MAP3D_INFO_CAMERA_Z 12
Define MAP3D_INFO_CAMERA_FOCAL_X 13
Define MAP3D_INFO_CAMERA_FOCAL_Y 14
Define MAP3D_INFO_CAMERA_FOCAL_Z 15
Define MAP3D_INFO_CAMERA_VU_1 16
Define MAP3D_INFO_CAMERA_VU_2 17
Define MAP3D_INFO_CAMERA_VU_3 18
Define MAP3D_INFO_CAMERA_VPN_1 19
Define MAP3D_INFO_CAMERA_VPN_2 20
Define MAP3D_INFO_CAMERA_VPN_3 21
Define MAP3D_INFO_CAMERA_CLIP_NEAR 22
Define MAP3D_INFO_CAMERA_CLIP_FAR 23

'=====
' MapperInfo() defines
'=====
Define MAPPER_INFO_ZOOM 1
Define MAPPER_INFO_SCALE 2
Define MAPPER_INFO_CENTERX 3
Define MAPPER_INFO_CENTERY 4
Define MAPPER_INFO_MINX 5
Define MAPPER_INFO_MINY 6
Define MAPPER_INFO_MAXX 7
Define MAPPER_INFO_MAXY 8
Define MAPPER_INFO_LAYERS 9
Define MAPPER_INFO_EDIT_LAYER 10
Define MAPPER_INFO_XYUNITS 11
Define MAPPER_INFO_DISTUNITS 12
Define MAPPER_INFO_AREAUNITS 13
Define MAPPER_INFO_SCROLLBARS 14
Define MAPPER_INFO_DISPLAY 15
Define MAPPER_INFO_NUM_THEMATIC 16
Define MAPPER_INFO_COORDSYS_CLAUSE 17
Define MAPPER_INFO_COORDSYS_NAME 18
Define MAPPER_INFO_MOVE_DUPLICATE_NODES 19
Define MAPPER_INFO_DIST_CALC_TYPE 20
Define MAPPER_INFO_DISPLAY_DMS 21
Define MAPPER_INFO_COORDSYS_CLAUSE_WITH_BOUNDS 22
Define MAPPER_INFO_CLIP_TYPE 23
Define MAPPER_INFO_CLIP_REGION 24
Define MAPPER_INFO_REPROJECTION 25
Define MAPPER_INFO_RESAMPLING 26
Define MAPPER_INFO_MERGE_MAP 27
Define MAPPER_INFO_ALL_LAYERS 28
Define MAPPER_INFO_GROUPLAYERS 29

```

```

Define MAPPER_INFO_NUM_ADORNMENTS          200
Define MAPPER_INFO_ADORNMENT               200
Define MAPPER_INFO_LABELS_SELECTABLE       30
Define MAPPER_INFO_HAS_CUSTOM_LABELS       31
Define MAPPER_INFO_CARTO_SCALE             32
Define MAPPER_INFO_CARTO_SCALE_EXPR        33

'-----
' Display Modes, returned by MapperInfo() for MAPPER_INFO_DISPLAY_DMS
'-----
Define MAPPER_INFO_DISPLAY_DECIMAL         0
Define MAPPER_INFO_DISPLAY_DEGMINSEC       1
Define MAPPER_INFO_DISPLAY_MGRS            2
Define MAPPER_INFO_DISPLAY_USNG_WGS84      3
Define MAPPER_INFO_DISPLAY_USNG_NAD27      4

'-----
' Display Modes, returned by MapperInfo() for MAPPER_INFO_DISPLAY
'-----
Define MAPPER_INFO_DISPLAY_SCALE            0
Define MAPPER_INFO_DISPLAY_ZOOM             1
Define MAPPER_INFO_DISPLAY_POSITION         2
Define MAPPER_INFO_DISPLAY_CARTOGRAPHIC_SCALE 3

'-----
' Distance Calculation Types from MapperInfo() for
MAPPER_INFO_DIST_CALC_TYPE
'-----
Define MAPPER_INFO_DIST_SPHERICAL           0
Define MAPPER_INFO_DIST_CARTESIAN           1

'-----
' Clip Types, returned by MapperInfo() for MAPPER_INFO_CLIP_TYPE
'-----
Define MAPPER_INFO_CLIP_DISPLAY_ALL         0
Define MAPPER_INFO_CLIP_DISPLAY_POLYOBJ     1
Define MAPPER_INFO_CLIP_OVERLAY             2

'=====
' MenuItemInfoByID() and MenuItemInfoByHandler() defines
'=====
Define MENUITEM_INFO_ENABLED               1
Define MENUITEM_INFO_CHECKED               2
Define MENUITEM_INFO_CHECKABLE             3
Define MENUITEM_INFO_SHOWHIDEABLE         4
Define MENUITEM_INFO_ACCELERATOR           5
Define MENUITEM_INFO_TEXT                  6
Define MENUITEM_INFO_HELPMSG               7
Define MENUITEM_INFO_HANDLER               8
Define MENUITEM_INFO_ID                    9

'=====
' ObjectGeography() defines
'=====

```

```

Define OBJ_GEO_MINX 1
Define OBJ_GEO_LINEBEGX 1
Define OBJ_GEO_POINTX 1
Define OBJ_GEO_MINY 2
Define OBJ_GEO_LINEBEGY 2
Define OBJ_GEO_POINTY 2
Define OBJ_GEO_MAXX 3
Define OBJ_GEO_LINEENDX 3
Define OBJ_GEO_MAXY 4
Define OBJ_GEO_LINEENDY 4
Define OBJ_GEO_ARCBEGANGLE 5
Define OBJ_GEO_TEXTLINEX 5
Define OBJ_GEO_ROUNDRAIDUS 5
Define OBJ_GEO_CENTROID 5
Define OBJ_GEO_ARCENDANGLE 6
Define OBJ_GEO_TEXTLINEY 6
Define OBJ_GEO_TEXTANGLE 7
Define OBJ_GEO_POINTZ 8
Define OBJ_GEO_POINTM 9

'=====
' ObjectInfo() defines
'=====
Define OBJ_INFO_TYPE 1
Define OBJ_INFO_PEN 2
Define OBJ_INFO_SYMBOL 2
Define OBJ_INFO_TEXTFONT 2
Define OBJ_INFO_BRUSH 3
Define OBJ_INFO_NPNTS 20
Define OBJ_INFO_TEXTSTRING 3
Define OBJ_INFO_SMOOTH 4
Define OBJ_INFO_FRAMEWIN 4
Define OBJ_INFO_NPOLYGONS 21
Define OBJ_INFO_TEXTSPACING 4
Define OBJ_INFO_TEXTJUSTIFY 5
Define OBJ_INFO_FRAMEITITLE 6
Define OBJ_INFO_TEXTARROW 6
Define OBJ_INFO_FILLFRAME 7
Define OBJ_INFO_REGION 8
Define OBJ_INFO_PLINE 9
Define OBJ_INFO_MPOINT 10
Define OBJ_INFO_NONEMPTY 11
Define OBJ_INFO_Z_UNIT_SET 12
Define OBJ_INFO_Z_UNIT 13
Define OBJ_INFO_HAS_Z 14
Define OBJ_INFO_HAS_M 15

'-----
' Object types, returned by ObjectInfo() for OBJ_INFO_TYPE
'-----
Define OBJ_TYPE_ARC 1
Define OBJ_TYPE_ELLIPSE 2
Define OBJ_TYPE_LINE 3

```

```

Define OBJ_TYPE_PLINE 4
Define OBJ_TYPE_POINT 5
Define OBJ_TYPE_FRAME 6
Define OBJ_TYPE_REGION 7
Define OBJ_TYPE_RECT 8
Define OBJ_TYPE_ROUNDRECT 9
Define OBJ_TYPE_TEXT 10
Define OBJ_TYPE_MPOINT 11
Define OBJ_TYPE_COLLECTION 12

```

```

'-----*
' RegionInfo() Defines
'-----*

```

```

Define REGION_INFO_IS_CLOCKWISE 1

```

```

'=====
' PrismMapInfo() defines
'=====

```

```

Define PRISM_MAP_INFO_SCALE 1
Define PRISM_MAP_INFO_BACKGROUND 4
Define PRISM_MAP_INFO_LIGHT_X 6
Define PRISM_MAP_INFO_LIGHT_Y 7
Define PRISM_MAP_INFO_LIGHT_Z 8
Define PRISM_MAP_INFO_LIGHT_COLOR 9
Define PRISM_MAP_INFO_CAMERA_X 10
Define PRISM_MAP_INFO_CAMERA_Y 11
Define PRISM_MAP_INFO_CAMERA_Z 12
Define PRISM_MAP_INFO_CAMERA_FOCAL_X 13
Define PRISM_MAP_INFO_CAMERA_FOCAL_Y 14
Define PRISM_MAP_INFO_CAMERA_FOCAL_Z 15
Define PRISM_MAP_INFO_CAMERA_VU_1 16
Define PRISM_MAP_INFO_CAMERA_VU_2 17
Define PRISM_MAP_INFO_CAMERA_VU_3 18
Define PRISM_MAP_INFO_CAMERA_VPN_1 19
Define PRISM_MAP_INFO_CAMERA_VPN_2 20
Define PRISM_MAP_INFO_CAMERA_VPN_3 21
Define PRISM_MAP_INFO_CAMERA_CLIP_NEAR 22
Define PRISM_MAP_INFO_CAMERA_CLIP_FAR 23
Define PRISM_MAP_INFO_INFOTIP_EXPR 24

```

```

'=====
' SearchInfo() defines
'=====

```

```

Define SEARCH_INFO_TABLE 1
Define SEARCH_INFO_ROW 2

```

```

'=====
' SelectionInfo() defines
'=====

```

```

Define SEL_INFO_TABLENAME 1
Define SEL_INFO_SELNAME 2
Define SEL_INFO_NROWS 3

```

```

Define  SEL_INFO_ISNEW                                4

'=====
' Server statement and function defines
'=====
'-----
' Return Codes
'-----
Define SRV_SUCCESS                                0
Define SRV_SUCCESS_WITH_INFO                      1
Define SRV_ERROR                                  -1
Define SRV_INVALID_HANDLE                         -2
Define SRV_NEED_DATA                              99
Define SRV_NO_MORE_DATA                           100

'-----
' Special values for the status associated with a fetched value
'-----
Define SRV_NULL_DATA                              -1
Define SRV_TRUNCATED_DATA                         -2

'-----
' Server_ColumnInfo() defines
'-----
Define SRV_COL_INFO_NAME                          1
Define SRV_COL_INFO_TYPE                          2
Define SRV_COL_INFO_WIDTH                         3
Define SRV_COL_INFO_PRECISION                     4
Define SRV_COL_INFO_SCALE                         5
Define SRV_COL_INFO_VALUE                         6
Define SRV_COL_INFO_STATUS                        7
Define SRV_COL_INFO_ALIAS                         8

'-----
' Column types, returned by Server_ColumnInfo() for SRV_COL_INFO_TYPE
'-----
Define SRV_COL_TYPE_NONE                          0
Define SRV_COL_TYPE_CHAR                          1
Define SRV_COL_TYPE_DECIMAL                       2
Define SRV_COL_TYPE_INTEGER                       3
Define SRV_COL_TYPE_SMALLINT                      4
Define SRV_COL_TYPE_DATE                          5
Define SRV_COL_TYPE_LOGICAL                       6
Define SRV_COL_TYPE_FLOAT                         8
Define SRV_COL_TYPE_FIXED_LEN_STRING              16
Define SRV_COL_TYPE_BIN_STRING                    17

'-----
' Server_DriverInfo() Attr defines
'-----
Define SRV_DRV_INFO_NAME                          1
Define SRV_DRV_INFO_NAME_LIST                     2
Define SRV_DRV_DATA_SOURCE                        3

```

```

'-----
' Server_ConnectInfo() Attr defines
'-----
Define SRV_CONNECT_INFO_DRIVER_NAME          1
Define SRV_CONNECT_INFO_DB_NAME              2
Define SRV_CONNECT_INFO_SQL_USER_ID          3
Define SRV_CONNECT_INFO_DS_NAME              4
Define SRV_CONNECT_INFO_QUOTE_CHAR           5
Define SRV_CONNECT_INFO_DATABASE_NAME        6
Define SRV_CONNECT_INFO_SERVER_NAME          7

'-----
' Fetch Directions (used by ServerFetch function in some code libraries)
'-----
Define SRV_FETCH_NEXT                        -1
Define SRV_FETCH_PREV                        -2
Define SRV_FETCH_FIRST                       -3
Define SRV_FETCH_LAST                        -4

'-----
' Oracle workspace manager
'-----
Define SRV_WM_HIST_NONE                      0
Define SRV_WM_HIST_OVERWRITE                 1
Define SRV_WM_HIST_NO_OVERWRITE              2

'=====
' SessionInfo() defines
'=====
Define SESSION_INFO_COORDSYS_CLAUSE          1
Define SESSION_INFO_DISTANCE_UNITS           2
Define SESSION_INFO_AREA_UNITS               3
Define SESSION_INFO_PAPER_UNITS              4
Define SESSION_INFO_DEF_NATIVE_BLKSZ         5
Define SESSION_INFO_DEF_NATIVEX_BLKSZ        6
Define SESSION_INFO_TEMP_PATH                7

'=====
' Set Next Document Style defines
'=====
Define WIN_STYLE_STANDARD                    0
Define WIN_STYLE_CHILD                       1
Define WIN_STYLE_POPUP_FULLCAPTION           2
Define WIN_STYLE_POPUP                       3

'=====
' StringCompare() defines
'=====
Define STR_LT                                -1
Define STR_GT                                 1
Define STR_EQ                                 0

```

```

'=====
' StyleAttr() defines
'=====
Define PEN_WIDTH 1
Define PEN_PATTERN 2
Define PEN_COLOR 4
Define PEN_INDEX 5
Define PEN_INTERLEAVED 6
Define BRUSH_PATTERN 1
Define BRUSH_FORECOLOR 2
Define BRUSH_BACKCOLOR 3
Define FONT_NAME 1
Define FONT_STYLE 2
Define FONT_POINTSIZE 3
Define FONT_FORECOLOR 4
Define FONT_BACKCOLOR 5
Define SYMBOL_CODE 1
Define SYMBOL_COLOR 2
Define SYMBOL_POINTSIZE 3
Define SYMBOL_ANGLE 4
Define SYMBOL_FONT_NAME 5
Define SYMBOL_FONT_STYLE 6
Define SYMBOL_KIND 7
Define SYMBOL_CUSTOM_NAME 8
Define SYMBOL_CUSTOM_STYLE 9

'-----
' Symbol kinds returned by StyleAttr() for SYMBOL_KIND
'-----
Define SYMBOL_KIND_VECTOR 1
Define SYMBOL_KIND_FONT 2
Define SYMBOL_KIND_CUSTOM 3

'=====
' SystemInfo() defines
'=====
Define SYS_INFO_PLATFORM 1
Define SYS_INFO_APPVERSION 2
Define SYS_INFO_MIVERSION 3
Define SYS_INFO_RUNTIME 4
Define SYS_INFO_CHARSET 5
Define SYS_INFO_COPYPROTECTED 6
Define SYS_INFO_APPLICATIONWND 7
Define SYS_INFO_DDESTATUS 8
Define SYS_INFO_MAPINFOWND 9
Define SYS_INFO_NUMBER_FORMAT 10
Define SYS_INFO_DATE_FORMAT 11
Define SYS_INFO_DIG_INSTALLED 12
Define SYS_INFO_DIG_MODE 13
Define SYS_INFO_MIPLATFORM 14
Define SYS_INFO_MDICLIENTWND 15
Define SYS_INFO_PRODUCTLEVEL 16
Define SYS_INFO_APPIDISPATCH 17

```



```

Define SYS_INFO_MIBUILD_NUMBER          18
Define SYS_INFO_MIFULLVERSION           19
Define SYS_INFO_IMAPINFOAPPLICATION     20
Define SYS_INFO_MAPINFO_INTERFACE       21
Define SYS_INFO_UNICODE                 22
Define SYS_INFO_CULTURE                  23
Define SYS_INFO_LOCALE                   24
Define SYS_INFO_MIRIBBONSTATE_VERSION   25
Define SYS_INFO_LICENSE_MODE             26
Define SYS_INFO_SUBSCRIPTION_PLAN       27

'-----
' Platform, returned by SystemInfo() for SYS_INFO_PLATFORM
'-----
Define PLATFORM_SPECIAL                  0
Define PLATFORM_WIN                      1
Define PLATFORM_MAC                      2
Define PLATFORM_MOTIF                    3
Define PLATFORM_X11                      4
Define PLATFORM_XOL                      5

'-----
' Version, returned by SystemInfo() for SYS_INFO_MIPLATFORM
'-----
Define MIPLATFORM_SPECIAL                0
Define MIPLATFORM_WIN16                  1
Define MIPLATFORM_WIN32                  2
Define MIPLATFORM_POWERMAC               3
Define MIPLATFORM_MAC68K                 4
Define MIPLATFORM_HP                     5
Define MIPLATFORM_SUN                    6
Define MIPLATFORM_WIN64                  7

'-----
' Interface Type, returned by SystemInfo() for SYS_INFO_MAPINFO_INTERFACE
'-----
Define MIINTERFACE_CLASSICMENU           0
Define MIINTERFACE_RIBBON                1

'=====
' TableInfo() defines
'=====
Define TAB_INFO_NAME                     1
Define TAB_INFO_NUM                      2
Define TAB_INFO_TYPE                     3
Define TAB_INFO_NCOLS                    4
Define TAB_INFO_MAPPABLE                 5
Define TAB_INFO_READONLY                 6
Define TAB_INFO_TEMP                     7
Define TAB_INFO_NROWS                    8
Define TAB_INFO_EDITED                   9
Define TAB_INFO_FASTEDIT                 10
Define TAB_INFO_UNDO                     11

```

```

Define TAB_INFO_MAPPABLE_TABLE 12
Define TAB_INFO_USERMAP 13
Define TAB_INFO_USERBROWSE 14
Define TAB_INFO_USERCLOSE 15
Define TAB_INFO_USEREDITABLE 16
Define TAB_INFO_USERREMOVEMAP 17
Define TAB_INFO_USERDISPLAYMAP 18
Define TAB_INFO_TABFILE 19
Define TAB_INFO_MINX 20
Define TAB_INFO_MINY 21
Define TAB_INFO_MAXX 22
Define TAB_INFO_MAXY 23
Define TAB_INFO_SEAMLESS 24
Define TAB_INFO_COORDSYS_MINX 25
Define TAB_INFO_COORDSYS_MINY 26
Define TAB_INFO_COORDSYS_MAXX 27
Define TAB_INFO_COORDSYS_MAXY 28
Define TAB_INFO_COORDSYS_CLAUSE 29
Define TAB_INFO_COORDSYS_NAME 30
Define TAB_INFO_NREFS 31
Define TAB_INFO_SUPPORT_MZ 32
Define TAB_INFO_Z_UNIT_SET 33
Define TAB_INFO_Z_UNIT 34
Define TAB_INFO_BROWSER_LIST 35
Define TAB_INFO_THEME_METADATA 36
Define TAB_INFO_COORDSYS_CLAUSE_WITHOUT_BOUNDS 37
Define TAB_INFO_DESCRIPTION 38
Define TAB_INFO_ID 39
Define TAB_INFO_PARENTID 40
Define TAB_INFO_ISMANAGED 41
Define TAB_INFO_ADSK_TEXTOBJECT 42
Define TAB_INFO_OVERRIDE_COORDINATE_ORDER 43
Define TAB_INFO_PERSIST 44
Define TAB_INFO_PREFER_HTML_FOR_INFO_TOOL 45
Define TAB_INFO_INUSE 46
Define TAB_INFO_DATA_FORMAT 47
Define TAB_INFO_CHARSET 48
Define TAB_INFO_MAP_BLOCKSIZE 49
Define TAB_INFO_READVERSION 50
Define TAB_INFO_EDITVERSION 51
Define TAB_INFO_READALLVERSION 52
Define TAB_INFO_LOCALE 53
Define TAB_INFO_CONNECTIONS_COUNT 54
Define TAB_INFO_SCHEMAS_COUNT 55
Define TAB_INFO_QUERY_STRING 56
Define TAB_INFO_ISHIDDEN 57
Define TAB_INFO_NUM_BASETABLES 58
Define TAB_INFO_BASETABLE_ALIASES 59
Define TAB_INFO_DOMINANT_OBJECT_TYPE 60

'-----
' Table type defines, returned by TableInfo() for TAB_INFO_TYPE
'-----

```

```

Define TAB_TYPE_BASE 1
Define TAB_TYPE_RESULT 2
Define TAB_TYPE_VIEW 3
Define TAB_TYPE_IMAGE 4
Define TAB_TYPE_LINKED 5
Define TAB_TYPE_WMS 6
Define TAB_TYPE_WFS 7
Define TAB_TYPE_FME 8
Define TAB_TYPE_TILESERVER 9

' Codes returned by TableInfo(<table-ref>, TAB_INFO_DOMINANT_OBJECT_TYPE).
Define TAB_UNKNOWN_OBJECTS 0
Define TAB_POINT_OBJECTS 2
Define TAB_LINE_OBJECTS 3
Define TAB_AREA_OBJECTS 4
Define TAB_MISC_OBJECTS 5
Define TAB_RASTER_OBJECTS 7

'-----
' Table format defines, returned by TableInfo() for TAB_INFO_DATA_FORMAT
'-----
Define TAB_DATA_FORMAT_NONE 0
Define TAB_DATA_FORMAT_DBASE 1
Define TAB_DATA_FORMAT_LOTUS 2
Define TAB_DATA_FORMAT_ASCII 3
Define TAB_DATA_FORMAT_EXCEL 4
Define TAB_DATA_FORMAT_NATIVE 5
Define TAB_DATA_FORMAT_RASTER 6
Define TAB_DATA_FORMAT_DAO 8
Define TAB_DATA_FORMAT_ODBC 10
Define TAB_DATA_FORMAT_NATIVEX 14
Define TAB_DATA_FORMAT_GPKG 15
Define TAB_DATA_FORMAT_SHAPE 16

'-----
' Defines used in LibraryServiceInfo function for what information to
return.
'-----
Define LIBSRVC_INFO_LIBSRVCMODE 1
Define LIBSRVC_INFO_LIBVERSION 2
Define LIBSRVC_INFO_DEFURLPATH 3
Define LIBSRVC_INFO_LISTCSWURL 4

'-----
' TableListInfo() defines
'-----
Define TL_INFO_SEL_COUNT 1

'-----
' TableListSelectionInfo() defines
'-----
Define TL_SEL_INFO_NAME 1

```

```

Define TL_SEL_INFO_ID                                2

'-----
' ThemeInfo() defines
'-----

Define THEME_INFO_TYPE                                1
Define THEME_INFO_ITEM_COUNT                          2
Define THEME_INFO_TABLE_ALIAS                        3
Define THEME_INFO_EXPRESSIONS_COUNT                  4

'-----
' Theme type defines returned by ThemeInfo() for THEME_INFO_TYPE
'-----

Define THEME_TYPE_RANGES                              0
Define THEME_TYPE_BARS                                1
Define THEME_TYPE_PIES                                 2
Define THEME_TYPE_GRADUATED                           3
Define THEME_TYPE_DOTDENSITY                          4
Define THEME_TYPE_UNIQUE                              5
Define THEME_TYPE_GRID                                6

'-----
' ThemeItemInfo() defines
'-----

Define THEME_ITEM_INFO_VISIBLE                        1
Define THEME_ITEM_INFO_LABEL                          2

'-----
' ThemeExpressionInfo() defines
'-----

Define THEME_EXPRESSION_INFO_STRING                   1
Define THEME_EXPRESSION_INFO_TYPE                     2

'-----
' Column type defines, returned by ThemeExpressionInfo() for
' THEME_EXPRESSION_INFO_TYPE
'-----

Define EXPRESSION_TYPE_CHAR                           1
Define EXPRESSION_TYPE_DECIMAL                        2
Define EXPRESSION_TYPE_INTEGER                        3
Define EXPRESSION_TYPE_SMALLINT                       4
Define EXPRESSION_TYPE_DATE                           5
Define EXPRESSION_TYPE_LOGICAL                        6
Define EXPRESSION_TYPE_GRAPHIC                        7
Define EXPRESSION_TYPE_FLOAT                          8
Define EXPRESSION_TYPE_TIME                           37
Define EXPRESSION_TYPE_DATETIME                       38

'-----
' RangeThemeInfo() defines
'-----

Define RANGETHHEME_INFO_ITEMS_COUNT                  1
Define RANGETHHEME_INFO_VARY_STYLE                   2

```

```

Define RANGETHHEME_INFO_REPLACE_STYLE          3
Define RANGETHHEME_INFO_OTHERS_LINE            4
Define RANGETHHEME_INFO_OTHERS_PEN             5
Define RANGETHHEME_INFO_OTHERS_BRUSH           6
Define RANGETHHEME_INFO_OTHERS_SYMBOL          7
Define RANGETHHEME_INFO_BASE_LINE              8
Define RANGETHHEME_INFO_BASE_BRUSH             9
Define RANGETHHEME_INFO_BASE_SYMBOL           10
Define RANGETHHEME_INFO_METHOD                 11
Define RANGETHHEME_INFO_ROUNDDBY              12

'-----
' Range theme vary style by defines returned by RangeThemeInfo() for
' RANGETHHEME_INFO_VARY_STYLE

'-----
Define RANGETHHEME_INFO_VARY_BY_ALL            0
Define RANGETHHEME_INFO_VARY_BY_COLOR          1
Define RANGETHHEME_INFO_VARY_BY_SIZE           2

'-----
' RangeThemeItemInfo() defines
'-----
Define RANGETHHEME_ITEM_INFO_LINE              1
Define RANGETHHEME_ITEM_INFO_PEN               2
Define RANGETHHEME_ITEM_INFO_BRUSH             3
Define RANGETHHEME_ITEM_INFO_SYMBOL            4
Define RANGETHHEME_ITEM_INFO_FROM              5
Define RANGETHHEME_ITEM_INFO_TO                6

'-----
' IndividualThemeInfo() defines
'-----
Define INDIVTHEME_INFO_ITEMS_COUNT             1
Define INDIVTHEME_INFO_VARY_STYLE              2
Define INDIVTHEME_INFO_REPLACE_STYLE           3
Define INDIVTHEME_INFO_OTHERS_LINE             4
Define INDIVTHEME_INFO_OTHERS_PEN              5
Define INDIVTHEME_INFO_OTHERS_BRUSH            6
Define INDIVTHEME_INFO_OTHERS_SYMBOL           7

'-----
' Individual theme vary style by defines returned by IndividualThemeInfo()
' for INDIVTHEME_INFO_VARY_STYLE

'-----
Define INDIVTHEME_INFO_VARY_BY_ALL             0
Define INDIVTHEME_INFO_VARY_BY_COLOR           1
Define INDIVTHEME_INFO_VARY_BY_SIZE            2

'-----
' IndividualThemeItemInfo() defines
'-----
Define INDIVTHEME_ITEM_INFO_LINE               1

```

```

Define INDIVTHEME_ITEM_INFO_PEN                2
Define INDIVTHEME_ITEM_INFO_BRUSH              3
Define INDIVTHEME_ITEM_INFO_SYMBOL             4
Define INDIVTHEME_ITEM_INFO_VALUE             5

'-----
' DotDensityThemeInfo() defines
'-----
Define DOTDENSITYTHEME_INFO_CONCE              1
Define DOTDENSITYTHEME_INFO_SQUARE            2
Define DOTDENSITYTHEME_INFO_SIZE              3
Define DOTDENSITYTHEME_INFO_COLOR            4

'-----
' GraduatedThemeInfo() defines
'-----
Define GRADUATEDTHEME_INFO_SYMBOL              1
Define GRADUATEDTHEME_INFO_VALUE              2
Define GRADUATEDTHEME_INFO_SHOW_NEG          3
Define GRADUATEDTHEME_INFO_SHOW_NEG_SY      4
Define GRADUATEDTHEME_INFO_GRADUATION        5

'-----
' Graduated theme graduation defines returned by GraduatedThemeInfo()
for
' GRADUATEDTHEME_INFO_GRADUATION
'-----
Define GRADUATEDTHEME_INFO_GRAD_ROOT          0
Define GRADUATEDTHEME_INFO_GRAD_CONST        1
Define GRADUATEDTHEME_INFO_GRAD_LOG          2

'-----
' BarChartThemeInfo() defines
'-----
Define BARCHARTTHEME_INFO_GRADUATION          1
Define BARCHARTTHEME_INFO_ORIENTATION        2
Define BARCHARTTHEME_INFO_PEN                3
Define BARCHARTTHEME_INFO_CHART_TYPE         4
Define BARCHARTTHEME_INFO_GRAD_STACKS        5
Define BARCHARTTHEME_INFO_INDEP_SCALES       6
Define BARCHARTTHEME_INFO_HEIGHT             7
Define BARCHARTTHEME_INFO_HEIGHT_UNIT        8
Define BARCHARTTHEME_INFO_HEIGHT_VALUE       9
Define BARCHARTTHEME_INFO_WIDTH             10
Define BARCHARTTHEME_INFO_WIDTH_UNIT         11
Define BARCHARTTHEME_INFO_BRUSH              12

'-----
' Bar chart theme graduation defines returned by BarChartThemeInfo() for
' BARCHARTTHEME_INFO_GRADUATION
'-----

```

```

Define BARChartTHEME_INFO_GRAD_ROOT          0
Define BARChartTHEME_INFO_GRAD_CONST         1
Define BARChartTHEME_INFO_GRAD_LOG           2

'-----
' Bar chart theme orientation defines returned by BarChartThemeInfo()
for
' BARChartTHEME_INFO_ORIENTATION
'-----
Define BARChartTHEME_INFO_ORI_CC              0
Define BARChartTHEME_INFO_ORI_TL              1
Define BARChartTHEME_INFO_ORI_TC              2
Define BARChartTHEME_INFO_ORI_TR              3
Define BARChartTHEME_INFO_ORI_CL              4
Define BARChartTHEME_INFO_ORI_CR              5
Define BARChartTHEME_INFO_ORI_BL              6
Define BARChartTHEME_INFO_ORI_BC              7
Define BARChartTHEME_INFO_ORI_BR              8

'-----
' Bar chart theme chart type defines returned by BarChartThemeInfo()
for
' BARChartTHEME_INFO_CHART_TYPE
'-----
Define BARChartTHEME_INFO_CHART_STA            0
Define BARChartTHEME_INFO_CHART_MUL            1

'-----
' BarChartThemeItemInfo() defines
'-----
Define BARChartTHEME_ITEM_INFO_BRUSH           1

'-----
' PieChartThemeInfo() defines
'-----
Define PIEChartTHEME_INFO_GRADUATION           1
Define PIEChartTHEME_INFO_ORIENTATION           2
Define PIEChartTHEME_INFO_PEN                   3
Define PIEChartTHEME_GRADUATED                   4
Define PIEChartTHEME_HALF_PIE                     5
Define PIEChartTHEME_DIAMETER                     6
Define PIEChartTHEME_DIAMETER_UNIT                 7
Define PIEChartTHEME_DIAMETER_VALUE                 8
Define PIEChartTHEME_START_ANGLE                   9
Define PIEChartTHEME_CLOCKWISE                    10

'-----
' Bar chart theme graduation defines returned by PieChartThemeInfo() for
' PIEChartTHEME_INFO_GRADUATION
'-----
Define PIEChartTHEME_INFO_GRAD_ROOT            0
Define PIEChartTHEME_INFO_GRAD_CONST            1

```

```

Define PIECHARTTHEME_INFO_GRAD_LOG                2

'-----
' Bar chart theme orientation defines returned by PieChartThemeInfo()
for
' PIECHARTTHEME_INFO_ORIENTATION
'-----
Define PIECHARTTHEME_INFO_ORI_CC                    0
Define PIECHARTTHEME_INFO_ORI_TL                    1
Define PIECHARTTHEME_INFO_ORI_TC                    2
Define PIECHARTTHEME_INFO_ORI_TR                    3
Define PIECHARTTHEME_INFO_ORI_CL                    4
Define PIECHARTTHEME_INFO_ORI_CR                    5
Define PIECHARTTHEME_INFO_ORI_BL                    6
Define PIECHARTTHEME_INFO_ORI_BC                    7
Define PIECHARTTHEME_INFO_ORI_BR                    8

'-----
' PieChartThemeItemInfo() defines
'-----
Define PIECHARTTHEME_ITEM_INFO_BRUSH                1

'-----
' RasterTableInfo() defines
'-----
Define RASTER_TAB_INFO_IMAGE_NAME                  1
Define RASTER_TAB_INFO_WIDTH                        2
Define RASTER_TAB_INFO_HEIGHT                      3
Define RASTER_TAB_INFO_IMAGE_TYPE                  4
Define RASTER_TAB_INFO_BITS_PER_PIXEL              5
Define RASTER_TAB_INFO_IMAGE_CLASS                 6
Define RASTER_TAB_INFO_NUM_CONTROL_POINTS          7
Define RASTER_TAB_INFO_BRIGHTNESS                 8
Define RASTER_TAB_INFO_CONTRAST                    9
Define RASTER_TAB_INFO_GREYSCALE                   10
Define RASTER_TAB_INFO_DISPLAY_TRANSPARENT         11
Define RASTER_TAB_INFO_TRANSPARENT_COLOR           12
Define RASTER_TAB_INFO_ALPHA                       13
Define RASTER_TAB_INFO_HANDLERNAME                 14
Define RASTER_TAB_INFO_LEGEND                     15
Define RASTER_TAB_INFO_LEGEND_XML                  16
Define RASTER_TAB_INFO_HANDLERFULLPATH             17
Define RASTER_TAB_INFO_INFLECTIONS                 18

'-----
' Image type defines returned by RasterTableInfo() for
RASTER_TAB_INFO_IMAGE_TYPE
'-----
Define IMAGE_TYPE_RASTER                            0
Define IMAGE_TYPE_GRID                              1

'-----

```



```
' Image class defines returned by RasterTableInfo() for
RASTER_TAB_INFO_IMAGE_CLASS
```

```
'-----
Define IMAGE_CLASS_BILEVEL          0
Define IMAGE_CLASS_GREYSCALE        1
Define IMAGE_CLASS_PALETTE          2
Define IMAGE_CLASS_RGB               3
```

```
'-----
' GridTableInfo() defines
'-----
```

```
Define GRID_TAB_INFO_MIN_VALUE      1
Define GRID_TAB_INFO_MAX_VALUE      2
Define GRID_TAB_INFO_HAS_HILLSHADE  3
```

```
'-----
' ControlPointInfo() defines
'-----
```

```
Define RASTER_CONTROL_POINT_X       1
Define RASTER_CONTROL_POINT_Y       2
Define GEO_CONTROL_POINT_X          3
Define GEO_CONTROL_POINT_Y          4
Define TAB_GEO_CONTROL_POINT_X      5
Define TAB_GEO_CONTROL_POINT_Y      6
```

```
'-----
' UndoInfo() defines
'-----
```

```
Define UNDO_INFO_COUNT              1
Define UNDO_INFO_UNDO_ACTION_NAME   2
Define UNDO_INFO_REDO_ACTION_NAME   3
Define UNDO_INFO_CAN_UNDO           4
Define UNDO_INFO_CAN_REDO           5
Define UNDO_INFO_ENABLED            6
Define UNDO_INFO_CURRENT_POS        7
```

```
'=====
' WindowInfo() defines
'=====
```

```
Define WIN_INFO_NAME                1
Define WIN_INFO_TYPE                3
Define WIN_INFO_WIDTH               4
Define WIN_INFO_HEIGHT              5
Define WIN_INFO_X                   6
Define WIN_INFO_Y                   7
Define WIN_INFO_TOPMOST             8
Define WIN_INFO_STATE               9
Define WIN_INFO_TABLE              10
Define WIN_INFO_LEGENDS_MAP         10
Define WIN_INFO_ADORNMENTS_MAP     10
Define WIN_INFO_ADORNMENTS_MAP     10
Define WIN_INFO_OPEN                11
Define WIN_INFO_WND                12
```

```

Define WIN_INFO_WINDOWID 13
Define WIN_INFO_WORKSPACE 14
Define WIN_INFO_CLONEWINDOW 15
Define WIN_INFO_SYSMENUCLOSE 16
Define WIN_INFO_AUTOSCROLL 17
Define WIN_INFO_SMARTPAN 18
Define WIN_INFO_SNAPMODE 19
Define WIN_INFO_SNAPTHRESHOLD 20
Define WIN_INFO_PRINTER_NAME 21
Define WIN_INFO_PRINTER_ORIENT 22
Define WIN_INFO_PRINTER_COPIES 23
Define WIN_INFO_PRINTER_PAPERSIZE 24
Define WIN_INFO_PRINTER_LEFTMARGIN 25
Define WIN_INFO_PRINTER_RIGHTMARGIN 26
Define WIN_INFO_PRINTER_TOPMARGIN 27
Define WIN_INFO_PRINTER_BOTTOMMARGIN 28
Define WIN_INFO_PRINTER_BORDER 29
Define WIN_INFO_PRINTER_TRUECOLOR 30
Define WIN_INFO_PRINTER_DITHER 31
Define WIN_INFO_PRINTER_METHOD 32
Define WIN_INFO_PRINTER_TRANSPRASTER 33
Define WIN_INFO_PRINTER_TRANSPVECTOR 34
Define WIN_INFO_EXPORT_BORDER 35
Define WIN_INFO_EXPORT_TRUECOLOR 36
Define WIN_INFO_EXPORT_DITHER 37
Define WIN_INFO_EXPORT_TRANSPRASTER 38
Define WIN_INFO_EXPORT_TRANSPVECTOR 39
Define WIN_INFO_PRINTER_SCALE_PATTERNS 40
Define WIN_INFO_EXPORT_ANTI_ALIASING 41
Define WIN_INFO_EXPORT_THRESHOLD 42
Define WIN_INFO_EXPORT_MASKSIZE 43
Define WIN_INFO_EXPORT_FILTER 44
Define WIN_INFO_ENHANCED_RENDERING 45
Define WIN_INFO_SMOOTH_TEXT 46
Define WIN_INFO_SMOOTH_IMAGE 47
Define WIN_INFO_SMOOTH_VECTOR 48
Define WIN_INFO_PARENT_LAYOUT 49
Define WIN_INFO_CLIENTWIDTH 50
Define WIN_INFO_CLIENTHEIGHT 51
Define WIN_INFO_PRINTER_RESOLUTION 52
Define WIN_INFO_TOPMOSTWHENFLOATING 53

```

```

'-----
' Window types, returned by WindowInfo() for WIN_INFO_TYPE
'-----

```

```

Define WIN_MAPPER 1
Define WIN_BROWSER 2
Define WIN_LAYOUT 3
Define WIN_GRAPH 4
Define WIN_BUTTONPAD 19
Define WIN_TOOLBAR 25
Define WIN_CART_LEGEND 27
Define WIN_3DMAP 28

```

```

Define WIN_ADORNMENT                32
Define WIN_LEGEND_DESIGNER          35
Define WIN_LAYOUT_DESIGNER          36
Define WIN_CUSTOM                    37
Define WIN_HELP                     1001
Define WIN_MAPBASIC                  1002
Define WIN_MESSAGE                   1003
Define WIN_RULER                     1007
Define WIN_INFO                     1008
Define WIN_LEGEND                    1009
Define WIN_STATISTICS                1010
Define WIN_MAPINFO                   1011

'-----
' Version 2 window types no longer used in version 3 or later versions
'-----
Define WIN_TOOLPICKER                1004
Define WIN_PENPICKER                 1005
Define WIN_SYMBOLPICKER              1006

'-----
' Window types which can be used in Open/Close Window statements
'-----
Define WIN_TABLE_LIST                2001
Define WIN_LAYER_CONTROL             2002
Define WIN_MOVE_MAP_TO              2003
Define WIN_WORKSPACE_EXPLORER       2004
Define WIN_WINDOW_LIST              2005
Define WIN_TOOL_MANAGER              2006
Define WIN_TASK_MANAGER              2007
Define WIN_CONNECTION_LIST          2008
Define WIN_NOTIFICATIONS             2009
Define WIN_SQL                       2010

'-----
' Window states, returned by WindowInfo() for WIN_INFO_STATE
'-----
Define WIN_STATE_NORMAL              0
Define WIN_STATE_MINIMIZED           1
Define WIN_STATE_MAXIMIZED           2
Define WIN_STATE_DOCKED              3
Define WIN_STATE_TABBED              4
Define WIN_STATE_AUTOHIDDEN          5
Define WIN_STATE_HIDDEN              6

'-----
' Print orientation, returned by WindowInfo() for WIN_INFO_PRINTER_ORIENT
'-----
Define WIN_PRINTER_PORTRAIT          1
Define WIN_PRINTER_LANDSCAPE         2

'-----
' Antialiasing filters, returned by WindowInfo() for

```

```

WIN_INFO_EXPORT_FILTER
'-----
Define FILTER_VERTICALLY_AND_HORIZONTALLY      0
Define FILTER_ALL DIRECTIONS_1                  1
Define FILTER_ALL DIRECTIONS_2                  2
Define FILTER_DIAGONALLY                        3
Define FILTER_HORIZONTALLY                      4
Define FILTER_VERTICALLY                        5

'-----
' GetPreference() defines
'-----

'-----
' Preference segment defines
'-----
Define PREFERENCE_SYSTEM                        0
Define PREFERENCE_TABLES                       1
Define PREFERENCE_BROWSER                      3
Define PREFERENCE_MAPPER                      4
Define PREFERENCE_INTERNATIONAL                6
Define PREFERENCE_PATHS                       11
Define PREFERENCE_LEGEND                      12
Define PREFERENCE_OUTPUT                      13
Define PREFERENCE_PRINTER                     14
Define PREFERENCE_WEBSERVICES                 15
Define PREFERENCE_IMAGE_PROCESSING            16
Define PREFERENCE_LAYOUT                      17
Define PREFERENCE_PDF                         18
Define PREFERENCE_PERFORMANCE                 19
Define PREFERENCE_MANAGED_WORKSPACE           50

'-----
' System Preference defines
' Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_SYSTEM_COLORDEFAULT                0 ' 0 = Monitor, 1=
Black&White, 2 = Color
Define PREF_SYSTEM_AUTOSAVEWORKSPACE           3 ' 0=OFF, 1=ON
Define PREF_SYSTEM_COPYTEXTCLIPBOARD           7 ' 0=OFF, 1=ON
Define PREF_SYSTEM_COPYGRAPHICSClipboard       8 ' 0=OFF, 1=ON
Define PREF_SYSTEM_PAPERUNITS                   9 ' 0=OFF, 1=ON
Define PREF_SYSTEM_UNDO_COUNT                  10 ' number of Undo
objects
Define PREF_SYSTEM_SUBSTITUTETTFONT            12 ' Display
Pre-version 4 symbols
Define PREF_SYSTEM_COPY_EMF_CLIPBOARD          13 ' 0=OFF, 1=ON
Define PREF_SYSTEM_COPY_OLEOBJECT              14 ' 0=OFF, 1=ON
Define PREF_SYSTEM_SAVEQUERYWORKSPACE          18 ' 0=OFF, 1=ON
Define PREF_SYSTEM_DATEWINDOW                  19 ' -1 = Use Current
Century, else Year value
Define PREF_SYSTEM_UNDO_SIZE                    20 ' Integer, Memory

```

```

size in bytes for Undo
Define PREF_SYSTEM_RESOLUTION          21 ' Export resolution
in dpi
Define PREF_SYSTEM_VMOPTION            22 ' 0 = GRID, 1=
Raster, 2 = Default
Define PREF_SYSTEM_DISTANCEUNITS      23 ' Returns Distance
Unit define
Define PREF_SYSTEM_AREAUNITS          24 ' Returns Area
Units define
Define PREF_SYSTEM_COPY_OLEOBJECT_NEW  25 ' 0=OFF, 1=ON
Define PREF_SYSTEM_COPY_ENHANCED_EMF_TYPE 26 ' 0 = EMF, 1=
EMF+, 2 = EMF+Dual
Define PREF_SYSTEM_CROSSHAIRCURSOR    28 ' 138 = Medium
crosshair, 164 = Large, 165 = Default
Define PREF_SYSTEM_COPY_WMF_CLIPBOARD 29 ' 0=OFF, 1=ON
Define PREF_SYSTEM_TRUECOLORCURSORS  30 ' 0=OFF, 1=ON
Define PREF_SYSTEM_WRITETOOLWINDOWSTOWORKSPACE 31 ' 0=OFF, 1=ON
Define PREF_SYSTEM_MULTIPLETOUCH      32 ' support
touchscreen
Define PREF_SYSTEM_DEFAULTFILEFORMAT  33 ' 5 = MapInfo
Native, 14 = MapInfo Extended
Define PREF_SYSTEM_DEFAULTCHARSETFORNATIVEX 34 ' String - Default
charset used for new tables
Define PREF_SYSTEM_WRITEUNICODEDATA   35 ' Encode workspaces
and Tab files

'-----
' Workspace Preference defines
'-----
Define PREF_WKS_PROMPTTOSAVE          1 ' 0=OFF, 1=ON
Define PREF_WKS_PROMPTTOSAVEOPTION    2 ' 0 = Automatic,
1 = Only when WOR is open

'-----
' Table Preference defines
'-----
Define PREF_TABLES_FILEOPENWHAT       0 ' Last Preferred
view
Define PREF_TABLES_OPENCHOICESCREEN   1 ' QuickStart
Define PREF_TABLES_CONNNNAME          2 ' String: Default
DBMS Connection
Define PREF_TABLES_CONNDRIVER         3 ' String: DBMS
Driver
Define PREF_TABLES_CONNSTRING         4 ' String: Connection
string

'-----
' Browser Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_BROWSER_RDTSORTORDER      0 ' Redistrict Sort
Order: 0=MRU, 1=Alpha,2=Not Ordered

```

```

Define PREF_BROWSER_ALTROWCOLORRED          1 ' RGB color Red
1-255
Define PREF_BROWSER_ALTROWCOLORGREEN        2 ' RGB color Green
1-255
Define PREF_BROWSER_ALTROWCOLORBLUE         3 ' RGB color Blue
1-255

'-----
' Mapper Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----

Define PREF_MAPPER_PRESERVESCALEONRESIZE     0 ' 0=Fit to Map,
1=Preserve Scale
Define PREF_MAPPER_HIGHLIGHT_BRUSH           2 ' Selection Brush
clause
Define PREF_MAPPER_SNAPTONODES               3 ' 0=OFF, 1=ON
Define PREF_MAPPER_MOVEDUPLICATENODES        4 ' 0 = None, 1=
Same Layer
Define PREF_MAPPER_PIXELTOLERANCE            5 ' SmallInt Pixel
tolerance value
Define PREF_MAPPER_TARGET_PEN                7 ' Target Pen clause
Define PREF_MAPPER_TARGET_BRUSH              8 ' Target Brush
clause
Define PREF_MAPPER_HIGHLIGHT_PEN             9 ' Selection Pen
Clause
Define PREF_MAPPER_SCROLLBARS                10 ' 0=OFF, 1=ON
Define PREF_MAPPER_SAVECOSMETIC              11 ' 0=OFF, 1=ON
Define PREF_MAPPER_SAVELABELS                12 ' 0=OFF, 1=ON
Define PREF_MAPPER_SAVETHEMATIC              13 ' 0=OFF, 1=ON
Define PREF_MAPPER_SHOWINFOTIPS              14 ' 0=OFF, 1=ON
Define PREF_MAPPER_DISTANCETYPE              15 ' 0=Spherical,
1=Cartesian
Define PREF_MAPPER_SHOWDEGREESMINUTESSECONDS 16 ' See
MAPPER_INFO_DISPLAY_DMS return values
Define PREF_MAPPER_LASTTHEMETYPE             17 ' String: Last
Default theme type used
Define PREF_MAPPER_GRIDOUTPUTFORMATHANDLER   18 ' String: Handler
for New Grids.
Define PREF_MAPPER_CLIPREGIONTYPE            19 ' SmallInt: returns
MAPPER_INFO_CLIP_TYPE
Define PREF_MAPPER_3DHARDWARE                 20 ' 0=OFF, 1=ON
Define PREF_MAPPER_AUTORASTERZOOMLAYER        21 ' 0=OFF, 1=ON
Define PREF_MAPPER_AUTOGRIDZOOMLAYER          22 ' 0=OFF, 1=ON
Define PREF_MAPPER_POINT_HIGHLIGHT_BRUSH      23 ' Target Pen
Clause for MPoints
Define PREF_MAPPER_POINT_TARGET_BRUSH         24 ' Target Brush
Clause for MPoints
Define PREF_MAPPER_AUTOSCROLL                 25 ' 0=OFF, 1=ON
Define PREF_MAPPER_DEFAULTPROJECTION          26 ' String: Table
Projection Preference
Define PREF_MAPPER_DEFAULTMBPROJECTION        27 ' String: Session
Projection Preference

```

```

Define PREF_MAPPER_DISPLAYSNAPRADIUS          28 ' 0=OFF, 1=ON
Define PREF_MAPPER_CARTOGRAPHICSCALE          29 ' 0=OFF, 1=ON
Define PREF_MAPPER_DISPLAYLAYERCONTROLTOOLTIPS 30 ' 0=OFF, 1=ON
Define PREF_MAPPER_AUTONODETOLERANCE          31 ' SmallInt - auto
  node tolerance value
Define PREF_MAPPER_DRAWLAYERSUNDERTHEMES      32 ' 0=OFF, 1=ON
Define PREF_MAPPER_READDEFAULTTHEMEMETADATA   33 ' 0=OFF, 1=ON
Define PREF_MAPPER_FINDSELECTIONAFTERPASTE    34 ' 0=OFF, 1=ON
Define PREF_MAPPER_ZOOMONFINDSELECTION        35 ' 0=OFF, 1=ON
Define PREF_MAPPER_ALPHAHIGHLIGHT             36 ' SmallInt: 255
=100% opaque, 0= transparent
Define PREF_MAPPER_ZOOMMOUSEWHEELONCURSOR      37 ' 0=OFF, 1=ON
Define PREF_MAPPER_ADORNAUTOMATICSCALING      38 ' 0=OFF, 1=ON
Define PREF_MAPPER_ADORN_BACKGROUND_BRUSH     39 ' Brush clause
for Adornment backgrounds
Define PREF_MAPPER_ADORN_BACKGROUND_PEN       40 ' Pen clause for
  Adornment backgrounds

'-----
' International Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_INTERNATIONAL_ADDRMATCH           0 ' 0=House number
BEFORE Street, 1=House number AFTER street
Define PREF_INTERNATIONAL_INSET               1 ' Inset value in
Distance Units
Define PREF_INTERNATIONAL_OFFSET              2 ' Offset value in
Distance Units
Define PREF_INTERNATIONAL_DISTUNIT            3 ' Returns define
for Distance units
Define PREF_INTERNATIONAL_ENABLEDISTANCESPECIFIED 4 ' Inset address
percentage value
Define PREF_INTERNATIONAL_INSET_DIST          5 ' Inset Distance
value
Define PREF_INTERNATIONAL_INSET_DIST_DISTUNIT 6 ' Returns define
for Distance units
Define PREF_INTERNATIONAL_USER_CLEAN           7 ' Substitutions
from MAPINFOW.ABB file
Define PREF_INTERNATIONAL_CLOSE_ADDR          8 ' Use Closest
Address
Define PREF_INTERNATIONAL_ALT_BDY             9 ' Match in alternate
boundary

'-----
' Path Preference defines
'Most return a string for directory paths unless otherwise indicated
'-----
Define PREF_PATH_MITABLEPATH                  0 ' New Tables Path
Define PREF_PATH_MIWORKSPACEPATH              1 ' New Workpsace
path
Define PREF_PATH_MIPROGRAMPATH                2 ' MapBasic Programs
path

```

```

Define PREF_PATH_MIDATAPATHS                      3 ' Alternate Search
  path if PREF_PATH_USETRAVERSAL = 1
Define PREF_PATH_USETRAVERSAL                      4 ' SmallInt: 0=No
  alternate search path, 1=Use alternate search path
Define PREF_PATH_CUSTOMSYMBOLPATH                  5 ' Path to custom
  symbols folder
Define PREF_PATH_IMPORTPATH                          6 ' Import files
  path
Define PREF_PATH_SQLQUERYPATH                       7 ' SQL Query path
Define PREF_PATH_THEMETEMPLATEPATH                  8 ' Theme templates
  path
Define PREF_PATH_MIQUERYPATH                        9 ' Saved Queries
  path
Define PREF_PATH_NEWGRIDPATH                       10 ' New Grid Files
Define PREF_PATH_REMOTETABLEPATH                   13 ' DBMS tables
  path
Define PREF_PATH_TABLESFROMSHAPEFILES               14 ' SHP table path
Define PREF_PATH_TABLESFROMWFS                     15 ' WFS table path
Define PREF_PATH_TABLESFROMWMS                     16 ' WMS table path
Define PREF_PATH_LAYOUTIMAGES                      17 ' Layout Image
  files path

'-----
' Legend Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_LEGEND_FNTTITLE                         0 ' Font Clause for
  Legend title
Define PREF_LEGEND_FNTSUBTITLE                      1 ' Font Clause for
  Legend Subtitle
Define PREF_LEGEND_FNTSTYLE                         2 ' Font Clause for
  Legend categories
Define PREF_LEGEND_PENBORDER                        3 ' Pen clause for
  legend frames
Define PREF_LEGEND_TITLEPATTERN                    4 ' Background brush
  clause
Define PREF_LEGEND_SUBTITLEPATTERN                 5 ' Background brush
  clause
Define PREF_LEGEND_STYLEPATTERN                    6 ' Background brush
  clause
Define PREF_LEGEND_STYLESIZE                        7 ' 0=Small, 1=Large
  swatches
Define PREF_LEGEND_PROMPTSAVEWORKSPACE             8 ' 0=OFF, 1=ON
Define PREF_LEGEND_CREATEPRE1150LEGENDS           9 ' 0=OFF, 1=ON
Define PREF_LEGEND_PAPERUNITS                      10 ' Returns
  PREF_SYSTEM_PAPERUNITS
Define PREF_LEGEND_SYNCDISPLAY                     11 ' 0=OFF, 1=ON
Define PREF_LEGEND_LINEWIDTH                       12 ' Sample Line
  width
Define PREF_LEGEND_REGIONWIDTH                     13 ' Sample region
  width
Define PREF_LEGEND_REGIONHEIGHT                    14 ' Sample region

```



```

height
Define PREF_LEGEND_DISPLAYGRID 15 ' 0=OFF, 1=ON
Define PREF_LEGEND_SNAPGRID 16 ' 0=OFF, 1=ON
Define PREF_LEGEND_AUTOUPDATE 17 ' 0=OFF, 1=ON
Define PREF_LEGEND_GRIDCOLORALPHA 18 ' Where 255 =100%
    opaque, 0= transparent
Define PREF_LEGEND_GRIDCOLORRED 19 ' Red Color
    component of RGB value
Define PREF_LEGEND_GRIDCOLORGREEN 20 ' Green
Define PREF_LEGEND_GRIDCOLORBLUE 21 ' Blue
Define PREF_LEGEND_GUIDECOLORALPHA 22 ' SmallInt: 255
    =100% opaque, 0= transparent
Define PREF_LEGEND_GUIDECOLORRED 23 ' Red Color
    component of RGB value
Define PREF_LEGEND_GUIDECOLORGREEN 24 ' Green
Define PREF_LEGEND_GUIDECOLORBLUE 25 ' Blue
Define PREF_LEGEND_AUTOFONTSIZESETTING 26 ' Increase sample
    size with font

'-----
' Output Preference defines
' Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_OUTPUT_USETRUECOLORDISPLAY 0 ' Raster True Color
Define PREF_OUTPUT_DITHERMETHODDISPLAY 1 ' SmallInt:
    1=Halftone, 2 = Error Diffusion
Define PREF_OUTPUT_PRINTMETHOD 2 ' 0=Direct to
    Device, 1=EMF method, 2=OSBM
Define PREF_OUTPUT_PRINTBORDER 3 ' 0=OFF, 1=ON
Define PREF_OUTPUT_TRANSPWHENPRINTVECTOR 5 ' Internal print
    handling for transparent vector
Define PREF_OUTPUT_USETRUECOLORPRINT 6 ' 0=OFF, 1=ON
Define PREF_OUTPUT_DITHERMETHODPRINT 7 ' SmallInt:
    1=Halftone, 2 = Error Diffusion
Define PREF_OUTPUT_EXPORTBORDER 8 ' 0=OFF, 1=ON
Define PREF_OUTPUT_TRANSPWHENEXPORTRASTER 9 ' Internal export
    handling for transparent raster
Define PREF_OUTPUT_TRANSPWHENEXPORTVECTOR 10 ' Internal export
    handling for transparent vector
Define PREF_OUTPUT_USETRUECOLOREXPORT 11 ' 0=OFF, 1=ON
Define PREF_OUTPUT_DITHERMETHODEXPORT 12 ' SmallInt:
    1=Halftone, 2 = Error Diffusion
Define PREF_OUTPUT_ROPTRANSPWHENPRINTINGRASTER 13 ' 0=OFF, 1=ON
Define PREF_OUTPUT_ROPTRANSPWHENEXPORTINGRASTER 14 ' 0=OFF, 1=ON
Define PREF_OUTPUT_SCALEPATTERNSPRINT 15 ' 0=OFF, 1=ON
Define PREF_OUTPUT_USEANTIALIASING 16 ' 0=OFF, 1=ON
Define PREF_OUTPUT_ANTIALIASINGTHRESHOLD 17 ' SmallInt value
    for antialias Threshold
Define PREF_OUTPUT_ANTIALIASINGMASKSIZE 18 ' SmallInt value
    for antialias Mask Size
Define PREF_OUTPUT_ANTIALIASINGFILTER 19 ' SmallInt: returns
    value from WIN_INFO_EXPORT_FILTER

```

```

Define PREF_OUTPUT_DISPLAYPOLYGONMODE          20 ' 0=OFF, 1=ON
Define PREF_OUTPUT_PRINTINGPOLYGONMODE          21 ' 0=OFF, 1=ON
Define PREF_OUTPUT_EXPORTINGPOLYGONMODE          22 ' 0=OFF, 1=ON
Define PREF_OUTPUT_SUBDIVIDEPRINTING            23 ' 0=OFF, 1=ON
Define PREF_OUTPUT_CLARIFYTEXTOVERWYSIWYG       24 ' Export clarify
    text
Define PREF_OUTPUT_DISPLAYPOINTSFAST            25 ' Fast Symbol
rendering

'-----
' Printer Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_PRINTER_USESPECIFICPRINTER          0 ' 0=Windows default
    printer, 1 = MapInfo printer
Define PREF_PRINTER_PAPERSIZE                    1 ' Returns papersize
    define, e.g.: 1=Letter, 2=Legal
Define PREF_PRINTER_ORIENTATION                  2 ' See
WIN_INFO_PRINTER_ORIENT
Define PREF_PRINTER_PRINTERNAME                  3 ' String: Printer
    name
Define PREF_PRINTER_SAVEPRINTERINTOWORKSPACE     4 ' Save printer
info in WOR
Define PREF_PRINTER_RESTOREPRINTERFROMWORKSPACE 5 ' Restore printer
info from WOR

'-----
' WebServices Preference defines
'Most return a SmallInt where 0 = Unchecked/Off and 1 = Checked/On,
unless otherwise indicated
'-----
Define PREF_WEBSERVICES_WFSCONNECTTIMEOUT        0 ' SmallInt: Connect
    timeout in seconds
Define PREF_WEBSERVICES_WFSSENDTIMEOUT           1 ' SmallInt: Send
    timeout in seconds
Define PREF_WEBSERVICES_WFSRECEIVETIMEOUT        2 ' SmallInt: Receive
    timeout in seconds
Define PREF_WEBSERVICES_WMSCONNECTTIMEOUT        3 ' SmallInt: Connect
    timeout in seconds
Define PREF_WEBSERVICES_WMSSENDTIMEOUT           4 ' SmallInt: Send
    timeout in seconds
Define PREF_WEBSERVICES_WMSRECEIVETIMEOUT        5 ' SmallInt: Receive
    timeout in seconds
Define PREF_WEBSERVICES_WMSMAXWIDTH              6 ' SmallInt: WMS
    Max Width in pixels
Define PREF_WEBSERVICES_WMSMAXHEIGHT            7 ' SmallInt: WMS
    Max Height in pixels
Define PREF_WEBSERVICES_ACCESTYPE                8 ' Proxy Access
    type: 0=System, 1=Use Direct, 2=Use Proxy
Define PREF_WEBSERVICES_PROXYSERVER             9 ' Proxy Server IP
Define PREF_WEBSERVICES_PORT                    10 ' Proxy Server
    Port

```

```

Define PREF_WEBSERVICES_BYPASSLOCAL          11 ' Use bypass
Define PREF_WEBSERVICES_BYPASSLIST          12 ' String: bypass
list
Define PREF_WEBSERVICES_WFSREFRESHUSINGMAP    13 ' Update mapper
filters with current mapper
Define PREF_WEBSERVICES_WEBGEOCODECONNECTTIMEOUT 14 ' SmallInt: Connect
timeout in seconds
Define PREF_WEBSERVICES_WEBGEOCODESENDTIMEOUT 15 ' SmallInt: Send
timeout in seconds
Define PREF_WEBSERVICES_WEBGEOCODERECEIVETIMEOUT 16 ' SmallInt: Receive
timeout in seconds
Define PREF_WEBSERVICES_STREETOFFSET          17 ' Offset value
from road
Define PREF_WEBSERVICES_CORNEROFFSET          18 ' Offset value
from corner
Define PREF_WEBSERVICES_UNITOFFSET          19 ' Returns
PREF_SYSTEM_DISTANCEUNITS
Define PREF_WEBSERVICES_GEOCODEDICTIONARY    20 ' SmallInt: See
return values for GeocodeInfo()
Define PREF_WEBSERVICES_GEOCODEBATCHSIZE     21 ' SmallInt: Points
to batch geocode
Define PREF_WEBSERVICES_WEBROUTINGCONNECTTIMEOUT 22 ' SmallInt: Connect
timeout in seconds
Define PREF_WEBSERVICES_WEBROUTINGSENDTIMEOUT 23 ' SmallInt: Send
timeout in seconds
Define PREF_WEBSERVICES_WEBROUTINGRECEIVETIMEOUT 24 ' SmallInt: Receive
timeout in seconds
Define PREF_WEBSERVICES_DRIVEREGIONBATCHSIZE 25 ' SmallInt: Points
to batch driving regions
Define PREF_WEBSERVICES_APPLYMULTIRESULTCODE 26 ' Mark Multiple
matches when automatic mode
Define PREF_WEBSERVICES_READTIMEOUTWEBTILESERVER 27 ' Read timeout
in seconds for tile servers
Define PREF_WEBSERVICES_REQUESTTIMEOUTWEBTILESERVER 28 ' Request timeout
in seconds for tile servers
Define PREF_WEBSERVICES_LIBRARYSERVICEACCESSMODE 29 ' 0=disabled,
1=catalog access, 2=catalog and library access
Define PREF_WEBSERVICES_OPENMANAGEDTABLESEEDITOR 30 ' open record
viewer/editor after saving managed tables
Define PREF_WEBSERVICES_BINGKEYFROMUSER      31 ' String: returns
custom Bing access key if used.

'-----
' Image Processing Preference defines
'-----
Define PREF_IMAGEPROCESSING_REPROJECTIONUSEINDEX 0 ' 0=Never, 1=Always,
2=optimized
Define PREF_IMAGEPROCESSING_RESAMPLEMETHODINDEX 1 ' 0=Cubic
Convolution, 1=Nearest Neighbor

'-----
' Layout Preference defines
'-----

```

```

Define PREF_LAYOUT_PROMPTSAVEWORKSPACE          3 ' 0=OFF, 1=ON

'-----
' Pdf Preference defines
'-----
Define PREF_PDF_LAYEREDPDF                        0 ' 0=OFF, 1=ON
Define PREF_PDF_SAVEGEOINFO                      1 ' 0=OFF, 1=ON
Define PREF_PDF_SAVEATTRIBUTEDATA                2 ' 0=OFF, 1=ON
Define PREF_PDF_AUTOOPENPDF                      3 ' 0=OFF, 1=ON
Define PREF_PDF_POSTSCRIPTSIM                    4 ' Use PDF Advanced
option: Enable Postscript Simulation
Define PREF_PDF_SAVEGEOINFOINNATIVEUNITS          5 ' Georegistration
Coordinate Units: 0=Map Units, 1=Degrees

'-----
' Performance Preference defines
'-----
Define PREF_PERFORMANCE_CONCURRENCYSETTING        0 '
0=None,1=Moderate,2=Intermediate,3=Aggressive,4=Full
Define PREF_PERFORMANCE_DROPINDEXSETTING          1 ' SmartIndexing:
0=Auto,1=On,2=Off

'=====
' Abbreviated list of error codes
'
' The following are error codes described in the Reference manual. All
' other errors are listed in ERRORS.DOC.
'=====
Define ERR_BAD_WINDOW                            590
Define ERR_BAD_WINDOW_NUM                        648
Define ERR_CANT_INITIATE_LINK                    698
Define ERR_CMD_NOT_SUPPORTED                     642
Define ERR_FCN_ARG_RANGE                         644
Define ERR_FCN_INVALID_FMT                       643
Define ERR_FCN_OBJ_FETCH_FAILED                  650
Define ERR_FILEMGR_NOTOPEN                       366
Define ERR_FP_MATH_LIB_DOMAIN                    911
Define ERR_FP_MATH_LIB_RANGE                     912
Define ERR_INVALID_CHANNEL                       696
Define ERR_INVALID_READ_CONTROL                  842
Define ERR_INVALID_TRIG_CONTROL                  843
Define ERR_NO_FIELD                              319
Define ERR_NO_RESPONSE_FROM_APP                  697
Define ERR_PROCESS_FAILED_IN_APP                 699
Define ERR_NULL_SELECTION                        589
Define ERR_TABLE_NOT_FOUND                       405
Define ERR_WANT_MAPPER_WIN                       313
Define ERR_CANT_ACCESS_FILE                      825

'=====
' Backward Compatibility defines
'
' These defines are provided so that existing MapBasic code will continue

```

```
' to compile & run correctly. Please use the new define (on the right)
' when writing new code.
```

```
=====
Define OBJ_ARC                      OBJ_TYPE_ARC
Define OBJ_ELLIPSE                  OBJ_TYPE_ELLIPSE
Define OBJ_LINE                     OBJ_TYPE_LINE
Define OBJ_PLINE                    OBJ_TYPE_PLINE
Define OBJ_POINT                    OBJ_TYPE_POINT
Define OBJ_FRAME                    OBJ_TYPE_FRAME
Define OBJ_REGION                   OBJ_TYPE_REGION
Define OBJ_RECT                     OBJ_TYPE_RECT
Define OBJ_ROUNDRECT                OBJ_TYPE_ROUNDRECT
Define OBJ_TEXT                     OBJ_TYPE_TEXT
```

```
'-----
' Codes for AdornmentInfo function to get info about an adornment win
'-----
```

```
Define ADORNMENT_INFO_TYPE          1
Define ADORNMENT_INFO_MAP_WINDOWID  2
Define ADORNMENT_INFO_IS_FIXED_POS  3
Define ADORNMENT_INFO_FIXED_POS_X   4
Define ADORNMENT_INFO_FIXED_POS_Y   5
Define ADORNMENT_INFO_FIXED_POS_UNITS 6
Define ADORNMENT_INFO_DOCKED_POS     7
Define ADORNMENT_INFO_DOCKED_OFFSET_X 8
Define ADORNMENT_INFO_DOCKED_OFFSET_Y 9
Define ADORNMENT_INFO_DOCKED_UNITS   10
Define ADORNMENT_INFO_BACKGROUND_PEN 11
Define ADORNMENT_INFO_BACKGROUND_BRUSH 12
Define ADORNMENT_INFO_SB_TYPE        20
Define ADORNMENT_INFO_SB_MAP_UNITS   21
Define ADORNMENT_INFO_SB_PAPER_UNITS 22
Define ADORNMENT_INFO_SB_BAR_LENGTH  23
Define ADORNMENT_INFO_SB_BAR_DRAW_LEN 24
Define ADORNMENT_INFO_SB_BAR_HEIGHT  25
Define ADORNMENT_INFO_SB_AUTO_SCALING 26
Define ADORNMENT_INFO_SB_CARTO_SCALE  27
Define ADORNMENT_INFO_SB_BAR_PEN      28
Define ADORNMENT_INFO_SB_BAR_BRUSH    29
Define ADORNMENT_INFO_SB_BAR_FONT     30
Define ADORNMENT_INFO_SB_DISPLAY_SCALE 31
Define ADORNMENT_INFO_SB_AUTOOFF_SCALE 32
Define ADORNMENT_INFO_SB_AUTOON_SCALE  33
Define ADORNMENT_INFO_SB_SCALE_STRING  34
Define ADORNMENT_INFO_SB_CARTO_VALUE   35
Define ADORNMENT_INFO_SB_CARTO_STRING  36
```

```
'=====
' Codes used to position Adornments relative to mapper
'=====
```

```
Define ADORNMENT_INFO_MAP_POS_TL    0
Define ADORNMENT_INFO_MAP_POS_TC    1
```

```

Define ADORNMENT_INFO_MAP_POS_TR      2
Define ADORNMENT_INFO_MAP_POS_CL      3
Define ADORNMENT_INFO_MAP_POS_CC      4
Define ADORNMENT_INFO_MAP_POS_CR      5
Define ADORNMENT_INFO_MAP_POS_BL      6
Define ADORNMENT_INFO_MAP_POS_BC      7
Define ADORNMENT_INFO_MAP_POS_BR      8
Define SCALEBAR_INFO_BARTYPE_CHECKEDBAR 0
Define SCALEBAR_INFO_BARTYPE_SOLIDBAR   1
Define SCALEBAR_INFO_BARTYPE_LINEBAR    2
Define SCALEBAR_INFO_BARTYPE_TICKBAR    3

'=====
' Coordinate system datum id's.  These match the id's from mapinfow.prj.
'=====
Define DATUMID_NAD27  62
Define DATUMID_NAD83  74
Define DATUMID_WGS84 104

'=====
' Assigning nullptr to c# ref type will destroy and allow GC to collect
them.
'=====
Define NULL_PTR      -1

'=====
' end of MAPBASIC.DEF
'=====

```

F - Введение в справку MapBasic

Добро пожаловать в *Справку MapBasic*, источник информации об операторах функциях и командах языка MapBasic.

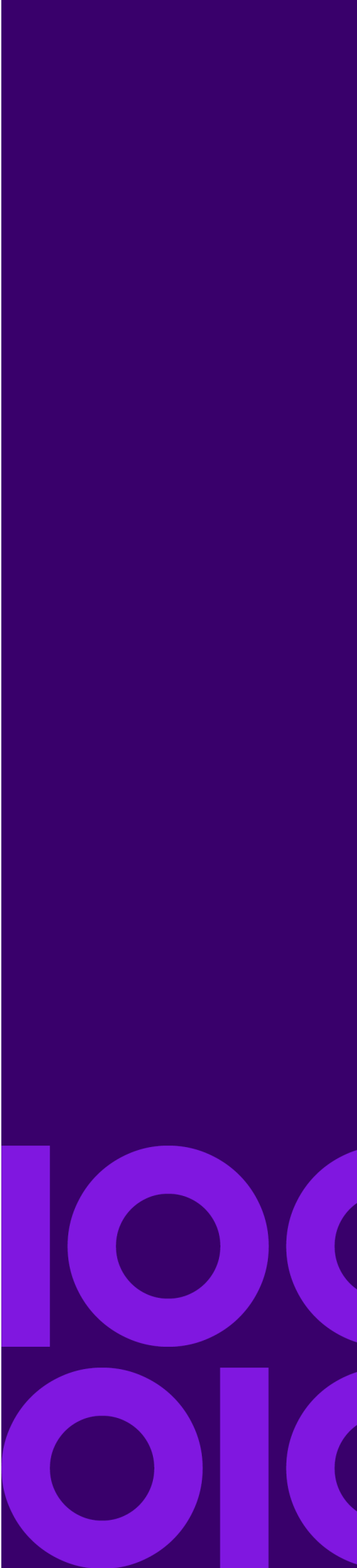
Если вы только начали работать с языком MapBasic, ознакомьтесь со следующими разделами:

- **Введение в MapBasic**
- **Новые и улучшенные операторы и функции MapBasic** на стр. 37
- **Сведения об операторах** на стр. 1222
- **Библиотеки HTTP и FTP** на стр. 1154
- **Библиотека XML** на стр. 1195

В этом разделе:

В этом разделе

Поиск раздела в справочной системе.....	1280
Справочная система.....	1280
О программе просмотра справки.....	1281
Просмотр справки на другом компьютере.....	1281
Команды меню.....	1282
Описание диалогов MapBasic.....	1301



Поиск раздела в справочной системе

Справочная система MapBasic позволяет быстро находить описания команд меню и диалогов.

Можно получить интересующую вас информацию несколькими способами:

1. используйте закладку **Содержание** и выберите интересующий вас раздел;
2. для быстрого поиска раздела воспользуйтесь закладкой **Указатель**. Введите в пустое поле первые буквы искомого слова. Будет выведен раздел, первые буквы названия которого наиболее полно совпадают с тем, что вы напечатали.
3. Для поиска по указанному слову используйте закладку **Поиск**. В пустое поле введите искомое слово (при этом MapInfo Pro выведет список слов, близких по своему написанию к искомому).

Для того, чтобы скопировать часть текста **Справки**, выберите текст, нажмите правую кнопку мыши и выберите в всплывающем меню команду **Копировать**.

Справочная система

MapBasic имеет несколько справочных систем:

- MapBasicHelp.exe содержит описание работы с функциями и операторами MapBasic, которое доступно из ленты MapBasic.
- ExtensibilityReferenceHelp.exe описывает объектную модель .NET API, которая позволяет настраивать и создавать элементы интерфейса пользователя, например, вкладки на ленте, кнопки, раскрывающиеся списки и галереи.
- IntegratedMappingReferenceHelp.exe описывает MapBasic Software Development Kit (SDK), который добавляет картографические функции в приложения.

Для того чтобы воспользоваться справочной системой за пределами программы, дважды щёлкните по файлу справки .exe, который находится в папке, где установлена программа MapBasic. Также можно скопировать этот файл в другое место на компьютере, например, на рабочий стол и запустить его отсюда.

О программе просмотра справки

MapBasic использует QT Assistant для просмотра справки. Подробнее об использовании программы просмотра смотрите документ, который вызывается из программы, если дважды щелкнуть по файлу `assistant.exe` в папке QT там, где установлена программа MapInfo Pro (например, в папке `C:\Program Files\MapInfo\MapBasic\QT`).

Просмотр справки на другом компьютере

Для того чтобы просмотреть *Справочную систему* на другом компьютере:

1. Скопируйте папку QT и файлы `.qhc` и `.qch` из папки, где установлена программа MapBasic, на другой компьютер, на котором требуется просматривать справку.

Пары файлов с расширением `.qhc` и `.qch` должны находиться в одной папке, чтобы можно было воспользоваться программой просмотра *Справочной системы*.

2. Запустите командную строку..

В меню **Пуск** напечатайте `Командная строка (cmd)` в окошке поиска, в Windows 8 переключитесь на экран **Приложения** и найдите её в разделе **Служебные – Windows**.

3. Смените директорию на папку QT и напечатайте следующую команду, в которой используется адрес файла `.qhc`:

```
assistant.exe -collectionFile C:\Users\Public\Documents\MapBasic.qhc
```

Убедитесь, что в качестве расширения введено значение `.qhc` для введённого имени файла `collectionFile`.

Попробуйте создать загрузочный (bat-файл), которым можно воспользоваться для запуска справочной системы. Файл будет содержать единственную команду:

```
C:\Users\Public\Documents\QT\assistant.exe -collectionFile
C:\Users\Public\Documents\MapBasic.qhc
```

.

Команды меню

Меню Файл

Меню **Файл** содержит команды, позволяющие создавать, открывать, закрывать, сохранять и печатать программы на языке MapBasic, а также закончить сеанс работы.

- Команда Новый (меню Файл)** на стр. 1284
- Команда Открыть (меню Файл)** на стр. 1284
- Заккрыть (меню Файл)** на стр. 1285
- Команда Заккрыть все (меню Файл)** на стр. 1285
- Команда Сохранить (меню Файл)** на стр. 1285
- Команда Сохранить копию (меню Файл)** на стр. 1286
- Команда Восстановить (меню Файл)** на стр. 1286
- Команда Компиляция из файла (меню Файл)** на стр. 1287
- Сборка из файла (меню Файл)** на стр. 1288
- Команда Настройка печати (меню Файл)** на стр. 1288
- Команда Печатать (меню Файл)** на стр. 1288
- Команда Печать в PDF (меню Файл)** на стр. 1289
- Команда Выход (меню Файл)** на стр. 1289

Меню Правка

Меню **Правка** содержит команды, позволяющие редактировать программу на MapBasic.

- Команда Отменить (меню Правка)** на стр. 1290
- Команда Вырезать (меню Правка)** на стр. 1290
- Команда Копировать (меню Правка)** на стр. 1290
- Команда Вставить (меню Правка)** на стр. 1291
- Команда Удалить (меню Правка)** на стр. 1291
- Команда Выбрать все (меню Правка)** на стр. 1291

Меню Поиск

Меню **Поиск** позволяет проводить поиск и замену текста, а также облегчает поиск ошибок, а также облегчает поиск ошибок.

Найти (меню Поиск) на стр. 1292

Найти ещё (меню Поиск) на стр. 1292

Команда Заменить и найти ещё (меню Поиск) на стр. 1293

Команда Следующая ошибка (меню Поиск) на стр. 1293

Команда Предыдущая ошибка (меню Поиск) на стр. 1293

Команда Перейти к строке (меню Поиск) на стр. 1294

Меню Проект

Меню **Проект** содержит команды компиляции и запуска программ на MapBasic, статистики программ и переключения активного окна.

Команда Выбрать файл проекта (меню Сборка) на стр. 1294

Команда Компилировать файл (меню Сборка) на стр. 1295

Собрать проект (меню Сборка) на стр. 1295

Команда Запустить (меню Сборка) на стр. 1295

Команда Информация (меню Сборка) на стр. 1296

Команда Показать/Скрыть ошибки (меню Сборка) на стр. 1296

Меню Окно

Если в MapBasic открыто несколько текстовых окон, то в меню **Окно** можно выбрать, какое из них сделать активным.

Команда Разложить все (меню Окно) на стр. 1296

Команда Сложить в стопку (меню Окно) на стр. 1297

Команда Разложить иконки (меню Окно) на стр. 1297

Команда Стил текст (меню Окно) на стр. 1298

Команда Список открытых окон (меню Окно) на стр. 1298

Меню Справка

Меню **Справка** дает доступ к *Справке MapBasic*. Файл справочной системы содержит описания всех операторов и функций языка MapBasic. *Справка MapBasic* также открывает доступ к

механизму перекрестных ссылок, который позволяет быстро находить описание нужных операторов.

Содержание (меню Справка) на стр. 1298

Команда Поиск (меню Справка) на стр. 1298

Команда Как пользоваться (меню Справка) на стр. 1299

Команда Проверка обновлений (меню Справка) на стр. 1300

О программе MapBasic (меню Справка) на стр. 1300

Команда Новый (меню Файл)

Команда **Новый** открывает новое окно **Правка**, в которое можно вводить текст новой программы.

Когда команда доступна?

Команда активна всегда.

Как получить доступ к этой команде?

- Выберите **Файл > Новый**.

Команда Открыть (меню Файл)

С помощью команды **Открыть** вы открываете файл для **редактирования** в отдельном окне. Команда открывает окно с текстом уже существующей программы на MapBasic (например, `dispatch.mb`, `dispatch.err`) со списком сообщений об ошибках, или текста файла рабочего набора MapInfo Pro. Напоминаем, что файл описания рабочего набора – это просто текстовый файл с операторами языка MapBasic.

В окне диалога **Открыть** имеется возможность открыть сразу несколько файлов. Для этого при выборе имен файлов держите нажатой клавишу **SHIFT** или клавишу **CTRL**.

Примечание: Файлы очень большого размера нельзя открыть в **текстовом** окне MapBasic. Способы обхода этого ограничения приведены в разделе *Ограничения текстового редактора MapBasic* в *Руководстве пользователя MapBasic*.

Когда команда доступна?

Команда **Открыть** доступна всегда.

Как получить доступ к этой команде?

- Выберите **Файл > Открыть**.

Заккрыть (меню Файл)

Команда **Заккрыть** закрывает активное окно **редактирования**. Если вы вносили в него изменения, MapBasic запросит: сохранить ли эти изменения, прежде чем закрыть окно. Команда **Заккрыть** доступна, если открыто хотя бы одно окно **Правка**.

Когда команда доступна?

Команда **Заккрыть** активна, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Файл > Заккрыть**.

Команда Заккрыть все (меню Файл)

Команда **Заккрыть все** закрывает все окна. Как и при выполнении команды **Заккрыть**, MapBasic предлагает в диалоге сохранить изменения или отказаться от них. Команда **Заккрыть все** доступна, если хотя бы одно окно **Правка** открыто.

Когда команда доступна?

Команда **Заккрыть все** доступна, если хотя бы одно окно **Правка** открыто.

Как получить доступ к этой команде?

- Выполните **Файл > Заккрыть все**.

Команда Сохранить (меню Файл)

Команда **Сохранить** сохраняет содержимое активного окна **Правка** на диске. Команда **Сохранить** доступна, если вносились какие-либо **Правка** изменения.

Когда команда доступна?

Команда **Сохранить** активна, если открыто хотя бы одно окно **Правка** с несохраненными изменениями в нем.

Как получить доступ к этой команде?

- Выберите **Файл > Сохранить**.

Команда Сохранить копию (меню Файл)

Команда **Сохранить копию** охраняет содержимое активного окна **Правка** в новом файле на диске. Команда **Сохранить копию** доступна, если хотя бы одно окно **Правка** открыто.

Когда команда доступна?

Команда **Сохранить копию** доступна, если хотя бы одно окно **Правка** если вносились какие-либо изменения.

Как получить доступ к этой команде?

- Выберите **Файл > Сохранить копию**.

Команда Восстановить (меню Файл)

Команда **Восстановить** отменяет все изменения, внесенные со времени последнего сохранения содержимого окна **Правка** в файл на диске. Команда **Восстановить** доступна, если вносились какие-либо изменения в окне **Правка**.

Когда команда доступна?

Команда **Восстановить** активна, если вносились какие-либо изменения в окне **Правка**.

Как получить доступ к этой команде?

- Выберите **Файл > Восстановить**.

Команда Charset (UTF-8/системный набор символов) (меню Файл)

С помощью этой кнопки можно выбрать требующийся набор символов либо UTF-8, либо системный.

Когда активна эта команда?

Эта команда активна всегда.

Как получить доступ к этой команде?

- Выполните команду **Файл > Набор символов > UTF-8/Системный набор символов**.

Команда Компиляция из файла (меню Файл)

Команда **Компиляция из файла** компилирует существующий MB-файл, считывая его прямо с диска, без открытия текстового окна **Правка** для этого файла. (В отличие от команды **Компилировать файл** из меню **Сборка**, которая компилирует модуль из активного **текстового окна**.) Используйте команду **Компиляция из файла** для компиляции программ, написанных во внешнем текстовом редакторе.

При обнаружении ошибок в программе, компилируемой командой **Компиляция из файла**, сообщения об ошибках сохраняются в текстовом файле с названием `filename.err`. Чтобы просмотреть этот файл, выполните команду **Файл > Открыть**.

Когда команда доступна?

Команда **Компиляция из файла** активна всегда.

Как получить доступ к этой команде?

- Выберите **Файл > Компиляция**.

Сборка из файла (меню Файл)

Команда **Сборка из файла** собирает приложение на основании заданного файла описания проекта без показа этого файла в текстовом окне **Правка**. (В отличие от команды **Собрать проект** из меню **Проект**, собирающей текущий проект.)

Когда команда доступна?

Команда **Сборка из файла** активна всегда

Как получить доступ к этой команде?

- Выберите **Файл > Сборка из файла**.

Команда Настройка печати (меню Файл)

Команда **Настройка печати** задает режимы печати для активного окна **Правка** (например, номер принтерного порта).

Когда команда доступна?

Команда **Настройка печати** доступна, если установлен хотя бы один принтер.

Как получить доступ к этой команде?

- Выберите **Файл > Настройка печати**.

Команда Печатать (меню Файл)

Используйте команду **Печать**, чтобы напечатать, если открыто хотя бы одно окно **Правка**.

Когда команда доступна?

Команда **Печатать** активна, если хотя бы одно **текстовое** окно открыто.

Как получить доступ к этой команде?

- Выберите **Файл > Печать**.

Команда Печать в PDF (меню Файл)

Используйте команду **Печать в PDF**, чтобы вывести в PDF-файл активное текстовое окно.

Когда команда доступна?

Команда **Печать в PDF** доступна, если открыто хотя бы одно активное окно.

Как получить доступ к этой команде?

- Выберите **Файл > Печать в PDF**.

Команда Выход (меню Файл)

Команда **Выход** завершает сеанс работы в среде MapBasic. MapBasic выдаст запрос, сохранять ли внесенные изменения.

Когда команда доступна?

Команда **Выход** активна всегда

Как получить доступ к этой команде?

- Выполните **Файл > Выход**.

Команда Последние используемые файлы (меню Файл)

Используйте названия **последних используемых файлов** чтобы быстро открыть те файлы, с которыми вы недавно работали. Когда файл будет выбран, он станет активным окном.

Когда команда доступна?

Последние используемые файлы доступны, если как минимум один файл уже открывался программой.

Как получить доступ к этой команде?

- Выберите **Файл > 1 имя_файла**.

Команда Отменить (меню Правка)

Команда **Отменить** отменяет результат последнего действия в активном текстовом окне **Правка**. При выполнении команды **Отменить**, MapBasic удаляет результат последнего действия, а в меню этот пункт превращается в **Повторить**. Выполнение команды **Повторить**, возвращает отмененные изменения.

Когда команда доступна?

Команда **Отменить** активна, если хотя бы одно окно открыто и внесено хотя бы одно изменение в окне **Правка**.

Как получить доступ к этой команде?

- Выберите **Правка > Отменить**.

Команда Вырезать (меню Правка)

Команда **Вырезать** переносит выбранный текст в буфер обмена, удаляя его из **текстового окна**. Помещенный в буфер обмена текст можно потом вставить с помощью команды **Вырезать** (см. ниже).

Когда команда доступна?

Команда **Вырезать** активна, если в одном из окон **Правка** выбран текст.

Как получить доступ к этой команде?

- Выберите **Правка > Вырезать**.

Команда Копировать (меню Правка)

Команда **Копировать** копирует выбранный текст в буфер обмена, не удаляя его с экрана.

Когда команда доступна?

Команда **Копировать** активна, если в одном из окон **Правка** или **MapBasic** выбран текст.

Как получить доступ к этой команде?

- выполните **Правка > Копировать**.

Команда Вставить (меню Правка)

Команда **Вставить** вставляет содержимое буфера обмена в активное окно **Правка** с позиции, где находится курсор. Если вы выберете текст в окне **Правка** и выполните команду **Вставить**, то текст из буфера обмена заменит выбранный текст.

Когда команда доступна?

Команда **Вставить** активна, если есть текст в буфере обмена и хотя бы одно окно **Правка** открыто.

Как получить доступ к этой команде?

- Выберите **Правка > Вставить**.

Команда Удалить (меню Правка)

Команда **Удалить** удаляет выбранный текст..

Когда команда доступна?

Команда **Удалить** активна, если в одном из окон **Правка** выбран текст.

Как получить доступ к этой команде?

- Выберите **Правка > Удалить**.

Команда Выбрать все (меню Правка)

Команда **Выбрать все** выбирает все содержимое активного текстового окна **Правка**.

Когда команда доступна?

Команда **Выбрать все** активна, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Правка > Выбрать все**.

Найти (меню Поиск)

Команда **Найти** осуществляет поиск заданной текстовой строки в активном окне **Правка**. Чтобы найти заданную строку поступайте следующим образом: наберите строку, которую Вы хотите найти, в диалоге Найти. Если требуется в процессе поиска различать большие и малые буквы, установите флажок **Различать строчные и прописные**.

Нажмите на кнопку **Найти**. MapBasic начнет поиск с текущей позиции курсора. Если заданная строка будет найдена, MapBasic пролистает окно до того места, где была найдена эта строка. Если строка не найдена, MapBasic подаст звуковой сигнал.

Когда команда доступна?

Команда **Найти** доступна, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Поиск > Найти**.

Найти ещё (меню Поиск)

Команда **Найти ещё** ищет следующее совпадение со строкой, заданной в предыдущем диалоге поиска.

Когда команда доступна?

Команда **Найти ещё** доступна если открыто хотя бы одно окно **Правка** и операция поиска была уже однажды выполнена.

Как получить доступ к этой команде?

- Выберите **Поиск > Найти ещё**.

Команда Заменить и найти ещё (меню Поиск)

Команда **Заменить и найти ещё** заменяет заданный в диалоге **Найти** текст, а затем ищет и выделяет следующее совпадение с заданной строкой.

Когда команда доступна?

Команда **Заменить и найти ещё** доступна, если открыто хотя бы одно окно **Правка** и была проведена операция поиска.

Как получить доступ к этой команде?

- Выберите **Поиск > Заменить и найти ещё**.

Команда Следующая ошибка (меню Поиск)

Команда **Следующая ошибка** позволяет находить и исправлять синтаксические ошибки. Если в программе обнаружены ошибки, MapBasic показывает список ошибок в нижней части окна **Правка**. Команда **Следующая ошибка** листает окно **Правка** до той строки программы, где была обнаружена заданная ошибка.

Когда команда доступна?

Команда **Следующая ошибка** активна, если при компиляции были обнаружены ошибки в окне **Правка**.

Как получить доступ к этой команде?

- Выберите **Поиск > Следующая ошибка**.

Команда Предыдущая ошибка (меню Поиск)

Команда **Предыдущая ошибка** листает окно **Правка** назад к предыдущей ошибке.

Когда команда доступна?

Команда **Предыдущая ошибка** активна, если при компиляции были обнаружены ошибки в окне **Правка**.

Как получить доступ к этой команде?

- Выберите Поиск > Предыдущая ошибка.

Команда Перейти к строке (меню Поиск)

Команда **Перейти к строке** позволяет пролистать текстовое окно **Правка** до строки с заданным номером.

Возможно, что ваша программа все еще содержит ошибки, хотя компиляция прошла успешно. При возникновении таких ошибок в момент выполнения приложения появляется диалог, в котором указана строка, где появилась ошибка. Обычно, после этого следует вернуться в среду MapBasic и изучить строку с указанным номером.

Когда команда доступна?

Команда **Перейти к строке** активна, если хотя бы одно окно **Правка** открыто.

Как получить доступ к этой команде?

- Выберите Поиск > Перейти к строке.

Команда Выбрать файл проекта (меню Сборка)

Команда **Выбрать файл проекта** открывает диалог, в котором вы можете выбрать один из имеющихся файлов проектов. Файл описания проекта – это текстовый файл, в котором перечислены модули, из которых состоит приложение. Выполнив данную команду, вы делаете один из файлов проекта активным и можете собирать соответствующий проект с помощью команды **Собрать проект**.

Когда команда доступна?

Команда **Выбрать файл проекта** активна всегда.

Как получить доступ к этой команде?

- Выберите Сборка > Выбрать файл проекта.

Команда Компилировать файл (меню Сборка)

Команда **Компилировать файл** компилирует программу (модуль) из активного окна **Правка**. Если компилятор обнаружил синтаксические ошибки, MapBasic покажет список ошибок в нижней части активного окна **Правка**. Если ошибок не зафиксировано, то MapBasic построит MBX-файл (если компилируется программа, состоящая только из одного модуля) или объектный (MBO) файл.

Когда команда доступна?

Команда **Компилировать файл** активна, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Сборка > Компилировать файл**.

Собрать проект (меню Сборка)

Команда **Собрать проект** собирает модули, перечисленные в текущем файле проекта в исполняемый файл приложения (если только не были обнаружены ошибки при сборке).

Когда команда доступна?

Команда **Собрать проект** активна, если открыт файл проекта.

Как получить доступ к этой команде?

- Выберите **Проект > Собрать проект**.

Команда Запустить (меню Сборка)

Команда **Запустить** посылает сообщение MapInfo Pro о том, что следует запустить на выполнение приложение из активного текстового окна **Правка**.

Когда команда доступна?

Запустить становится доступна после компиляции файла (командой **Сборка > Компилировать файл**).

Как получить доступ к этой команде?

- Выберите **Проект > Запустить**.

Команда Информация (меню Сборка)

Команда **Информация** показывает статистическую информацию о программе из активного окна **Правка**.

Когда команда доступна?

Информация активна, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Проект > Информация**.

Команда Показать/Скрыть ошибки (меню Сборка)

Одна из команд **Показать/Скрыть ошибки** открывает или закрывает список сообщений об ошибках для активного окна **Правка**. Если список ошибок показывается в окне, то меню содержит строку **Скрыть ошибки**. Если список ошибок не показывается, то меню содержит строку **Показать ошибки**.

Когда команда доступна?

Команда **Показать/Скрыть ошибки** активна, если имеется окно **Правка**, которому соответствует хотя бы одно сообщение об ошибке.

Как получить доступ к этой команде?

- Выберите **Проект > Показать/Скрыть ошибки**.

Команда Разложить все (меню Окно)

Команда **Разложить все** располагает окна **Правка** так, чтобы все они были видны.

Когда команда доступна?

Команды этого меню активны, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Окно > Разложить все**.

Команда Сложить в стопку (меню Окно)

Команда **Сложить в стопку** располагает окна друг над другом (каскадом).

Когда команда доступна?

Команды этого меню активны, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Окно > Сложить в стопку**.

Команда Разложить иконки (меню Окно)

Команда **Разложить иконки** переставляет иконки, соответствующие минимизированным окнам **Правка**. Чтобы временно свернуть окно **Правка** в иконку, нажмите кнопку минимизации справа от строки заголовка.

Когда команда доступна?

Команды этого меню активны, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Окно > Разложить иконки**.

Команда Стил ь текста (меню Окно)

Команда **Стил ь текста** позволяет выбрать шрифт, которым следует показывать текст в окне. Этим шрифтом показывается весь текст в окне.

Когда команда доступна?

Команды этого меню активны, если открыто хотя бы одно окно **Правка**.

Как получить доступ к этой команде?

- Выберите **Окно > Стил ь** текста.

Команда Список открытых окон (меню Окно)

Нижняя часть меню Окно содержит перечень открытых окон **Правка**. Чтобы сделать окно **Правка** активным (т.е. поместить его поверх других), выберите название окна в меню Окно.

Содержание (меню Справка)

Команда **Содержание** открывает окно **Справка**, откуда можно переходить к интересующим вас темам. Для этого вы указываете на их заголовки или, выбирая тему в диалоге (для этого нажмите кнопку **Поиск**) – откроется диалог **Найти**.

Когда команда доступна?

Эта команда активна всегда.

Как получить доступ к этой команде?

- Выберите **Справка > Содержание**.

Команда Поиск (меню Справка)

Команда **Поиск** открывает одноименный **диалог**.

Команда **Поиск** обеспечивает доступ к панели поиска *Справки MapBasic*. Вы можете вводить ключевые слова, по которым будут найдены операторы, функции или переменные MapBasic API.

Когда команда доступна?

Команда **Поиск** активна всегда.

Как получить доступ к этой команде?

- Выберите **Справка > Поиск**.

Команда Как пользоваться (меню Справка)

Команда **Как пользоваться** открывает тему, описывающую порядок работы с подсказками *Справки MapBasic*. Электронная *Справка MapBasic* основана на *Справочнике MapBasic*.

Когда команда доступна?

Команда **Как пользоваться** активна всегда.

Как получить доступ к этой команде?

- Выберите **Справка > Как пользоваться**.

Команда Справочник пользования (меню Справка)

Если нажать кнопку **Справочник пользования**, MapBasic открывает *Справочник пользования MapInfo Pro*.

Когда активна эта команда?

Команда **Справочник пользования** всегда активна.

Как получить доступ к этой команде?

- Выполните команду **Справка > Справочник пользования**.

Команда **Общественная загрузка** (меню **Справка**)

Когда нажимается кнопка **Общественная загрузка**, MapBasic открывает веб-сайт, с которого можно загрузить нужные MapInfo Pro или MapBasic ресурсы.

Когда активна эта команда?

Команда **Общественная загрузка** всегда активна.

Как получить доступ к этой команде?

- Выполните команду **Справка > Общественная загрузка**.

Команда **Проверка обновлений** (меню **Справка**)

Команда **Проверка обновлений** открывает страницу Web-сайта Precisely, которая посвящена доступным обновлениям программы. Вы попадёте на Web-страницу, на которой представлены обновления MapBasic API.

Когда команда доступна?

Команда **Проверка обновлений** активна всегда, если у вас подключены к Интернету.

Как получить доступ к этой команде?

- выполните **Справка > Проверка обновлений**.

О программе MapBasic (меню **Справка**)

Команда **О программе MapBasic** показывает диалог **О программе MapBasic**, в котором содержится информация об авторских правах и номер версии программы.

Примечание: Многие подсказки содержат краткие примеры программ. Можно скопировать любой из этих примеров в буфер обмена, а затем вставить в свою программу. Для этого выполните команду **Правка > Копировать** в окне **Правка Справочной системы** или переместите выбранный фрагмент мышкой из окна **Справки** в вашу программу.

Когда команда доступна?

Команда **О программе MapBasic** активна всегда

Как получить доступ к этой команде?

- Выполните команду **Справка > О программе MapBasic**.

Описание диалогов MapBasic

Диалог Компилировать программу

Просмотр

Найдите папку и файл для компиляции. Файлы показываются в окошке. Выбрать нужный файл можно, дважды указав на него мышкой.

Кнопка Наверх

Позволяет подняться в иерархии папок на один уровень выше.

Кнопка Рабочий стол

Показывает содержимое вашего Рабочего стола.

Кнопка Создать папку

Создает папку с временным названием.

Кнопка Список

Выводит список названий файлов и папок без указания их свойств (даты изменения, размера и т.п.)

Кнопка Детали

Представляет полную информацию о файлах и папках: их название, размер, тип, дату последнего изменения.

Окошко с файлами и папками

Показывает папки и файлы в заданном каталоге. Двойное нажатие на папку или файл открывает его.

Окно быстрого доступа к каталогам

Кнопки в левой панели диалоге позволяют открыть стандартные папки системы: Недавние документы, Рабочий стол, Мои документы, Мой компьютер и Сетевое окружение. Нажатие на одну из этих пиктограмм меняет текущий каталог в диалоге на тот, который ассоциируется с пиктограммой.

Имя файла

Введите полное имя файла для компиляции.

Тип файлов

Выберите типы файлов; "Программы (*.mb)", "Заголовки программ (*.def)", "Файлы ошибок (*.err)" или "Все файлы (*.*)".

Открыть

Нажмите, чтобы начать компиляцию программы.

Отмена

Отменяет выбранные в диалоге настройки.

Диалог Поиск и замена

Позволяет находить слова и фрагменты текста и автоматически заменять их другими.

Найти

Текст, который надо найти.

Заменить на

Заменить на текст из этого окошка.

Различать строчные и прописные

Установите флажок, чтобы различать регистр букв.

Найти (кнопка)

Найти следующий фрагмент искомого текста.

Заменить все

Произвести замену искомого фрагмента на заменяющий во всем файле.

Отмена

Отменяет выбранные в диалоге настройки.

Диалог Собрать проект

Просмотр

Найдите папку и файл проекта для сборки. Файлы показываются в окошке. Выбрать нужный файл можно, дважды указав на него мышкой.

Кнопка Наверх

Позволяет подняться в иерархии папок на один уровень выше.

Кнопка Рабочий стол

Показывает содержимое вашего Рабочего стола.

Кнопка Создать папку

Создает папку с временным названием.

Кнопка Список

Выводит список названий файлов и папок без указания их свойств (даты изменения, размера и т.п.)

Кнопка Детали

Представляет полную информацию о файлах и папках: их название, размер, тип, дату последнего изменения.

Окошко с файлами и папками

Показывает папки и файлы в заданном каталоге. Двойное нажатие на папку или файл открывает его.

Окно быстрого доступа к каталогам

Кнопки в левой панели диалоге позволяют открыть стандартные папки системы: Недавние документы, Рабочий стол, Мои документы, Мой компьютер и Сетевое окружение. Нажатие на одну из этих пиктограмм меняет текущий каталог в диалоге на тот, который ассоциируется с пиктограммой.

Имя файла

Введите полное имя файла сборки.

Тип файлов

Выберите тип файла для сборки; по умолчанию это "Проекты (*.mbp)".

Открыть

Нажмите для начала сборки выбранного файла.

Отмена

Отменяет выбранные в диалоге настройки.

Диалог Открыть программу

Позволяет **открыть файлы** следующих типов: "Программы (*.mb)", "Заголовки программ (*.def)", "Файлы ошибок (*.err)" или "Все файлы (*.*)". Выбрав "Все файлы (*.*)", вы увидите в списке файлы всех типов и вам придется различать их по расширениям.

Просмотр

Найдите папку и файл. Файлы показываются в окошке. Выбрать нужный файл можно, дважды указав на него мышкой.

Кнопка Наверх

Позволяет подняться в иерархии папок на один уровень выше.

Кнопка Рабочий стол

Показывает содержимое вашего Рабочего стола.

Кнопка Создать папку

Создает папку с временным названием.

Кнопка Список

Выводит список названий файлов и папок без указания их свойств (даты изменения, размера и т.п.)

Кнопка Детали

Представляет полную информацию о файлах и папках: их название, размер, тип, дату последнего изменения.

Окошко с файлами и папками

Показывает папки и файлы в заданном каталоге. Двойное нажатие на папку или файл открывает его.

Окно быстрого доступа к каталогам

Кнопки в левой панели диалоге позволяют открыть стандартные папки системы: Недавние документы, Рабочий стол, Мои документы, Мой компьютер и Сетевое окружение. Нажатие на одну из этих пиктограмм меняет текущий каталог в диалоге на тот, который ассоциируется с пиктограммой.

Имя файла

Введите полное имя открываемого файла.

Тип файлов

Выберите формат, в котором будет сохранен файл.

Открыть

Нажмите для открытия файла.

Отмена

Отменяет выбранные в диалоге настройки.

диалог Настройка печати

В диалоге контролируются размер, подача, ориентация и поля распечатки.

Бумага**Размер**

Выберите размер листа из списка.

Источник

Выберите источник листов из списка.

Ориентация**Книжная**

Печать параллельно короткой стороне листа.

Альбомная

Печать параллельно длинной стороне листа.

Поля (мм)

Стандартная единица измерений - дюймы. Единицы измерений можно поменять в диалоге **Режимы**. Нажмите кнопку **Системные**, чтобы открыть диалог Системные режимы и выберите единицы измерения из списка **Для отчетов**.

Влево

Задайте размер левого поля.

Вправо

Задайте размер правого поля. Эти режимы используются для односторонней печати. Если выбран режим двусторонней печати, этот режим переименовывается во Внешнее.

Верх

Задайте размер верхнего поля.

Низ

Задайте размер нижнего поля.

ОК

Применяет произведенные установки.

Отмена

Отменяет выбранные в диалоге настройки.

диалог Настройка печати (Выбор принтера)

В этом диалоге можно выбрать параметры печати для выбранного размера листа.

Имя

Выберете принтер из списка.

Свойства

Нажатие на эту кнопку открывает диалог с настройками для данного принтера. Здесь можно настраивать ориентацию листа (книжную или альбомную), порядок печати (как обычно или в обратном порядке) и количество страниц на листе. Для многих принтеров существуют также дополнительные настройки, открываемые одноименной кнопкой. В дополнительные настройки обычно входят средства повышения качества печати, предоставляемые данным принтером.

Состояние

В этом поле демонстрируется текущее состояние принтера и готовность его к печати. Это поле предназначено только для чтения.

Тип

Показывает тип принтера, выбранного в поле **Имя**. Это поле предназначено только для чтения.

Где

Показывает, в каком месте сети подсоединен данный принтер. Это поле предназначено только для чтения.

Комментарии

Комментарии, которые вводит системный администратор. Это поле предназначено только для чтения.

Сеть

Кнопка доступа к сети и поиска других доступных принтеров. Для печати на таком принтере вам может понадобиться специальный драйвер.

ОК

Нажав эту кнопку, вы запоминаете настройки и возвращаетесь в диалог **Настройка печати**.

Отмена

Возврат в диалог **Настройка печати** без изменений.

Диалог Подбор шрифтов

Позволяет **выбрать шрифт** и режимы изображения текста.

Шрифт

Выберите шрифт и размер из списка Шрифт.

Размер

Выберите размер букв из списка Размер.

Жирный

Буквы в тексте будут **полужирными**.

Курсив

Буквы в тексте будут *наклонными*.

Подчеркнутый

Буквы в тексте будут подчеркнутыми.

ОК

Применяет произведенные установки.

Отмена

Отменяет выбранные в диалоге настройки.

диалог Печать

Команда **Печать** позволяет выбрать и настроить принтер и запустить процесс печати. Принтеры устанавливаются в среде Windows (через **Панель управления**). Информацию об установке принтеров вы можете найти в документации по системе Windows.

Принтер

Имя

Список установленных на этом компьютере принтеров.

Статус, Тип, Порт, Заметки

Информация о текущем состоянии выбранного принтера.

Кнопка Свойства

Нажав на эту кнопку, можно вызвать диалог настройки выбранного принтера. Настройки индивидуальны для каждой модели принтера.

Печать в файл

Выводит задание на печать в файл вместо самого принтера. Будет предложено указать имя файла и его файловый адрес.

Печатать**Все или Страницы**

Следует указать, печатать ли все страницы документа или только отдельные листы.

Копии**Число копий**

Можно указать количество копий документа.

ОК

Применяет произведенные установки.

Отмена

Отменяет выбранные в диалоге настройки.

Диалог Сохранить программу

Сохранить как

Выберите диск, папку, и файл для сохранения, импорта или создания таблицы.

Кнопка Наверх

Позволяет подняться в иерархии папок на один уровень выше.

Кнопка Рабочий стол

Показывает содержимое вашего Рабочего стола.

Кнопка Создать папку

Создает папку с временным названием.

Кнопка Список

Выводит список названий файлов и папок без указания их свойств (даты изменения, размера и т.п.)

Кнопка Детали

Представляет полную информацию о файлах и папках: их название, размер, тип, дату последнего изменения.

Окошко с файлами и папками

Показывает папки и файлы в заданном каталоге. Двойное нажатие на папку или файл открывает его.

Окно быстрого доступа к каталогам

Кнопки в левой панели диалоге позволяют открыть стандартные папки системы: Недавние документы, Рабочий стол, Мои документы, Мой компьютер и Сетевое окружение. Нажатие на одну из этих пиктограмм меняет текущий каталог в диалоге на тот, который ассоциируется с пиктограммой.

Имя файла

Отображает имя файла с расширением MB. Можно изменить имя файла. MapBasic поддерживает длинные имена файлов. Можно задать имя файлу длиной до 260 символов. Кроме того, любое имя в спецификации маршрута к вашему файлу может содержать до 255 символов. В именах файлов допускаются точки и пробелы. Если используется более одной точки, следите за тем, чтобы после последней стоял правильный тип файла. Например, так:
`\\STATE\CITY.MB`

Выбор типа при экспорте окна

Позволяет открыть файлы следующих типов: **"Программы (*.mb)"**, **"Проекты (*.mbp)"**, **"Заголовки программ (*.def)"** или **"Все файлы (*.*)"**. Выбрав "Все файлы (*.*)", вы увидите в списке файлы всех типов и вам придется различать их по расширениям.

Сохранить

Нажмите для сохранения файла.

Отмена

Отменяет выбранные в диалоге настройки.

Диалог Выбор файла проекта

Просмотр

Найдите папку и файл. Файлы показываются в окошке. Выбрать нужный файл можно, дважды указав на него мышкой.

Кнопка Наверх

Позволяет подняться в иерархии папок на один уровень выше.

Кнопка Рабочий стол

Показывает содержимое вашего Рабочего стола.

Кнопка Создать папку

Создает папку с временным названием.

Кнопка Список

Выводит список названий файлов и папок без указания их свойств (даты изменения, размера и т.п.)

Кнопка Детали

Представляет полную информацию о файлах и папках: их название, размер, тип, дату последнего изменения.

Окошко с файлами и папками

Показывает папки и файлы в заданном каталоге. Двойное нажатие на папку или файл открывает его.

Окно быстрого доступа к каталогам

Кнопки в левой панели диалоге позволяют открыть стандартные папки системы: Недавние документы, Рабочий стол, Мои документы, Мой компьютер и Сетевое окружение. Нажатие на одну из этих пиктограмм меняет текущий каталог в диалоге на тот, который ассоциируется с пиктограммой.

Имя файла

Введите полное имя открываемого файла.

Тип файлов

Выберите тип файла для сборки: "Проекты (*.mbp)" или "Все файлы (*.*)".

Открыть

Нажмите эту кнопку для открытия файла.

Отмена

Отменяет выбранные в диалоге настройки.



2 Blue Hill Plaza, #1563
Pearl River, NY 10965
USA

www.precisely.com

© 1985, 2020 Precisely. Все права защищены.