

precisely

EngageOne Designer

User's Guide

Version 6.6.11



Table of Contents

1 - Licensing

Keycode entry.....	11
--------------------	----

2 - Getting started

Start-up and login.....	13
Logon details.....	13
Workflow.....	14
Design basics.....	15
Using the Designer.....	19
Input data and the data format.....	25
Publishing and production.....	27
Audit Trail.....	27
Designer preferences.....	28
Errors and information.....	29
Filtering by name using wildcard matching.....	30

3 - Searching for assets

Accessing search features.....	32
Working with the search feature.....	33
Working with the advanced search feature.....	34

4 - Working with the editor

The editor workspace.....	57
Editing features.....	58

5 - Creating paragraphs

Entering and formatting text.....	69
Positioning text.....	77

Bullets and numbering.....	85
Pagination.....	86
Conditional paragraphs.....	88
Merging paragraphs.....	89
Importing paragraph content.....	89
Symbols.....	90
System variables.....	91
URL links.....	93

6 - Working with graphics

Using graphics.....	96
Anchors and positioning.....	96
Drawing shapes.....	98
Using images.....	99
Text boxes.....	101
Selecting and adjusting graphics.....	103
Layout.....	107
Order of graphics.....	108

7 - About page layouts

Working with the page layout	110
Page setup.....	111
Columns.....	115

8 - About input data

Working with input data.....	117
Using data aliases within text.....	117
Repeating data and transaction tables.....	120

9 - About design files and resources

Working with design files and resources.....	131
Version control.....	132
Using fonts and images.....	141
Custom code pages.....	146

10 - About the logic map

Using the logic map.....	150
Restricting active objects.....	151
Bookmarks.....	152
Notes.....	152
Searching the logic map.....	153
Logic map errors.....	154

11 - About publications and documents

Working with publications and documents.....	156
Using multiple documents.....	160
Sharing and synchronization.....	162
External documents.....	165
Working with managed publications.....	169

12 - Working with values

Adding values to objects.....	171
About data types.....	172
Formatting values.....	173
Variables.....	178

13 - Page and job handling

Page presentation and processing.....	183
Generating new pages.....	184
Page Setup view.....	185
Working with reserved areas.....	191

Working with Margin Notes.....	194
Sections and page control.....	198
Customizing columns.....	205
Using run-time actions.....	214

14 - Creating conditional logic

Conditional expressions.....	218
Condition.....	220
Page check.....	221
Loop.....	222
Case.....	224
Data driven character styles.....	226

15 - Functions

About Designer Functions.....	231
Address block.....	231
Arithmetic.....	234
Barcodes.....	238
Concatenation.....	254
Lookup tables.....	255
Substring.....	256
System variables.....	257

16 - About tables

Using tables.....	262
Table positioning.....	262
Basic editing.....	263
Borders and shading.....	264
Headers and footers.....	265
Pagination.....	266

17 - Keyed objects

Using keyed objects.....	268
About key maps.....	268
Creating the keyed object.....	270

18 - Charts

Which chart type to use.....	273
Using plug-in charts.....	274
Using native charts.....	279

19 - Advanced data handling

Sample data and data mapping.....	304
Associating and editing data maps.....	304
Using a repeating data control.....	305
Creating a repeating data control.....	306
Repeating data dialog.....	308

20 - Localization

About locales.....	311
Localizing document designs.....	312

21 - Production journals

Working with Production journals.....	316
---------------------------------------	-----

22 - Interfacing with message applications

Message areas and data fields.....	321
Importing the message environment file.....	321
Message boxes and areas.....	322
Message streaming.....	324
Resolving message data fields.....	325
Message signals.....	327
Production & preview	329
Customizing the audit trail.....	330

23 - Interfacing with Vault

Working with Designer documents.....	332
Creating the index.....	332

DIJ settings for use with Vault.....	333
Table of contents.....	334

24 - Interfacing with EngageOne Deliver

Datastreams available for EngageOne Deliver..	336
DIJ settings for use with EngageOne Deliver....	336

25 - Publishing content for EngageOne Communicate

Connecting to your EngageOne Communicate account.....	338
Dynamic attachments	342
Working with tagged content.....	343
Version Label and Audit Trail.....	346

26 - Working with EngageOne Video design objects

Workflow.....	348
---------------	-----

27 - Creating EngageOne templates

EngageOne - operating modes.....	350
Working with interactive documents.....	351
Working with multiple languages.....	354
EngageOne data.....	355
Prompt groups.....	386
Building the EngageOne template.....	391
Previewing an EngageOne template.....	397
Publishing for EngageOne.....	400
Import and exporting interactive data definitions.	415

28 - Active Content

About Active Content.....	429
---------------------------	-----

Creating a basic Active Content object.....	429
Using Active Content as a function.....	432
Using Active Content groups.....	435
Using Active Content to update publications in production.....	441
Using Active Content with EngageOne.....	443

29 - Review and approval

Workflow.....	447
Working with managed projects and publications in Designer.....	449
Submitting assets for review.....	451
Review comments.....	451
Merging managed projects.....	453

30 - Controlling the output environment

Spot color printers.....	455
Controlling printer features.....	457
PostScript DSC.....	458
Adding output comments.....	458
PDF Annotation.....	459

31 - Working with Designer input data

Supported input data formats.....	463
About Designer input data	463

32 - About data formats

Creating a data format.....	472
-----------------------------	-----

33 - About data mapping

Data mapping.....	496
-------------------	-----

34 - About data dictionaries

Working with data dictionaries.....	505
-------------------------------------	-----

35 - About the production cycle

Into production	509
-----------------------	-----

36 - About Generate

Running Generate.....	513
-----------------------	-----

37 - Output, media and resources

Logical pages and media layout.....	521
Font and image resources.....	523
Format of the output datastream.....	524
Support for ideographic text.....	524
Creating output device objects.....	524
AFP.....	526
PCL.....	536
Postscript.....	541
PDF.....	549
RTF.....	560
Line Data.....	563
HTML.....	566
Specifying a file template.....	585
Using resource maps.....	586

38 - About publishing and deployment

Publishing and deployment.....	592
--------------------------------	-----

39 - Access control and security

User accounts.....	620
User roles.....	622
User groups.....	625
Security.....	626
Securing data transfer.....	629
Sessions.....	630

40 - Managing Resources

Private/public resources.....	632
Working with resource objects.....	632

41 - Repository management

Designer Repository Configuration.....	639
Snapshots.....	647

42 - Appendix A Glossary

2of5.....	657
3of9.....	657
A.....	657
Action.....	657
Active Content.....	657
ACU.....	658
Address block.....	658
Administration.....	658
Adobe font metrics.....	658
Advanced preview.....	658
AFP (Advanced Function Presentation).....	659
AFPDS (Advanced Function Presentation Data Stream).....	659
Anchor.....	659
Anchored.....	659
Application Import utility.....	659
Arithmetic.....	660
ASCII.....	660

Asset.....	660
Audit trail.....	660
Axes, axis.....	660
B.....	661
Background.....	661
Bar.....	661
Barcode.....	661
Base 14 fonts.....	661
Batch publishing.....	662
Bidirectional text.....	662
Bitmap or BMP.....	662
Body/body area.....	662
BookBill.....	662
Bookmark.....	663
Bounding box.....	663
Branch.....	663
Browser.....	663
Bundle.....	663
C.....	664
Carriage return.....	664
Case.....	664
Casting.....	664
Cell.....	664
Centralized printer.....	665
Character set.....	665
Chart.....	665
Chart element.....	665
Checkpoint.....	665
Child object.....	666
Clickable image.....	666
CMYK.....	666
Code128.....	666
Coded font.....	666
Code page.....	667
Column.....	667
Column layout.....	667
Communicate account.....	667
Composite resource.....	667
Concatenation.....	668
Condition.....	668
Conditional paragraph.....	668
Constant.....	668
Content Author.....	668
Control object.....	669
Cross browser.....	669

Currency.....	669	EngageOne Compose Interactive Editor.....	678
<i>D</i>	669	EngageOne Deliver.....	678
Data alias.....	669	EngageOne Video.....	678
Database.....	670	EPS.....	679
Data dictionary.....	670	Eye-catcher.....	679
Data field.....	670	External document.....	679
Data Flow.....	670	Express Batch.....	679
Data format & Data Format Editor.....	670	<i>F</i>	679
Data map.....	671	Field.....	680
Data input user exit.....	671	Filestore.....	680
Data record.....	671	Fixed anchor.....	680
Date.....	671	Folder.....	680
DBCS.....	671	Font.....	680
Default page header/footer.....	672	Font scaling barcode.....	681
Deployment.....	672	Font subsetting.....	681
Design review and approval.....	672	Footer.....	681
Delimited data.....	672	Form.....	681
DFE.....	672	Formatting marks.....	681
Dictionaries.....	673	Formula.....	682
DIJ - Document Interchange Journal.....	673	<i>G</i>	682
DIME.....	673	Generate.....	682
DOC1ACU.....	673	Governor field.....	682
DOC1DIME.....	673	<i>H</i>	682
DOC1GEN.....	674	Hard return.....	683
DOC1Publish.....	674	Header.....	683
DOC1RPU.....	674	HIP.....	683
Document.....	674	Host.....	683
Document area.....	674	HTML.....	683
Document Editor.....	675	<i>I</i>	684
Document logic.....	675	Image.....	684
Document section.....	675	Indentation.....	684
Dot leader.....	675	Indexes.....	684
Double byte fonts.....	675	In-line anchor.....	684
Dpi.....	676	Input data.....	685
Draft.....	676	Insertion point.....	685
DSC comments.....	676	Integer.....	685
<i>E</i>	676	Interactive Data Editor (IDE).....	685
EBCDIC.....	676	Interactive variable.....	685
Editable text.....	677	<i>J</i>	686
eHTML.....	677	Journal.....	686
Embedded resources.....	677	<i>K</i>	686
EMFE (Extract Merge and Format Engine).....	677	Keycodes.....	686
EngageOne.....	677	Key field.....	686
EngageOne Communicate.....	678	Key map.....	687
EngageOne Compose ActiveX Editor.....	678	Keyed object.....	687

Keyed record data.....	687	<i>P</i>	696
Keyed user exit.....	687	Page check.....	696
<i>L</i>	687	Page setup.....	696
Legend.....	688	Paragraph.....	697
Lines.....	688	Parameter.....	697
Line data.....	688	PCE.....	697
Line drawing barcode.....	688	PCL.....	697
Locale.....	688	PDF, PDF/A.....	697
Lock/unlock.....	689	Pel (picture element).....	698
Logic map.....	689	Physical page.....	698
Logical page.....	689	Pica.....	698
Logo.....	689	Point.....	698
Lookup table.....	689	Post Composition Engine.....	698
Lookup table user exit.....	690	PostNet.....	699
Loop.....	690	Postscript.....	699
Label.....	690	Preview.....	699
<i>M</i>	690	Privilege.....	699
Managed project.....	690	Production job.....	699
Mandatory records.....	691	Project.....	700
Margin.....	691	Project services.....	700
Margin note.....	691	Prompt group.....	700
Merge.....	691	Prompt layer editor.....	700
Message.....	691	Prompt loop.....	700
Message area.....	692	Public.....	701
Messages file.....	692	Publication.....	701
Message box.....	692	Publication data set.....	701
Message stream.....	692	Publication Editor.....	701
Message audit trail.....	692	Publish.....	701
Message environment file.....	693	PURL.....	702
MVS.....	693	<i>R</i>	702
<i>N</i>	693	Rasterize.....	702
Navigator.....	693	RDW (record descriptor word).....	702
Nested table.....	693	Record.....	702
Non-repeating data.....	694	Remaining space.....	703
Note.....	694	Repeating data.....	703
Number.....	694	Repository.....	703
<i>O</i>	694	Repository Configuration Tool.....	703
OpenType font.....	694	Reserved area.....	703
OPS (override production settings).....	695	Reserved area entry.....	704
Orphan.....	695	Resource map.....	704
Output datastream.....	695	Return code.....	704
Output device.....	695	Revert.....	704
Outsourcing.....	695	Revision.....	704
Overlay.....	696	RGB.....	705
Owner password.....	696	RPU (Resource Pack Utility).....	705

RTF (Rich Text Format).....	705
Rollback.....	705
Row.....	705
Rulers.....	706
S.....	706
Sample data.....	706
Scope.....	706
Section.....	706
Segmented publishing.....	707
Set.....	707
Shape.....	707
Sheet.....	707
Short Message Service (SMS).....	707
Snapshot.....	708
Soft return.....	708
Sort.....	708
Spell checking.....	708
Start of publication record.....	708
Static table.....	709
String.....	709
Substring.....	709
Symbol.....	709
Symbolic reference.....	709
System variable.....	710
T.....	710
Tag logical element (TLE).....	710
Template.....	710
Text box.....	711
Time.....	711
Transaction table.....	711
TrueType font.....	711
U.....	711
Unlock.....	712
URL.....	712
User.....	712
User exit.....	712
User group.....	712
User password.....	713
V.....	713
Variable.....	713
Value.....	713
Vault.....	713
Vector offset.....	714
VPS.....	714
VSAM.....	714

W.....	714
Widow.....	714
X.....	715
XML.....	715
A.....	715
Zoom.....	715

43 - Appendix B Formula expressions

Working with formula conditions.....	717
--------------------------------------	-----

44 - Appendix C Interchange Journals

Creating a DIJ.....	726
---------------------	-----

1 - Licensing

To be able to use Designer, the keycode provided by your license must be entered into the Designer. These keycodes control access to the features in the Designer. Only system administrators can enter and update keycodes.

In this section

Keycode entry.....11



Keycode entry

The first time a system administrator logs on to the Designer they will be prompted for a customer name and corresponding keycodes. If the keycodes do not work or do not provide the expected functionality contact Precisely Technical Support.

To enter keycodes:

1. When you start the Designer for the first time the License Tool dialog box will be displayed. – or – select **Tools/Keycodes** from the navigator.
2. Once the License Tool dialog box has been invoked enter your Customer Name and click **Add**.
3. In the Edit Keycode dialog box enter the keycode and click **OK**.
4. Repeat the previous step for each keycode you have. Click **OK** in the License Tool dialog box when all the keycodes have been successfully included.

The screenshot shows the 'Licence Tool' dialog box. At the top, there is a 'Customer Name' field with the text 'Dotonomic' entered. Below this is a 'KeyCodes' section containing a table with three columns: 'Key Code', 'Product', and 'Expiry Date'. The table has three rows of data. To the right of the table are 'Add' and 'Remove' buttons. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

Key Code	Product	Expiry Date
[blurred]	Designer/Windows	20/12/2012
[blurred]	Generate/Windows	20/12/2012
[blurred]	PCE/Windows	20/12/2012

- Clicking Add invokes the Add Key Code dialog box. The keycode entered must correspond to the Customer Name or you will not be able to save it
- To enable the OK button the key code you enter must meet the expected format.

2 - Getting started

Designer and **Generate** provide an integrated environment for designing and publishing personalized customer communications. When you first login to the system the **Designer** will be launched. Depending on your user profile you can design and create documents and publications, manage and format the data that will be used, and prepare your documents for publishing.

Your user profile also determines the types of resources you are allowed to work with. If you have created a new repository when installing the **Designer** (typically done by the first **Designer** user at a site) you will have **Administrator** rights by default and have access to all file types.

In this section

Start-up and login.....	13
Logon details.....	13
Workflow.....	14
Design basics.....	15
Using the Designer.....	19
Input data and the data format.....	25
Publishing and production.....	27
Audit Trail.....	27
Designer preferences.....	28
Errors and information.....	29
Filtering by name using wildcard matching.....	30



Start-up and login

You can start Designer from the desktop icon or the **Start** menu. The login screen will appear. Enter a user name and password. The first time you log on (in all operating systems) enter the default Administrator account name and password. Both are 'admin' (use lower case characters).

If Designer has previously been configured to allow multiple repositories to be used, then use the **Options** button to select the repository you want to use. Example key definition

Logon details

Enter your user name and password to gain access to the Designer. Passwords are case sensitive. If you have access to more than one repository, use the **Options** button to select the one you want to use.

If you have problems logging on please contact your Designer Administrator or whoever was responsible for setting up your user account.



The screenshot shows a Windows-style dialog box titled "Logon". The main heading is "Welcome to EngageOne® Designer". Below the heading, it says "To logon enter your user name and password." There are two input fields: "User Name" with the text "admin" entered, and "Password" which is empty. At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Workflow

Within the Designer the document editor allows you to combine page layouts and document logic with references to variable data typically coming from your customer databases. Specialized controls allow you to handle the repeating data – typically transaction records or similar – that is commonplace in these kind of communications.

In the production environment such variable data is provided to Generate in an input data file. Before a publication can be put into production the type and structure of the input data must be defined within a data format. However, you do not need a data format to work with variable data in your designs. Data aliases are used to allow you to indicate indirectly the records and fields that will be available at production time. These are grouped into data dictionaries to allow reuse and control. Once a data format has been defined you can map the data aliases to the actual structures in the input data.

Document designs can be optionally included in a structured review and approval process through the integration of Designer with EngageOne Interactive. Refer to the *EngageOne Interactive User's Guide* for further information.

Document designs and other design resources are stored in a repository that provides:

- **Version control** on page 132
- Multi-user **Access control and security** on page 619
- **Review and approval** on page 446

The Designer also allows you to specify the settings that will control a publication in the production environment, such as, information about the intended output device and the host system on which the job will run. At publication time, the document designs, data format, production information, and the resources actually required to present the output from the publication are merged into a single file. This HIP file is used by **Generate** to manage production jobs.

Generate produces one or more output datastreams in a format suitable for your target output devices.

Design basics

A publication is the Designer file that is actually put into production. It references one or more document files that contain the actual page layouts and associated logic. If required, you can use the same document in multiple publications.

To start work on a new publication: in the navigator select a project for the publication and from the **File** menu select **New/Publication**. In the **Publication** dialog box enter a unique name and an optional description. A new document will be created automatically and the document editor window will be launched with this file loaded.

When you start work on a new document you will typically be creating material for the body area of the pages to be created. The area available for the body is the space that remains once page margins, headers and footers, and any reserved areas have been taken into consideration. Content in the body area flows onto new pages as required.

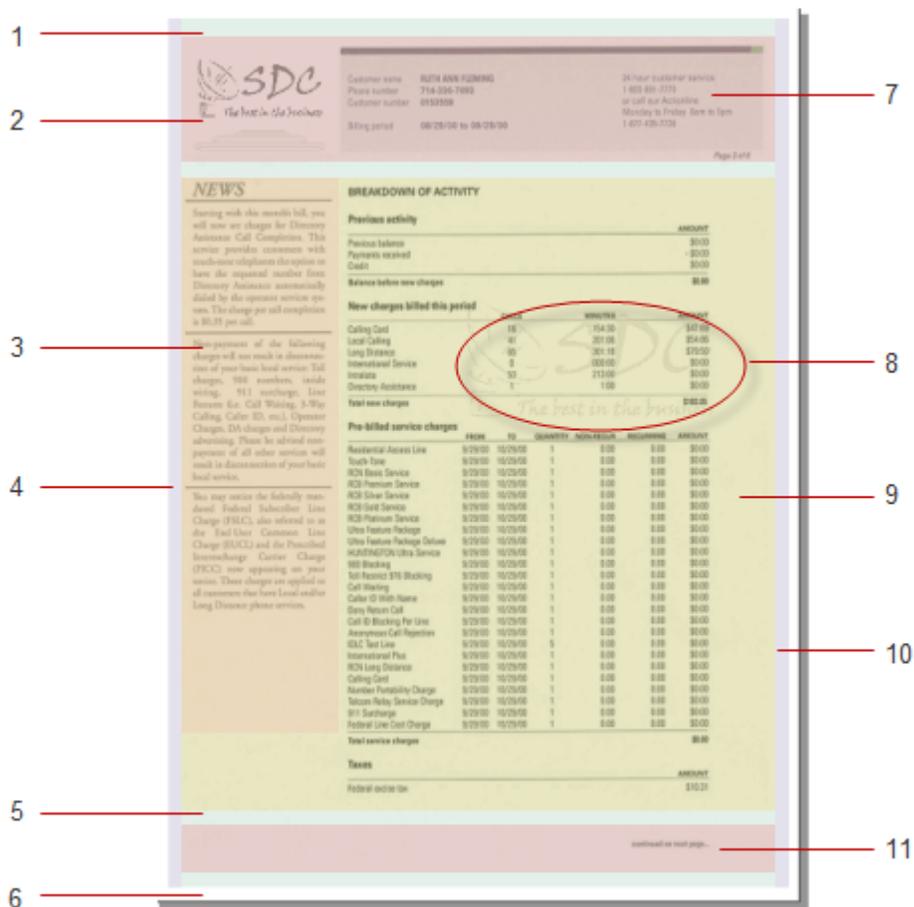
Documents also have a background layer which you can use to provide a watermark or similar effect. You can also create a reserved area on the background to specify that part of the page is unavailable for the body area. Text and other content can be added to a reserved area as normal and it is possible to update the content dynamically if required.

Headers and footers can be used to specify content that is to be repeated at the top or bottom of each page. The content of headers and footers will often include system variables that provide page numbering functions and other information about the run-time environment.

You can create text boxes that act as a container for specific content. You can use a text box simply to highlight particular content or as a reserved area which text will flow around.

All graphics including generic shapes, text boxes, charts and references to external image files are positioned relative to their predecessor in the document flow by default. You can also use fixed or inline (within paragraph text) anchor positions.

Page structure



Legend:

1	Top margin	7	Text box
2	Header	8	Graphic on the background layer
3	Reserved area on the background layer	9	Body

4	Left margin	10	Right margin
5	Bottom margin	11	Footer
6	Unprintable area		

Including data references

At the design stage the Designer uses indirect references to data elements that will be available in the input data file to be used in the production environment. By using such data aliases and data records you are able to design publications without needing to define the structure of the expected input data in advance. These aliases are mapped to actual elements of input data before the publication is put into production. It also means you can easily re-associate the aliases with new data structures if required.

You can create new data aliases or records as you work or reference existing ones. Data aliases are grouped together into a **data dictionary** that, for example, represents all the data available to a particular type of application.

You can insert references to most data aliases directly into paragraph text as required. However, when you need to reference elements within a **repeating data** structure – records or fields that appear iteratively such as transaction details – you will need to use a transaction table to control how they are positioned.

Tables and transaction tables

The **Designer** uses tables for two distinct functions. A static table is used to align text and graphics in rows and cells or generally to help with positioning objects on a page (as with most word processing tools). A transaction table is used to control how repeating data is presented; such as, how multiple iterations of the same data structure are ordered and how they are positioned in relation to one another.

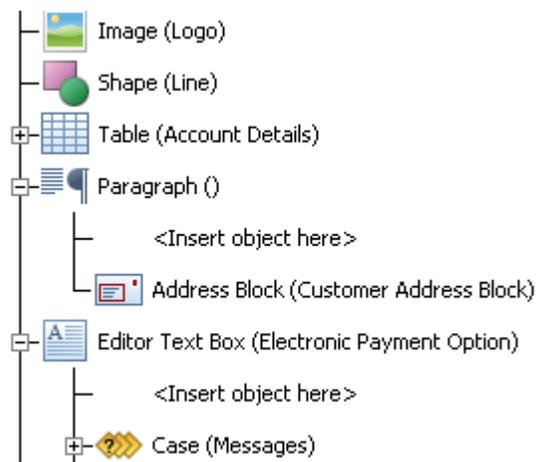
The **Table Wizard** allows you to create the basic structure quickly and easily add controls for any type of table. Once created you can modify the table layout and settings as required.

Document logic

When you are working with document designs you are actually doing two things:

- You are designing the look of the document pages. For example, the text and graphics to be presented and their relative positions.
- You are specifying the sequence of objects and associated logic that will determine how the publication is processed in the production environment.

This sequence of objects is expressed in the form of a tree in the logic map showing the document logic and processing sequence, and enables you to provide parameters to objects. Objects can generally be inserted into the document both in the logic map and directly in the document view.

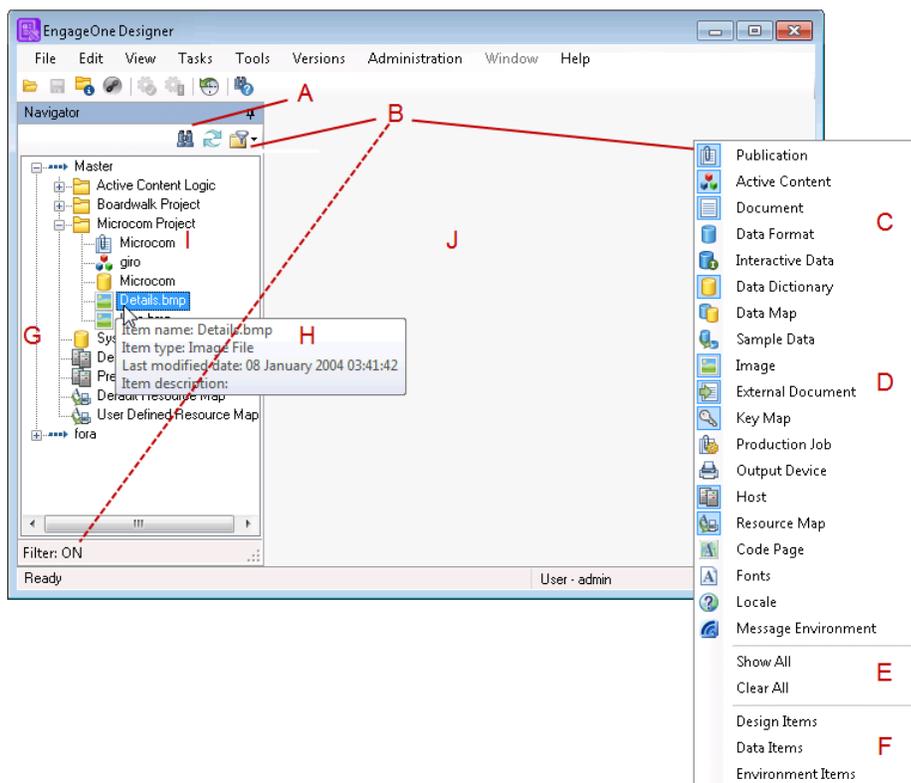


Using the Designer

The Designer window is a flexible workspace and opens initially showing the navigator window which lists the resources available in the Designer. These can be opened for viewing and editing in the workspace where each will open as a tabbed window, refer to [Using the workspace](#) on page 22 for further information.

Using the navigator

The navigator allows you to access and control all the **Designer** resources that you are allowed to use. It is presented in the form of a tree consisting of one or more projects which contain the resources. A project would typically contain work items that are used for a common purpose – for example, you might have projects for investigation, development, and production. Projects can be subdivided into folders. Also listed in the tree are items that are available to all projects such as fonts and locales.



Legend:

- **A** - use the binocular button to search for items in the navigator.
- **B** - use folder/filter button the view to show specific types. Filter activation is indicated at the bottom of the navigator.
- **C** - use Ctrl+Click to select more than one; click out of the list to close it
- **D** - you can select individual items to show.
- **E** - show all or none ...
- **F** - ... or show a pre-defined selection.
- **G** - the navigator lists all resources to which you have access.

- **H** - hover over an item to show its properties.
- **I** - expand the list as required and double click an item to open it.
- **J** - the selected item will open in the workspace.

To add a new item in the navigator: browse to the required project or folder and on the **File** menu select **New**. Alternatively, you can right-click in the navigator and select **New**. Select the required item from the drop-down list. Note that only items that are valid in the selected tree position will be available.

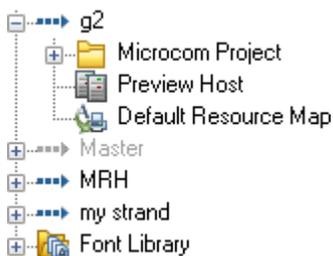
To delete an item: highlight the item in the navigator and on the **File** menu click **Delete**. Alternatively, you can right-click the file to be deleted in the navigator and select **Delete** from the context menu. The resource will be marked as deleted. You will only be allowed to delete items that are not in use by anything else. See [Deleting](#) on page 634 for more information.

The navigator view can become crowded with items that are not currently in use. You can simplify this by applying a filter to the types of item that are shown and by hiding projects that are not currently in use.

To filter item types: use the filter toolbar option to select and show the required types. Alternatively, you can use one of the predefined filters **Show all**, **Clear all**, or **Design items**.

Note: An indication that a filter has been applied is shown at the bottom of the navigator. You can use filters to search for an item by clearing all other types and viewing that item type only.

To hide a project in the navigator: browse to and right-click on the required project and select **Hide item** from the drop-down list. You will not be able to work on projects that are hidden. Clear the option to show the project and be able to work with it again.



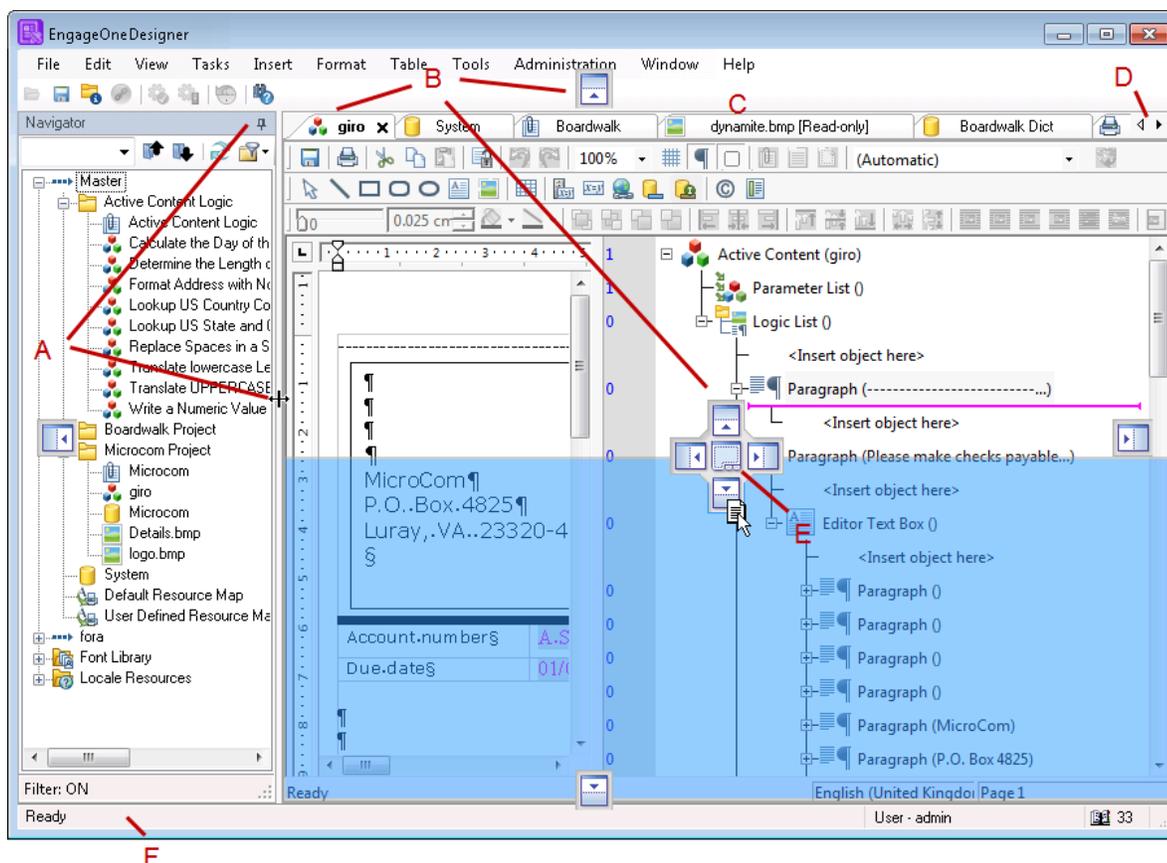
Hidden projects are shown grayed out and cannot be expanded.

To hide/show hidden items: on the **View** menu, clear **Show hidden items**. Select the option to show them again.

The navigator is by default docked on the left hand side of the **Designer** workspace. As with other windows in the **Designer** you can undock it, and use the auto-hide feature.

Using the workspace

The default view of the **Designer** workspace includes the navigator pane, with all other open objects in the form of tabbed panes. Any pane in the workspace can be re-docked to a different part of the workspace as required by dragging the tab over a docking control.



Legend:

- **A** - panes can be auto-hidden using the pin icon.
- **B** - drag a tab over a docking control to relocate it as indicated.
- **C** - reorder tabs by dragging them along the tab bar.
- **D** - scroll to hidden tabs.
- **E** - dragging a pane to the center changes it to a tabbed window.
- **F** - the status bar.

To restore the workspace view: on the **Window** menu click **Restore**. All windows will be restored to their default positions in the workspace.

Navigator keyboard shortcuts

The following keyboard shortcuts can be used within the Designer workspace to complete these common tasks more quickly:

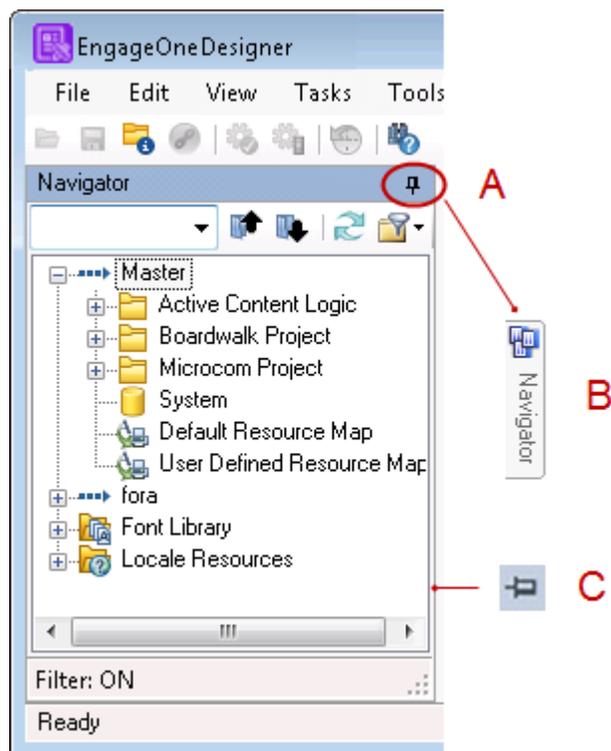
- **F2** - Rename
- **CTRL + C** - Copy with Project

Note: This also copies the full object path to the system clipboard, allowing it to be pasted into any editor for scripting or diagnostics.

- **CTRL + V** - Paste within Project
- **ALT + Enter** - Properties
- For Navigation through the tabbed panes in the Designer Workspace:
 - **CTRL+SHIFT+TAB** - run forward through the tabs
 - **CTRL+TAB** - run backwards through the tabs
- **CTRL + ENTER** - Used when entering description text in an asset's property. This shortcut allows you to add a line break to the text.

Using the auto hide feature

The auto-hide feature is available on a docked window and will only show the window when it is actively being used. This can be useful if you only refer to the window now and then and, for example, need the space for editing a document design.



Legend:

- **A** - click (un-pin) the pushpin to turn on auto-hide.
- **B** - the window will be reduced to a tab entitled Navigator at the side of the workspace. Hover over the tab to display the window. When the pointer is moved away from the window it will automatically be hidden again.
- **C** - click the pushpin again to re-dock the window.

Input data and the data format

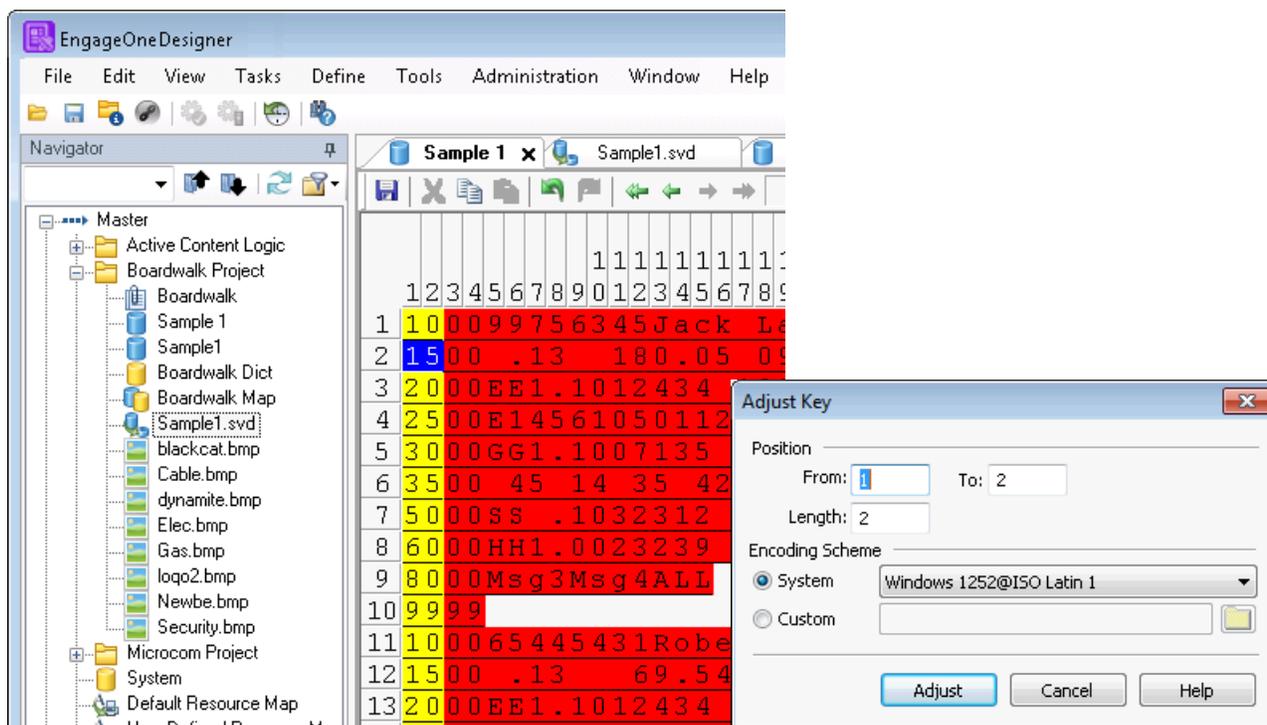
A data format defines the structure of the input data file that will be used with a Designer publication in the production environment. Any publication that needs to use input data must include a reference to a data format (as part of a **Host** settings object) before it can be put into production.

Designer supports three general types of input data:

- **Keyed Records** – a line data or 'flat' file where each line is an individual record. Record types are identified according to the contents of a consistent field within each line of data (the key field). Other fields within the records must be individually identified. You will need to supply a sample data file that acts as the template on which the record and field definitions are built.
- **Delimited** – a variation of keyed record data where field locations are automatically identifiable to the Designer by the presence of one or more consistent delimiter characters between each field.
- **XML** – the structure of the data is automatically defined by the XML but note that the data types of individual fields must still be defined. Note that **XML** support is an optional extra and is only enabled to users who install Designer with the appropriate license keycodes.

You can import a sample of the intended input data and from this create a data format file. You can then use the data format editor to define the detailed structure and content of the data. You can also import a predefined data format from Designer and apply it to your sample data. Note that applications from DOC1 Series 5 are automatically upgraded when opened in Series 6.

The screenshots below shows the use of the data format editor with keyed record sample data:



In the document editor, references to input data elements are stored as part of a data dictionary. At publication time, a data map provides a cross reference between the data elements defined within a data dictionary and those in the assigned data format. If required you can export a data dictionary from a data format so that the document design process can be aware of the exact data elements that will be available.

Publishing and production

Publications are prepared for the production environment using a **Publish** task in EngageOne Designer. You can choose to produce a single HIP file that will control the job in the production environment. This HIP file can also contain the resources required to print or present application output on the selected output devices. You can also, choose to publish the resources and the design separately. The settings you specify when publishing can be saved in a production job object for reuse. Depending on the type of publishing task selected, before you can publish a publication design you will need to provide information about the printer/browsers for which output is to be created in one or more output device objects. You will also need to create a host object that provides operating system and deployment information regarding the intended production platform.

The publishing process also updates a data map to provide links to the data format for any unresolved data aliases in your publication designs.

At the completion of a publishing task a HIP file is created and is deployed to the location indicated in the selected host object. If the necessary connectivity is available, deployment can automatically copy the HIP to your host platform using FTP or APPC connections or you can deploy the file locally and manually copy it to your production platform.

In the EngageOne Server environment, font and image resources are placed by the EngageOne publishing task within independent files that must be created and deployed in conjunction with requests from the EngageOne Server. For more information about EngageOne see [Creating EngageOne templates](#) on page 349.

In the production environment Generate is the batch program that processes jobs on your chosen host system. You will need to specify the HIP file that controls the publication as a parameter when starting the program. Generate merges input data with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

Audit Trail

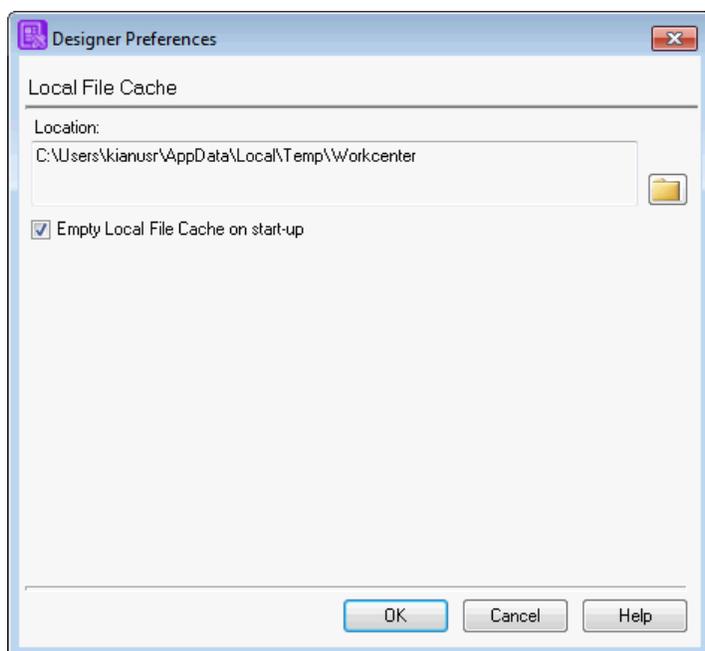
User activity in the Designer can be logged if required, and extracted for audit trail purposes or similar. This feature is turned off by default and can be enabled by the Designer administrator for your installation. Refer to the EngageOne Designer Utilities Guide for further details.

Designer preferences

The Designer provides a number of general options that enable you to customize your system. These Designer preferences enable you to do the following:

- change settings for the **Local File Cache**. This is where files in the **Design Repository** are placed when they are in use. You can change the location for the file cache, as well as set it to be cleared each time you restart the Designer.

To access the Designer preferences: from the **Tools** menu select **Designer Preferences**.



- Select **Empty Local File Cache on start-up** to clear the **Local File Cache**.
- Use the Browse button to change location for the **Local File Cache**.

Errors and information

In the Designer, errors, warnings, and information messages from any activity are shown in the **Event Log**.

Messages have a unique message **ID** associated with them which you should reference if you need to contact your product supplier when reporting a problem. A severity tag is also displayed with each message:



Information only. No user action required.



Warning. An abnormal event that may require user action.



Error. An abnormal event that cannot be corrected without user intervention.

To display the event log: click on the logs icon in the status bar. Click on the event log book icon which is displayed at the bottom right of the screen to view the log. The number adjacent to the icon indicated the total number of errors logged excluding warnings and information entries.

In the production environment messages are written to the standard output device for the platform.

Filtering by name using wildcard matching

Filtering by name using wildcard matching is permitted when performing certain Designer tasks, as follows:

- Data Dictionary and Data Map Generation Wizard (Update Existing Data Dictionary), refer to [Running a publishing task](#) on page 602.
- Adding publications/resources to a production job, refer to [Generating a dictionary and data map](#) on page 501

Click on the  button to perform case-insensitive, wildcard matching.

Example	Description
* (or an empty filter)	matches everything
A*	matches everything starting with 'A', in this case Arial is matched and My Arial is not matched.
A	matches everything with an 'a' in the name, in this case Arial and My Arial would be matched.
L*image*	matches item names starting with 'L' and containing 'image'. In this case Logolmage.jpg and LionsShareImages.png are matched, however, MyLogolmage.png is not matched.

Note: You can use multiple wildcards as required.

3 - Searching for assets

Designer provides two search options for finding assets in your Designer repositories:

Search - allows you to perform searches for assets with a particular name in your repository.

Advanced search - provides more sophisticated search capabilities using search queries.

In this section

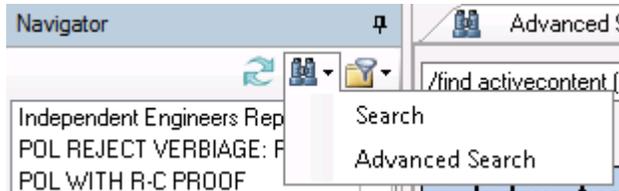
Accessing search features.....	32
Working with the search feature.....	33
Working with the advanced search feature.....	34



Accessing search features

Use the binocular icon from the navigator.

Click on the drop-down arrow and select the type of search you require as shown below:



To activate the Search via the keyboard: Press Alt+E+S

-or-

Click the F3 shortcut key

To activate the Advanced search via the keyboard: Press Alt+E+V

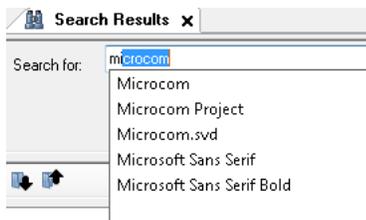
-or-

Click the Ctrl+F3 shortcut key

Note: The **Advanced Search** option is only available when advanced search feature is installed.

Working with the search feature

Select the **Search** option by clicking on the binocular icon in the navigator to perform a basic named search. As you type in the search box, a list of potential matches is shown beneath the search entry field:



You can either select the required asset from the drop-down list or type the complete name of the asset you are looking for. Press **ENTER** or click on **Search** button to populate the search results window.

Working with the advanced search feature

The advanced search feature allows you to:

- Create complex searches to find specific asset types and their associated properties using Designer's built-in search query language.
- View search query results in a grid format, allowing you to drill-down into further detail on a selected search result.
- Use context menu options to:
 - Locate and open the selected asset in the navigator.
 - Determine where the selected asset is used and the resources it uses.

Example queries:

The following query returns all Arial fonts in your repository:

```
/find font "arial"
```

The following query returns all images named logo in your repository:

```
/find image "logo"
```

The search query returns results in a grid format, allowing you to drill-down into further detail on a selected search result. For more information, refer to [Performing an advanced search](#) on page 35.

Performing an advanced search

This section is an introduction to the advanced search feature with examples. For detailed information about asset types, properties and operators, refer to [Asset types and properties](#) on page 42. For more detailed search examples refer to [Search query examples](#) on page 53.

To perform an advanced search:

- Enter text directly into the search entry field; this executes a text search, returning assets containing the search text in the:
 - asset itself
 - asset's name
 - asset's description
 - asset's path.
- Enter a */find* type query. You must specify the type of asset you are looking for and optional properties associated with the chosen asset. For detailed information about indexed assets, properties and the search query syntax, refer to [Asset types and properties](#) on page 42.

Note:

- The length of your search cannot exceed 1024 characters.
- You can delimit property values within your search query either with ' or " characters.

For example:

`/find text 'a'` and `/find text "a"` are valid search queries.

- Property values are not case sensitive.

The following asset types are indexed and can be used as the subject of a search query:

• Publications

```
/find publication name 'test'
```

Returns a list of publications with **test** anywhere in the name of the publication.

Note: that the keyword **pub** can be used in place of **publication**, e.g.

```
/find pub name 'test'
```

• Documents

```
/find document text 'enter'
```

Returns a list of documents with paragraph text containing **Enter**.

Note: that the keyword **doc** can be used in place of **document**, e.g.

```
/find doc name 'enter'
```

- **Active content**

```
/find activecontent publishable 'true'
```

Returns a list of all publishable active content items.

Note: that the keyword **ac** can be used in place of **activecontent**, e.g.

```
/find ac name 'true'
```

- **Fonts**

```
/find font typeface ('Arial' or 'Times') pointsize (greater '11.5' and  
less '14') usedby activecontent
```

Returns a list of specific fonts within a range of point sizes that are only used by active content.

- **Images**

```
/find image name 'logo' imageformat ('GIF' or 'PNG') colorformat is  
'RGB24' resolution less '96' depth '24' creationdate after '2014-04-29'
```

Returns a list of all images:

- containing **logo** anywhere in the image's name.
- that are either GIF or PNG format.
- where the colorformat is **RGB24**
- with a resolution of less than 96 dpi
- having a color depth of 24
- created after the 29th April 2014

- **Data Alias**

```
/find dataalias name 'total_amount' usedby pub
```

Returns a list of data aliases used by publications.

Note: that the keyword **alias** can be used in place of **dataalias**, e.g.

```
/find alias name 'total_amount' usedby pub
```

- **Data Field**

```
/find datafield name 'total_amount'
```

Returns a list of data fields with 'total_amount' in the name.

Note: that the keyword **field** can be used in place of **datafield**, e.g.

```
/find field name 'total_amount'
```

Loose and Exact matching

There are two methods of matching for string properties that contain text, Loose matching and Exact matching. Loose matching is based on words in the text and is not order-specific. Exact matching is based on the characters in the text and allows the use of wildcards.

Loose match search

Loose matching example:

```
/find publication name 'copy right'
```

This search query will return results where the word **copy** or the word **right** appear anywhere in the name of a publication such as "Copy of statement", "Your right to cancel" and "Right copy". The query would not return publications named "copyright".

Exact match search

Exact matching examples:

```
/find publication name IS 'Copy'
```

Finds any publications in the current repository called **copy**.

Exact match examples using wildcards

Wildcards are special characters that can stand in for unknown characters in a text value.

Note:

* - matches any sequence of characters

? - matches any single character

The following exact match search queries use wildcards to find named publications :

```
/find publication name is "copy?????"
```

Finds publications in the current repository whose name starts with **copy** followed by 5 characters such as **copyright**.

```
/find publication name is "copy*"
```

Finds publications named **copy**, **copy of right** and **copyright**.

Note: paragraph matches returned from a wildcard search are **not** highlighted in the results grid.

The USED BY query clause

The **usedby** clause allows you to identify assets that are used by another set of assets.

For example, you may want to search for all Arial fonts that are above 8 point which are used by any active content items in the project called "TestProj". This search can be achieved using the following type of query:

```
/find font typeface 'Arial' pointsize greater '8' usedby activecontent
path is 'TestProj/*'
```

The **usedby** clause can be preceded by the **not** keyword. For example, to find data aliases in the Master project that are not used by any publication, document or active content, the query would be:-

```
/find alias path is 'master/*' not usedby pub or ac or doc
```

There are a few points to note when using the **usedby** keyword:

- Adding a **usedby** clause to a query can make it slower. Adding property filters to the query will narrow down the search and improve performance.
- The number of search results shown under the search query field will be an indicative value until all the results have been retrieved.

The USING query clause

The **using** clause allows you to identify assets that use another set of assets.

For example, you could search for all active content assets in the project called "TestProj" that use Arial fonts that are above 8 point. This search can be achieved using the following type of query:

```
/find activecontent path is 'TestProj/*' using font typeface 'Arial'
pointsize greater '8'
```

The **using** clause can be preceded by the **not** keyword. For example, to find publication, document or active content assets that are not using any data aliases in the "master" project, the query could be:-

```
/find pub or ac or doc not using alias path is 'master/*'
```

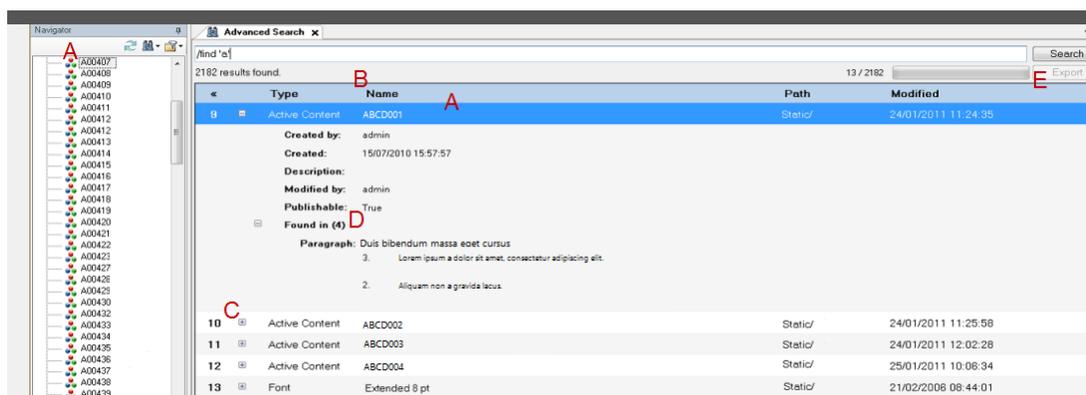
There are a few points to note when using the **usedby** keyword:

- Adding a **using** clause to a query can make it slower. Adding property filters to the query will narrow down the search and improve performance.

- The number of search results shown under the search query field will be an indicative value until all the results have been retrieved.

Search results

Use the advanced search to find assets in your repository by typing your search query. Refer to [Performing an advanced search](#) on page 35 for further information. Results of your search are displayed in a grid format. The initial results are displayed in a summary form as shown below:



Legend:

- **A** - Double click on a result in the grid to locate the asset in the Navigator treeview;
 - Note that for data field/alias search results, the interactive data asset to which the alias/field belongs to will be highlighted in the Navigator.
- **B** - Sort by clicking on the column titles;
- **C** - Click on the  button to expand the required search result. Expanding the search result will show properties for the selected asset type. For example, for fonts, attributes such as typeface, bold, italic etc. are presented. Expanding an image search result presents attributes such as image format, resolution, and so on.
- **D** - For queries that only specify a text value, such as */find 'copy right'*, the results will include text from matched paragraphs for the specific asset.
- **E** - Click on the **Export** button to export your search query results to a CSV formatted file. Refer to [Exporting search results](#) on page 40 for further information.

Note: The text in the search results is only shown for specific queries and is not always highlighted.

Context menu options

Use the right mouse button on a selected search result to access the following shortcuts:

- **Select in Navigator** - Locates the selected asset in the navigator. Shortcut key **Ctrl+N**

- **Open...** - Locates and opens the selected asset in the navigator. Shortcut key **Ctrl+O**
- **Expand Details** - Expands or collapses the selected search result.
- **Show dependency** - Presents information such as where the selected asset is used and the resources it uses. Refer to [Displaying object associations](#) on page 635 for further information. Shortcut key **Ctrl+D**

Sorting results

Method 1: Sort the results of your query by specifying the order in the query.

Find the paragraph text "abc", look in all asset types that contain paragraph text. Order results by asset type then name in descending order:

```
/find text "abc" order type name desc
```

Find the paragraph text "abc", show only assets that are used by documents. Order results by name in ascending order:

```
/find text "abc" usedby document order name asc
```

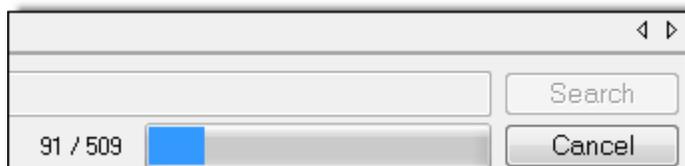
Note: If the asc/desc keywords are omitted the default, ascending order is applied.

Method 2: The number of times a column header is clicked determines the sort order of the results:

- Once, to sort in ascending order;
- Twice, to sort in descending order;
- Three times to remove the sort.

Exporting search results

You can export the results of your search by clicking the Export button located above the search results grid. You will be prompted to save the exported CSV file in your local environment. An indication of progress is presented during the export process, as shown below.



Once the export has been initiated the Export button is replaced by the Cancel button allowing you to stop the export process if required.

The resultant CSV file contains the results of the search query and, also, dependent information for each asset found. Because of this, the number of rows in the exported file will not be the same as that of the result grid. The exported file contains the following information:

- **Record Type**

- The Match row indicates where the asset was found.
- The Dependent row(s) indicates the parent of the match record.
- The Query row presents details of the search query used to generate the report. Note that most columns are blank for Query records.

Note: only the assets indexed by advanced search feature are displayed in the match and dependent rows. The asset **types** currently indexed are publications, public documents, active contents, images and fonts.

What are Match and Dependent report rows ?

In the example:

/find font 'arial' pointsize '20'

The search will find the Arial 20pt font - this is the Match

Arial 20pt is used in documentA, documentB,documentC and active contentD - these are dependents.

- **Type** - indicates the type of asset, e.g. font, active content, document, etc.
- **Name** - the asset's name.
- **Path** - the location of the asset in the repository.
- **Description** - the search query used.
- **Modified** - last modification date.
- **Modified By** - the user who last modified the asset.
- **Hits** - for **text** searches only; the number of times the paragraph text appears in the matched asset.
- **Repository Id** - the repository alias.

Note: In the case of single repository installations where there is no repository alias, this is the name of the database.

- **Id** - is a unique id for the asset.
- **Match Id** - refers to Dependent records, this is the unique id of the related Match asset.

Note that the number of **Match** rows exported and the number of **Dependent** rows exported for each **Match** row is limited to 50,000. The export process can be a lengthy operation for large result sets.

Asset types and properties

Type	Property	Validation	Description
any	all type specific properties or common properties		This is the default type that represents all types where a type is not supplied, for example: <code>/find text "A text"</code>
publication	text	String	Paragraph text for publication, for example: <code>/find publication text "A text"</code>
	name	String	A publication, for example: <code>/find publication name "My pub"</code>
document	text	String	Document paragraph text, for example: <code>/find document text "My text"</code>
	name	String	A specific document, for example: <code>/find document name "My pub"</code>
activecontent	text	String	Paragraph text for active content, for example: <code>/find activecontent text "PublishableAC"</code>
	name	String	An active content, for example: <code>/find activecontent name "My AC"</code>
	publishable	Boolean	This value is set to True when the active is Publishable and False when it is not.

Type	Property	Validation	Description
font	name	String	A font, for example: <pre>/find font "Arial" pointsize "10.5" charset "ansi"</pre>
	typeface	A string - for example: "Arial"	
	fontFormat	Fixed String	Font format type, one of: <ul style="list-style-type: none"> • Type1 • TrueType • OpenType • OpenTypeCff
	pointsize	Number - for example: 11.5	
	bold	Boolean	When the weight is greater than or equal to 700 then this value will be set to True otherwise it will be set to False .
italic	Boolean	This value is set to True if font is italicized otherwise it will be set to False .	

Type	Property	Validation	Description
	charSet	Fixed String - for example: "Ansi"	<p>The character set name, it can be one of the following values:</p> <ul style="list-style-type: none"> • Ansi, • Default • Symbol • Mac • ShiftJis • Hangul • Johab • Gb2312 • ChineseBig5 • Greek • Turkish • Vietnamese • Hebrew • Arabic • Russian • Thai • EastEurope • Unicode • Oem
	fontWeight	Integer value, for example: "400"	<p>In general:</p> <ul style="list-style-type: none"> • 400 - normal • 700 - bold
	originalFontName	String - for example: "MyCustomFont.ttf"	
image	name	String	<p>An image, for example:</p> <pre>/find image "Logo1"</pre>
	imageFormat	Fixed String - for example: "PNG"	

Type	Property	Validation	Description
------	----------	------------	-------------

imageformat can be one of the following values:

- BMP Windows/OS2 bitmap
- EPS Adobe Encapsulated PostScript file
- EPS2 Adobe Encapsulated PostScript (embedded JPEG)
- FS10 IBM IOCA FS10
- FS45 IBM IOCA FS45
- GIF CompuServe Graphics Interchange
- JPEG JPEG or JFIF compliant
- PCL HP PCL raster
- PCX ZSoft IBM PC Paintbrush
- PNG Portable Network Graphics
- PPML Personalised Print Markup Language Format
- PSEG IBM IOCA Page Segment
- TIFF Tagged Image Format
- VIPP Variable Data Intelligent PostScript Printware Image format
- VPS Variable Print Specification PostScript Image Dictionary format
- XrxImg Xerox Image
- LarMeta Legacy Designer image format
- AfpOvl AFP Overlay
- XrxFrm Xerox Form
- ICO Microsoft icon
- JNG JPEG Network Graphics
- MIFF Magick Interchange Format File
- MNG Multiple-image Network Graphics
- P7 Xv Thumbnail format
- PBM Portable bitmap format
- PCLBW HP PCL raster (Black and white)
- PDF Portable Document Format
- PGM Portable graymap format
- PICT Apple Macintosh QuickDraw/PICT
- PIC Alias/Wavefront RLE image
- PNM Portable anymap format
- PPM Portable pixmap format
- PS Adobe PostScript
- PSD Adobe PhotoShop bitmap
- SCT Scitex HandShake
- SGI Irix RGB image
- SUN Sun Raster file
- SVG Scalable Vector Graphics
- TGA Truevision Targa image
- VDX Variable Print Specification PostScript Image Dictionary Format
- WBMP Wireless Bitmap image
- WPG Word Perfect Graphics

Type	Property	Validation	Description
	resolution	Integer	
	width	Integer	The image width in pixels
	height	Integer	The image height in pixels
	depth	Integer	The color depth (generally 1, 8 or 24 but could be other)
	colorFormat	Fixed String	Color encoding type, it can be one of the following values: <ul style="list-style-type: none"> • BW (black and White) • RGB8 • RGB16 • RGB24 • CMYK - 32 bit CMYK • Unknown - may occur in older repositories
dataalias	aliastype	Fixed String	The data alias type, can be one of the following values: <ul style="list-style-type: none"> • Field • Record
	datatype	Fixed String	The data type, can be one of the following values: <ul style="list-style-type: none"> • Integer • Number • String • Date • Time • Currency
	datapath	String	
	dictionaryname	String	
	dictionarypath	String	

Type	Property	Validation	Description
	mapname	String	
	mappath	String	
	formatname	String	
	formatpath	String	
	interactive	Boolean	<p>Limit the search to Interactive publications only, can be one of the following:</p> <ul style="list-style-type: none"> • true • false • yes • no
datafield	datatype	Fixed String	<p>The data type, can be one of the following:</p> <ul style="list-style-type: none"> • String • Date • Number • Decimal • NativeBinary • LsbBinary • MsbBinary • PackedDecimal • SignOverstruck • Integer • Time
	datapath	String	
	formatname	String	
	formatpath	String	
	interactive	Boolean	

Common properties

The table below lists common properties that can be applied to most asset types:

Type	Property	Validation	Description
publication document activecontent	text	String	Paragraph text, for example: <code>/find publication text "Summary"</code> <code>/find text "Summary"</code>

Type	Property	Validation	Description
publication document activecontent font image any	anyproperty		This is the default property and is applied where a property is not provided, Only String properties will be searched, for example: <code>/find "Arial"</code>
	name	String	The name of the asset, for example: <code>/find publication name "Prod"</code>
	creationdate	Datetime	The date and time that the asset was created, for example: <code>/find document creationdate "2017-05-12"</code>
	creatorname	String	The Designer user name of the person who created the asset, for example: <code>/find document creatorname "Tommy Atkins"</code>
	description	String	The description of the asset entered in Designer, for example: <code>/find document description "some text"</code>
	modificationdate	Datetime	The date and time that the asset was last modified, for example: <code>/find document modificationdate "2017-05-12"</code>
	modifiername	String	

Type	Property	Validation	Description
			<p>The Designer user name of the person who last modified the asset, for example:</p> <pre>/find document modifiername "Tommy Atkins"</pre>
	path	String	<p>The path of the asset in the Designer Navigator tree.</p> <pre>/find document path is "Project/Folder/"</pre>
any	type	Fixed String	<p>Any type, can be one of the following values:</p> <ul style="list-style-type: none"> • activecontent • document • font • image • publication <pre>/find any type ("font" or "image")</pre>

Query operators

Operator	Asset	Property value	Value type	Description	Example
or	✓	✓	any	Logical operators that can be used both for asset's and value's logical operations	<code>/find text ("abc" or "def" and "ghi")</code>
and	✓	✓	any		<code>/find text ("abc" and "image")</code>
not	✓	✓	any		<code>/find text is "Te*"</code>
is	not applicable	✓	string, fixed string (except Type property)	Exact match operator for text search	<code>/find font name isnot "*10*"</code>
isnot	not applicable	✓	string, fixed string (except Type property)	Exact match operator for text search	

from	not applicable	✓	date, numeric	A range for date/number greater or equal than a provided date/number	/find document modification-date (after "2015-12-31" and before "2017-01-01" or "2017-01-05")
to	not applicable	✓	date, numeric	A range for date/number lower or equal than a provided date/number	/find document creationdate (from "2016-01-01" and to "2016-12-31" and (not "2016-12-25"))
after	not applicable	✓	date	A range for date greater than a provided date	
before	not applicable	✓	date	A range for date lower than a provided date	
greater	not applicable	✓	numeric	A range for number greater than a provided number	/find font pointsize greater "18"
less	not applicable	✓	numeric	A range for number lower than a provided number	/find font fontweight less "500"

Default operators:

If an operator is not specified between a property and value in the search query, the default operator is used. With the exception of **String** properties, the property has to be the same as the value specified, for example:

```
/find resolution '96'
```

will find images where the resolution is 96.

For **String** properties, the default operator is a loose match which matches based on the words in the value specified in any order, for example:

```
/find text 'copy right'
```

will find active contents, documents and publications with the words **copy** or **right** in their paragraph text such as **Copy of statement**, **Your right to cancel** and **Right copy**.

Note: This query will not match the word **copyright** in paragraph text.

Search query examples

Text searches

Find any asset with the paragraph text **abc**:

```
/find text "abc"
```

Find documents with the paragraph text **abc**:

```
/find document text "abc"
```

Find active content with the paragraph text **abc**:

```
/find activecontent text "abc"
```

Exact match - find assets with paragraph text containing **1.5**

```
/find text is "1.5"
```

Find any asset with the paragraph text **abc**. Order results by asset type and name

```
/find any text "abc" order type name
```

Note: since **any** is the default asset type it can be omitted as follows:

```
/find text "abc" order type name
```

Find documents with paragraph text **abc**, order result by the document name.

```
/find document text "abc" order name
```

Find assets with the paragraph text **abc**, show for all assets except activecontent; order result by name:

```
/find any text "abc" type not "activecontent" order name
```

Asset searches

Find all **Arial** fonts:

```
/find font typeface "Arial"
```

Find all **Corporate** fonts with a pointsize of **11.5**

```
/find font typeface "Corporate" pointsize "11.5"
```

Find all fonts starting with **Corporate** in the name:

```
/find font name is "Corporate*"
```

Find all **Corporate** or **OldCorporate** fonts with a pointsize of **11.5**:

```
/find font typeface ("Corporate" or "OldCorporate") pointsize "11.5"
```

Find all **Corporate** or **OldCorporate** fonts with a pointsize of **11.5** used by active content:

```
/find font typeface ("Corporate" or "OldCorporate") pointsize "11.5"
usedby activecontent
```

Find image assets that have specified width, height, depth and colorformat properties

```
/find image width "100" height "200" depth "24" colorformat "RGB8"
```

Find image assets that have a resolution greater than 300dpi and order results by name

```
/find image resolution greater "300" order name
```

Find any assets that are used by active content

```
/find any name is "*" usedby activecontent
```

Note: since **any** is the default asset type it can be omitted as follows:

```
/find name is "*" usedby activecontent
```

Find any asset that contain **12** in its name that is not a font and is used by document; order the result by type and name

```
/find name "12" type not "font" usedby document order type name
```

Other wildcard search queries

/find text is "*" - shows all assets containing paragraph text.

/find font is "*" - shows all fonts in the database.

Searching for the * character

`/find "*" - shows all assets containing * in the asset's name or, paragraph text.`

`/find text "*" - shows all assets containing * in their paragraph text.`

`/find font "*" - shows all fonts with the character * in the font's name.`

4 - Working with the editor

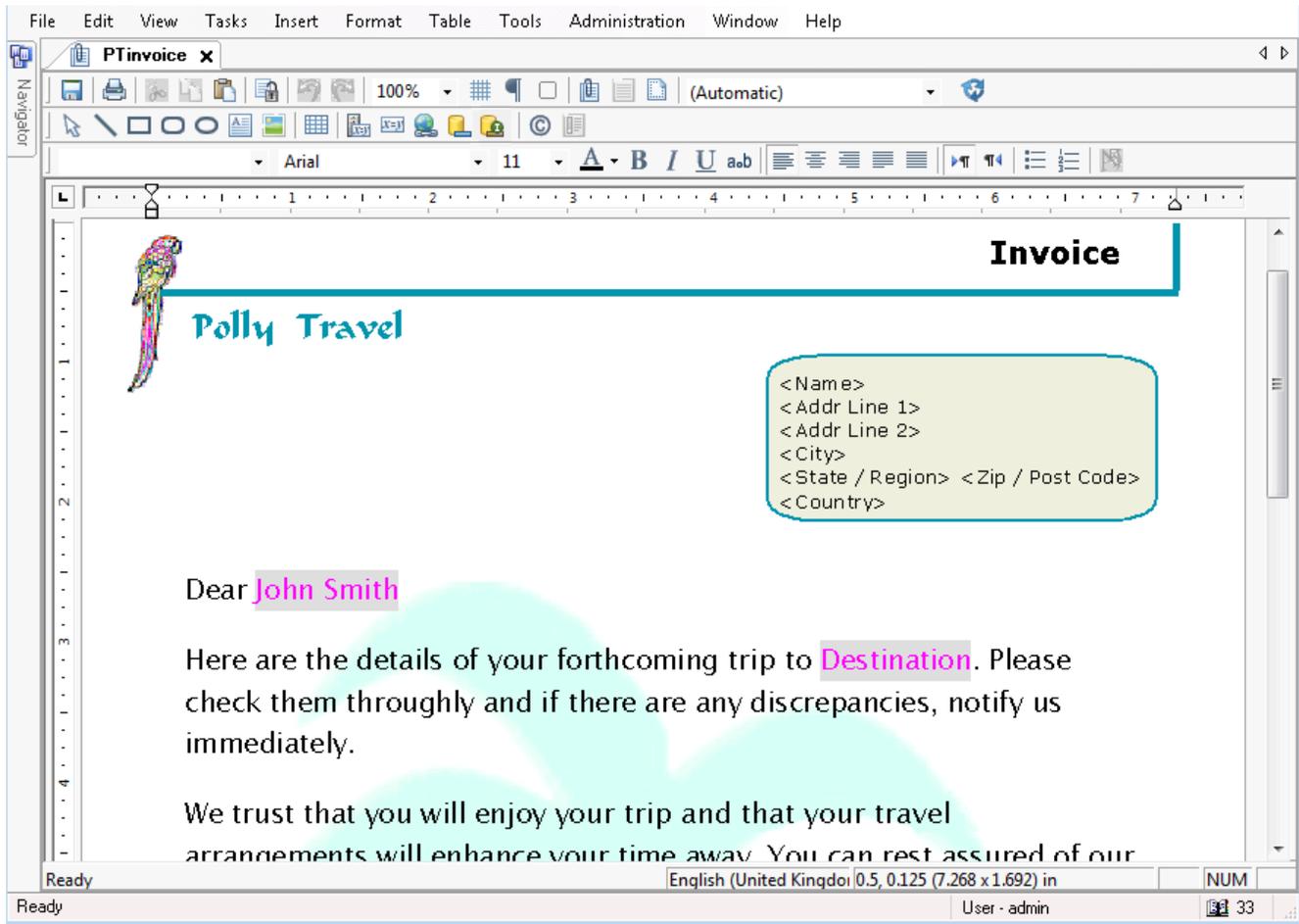
The document editor is where you design actual page layouts. When you create a new publication the editor will be invoked with a blank page ready for your input. As well as the main page work space, the logic map view will be enabled when the editor is launched. Refer to [Using the logic map](#) on page 150 for details.

In this section

The editor workspace.....	57
Editing features.....	58



The editor workspace



- Use the toolbar buttons as shortcuts for settings and functions. Choose which toolbars you see using the View menu.
- The top ruler can be used to set tab stops and indents for the selected paragraph. Right-click on it for options
- The coordinates indicator gives information about offset and size when a graphic is selected.
- Variable data is highlighted in the document area in magenta colored text.
- Click on the book icon on the bottom right of the workspace to enable event log information.

To create a new publication or document: in the navigator **File** menu select **New Publication**.

By default, you will be working with the document view where content intended for the body area of the document is created. You may also want to work with the page setup view which allows you to

create content for headers and footers and the background layer. Both views share the same editing features. See [Page Layout](#) for details of these views and the page areas they allow you to define.

Editing features

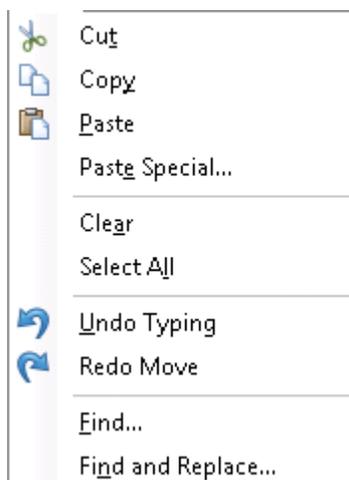
The editor provides standard editing features that enable you to create documents. The menus and toolbars give you access to all the editing options. The layout of these can be customized to some extent.

Text and graphics can be copied and moved by selecting it and using the clipboard as a temporary store for it. If you make a mistake when editing, you can retrace your steps using the undo and redo options.

Menus and toolbars

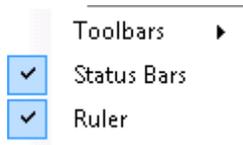
The menu gives you access to most of the editor functions and commands. You will also be able to access many of these using the toolbar buttons or keyboard access keys, both of which will be shown on the menu.

Note: You may have to press the <Alt> key in order to see the access keys.



Buttons are grouped together into toolbars with similar editor functions. You can hide toolbars to simplify the layout of the editor window. You can also move the toolbars in relation to one another to suit your needs.

To hide or display a toolbar: on the **View** menu click **Toolbars** and then clear or select the relevant toolbar check box.



To move a toolbar: drag its docking bar to a new position.

Measuring and positioning aids

You can customize the unit of measure (UOM) used by the editor to suit your requirements. The UOM specified is used in rulers and coordinate displays.

The preferred UOM is also used as the default in dialogs that require a measurement although you can override this as required. You can specify a different UOM simply by adding one of the following abbreviations after the number.

UOM	Specify	Example
inches	in	2.1 in
centimeters	cm	1.2 cm
millimeters	mm	32 mm
points	pt	12 pt
picas	pi	2 pi

The on-screen rulers help you with positioning objects accurately. The top ruler is also used when setting tab stops and indents for paragraph text. You can turn off the ruler display if required by using the **View/Ruler** option.

Note: For information about setting tab stops and indents using the ruler, see [Positioning text](#) on page 77.

You can also use the grid to help with placement. You can set the grid size to your preference and optionally have graphics snap to grid intersections when you are creating or moving them.

To set all grid, ruler and UOM preferences: from the **Tools** menu select **Options** and apply settings as required.

To override ruler settings and UOM: right-click on the ruler and select from the drop-down list.

To turn the grid on and off: click on the **Toggle Grid** button– or –on the **View** menu click **Grid**.

Selecting content

When you want to move, delete, or overwrite design elements you must first select what you want to work with. You can do this either by using the mouse or via the keyboard. Refer to Keyboard shortcuts in this section for further information.

To select text and graphics using the mouse: use one of the following shortcuts:

Anything:	drag across it from beginning to end.
A word:	double-click in it.
A paragraph:	triple-click in it.
Start a selection:	click at start of selection.
End a selection:	hold down SHIFT and click at end of selection.

To cancel a selection: click anywhere else on the page or press any arrow key.

To delete selected elements: use the **Delete** key on the keyboard or click **Clear** from the Edit menu.

Using the clipboard

You can copy, cut and paste design elements using the system clipboard. This is an easy way of transferring content both within the Designer editor and to and from other applications.

Refer to [Keyboard shortcuts](#) on page 64 for further information.

Undo and redo

If you create, edit or format a design element by mistake, you can reverse the action by using the **Undo** command. Similarly, if you undo an action by mistake, you can use the **Redo** command to

switch back again. You can undo actions progressively, each time restoring to the state before the previous change was made, up to a limit of 10 previous actions.

To undo an action: click the **Undo** toolbar button

-or-

from the **Edit** menu click **Undo**.

Note that the action that will be canceled is shown in the **Undo** command.

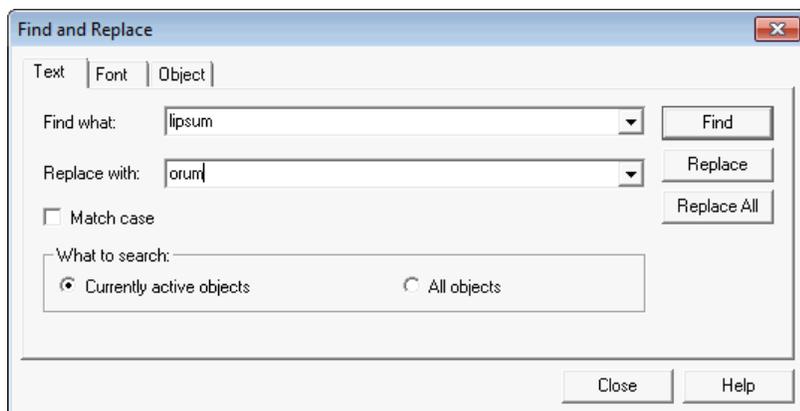
To reverse an undo action: click the **Redo** toolbar button

-or-

from the **Edit** menu click **Redo**. Note that the action that will be reversed is shown in the **Undo** command.

Find and replace

You can search and make changes to text, fonts, character styles and value objects in your design documents by using the EngageOne Designer find and replace feature. You can replace specific instances of text and value objects or globally change every occurrence within the document. For fonts and styles you can replace every occurrence in both Design view and Page Setup view.



- The Font tab is used to find and replace the font, font style, size, color and language.
- The Object tab is used to find and replace values, for example data aliases and variables.
- You can search for text as shown in the editor or, extend the search to the logic map including all hidden objects, (such as conditional). The editor will be searched first, then the logic map
- You can extend the search to the logic map including all hidden objects

Note: In a Preview for EngageOne you can only find and replace text in an editable text section. Also the What to search options are not applicable for EngageOne. For information on searching the logic map see [Using the logic map](#) on page 150. For details on working with values see [Adding values to objects](#) on page 171

To find and replace: from the **Edit** menu select **Find and Replace**. Select the relevant tab to search for **text**, **fonts** or **objects**.

Refresh

Refresh

The refresh option is used to refresh the view of the document being edited in the editor.

To refresh the editor view: select *View/Refresh*

-or-

press F5.

Boundary indicators

You can optionally choose to show the outline of layout areas such as text boxes, address blocks, charts and so on. Boundaries are indicated by a gray outline around the layout area. Select the object by clicking anywhere in the layout area or on its border.

To switch boundaries on and off: click the toolbar button

-or-

from the **View** menu choose **Show Boundaries**.

Note: By default boundaries are switched on.

Keyboard shortcuts

You can use keyboard shortcuts to move the focus to a frequently used menu option or command without using the mouse. This section lists the keyboard shortcuts available in the document editor.

To activate the menu bar via the keyboard: Press the Alt key once, each menu option identified by a single underlined letter which indicates its shortcut key. Pressing on the Alt key again will return focus to the editor.

To navigate the menu bar via the keyboard:

1. The required menu is presented by pressing the appropriate shortcut key. You navigate between menu options using the Left and Right keys.
2. You can navigate through the menu using the following cursor keys:
 - Use the up and down arrows to move up and down the list of menu items
 - Use the Esc key to cancel the menu
 - Use the Right or Enter keys to open a submenu
 - Use Left or Esc keys to close a submenu and return to the parent menu.

Working with toolbar keyboard shortcuts

When working with combo boxes use the following to navigate and select items:

- To move up one item use the up arrow
- To move down one item use the down arrow
- To select an item from the combo box press the return key

To exit from the toolbar control, press the “Esc” (Escape) key.

Toolbar options

Clipboard

Cut	CTRL+X
Copy	CTRL+C
Paste	CTRL+V

Actions

Undo	CTRL+Z
------	--------

Redo	CTRL+Y
------	--------

View

Select magnification from the zoom box	CTRL+SHIFT+Z
--	--------------

Toggle grid	F9
-------------	----

Toggle Formatting Marks	CTRL+SHIFT+8
-------------------------	--------------

Toggle Boundaries	CTRL+SHIFT+9
-------------------	--------------

Language

Select document Language box:	CTRL+SHIFT+E
-------------------------------	--------------

Text formatting

Select the font from the font box:	CTRL+SHIFT+F
------------------------------------	--------------

Select font size from the size box:	CTRL+SHIFT+P
-------------------------------------	--------------

Select text color options:	CTRL+SHIFT+C
----------------------------	--------------

Make selected text bold:	CTRL+B
--------------------------	--------

Make selected text italic:	CTRL+I
----------------------------	--------

Underline selected text:	CTRL+U
--------------------------	--------

Keep lines together:	CTRL+K
----------------------	--------

Align left:	CTRL+L
-------------	--------

Center text:	CTRL+T
--------------	--------

Align right:	CTRL+R
--------------	--------

Justify:	CTRL+J
Full justify:	CTRL+SHIFT+J
Start a bulleted list:	CTRL+SHIFT+L
Start a numbered list:	CTRL+Alt+L

Other keyboard shortcuts

Spell check:	F7
Invoke the Font dialog:	CTRL+D
Display the font dialog when working with editable text:	CTRL+D

To select paragraph design elements:

A character to the right:	SHIFT + Right arrow
A character to the left:	SHIFT + Left arrow
A line before:	SHIFT + Up arrow
A line after:	SHIFT + down arrow
To end of line:	SHIFT + END
To beginning of line:	SHIFT + HOME
All	CTRL + A or on Edit menu click Select All

To navigate through frames/objects:

Switch between editor and logic map views:	F6
--	----

Move to next text box or image: CTRL+6

Move to previous text box or image: CTRL+SHIFT+6

5 - Creating paragraphs

Paragraphs are created by typing with the insertion point positioned in the editor. Once a paragraph has been created you can then select it from the logic map to display the Paragraph dialog box. This enables you to configure a number of settings for the selected paragraph. Follow the links below for instructions on performing basic paragraph formatting.

In this section

Entering and formatting text.....	69
Positioning text.....	77
Bullets and numbering.....	85
Pagination.....	86
Conditional paragraphs.....	88
Merging paragraphs.....	89
Importing paragraph content.....	89
Symbols.....	90
System variables.....	91
URL links.....	93



Entering and formatting text

Paragraphs are created simply by typing with the insertion point positioned in the body area of the document view or in a control area such as a text box or header and footer space. A new paragraph is automatically created when you press the return key. You can also create new paragraphs by double-clicking in any open space below the previous paragraph in the document view

Text will automatically wrap onto new lines when the available width is exceeded, using hyphenation when specified, although you can force particular text strings to be kept together on the same line if required. Pressing the Shift and Return keys together forces a new line without creating a new paragraph. In the main body area text also flows onto a new page when the current page is full.

When you start typing, the text is created using the active format – the font being used, its style, color and so on. You can change the format at any time or amend the format of existing text as required. When you move the insertion point into existing text, the current format will be picked up from the text to the right of the insertion point. You can, of course, change the format before typing.

Text attributes are normally specified using the text formatting toolbar.

Note: You can specify a global default font, see [Default editor font](#) on page 73.

Changes to the formatting of text will be applied to any highlighted text, or, if none is highlighted then to any new text that is entered at the insertion point.

To show the text formatting toolbar: select **View > Toolbars > Text Formatting** from the menu.

To adjust the formatting of text: highlight the text to be changed and either use the text formatting toolbar options– or –on the **Format** menu click **Font**. This will invoke the **Font** dialog box where you can format the selected text.

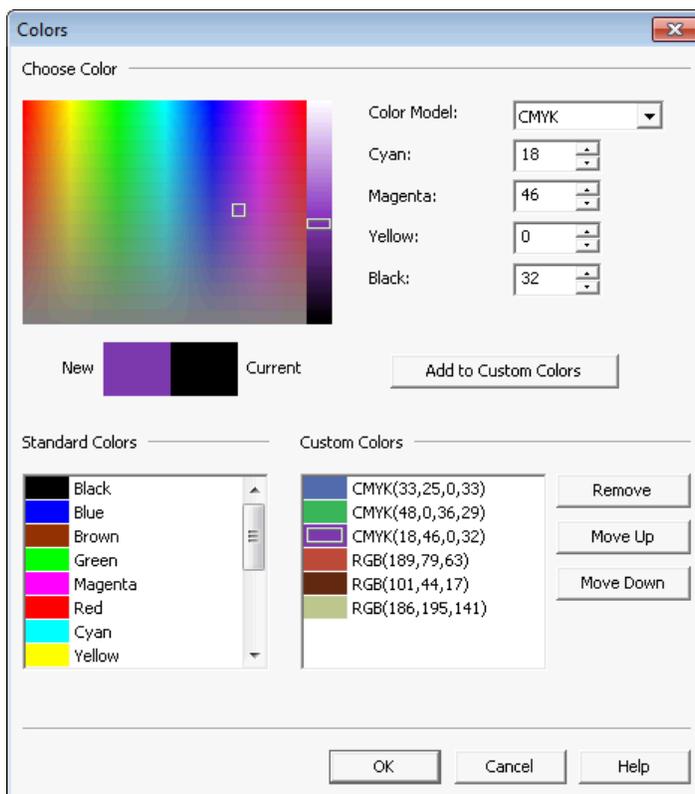
To stop a line breaking at a space: use Shift+Ctrl+Space to insert a non-breaking space character.

To keep text together: highlight the relevant text and on the **Edit** menu select **Keep together**– or –use the **Keep together** button on the text formatting toolbar. This replaces all spaces with non-breaking spaces to keep as much of the text as possible on a single line. Hyphenation will not be applied. The **keep together** attribute will remain in force and be applied to all new text until specifically cleared.

Note: If you apply Keep together to a data alias it will be applied to all the text inserted by the data alias.

Color

The **Color** dialog box enables you to select from existing default colors or you can create your own custom colors. Colors can be specified in either CMYK or RGB color models.



- Custom colors can be created by entering the specific CMYK/RGB values or by selecting the color from the spectrum.
- Standard Colors cannot be edited or deleted. Custom Colors are listed with their CMYK or RGB values and can be deleted.

Right to left text entry

The default direction of paragraph text is determined by the active Windows input language (see **Default editor font** on page 73). This is generally left to right, but for languages such as Arabic, would be right to left.

When the direction is right to left, text is entered using the keyboard in the same manner as if typing left to right text. Paragraphs can include bidirectional text and you must ensure that the direction of the paragraph is correct.

To set the direction of a paragraph to right to left: invoke the corresponding **Paragraph** dialog box. Under **Presentation** set the **alignment** as required and the **direction** to **right to left**.

-or-

use the  **Right-to-left** button on the Text formatting toolbar.

Bidirectional text entry

When working with a combination of both left-to-right and right-to-left text in a paragraph, it is important to note the following:

- You must ensure that the paragraph direction is set to right-to-left. The behavior of bidirectional text inside a left-to-right paragraph is undefined.
- Where bidirectional text can appear in a field, the language of the field must be set to either Arabic or the Unicode language.
- Paragraph labels displayed in the logic map are displayed as left-to-right regardless of the paragraph setting. This may result in inconsistencies between the layout in the editor and the layout in the logic map.
- The placement of punctuation characters within bidirectional text is dependent on the characters surrounding the punctuation, and, in certain situations, on the language setting of the punctuation characters. Punctuation in this context means all non-letter and non-digit characters in the ASCII table (including space), plus non-breaking space, currency symbols, Arabic comma, Arabic decimal separator and Arabic thousands separator.

The following rules are used to determine the placement of punctuation characters in a right-to-left paragraph:

Open bracket

- Default placement (see below) if there is no corresponding close bracket.

- Right-to-left if any of the text between the brackets is right-to-left.
- Left-to-right if all of the text between the brackets is left-to-right.

Close bracket

- Default placement if there is no corresponding open bracket.
- Same direction as the corresponding open bracket (but with the opposing glyph).

Decimal point, thousands separator, colon and non-breaking space

- Left-to-right when directly between two digits (in Latin or Arabic script).
- Default placement otherwise.

Arabic decimal and thousands separators

- Left-to-right when immediately after a number, even if no digit follows the separator.
- Default placement otherwise.

Percent sign and currency symbols

- Left-to-right when immediately after a number which follows left-to-right text.
- Default placement otherwise.

Default placement following a number

- Left-to-right when directly between two digits and the language of the punctuation is Unicode.
- Excluding space, left-to-right when directly between a digit and left-to-right text.
- Same direction as the text before the number when at the end of text in brackets.
- Same direction as the text before the number when the text after the punctuation is in the same direction.
- Right-to-left otherwise.

Default placement following text, or at the beginning of a line

- Same direction as the text after the punctuation when at the beginning of a line and the language of the punctuation is Unicode.
- Same direction as the text before the punctuation when at the end of a line and the language of the punctuation is Unicode.
- Same direction as the text before the punctuation when at the end of text in brackets.
- Same direction as the surrounding text when between text of the same direction.
- Right-to-left otherwise.

Default editor font

The editor's default font is used for text entered in publications and presentation objects, such as address blocks and text boxes. The default font is set using the **Options** dialog box and enables you to choose the font, point size, and so on.

You can use the formatting toolbar to change font attributes without affecting the default font. Changes made to the default font do not affect the text in existing publications and presentation objects.

To change the default font: select **Tools/Options**. Click the **Edit** tab and choose the required font.

Changing the input language

The input language controls how characters will be mapped to the font used within an output datastream, for example the code page to be used. Normally the intended language of text is automatically assumed from the active Windows input language when characters are entered. In some exceptional circumstances this may not be the case and you will need to set the intended language for specific text. You can use the Unicode Language option for complex scripts and languages that do not map to a standard Windows character set.

To set the language of existing text: highlight the text you want to change and select **Tools/Set language**. Select a supported language and click **OK**.

NOTE: When creating your document ensure that you have the correct input language set or you may be unable to select some language specific fonts in the editor.

Changing the language mapping

Normally the default encoding for an input language is sufficient. However, there may be situations when you may want to change that encoding, for instance to view a field from a data format that uses a different encoding scheme to the one used by the publication. This option is only used for viewing content in the Editor and is done to prevent the Windows operating system from automatically substituting fonts in a publication.

An additional option is available for changing the language mapping when text is entered in the Editor using a Windows Input Method Editor (IME). If this option is selected when characters are entered into a document using the IME, the characters displayed in the document will use the mappings that correspond to the custom code page in the Designer and not the character set associated with the language and font of the IME.

To change the language mapping:

1. Select **Tools/Options** and select the **Language/Encoding mapping** tab.
2. Select the language you want to change and click **Modify**.

3. Select either a **System** or **Custom** code page you want to use from the **Encoding** list and click **OK**.
4. The new mapping is now displayed in the **Options** dialog box. Click **OK**.

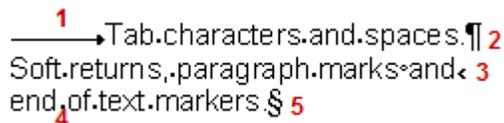
NOTE: Changing the encoding for one language, may impact a number of other languages that share the same encoding. These languages are displayed in the Affected Languages list.

To use custom code pages for keyboard entry: select **Tools/Options** and select the **Language/Encoding mapping** tab and click the **Use custom code pages for keyboard entry** option.

To reset the language mapping: select **Tools/Options** and select the **Language/Encoding mapping** tab and click **Reset All**.

Formatting marks

Formatting marks are non-printing characters that indicate controls within paragraphs such as end markers, spaces, tab characters and so on. Displaying them shows you exactly what you have typed and can help to highlight layout problems, for example, where you have typed several spaces rather than a tab character. Hiding them shows you more how the finished page might look and also removes empty paragraph objects from the logic map. Even if formatting marks are showing, they will not be in the final output.



 1 → Tab.characters.and.spaces.¶ 2

 Soft.returns,paragraph.marks and ↵ 3

 end.of.text.markers § 5

Table 1: Legend

1	tab
2	paragraph mark
3	soft return
4	space
5	end of section

To show/hide formatting marks: click the **Formatting Marks** toolbar button or select **View/Show all Paragraphs** to switch them on or off.

Hyphenation

Text in most Latin based languages can be automatically hyphenated by Designer if required. This is initially set for the entire document but can be overridden as necessary for specific paragraphs.

Note: Text will not be hyphenated if the **keep text together** option is in force. Also, text in right-to-left paragraphs can not be hyphenated.

To set hyphenation for a document in either the Publication view or the Document view double-click on the relevant document. In the **Document** dialog box select or clear the **Enable hyphenation** option.

To turn off hyphenation for a specific paragraph: select the relevant paragraph and on the **Format** menu click **Paragraph**. In the **Presentation** section of the **Paragraph** dialog box select the **Don't hyphenate** option.

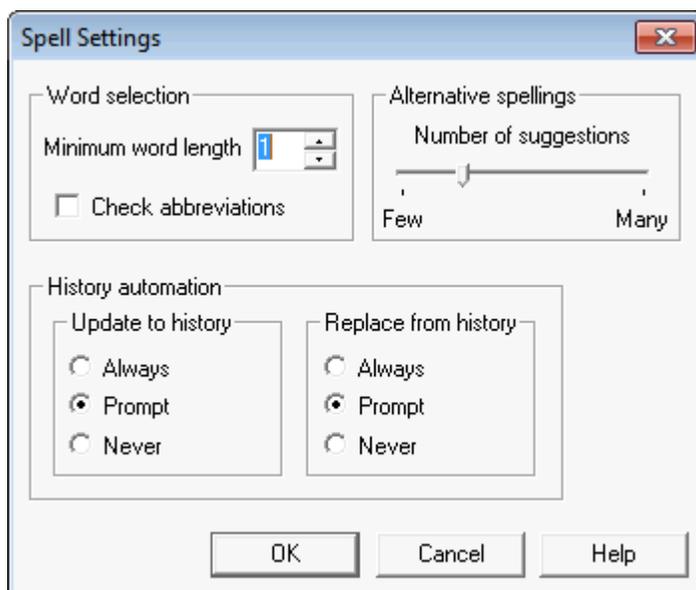
Spell checking

Spell checking is performed using the dictionary associated with the current Designer locale. Each locale has a default dictionary associated with it. You can change the locale's dictionary, but you must ensure that the language corresponds to the locale for the spell checker to be accurate.

Note: For more information on locales, see [About locales](#) on page 311.

If any text is selected in the editor, then only this will be checked, otherwise, all text will be checked. You can customize the way the spell checker works by changing the spell settings – for instance, whether to check abbreviations, whether to remember previous changes.

Note: Spelling settings are not applicable for EngageOne templates when using the preview feature. See, [Creating EngageOne templates](#) on page 349.



- You can ignore words under a certain length using the **Word selection** options.
- Alternatives are suggested for words not found in the dictionary. You can also look for more as required when checking the spelling using the **Alternative spellings** option.
- Corrections can be remembered in your local history and used for future spell checks via the **History automation** options.

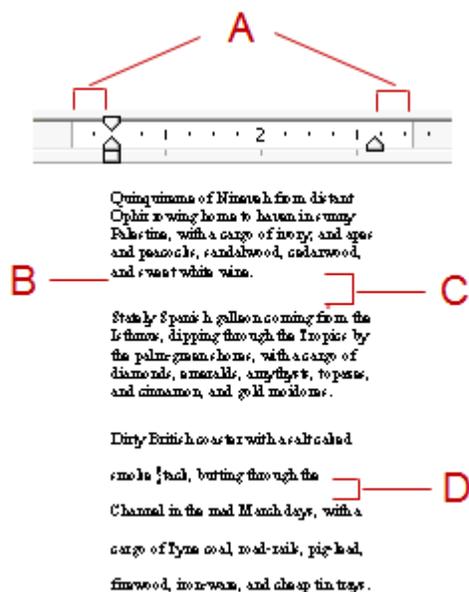
To run the spell checker: on the **Tools** menu select **Spell check**.

Note: The keyboard shortcut for the spell checker is **F7**.

To change the spell settings: on the **Tools** menu select **Spell settings** and change the settings as required.

Positioning text

Paragraphs are positioned in relation to one another and the edges of the working area; for example, the margins of the body area or the sides of a text box or table cell. You can customize how they are aligned (or justified) between the left and right edges, any indentation to be used from those edges, the line spacing within individual paragraphs, and the gap required between adjacent paragraphs. Tab stops can also be used to align text within a paragraph.



Legend:

- A - indentations
- B - alignment
- C - paragraph spacing before/after
- D - line spacing

Paragraph alignment

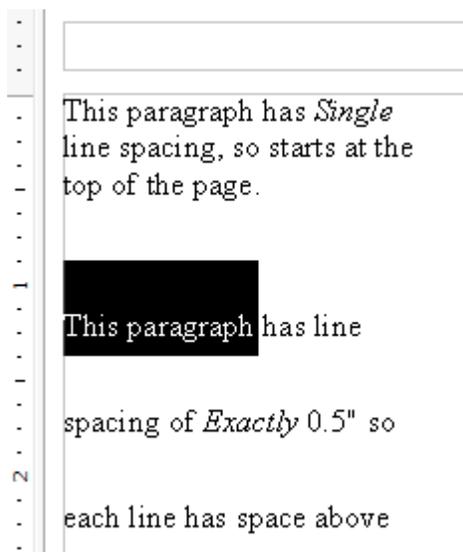
Paragraphs are aligned between their left and right indentations. Alignment is applied to the whole paragraph, so if you want a different layout for certain lines, you must create a separate paragraph for them.

For paragraphs you can specify both the amount of space to be inserted after the preceding paragraph, and the amount of space before the following paragraph. If consecutive paragraphs have conflicting 'before' and 'after' values then the greater value will be used – for example if two paragraphs have the space above set to 0.1" and the space after set to 0.2", then the paragraph spacing will be 0.2".

Line spacing determines the space between the lines within a paragraph.

Single:	represents the standard amount of line spacing for the largest font in the paragraph. 1.5 lines and Double are increments of this default.
Multiple:	as for Single , but uses the value specified as a multiple of the line, e.g. 1.2 will increase the line spacing by 20%; 0.8 will decrease it by 20%.
At least:	lines will be spaced by at least the specified amount, but will be more as required by the content of each line.
Exactly:	lines are spaced using the specified value, regardless of the size of the content of a line. Note: Only Exactly will space all the lines in the paragraph the same. All other settings can produce different line spacings within a paragraph depending on the content of each individual line, for example if there is an inline graphic.

Line spacing is measured from the bottom of the line upwards, so if you specify a value in **At least** that is much larger than the font, the additional space is above the line. This can affect the gap between paragraphs as the paragraph spacing is applied in addition to line spacing.

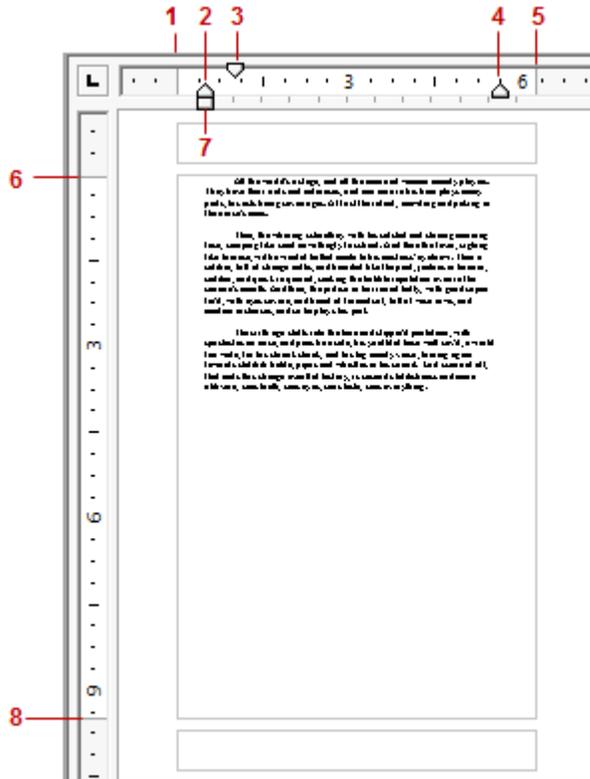


Paragraph and line spacing should be used in preference to inserting blank paragraphs between text, as it gives more control over the overall layout on the page. For instance, if a paragraph doesn't quite fit you may be able to reduce the paragraph spacing and the line spacing so that it does – say by using **Multiple** to reduce the line spacing by a small amount, e.g. 0.7 of a line.

To adjust paragraph or line spacing: on the **Format** menu click **Paragraph**. In the **Paragraph** section of the **Paragraph** dialog box set the required **Spacing** options.

Indentation

Paragraphs are, by default, positioned at the left and right edges of the working area (for the body area, these are set by the page margins). However, you can specify a gap between these edges and the paragraph by applying indentation.



- Note that Indents are measured from the margin, not from the edge of the page.

Table 2: Legend

1	left margin
2	hanging indent
3	first line indent
4	right indent

5	right margin
6	top margin
7	left indent
8	bottom margin

To indent a paragraph: move the left and right indent markers on the ruler to the required positions— or —on the **Format** menu click **Paragraph**. In the **Indentation** section of the **Paragraph** dialog box set the **Before text** and **After text** indentation.

Indentation

Before text: 0.25 in

After text: 0.5 in

Special: (none)

By: 0 in

Indenting the first line

You can indent the first line of a paragraph in relation to other lines it contains.

To indent the first line: move the first line indent marker on the ruler to the required position

– or –

on the **Format** menu click **Paragraph** and select **First Line** from the **Special** indentation drop-down list. Enter the amount to indent in the **By** field.

Indentation

Before text: 0.25 in

After text: 0.5 in

Special: First line

By: 0.25 in

Hanging indent

A hanging indent is where all lines except the first one are indented. The hanging indent is measured from the left, or first line indent.

Indentation

Before text: 0.25 in

After text: 0.375 in

Special: Hanging

By: 0.5 in

To create a hanging indent: move the hanging indent marker on the ruler to the required position

– or –

on the **Format** menu click **Paragraph**. Select **Hanging** from the **Special** indentation drop-down list. Enter the amount to indent in the **By** field.

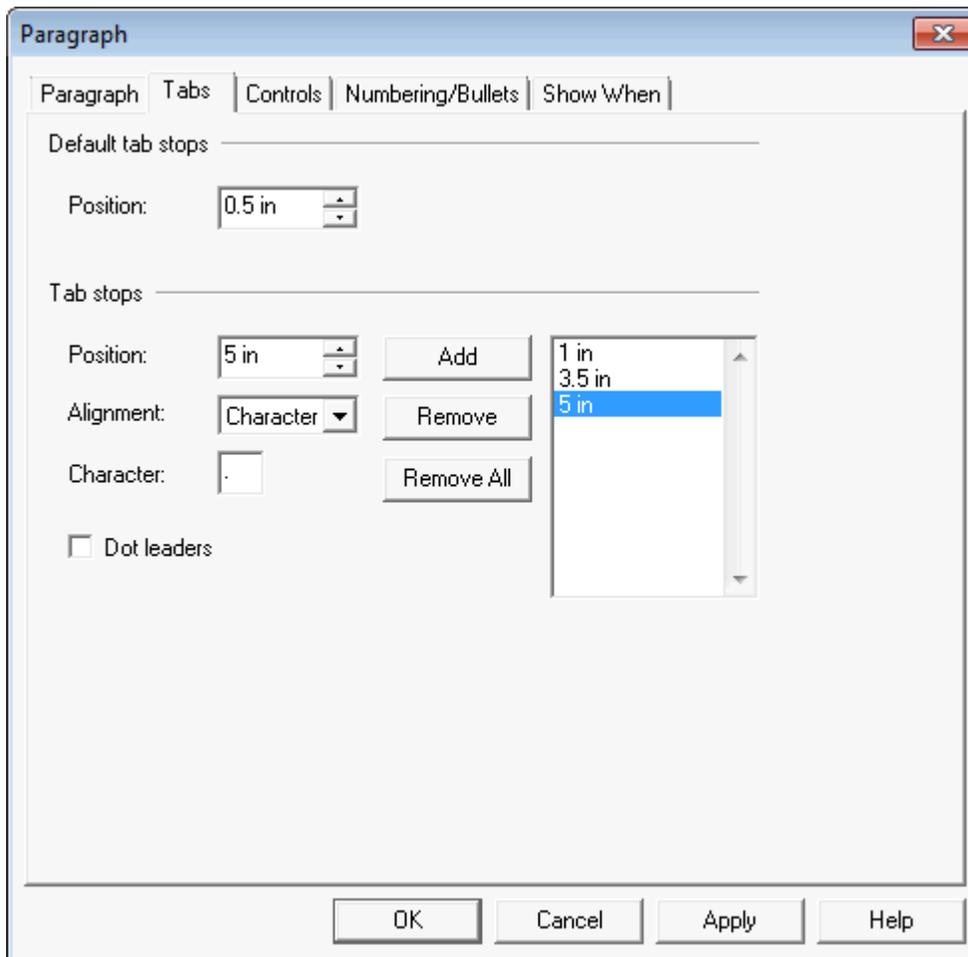
Tab stops

Any number of tab stops can be specified in a paragraph. They can be left, right, centered, or aligned on a character. A character tab will align on the first occurrence of the specified character. If this is not in the text, then a tab the same as the paragraph justification will be used.

If there are no tab stops specifically set in a paragraph, then, by default, tab stops will be set at regular intervals on the ruler. These will be the same as the paragraph alignment, for example right tabs in a right aligned paragraph.

<u>Tab selector</u>	<u>Tab stop</u>
L	left
⌵	centered
J	right
⌵	character

To insert a tab stop: click the tab selector on the ruler until the required tab stop is shown. Then click on the ruler at the position where you want to place the tab stop. The tab stop will be inserted on the ruler– or –on the **Format** menu click **Tabs**. The **Tabs** page of the **Paragraph** dialog box will be displayed. Specify a tab stop and click **Add**.



- Set default tab stops for the document using the **Default tab stops/Position** control.
- Set the position and alignment of a tab stop to be added or changed using the **Tab stops** options.
- Use the **Add** button to create a new tab stop.
- Use the **Remove** button to delete all tab or the selected tab stop.
- Enter a **Character** to align on for Character alignment, e.g. a decimal point.
- Select **Dot leaders** to add leading dots to a tab stop.

To move a tab stop: either select the tab stop on the ruler and drag it to its new position

– or –

on the **Format** menu click **Tabs**, select the relevant tab stop from the list and enter the new position under **Tab Stops/ Position**.

To modify an existing tab stop: on the **Format** menu click **Tabs**, select the tab stop from the list and change its **Alignment**, **Character** and **Dot leaders** as required.

To remove a tab stop: either select the tab stop on the ruler and drag it off the ruler

– or –

on the **Format** menu click **Tabs**, select the tab stop from the list and then click **Remove**, or click **Remove All** to delete all tab stops.

To remove all tab stops: on the **Format** menu click **Tabs** and then click **Remove All**.

To set default tab stops: on the **Format** menu click **Tabs**. Enter the interval between tab stops under **Default Tab Stops**.

Dot leaders

Dot leaders can be used between tab stops to help readability. When a leader is applied to a tab, a row of dots is inserted between the end of preceding text and the current tab stop. This is typical of table of contents and similar styles.

To insert a dot leader: on the **Format** menu click **Tabs**, select the tab stop from the list and tick **Dot leaders**.

Rotating text

Text can be rotated only when it is in a text box, see [Rotating text](#) on page 103 for details.

Bullets and numbering

Bullets or numbering can be applied to paragraphs if you want to emphasize them or to create lists. When they are applied the paragraph indentation will automatically be changed to suit. Numbering can be reset at any time back to '1' and you can add a character to follow the number, for example, a dot. You can specify a different font (or font attribute) for the bullet or numbering character. By default, bullets use the '.' character, but you can change this, if required, to a different character.

To set bullets or numbering: use the **Bullet** or **Numbering** toolbar buttons

-or-

on the **Format** menu click **Paragraph** and on the **Numbering/Bullets** page select the style that you want.

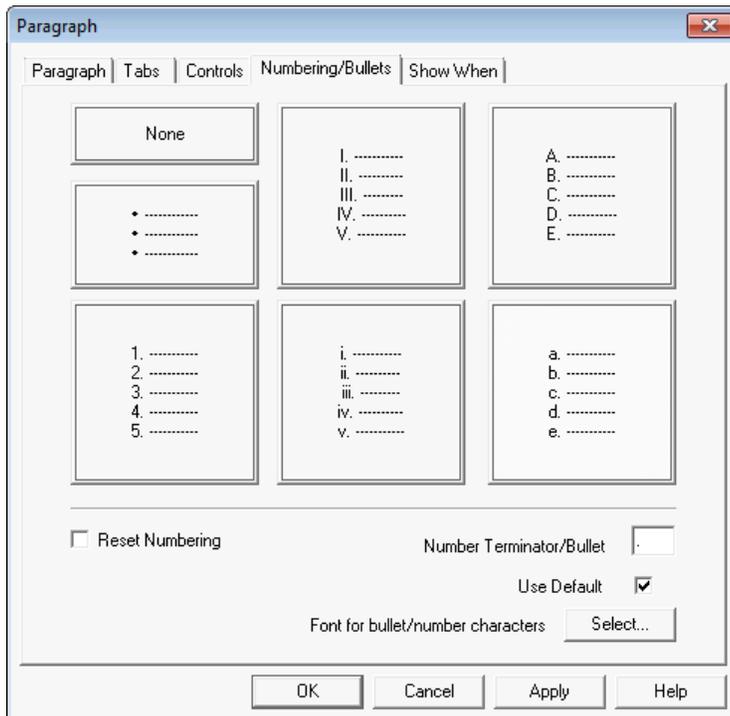
A hanging indent is automatically applied with a tab between the bullet/number and the text. Drag the tab marker on the horizontal ruler to change the indent, using the line as a guide.

To format bullets and numbering: on the **Format** menu click **Paragraph** and on the **Numbering/Bullets** page select formatting as required.

To turn off bullets or numbering: de-select the appropriate toolbar button.

-or-

from the **Format** menu click **Paragraph** and on the **Numbering/Bullets** page select **None**.

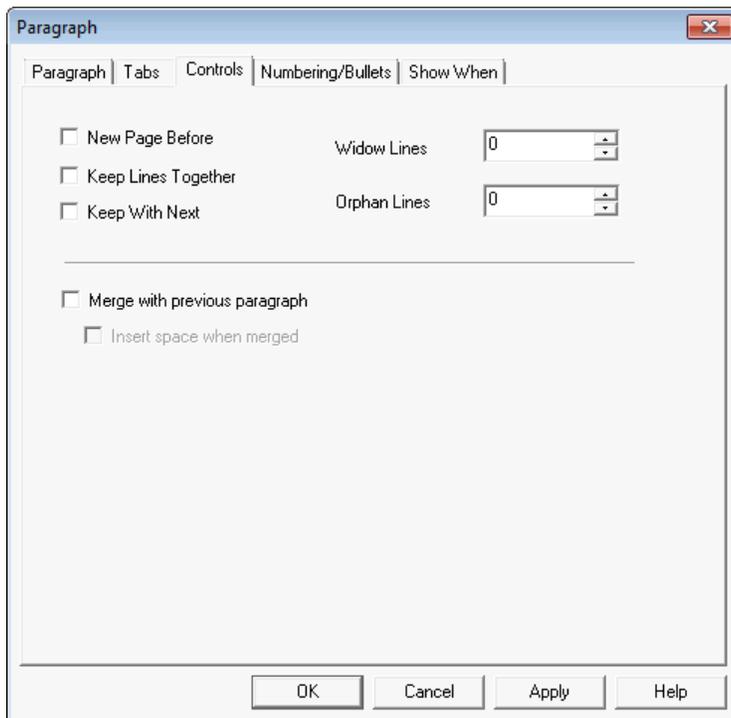


- Use the **Reset Numbering** option to set the numbering to 1.
- Use the **Number Terminator/Bullet** option to enter a character to use as a bullet or to follow the number, or use the **Default** character.
- Use the **Font for bullet ...** option to select a different style for the bullet or numbering.

Pagination

There are various options you can use that affect how a paragraph is positioned and the text flow. You can force the paragraph to start at the top of a page, keep a minimum number of lines together if a page break splits a paragraph (widows and orphans), or never allow a page break to split a paragraph so that the entire paragraph is kept together.

You can also specify that a paragraph is always kept with the following paragraph.



- Use the **New Page Before** option to start the paragraph at the top of the next page.
- Use the **Keep Lines Together** option to prevent the paragraph being split across a page break.
- Use the **Keep with Next** option to ensure that the next paragraph is always kept with the current one.

To control pagination: on the **Format** menu click **Paragraph** and on the **Controls** tab select the options required.

Widows and orphans

Widows and orphans occur when a paragraph is split across a page break, with the bulk of the paragraph on one page and the remainder on the other.

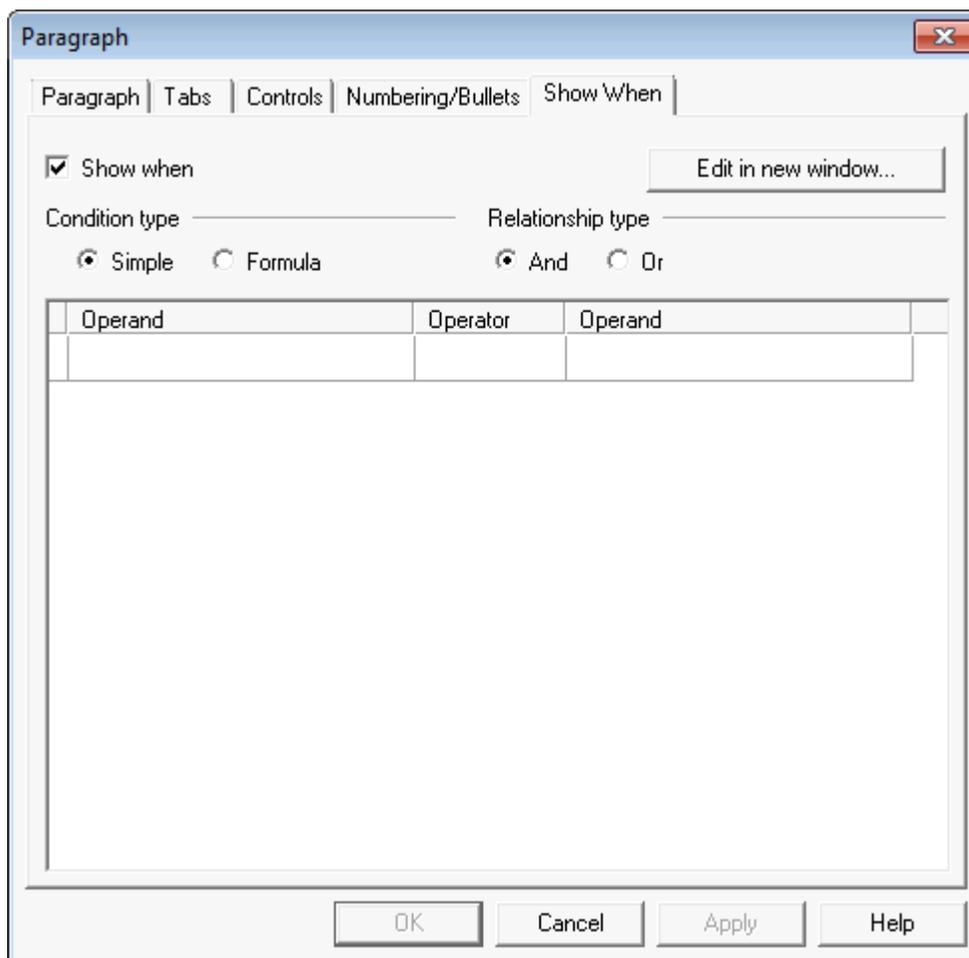
A widow is a few lines at the beginning of a paragraph that are separated from the rest of the paragraph by a page break. An orphan is a few lines at the end of a paragraph that are separated from the rest of the paragraph by a page break.

Specify the number of lines that you want kept together at the bottom or top of a page. Orphan control set to '2' ensures that at least two lines are kept together.

To specify widows and orphans: on the **Format** menu option click **Paragraph** and on the **Controls** tab enter the number of **Widow Lines** or **Orphan Lines** required.

Conditional paragraphs

Paragraphs can be made conditional so that they appear when certain criteria are met, for instance, you may want to show a paragraph only if a customer has a certain type of account. You will typically want to use values from data aliases and variables that have been made available to the publication as part of the condition.



- Select the **Show When** option to make the paragraph conditional.
- Use a **Condition type** option to choose the type of condition.
- Use a **Relationship type** option when more than one condition is defined.
- Enter your condition in the **Operand-Operator-Operand** area of the page. Refer for [Creating conditional logic](#) on page 217 for further details.

Note: If you make a paragraph conditional it will not be shown in the editor if the specified condition evaluates to false. To change the settings you will need to use the logic map to work with the relevant paragraph object. See [Using the logic map](#) on page 150 for more information.

To make a paragraph conditional: on the **Format** menu option click **Paragraph**. On the **Show When** tab of the **Paragraph** dialog box check the **Show When** option and enter the condition as required.

Merging paragraphs

You can optionally join content from two adjacent paragraphs to form a single paragraph in your output. This would typically be used to join text from conditional logic or from within Active Content objects. Paragraphs are merged by selecting the required paragraph object in the logic map.

- Merge with previous paragraph
- Insert space when merged

When merging paragraphs you need to be aware of the following:

- Joined paragraphs continue to appear as separate objects in the logic map.
- Paragraphs are considered to be 'adjacent' if they are not separated by another content object in the sequence of logic map objects. For example, two paragraphs separated by one or more condition objects can be merged but not where separated by a graphic object.
- When paragraphs are merged, paragraph attributes are gathered solely from the first paragraph in the sequence. Character attributes are unaffected by a merge.

To specify the merge option: open the paragraph object in the logic map and from the **Controls** tab specify the required merge options.

Importing paragraph content

You can include RTF content created outside Designer in your design layout. You can import the contents of an RTF file directly into your document, and also copy and paste the RTF – see [Using the clipboard](#) on page 60 for details.

The text, fonts, colors, tabs and justification used in the RTF are included when it is imported. Windows supported images, either embedded or referenced, are also included. Page layout information, such as page margins and page size, is not included.

Note: RTF content can also be inserted into documents by using an RTF field. For more information, see [RTF fields](#) on page 176.

Text in RTF format may be slightly different depending on which application it was created by. Should you experience inconsistencies, open the RTF file in Windows WordPad and save it (as an RTF file). You can then import from this file.

To import RTF content: from the **Insert** menu select **Presentation Objects/RTF**. Browse to the RTF file and click **Open**.

Symbols

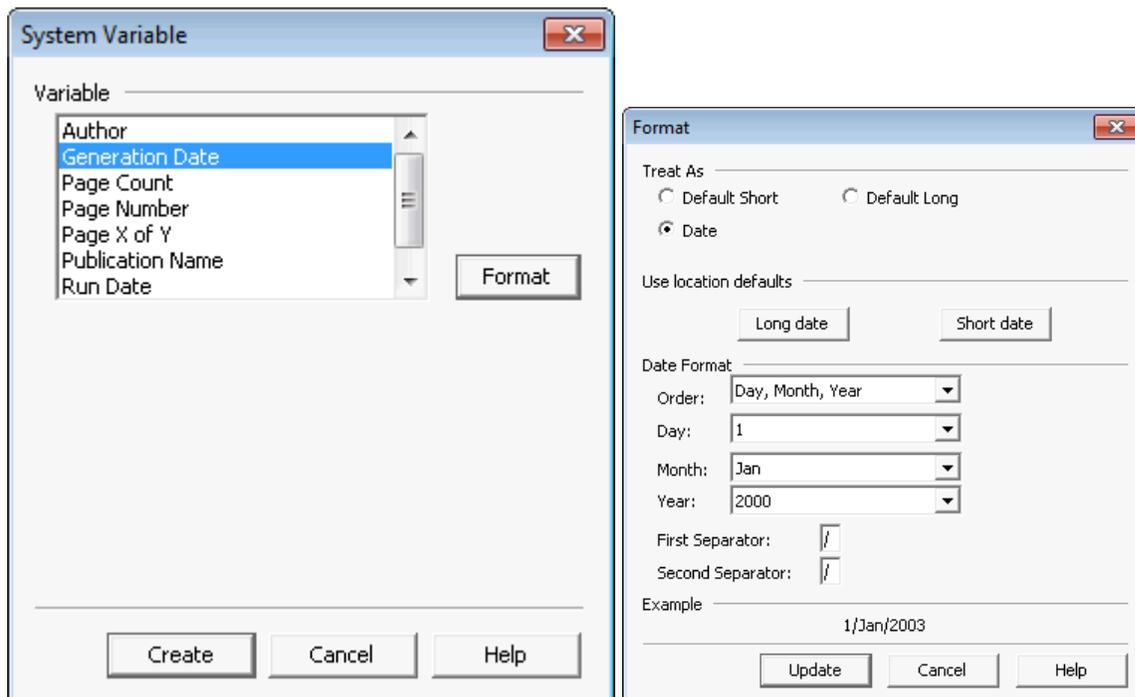
Although most commonly used characters and symbols can be entered from the keyboard, some require several key presses, for example, to type in the registered mark ® in most common fonts, you would have to type Alt+0174. A menu option is provided which accesses the Windows Character Map, where you can choose the character or symbol you want to enter.

To insert a symbol: on the **Insert** menu click **Symbol** or click the "copyright" toolbar button. In the **Character Map** select the required font and choose the characters you want. Copy them and then paste them into your document.

System variables

A system variable allows you to insert dynamic information that will be resolved at production run time, for example the system date or the current page number being processed. Some options generation date or page numbers. It can be formatted in various ways, depending on the information being shown, for instance, the page number can be Arabic numbers (1,2,3...), Roman numerals (i,ii,iii...), or alpha (a,b,c).

To insert a system variable: on the **Insert** menu click **Field > System Variable** or click the toolbar button. Select the required variable from the **Variable** list and specify any options that are required. You can format some variables by clicking the **Format** button – see **Formatting values** on page 173 for more information.



System Variables in paragraphs

Author	The user that created the publication being processed.
--------	--

Generation date	The system date when the HIP file was created for the production job by a publish task.
-----------------	---

Note: Only the date of the publication HIP is used by Generate.

System Variables in paragraphs

Page number	The current page number. Select whether you want the page number within the entire job or within the current publication or document , the format that you want the numbering to be in and whether or not the page number is to be shown on the first page.
Page count	The total number of pages in the current document. If Page count is required for multiple documents then you must use the Continue page count option in the Document dialog box. To use it for a complete publication, this option must be set for all documents except the first one.
Page X of Y	The current page number out of the total number of pages in the document being processed, for instance: 'Page 2 of 5'. Page X of Y is only displayed correctly when used in paragraphs that are left, right or center justified. The formatting of this option can be modified via Locales in the navigator. If Page X of Y is required for multiple documents then you must use the Continue page count option in the Document dialog box, see Using multiple documents on page 160 for further information. To use it for a complete publication, this option must be set for all documents except the first one.
Publication name	The name of the publication being processed.
Run date/time	<p>The system date or time when Generate began processing the production job.</p> <p>Note: An extended range of system variables are available for use as values within other objects. See System variables on page 257 for details.</p>

URL links

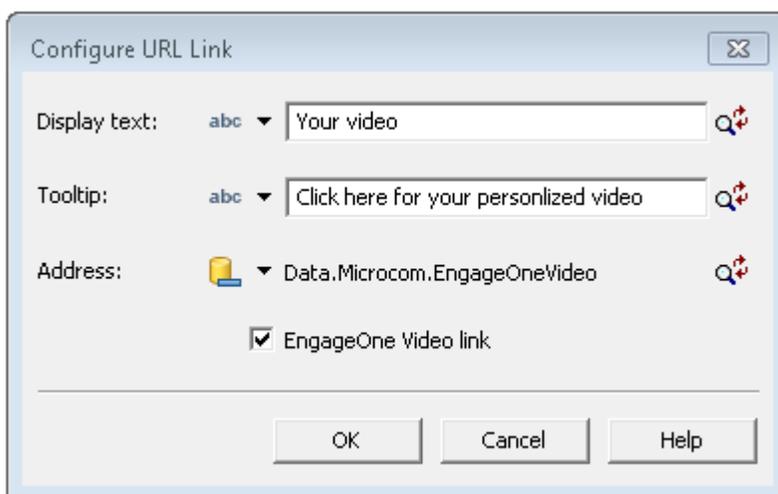
URL links can be included in EngageOne applications that are intended for web enabled output. If they are used in non–web enabled output then the display text or graphic is still shown, but the link will be inactive.

Strings, variables and data aliases can be used for the URL link values. For information on data aliases, see [Using data aliases within text](#) on page 117 and for details on variables, see [Variables](#) on page 178.

Note the following when using URL links:

- No check is made that the URL link references a valid address.
- The use of variable indexes is not supported for the Display text, Tooltip or address options.

To create a URL link: in the **Insert** menu click Field/URL Link – or – click the Field/URL Link toolbar button. In the **Configure URL Link** dialog box specify the options for the link.



- Use the **Display Text** option to define the text to be displayed on the page.
- Use the **Tooltip** option to display information when hovering over the link.
- Use the **Address** option to define the URL address of the web site.
- Use the **EngageOne Video link** option to associate the URL to a Personalized URL (PURL). The PURL is used to link to a EngageOne Video project.

Note: This option is only available when a data alias is associated to the **Address** field. Refer to [Working with EngageOne Video design objects](#) on page 347 for further information.

To create a URL link for an image: select the image and from the **Format** menu click on **Shape**. In the **Image** tab, select **Clickable image** and specify the options for the link. Use the **Alternative text** option to specify the tooltip. This text is also shown if the image cannot be displayed.

To edit a URL link: highlight or click in the URL and in the **Format** menu click **URL Link Field**. Change the options for the link as required in the **Configure URL Link** dialog box.

- Tooltips are presented when hovering over the link.
- Press **Ctrl+left-click** to activate link.

To test a URL link: hover over the link and hold down Ctrl and click the left mouse button. Your internet browser will start up with the URL in the address.



6 - Working with graphics

In a document design you can reference image files created elsewhere or you can draw simple graphics using the editor's drawing tools. These are known collectively as shapes. Image files imported from your output device environment are treated the same as locally generated images.

In this section

Using graphics.....	96
Anchors and positioning.....	96
Drawing shapes.....	98
Using images.....	99
Text boxes.....	101
Selecting and adjusting graphics.....	103
Layout.....	107
Order of graphics.....	108



Using graphics

All local images used by a publication are automatically converted to a format suitable for your target output device at publishing time. These are stored as part of the HIP file used by Generate.

Text boxes are treated as graphics. The rectangle shape of a text box can be filled and given a border as for other graphics and it has the same controls for flowing into paragraph text. Unlike other graphics however, a text box is intended to contain other presentation objects.

Individual graphics can be rotated independently within a document layout. You can use a text box to rotate a group of presentation objects including text.

You can control how graphics interact with paragraphs by specifying how text flows in relation to individual graphics and how much margin space is required where text flows around.

You are able to specify the order in which overlapping graphics are presented and can use the tools provided to align and space graphics in relation to one another.

Anchors and positioning

A graphic is always positioned in relation either to a paragraph or to a page; this is known as its anchor. There are three different types of anchor: inline, anchored, and fixed.

When a graphic is inserted it is initially anchored according to the type of graphic. When you move the cursor over the graphic the type of anchor used is displayed. You can change the anchors of individual graphics once they have been inserted.

The following anchor types are available:

In-line

Positioned as a character within paragraph text. Paragraph line spacing will be increased to accommodate the graphic, unless it has been set to an 'exact' value. In-line is the default for images.

Anchored	Positioned relative within the main document flow. In the vertical plane the object is offset in relation to the previous paragraph or the top of the current container if no paragraph exists. In the horizontal plane the offset is always in relation to the left side of the container (page, text box, and so on). Anchored is the default for shapes, such as lines, rectangles, ovals and text boxes.
Fixed	Positioned at a specified offset in relation to the top left corner of the page where it is inserted or multiple pages if used on the background. Fixed is the default setting, and only anchor option, for images inserted on the background.

Note: Not all anchoring methods are applicable to all types of graphic.

To change the anchor type: select the graphic on the page

– or –

highlight the graphic in the logic map. From the **Format** menu select **Shape**. In the **Layout** tab under **Flow** select the required anchor type and click **OK**.

Coordinates and dynamic positioning

Graphics using anchored or fixed methods are typically positioned by dragging the object to the required position in the page layout or by specifying coordinates as constant values using the shape properties, see [Moving and resizing](#) on page 104 for details.

Note: When positioning graphics using variable data it is vital to carry out the appropriate level of testing to ensure correct placement in the production environment. You can specify how errors are handled when a shape is placed outside the page area. See [Publishing and deployment](#) on page 592.

You can also use any object or function that supplies a value to specify coordinates. Where such values are provided by variable data such as data fields it follows that the graphic can have dynamic offsets that differ from document to document.

To position using coordinates: select the shape in the editor and on the **Format** menu click **Shape**. In the **Layout** tab select the required unit of measurement and select whether to measure from the edge of the **page** or the current **column**. Choose the appropriate value object using the value option button for the X and Y offsets. See [Working with values](#) on page 170

Drawing shapes

You can create simple graphics directly in the editor by using the shape drawing tools. These include lines, rectangles, and ovals. You can customize the line thickness, shading, fill color and so on. used for such shapes and they can, except for lines, be rotated to any angle, see [Rotation](#) on page 106 When you create a new shape the attributes of the previously created shape will be used. You can adjust these by editing the shape's attributes once it has been created.

Attention: Restrictions inherent in some output datastreams mean it may be necessary to downgrade or omit some types of shapes depending on your target output device.

Graphics are, by default, anchored to a paragraph. However you can change this once the graphic has been drawn, see [Anchors and positioning](#) on page 96.



Line



Rectangle



Rounded Rectangle

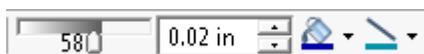


Oval

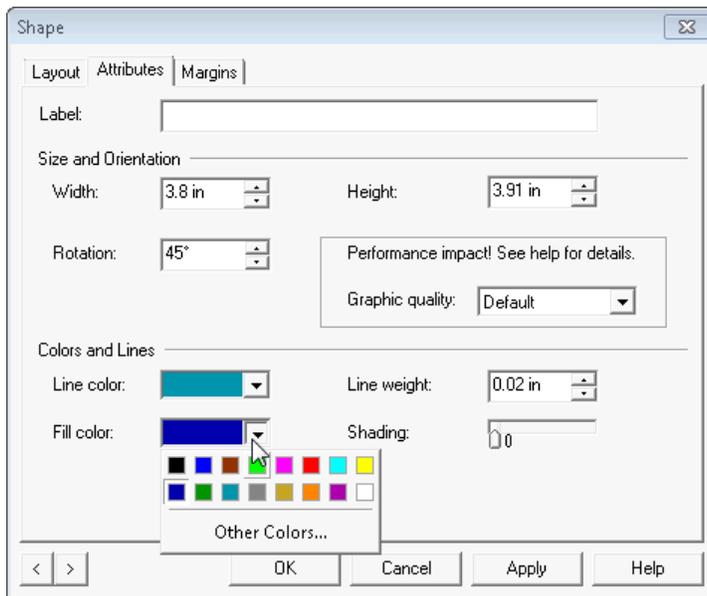
To draw a shape: click in the paragraph that you want the graphic anchored to and on the **Insert > Presentation Objects** menu select the required shape or click the appropriate toolbar icon. Click where you want to start drawing and drag out the shape to the required size. Holding down SHIFT while drawing will create a square, a circle, or a horizontal or vertical line.

To change shape attributes: select the shape in the editor and on the **Format** menu click **Shape** or

right-click on the object in the logic map and select **Properties**. Use the **Attributes** tab to set the required color, size, shading, rotation, graphic quality and line settings. Use the arrows at the bottom left of the dialog box to move to the previous or next shape in your design.



- Use the slider in the above image to alter the shading
- Select the line-weight, fill color and line color as required



To change the corners of a rounded rectangle: select the rectangle and drag the internal handle. Both the amount and the shape of the curve will change.

Using images

You are able to reference images that are already stored within the EngageOne Designer repository and that are in the same project as the current document, publication, or Active Content.

Note: In some installations these facilities may be restricted by the rights specified in your user profile.

If the resolution is not already specified in an image, a default resolution of 96 dpi is used when it is inserted into a document. Images are, by default, anchored inline within a paragraph. However you can change this once the image has been inserted, see [Anchors and positioning](#) on page 96. Images can also be rotated to any angle.

Clickable images

An image can also be used as a URL link (for PDF, HTML and eHTML datastreams only) by selecting the **Clickable image** option. See [URL links](#) on page 93 for details.

Note: The use of variable indexes is not supported for the Display text, Tooltip or address options.

Note: Use the **EngageOne Video link** option to associate the image's URL to a Personalized URL (PURL). The PURL is used to link to a EngageOne Video project. This option is only

available when a data alias is associated to the **Address** field. Refer to [Working with EngageOne Video design objects](#) on page 347 for further information.

If you are working with more than one language you can change the image according to the active locale by selecting the **Multiple languages** option. See [Localization](#) on page 310 for details.

To insert an image:

1. Click in the paragraph where the image is to be inserted.
2. From the **Insert > Presentation Objects** menu select **Image** -or- click the **Image** button in the Objects toolbar.
3. Use the **Open** dialog box to select the image you want to insert.

To change the image used: select it and from the Format menu choose Shape. Select the Image tab in the Image dialog box and enter or browse for a different image.

To rotate an image: select the image in the editor and on the **Format** menu click **Shape**. Use the **Rotation** option on the **Attributes** tab to specify the number of degrees that the image is to be rotated clockwise. The image will be rotated on its central point. See [Drawing shapes](#) on page 98 and [Rotation](#) on page 106 for more information.

Images can also be inserted using a keyed object. This allows different images to be called depending on the value of a key, for example a data alias – see [Keyed objects](#) on page 267 for details.

Note: it is important to be aware of the following when working with images:

- Inconsistent results may occur such as loss of quality or truncation when using SVG images in your template design. It is recommended that such images are converted to JPEG format at the appropriate resolution and then imported into the repository for use in your document design.
- The use of transparent or layered images is not supported.

Overlays

Overlays, the term used by IBM for an electronic form, can be inserted into your document in the same way as images. However you should bear in mind that they can only be placed in a fixed position on the background layer of your document – see [Background](#) on page 113 for more information.

To insert an Overlay: In the Page Setup view click in the background where the overlay is to be inserted. From the **Insert > Presentation Objects** menu select Overlay. Use the Open dialog box to select the overlay you want to insert.

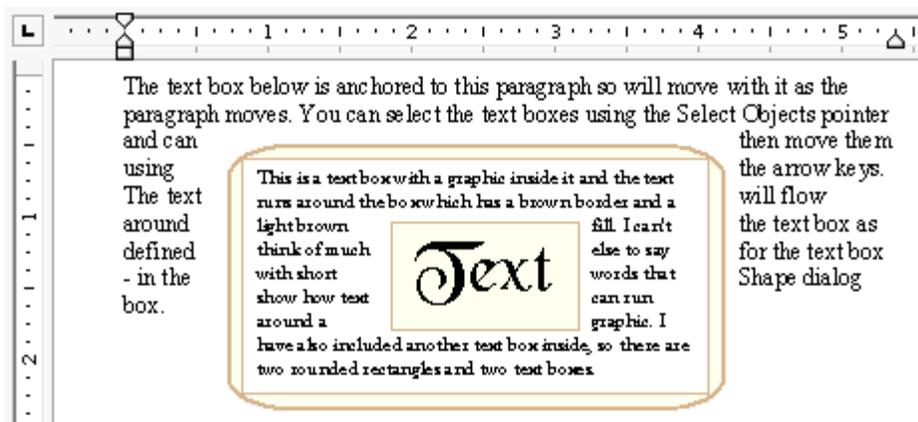
Text boxes

A text box acts as a container for specific page content and can contain both paragraph text and graphics. You can use a text box simply to highlight particular content or as a reserved area which other text can flow around.

Text boxes are rectangular shapes and are treated like graphics for the purposes of positioning and customizing. They can be inserted anywhere in a document design, for instance, in the body area, on the background, in headers and footers and even inside other text boxes.

The dimensions of a text box are as drawn. If content does not fit you must resize the text box appropriately.

Note: It is important to note, however, that variable data can cause the text box to overflow in production. Any content that does overflow the available space will be lost.



The position of a text box (anchored to a paragraph) within the document flow may itself be affected by variable content that precedes it. For instance, if you place a text box after a transaction table, its vertical start position will be influenced by the number of transactions in the table. If there is insufficient space to fit the text box at the bottom of the current flow area, typically a page, it will by default be moved to the following page.

If you want an anchored text box to appear only when there is sufficient space in the current flow area, then set the **Do not place where an overflow would occur** option. This causes the text box to be discarded when it does not fit.

Text boxes are, by default, anchored to a paragraph. This can be changed once the text box has been drawn; see [Anchors and positioning](#) on page 96.

Note: You can also rotate text within a text box at 90° intervals without rotating the text box itself. For more information, see [Rotating text](#) on page 103 for details.

If a header or footer contains a text box then the text box and its contents can be rotated within the header or footer.

Creating a text box

To create a text box:

1. Click in the paragraph where you want the text box anchored and from the **Insert > Presentation Objects** menu select Text Box -or- use the **Text Box** button in the Objects toolbar.

2. Create the box.

If you are using a paragraph or fixed anchor click in the page layout where you want to start drawing and drag out the shape to the required size.

If you are using an inline anchor, a default sized shape will be created within the paragraph ready for you to adjust.

3. Add content to the text box.

For paragraph text click inside the box and start typing. Create graphics, including other text boxes, inside the box as normal.

4. To change text box line and fill settings select the text box in the editor and on the **Format** menu click on **Shape**. Use the **Attributes** tab to customize the settings. See [Drawing shapes](#) on page 98 for more information.

Changing the look of text boxes dynamically

The look of a text box, lines, fill, and so on, can be changed during production. For instance, you might want a different color for a text box that contains a negative value. This involves the use of the Action option and would generally be in some kind of conditional logic. When the condition is met, the change is applied, otherwise the original settings are used.

1. In the logic map mark the insertion point in the text box to be changed and from the **Insert** menu select **Control Objects > Action**.
2. In the **Action** dialog box select one of the following from the list and change the setting as required. **Change border color, Change fill color, Change shade percent, Change side thickness.**

Rotating text boxes

You can use a text box to rotate content to any angle. The dimensions and shape of the text box will be maintained but the bounding box containing the rotated objects will be automatically resized as required. You may have to adjust the surrounding objects to fit.

To rotate a text box: select the box in the editor and on the **Format** menu choose **Shape**. Use the **Rotation** option on the **Attributes** tab to specify the number of degrees that the image is to be rotated clockwise. The text box will be rotated on its central point. See [Drawing shapes](#) on page 98 and [Rotation](#) on page 106 for more information.

Note: URLs inside rotated text boxes will be converted to text and lose their hyperlink. However the link will still be available while editing the text box.

Rotating text

Text that is inside a text box can be rotated. It is rotated 90° clockwise, and can be rotated more than once until it is in the required plane. Text in headers and footers can also be rotated, see [Headers and footers](#) on page 185

Note: Any image or graphic that is also in the text box will not be rotated. You will have to provide it in the required rotation.

The text box itself will not be rotated, although it will adjust to cater for extra lines. However, you may need to resize it to the required dimensions once the text has been rotated.

To rotate text within a text box: place the pointer inside the text box and on the **Format** menu click on **Rotate** – or use the Rotate button in the Text Formatting toolbar.

Selecting and adjusting graphics

In order to adjust a graphic, you must first select it. When it is selected, handles are displayed on the corners and sides of a bounding box. The pointer will indicate how the graphic is anchored.

You can also select an existing graphic from the dialog box, cycling through them in the order they were created. This means that when you have changed the attributes for one graphic, you can move on to another one and change it without closing the dialog.

Note: White handles indicate that the graphic is open for input, meaning a text box is ready for you to type in. Black handles indicate that the graphic is just selected, and can be moved, or if appropriate, resized.

To select a graphic: click anywhere in the graphic or on its border.

To select multiple graphics: select the first graphic and then hold down SHIFT whilst you click the others

– or –

use the **Select Objects** toolbar button and drag across all the graphics, ensuring that you enclose them all completely.

To view the next or previous graphic: on the relevant shape formatting dialog box use the next or previous buttons.

To select all graphics: click the **Select Objects** toolbar button to turn the select mode on and on the **Edit** menu click **Select All**.

To deselect a graphic when many are selected: hold down SHIFT and click the graphic. The handles will disappear from the graphic when it is deselected.

Moving and resizing

Graphics other than those using an in-line anchor can be moved either directly within the editor window, or with precision using the settings in the dialog box. Graphics can be resized in the same way.

You can also align graphics to each other, or move them so that the space between them is evenly distributed.

Note: If you want to move graphics in relation to text, rather than other graphics, then you must use the text runaround properties. See [Working with graphics](#) on page 95.

To move a graphic: hover over the graphic and when the movement pointer is displayed, drag to its new position– or –select it in the editor and use the arrow keys to nudge it– or –select it in the editor and on the **Format** menu click **Shape**. Under the **Layout** tab enter the new position for the top left corner of the graphic. You can also use any object or function that supplies a value to specify co-ordinates to position the graphic, see [Coordinates and dynamic positioning](#) on page 97 for details.

To resize a shape: select the graphic and drag any of the handles to the new size.

-or-

on the **Format** menu click **Shape**. Under the **Attributes** tab enter the new **Width** or **Height** amounts as appropriate. If you enter a value of zero for the width or height of a line, then the line will become horizontal (width=0) or vertical (height=0).

Note: You can not resize referenced images in the editor. If you need to do so, first resize the original using a suitable graphics program or utility and then reference the new resource in the editor.

Aligning

If you have several graphics on a page, you can line them up with each other. They can be aligned on their edges or so that the middle of each graphic lines up.

All graphics that are selected will be aligned with each other. They are aligned inside an imaginary box drawn around all the selected graphics, for instance, aligning graphics on the left will place them against the left of this box; aligning them in the center will align the centers of each graphic in the center of the box. It is important to ensure that the relevant graphic is in the correct position before alignment, for example, the right most graphic when aligning right.

To align graphics: select the graphics and on the **Format** menu click **Align or Distribute** and select one of the following alignment options:

Table 3: Alignment

Left	align with the leftmost graphic
Centered	between the leftmost and the rightmost graphics
Right	align with the rightmost graphic
Top	align with the topmost graphic
Middle	between the topmost and the bottommost graphics
Bottom	align with the bottommost graphic

Distributing

Graphics can be distributed so that the space between each graphic is the same. The graphics on the outside stay in the same position, and all those in between are moved so that the space between the borders of each graphic is evenly distributed. This is only available when there are three or more graphics selected.

To distribute graphics: select the graphics, and on the **Format** menu click **Align or Distribute** and select one of the following distribution options:

- Distribute horizontally – left to right
- Distribute vertically – top to bottom

Rotation

Graphics can be rotated to any angle in one degree increments. This includes images, text boxes, address blocks, barcodes and shapes – apart from lines. The bounding box containing the rotated object will be automatically resized as necessary and you may need to adjust the surrounding objects to fit.

Note: Simple text rotation can be performed within a text box. For more information, see [Rotating text](#) on page 103 . This does not have performance or quality issues.

You can also set the quality of the rotated graphic as this will affect both the time taken to generate and the size of the output file, which can be considerably increased when using rotated graphics.

To rotate a graphic: select the graphic object in the editor and on the **Format** menu click **Shape** or right-click on the object and select **Format Shape**. Use the **Rotation** option on the **Attributes** tab to specify the number of degrees that the image is to be rotated clockwise. The object will be rotated on its central point.

To edit the contents of a rotated graphic: double-click on the object in the editor or right-click on the object and select **Edit text**. The shape will be temporarily unrotated for editing.

Note: You can only resize a rotated object while it is being edited.

To set the quality for a rotated graphic: in the **Attributes** tab of the **Shape** dialog box set the **graphic quality**:

- **Default** uses the resolution that is specified for the intended output device – see [Output, media and resources](#) on page 520.
- **Low** gives the best speed to generate but the lowest image quality.
- **High** will take much longer to generate but will be a reasonable quality of image.
- **Fine** is only available for barcodes and produces a very high quality image that is suitable for scanning.

Note: Low, medium and high settings will override any other resolution settings (in the output device settings) for this graphic. The following section shows more limitations when using advanced graphics.

Limitations of advanced graphics

Some advanced graphics features such as rotation need to rasterize the graphic (convert to pixels) to generate the desired output. This rasterization process, and hence the graphics it generates, has the following restrictions:

- Only these drivers are supported – **AFP, PCL, Postscript, PDF and eHTML**.

- Complex scripts are not supported.
- The time taken to generate the output and the size of the output file can be considerably increased.

Layout

Depending on the type of anchor, you can choose how adjacent text should flow around the graphic or whether the text should appear behind or in front of it, and its alignment across the page.

Graphics that are anchored inline are treated as just another character in the text and the line spacing will normally be increased automatically to accommodate them.

To select the runaround properties for a graphic: select the graphic and on the **Format** menu click on **Shape**. On the **Layout** tab click the required **Wrapping style**:

Square	Text will flow evenly around the graphic.
Top and Bottom	Text will flow down to the graphic and continue after it.
Left	Text will only be placed to the left of the graphic.
Right	Text will only be placed to the right of the graphic.
Behind	Text will flow over the graphic.
In Front	The graphic will sit on top of any text.

To select the alignment for a graphic: select the graphic and on the **Format** menu click **Shape**. On the **Layout** tab set the **Flow** to **Anchored** and click the required **Horizontal alignment**:

Other	The graphic will be aligned according to the coordinates specified in Position .
Left	The graphic will be aligned with the left margin of the paragraph, with text flowing to the right.
Center	The graphic will be placed centrally between paragraph margins, with text flowing to the right and to the left.

Right

The graphic will be aligned with the right margin of the paragraph, with text flowing to the left.

Runaround margins

When a graphic is placed on the page you can specify how far away any text that flows around it should be. The margins and the runaround properties together determine how the graphic sits within the content of the page. Some graphics only have external margins, but shapes that can contain text, such as a text box, address block, header, and footer have internal margins as well. These determine how far text and graphics that are inside the graphic are from the border.

To set the margins of a graphic: select the graphic and on the **Format** menu click on **Shape**. On the **Margins** tab enter the required **Internal** or **External** margins.

Order of graphics

As graphics are inserted onto the page they are initially layered in the order they are placed on the page. This means that later graphics will be displayed over earlier graphics if they overlap. You can adjust this order as required.

You cannot move graphics in front of or behind text in this way. You must use the runaround properties of the graphic to do this – see [Layout](#) on page 107.

Note: You can cycle through graphics in their 'layer' order by using the previous and next buttons at the bottom of the **Image** dialog box.

To move a graphic forwards or backwards one layer: select the graphic and on the **Format** menu point to **Order** and select either **Bring Forward** or **Send Backward** buttons.

To move a graphic in front of or behind all other graphics: select the graphic and on the **Format** menu point to **Order** and select **Bring To Front** or **Send To Back** buttons.

Note: The order of graphics output by Generate will vary across output devices. For example, PDF will honor any order settings when layering graphics, but other output devices may not. In particular, unpredictable results have been observed for AFP generated output. Refer to [AFP](#) on page 526 for further information.

7 - About page layouts

The primary area where content is added to document designs is known as the body which consists of one or more columns within the body area.

In this section

Working with the page layout	110
Page setup.....	111
Columns.....	115



Working with the page layout

Content will automatically flow onto new columns and pages as required. The space available for the body is constrained by the overall page size you specify and the position of any other layout areas such as the margins, headers and footers that you define.

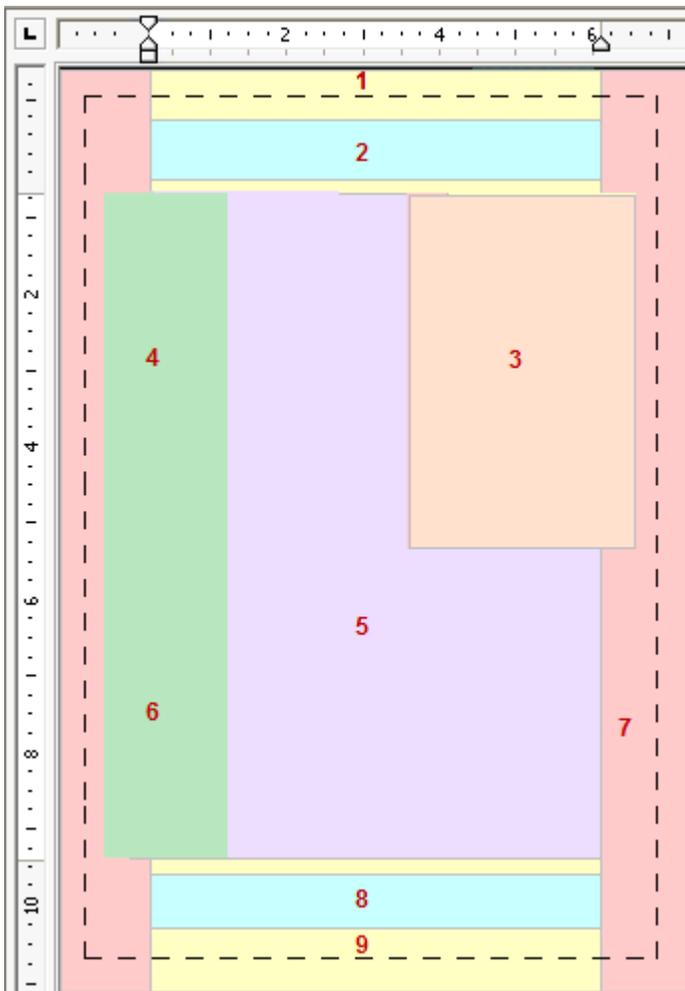


Table 4: Legend

1	top margin
2	header

3	reserved area on the background
4	margin note area
5	body area
6	left margin
7	right margin
8	footer
9	bottom margin
10	non-printable area

Documents also have a background layer, which you can use to:

- Add content 'behind' the body for example, to provide a watermark or similar
- Include a reserved area, to specify that part of the page area is unavailable for content in the body area. It is possible to update the content of a reserved area dynamically if required.
- Include a margin note area that is used to control the placement of margin notes. For more information see [Working with Margin Notes](#) on page 194.

Headers and footers can be used to specify content that is to be repeated above and below the body area. They can also be used above and below individual columns.

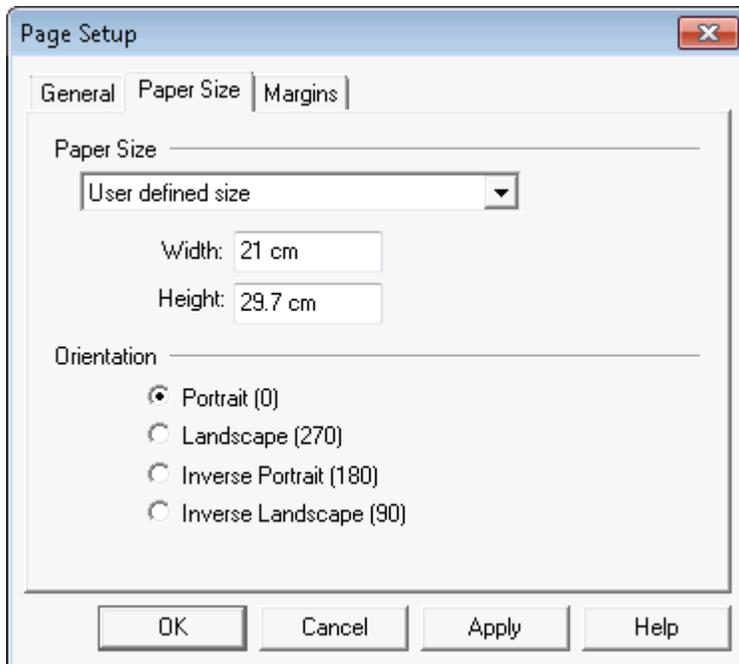
In all these areas of the page you can create text boxes that act as a container for specific content. You can use a text box simply to highlight particular content or as a reserved area around which text will flow.

Page setup

The **Page setup** view – available from the **View** menu – allows you to specify the various options that define the look and layout of the page, such as margins, background, headers and footers.

Page size and orientation

The page size is set for a complete document in the Page Setup view. This means that initially all pages in a publication are the same size. If your publication requires more than one page size, then you must create a new document for each different page size.



- Select a pre-defined paper size or select **User Defined Size** to define your own.
- Enter the width and height when defining your own size.
- Select the default orientation for the document.

Unlike page size, you can have different page orientations within a single document. The orientation is used to determine both the top of the logical page and how the physical page is printed. As well as portrait and landscape you can use the **Inverse Portrait** and **Inverse Landscape** options to specify that the page is printed the opposite way round. Inverse pages will not be displayed as such in the Designer. To change the orientation of a page you must use a document section. For more details, see [Sections and page control](#) on page 198.

To set the page size: select **View/Page Setup** and double-click Page Setup icon in the logic map. The **Page Setup** dialog box is displayed. Click the **Paper Size** tab and apply the necessary changes.

Margins and body area

Margins are the space around the edge of the page that are not available as part of the body area. They are measured from the edge of the page and you should set the size of the margins to meet the requirements of your document designs.

Many printers have an area around one or more edges that cannot be printed on. You can specify dimensions for this along with your user defined margins so that your page design keeps within the printable area.

By default, the area available for the main document body is automatically defined by the space available once margins, page headers and footers, and any reserved areas have been taken into consideration. If you want to change the body area dimensions, change any of these.

To define margins: select **View/Page Setup** and double-click Page Setup icon in the logic map. The **Page Setup** dialog box is displayed. Click the **Margins** tab and apply the necessary changes.

Background

For each page in the document view there will be a corresponding page in the background layer. These pages display the content from the document foreground grayed out beneath the background objects for that particular page.

By default, content on the background layer appears on all pages of the document although you can use conditions to restrict objects to particular pages if required. For instance, if you place an object on the third background page it will appear on every page of the document unless you create a condition to limit it to a specific page.

All objects on the background layer have a fixed position. You can place graphics directly on the background. All other presentation objects will need to be included in a text box that establishes the required position on the page. Objects on the background layer can use the entire page area if required and are not restricted by margins.

Where background content appears in the same place as content in the body area, the background material will be placed 'behind' the other content. You may want to use this to achieve a watermark effect or similar.

A reserved area is a special object only available to the background. It defines space on the page that cannot be used by content within the main document body and this will flow around the reserved area as necessary.

You can use any normal presentation feature directly within reserved areas. Additionally, you can dynamically add or replace the content within a reserved area within the main document logic.

For information about creating and updating reserved areas see [Working with reserved areas](#) on page 191

To work with the background: click the Page Setup toolbar button– or –from the **View** menu choose **Page Setup**.

To add text to the background: create a text box, position it as required and then type text into it. See [Creating paragraphs](#) on page 68.

To add graphics to the background: use the available background area to add graphics as required. See [Using images](#) on page 99 for details.

A margin note area is a defined space in the background, which is used to control the positioning of margin notes. See [Working with Margin Notes](#) on page 194 for more details. Multiple margin note areas can be defined in the **Page Setup** view.

Page headers and footers

Headers and footers contain content to be placed above and below the body area on each page of a document. You may, for instance, want to add a paragraph that includes the date in the page header or use a footer that shows the current page number.

Headers and footers are placed before and after the body area and are constrained by the left, right margins and before/after spacing. You must ensure that they are large enough to cope with the expected header and footer content. They are defined as part of the background layer.

The start and end of the body area are defined by the top/bottom page margins or the header/footer, whichever is larger.

Note: Content defined for headers and footers does not have to appear on all pages. You can place a paragraph conditionally depending on the current page number or other criteria. You can also use conditional objects with header/footer logic. See [Page and job handling](#) on page 182 for details.

To work with page headers and footers: change to the **Page Setup** view and on the **View** menu choose **Header** or **Footer** as appropriate. Enter text and graphics into the header or footer as normal.

Note: The combined size of the header and footer cannot be more than 75% of the page, thus leaving a minimum of 25% for the body area.

To resize a page header or footer: click in the area and drag the appropriate handle as required– or –in the Page **Setup** logic map, highlight the header or footer and from the **Format** menu select **Shape**. On the **Header and Footer** page, set the **Custom height** as required.

You can adjust header or footer attributes as with any shape object. See [Using images](#) on page 99 for details.

Columns

By default Designer uses a single body area for positioning main document content. If required you can split this into multiple columns. Such layouts are often known as newspaper columns. Content that is designed to appear within the main document body will automatically flow through the available column areas, top-to-bottom and left-to-right or, if required, right-to-left.

You can use the same layout for columns on all pages, use different layouts for different pages or use a combination of column layouts on a single page. For instance, the first page might have a paragraph of introduction at the top that spans the page; two or more columns may start part way down; the middle pages might contain just columns and the last page may have information following the columns such as a payment slip.

When working with multiple columns content is created in the normal way and there are no restrictions on the type of content you can include. Paragraph text will wrap within the available column width as with a single body area and will flow around anchored graphics and background items as normal.

Note: Care should be taken when using anchored graphics to ensure that they fit as required with the column boundaries.

To use columns: click in the paragraph where you want the columns to start and use the **Columns** toolbar button, choosing the number of columns required. This will create a basic column layout from the current position with no headers or footers, and equal width columns on the whole available page.

Note: You can use the logic map to specify more advanced column layout requirements, see [Customizing columns](#) on page 205 for more information.

8 - About input data

Input data is variable information from your business environment that can be used as part of your documents.

Generate uses an input data file that contains a set of such information for each publication to be produced. Typically this is a data base extract that has been prepared for use with Designer.

In this section

Working with input data.....	117
Using data aliases within text.....	117
Repeating data and transaction tables.....	120



Working with input data

Within design files, references to input data are created as data alias or data record objects that are aliases for the actual data elements that will be available in the input data file. Such objects provide indirect references to the data so you do not need to know the exact data elements required when creating your design. The fields are mapped to actual elements of input data before the publication is put into production.

Note: You may have access to multiple dictionaries depending on your environment and user privileges. In larger installations dictionaries will often be created and maintained by the Designer Administrator. See [Working with data dictionaries](#) on page 505 for more information.

Such aliases to records and fields are themselves grouped together into data dictionaries that, for example, represent all the data available to a particular type of job or a sub-set of elements that are referenced frequently.

Broadly speaking there are two types of data elements: non-repeated and repeated.

Non-repeating data typically contains information such as account numbers, names, addresses and so on. Fields that contain non-repeating data appear only once in the input data for a single publication and can be used directly within paragraph text.

Repeating data typically contains transaction type information such as account entries or itemized telephone calls. Repeating data is iterated within the data for a single publication, for example the same data structure appears multiple times consecutively. The data structure itself can consist of one or more elements that each need to be referenced as a separate data alias when used in a document design. You will normally want to use a transaction table to control how the iterations of such elements are placed within document designs.

Using data aliases within text

Fields from non-repeating data elements can be inserted directly into paragraph text as required.

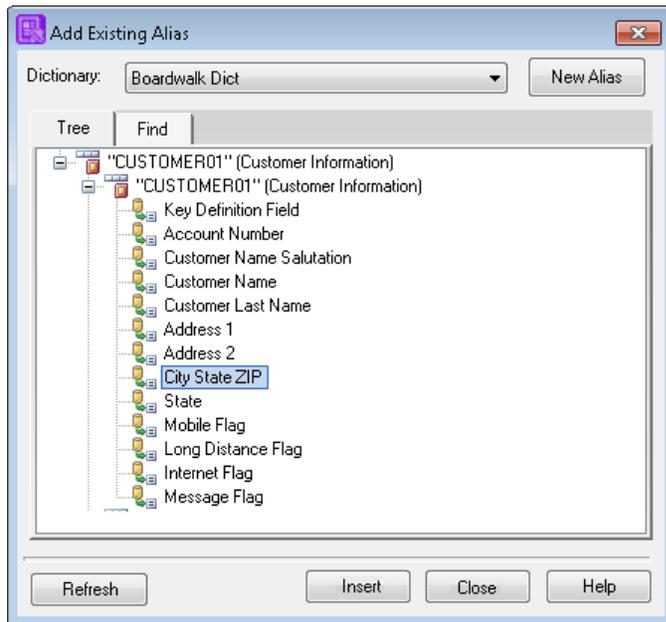
Note: References to fields can also be used to provide parameters for functions and other objects requiring values. The values provided by a field can also be stored and manipulated within a variable before being output. See [Working with values](#) on page 170 for more information.

When creating a data alias you will need to define its expected data type (string, date, number and so on) so that the correct formatting can be applied to it. You will also normally want to supply an example of the data expected in the field so that the document view presents realistic content.

Reference an existing data alias

As you enter your paragraph text you can reference existing data aliases.

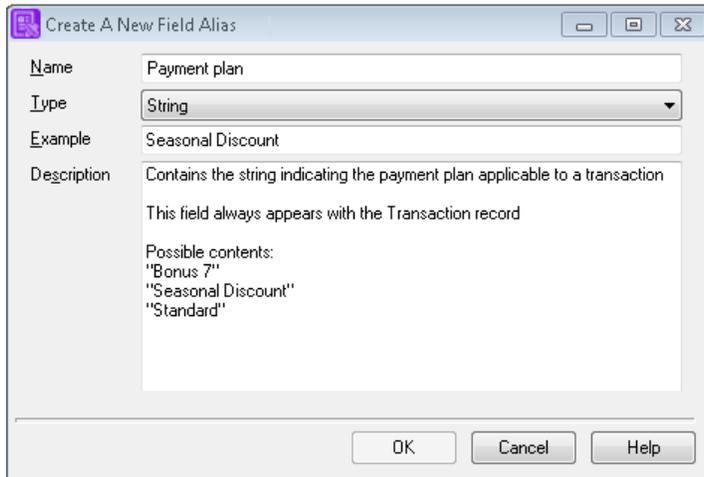
1. Position the cursor as required within a paragraph.
2. On the **Insert** menu select **Field > Data Alias**
3. In the **Add Existing Alias** dialog box select the data dictionary the field belongs to (if you have access to multiple dictionaries)
4. From the **Tree** view click on the data alias that you want to reference and click **Insert**.



Note: You can use **Find** to search for an existing data alias. For more details, see [Working with data dictionaries](#) on page 505.

Creating a new data alias

As you enter your paragraph text you can reference existing data aliases or create new ones assuming you have the relevant user privileges. If you intend to create new data aliases you need to be confident that appropriate data elements will be available in the input data files that will eventually be used with the production job.



1. Position the cursor as required within a paragraph.
2. On the **Insert** menu select **Field/Data Alias**.
3. In the **Add Existing Alias** dialog box select the data dictionary in which the new field will be grouped. See [Working with data dictionaries](#) on page 505 for more information.
4. In the **Add Existing Alias** dialog box click the **New Alias** button.
5. Enter the details in the **Create A New Field Alias** dialog box and click **OK** to create the new reference.

To change to a different data alias reference: select and then right-click on it in the document view and select **Format** from the context menu. Click the **Value** button and the **Add Existing Alias** dialog box is displayed, enabling you to change data aliases.

To format data alias content: right-click on the field in the document view and select **Format** from the context menu and in the **Field** dialog box, click the format button . For more detailed information see [Formatting values](#) on page 173. This relates to the content of the field and not how it looks on the page. You can, of course, change the font and text properties assigned to the field output in the usual way – see [Positioning text](#) on page 77 for details.

Repeating data and transaction tables

A transaction table provides the special controls required to place data elements that appear iteratively within input data.

Note: A repeating data object can also be used to handle iterative data where it needs to be processed without actually placing fields within a document design or where you need more programmatic control of field placement. See [Advanced data handling](#) on page 303 for details.

Like a regular table, a transaction table is created using the table wizard and its cosmetic attributes can be customized in the same way. See [Using tables](#) on page 262 for details.

The main areas where a transaction table differs are:

- You will need to specify the **Control Object**, for example identify the repeating data element to be placed within the table, or specify a repeat count.
- The number of rows specified for the table are automatically multiplied for each iteration of the repeating data that appears
- The table layout can appear within multiple presentation columns on the same page.

Note: If you add data aliases that are not part of the control data, for example repeating data, they will be treated as static elements.

Initially, only data aliases from the repeating data element are added to the cells of the table as part of the wizard. Once the wizard has created the table however, you can add any other text or graphics to the table cells in the normal way. Static elements of this kind will be repeated every time an iteration of the relevant row is generated by the repeating data.

As the repeating data creates additional table rows new pages are automatically generated when the height of the table exceeds the available page space.

By default, the repeating data will be presented according to the order in which it is provided in the input data file. If required, you can specify a sort that re-sequences the data prior to being placed in the table – this is available only when a data element has been selected to control the repeat.

To create a transaction table:

1. Make sure the cursor is positioned within the paragraph or object immediately before where the table is to be inserted.
2. On the **Table** menu select **Table Wizard**.
3. In the wizard choose Transaction as the table type.
4. Define a grid for the data.

On the **Rows and Columns** page specify the **Number of rows** and **Number of columns** that form the required layout per transaction. Typically you will want to create a grid that has a cell for one iteration of each field from your repeating data.

5. Select the **header** and **footer** options if required.

If selected an extra table row will automatically be inserted before/after the main grid. You can add additional rows to the header/footer once the table has been created.

6. Specify a control object.

Use the Control Data page to identify the repeating data element that will be presented in the table. Using the **Select** button will invoke the **Add Existing Alias** dialog box where you can either create a new alias or select from the list of existing repeating data elements already defined in the available data dictionaries. See [Working with input data](#) on page 117 for more information about data aliases and data dictionaries.

7. Map fields to table cells.

Use the Cell Contents page to identify where fields from the repeating data will be placed in the table.

B Cell	C Title	Field
1	Total Amount Due	Summary Total Amount Due
2	Previous Due Date	Last Month Due Date
3	Previous Amount Due	Key Definition Field
4	Payment Received	Summary Total Amount Due
5	Due Date	Last Month Due Date

Legend:

- **A** - If your grid has multiple transaction rows use the Row number control to show other rows.
- **B** - Cell numbering corresponds to table columns working left to right.
- **C** - The data aliases available correspond to the control object specified in step 6.
- **D** - Title information will be used in the header rows if specified.

8. Specify data sorting options if required.

See [Sorting](#) on page 126 for further information.

9. Specify the layout of presentation columns.

See the following section [Presentation](#) on page 123 for more information.

10. Specify general table attributes on the General Options page as required.

Selecting **Keep on same page** will attempt to keep an entire transaction table in the same presentation column.

11. Add other data as required.

Once the table has been created it will be populated with the specified fields from the repeating data element. Header and footer rows may also contain data. However, you will often want to add other text and graphics or fields outside the repeating data element itself. Click in a cell and then add new data as required.

Note: When inserting other fields you must click in the cell on the page in the document view, not in the logic map.

12. Include any table functions as required.

Running totals and transaction count can be included – see [Counts and totals](#) on page 126 for details.

To change the control data or specify advanced table controls: right-click within the table and select **Table Properties** from the context menu.

When viewing a transaction table in the editor, you can restrict the number of transactions to be shown in the document view. This could be useful if, for instance, you want to see how a table looks, without the need for real data. Use any value in the table, and set the repeat count.

Attention: This has influence only during editing and has no effect on how data is handled in preview or production.

To restrict the number of transactions displayed while editing: right-click within the table and select **Table Properties** from the context menu. In the **General** page specify the **Number of transactions** as required.

2	Across then down
3	Balanced columns. Use the Span option on a row for it to span across all columns as shown in the Header string.

- After the table has been created you can adjust the width of any individual table column as required and you can customize the line and fill options of all cells.
- You can also specify that additional presentation columns are to be used and indicate how the repeating data and associated static elements should flow through the multi-column layout.
- When working with multiple presentation columns you will need to ensure that the width of your table allows enough room for the number of columns you select.

As with regular tables, a transaction table can include header and footer rows. The content of header rows is repeated at the top of each presentation column. Footer rows are presented below the last row generated by the repeating data. When multiple presentation columns are being used you can request that the header or footer rows are expanded to span across the entire set of columns if required.

If you have entered **Title** text for any table columns this will appear in the relevant header cell by default. You can move this if required.

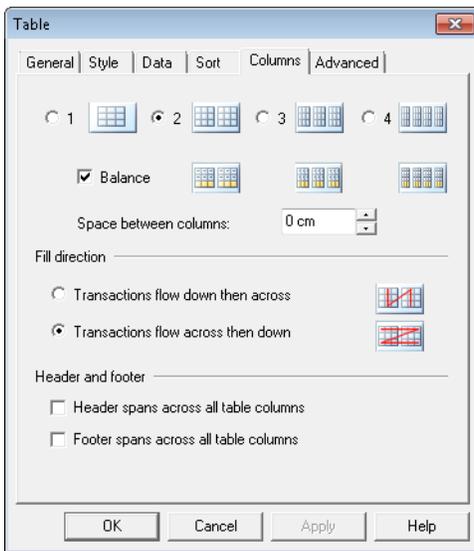
Widow and orphan settings for transaction tables relate to complete sets of transaction rows. For example, if the table has 3 transaction rows and you specified a widow value of 2, at least 6 rows would actually be kept together prior to a page throw.

To change or add to table data: edit cell contents created by the wizard in the normal way. You can add, replace or delete data aliases as required. You can also add any static data such as paragraph text and graphics.

To customize table size and style: you can adjust column widths and row height settings, add or remove cells and set their borders and shading as for a regular table. See [Using tables](#) on page 262 for details.

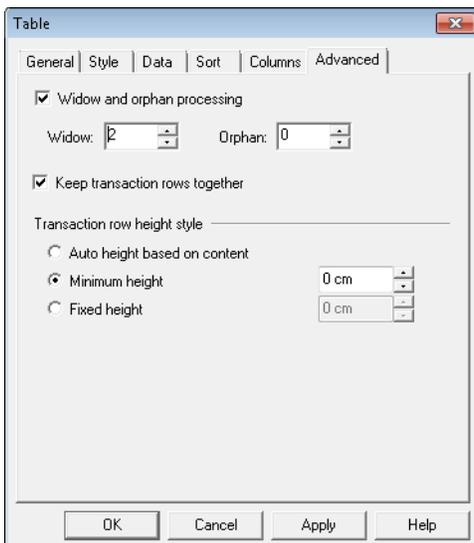
To change the presentation column layout: right-click within the table and select **Table Properties** from the context menu. Use the **Columns** page as required.

- Select the layout you want for the table on the page and specify if you want to balance the fill evenly across multiple columns using the **Balance** option.
- Choose the direction in which rows will be added. In a multi-column layout you have the option to flow the transactions:
 - down the column and then across to the next column.
 - across corresponding rows in multiple columns and the down.
- Spread the header or footer across all table columns if required.



To change settings for transaction row: click anywhere in the table and from the **Table** menu select **Properties/Table**. Choose settings from the Advanced page as required. Selecting **Keep transaction rows together** prevents transaction rows and any corresponding headers and footers from an individual repeating data element from being split across columns.

- Use the **Orphan** control to set the minimum number of transactions to be kept together at the bottom (widow) and top (orphan) of multi-column tables.
- Use the **Keep transaction rows together** option to ensure Transaction rows are not be split across columns.
- Use the **Transaction row height style** options as required.

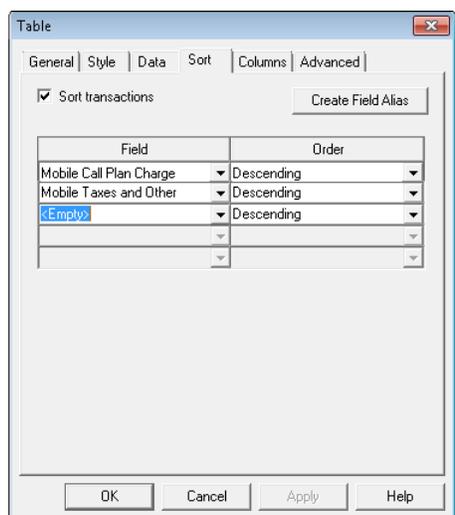


Sorting

The repeating data to be placed within a transaction table can be sorted before being output. The contents of one or more fields within the repeating data are used as the sort keys. Data in each category can be sorted either in ascending or descending order.

Sorting is based on the sequence of characters in the Unicode table. Results should be fully predictable when data contains standard ASCII-based characters but using sorting with other characters, particularly those from ideographic languages, may produce unexpected results.

To specify sorting options: right-click within the table and select **Table Properties** from the context menu. On the Sort page check the **Sort transactions** option. Enter the appropriate sort criteria using the dialog options.



- Sort order of multiple fields is top-to-bottom.
- To delete an existing key select **<Empty>**.
- Click **Create Field Alias** if the repeating data alias is to be used as sort criteria has not yet been aliased.

XML sorting

When sorting XML child elements you must ensure they are treated as fields within their parent record. Refer to [Sorting child elements](#) for further information.

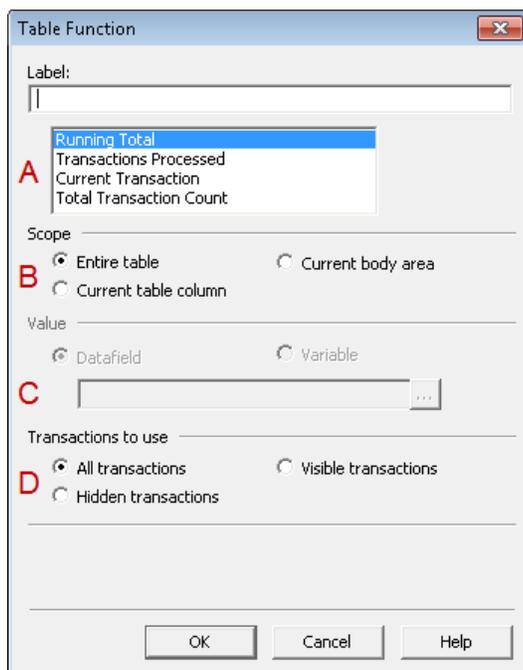
Counts and totals

Entries in a transaction table are automatically counted and numeric values are dynamically totaled.

You can use table functions to output the current count and total values within a table (or use them in table logic), as with any other value – see [Working with values](#) on page 170. The calculation of these counts and totals can be restricted to just the current column or page if required.

The table functions **Transactions processed** and **Running total** are only updated when each transaction has finished being processed, so if they are used within a transaction, the value will be that of the previous transaction. For instance, if you want to include a running total of a data alias in each transaction, you must add the current data alias to the running total.

Calculations with page and column scope can only be used in the **Table Header**, **Table Footer** or **After Last Occurrence** sections, where the final content of the page or column is known.



Legend:

- **A** - Select the type of calculation to be used ...
- **B** - ... and how much of the table is to be used.
- **C** - For a running total, choose the data alias or variable to be totaled.
- **D** - If the table has logic that only shows a subset of transactions, select which are to be included in the calculation by selecting the Transactions to use options.

To include the transaction counts and totals etc: in the document view, position the pointer in the table cell where you want the count or total, and from the **Insert** menu select **Field/Table Function**.

Running total: Running total of specified value.

Transactions processed: The number of specified transactions processed up to that point, starting at 0.

Current transaction: The number of all transactions currently processed by table logic, starting at 1.

Total transaction count: The total number of all transactions available to the table logic.

Storing counts for later use

The **Current transaction** and the **Total transaction count** can be saved in variables for use elsewhere in the publication.

To store the transaction counts for later use: right-click within the transaction table and select **Table Properties** from the context menu. On the **Data** page, select the transaction counts you want to use elsewhere and specify the variable in which they are to be saved – see **Variables** on page 178.

Nested tables

While it is not possible to create a table within a cell of an existing table, it is possible to nest a table between major row elements in the **Detail** section of another table. This is useful for presenting transaction data on multiple levels such as in the following example:

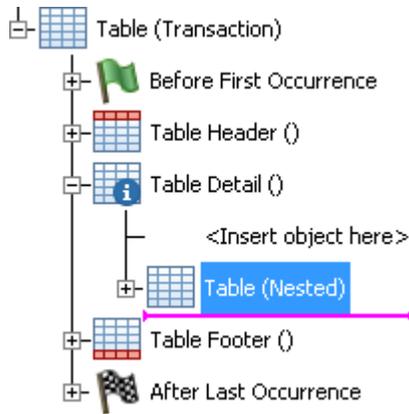
Your Transactions for May		
Market Tracker Bonds		
Standard Rates	Total cost: 35.3390	Commission: 1.00
Bought 5 at 7.0678	Total cost: 60.0100	Commission: 3.00
Sold 10 at 6.0010		
Bonus Rates	Total cost: 40.0250	Commission: 0.75
Bought 5 at 8.0050		
Your new holding: 145		
Far East High Yield Scheme		
Standard Rates	Total cost: 33.6000	Commission: 1.25
Bought 16 at 2.1000		
Your new holding: 1006		
Thank you for your business		

To do this you will need to use the logic map to mark the insertion point in the **Detail** section of an existing table when creating a second or third level table.

Nested tables can have headers and footers as with a regular transaction table. You can configure which of these should be shown if a new page is inserted while more than one level of table is being processed.

Creating and configuring nested tables

To create a nested table:



1. Display the logic map if it is not already available.
2. Mark the insertion point within the existing table.

Expand the table object and click on the **Table Detail** object.

3. Select **Table > Table Wizard** from the menu.
4. Use the table wizard to create the table settings in the normal manner.

9 - About design files and resources

At its simplest a **Designer** publication consists of a single private document design. In more sophisticated scenarios however, a publication can be made up of many different design files which are maintained independently and which can be shared by other publications. Each file is stored and controlled within the **Designer** repository.

In this section

Working with design files and resources.....	131
Version control.....	132
Using fonts and images.....	141
Custom code pages.....	146



Working with design files and resources

The following types of design files can be created:

- **Publication** – an assembly of one or more related documents.
- **Document** – the basic design unit of a publication. Typically, publications have only a single document design which is made available automatically when the publication is created. However, you can create additional document designs for conditional selection or reuse within a range of similar publications.
- **Active Content** – a file containing a segment of document logic and presentation objects; an **Active Content** object is called as a function by a document.
- **External Documents** – pre-composed documents in a supported external format that can be inserted into publications.
- **Images** – graphic files that have been imported into the repository so they can be shared by multiple design files.

Additionally, design files may make use of fonts that have been imported from your output device environment or specifically shared. These are also stored as separate files within the **Designer** repository.

Before being published a publication must be linked with a data format that describes the structure and content of the input data file to be used in the production environment. Within document designs, references to input data are created as data record or data alias objects that are aliases for the actual data elements that will be available in the input data file. Such aliases are stored for subsequent reuse within one or more data dictionary files. When publishing you will need to create or reference a data map that links the data aliases used in the design with the actual elements in the data format.

Depending on your working practices, you may decide to create some or all of these data related resources in advance of actually creating the document design. You can generate a data dictionary and a data map file from a data format if required.

All design files and resources that are available for you to work with can be found and manipulated using the navigator.

For more information about:

Creating a basic document design see [Working with the editor](#) on page 56.

Working with multiple documents design and external documents [Working with publications and documents](#) on page 156.

Creating Active Content files see [Active Content](#) on page 428.

Importing font and image resources see [Using fonts and images](#) on page 141.

Adding data aliases to paragraphs and transaction tables and creating data formats, data maps and data dictionaries see [About Designer input data](#) on page 463.

Basic navigation of files in the repository see [Using the navigator](#) on page 20.

Advanced repository topics including the use of revision control and management tools see [Managing Resources](#) on page 631.

Finding assets using the search features see [Performing an advanced search](#) on page 35 .

Version control

Version control allows you to group resources and objects together in projects for a common purpose, such as grouping together all the resources used in a publication for a particular marketing campaign or for new development.

A project can be subdivided into folders if required, giving more structure to the organization of your work. You can branch a project, folder, or anything within them, at any time. This will create a new project which could, for example be used for separate development, leaving the original unchanged.

Note: If you are not going to use the version control feature you must have at least one project in which to create most objects. Folders can be used within the project to provide a hierarchy.

Projects can also be merged. This would typically be for merging back changes once they have been tested, but any two projects can be merged together. You are given the opportunity to resolve any conflicts that would result from the merge.

All items other than code pages, fonts and locale resources can be version controlled. At any time you can mark a point in your work by applying a version label. This enables you to roll back to that version if required.

To create a new project: on the navigator select **Versions > New Project** from the menu and enter a unique name and optional description. A project with the minimum resources will be created in the navigator.

To create a new folder: on the navigator browse to the required project and select **File/New/Folder** from the menu and enter a unique name and optional description. An empty folder will be created in the current folder or project.

Branching

Create a copy of an item and work on it independently of the original. A branch can be merged back with the original, or any other project, at any time.

Any conflicts that arise from the merge must be reconciled as part of the merge process. For example, you may wish to create a branch from the production project for development. The two projects will coexist independently of each other until the development project is ready to be merged back into

the production project. This means that there is no risk to production whilst development is taking place.

Note: when branching you need to be aware of the following:

- Default Resource Map, Code Pages, Font Library and Locale Resources cannot be branched.
- Security assignments made on the original will be preserved on the branched version.

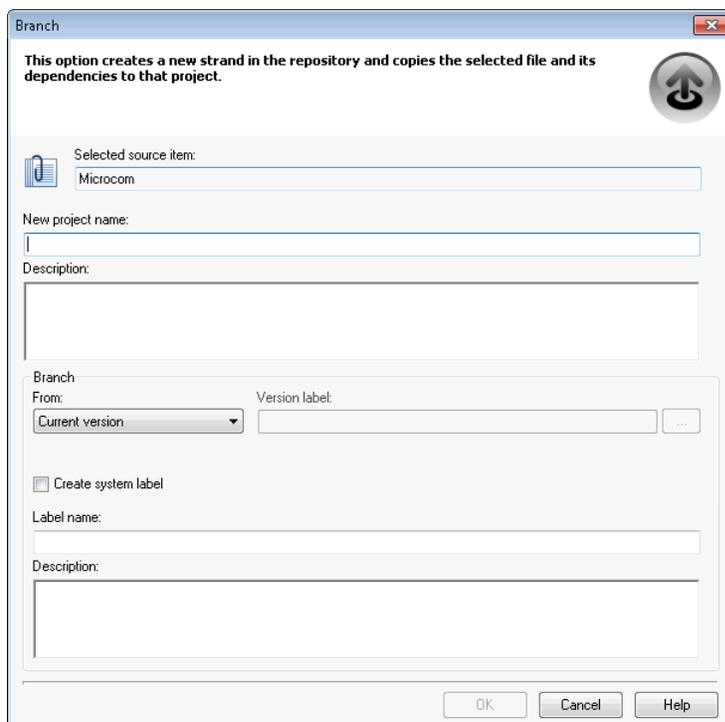
To create a branch:

1. From the main menu select **Versions > Branch** and choose whether to branch from a **Project** or from the **Selected item** in the navigator.

Use the **Branch** dialog box to specify the branch options.

2. The new project will be created in the navigator along with any files that it depends on, such as default production job, output device, host, resource map and any resources it requires.

Note: Empty folders in the original project will not be in the new project.



- Select a **source project**, or if you are branching from the **selected item** then it will be displayed here.
- Choose the version **from** which to create the branch. This can be the current version, or you can specify an existing **version label**. Use the Find button to search for a label.
- When branching from the current version you can either use the default **system label** (date and time) or enter your own **label** and optional **description**. These are shown in the **History view & Find**

Note: You can also branch, merge and label items in the history view, see [Version history](#) on page 138.

Merging

Changes that have been made on branched projects can be merged back with the original project, for example merging new development back into a live project.

You can merge any version, current or previous, of the branched project and also merge back to any other project if required. The merged items will be generated as a new version in the target project.

A comparison is made of the source and the target versions and shows which objects have changed, been added or removed, or are unchanged, and you can choose whether or not, and how to, apply each change.

The following actions can be taken when merging:

Skip:	The item will remain as it is in the target project. This is the default for items that have not changed.
Add:	This is for new items that have been added, which will be included in the target project.
Merge content:	This is for items that must be updated to preserve the integrity of the new version, such as resource maps. The content of the source item will be merged into the target item to produce an item with entries from both projects.
Replace:	The source item will replace the target item. This is the default for items that have been updated in the source project.

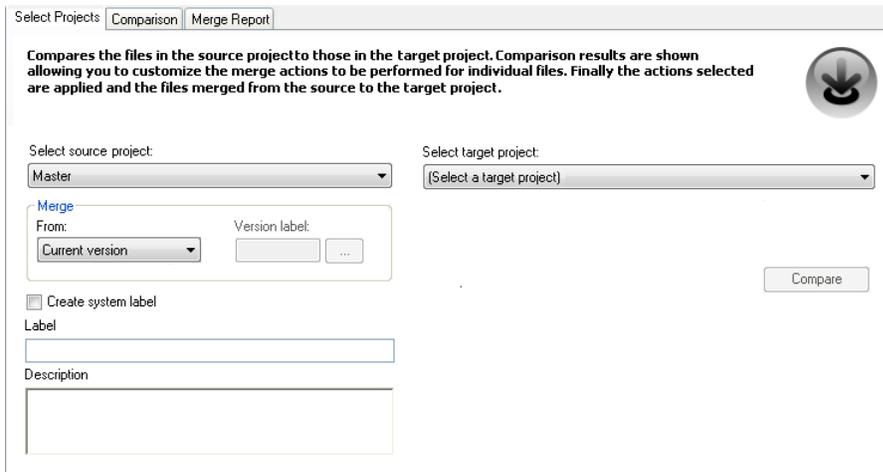
Selecting the same source and target project allows you to restore a previous version, with the opportunity to include or skip the changes that have been made since that version.

Note that you must ensure that none of the resources are in use by anyone before merging.

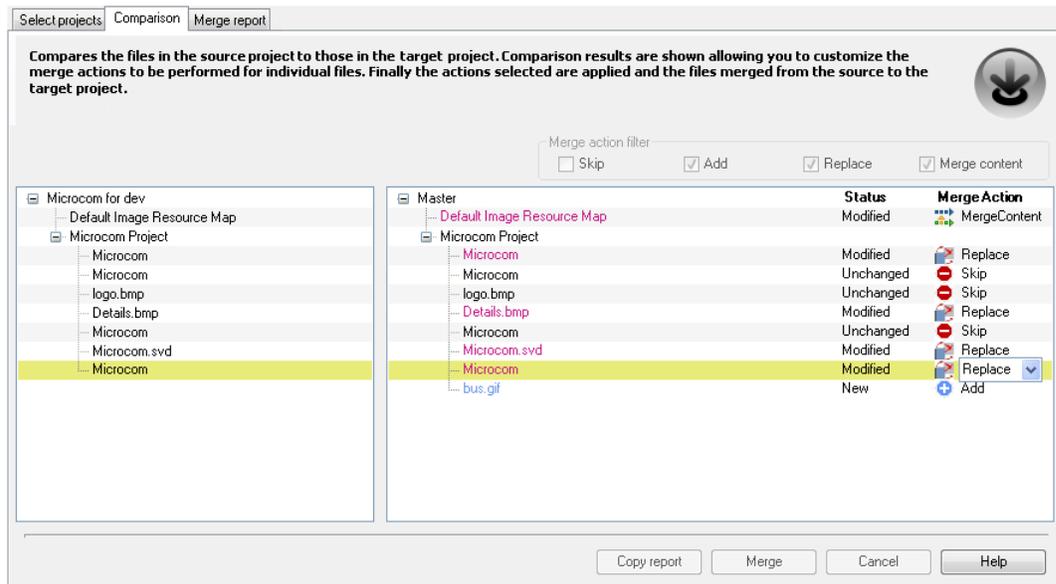
To merge projects:

1. From the main menu select **Versions/Merge** and choose whether to merge a **Project** or the **Selected item** in the navigator;– or –select the item in the navigator, right-click on it and from the context menu select **Versions/Merge**.

2. Use the **Select projects** window to specify the projects you want to merge and use the **Compare** button to start the merge process.



- Select a **source item**, or, if you are merging the **selected item** then it will be identified.
 - Select the **target project** that you want to merge into.
 - Choose the version **from** which to merge. This can be the current version or, you can specify an existing **version label**. Use the browse button to search for a label.
 - Specify a label for the new merged version
 - You can either use the default **system label** (date and time) or enter your own **label** and optional **description**. These are seen in the History view & Find.
3. Use the **Comparison** window to customize the actions to be taken in the merge process for each item.
 4. A report of the merge actions will be produced in the **Merge Report** window. As in the **Comparison** window, you can **copy** the report as plain text to the clipboard for use in a text editor. Click the **Merge** button to update the target project as specified. The source project will still be available for further use as required.
 5. Click on the **Compare** button to start the merge process by comparing the two projects.



- You can also branch, merge items in the history view –see [Version history](#) on page 138.
- Items in the project being merged are listed in the left pane. Right-click on an item to view it, its properties or its history.
- The right hand pane shows how they compare with all the items in the target project. Items in red have been modified; items in blue are new. You can override the default **Merge Action** to be taken for the selected item.
- You can copy a report of the comparison as plain text onto the clipboard for use in a text editor by clicking on the **Copy report** button.
- Use the **Merge action filter** to narrow down the list of items presented in the right hand pane.

Version labeling

Any time that an action is performed on a project, such as branch, merge, or publish using the Publish Wizard, then a version label is applied.

Enter your own label or you can use the default of the date and time. Each version label must be unique and can include a description. The label is shown in the History view when you hover over a version and the label and description are shown in search results.

Note: Once a label has been applied it cannot be changed.

A version label can be applied manually to the current version at any time, thus creating a snapshot of that project.

To apply a version label: highlight the relevant project in the navigator and from the **Versions** menu select **Version Label > Apply**

-or-

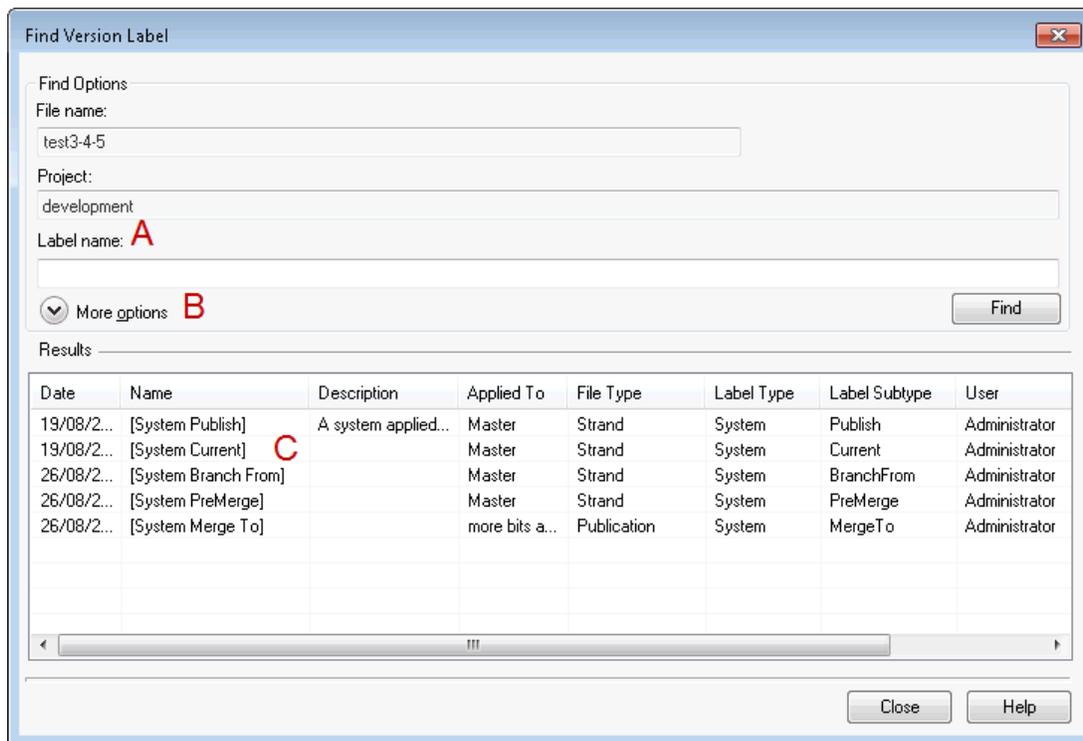
right-click on the project in the navigator and select **Versions > Version Label > Apply**. In the Version label dialog box, enter a unique version label and an optional description. You can also label items in the history view **Version history** on page 138.

To restore a previous version label: you can branch from that version and merge it with the current version. This will allow you to resolve any conflicts that may have occurred. Alternatively you can merge a previous version into the same project – see **Merging** on page 134.

To find a version label: in the navigator right-click on an object and on the context menu select **Versions/Version label/Find** – this will search within this object only;

-or-

from the **Versions** menu select **Version label/Find** – this allows you to specify what to search. Use the **Find version label** dialog box to specify your search criteria and to perform the search.



Legend:

- **A** - enter the complete label to search using ? for a single character wildcard and * for any number of characters. Alternative leave the **Label name** empty and click on the **Find** button to list all items.
- **B** - use the **More Options** control to specify additional filter options such as the type of activity, user and dates.
- **C** - double click an item in the **Results** pane for more details. Columns can be sorted by clicking in the header.

To delete a label: right-click on the required label in the history view to select it and on the context menu click **Delete**. The label is marked for deletion with a red cross and will actually be deleted when

you close the history view and have confirmed deletion. You can only **undo delete** on a label before the history view is closed.

Note: You cannot delete all labels; at least one label must remain.

Version history

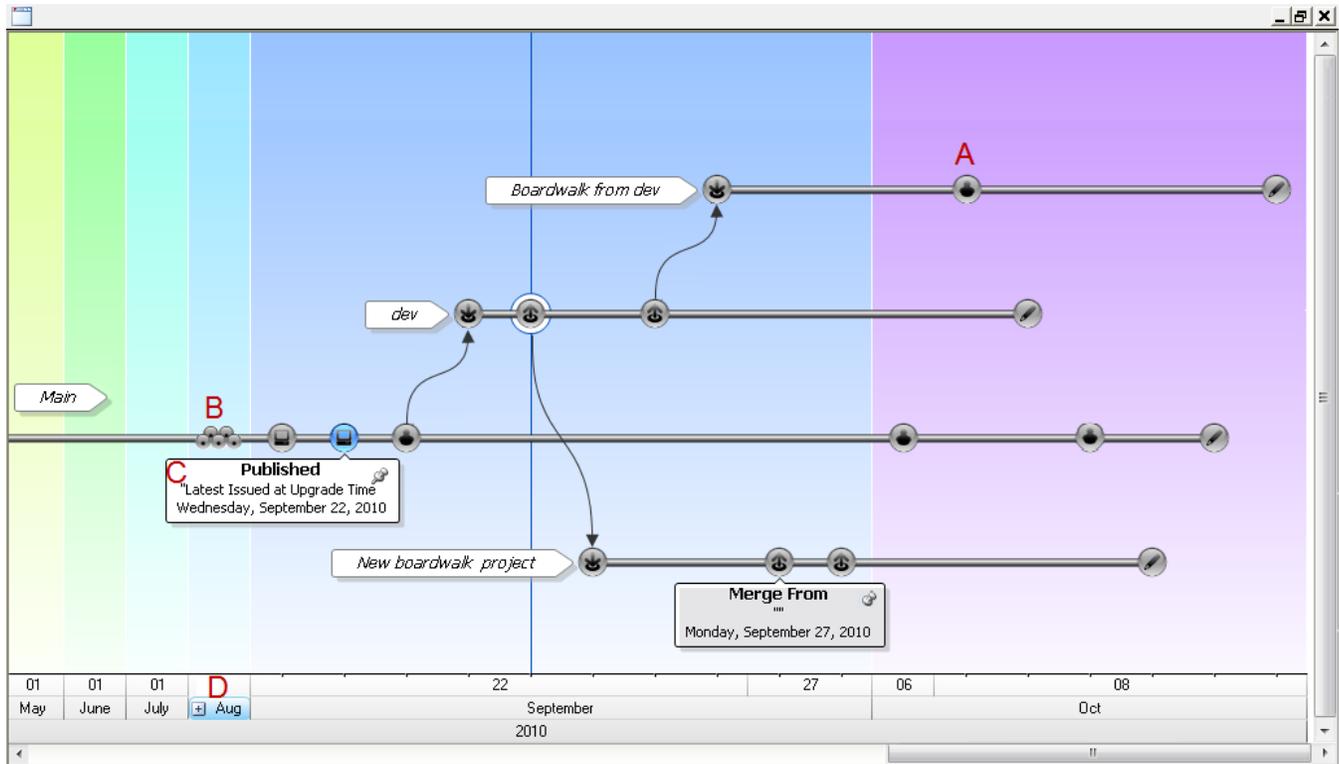
The complete version history of an object is shown in diagrammatic form in the History view. This covers everything from the initial creation of the original object to the current version and shows all branches, merges, publishing and labeling. It is set out in chronological order as a timeline. Any part of the timeline, day, month or year, can be collapsed and expanded as required.

You can use the history view to branch, merge, publish by label, apply and delete labels from any of the versions in the diagram.

To view the history of an object: select the object in the navigator and on the **Versions** menu click View history -or- right-click on the object in the navigator and on the context menu select **Versions > View History/**.

To branch, merge, publish by label, or label an object in the History view: right-click on the required label in the history view to select it and on the context menu click the relevant option. The appropriate window or dialog box will be displayed, for details see [“Branching”](#), [Merging](#) on page 134, [Working with production jobs](#) on page 618 and [“Version labeling”](#).

Version history view



Legend:

- **A** - click on a label to select it. Open, branch, merge, and so on, are performed on the selected item.
- **B** - the collapsed label symbol indicates that there are multiple labels located in the portion of the time line.
- **C** - hover over a label to show its properties and click on the pin icon to pin it to the page.
- **D** - hover over any part of the time line presented at the bottom of the screen and click on the + or - icons to collapse or expand the item.

Key to symbols used in the version history view:



Branch/merge from



Branch/merge to



Pre-merge

	Current version
	System label
	User label
	Locked
	Deleted item
	Export snapshot
	Import snapshot
	Collapsed labels

To delete a label: right-click on the required label in the history view to select it and on the context menu click Delete. The label is marked for deletion with a red cross and will actually be deleted when you close the history view and have confirmed deletion. You can only undo delete on a label before the history view is closed.

Note: You cannot delete all labels; at least one label must remain.

Using fonts and images

Typically, standard Windows-based resources are used when designing publications. When creating text you can select a font from the list of resources available to the local computer. When including images you can select a graphics file directly from the local file system.

However, where it is important to design using resources only available in your existing output device environment you may need to import fonts and images into the repository. You may also want to add resources to the repository to make them sharable by other publications or users.

Images

Designer can support a large range of image types including most common **PC**-based formats as well as those used with supported output devices.

When working with a design file you can reference images and overlays that are in the same project in the Designer repository. Images and overlays are added to the repository from the navigator. See [Using images](#) on page 99 for more information.

Importing images

To add an image to the repository: in the navigator browse to the required project or folder and on the **Tasks** menu select **Import/Images**. In the **Open** dialog box browse to and select the images you want to add and click **Open**.

Exporting images

If required, you can export images from your Designer repository to your Windows file system as follows:

In the navigator browse to the required project or folder and select the image you wish to export. On the **Tasks** menu select **Export/Images** and save as required.

Note: To use this feature you must have a user role that allows the export of **Design resources**. Refer to [User roles](#) on page 622 for further information about roles and associated privileges.

Overlays

AFP overlays typically contain pre-defined standard content such as corporate letter headings. They are often stored and used directly in the print environment and applied to output dynamically rather than being embedded in the output data stream itself.

Where this is the case the overlay is simply referenced with the data stream. Where embedding is required **Generate** always passes an imported resource in its native format directly to the output data stream. In Designer overlays can only be included as part of a document background.

Note: AFP overlay names are restricted to a maximum of eight characters. See also **Font and image resources** on page 523.

Exporting overlays

If required, you can export overlays from your Designer repository to your Windows file system as follows:

In the navigator browse to the required project or folder and select the overlay you wish to export. On the **Tasks** menu select **Export/Overlays** and save as required.

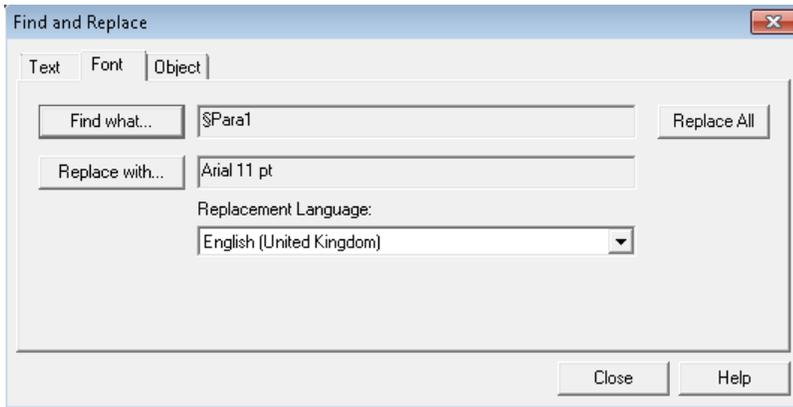
Note: To use this feature you must have a user role that allows the export of **Design resources**. Refer to **User roles** on page 622 for further information about roles and associated privileges.

Fonts

When an existing publication is opened Designer searches the local machine to see if all the referenced fonts are available. If not, they are copied from the repository into the local font directory and automatically registered with the operating system. If you prefer, you can install all the fonts currently in your repository onto your local machine. This would be useful if, for instance, you have restored a different repository from backup.

Fonts imported into the repository are converted into Windows TrueType files but note that these will be bitmap rather than outline resources. This means that it is not possible to select a different style, size or color when working with an imported font. If you want to work with variations of such fonts you must import each variation required from a separate resource file.

- Font names for AFP and Xerox must not exceed 8 characters in length.
- The fonts available are limited to those in your local font directory.
- Replacing the font replaces all instances of it in both **Design** view and **Page Setup** view.



Note: In certain circumstances imported fonts may not present correctly in the Designer editor. To avoid this issue, ensure the **Smooth edges of screen fonts** is **not** selected. This option is available from the Control Panel or, by typing **SystemPropertiesPerformance** in Windows Search.

1. **To import printer fonts into the repository:** in the navigator browse to and select the **Font library** and on the **Tasks** menu select **Import/Fonts**.

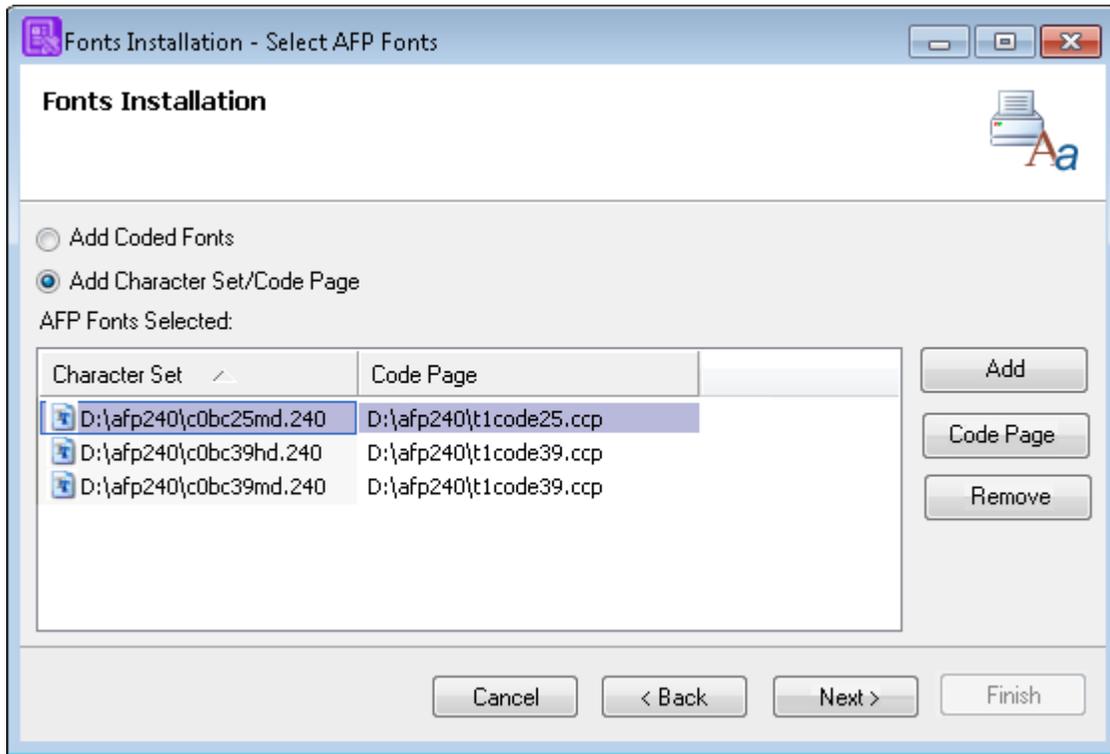
Note: You may need to use the filter to see fonts. Use the **Fonts Installation** dialog box to select the type of fonts to be imported and then identify the appropriate files.

2. **To install repository fonts to your local machine:** on the navigator **Tasks** menu select **Install Fonts from Repository** and the dialog box is displayed. Select the fonts you want to install from the list and click **Install**.
3. **To change fonts used in a document:**
 - a) In the document editor select **Edit/Find and Replace** – the **Find and Replace** dialog box is displayed.
 - b) Click the **Font** tab and click **Find** what to display the **Fonts in use** dialog box.
 - c) Select **All replaceable fonts** from the drop-down list and select the appropriate font or character style for the find and replace action.
 - d) Click **Replace All** and all occurrences of the font or character style in the Design view and Page Setup view are replaced.

AFP

Designer can import raster type fonts (not vector or outline types). You can import single-byte AFP fonts either by reference to a coded font file or to a character set/code page pair.

All single-byte fonts are assumed to be compatible with the IBM International **code page 1148**. Other single-byte code pages are not currently supported for font import. Coded font files contain references to the relevant character sets and code page files; these additional files are assumed to be present in the same directory as the coded font. **DBCS** AFP fonts can only be referenced by coded font name; you will need to specify explicitly the language encoding of the font to be imported.



- Choose between importing coded fonts or character sets. You cannot import both types at the same time.
- When importing character sets you must include a corresponding code page for the import to complete successfully.

Design font recommendations for AFP output

When designing application content for Publish to AFP output the following should be considered before using a specific Design font type

Windows True Type (.TTF/.TTC) Fonts

When designing with Windows True Type fonts, the recommended AFPDS device font format is the AFPDS device default, AFP Outline (FOCA Type1)

For True Type fonts, the older font format AFP Bitmap (FOCA) is **not** recommended as overprinting and incorrect character positioning will occur on the output AFP bitmap font for any Windows font attribute change such as font style, point size, language change, etc.

Imported AFP Bitmap fonts

For older AFP printers and some non IBM AFP printers which do not support AFP Outline Type1 fonts, you must source your own AFP Bitmap fonts and import these for use in the design editor. Subsequently publishing the publication containing these fonts to an AFPDS device whose resolution matches that of the imported Bitmap fonts is the only method guaranteed to produce accurate AFP Bitmap font output.

Publishing to AFP Outline font Exceptions

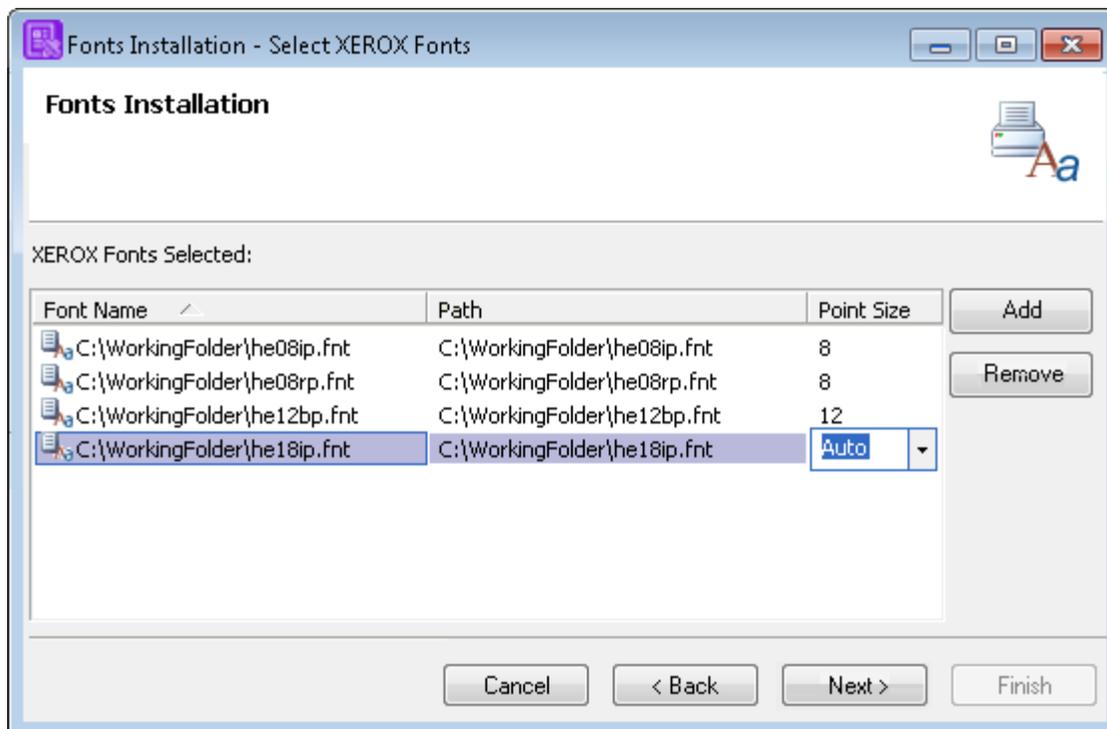
The following exceptions to a request to Publish to AFP Outline (FOCA Type1) will produce AFP Bitmap fonts and are expected ;

- Use of Designer Standard / Font Scaling Barcodes
 - Use Line Drawing Design Barcodes as an alternative
- Use of Windows True Type Symbol fonts such as Wingdings and variants
- Use of Open Type Format (PostScript Outlines) .OTF fonts
 - Map design .OTF fonts to imported AFP Bitmap fonts to ensure output fidelity

Xerox

Designer can import the FNT file format used with Xerox printers.

These are bitmap fonts with specific point sizes. In most cases the point size information is contained within the font itself but where this is not the case you may specify this as part of the import process.



Click on the Point size and select AUTO if a Xerox font contains point size information. If not, enter it directly.

Windows

Windows fonts are normally automatically added to the repository as you use them in publication designs. However, you can specifically import them in advance if required. This may be useful if you want to make specific fonts available to users in a shared development environment.

Custom code pages

Custom code pages enable users to extend the number of code pages available in **Designer**. By first exporting an existing code page and then modifying it in an appropriate text editor, you can then import it back into **Designer** as a custom code page. Currently **Designer** enables you to use custom code pages with data formats, lookup tables, **AFP** font generation and for viewing content in the **Designer Editor**.

This feature is aimed at users who use extended character sets for double-byte and multi-byte languages, such as Chinese, Japanese and Korean. Languages that use single-byte code pages, such as English and German cannot make use of custom code pages.

As the large number of characters included in a code page makes the task of creating one from scratch unrealistic, it is recommended that existing system code pages are exported as templates for creating custom code pages.

Code pages are closely linked with both languages and subsequently fonts in the product. Using a custom code page with an unsuitable language or font can result in unexpected behavior. These issues normally arise if a double-byte or multi-byte code page is applied to a single-byte language or if a font associated with a single-byte language, such as Arial, is being used.

Note: Support of custom code pages is based on IBM's open source ICU project. See <http://www-306.ibm.com/software/globalization/icu/index.jsp> for details.

Importing code pages

When creating custom code pages that will be imported into **Designer**, they must be in either unicode mappings (**ucm**) or comma separated (**csv**) file format. You must ensure that any changes made to a code page outside **Designer** conforms to the expected syntax for the file, as there is no validation of the code page when it is imported into the **Designer**.

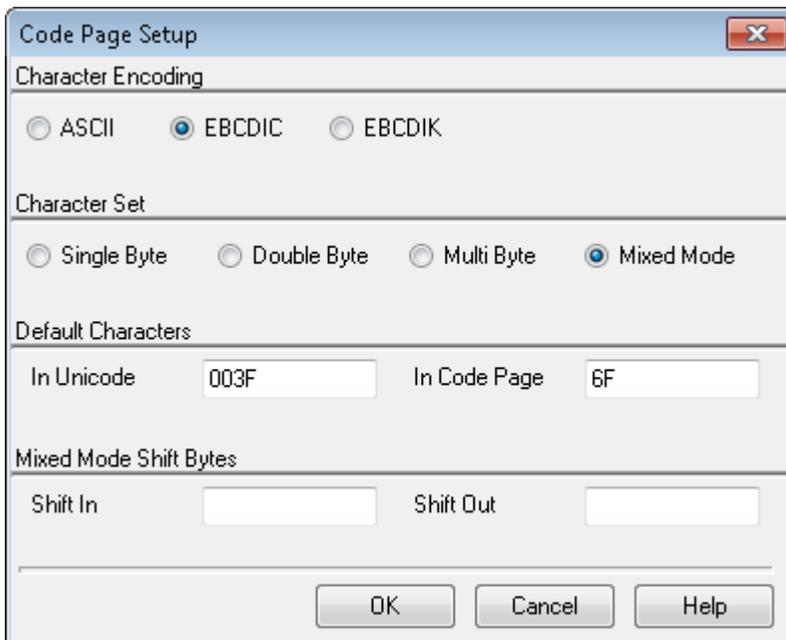
To import a code page:

1. In the navigator browse to the required project or folder and on the **Tasks** menu select **Import/Code page**. In the **Import Code Page** dialog box navigate to the required ucm or csv file and click **Open**.

- In the **Code Page Setup** dialog box apply the required settings for the code page and click **OK**.

```
<mb_cur_min> 1
<uconv_class> "MIXMODE"
<code_set_name> "JEF EBCDIC"
<icu:charsetfamily> "EBCDIC"
<mb_cur_max> 2
<subchar> \x00\x3F
<subchar 1> \x6F
0xA7,0xA1F8,0
0xA8,0xA1AF,0
0xB0,0xA1EB,0
0xB1,0xA1DE,0
```

- The import process will scan the file being imported and populate the Code Page Setup options accordingly.
- The header information included in the code page is used for setting the options during the code page import.



Code Page Setup options

Character Encoding

Designer supports the following encodings: ASCII, EBCDIC and EBCDIK (an extension of EBCDIC that contains Katakana characters).

Code Page Setup options

Character Set	<p>The code page can be based on one of the following character sets:</p> <p>Single byte – each character consists of a single byte.</p> <p>Double byte – each character consists of two bytes.</p> <p>Multi byte – each character can be made up of between one and six bytes.</p> <p>Mixed mode – each character will be either single byte or double byte. The transition between the two is done using shift out and shift in characters.</p>
Default Characters	<p>These characters are used as replacement values when there are incomplete mappings between the Designer code page and the unicode values contained in the UCM or CSV file. Typically the In Unicode and In Code Page characters used will substitute a question mark for missing mappings. This option is used when the fallback indicator calls for a substitution character.</p>
Mixed Mode Shift Bytes	<p>These options are only enabled if the Mixed Mode character set is selected. They indicate the start (Shift Out) and end (Shift In) of a group of double byte characters included amongst single byte characters.</p>

Exporting code pages

It is suggested that the system code pages included in Designer are used for creating custom code pages. To this end, you can export system code pages from Designer. These can then be edited and imported back into the Designer.

1. In the navigator browse to the required project or folder and on the **Tasks** menu select **Export/Code page**.
2. In the **Export Code Page** dialog box select a **System** or **Custom** code page, select a **CSV** or **UCM** output type and click the **Browse** button in the **Output File** field.
3. In the **Save Code Page** dialog box browse to the required folder and enter a name for the code page. Click **Save**.
4. Review the options in the **Export Code Page** dialog box and click **OK** to export the code page.

10 - About the logic map

When you are editing design files the logic map view allows you to see a representation of the objects that make up the design and how they relate to one another. It is a visualization of the sequence in which the objects will be processed when the design is executed in the production environment.

In this section

Using the logic map.....	150
Restricting active objects.....	151
Bookmarks.....	152
Notes.....	152
Searching the logic map.....	153
Logic map errors.....	154

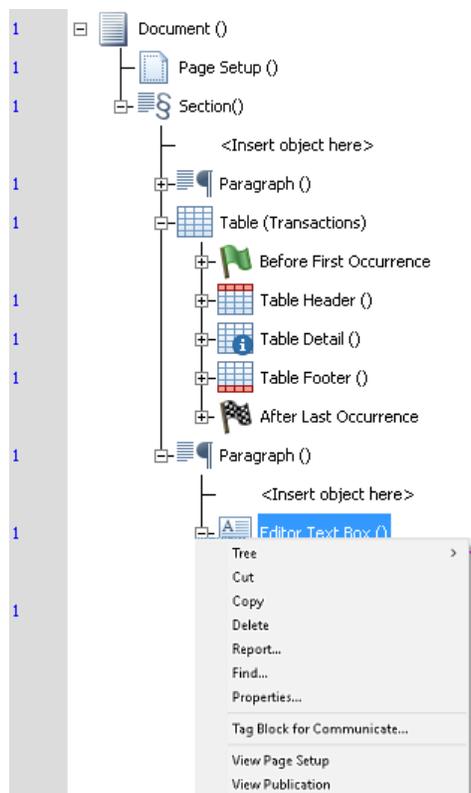


Using the logic map

The logic map is used to indicate where new objects are to be inserted. In order to work with variables, functions and other advanced features you will need the logic map to be available.

The logic map can also be used to move, copy/paste, and delete objects or groups of objects. Select an object and use standard keyboard controls or menu options to do so.

You can choose to deactivate a group of objects temporarily from the design using the Bundle feature.



- **Processing sequence:** The top-to-bottom order indicates the sequence in which objects will be processed.
- **Expanding and collapsing:** Many objects are groups that have child objects. Click on + icon to open a collapsed group; click on - icon to collapse an expanded group. Or right-click on an object and select the expand or collapse option from the **Tree** menu.
- **Insertion point:** Clicking on an object selects it as the insertion point. When a new object is created it is added immediately below this point.
- **Annotation bar (shaded left-side):** This displays additional information such as the page number within the design in which an object appears.

- **Context menu:** Right-click on an object to see its context menu (if any). The contents will vary depending on the type of object and its context within the logic map.

To move an object: drag it to a new location using the mouse.

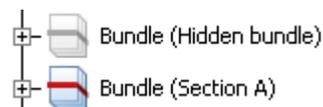
-or-

Use the **Cut/Copy/Paste** options from the context menu.

To print part of the logic map: right-click on a group object and on the context menu click **Report**. All objects that form part of the tree for the group are printed.

Restricting active objects

You can use the Bundle feature in the logic map to withdraw objects within a design file from being displayed or being processed. This can be a useful aid when testing, enabling you to concentrate on specific parts of the logic.



When hidden attributes have been specified, the bundle icon appears 'clear' in the logic map. If the contents are specified as hidden from the editor, only the bundle object itself is displayed in the logic map.

To create a bundle group

1. Mark the insertion point within the logic map.
2. Select **Insert/Tool Objects/Bundle** from the menu.
3. In the **Bundle** dialog specify the required 'hide' options as detailed below and click **Create**.

Editor - Do not show in the document.

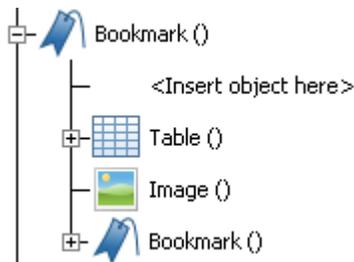
Publishing/Preview - Do not include when the document is published or previewed.

4. In the logic map move all objects to be hidden into the bundle group. Click the appropriate check boxes to indicate when the bundle is to be hidden as follows:

To reactivate hidden bundle objects: double-click on the Bundle icon in the logic map and use the dialog to turn off the existing 'hide' options.

Bookmarks

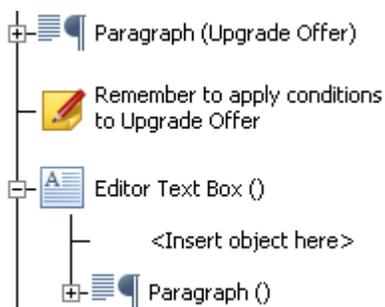
You can provide bookmarks for applications that generate PDF output. As well as grouping logic map items within a bookmark object, you can also nest bookmarks to help create the required bookmark structure.



To create a bookmark: mark the insertion point in the logic map and from the **Insert** menu select **Tool Objects/Bookmark**. Resolve the **Title** which provides the value for the bookmark.

Notes

A note object allows you to provide a comment in the logic map.

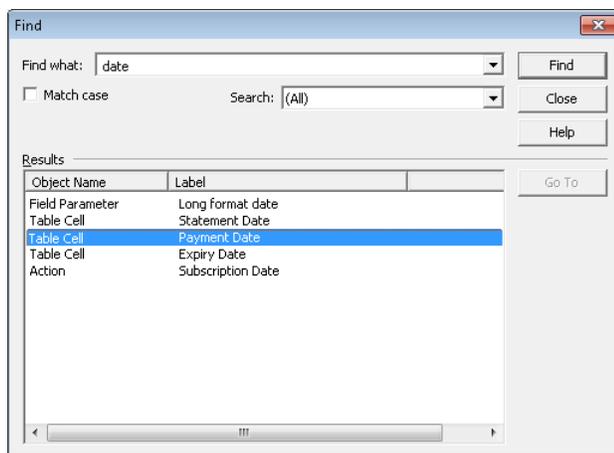


To create a note: mark the insertion point in the logic map and from the **Insert** menu select **Tool Objects/Note**. Enter the comment. The text does not automatically wrap so you must use carriage returns where required.

Searching the logic map

You can use the Find option to search the logic map. Find enables you to search for logic map objects by looking for the object or its label. You can search the entire logic map or narrow your search to specific objects.

To find a logic map object: select **Edit/Find** and the **Find** dialog box is displayed. Enter the type of object or its label in the **Find what** field. Use **Match case** and the **Search** list as needed and click **Find**.



- The **Search** list includes the logic map objects currently being used in the document, enabling you to focus the search further.
- Selecting an object in the **Results** list and clicking **Go To** will highlight the object in the logic map.

Logic map errors

From time to time your publication logic may contain inconsistencies that may prevent it from being published. For instance, you may be supplying an object with a value that cannot be converted to the appropriate data type. Or you may have used a new variable within the design but never created logic to assign a value to it.

The Document, Publication and Active Content Editors evaluate your logic and provide information about this kind of problem within the Logic Status view. In the view you may click on a message to be shown the objects to which it relates (where applicable).

Attempting to insert objects in the logic map that cannot be placed at the selected insertion point causes the **Insertion Warning** dialog box to be displayed. This acts similar to the Logic Status view.

To view logic errors: select **View > Logic Status** from the menu.

11 - About publications and documents

Although many Designer applications consist of a single document design others often need to combine document types (both internal and external) into a single production job. For this reason all Designer application designs have a high-level Publication object that allows you to build such document combinations and to set global attributes for the entire job.

In this section

Working with publications and documents.....	156
Using multiple documents.....	160
Sharing and synchronization.....	162
External documents.....	165
Working with managed publications.....	169



Working with publications and documents

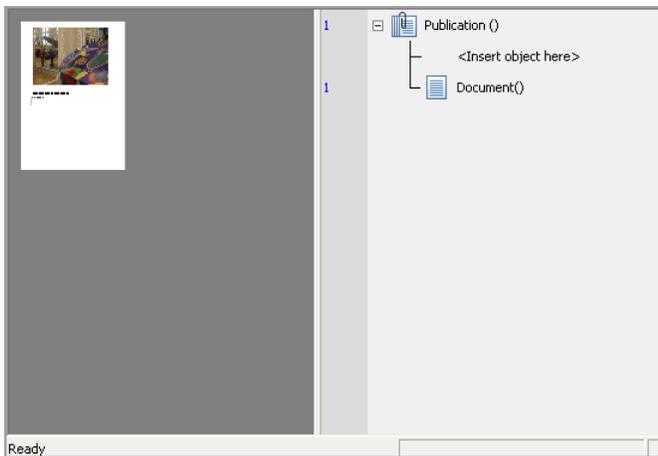
When you create a new publication what you actually create is a new document design within a publication object. By default, you will be shown the Document Editor with the new document loaded. However, at any time you can switch to the Publication Editor that allows you to add further document designs required.

Note: When you edit an existing publication the Document Editor will be shown first if it contains a single document. Otherwise the Publication Editor is loaded first.

To switch to the Publication Logic: click the **Publication logic** toolbar button

-or-

right-click in the logic map of the Document Editor and from the context menu select **View Publication**.



- The Publication View has a preview pane that shows thumbnails of the documents in the current publication.
- The logic map behaves the same as in the Document Editor but with a limited set of objects.

The Publication Editor has a logic map like the Document Editor but with a limited set of supported objects. This determines the sequence of documents within the publication and allows you to create conditional logic to decide when to use them if required. You may also add to a journal and use some action objects within publication logic.

See [Creating conditional logic](#) on page 217 for information about creating page check, condition, case, and loop objects. See [Working with Production journals](#) on page 316 for information about using journals.

A publication has a set of attributes that apply to all the documents it contains (although some may be overridden in other objects). In particular you can assign the sample data file used to provide example data in the document designs. Other optional attributes provide settings that will be used

to configure document level information in the output datastream when the publication is used in the production environment. Note that any variables used to set attribute values must be specified in the publication logic map before the first document.

To set publication attributes: double-click the Publication object in the Publication Editor logic map and the **Publication** dialog box is displayed.

Publication attributes

Job Options

Label	Identifies the publication in the logic map.
Data map	Allows you to assign a data map to the publication. This links data aliases used in document designs with actual data elements to be used in the production environment. See Advanced data handling on page 303 for details.
Sample data	Allows you to specify the sample data file that will be used to provide example data in document designs. See Data mapping on page 496 for more information.
Locale	Sets a default locale to be assigned to new documents added to the publication. Note: This is a default only and may be overridden for individual documents and sections within documents. For more information see About locales on page 311.
Spell Dictionary	Select a dictionary for your publication that can be used with the publication's locale. Like the locale, this setting can be overridden for individual documents, sections and Active Content.
Automatically synchronize with referenced files	See About synchronization on page 163 for details.

Publication attributes

Cross Browser Support

This option is applicable to publications containing interactive data intended for EngageOne. When selected, the final customer communication may be created within any supported browser without the need to download and install the ActiveX version of the Interactive editor. Refer to [Cross browser support](#) on page 402 and [How to publish for EngageOne](#) on page 405 for further information.

Note: When this option is selected:

- The template can then be used in both ActiveX environments and non-ActiveX environments.
- If the template contains designer logic objects that are not compatible with the EngageOne cross-browser environment, you will be informed of the restrictions that apply and publishing will stop.

Document information

These attributes allow you to set values that are required as document level settings in some output datastreams. They are ignored when creating datastreams other than those indicated although you may use them to provide reference information within your publication design as required. To enable these options select **Use Publication Attributes**.

Author

Select a value that identifies an author for each document. This setting is supported for PDF output.

Title

Select a value that provides a title for each individual document. This setting is supported for:

- eHTML and PDF in Compound and SPLIT output only .
- RTF SPLIT output when using the %3 option within the Generate output name

Note that this setting is not supported for PDF Merged output. For eHTML output, it is used as the <Title> element. See [HTML for e-mail](#) on page 575 for more details.

Subject

Select a value that provides a subject description for each document. This setting is supported for PDF output.

Keywords

Select a value that provides one or more keywords for document search functions when the document is opened in a browser. This setting is supported for PDF output.

Publication attributes

Password	Select a value that provides a user password required to gain access to a document. This setting is supported for PDF output and will be automatically encrypted by Generate when producing a PDF datastream.
Internet email address	Select a value that identifies the email address to be associated with each document. This setting is supported for PDF output and will be used when the emailing function in the DOC1DIME utility is used.
Internet email template	Select a value that acts as a string to be passed as a property into the email. This string is then available for use by other software packages. This option is only enabled when the Internet email address option is checked.
Permissions	For PDF output the checked options indicate the permitted level of access to documents when opened by a recipient.

Style Settings

The **Style List** contains all the styles included in the repository that are available to the publication. These styles will be available from the **Styles** drop-down list when editing a document. You can add style names in the **Publication** dialog box, but the style's attributes can only be applied in the Document or Active Content editor. The **Remove** button can be used to remove the selected style from the list or permanently from the repository. For more information on styles, [Data driven character styles](#) on page 226.

Journal Options

This allows you to create and specify job-level journals such as start and end of job and DIJ. Journals that are to be generated within a publication are specified using the journal object in the logic map – see [Working with Production journals](#) on page 316 for details.

Create Document Interchange Journal	Check this option if the output datastream created by this publication is to be stored in EngageOne Vault or used by EngageOne Deliver , and therefore requires a Document Interchange Journal (DIJ) to act as the index of documents it contains. This will enable the Edit button where you can specify the details that will actually be included in the index. See Interfacing with Vault on page 331 for more information.
Create Start of job journal	A journal will be created and records added once only, before any publications or documents are processed. If the production job consists of multiple publications, then all journals will be processed at the start. Use the Edit button to specify the contents – see Working with Production journals on page 316 for details.

Publication attributes

Create End of job journal

As above but journals will be generated after the last publication is processed.

Using multiple documents

By default, a publication is made up of a design for a single type of document. However, you can add as many document designs as required to a publication providing any input data they use can be provided within the same publication data set. A publication data set consists of the data between records defined as the **Start of publication** in a data format. For more information see [About Designer input data](#) on page 463

You may want to break down your publication into multiple documents if, for example you:

- Have document designs you want to share with other publications, see [Sharing and synchronization](#) on page 162 for details.
- Have different document designs for a range of output devices.
- Want to select document designs conditionally.
- Want to use the same publication data for different document designs.

There are two different methods for including documents conditionally within a publication. One method is to use a condition within the publication logic map. For more details see [Creating conditional logic](#) on page 217. You can also use the **Show When** option for specific documents. When this type of condition is placed within a document you can include it in multiple publications without having to recreate the condition each time it is used.

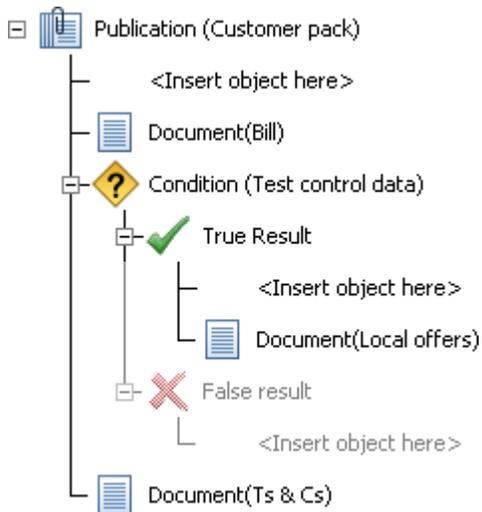
By default, all documents will be included in all the output datastreams produced by a publication. If you have document designs that are only suitable for specific output methods you can explicitly assign the appropriate types of output datastream to the document. In the production environment Generate will only use the document design when it is writing to suitable output datastreams.

To create a new document within a publication:

1. Switch to the Publication Editor.

Click the **Publication Logic** toolbar button – or –right-click in the logic map and from the context menu select **View Publication**.

2. Adjust publication logic if required.



In the Publication Editor logic map you can create any new conditional logic that may be required to select the document for processing.

3. Add a document

In the logic map mark the insertion point as required. From the **Insert** menu select **Document/New**.

4. Specify document attributes

Double-click the new document icon in the logic map to display its attributes.

Assign a label and use the **Locale** field if you want to override the default locale for the new document. If you choose a new locale you should also select a corresponding dictionary for the spell checker. Where you intend to use paragraph localization you should also assign the languages for which paragraph variants are required using the **Languages** tab. See [Localization](#) on page 310 for information about locales and paragraph localization.

Check the **Continue page count** option if you want the Document Page Count system variable to retain the number of pages generated by the previous document in the publication (if any). See [System variables](#) on page 257 for more information.

If appropriate select one or more output datastreams for the document from the **Supported Output Devices** tab. The default output setting for this is **All**.

5. Edit the new document

Right-click the document icon in the logic map and select **View Document**. The Document Editor is invoked with the document loaded.

To make a document conditional in a single publication: in the Publication Editor logic map insert a Condition from the **Control Object** menu

To make a document conditional in all publications in which it is used: in the Publication Editor double-click the document in the logic map and the **Document** dialog box is displayed.

Click the **Show When** tab, select the **Show When** check box and enter the required condition. For details on creating conditions, see [Creating conditional logic](#) on page 217.

Note: Conditional document logic is processed regardless of whether it appears in the printed output. Therefore, you must be careful when performing calculations in your conditional document logic, for example, page number increments for documents that will not appear in the final output.

To use an external document in your publication: in the Publication editor double-click on the relevant document and in the **Document** dialog box **External Document** page, browse for the file you want to import, and click **Update**. Each page of the imported document will be placed behind everything else on the pages in your publication. If necessary, additional pages will be created to accommodate each imported page. See [External documents](#) on page 165 for more information.

To update an external document: import the updated external document (see above) ensuring that the name is the same. All publications that use this external document will then use the updated document.

To remove an external document from the publication: double-click on the relevant document and in the **Document** dialog box **External Document** page, click the **Delete** button.

Sharing and synchronization

You can reuse parts of your publication design by creating Active Content files that contain commonly used sections of logic and presentation objects. Active Content can be included in your publication or document design. If it is being edited then the last saved version will be used. For more information see [Active Content](#) on page 428

You are also able to share complete document designs among multiple publications. By default however, documents are solely available for use with the publication in which they are created and you will need specifically to make them public before they can be reused.

Documents included in a publication will always inherit data assigned to the publication. When editing public documents you can associate data map and sample data independently of the publication if required.

To make a document public: click on the appropriate document icon in the publication editor and select **Edit > Make Public** from the menu.

To add an existing public document to a publication: In the publication editor mark the insertion point in the logic map. From the **Insert** menu select **Document > Existing**. Use the dialog box to select from the available documents. Remember that all documents in the same publication must share the same input data file and publication data set definition.

To edit a public document from a parent publication: in the publication editor select and right-click on the required public document in the logic map and choose the **Open option** from the context menu

-or-

open the public document in the document editor, right-click the **Read Only** document object in the logic map and select **Open** from the context menu.

About synchronization

Active Content and public documents can be edited either independently by launching them from the navigator or directly within the context of a parent document or publication that makes use of them.

If you choose to launch a shared file from a parent editor certain attributes from the parent are automatically passed to the new editor session:

- Data map and sample data assignments.
- Variable assignments.
- Locale settings.
- Highlight color definitions.

When working in this way you may want to use the **Synchronize with open files** option to refresh this exchange of data to reflect the current state of associated files open in other editor sessions. If you choose to synchronize with open files from a child editor session the parent is updated automatically. You need to be aware that changes are simply made available to other editor sessions containing related information and must be synchronized on an individual basis to view changes.

Note: If you are editing Active Content in a conditional path that is not currently active, variables, locale and highlight color assignments may not be the most recent.

To synchronize the current editor with open files: from the **View** menu select **Synchronize Editor/With Open files**.

If you launch a file independently the editor does not, by default, have access to such 'shared' values that may be provided or updated by other files that it references. For example, if a document includes an Active Content file that sets values for its parent you may often see <Not Set> when such values are displayed. You can use the **Synchronize with referenced files** option to access such lower level logic and update the display to take account of it.

To synchronize the current editor with referenced files: click the **Synchronize with referenced files** toolbar button

-or-

from the **View** menu select **Synchronize Editor/With referenced files**.

If required you can make this type of synchronization the default behavior for a publication by choosing the **Automatically synchronize with referenced files** attribute. Note, however, that for complex applications and those using many referenced files this may significantly affect the load time of the publication. See [Working with publications and documents](#) on page 156 for details of setting publication attributes.

External documents

In addition to building a publication using multiple Designer document designs you can also import PDF and Quark generated EPS files into the Designer and include them as pre-composed documents within a publication. Once included in a publication, the complete external document is viewable from the navigator. In order to import external documents you must ensure that Ghostscript is installed, refer to the Designer Installation Guide for further details.

Note: The ability to import external documents is indicated by your product license.

The external document itself is treated as a single object, similar to an image, regardless of its size or content. It is placed behind the page content, including any background and each page is aligned from the top left corner. If necessary, extra pages will be created in the publication to contain each imported page.

Note: You cannot change the actual content of the imported external document although you can re-import an updated document.

Importing External documents

To import a PDF or EPS document into the Designer: in the navigator browse to the required project or folder. On the **Tasks** menu select **Import/External Documents**. Browse to the required file and click **Open**. The external document is imported into the Designer and is ready to be used in a publication.

Exporting External documents

If required, you can export external documents from your Designer repository to your Windows file system as follows:

In the navigator browse to the required project or folder and select the external document you wish to export. On the **Tasks** menu select **Export/External Documents** and save as required.

Note: To use this feature you must have a user role that allows the export of **Design resources**. Refer to [User roles](#) on page 622 for further information about roles and associated privileges.

Viewing External documents

To view an external document: browse to and open the relevant document in the navigator.

Restrictions

- Password protected PDFs cannot be imported
- External PDF documents may not be used together with EngageOne Interactive documents

Importing PDF documents using a custom profile

This section describes the steps you need to follow to create a custom profile for importing PDF external documents to achieve optimal output sizes and quality.

It is important to note that profile changes are applied only when the PDF is imported. If you create a new or change an existing profile for an existing document, the document must be re-imported to implement the profile change.

Creating a custom profile

1. Locate the conversion profile folder. In a default client installation the folder can be found at the following location:

```
C:\Program Files (x86)\EngageOne\EngageOne  
Compose\Designer\Client\Import\Profiles
```

2. Copy the pdfimport_default.json import profile file and rename it. For example, if you are importing a file called PolicyOptions.pdf you could name the profile file pdfimport_PolicyOptions.json.
3. Open the new import profile file with a text editor and locate the "name" setting. The default is:

```
"name": "Default"
```

4. Change the "name" setting to the name of the PDF file you wish to import, for example:

```
"name" : "PolicyOptions"
```

Note: that you should not include the file extension.

5. Locate the "options" setting and modify as required, refer to [Consideration for PDFs containing transparent graphical effects](#) on page 168 and [Variations in text color](#) on page 168 for information on these settings for specific scenarios.
6. Save the file.
7. In Designer, delete the PDF document if it has already been imported and then import the original PDF file. As a unique profile file now exists for the PDF file; it will be used in place of the default.

Consideration for PDFs containing transparent graphical effects

PDF pages containing transparent graphical effects are rendered as images for inclusion in PostScript EPS files. The default conversion profile settings may create EPS files of such size that production performance and PostScript output sizes are adversely affected. In many cases it is possible to reduce the size of such files by creating a custom profile and re-importing the PDF. For information on creating a custom profile refer to [Creating a custom profile](#) on page 167. The example that follows can be used to ensure optimal output sizes.

```
{
  "type": "PDFIMPORT",
  "name": "PolicyOptions",
  "description": "My custom profile to import PolicyOptions.pdf",
  "options": "-M defaultall -m ErewriteL3a -M keepsrccolor -f",
  "logDirectory": ""
}
```

Variations in text color

In certain circumstances, imported PDF text may be affected by its background color. In this scenario, variations in text color may be observed. In many cases, you can overcome this by using the `-m Edirect` and `-M transmode=1` options in your custom PDF import profile, as shown in the example below. For information on creating a custom profile refer to [Creating a custom profile](#) on page 167.

```
{
  "type": "PDFIMPORT",
  "name": "Default_Opaque Level2Postscript",
  "description": "An alternative PDF import conversion to correct background dithering.",
  "options": "-M defaultall -m Edirect -M keepsrccolor -M transmode=1 -f",
  "logDirectory": ""
}
```

Working with managed publications

Project managers working in the Interactive environment can define projects containing publications. These publications are managed by review and approval workflows and are visible in Designer as a new project branched from your designated Master project. For detailed information about managed projects and publications, refer to [Review and approval](#) on page 446.

12 - Working with values

Working with values contains information on the following:

Adding values to objects – values provided by data aliases, variables, system variables, output from functions or constants can be used by objects to provide input to their function.

Data types – all values stored or output by a publication have an associated data type that tells Designer how to handle their contents.

Formatting values – each data type has a default presentation format. Designer enables you to change the format if necessary.

Variables – variables are used to store values for subsequent use.

In this section

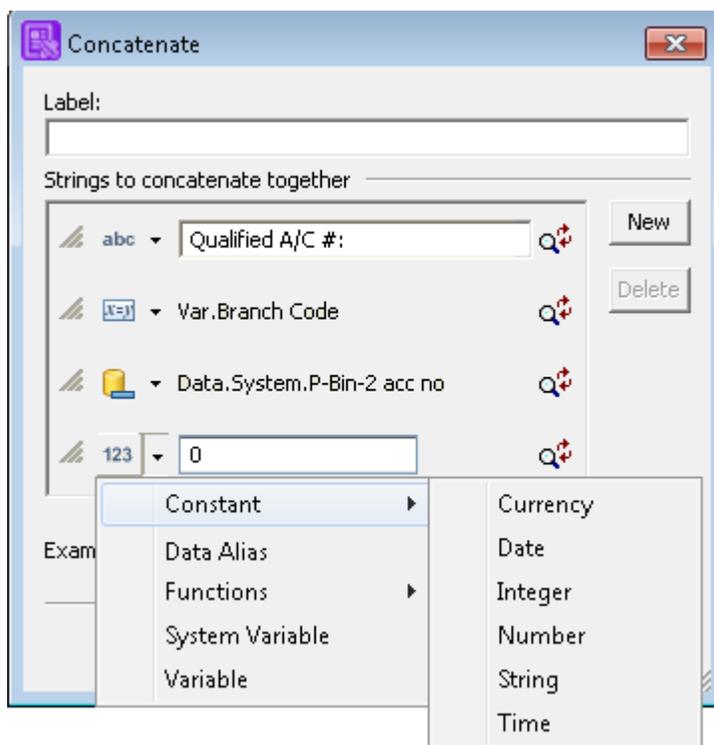
Adding values to objects.....	171
About data types.....	172
Formatting values.....	173
Variables.....	178



Adding values to objects

Many objects require one or more values to be provided as input to their function. For instance, a table lookup requires a text parameter; a condition needs two values of the same type to compare; and a substring requires the text to be manipulated plus offset and length parameters.

Values can be provided by data aliases, variables, system variables, the output from functions or you can simply enter a constant. You can also use formula expressions that allow you to create more complex parameters or as a more convenient way of adding compound functions. See [Appendix B Formula expressions](#) on page 716 for details. You identify the item that will provide the value (or the value itself in the case of constants) as part of the appropriate object dialog box.



- Type and name of the object providing the value.
- Add or remove items in a values list using the **New and Delete** buttons.
- For a constant, select its data type and enter the value required.
- Click the drop-down arrow to select the type of object to provide the value.
- Click the icon to select an existing object or create a new one.
- Use the magnifying glass icon to format your value.

About data types

All values that are stored or output by a publication have assigned data types that tell Designer how to handle their contents. Such data types may be explicitly or implicitly defined depending on the object providing the value and the circumstances in which it is used. For example, when you create a new variable or data alias you must explicitly specify the type of data it will hold but when you use the concatenation function its output is implicitly a string.

Designer supports the following data types:

String:	a string of characters or RTF with basic text formatting. String values provided by a data alias have a limit of 32k characters.
Number:	a decimal number with a maximum of 24 digits before the decimal place and 8 after.
Currency:	numeric data representing monetary values with a maximum of 24 digits before the decimal place and 8 after.
Integer:	a whole number up to 9 digits long.
Date:	a calendar date.
Time:	a time of day value.

The “E” scientific notation is supported in Number data type for use within input data.

Note: Other forms of scientific notation are not supported.

Scientific notation is not supported in variables, interactive data prompts or the interactive data Default value and any user entry field.

Some types of data may be temporarily converted at the point where they are used to provide a different type. For example, a number can be adjusted to a string for use with text related functions.

Note: Calculations that result in number values with whole or fractional parts longer than the supported maximum will result in data being truncated. For instance, the calculation 0.00009×0.0001 will produce the result 0.00000000 as the accurate result of 0.000000009 is too long for the fractional part of the Number data type.

Formatting values

All objects that provide a value have a default presentation format that is assumed from their data type, the circumstances in which they are used and the Designer locale that is active when the object is processed. For example, if a locale based on regional settings for the United States is active a value of type **Date** will typically be assumed to have the format mm/dd/yyyy. Similarly, a value of type **Number** will have a default output that includes two decimal places using the period character as the decimal point and will always be assumed to be a decimal value unless specifically cast to type **Integer**.

An initial locale is specified for each publication. This can be altered during processing by using a **Change locale** action.

You can override the default formatting when an object value is:

- Included in a presentation object.
- Used with a function requiring string input.
- Used to update a variable of a dissimilar type.

Note: Along with setting defaults for design objects, Designer locales are also used to specify custom or variable formatting of actual document content to support regional differences. For instance, you can specify the names to be used for days of the week or months, or the character to be used for the decimal point. Different locales can be activated during a production run so that these preferences are used to format output, see [About locales](#) on page 311 for details of creating and editing a locale.

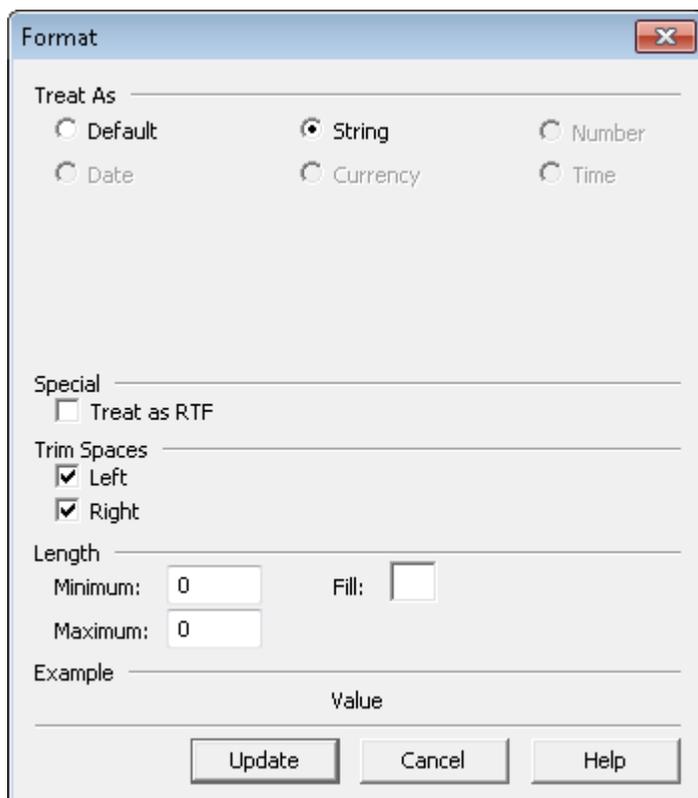
Where mixed data types are used as part of the same object Designer will attempt to convert (cast) values automatically as appropriate. For instance a string value of 15 August 2005 can be used to update a date variable. When there is ambiguity between data types used in functions or conditional comparisons you are also able to cast values explicitly to another data type.

Where values cannot be converted (either automatically or explicitly) an error will occur.

To specify formatting of a data alias or variable placed on a page: right-click on the field in the document editor and select **Format** from the context menu

– or –

double-click the corresponding **Field Parameter** object in the logic map. Click the format button and select the required options in the **Format** dialog box.



- Select how you want the value to be treated and set the options for this type of value.

To specify formatting & casting of variables used within functions or objects: in the appropriate object dialog box click the format button adjacent to the value to be formatted. Select the required options in the **Format** or **Casting** dialog box.

Format dialog box

String options

Special

Use the **Treat as RTF** option to identify a string value as being in RTF format. This can be either embedded **RTF code** or a reference to an **external RTF filename**. See [RTF fields](#) on page 176 for more details.

Trim spaces

If selected, any spaces to the **Left** or **Right** of other field content are removed before the field is placed. This is not available for RTF fields.

Format dialog box

Length	If the number of characters in the field is less than the Minimum length the remaining positions will be padded with the Fill character . The space character is used for this purpose unless specified otherwise. This is not available for RTF fields.
--------	--

Number options

Format	The Negative and Positive formats define optional text strings to be appended to number fields when the value is less than zero or greater than zero respectively. You can select from the list of standard formats or enter your own string. The Zero as options allow you either not to display values where the value is zero or to display a string of up to 10 characters in its place. For instance, you may want to show the word 'Nil' rather than 0.
--------	--

Decimal	Decimal places indicates a fixed number of decimal places to be shown by this field. Positions not supplied by input data are padded with zeros. If Exclude decimal point is checked decimal points in the field value will be ignored. For example, if the field contains 120.07 it will be formatted as 12007. If Include thousands separator is checked a separator character will be used to group digits in number values in excess of 1000. For example, assuming the separator character is a comma 2108910 becomes 2,108,910. If Round to last decimal is checked the last digit in the number will be rounded up or down to represent any remaining fraction when the number of decimal places in the value exceeds that specified. If this is not checked the number will simply be truncated to the specified number of places. For example, assuming 2 decimal places are specified, 100.009 becomes 100.01 if rounding is checked otherwise it becomes 100.00.
---------	---

Length	As for String formats.
--------	-------------------------------

Date options

Default Short	Uses the short date format of the active locale.
---------------	--

Default Long	Uses the long date format of the active locale.
--------------	---

Use location defaults	Use these buttons to set all Date format options to the styles defined for either long or short dates in the active locale.
-----------------------	--

Format dialog box

Order Specifies the order in which the three components of a date field will be presented.

Note: The use of some options will mean that one or more components will not be used.

Day, Month and Year Specifies the style to be used to represent each of the date components.

First and Second separator Specifies the characters that will be used to separate the three components of a data alias. For example in the date: 23,Jan.2007, is the First Separator and . is the Second Separator.

Currency options

Options are identical to Number formats with the exception of:

Currency symbol Select a character to be used as a prefix for the value.

RTF fields

You can dynamically include content in Rich Text Format (RTF) by passing it within a value object such as a data alias or a return from a lookup table. Such objects can either contain actual RTF code or be a reference to an external file that contains RTF. RTF code would normally be included using RTF input data – see [Advanced data handling](#) on page 303.

Note: You cannot pass RTF content to certain object types such as, plug-in charts, native charts, or barcode objects.

You can also import the contents of a complete RTF file directly into your document. See [Importing paragraph content](#) on page 89.

In the production environment Generate will format the RTF as a single uninterrupted piece of content at the point at which it is placed on the page by the publication logic. The RTF content will automatically flow onto new pages as appropriate. It is the user's responsibility to make sure that such content is only placed where it can be handled correctly; for example, RTF placed in a text box or other container may be truncated.

Note: The Preview feature will only be able to include content from a referenced external RTF file if the reference is to a file that is available to the Designer.

Simple text formatting, such as bullets, indents, and font decoration can be used in the RTF field – see [RTF](#) on page 560 for a list of restrictions. The fonts used by the RTF field should be available to the Designer, otherwise the closest matching fonts from the Designer repository will be used. If required, you can make the fonts available when publishing – see [Running a publishing task](#) on page 602 in the Publish Wizard options. Nested bullets, tables, boxes, lines, and images are not supported. Hard returns can only be used when the RTF is supplied as XML input data.

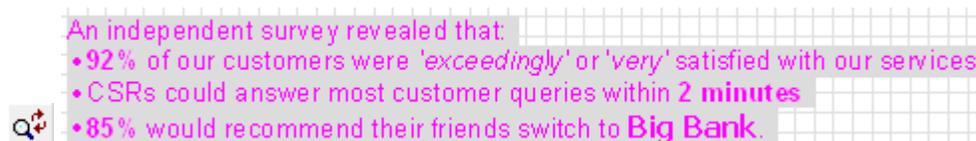
You can use variables within an RTF data alias using the DocVariable field option in Microsoft Word. When you create the RTF in Word you can include a reference, by name, to a variable contained in a Designer application. Once the variable has been set in your design document it will be displayed when the RTF field is inserted. The variable in Designer must be of type **string** and **document** scope. For more details on using Word's field option consult your Microsoft documentation.

When an RTF field is placed in a document it is displayed in RTF code. The field must be designated as RTF for it to be presented correctly in the Designer and processed accordingly by Generate. See [Formatting values](#) on page 173 for more details.

- Right-click on the RTF field and select **Format**, followed by the **Format** button. This enables you to use the **Treat as RTF** option to display the field in the Designer.



- For an external RTF file the value of the field will be shown, e.g. <External RTF:\dir\myRTF.rft>



- RTF **code** previews and prints correctly, including the text formatting and soft return bulleted list.

An independent survey revealed that:

- **92%** of our customers were '*exceedingly*' or '*very*' satisfied with our services
- CSRs could answer most customer queries within **2 minutes**
- **85%** would recommend their friends switch to **Big Bank**.

Casting

In most circumstances the data type required by a particular object is implicit and if you specify a value of a different type it will, where possible, automatically be converted to the required type. For example, if you are storing a number in a currency variable, the number would be converted to currency automatically.

In some situations however, the expected data type can be ambiguous, particularly when working with mixed type values as part of arithmetic or conditional functions. In these circumstances you should explicitly cast the values to the required type.

Where appropriate the **Casting** dialog box is automatically invoked in place of the **Format** dialog box when you select the format button within object dialog boxes.

Note: Casting provides a single conversion of a value as input for the object for which it is specified. It does not influence the original object value which retains its original data type.

Variables

Variables store values for subsequent use and can be updated as required. You can use a variable to provide input to an object or output the value it holds within a paragraph.

For example, you may want to use a variable to add number values from different sources before presenting the total, or to evaluate and hold the most recent date from a range of date fields.

A variable is an array which can be used to store multiple values of the same type. Different elements within the array are referenced by an index. The index is identified by an integer from 1 to n. The index can be provided by a value object, such as an arithmetic function or a data alias.

To use a Variable to provide the index for an array its scope cannot be Local. If the scope of a variable is changed to Local it invalidates any index value.

Variables can hold values of any of the supported data types but each individual variable can only accept data of a single type. The data type expected by a particular variable cannot change within the same publication.

By default, the value stored in a variable is cleared at the start of each new publication. Users with the appropriate design privileges can, if required, either extend this scope to an entire job or limit it to individual documents, sections, or Active Content.

New variables can be created as they are used or set. When inserting a variable into a document, you must ensure that, in the logic map, the variable is assigned the required value before the paragraph object in which it is included. Until it has a value, it will be displayed as <NOT SET>.

Variables that have an empty value are displayed as <EMPTY> when the formatting marks option is used.

To insert a variable in a document:

1. In the document view place the cursor in the relevant paragraph.
2. From the **Insert** menu select **Field/Variable**. The **Variable** dialog box is displayed.
3. In the **Variable** dialog box either select an existing variable or create a new one.

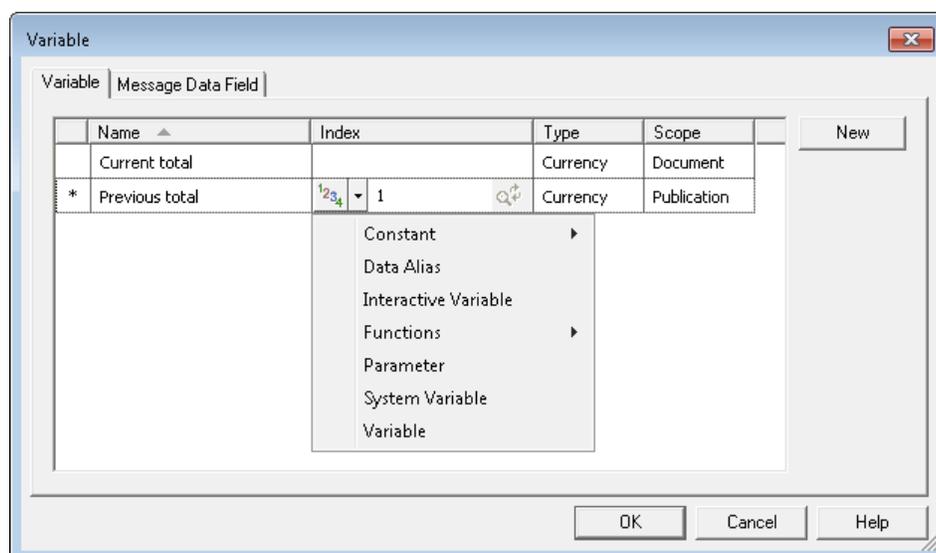
If the variable is an array click in the index field and select a value for the index number.

4. Click the **OK** button.

If you want to format the variable, see [Formatting values](#) on page 173 for details of the options available.

To use variable contents within another object: see [Adding values to objects](#) on page 171

To create a new variable: in the **Variable** dialog box click the **New** button and enter a new name in the empty entry. Select an index entry if the variable is to be an array. Specify the attributes of the variable and click **Update**. You can then assign a value to the variable.



- An asterisk indicates the selected variable.
- Select a value for the index if you want a variable array.
- Select the type of data the variable will hold.
- Set the scope of the variable if you have the necessary user privileges.

Scope options

Job

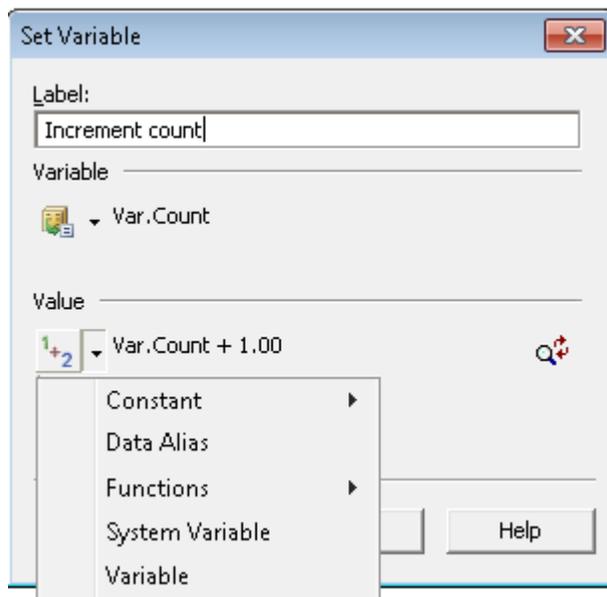
The value held in the variable is never cleared. You can use variables with Job scope to provide counters or totals across an entire production run for example.

Scope options

Publication	The value held in the variable is cleared at the start of each new publication.
Document	The value held in the variable is cleared at the start of each new document within a publication. Use this setting if you need to reuse the same logic across multiple document designs or to use the same variable name for different purposes in other documents within the publication.
Local	The value held in the variable is cleared at the start of each new section or Active Content.

To assign a value to a variable:

1. Mark the insertion point in the logic map where you want to set the value of the variable.
2. From the **Insert** menu select **Set Variable**. Choose or create a variable in the **Variable** dialog box and select **OK**.



3. The **Set Variable** dialog box is displayed. Provide a label and set the value, type and format for the variable and click **Create**.

You may often want to use a function such as arithmetic or concatenation to add to existing contents rather than just replacing them. See [Adding values to objects](#) on page 171 for details.

Note: Assigning a value actually creates a **Set Variable Value Action** object in the logic map.

Deleting variables

Variables that are not used are automatically deleted when the document is closed. Only the variables that are used are saved. As such, you cannot specifically delete a variable. If you do want to delete one, you must ensure that it is not used by any document or publication. It will then be deleted automatically on closing.

13 - Page and job handling

This chapter highlights features used to control how a document's pages are presented and processed.

In this section

Page presentation and processing.....	183
Generating new pages.....	184
Page Setup view.....	185
Working with reserved areas.....	191
Working with Margin Notes.....	194
Sections and page control.....	198
Customizing columns.....	205
Using run-time actions.....	214



Page presentation and processing

The following features can be used to control presentation and processing.

New pages – are created automatically or can be explicitly inserted at any point in a document.

Page Setup view – this view contains logic that is performed each time a document generates a new page. This includes headers and footers, as well as presentation objects placed on the background.

Reserved areas – are used to set aside space on the page for dynamic update and are defined in the Page Setup view.

Margin note areas – are used to control the positioning of margin notes, which are used for presenting feature information. Margin note areas are defined on the background in the Page Setup view.

Sections – can be used in the Document view to override the default page handling attributes that were specified in the Page Setup view.

Remaining space – allows objects to be placed in space at the bottom of a page that would otherwise be unused. **Columns** – are used to divide the body area into up to eight flow-able columns.

Run-time actions – using the action object you can specify some general actions that can affect the processing of the job at production time.

Generating new pages

New pages are automatically created when the body area of the current page is full. However you can also use the logic map to insert an explicit new page at any point.

The screenshot shows the 'New Page' dialog box with the 'New Page' tab selected. The 'Label' field is empty. The 'Restart on odd page for duplex output' and 'Manage Remaining Space' checkboxes are unchecked. The 'Create', 'Cancel', and 'Help' buttons are visible at the bottom.

The screenshot shows the 'New Page' dialog box with the 'Headers and Footers' tab selected. Under the 'Header' section, the 'Hide' radio button is selected. Under the 'Footer' section, the 'Hide' radio button is also selected. The 'Create', 'Cancel', and 'Help' buttons are visible at the bottom.

- **New Page** tab
 - **Restart on odd page** ... — Print this page on the front when printing double sided.
 - **Manage Remaining Space** — allow content to be placed in empty space at the bottom of the previous page. See [Remaining space](#) on page 201.

- **Headers and Footers** tab
 - **Show/Hide** — **Show** or **Hide** headers/footers from this page on until changed again.
 - **Hide on this page** — Override the current setting and hide the header/footer on this page only. Then revert back to original setting for subsequent pages.
 - **Continue on previous setting** — use previous page's header setting.

To insert a new page:

1. Mark the insertion point in the document logic. The new page is actioned as soon as the object is encountered.
2. On the **Insert** menu click **Control Objects/New Page**.
3. Use the tabs in the **New Page** dialog box to set header and footer options and the duplexing and remaining space settings.

Page Setup view

The Page Setup view contains logic that is performed every time a document generates a new page. Page Setup has a range of logic map groups that are executed in a fixed sequence for each page generated by a document. The table below identifies these groups and what they might be used for.

To work with the Page Setup view: click the toolbar button  or right-click in the logic map and select **View Page Setup** from the context menu.

Headers and footers

The **Default Header** and **Default Footer** objects are used to define the height of headers and footers in the current document and to format how these areas appear on the page. You can conditionally adjust the height of the header/footer areas within the Default logic. These settings apply to the entire document and cannot be adjusted in any other logic map groups.

Header Content and **Footer Content** objects are used to add content to header and footer areas. These can be created anywhere in the page setup view and are often used within conditions where selective content is required. For instance, you may want to include a different header/footer style on odd and even pages.

Note: Only one header or footer is active at a time. If more than one **Header Content** or **Page Content** object is encountered while processing the page setup logic only the content of the last object processed will be used.

You can also include content for the header/footer with the **Default Header** and **Default Footer** groups but this will be replaced if **Header Content** or **Footer Content** objects are processed elsewhere in the page setup logic.

Headers and footers can be turned on or off individually, effective for the entire document. The page content will then use any additional space. The default is for headers and footers to be on.

Note: Where the **Page Setup** view is used without the logic map being available all presentation objects are automatically added to the **Default Header** and **Default Footer** groups.

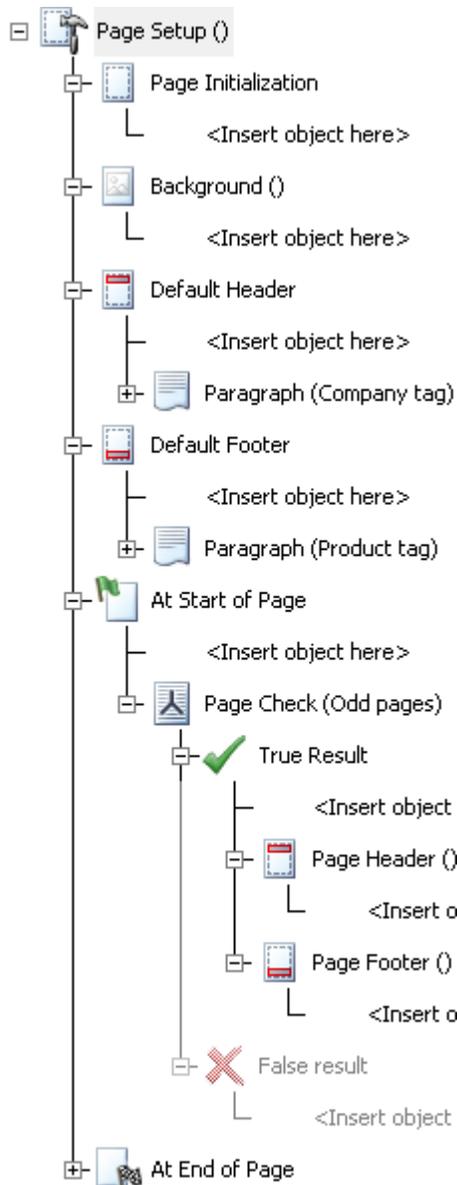
To add basic header or footer content: click inside the header or footer area and create paragraphs and graphics as normal. The objects are added to the Default Header/Footer group.

Page Setup group	Logic is executed	Can be used for
Background	Once the page initialization logic is complete.	<p>Adding content to the background layer for a watermark effect or similar (see Page and job handling on page 182).</p> <p>Creating reserved areas that are unavailable to the main document body.</p> <p>Creating margin note areas for placing feature content in a margin note, positioned adjacent to related information in the main body area.</p>
Default Header/Footer	Once the background logic is complete.	<p>Setting the height of header and footer areas.</p> <p>Adding default content for headers/footers (which is used unless a Page Header/Footer is encountered).</p>
At Start of Page	Once the default header/footer logic is complete.	<p>Any processing to be performed only at the start of a page such as updating a variable.</p> <p>Note: This should contain logic only; so Active Content that produces physical content should not be included.</p>

Page Setup group	Logic is executed	Can be used for
At End of Page	When a page in a document is about to end.	Any processing to be performed only at the end of a page such as incrementing counters. Note: This is primarily for logic, see note above, although you may include additional content on the current page provided it is included in a fixed anchor text box.

Controlling headers and footers

To change header/footer content conditionally:



1. In the Page Setup view mark the insertion point in the logic map.
You can use the At Start of Page and At End of Page groups to update the header/footer contents.
2. Create a condition object.
You will typically want to use a Page Check object to set the header/footer according to the current page number within the document. You can also use a regular condition, case or loop object as required. See [Creating conditional logic](#) on page 217 for details.

3. Create a new Header or Footer Content object.

Mark the insertion point in the True or False branch of the condition and from the **Insert** menu select **Page Header** or **Page Footer** .

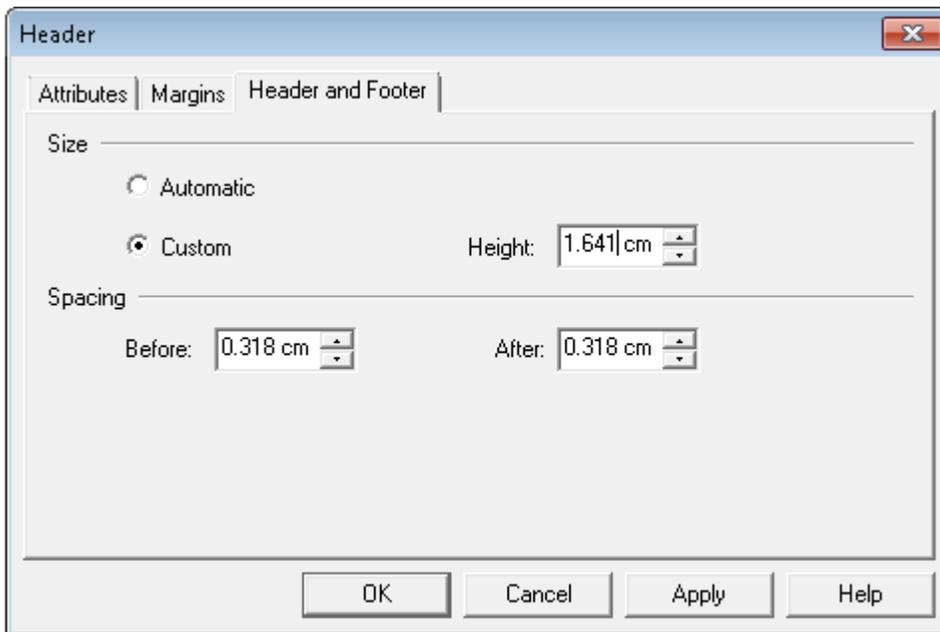
4. Add the new content.

Click within the header/footer in the document editor view and create paragraphs and graphics as normal.

To format header/footer areas: double-click Default Header/Footer or the appropriate Page Header/Footer object in the logic map. Use the **Header** or **Footer** dialog box to specify custom settings.

The screenshot shows the 'Header' dialog box with the following settings:

- Attributes** | Margins | **Header and Footer**
- Label:
- Size and Orientation**
 - Width: 8.884 cm
 - Height: 3.813 cm
 - Rotation: 0°
- Colors and Lines**
 - Line color: Blue
 - Line weight: 0 cm
 - Fill color: White
 - Shading: 0
- Buttons: OK, Cancel, Apply, Help



- **Attributes** page: Add a border or to fill the header/footer.
- **Header and Footers** page:
 - Use the **Size** options to allow for automatic adjustment of the header/footer area to accommodate the content.
 - Use the **Spacing** options to specify:
 - **Before:**Header - the distance from the top of the page. Footer - the body text to the edge of the footer.
 - **After:** Header - the distance from the edge of the header to the body text. Footer - the distance from the bottom of the page to the edge of the footer.

Text that is inside a header or footer can be rotated 90° clockwise. You can rotate more than once until the text is in the required plane.

To rotate text in a header/footer: select the text to be rotated in the header/footer area and on the **Format** menu select **Rotate Text** – or use the **Rotate** button in the Text Formatting toolbar.

To change the header/footer size conditionally:

1. Mark the insertion point in the logic map. Header/footer size can only be changed within the Default Header or Default Footer Groups.
2. Create a condition object. See [Creating conditional logic](#) on page 217 for details.
3. Specify the new size.

Mark the insertion point in the relevant branch of the condition and from the **Insert** menu select **Presentation Objects/Headers and Footers/Header Size** or **Footer Size**. Use the dialog box to enter the new size and adjust the spacing.

To turn headers/footers off: double-click Page Setup in the Page Setup view and in the **Page Setup** dialog box, select **Turn Headers off /Turn Footers off**.

To turn headers/footers on: double-click Page Setup in the Page Setup view and in the **Page Setup** dialog box, clear **Turn Headers off /Turn Footers off**.

Working with reserved areas

A reserved area defines a page space that has been set aside for dynamic update. For example, you may want to use a reserved area to create a personalized information box adjacent to a customer's transaction details. You could selectively add content to the box in response to particular types of transaction, that are processed.

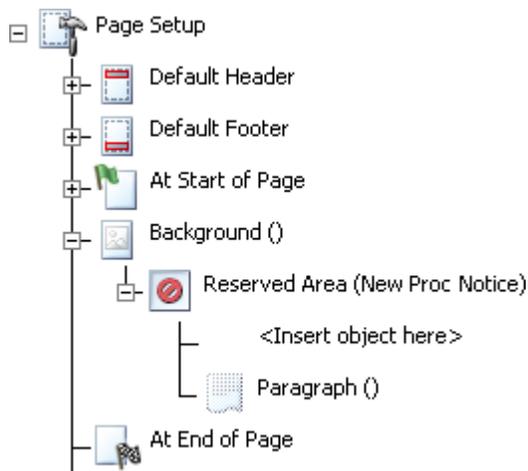
A reserved area is always defined as part of the Page Setup background layer but influences the presentation of objects within the body area where these overlap. Normally you will want your body content to flow around the reserved area although all the regular text flow options are available.

By default, reserved areas apply to all pages although you can use conditions within the Background to selectively include them on particular pages if required.

Initial content for a reserved area can be specified on the background if required. The content is updated by using a Reserved Area Entry object within your main document logic. When updating, you can choose to append to any existing material or clear the reserved area before adding the new content.

A reserved area always has the same position and size on every page on which it appears but you can create any number of reserved areas as required. A reserved area can contain any type of presentation object except for another reserved area.

In terms of positioning, color, line, and fill attributes and text runaround properties, a reserved area behaves exactly the same as a text box or other shape. See [Working with graphics](#) on page 95 for details.



The initial content of a reserved area can be inserted in the Page Setup view either by entering it directly or by inserting objects in the **Reserved** object in the logic map. This content can be updated dynamically via the logic in the document view.

Creating reserved areas

A reserved area is always defined as part of the Page Setup background layer but influences the presentation of objects within the body area where these overlap. Normally you will want your body content to flow around the reserved area although all the regular text flow options are available.

By default, reserved areas apply to all pages although you can use conditions within the Background to selectively include them on particular pages if required.

1. In the Page Setup view, mark the insertion pointer in the **Background** object in the logic map and from the **Insert** menu select **Tool Objects/ Reserved Area**.
2. Click in the page layout and drag the pointer where you want the box to be drawn.
3. Enter initial content (if any).

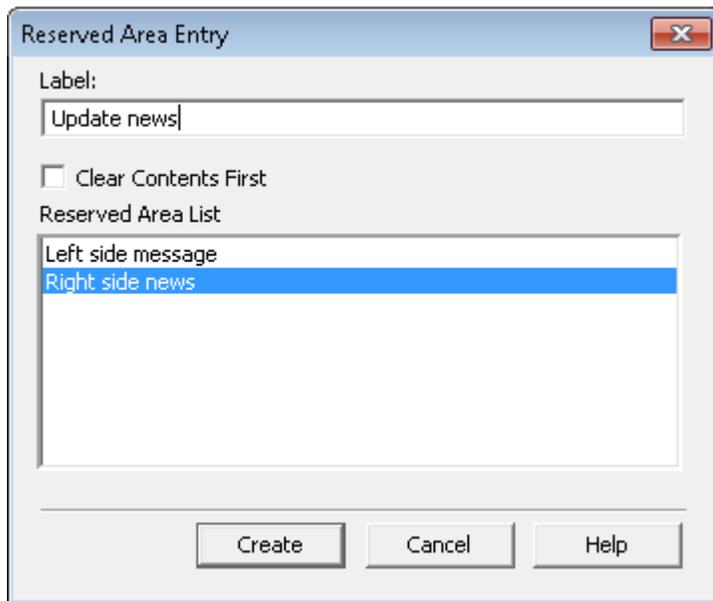
You can create paragraph text and graphics as with a text box. Content will wrap and flow down the available space as with the main document body.

Note: If the reserved area is not large enough, it will grow downwards as you add new lines in the background, but will not wrap onto a new page. Content added via the document view will be truncated if it does not fit.

Creating a reserved area entry

Initial content for a reserved area can be specified on the background if required. The content is updated by using a Reserved Area Entry object within your main document logic. When updating, you can choose to append to any existing material or clear the reserved area before adding the new content.

1. In the document view mark the insertion point.
Click on an object in the logic map to indicate the point at which the update will be performed.
2. Select **Insert/Tool Objects/ Reserved Area Entry** from the menu.



3. Use the **Reserved Area Entry** dialog box to specify update settings.
4. Click **Create**.
5. Add new content.

In the document editor the selected reserved area will become active. Create paragraphs and graphics as normal.

Working with Margin Notes

Margin notes are typically used to present related feature information adjacent to content in the main document flow but not interrupting it.



Customer name **RUTH ANN FLEMING**
 Phone number **714-336-7093**
 Customer number **0153559**
 Billing period **08/29/00 to 09/29/00**

24 hour customer service
 1-800-891-7770
 or call our Actionline
 Monday to Friday 8am to 5pm
 1-877-435-7726

Page 3 of 6

Please let us know if your name, address, or any other personal details have changed

BREAKDOWN OF ACTIVITY

Previous activity	AMOUNT
Previous balance	\$0.00
Payments received	- \$0.00
Credit	\$0.00
Balance before new charges	\$0.00

Reduce your charges with our Buddies scheme!



YOU QUALIFY!
 free calls after 6pm

Residential Access Line	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Touch-Tone	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCN Basic Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCB Premium Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCB Silver Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCB Gold Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCB Platinum Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Ultra Feature Package	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Ultra Feature Package Deluxe	9/29/00	10/29/00	1	0.00	0.00	\$0.00
HUNTINGTON Ultra Service	9/29/00	10/29/00	1	0.00	0.00	\$0.00
900 Blocking	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Toll Restrict 976 Blocking	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Call Waiting	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Caller ID With Name	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Deny Return Call	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Call ID Blocking Per Line	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Anonymous Call Rejection	9/29/00	10/29/00	1	0.00	0.00	\$0.00
IDLC Test Line	9/29/00	10/29/00	5	0.00	0.00	\$0.00
International Plus	9/29/00	10/29/00	1	0.00	0.00	\$0.00
RCN Long Distance	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Calling Card	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Number Portability Charge	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Telcom Relay Service Charge	9/29/00	10/29/00	1	0.00	0.00	\$0.00
911 Surcharge	9/29/00	10/29/00	1	0.00	0.00	\$0.00
Federal Line Cost Charge	9/29/00	10/29/00	1	0.00	0.00	\$0.00

Margin notes behave similarly to text boxes. You can place text, images and charts inside margin notes but you cannot insert message boxes, tables, reserved area entries, or Active Content. For example, you may want to use a margin note to highlight particular information presented in a document that should be verified by the customer, or advertise a special offer that relates to a particular type of transaction.

Example margin notes are shown as blue areas in the diagram; the yellow portion indicates the margin note area.

You can place a margin note inside Active Content without a margin note area being present. The margin note will be included in the logic map but not shown in the document until an associated margin note area is created.

Margin notes are anchored to a paragraph in the main flow, or in a table cell. If the paragraph changes position the margin note will move accordingly. Similarly, if one overflows a page then they both move to the next page. The same behavior exists within a table cell.

Note: It is not recommended that margin notes are used with columns as the same margin note area is used for all columns on a page. Margin notes from the second and subsequent columns will be placed over those from previous columns.

If a margin note is vertically offset from its anchor to such an extent that it would be positioned off the page, or cause the overprinting of headers, the margin note will be positioned at the top of the flow area on a new page. Clicking on the margin note will display a screen indicator, which shows the relationship between a margin note and its anchor.

Creating margin note areas

Margin notes are used in conjunction with a margin note area, which reserves space in the document design for controlling the horizontal placement of margin notes.

To create a margin note area:

1. In Page setup view mark the insertion point in the Background object in the logic map. From the Insert menu select **Tool Objects > Margin Note Area**
2. Click in the page layout and drag the pointer where you want the box to be drawn. The width of the area is set by the drawing action. The height is fixed to the extent of the page, excluding any header and footer.

To specify margin note area settings: select the area and on the **Format** menu click **Shape**

– or –

right-click the Margin Note Area object in the logic map and select **Properties** from the context menu. Color, line, and fill attributes, and text runaround properties are handled in a similar way to any other shape object. If using multiple margin note areas in a document it is recommended that you use the Attributes tab to assign a unique label. For more information, see [Working with graphics](#) on page 95.

Creating margin notes

Margin note areas are defined as part of the **Page Setup** background layer and have no presentation content of their own. The content is provided by the margin note, which is created within the main document logic.

To create a margin note:

1. In document view click within the paragraph that you want the margin note to be anchored to.

Note: You cannot create a margin note in a document unless a margin note area is present.

2. From the **Insert > Presentation Objects** menu, select **Margin note**– or –use the **Margin Note** button in the Objects toolbar.
3. Create the margin note box.

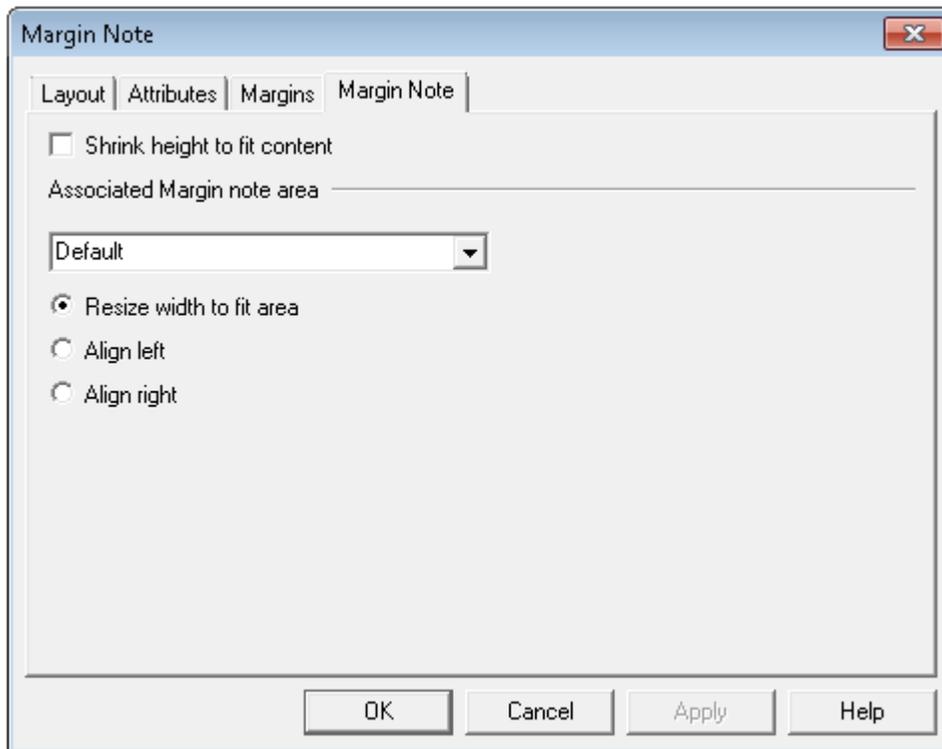
Click in the page and drag out the shape to the required size. The margin note will automatically be positioned in the default margin note area, adjacent to its anchor object and will be resized to fit the width of the margin note area.

4. Add content as required.

Changing margin note settings

Height, area and identifying information can be changed using the **Margin Note** tab

1. Select the area and from the **Format** menu click **Shape** – or –right-click the **Margin Note** logic entry and select Properties from the context menu.
2. The **Margin Note** dialog box is displayed. You can adjust margin note attributes as with any other shape object. See [Working with graphics](#) on page 95 for details. Note that you cannot change the width of the margin note area if the Resize width to fit area option is selected.
3. Use the **Margin Note** tab to set optimal height and width, adjust horizontal alignment, or to change margin note area association. If associating a margin note placed within Active Content that is wider than the margin note area, the content will not be reformatted. You will have to adjust the margin note manually.



Reduces the height (in Generate) of the margin note box to fit the text.

Note that the default margin note area is logically the first one placed on the page.

Select or type in the name of the margin note area that you want to associate the margin note with.

Sections and page control

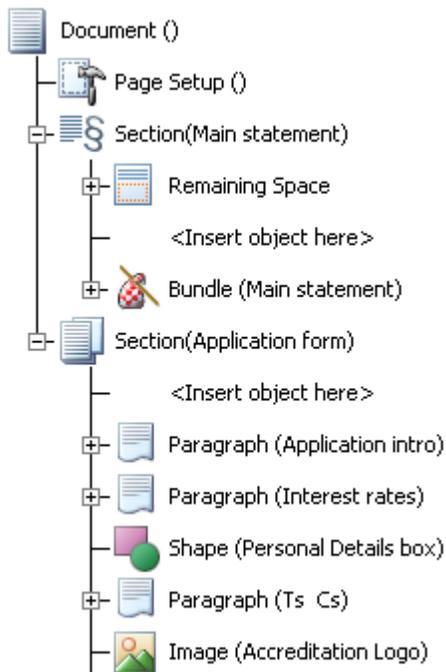
When new pages are generated they will, by default, use the page handling attributes specified for the document in the current Page Setup object and the initial Section in the Document view. These attributes include:

- Page orientation
- Page and sheet throws
- Header and footer use
- Remaining space
- Column layout

By using additional sections you can override these defaults for particular parts of your document logic. For example, you may have a document design where the final page is an application form

requiring landscape orientation, a different configuration of headers and footers, and which always needs to be printed on a new physical sheet on the printer.

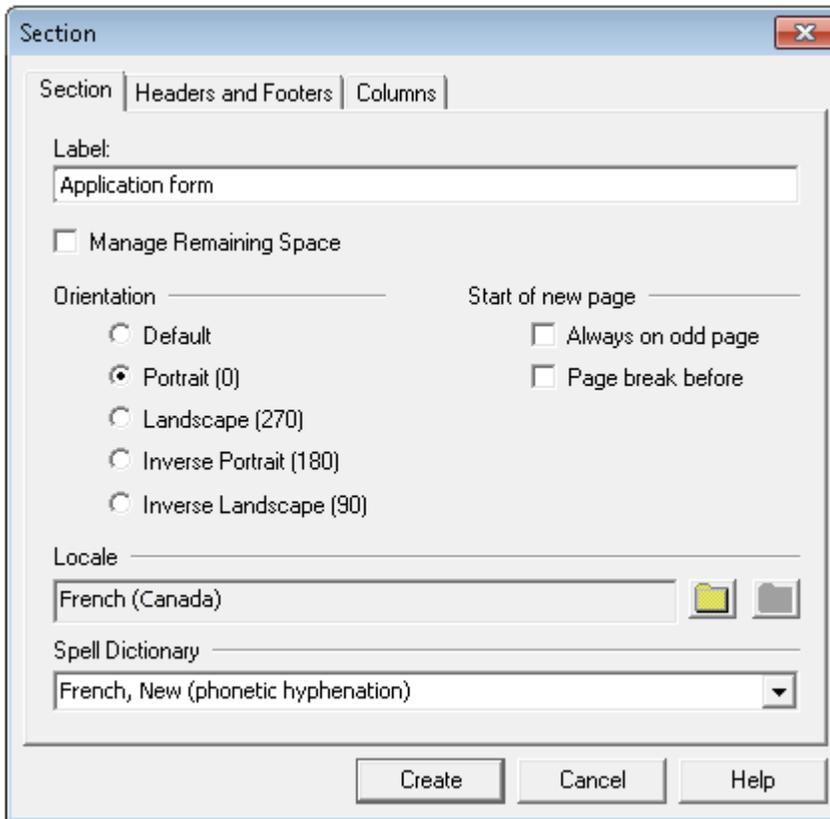
For such requirements you must include a section within your document logic and include presentation objects that require the special page attributes within the section object. Note that a document is automatically created with a single section and all presentation objects will be added to this group in the absence of other sections.



To create a new section:

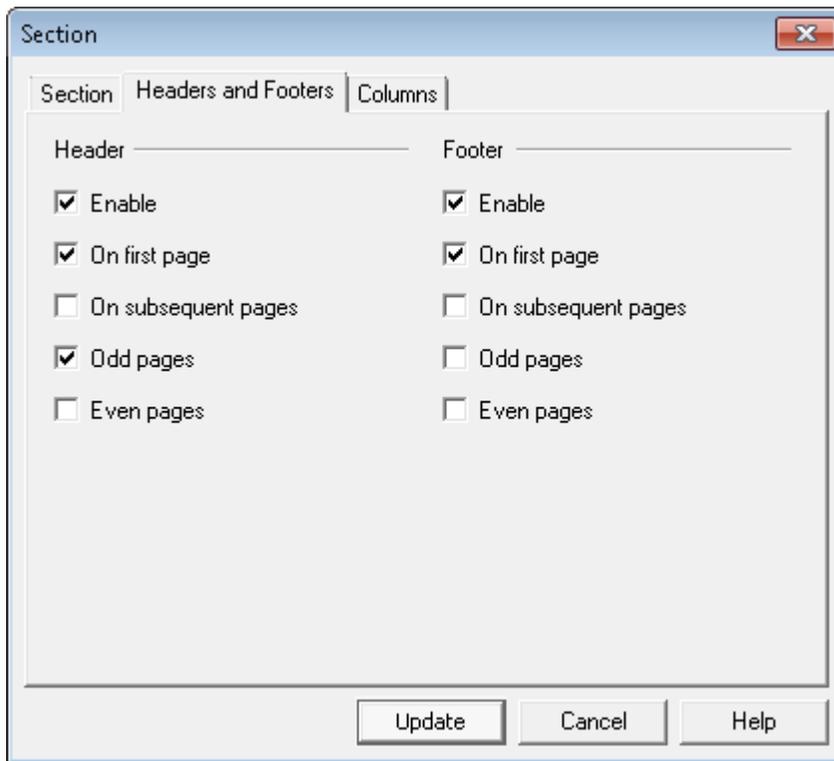
1. Mark the insertion point in the document logic.

The page handling attributes of the section will apply as soon as a new page is generated by the objects within it. You can specifically request a new page as part of the section itself if required.
2. On the **Insert** menu click **Section**.
3. In the **Section** dialog box specify the page handling attributes you require.
4. Add the objects to which the page handling attributes are to apply within the section.



Section tab:

- Use the Manage Remaining Space option to enable objects to be placed in unused space at the end of the section.
- Choose **Portrait**, **Landscape**, **Inverse Portrait**, or **Inverse Landscape** to override the **Default** layout for objects within the section.
- Use **Always on odd page** to enforce a new sheet of paper when printing double sided.
- Use **Page break before** to initiate a new page immediately with these attributes.
- The **Spell Dictionary** must correspond to the language of the **Locale**.



Headers and Footers tab — use these options to customize the header & footer configuration for pages created by the section.

Columns tab — See [Controlling column content](#) on page 209 for information on working with columns.

Note: You can also specify a new default locale that will be active while the section is processed. See [About locales](#) on page 311 for details.

Remaining space

Once regular content has been placed in the body area there may still be sufficient space left for additional content. This can occur at the end of a section or when a new page has been created and the previous page is not full. This remaining space, or white space, can be used for optional content such as advertisements or promotional material.

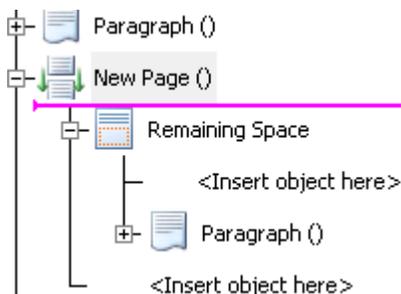
Most types of presentation object can be placed in remaining space and you can use conditions and loops on them in the logic map. Objects are placed in the sequence defined by the logic map and must fit completely to be included. You can choose either to stop processing remaining space the first time there is no room for an object, or to continue trying all subsequent objects in turn to see if they will fit, discarding any that don't.

NOTE: The following objects are not allowed in remaining space: section, column layout, reserved area, margin note area, overlay, margin note, message box, message stream, transaction tables, repeating data, new page actions.

Groups of objects, such as tables and text boxes are treated as single items, as are paragraphs that have the 'keep with next' option set. Active Content can be treated either as a single item or as separate, individual objects. Content that cannot be placed is not retained for use in other remaining space areas. Any invalid objects encountered will cause remaining space processing to stop (with an error) but valid objects that have already been placed will remain.

NOTE: If Active Content contains an object that is not valid for remaining space then, as a single item, it will be rejected completely or, as independent objects, those that have already been placed will remain, with the rest being rejected.

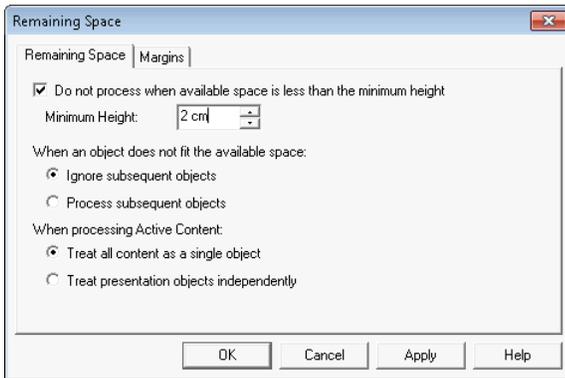
Remaining space can be specified to appear at the end of the current section or at the end of the previous page before a new page object. The remaining space area starts from below the lowest content in the page body area or the bottom of column body and column footers (excluding graphics) to the bottom of the page body area. The width occupies the entire body area, wrapping round reserved areas and margin note areas as usual. Content in the remaining space will flow from the top and around any graphics that extend beyond the lowest text in the body area.



To enable remaining space management: in the **Section** dialog box or the **New page** dialog box select the **Manage remaining space** option. A **Remaining space** object will be associated with the section or new page. Use the logic map or the document view to add objects as required.

To switch off remaining space management: in the relevant **Section** or **New page** dialog box clear the **Manage remaining space** option. The **Remaining space** object and all its contents will be removed from the logic map and the publication but will still be available should you wish to enable it again.

To specify remaining space options: right-click on **Remaining space** in the logic map and select **Properties** from the context menu. Specify the options as required.



- Use the **Minimum Height** option to prevent placement of the content if there is less than the specified amount of space available.
- Use the **When an object does not fit ...** options to either ignore all subsequent objects if an object does not fit or, try to fit subsequent objects.
- Use the Active Content processing options to treat Active Content as one or many objects.

To set internal margins:

right-click on **Remaining space** in the logic map and select **Properties** from the context menu. Use the **Margins** tab to specify the internal margins.

Viewing the remaining space area

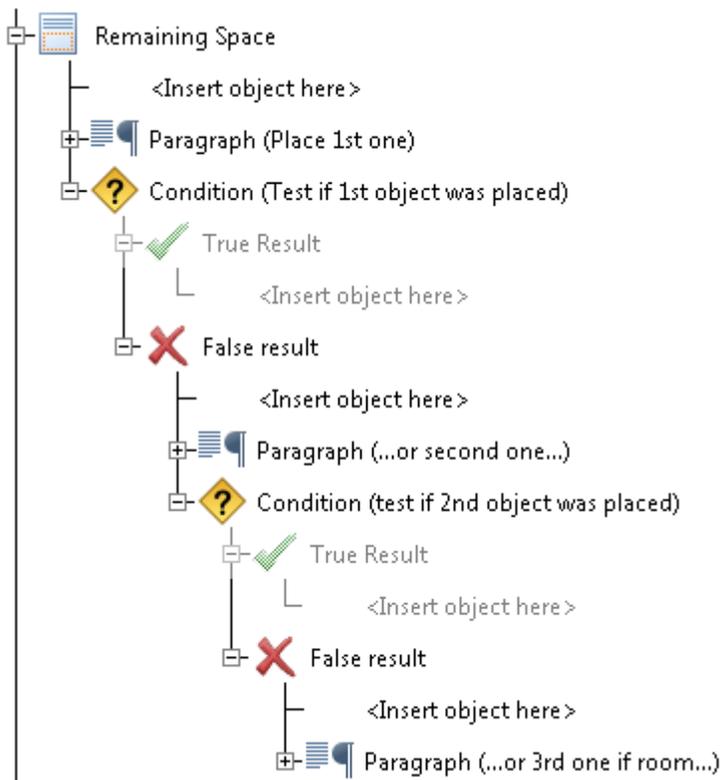
The contents of remaining space area are shown grayed out when the focus is in the main body of the page. If there is not enough room in the editor to show the remaining space area, a small 'plus' icon will be shown to indicate the presence of remaining space. Clicking on this will display the entire contents and the icon will change into a 'minus' sign. Click the 'minus' icon or in the main body to close the remaining space area.

To view remaining space area:

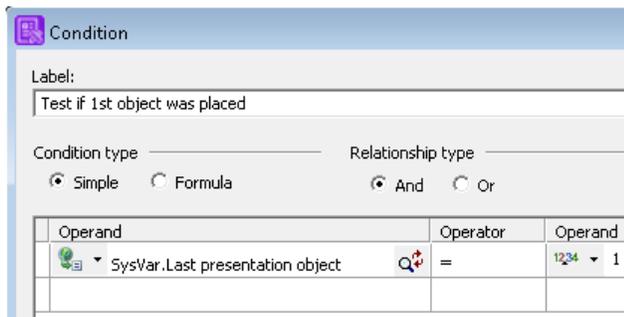
Either click on the **remaining space** object in the logic map – or –click in the remaining space area in the editor (or on the icon).

Controlling which objects are placed

You can use conditional logic – conditions and loops – to control which objects may be placed in the remaining space area. For example, if you only want a single object placed regardless of how many would actually fit in the space available, you can use the variable **Last presentation object** to check when an object has been placed. In this example you would also need to select the **Process subsequent objects** option in the **Remaining Space** dialog box.



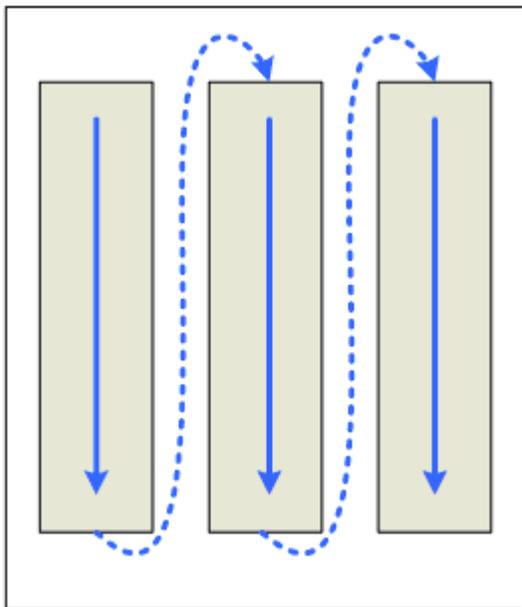
Use the `SysVar.Last` presentation object to test if the previous object was placed. Returns: 1 if the last object was placed. 0 if the last object was not placed and -1 when it is used out of context, for example because of paragraph **keep with next** settings.



Customizing columns

The body area that is used for positioning main document content can be split into multiple columns, up to a maximum of eight in a single layout. Content flows top-to-bottom within a column and when an overflow occurs will continue in the next column in sequence in the layout. The sequence can be either left-to-right or right-to-left. Content will automatically flow onto new pages as required, continuing to use the same column layout. You can switch to a different layout at any time.

See [Customizing columns](#) on page 205 for information on generating simple columns.



- Content flows into the same column layout on the next page and can then switch to a new column layout.

Headers and footers can be applied to columns. These can be used in addition to, or instead of, page level headers and footers – see [Controlling column headers and footers](#) on page 211 . They can also span across all columns. You can specify different header or footer content for regular use, for where content flows into a new column, and for where content flows into a new page.

Note that customized column layouts with no content in the column body area will not be visible on the Designer editor page or Generate output even if the column layout contains content in the column header or footer.

Column layouts

A column layout is used to define the appearance of columns on a page. You can specify any number of layouts for a publication, switching between them as required. Most column layouts will be designed automatically to occupy the available width and height of the body area but you can limit the dimensions if required. A canvas is used to define the overall body size within which the column layout will be designed. This enables you to specify layouts for any page size or orientation, irrespective of the current document's page sizes.

Column layouts can be defined as part of a document section or whenever you switch a column layout in the document logic. A column layout can be reused anywhere in the current document and any other document designs in the publication.

Note: that widow/orphan keep with next and keep together paragraph settings are ignored when switching column layouts.

If no column layout is defined then the default layout is used. This consists of a single column with no headers or footers that fits in the available body space. The default layout cannot be edited or deleted.

Create/edit column layouts

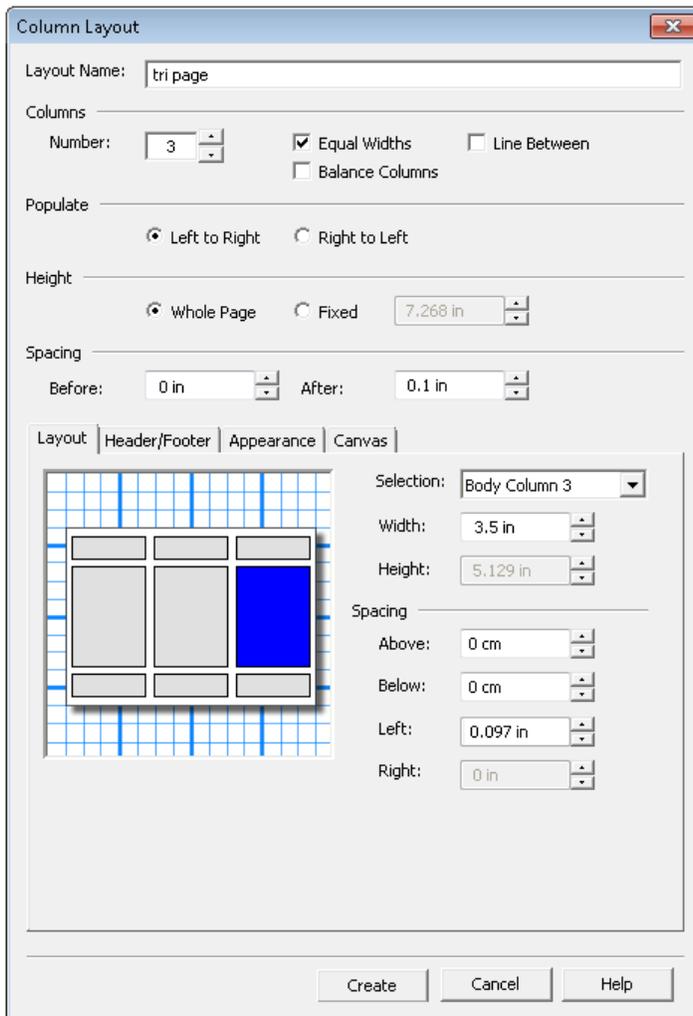
To create or edit a column layout:

1. From the **Insert** menu select **Control objects/Columns/Switch layout**. to invoke the **Switch layout** dialog box (see **To switch column layouts** section that follows);– or –use the **Columns** button in the Logic toolbar. and double-click on the **Switch Column Layout** object in the logic map;– or –invoke the **Section** dialog box, and click the **Columns** tab.

NOTE: The Section Dialog box is invoked either by double-clicking on Section in the logic map or by creating a new section using Insert / Section. See **Sections and page control** on page 198 for more information.

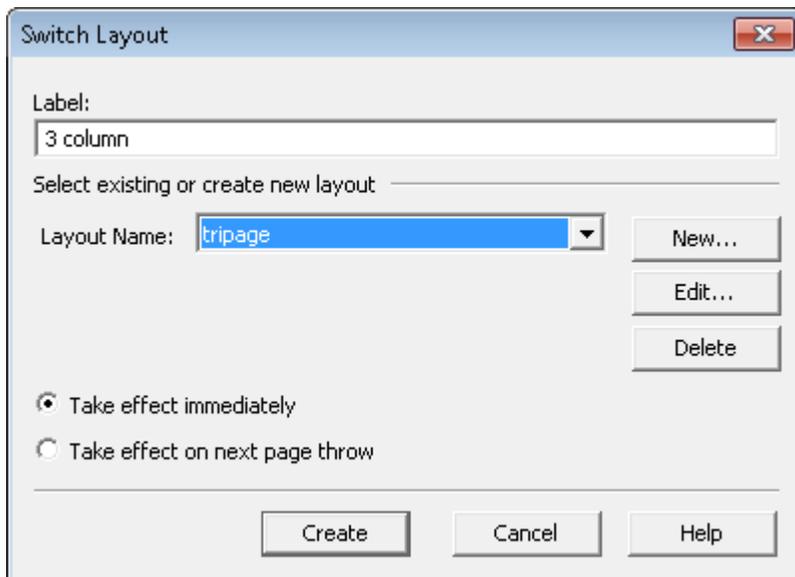
2. Select **New** or **Edit** and use the **Column layout** dialog box to configure the column layouts as required.

Note that any changes made to the column layout will be applied to all instances where the layout is used.



- **Height** options — allow columns to take up the whole of the available space on the page or, you can limit the overall height – (this includes headers and footers).
- **Spacing** options — between column layouts only and is applied before all headers and after all footers. This is not included in the Fixed height if specified.
- **Tabs** — configure the layout of the columns, add headers and footers (see page 142), and set the column boundaries (see page 142). Note that items will be shown in red if they still need to be configured or do not fit the canvas.
- **Selection** drop-down — click on a column or header or footer and configure its Width and Height.
- **Layout/Spacing** — only spacing between column items is specified here. Use the Canvas tab to specify other measurements. Left and right spacing can only be specified when equal widths is not set.

Switching column layouts



- You can use an existing layout or create a new one.
 - The new column layout can be used immediately or when the next page is generated.
1. In the document editor mark the insertion point in the logic map– or –click in the paragraph after which you want to specify the switch.
 2. From the **Insert** menu select **Control objects/Columns/Switch layout**. Select the required layout and choose when the change is to take effect.

To delete a column layout:

invoke the **Section** dialog box, and click the **Columns** tab. Select a column layout and click **Delete**.

Note: If you delete a column layout that is in use elsewhere in the publication then the default column layout will be used instead.

An important note about switching to a new column layout

When switching to a new column layout, the layout's position is based on the previous layout's lowest position. This may be the lowest print position of main flow items in the previous layout or below the previous layout's footers if these exist. For the default layout with a single column, the lowest position would be based on the last paragraph, table, or inline object (e.g., shape, image, chart). The boundaries of existing objects with fixed or anchored position are not considered when positioning the new layout. Note that column headers and footers, like page headers/footers, are not main flow items, and their contents do not wrap around other objects on the page.

Lines between column layouts

A line can be drawn between columns, giving them a visual separation. The line is central between columns and is drawn from the top of the column to the bottom, including any column headers and footers (but not spanning headers/footers). Lines will only be drawn between two columns with content, and will not be drawn next to an empty column. You can customize the color and weight of the line.

To insert a line between columns: in the **Column layout** dialog box select the **Line between** option. Then select the **Appearance** tab and specify the color and weight as required.

Controlling column content

A column break can be used to force subsequent content of the column to start at the top of a new column, generating a new page if necessary. Note that a column break will take precedence over balancing.

To insert a column break:

1. In the document editor mark the insertion point in the logic map– or –position the cursor in the paragraph after which you want the column to break.
2. From the **Insert** menu select **Control objects/Columns/Column break**.

The content of columns on the final page of the layout can be balanced so that they are of similar height.

Note that the quality of balancing may be affected when used in conjunction with:

- tables
- widow and orphan settings
- anchored objects that are placed above the paragraph, or that overflow outside the column
- shapes that are large in relation to the content to be balanced

To balance columns: in the **Column layout** dialog box select the **Balance columns** option. Note that the use of balancing columns can cause a reduction in performance time.

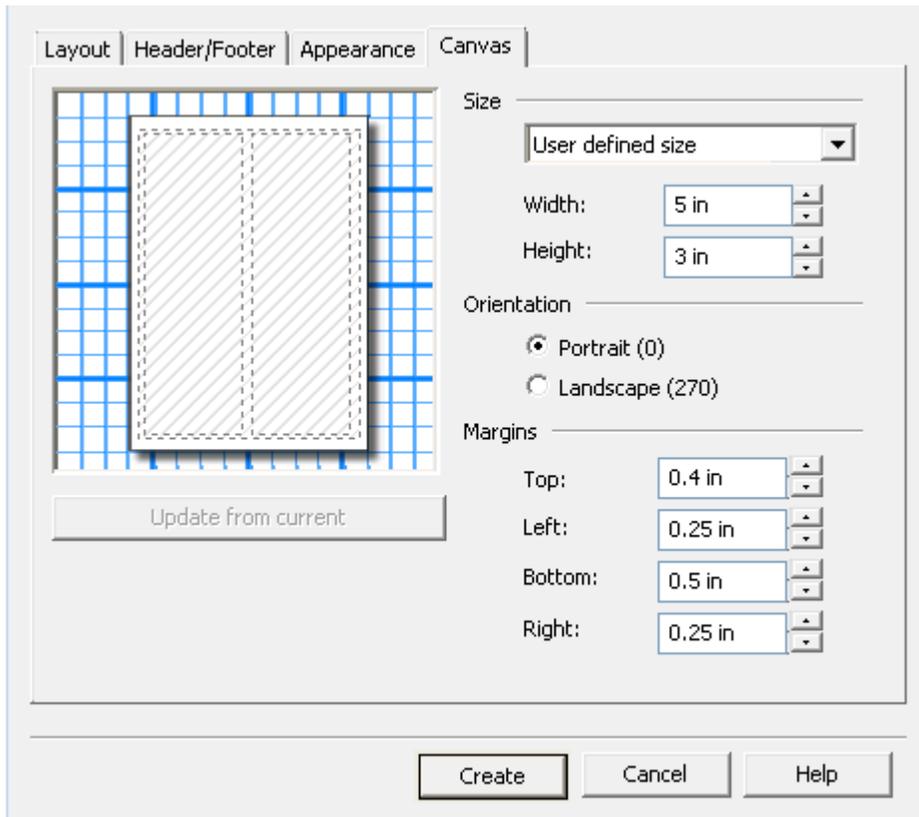
Note: To ensure columns are correctly balanced the following options are not available:

- **Keep with Next**
- **Keep Lines Together**
- **Widow Lines/Orphan Lines**
- **Merge with previous paragraph**

The column canvas

The column canvas allows you to specify the overall size of a column layout and is used for calculating the dimensions of the columns. You can use the size of the current page or specify a different size or orientation.

To specify a column canvas: in the **Column layout** dialog box select the **Canvas** tab and set the options as required.



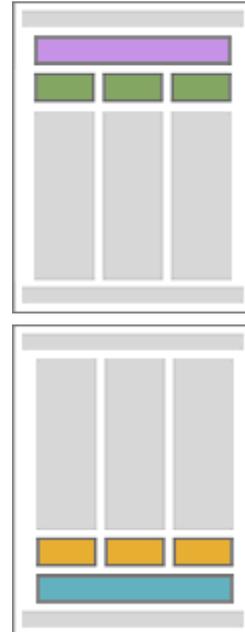
- Choose from a pre-defined size, or define your own by specifying measurements for the canvas.
- Click on the Update from current button to update the canvas using the dimensions of the current page in the editor.
- Margins are included in the size of the canvas and should meet the requirements of your document design.

Controlling column headers and footers

Headers and footers

The headers/footers used in columns can all contain the same content or can use different content as required. This can be changed at any time in the processing, or when the columns overflow either into the next column or into the next page. This would allow, for example, 'Continued' to be placed in all headers except the first one. If required, headers and footers can also span across all columns in the layout, allowing you to use the entire width of the column layout for the content.

Column headers and footers are entirely separate from page headers and footers which can also be used on the same page.



Overflow headers and footers

By selecting overflow header the regular column header will be used in the first column and the overflow header in all subsequent columns. By selecting overflow footer the regular column footer will be used in the last column and the overflow footer in all previous columns.



Page overflow headers and footers

The page overflow header is used in the first column of all pages except the first page and the page overflow footer is used in the last column of all pages except the last page. Other headers and footers are as required by the other options.

Note that if both column and page overflow options are specified, then the page overflow will take precedence over the column overflow.



Legend:

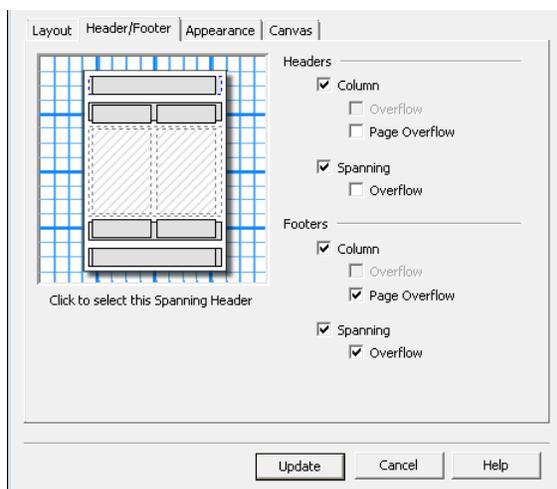
- Grey — regular headers/footers
- Yellow — Overflow footers
- Green — Overflow headers
- Light green — Page overflow headers
- Pink — Page overflow footer

Spanning overflow headers and footers

When **spanning overflow header** is selected the regular spanning header will be used on the first page and the spanning overflow header on all subsequent pages. When **spanning overflow footer** is selected the regular spanning footer will be used in the last page and the spanning overflow footer in all previous pages.

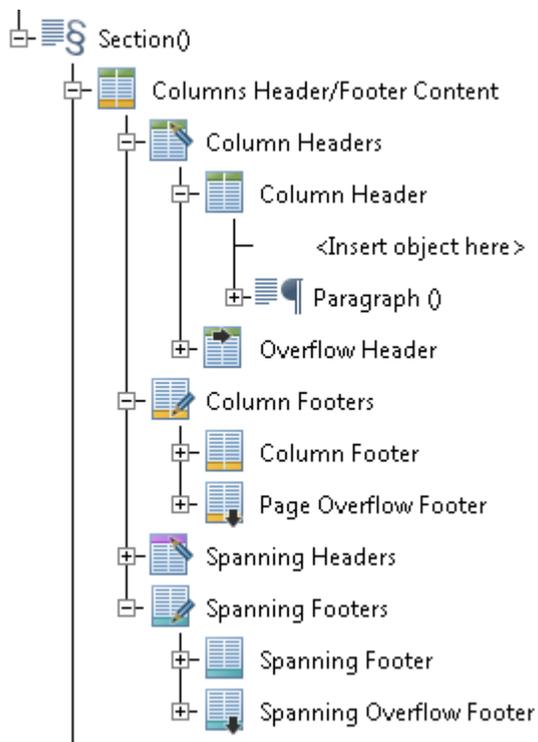
To configure headers and footers:

1. In the **Column layout** dialog box select the **Header/Footer** tab and choose the required options.



2. In the **Layout** tab select the header or footer and specify its size and spacing.
3. Add content using the logic map or in the relevant header or footer in the document view.

In the logic map a group for regular content will always be created when headers/footers are enabled. Additional logic map groups will be created for each overflow option selected. Each group should be separately populated with presentation objects – content is not shared between the groups.

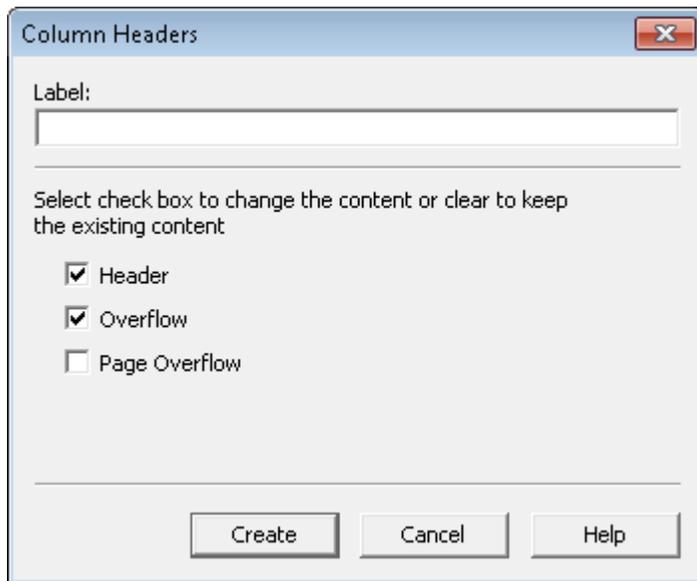


- Use the **Header** and **Footer** check boxes to turn on regular headers and footers.
- The **Overflow** header is used in all overflow columns.
- The **Page** overflow header is used in the first column of all overflow pages.
- The **Spanning** overflow header is used in all overflow pages.
- The **Overflow** footer is used in all columns that cause an overflow – i.e. it is not used in the very last column.
- The **Page** overflow footer is used in the last column of all pages that cause an overflow – i.e. it is not used on the very last page.
- The **Spanning** overflow footer is used in all pages that cause an overflow – i.e. it is not used on the very last page.
- Note that logic map groups are added for each option selected as illustrated.

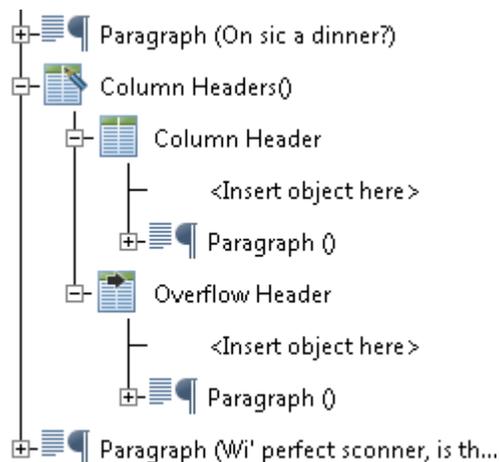
The content of headers and footers can be changed dynamically within the logic. The change will take effect the next time the header or footer is used and will apply to subsequent usage.

To change the content of a column header or footer:

1. In the document editor mark the insertion point in the logic map– or –position the cursor in the paragraph after which you want the new header or footer to be used.
2. From the **Insert/Control objects/Columns** menu select either **Column headers**, **Column footers**, **Spanning headers** or **Spanning footers** and select which types of header or footer you want to change.



3. In the logic map a group for each selected type will be created. These should be separately populated with presentation objects – content is not shared between groups.



4. Logic map groups are added for each option selected but will be grayed out if they were not specified for the current column layout.

Using run-time actions

There are some general actions that affect the processing of a job at production time. They are specified using the **Action** object and can be inserted at most places in the logic.

To create an action: mark the insertion point as required within your document logic and select **Insert/Control Objects/Action** from the menu. Select the required type in the **Action** dialog box:

Abort

Abort the current publication or the production job immediately, and output a message to the standard output device on your host machine.

The screenshot shows a dialog box titled 'Abort'. At the top, there is a text input field labeled 'Message to display'. Below this field, there are two radio button options: 'Job' (which is selected) and 'Publication'.

- Type in a message to be output when the abort is actioned.
- Select whether to abort the current publication only or the complete job.

Print message

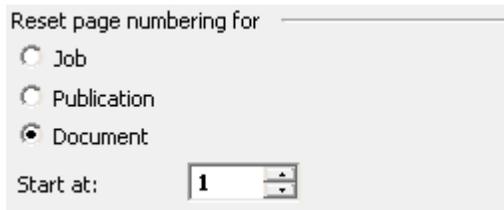
Output a message to the standard output device on your host machine. Use the value button to select a value be used for the message.

The screenshot shows a dialog box titled 'Value'. It has a dropdown menu with 'abc' selected and a text input field. Below the input field is a list of value types: Constant, Data Alias, Interactive Variable, Functions, Parameter, System Variable, and Variable. A 'Help' button is located to the right of the list.

- For information on using values, see [Adding values to objects](#) on page 171.

Reset page number

Reset page numbering at job, publication or document level and specify the page number you wish to start at.



Note that this action causes inconsistent page counts to be generated for Page X of Y numbering and should be avoided if this system variable is present in your document design.

Other actions

For information on using other actions, see the following:

Change border color – see [Text boxes](#) on page 101

Change fill color – see [Text boxes](#) on page 101

Change highlight color – see [Spot color printers](#) on page 455

Change locale – see [About locales](#) on page 311

Change shade percent – see [Text boxes](#) on page 101

Change side thickness – see [Text boxes](#) on page 101

AFP Tagged Logical Element – see [Building the EngageOne template](#) on page 391

PostScript DSC – see [Postscript DSC comments](#) on page 541

Return value – see [Active Content](#) on page 428

Select input tray – see [Controlling the output environment](#) on page 454

Select output bin – see [Controlling the output environment](#) on page 454

Set customer ID – see [Customizing the audit trail](#) on page 330

Set message audit trail – see [Customizing the audit trail](#) on page 330

Set parameter variable – see [Active Content](#) on page 428

Set variable – see [Resolving message data fields](#) on page 325

Simplex/duplex/tumble – see [Controlling the output environment](#) on page 454

14 - Creating conditional logic

Designer has the following conditional logic features:

- Condition queries an expression and executes either a **True** or **False** logic map path depending on the result.
- **Page check** is a specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified.
- **Loop** queries an expression and continuously executes a logic map path until it is no longer **True** or until a **Break Loop** is encountered. If required, **Continue Loop** can be used to restart the conditional logic without processing the remaining objects.
- **Case** selects one of a range of conditional paths when a comparison value is matched.

You may nest any of these objects within one another to form more complex logic as required.

NOTE: You can also use the Show When attribute of individual paragraphs to include them conditionally without the need to use one of the above objects. Refer to [Conditional paragraphs](#) on page 88 for details.

In this section

Conditional expressions.....	218
Condition.....	220
Page check.....	221
Loop.....	222
Case.....	224
Data driven character styles.....	226



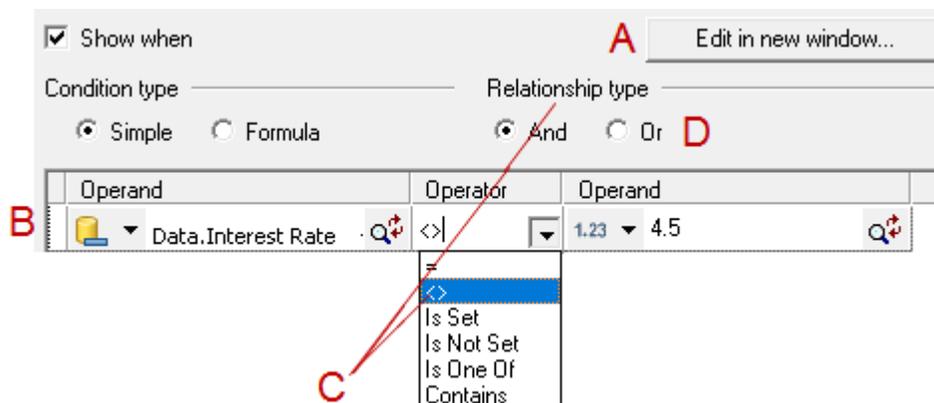
Conditional expressions

The expressions used in Condition and Loop objects can be specified either using a **Simple** method with arguments specified in a structured format or by coding the arguments using a **Formula** expression. Values can be specified as constants or be supplied by data aliases, variables or calls to system or Active Content functions.

The values being queried must be of the same type but in many circumstances you can temporarily reformat the data to ensure consistency. See [Formatting values](#) on page 173 for details.

To code a simple condition click the **Simple** option and enter a comparison argument using the fields provided in the object dialog. Use the operand controls to specify input values for the comparison.

You may add further arguments as required. If the AND operator is specified all arguments must be true for the **True** logic to be executed. If the OR operator is specified the **True** logic will be executed if any of the arguments are true.



Legend:

- **A** - This option allows you to resize the window making it easier to view long or complex formula conditions.
- **B** - Click in the first available **Operand** field to select a value type and name. See [Adding values to objects](#) on page 171 for details.
- **C** - The operators available in a condition vary depending on the value type of the operand.

Note:

- Two particular comparisons, **Is Set** and **Is Not Set**, query the status of a data alias (to see if it is available in the current data set) or a variable (to see if it is currently in scope).
- **Is One Of** and **Contains** are typically used when processing checkbox selections in your document logic. Note that these operators are available only if both the left and right operands are strings.

- **Is One Of** - returns true if a case-sensitive match for the left operand string is found in the right operand choice list. For example "Option4" <Is One Of> "Option1 Option2 Option4" would be true.
 - **Contains** - returns true if a case-sensitive match for the right operand string is found in the left operand choice list. For example "Option1 Option2 Option4" <Contains> "Option4" would be true.
- **D** - Statements in simple conditions can have either an **And** or an **Or** relationship. You cannot use both types unless you code a Formula.

To code a formula click the **Formula** option and enter the appropriate syntax. For detailed information see [Appendix B Formula expressions](#) on page 716.

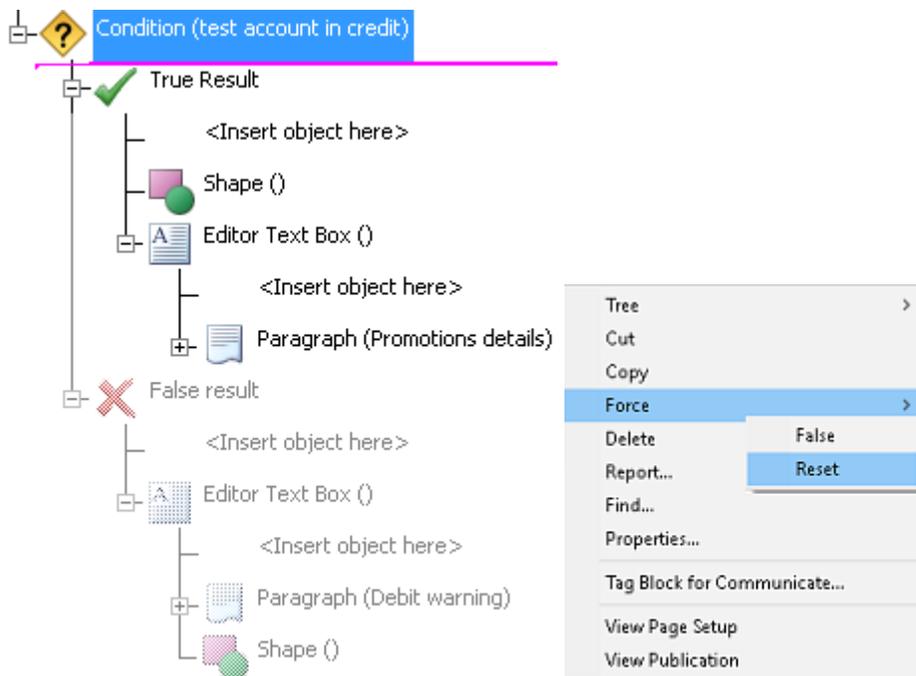
Condition

A condition object queries an expression and executes either a **True** or **False** logic map path depending on the result.

To create a condition:

1. Mark the required insertion point within the logic map.
2. Select **Insert/Control Objects/Condition/Condition** from the menu.
3. In the **Condition** dialog select either **Simple** or **Formula**.
4. Create the required query. See [Conditional expressions](#) on page 218 for details.
5. Click **Create** and populate the logic tree.

In the logic map a condition object is automatically created with True and False paths. Add the required logic to each.



To help when designing a document, you can force the True or False path to be active. This enables you to see the effect of each path in the condition. You can reset all paths in the document by selecting **Insert/Control Objects/Condition/Reset All**.

To delete a simple condition argument: in the **Condition** dialog box select the argument you want to remove and press **Delete**.

Page check

Page check is a specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified. You can perform a simple check to see if the current page number is odd or even or you can reference one or more specific pages either individually or within ranges.

As with a regular condition, a **True** or **False** logic map path is executed depending on the result of the check.

To use a page check:

1. Mark the required insertion point within the logic map.
2. Select **Insert/Control Objects/Condition/Page Check** from the menu.
3. In the **Page Check** dialog select either **Job**, **Publication** or **Document** to indicate the scope of the check.

In each of these, the page check assumes a starting point of page 1.

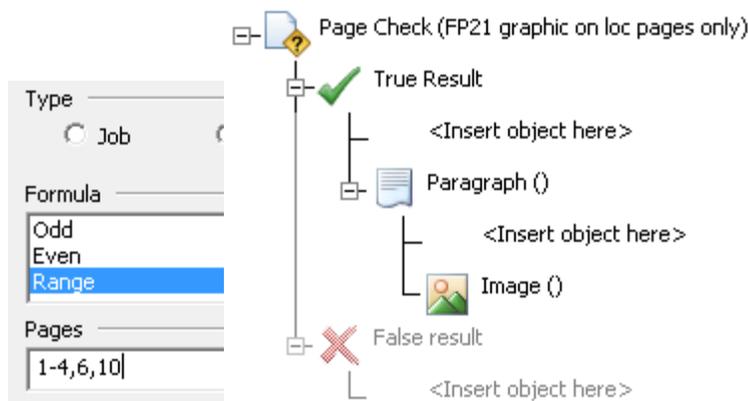
4. Specify the page formula to be matched.

Selecting **Odd** will mean the True path is executed for page 1, 3, 5, etc. Similarly, pages 2, 4, 6, etc. will execute the True path when **Even** is selected.

If **Range** is selected, indicate the page numbers that will cause the True path to be executed in the entry field, for example: 1, 7, 111-4, 81-10, 20+

5. Click **Create** and populate the logic tree.

In the logic map a page check object is automatically created with True and False paths. Add the required logic to each.



Loop

A loop object queries an expression and continuously executes a logic map path until it is no longer **True** or until a **Break Loop** is encountered. If required, **Continue Loop** can be used to restart the conditional logic without processing the remaining objects. Loops can be nested within each other.

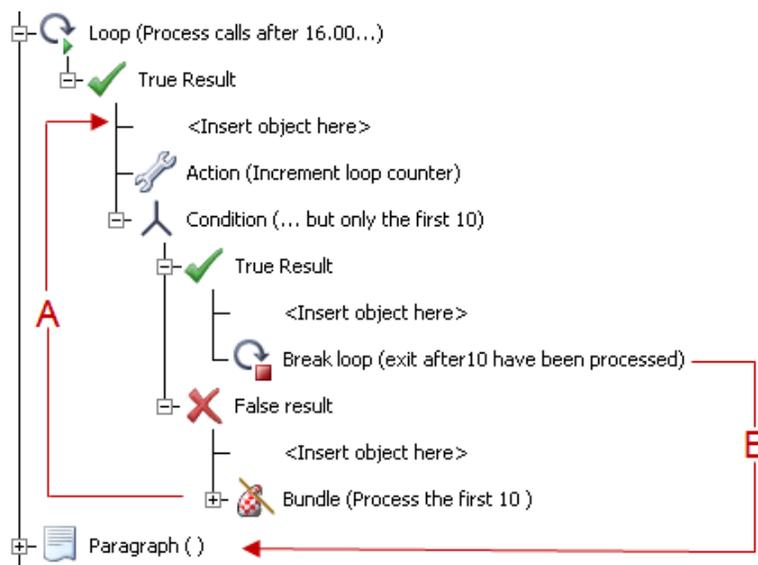
If the conditional expression does not evaluate to true when the loop object is first encountered the True path will not be executed.

To create a loop:

1. Mark the required insertion point within the logic map.
2. Select **Insert/Control Objects/Loop/Conditional Loop** from the menu.
3. In the **Loop** dialog box select either the **Simple** or **Formula** condition type and set the **Maximum Loop Iterations**.
4. Create the required query. See [Creating conditional logic](#) on page 217 for details.
5. Click **Create** and populate the logic tree.

In the logic map a loop object is automatically created with a True path. Add the objects that you want to be processed while the conditional expression is true.

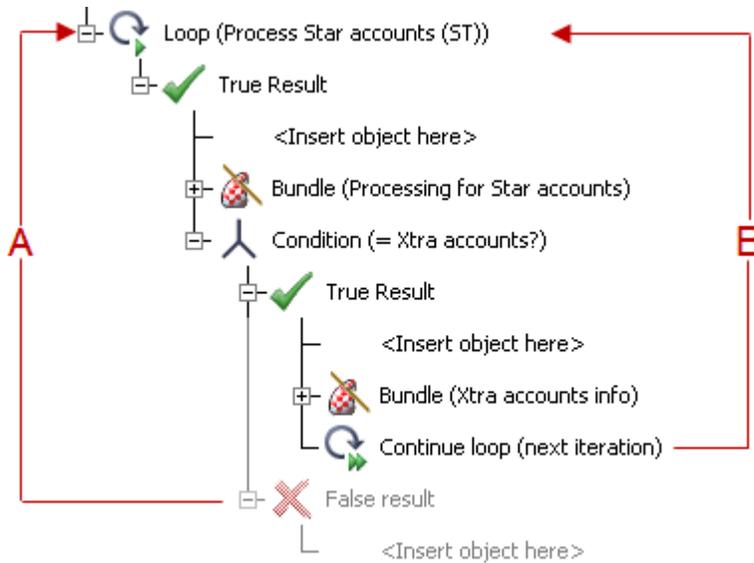
To break out of a loop: mark the insertion point within the True path and select **Insert/Control Objects/Loop/Break** from the menu.



Legend:

- **A** - Return to the top of the loop
- **B** - Do not process any more of the loop

To include specific processing within a loop: you will typically want to use another conditional object within the True path to evaluate a specific status and include any specific processing. On the appropriate path within the embedded object mark the insertion point and select **Insert/Control Objects/Loop/Continue** from the menu.



Legend:

- **A** - Repeat until the loop condition is false
- **B** - Return to the top of the loop

Case

A case object selects one of a range of conditional paths when a comparison value is matched. The value is defined within the main case object and is compared with the values defined for the **Case Test** objects within the case group. The logic associated with the first case test that matches the comparison value is executed. If no value is matched an **Otherwise Result** logic path is executed.

Values can be specified as constants or be supplied by data aliases or any other value object. All values must be of the same type but in many circumstances you can temporarily reformat the data to ensure consistency. See [Formatting values](#) on page 173 for details.

To create a case group:

1. Mark the required insertion point within the logic map.
2. Create the case object and supply the comparison value.

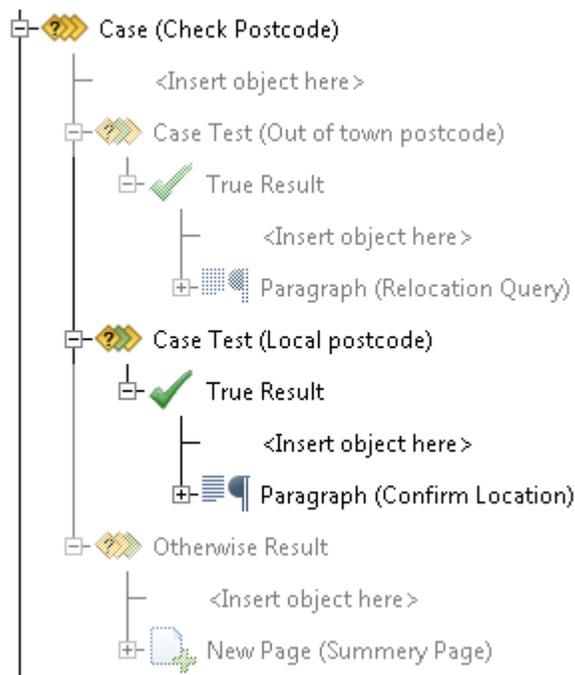
Select **Insert/Control Objects/Condition/Case** from the menu.

In the **Case** dialog box give the object a reference Label and provide the value used for the comparison, see [Adding values to objects](#) on page 171 for details.

3. Create case tests.

Mark the insertion point on <Insert object here> and select **Insert/Control Objects/Condition/Case Test** from the menu. The expanded case test object will contain a place holder and a True Result group.

Resolve the place holder with an object containing the value that will be compared against the main case value.



Add the logic to be executed when the value is matched to the **True** path

Repeat for each case test required.

4. If required, add logic to the **Otherwise** path that will be executed if no case test has been matched.

Data driven character styles

You can use style lists to change the presentation of text dynamically according to conditional logic.

Style lists are objects that you can place within your document logic and are themselves containers for one or more styles.

A style itself contains the actual text formatting instructions. It always specifies a font and point size and can also include font decoration settings such as color, bold, underline and so on. A style can be applied to most text elements including data aliases, address blocks and charts.

NOTE: Character styles are not applied cumulatively, i.e. you cannot apply two or more styles to the same text and get the joint effect of both. Only the last style applied will be used. Therefore each style should have all the settings required to present the text according to your needs.

At production time the active style list is the last such object to be encountered in the logic. Before a style list is encountered no styles are used and all text elements are presented using their regular settings. Once a style list is encountered any text that follows the object within the logic will adopt the settings of the styles it contains wherever a style name from the list has been applied. Subsequent style lists within the logic may amend some or all of the named styles. If a style name does not appear in a style list its settings remained unchanged from the previous time a style list included it.

Style lists are not stored in the repository as independent files. However, you can reuse the logic you set up to control styles by including it in an Active Content file and including this in any design where you need the same functionality.

You will typically want to use conditional objects to choose when to invoke particular style lists. Remember to make sure that text to be influenced by a style list appears after the list objects within the document logic.

In the following example a case object is used to create the necessary conditions for dynamically changing the character styles.

Why not take advantage of the great rates available to you **RIGHT NOW** and upgrade to a **Big Bank Gold Card**.

Benefits include:

0% on purchases for 6 months.
0% on balance transfer for 6 months.
After the introductory period you will be charged at the typical rate of **15.9% APR**.

New enhanced security features.
No annual fee.

Why not take advantage of the great rates available to you **RIGHT NOW** and upgrade to a **Big Bank Platinum Card**.

Benefits include:

0% on purchases for 9 months.
0% on balance transfer for 9 months.
After the introductory period you will be charged at the typical rate of **12.9% APR**.

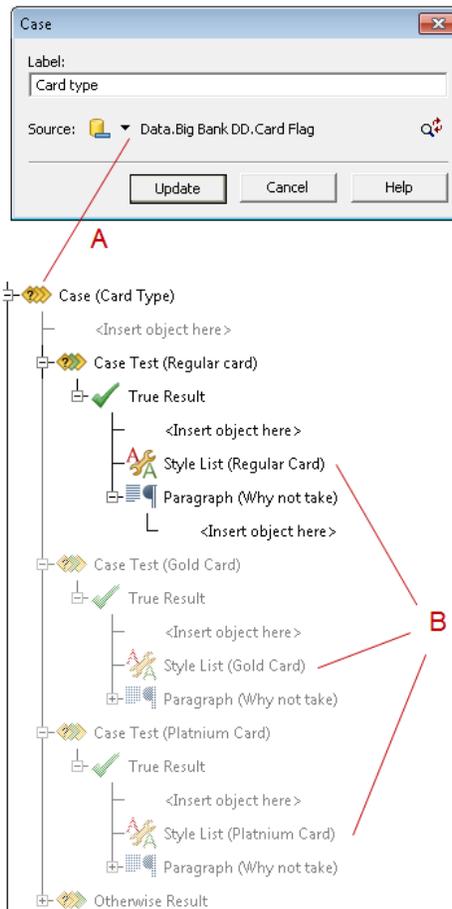
New enhanced security features.
No annual fee.

Why not take advantage of the great rates available to you **RIGHT NOW** and upgrade to a **Big Bank Titanium Card**.

Benefits include:

0% on purchases for 1 whole year.
0% on balance transfer for 1 whole year.
After the introductory period you will be charged at the typical rate of **10.9% APR**.

New enhanced security features.
No annual fee.



Legend:

- **A** - The Case object is based on the **Card Flag** data element in the input data.
- **B** - As the **Card Flag** data element changes with the publication data sets, the style list associated with each case test will change the formatting of the font.

To set character styles dynamically:

1. In the document's logic map insert a Case object and create a case group. For detailed information on case groups, see **Case** on page 224 for further information. Note that you can also use other Conditional Logic, such as Page Checks, Conditions and Loops.
2. For each case test, mark the insertion point in the logic map and select Insert/Control Objects/Set Style to place a Style List object.
3. Double-click the initial Style List object in the logic map and the Style List dialog box is displayed.
4. Click Add to display the Set Style dialog box.
5. Enter a name (to a maximum of 32 characters) for the style created in the General tab and then select the Font tab to set the attributes for the character style. Repeat the step for each character style required in the document. Note that the initial Style List object is used to set up the character styles that will be dynamically changed by each case test.

6. For each subsequent Style List object repeat steps 5 to 7, however, in the General tab of the Set Style dialog box select one of the existing character styles from the list and then select the Font tab to set its attributes.
7. Apply the character styles to the necessary text. [Data driven character styles](#) on page 226

Using character styles

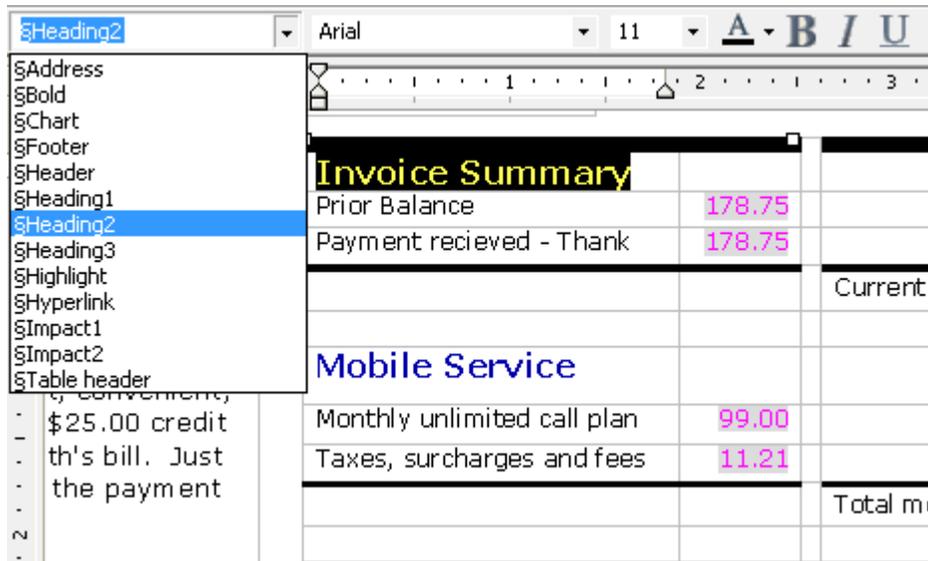
After creating style lists and character styles you can apply them to text, as well as edit and remove them from Style Lists.

1. To apply a character style to text, highlight the appropriate text in the editor and select the character style from the list in the toolbar– or –highlight the text and right-click it to display the context menu. Select **Font** to display the **Font** dialog box. Click the **Style** option and select a character style from the list.
 - a) All the character styles contained in the repository are included in the Styles drop-down list. This includes styles from other publications. You must therefore ensure that you use character styles that are available to your document for the formatting to take effect.
 - b) To modify an existing style. double-click the Style List object in the logic map
2. Select a character style and click Edit to display the Set Style dialog box.
3. Choose the **Font** tab and click **Select** to make the necessary changes to the font attributes and click **OK**.
4. Select a different character set from the list, if necessary, and click **Update**.
5. To remove a character style from the style list:
 - a) Invoke the **Style List** dialog box and click **Remove** to display the **Remove Style** dialog box.
 - b) Select the character style to remove from the drop-down list.
 - c) Select either **Remove from List** or **Remove from List and Repository** and click **Update**. If you select **Remove from List**, the character style name remains in the repository and is still included in style lists for other objects.
6. To remove a character style from text highlight the text that you want to remove the character style from change the font attributes as required using the toolbar options.

Setting character styles dynamically

1. In the document's logic map insert a Case object and create a case group.
For detailed information on case groups, see [Case](#) on page 224. You can also use other Conditional Logic, such as Page Checks, Conditions and Loops.
2. For each case test, mark the insertion point in the logic map and select **Insert/Control Objects/Set Style** to place a Style List object.

- Double-click the initial **Style List** object in the logic map and the **Style List** dialog box is displayed.
- Click **Add** to display the **Set Style** dialog box.



- Enter a name (to a maximum of 32 characters) for the style created in the **General** tab and then select the **Font** tab to set the attributes for the character style.

Repeat the step for each character style required in the document. The initial Style List object is used to set up the character styles that will be dynamically changed by each case test.

- For each subsequent **Style List** object repeat steps 5 to 7, however, in the **General** tab of the **Set Style** dialog box select one of the existing character styles from the list and then select the **Font** tab to set its attributes.
- Apply the character styles to the necessary text.

15 - Functions

A number of functions are available within the Designer environment. This chapter gives detailed information on using the built-in functions.

In this section

About Designer Functions.....	231
Address block.....	231
Arithmetic.....	234
Barcodes.....	238
Concatenation.....	254
Lookup tables.....	255
Substring.....	256
System variables.....	257



About Designer Functions

Designer includes the following functions:

Address block:	this function automatically formats address data based on the chosen country. The resulting address structure is output and positioned as a single object
Arithmetic:	performs a calculation on two input values and outputs the result.
Barcodes:	converts strings and numbers to the necessary sequence of lines required for the chosen barcode type.
Concatenation:	this function enables you to join text values from two or more objects to form a single string.
Lookup table:	enables you to replace predefined keywords with text strings at production time.
Substring:	this function extracts one or more consecutive characters from within a text string.
System variables:	can be used to access run-time statistics and information from the productions environment.

Address block

The address block function automatically formats name and address data into a style that is the standard for a named country. The resulting address structure is output and positioned as a single object.

The output lines of the address block are automatically aligned vertically and the number of lines actually placed is adjusted to compensate for empty input values. The output lines are placed within a notional outline that allows enough horizontal space to accommodate all possible address lines. You can choose to align the block either at the top or bottom of the outline space.

The font and line spacing used for the output lines is set for the entire object although you can override the settings of individual components as required. While you can change the font style, size and decoration for the text of an address block, you cannot change the justification or add bullets and numbering.

In terms of positioning, rotation, color, line and fill attributes and text runaround properties, an address block behaves exactly the same as a text box or other shape. See [Using graphics](#) on page 96 for details.

To insert an address block:

1. In the document editor mark the insertion point in the logic map– or –position the cursor in the paragraph to which the address block is to be anchored.
2. From the **Insert** menu click **Presentation Objects/Address Block**.

- The dialog is populated with standard address settings for the selected country.
- Toggle any optional specific address components as required.
- Up to 5 additional address lines can be added.
- An example as to the format of the address is presented.

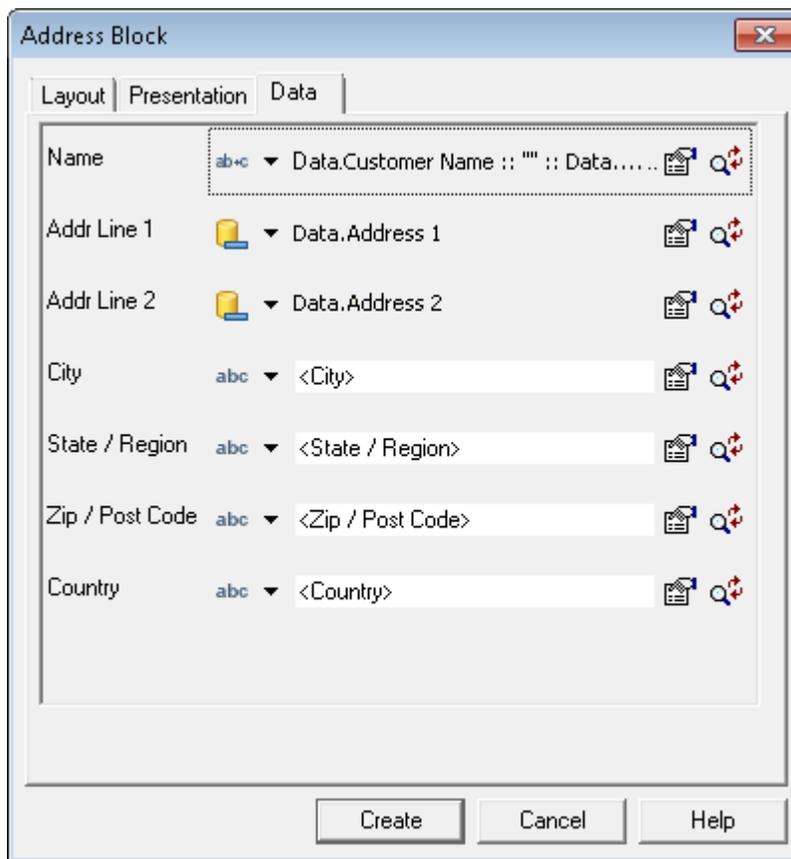
3. In the **Address Block/Layout** dialog box use the **Format** list to select the country for which the address is to be formatted.
4. On the **Presentation** page select the font, text alignment, direction and line spacing to be used for output lines unless overridden.

Line spacing is either:

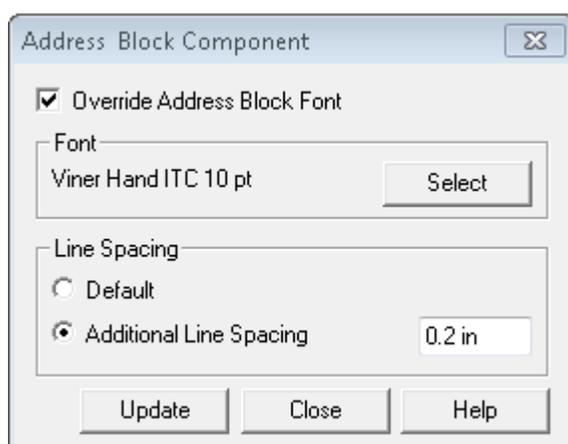
Automatic – the line spacing value contained in the default font

Absolute – a fixed gap between lines that you define. This spacing will be applied regardless of the font size.

5. On the **Data** page resolve all the address component fields with values.



6. Click on the hand icon to override presentation settings for a component and customize its settings in the **Address Block Component** dialog.



7. Select the **Create** button to place the object on the page.

Once placed you can reposition it and alter its font and properties as required.

To change the font of a component in an address block: highlight the component and select the required font, size, color etc.

To change the properties of an address block: right-click on its border and from the context menu select **Format Shape**. Use the dialog box to change the properties. See [Using graphics](#) on page 96 for more information.

Arithmetic

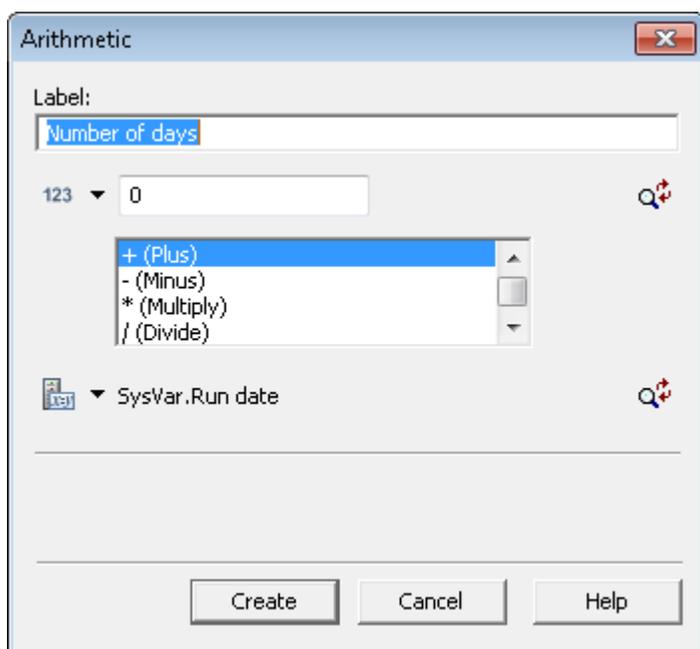
An arithmetic function performs a calculation on two input values and outputs the result. Each value can be of any type, but must be cast to the same type as the result, see [Working with values](#) on page 170 for details.

To create an arithmetic function:

1. From the **Insert** menu select **Field/Arithmetic** and proceed to step 3– or –create an object to receive the output.

You can update a number or date variable with the function output or use it as input to any other object that requires a number or date value. See [Working with values](#) on page 170 for details.

2. When specifying a value for the object select **Functions/Arithmetic** as the type.
3. In the **Arithmetic** dialog box specify the calculation.



- Provide two values for the calculation.
- Select an operator from the list.
- Cast the value when it does not match the expected type.

Arithmetic operators

Plus	Valid for numbers, integers and dates, the result will be the addition of the values – i.e. A+B.
Minus	Valid for numbers, integers and dates, the result will be the subtraction of the second value from the first value – i.e. A-B.
Multiply	Valid for numbers and integers, the result will be the multiplication of the values – i.e. A*B.
Divide	Valid for numbers and integers, the result will be the division of the values – i.e. A/B. Note that if both values are integer, then the result will be integer, i.e. truncated; however, if at least one of the values is a number, then the result will also be a number.
Modulo	Valid for integers only, the returned value will be the remainder that results from the modular division of the values – i.e. (A/B). For example 10/3 gives a remainder, or modulo, of 1.

Arithmetic operators

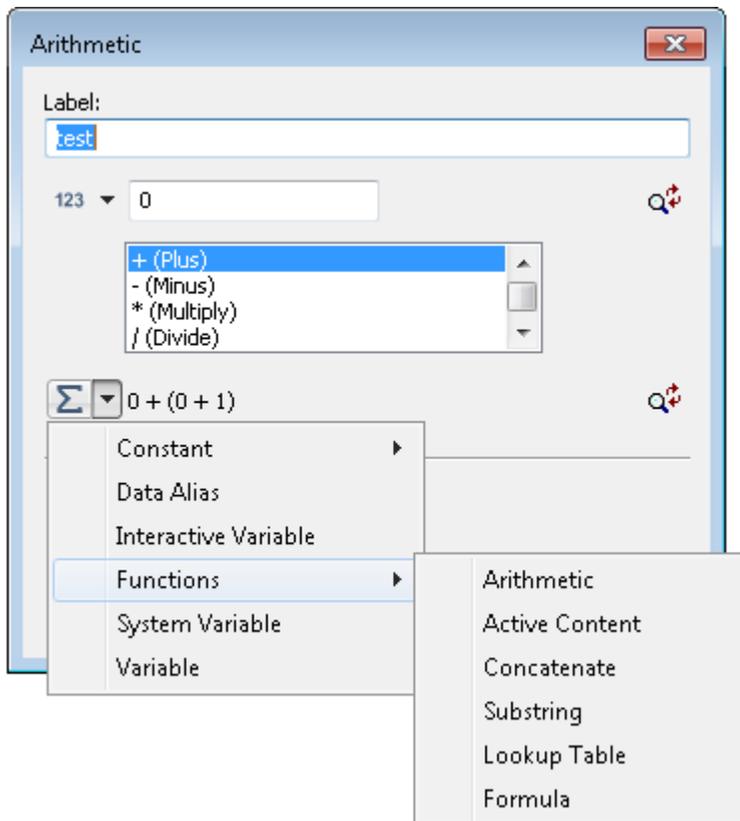
Bitwise AND'

Valid for integers only, 'AND' compares the two values bit by bit. If every bit in position 'x' of both values is 1, bit 'x' in the result is set to 1, otherwise 0.

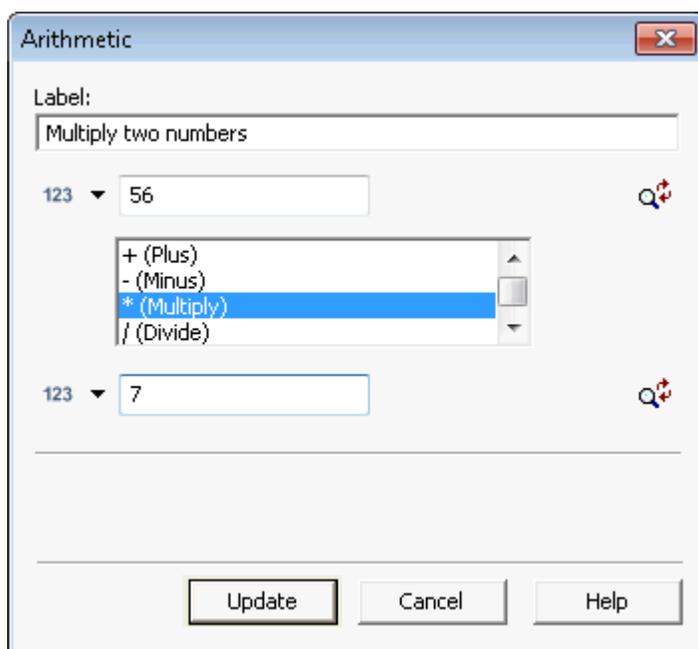
Bitwise OR'

As above except that if a bit in position 'x' of either of the values is 1, bit 'x' in the result is set to 1.

You can nest arithmetic functions to perform more complex equations if required. The evaluation sequence will follow the order in which the objects are nested, from top to bottom as in the following example.



- To build more complex calculations first create an initial arithmetic function, then specify one or both parameters as another arithmetic.
- This example will calculate $1234 / (56 \times 7)$



Date calculations

Calculations can be performed on date values with the following restrictions:

- The operator used must be either addition or subtraction.
- The first value must contain a date value.
- The second value indicates the time to add or subtract from the date. This can be specified as:

an integer

– of the number of days

a string

– indicating the number and type of time units

a date

– for subtraction only.

Examples of time units:

2W = two weeks

5M = five months

3Y = three years.

Barcodes

The barcode function converts strings and numbers to the appropriate sequence of lines that are required for the selected barcode. EngageOne Designer includes support for **Standard**, **Font scaling** and **Line drawing** barcode types. The barcode fonts are automatically installed in **Designer** and available for all drivers and variants. See below for the list of barcodes and their settings.

Standard and **font scaling** barcodes generate textual output and make reference to special fonts within the output datastream produced by **Generate**. As with all other text output the appropriate fonts will be automatically included in the resources when published. By default these and all other fonts are embedded in the output datastream, but if you choose not to embed you will need to extract the resources from the **HIP** file and make them available to your output device. Refer to **Running a publishing task** on page 602 for details of font embedding options and to the **Production Guide** for information on using **DOC1RPU** to extract resources from a **HIP** file.

Line drawing barcodes are drawn without the use of fonts; native drawing methods are used instead, such as lines or rectangles.

Note: if you intend using barcodes in PDF output it is recommended to use line drawing barcodes where supported for the relevant barcode type.

For Standard or Font scaling barcodes of PDF417 barcode type, where the barcodes have been created in Designer versions prior to 6.6 Service Pack 6, the Designer Client user will be required to manually update the associated Windows barcode fonts to ensure that the latest PDF417 barcode font versions are used as follows:

1. Close Designer and uninstall the previous PDF417 barcode windows fonts in the Windows Font folder;
 - PD2w Regular
 - PD3w Regular
 - PD4w Regular

Note: It may be necessary to restart the computer to allow this.

2. Clear Designer local file cache.

Note: Tools \ Designer Preferences \ Empty Local File Cache on startup

3. Clear the resource cache used for EngageOne publishing.

Note: Tools \ Resource Cache Management \ Printer Identifier \ Clear Cache

4. Publish HIP for Generate or Template ZIP for EngageOne as appropriate, and execute Generate to obtain updated barcode font output.

Creating a barcode

1. Place the cursor in the document area where you want the barcode.
2. Select **Insert/Presentation Objects/Barcode** and the **Barcode** dialog box is displayed.
3. Select the **Standard**, **Font scaling** or **Line drawing** mode of operation.
4. Select the required barcode and resolve any barcode specific settings.

To change the properties of a barcode: right-click on its border and from the context menu select **Format shape**. Use the **Barcode** dialog box to change the properties. See [Using graphics](#) on page 96 for more information.

The Barcode dialog

The screenshot shows the 'Barcode' dialog box with the following settings:

- Label:** EOV Barcode
- Value:** Data.id_dict.purl2
- Code page:** Windows 1252@ISO Latin 1
- Mode:**
 - Standard
 - Font scaling
 - Line drawing (AFP, eHTML, PCL, PDF, Postscript & VPS)
- Barcode group:** QR Code barcode
- Type:** EngageOne Video
- Presentation:**
 - 90 Degree Orientation
 - Color:** Black
- Options:**
 - Mode:** Binary
 - Security:** 3
 - Scale:**
 - Not applicable
 - Not applicable
 - Width of single cell:** 0.05 in
 - Not applicable
 - Not applicable

Buttons at the bottom: Create, Preview, Cancel, Help.

- **Standard** barcodes can be used with any output device. **Font scaling** and **Line drawing barcodes** have limited output device compatibility.
- **Preview** displays the barcode in a separate window based on the **Value** and **Options** selected.

Linking to an EngageOne Video project

Use the EngageOne Video type to place a font scaling or line drawing QR Code barcode at the appropriate position in the document. The QR Code represents a Personalized URL (PURL) that links to an EngageOne Video project. In most cases the PURL will be generated by the system and provided through a data field, so you should select the appropriate data alias as the barcode value. Refer to [Working with EngageOne Video design objects](#) on page 347 for introductory information. Note that at design-time the PURL value will be whatever default has been applied to the interactive data alias or sample data used. If the length of the PURL changes in production the size of the generated QR Code may be different.

Previewing barcodes

Previewing barcodes

You can choose to preview a barcode to verify that it has been configured correctly and to see how it will be presented in the document.

Note: Only font scaling and Line Drawing barcodes can be previewed. The option is disabled for standard barcodes.

To preview a barcode: configure the barcode as required, refer to [Creating a barcode](#) on page 240 . If it contains an invalid value a warning message box is displayed, otherwise you will be able to view the barcode.

Using Carriage Return/Line Feed characters in barcodes

The delineation of your data configured in the data format may not accommodate the input of Carriage Return and Line Feed characters. In this scenario, if you wish to use these characters within a barcode value, the Paragraph Separator (Unicode value U+2029) and Line Separator (Unicode value U+2028) characters can be used. These characters will be replaced with Carriage Return and Line Feed characters in the output.

If you wish to input these characters within the Barcode dialog in Designer, it is possible to paste a Line Separator character into a value field, however you cannot do this for the Paragraph Separator. If a Paragraph Separator is required then use a lookup table created with a text editor that supports entry of the character. Note that Windows Notepad is not such an editor. The lookup table must be saved in Unicode encoding (for example, UTF-8), and this must be specified when publishing. Refer to [Lookup tables](#) on page 255 for further information.

Note that if the Paragraph and Line Separator characters are to be used in the input data then the encoding scheme for the field in which they appear must also be set to Unicode.

A Line Separator character can be entered in the editor by pressing **Shift+Return**. This can then be selected by pressing **Shift+Left** and copied to the clipboard. Any Unicode value can be entered into Microsoft Word using the **Alt+X** command. For example, to enter a Paragraph Separator, type U+2029 followed immediately by **Alt+X**, then press **Shift+Left** to select it.

Barcode reference

The sections that follow presents detailed information about supported barcode types.

Standard barcode

Standard barcodes can be used with any output device. To be compatible with each output device there are a limited number of barcodes and associated options available when using standard mode. Apart from changing the density, to enable other scalability options you must use font scaling barcodes.

Standard barcode settings

Label	Enter an optional name for the barcode object that will appear in the logic map.
Value	Enter the value you want to use for the barcode. See Adding values to objects on page 171 for more information. It is important to note that barcode values can be a maximum of 1024 characters. Longer values will cause Designer and Generate to issue warning messages and the value will be truncated when processed by Generate.
Mode	Standard mode barcodes can be used with any output device. Refer to Font scaling barcodes on page 246 and Line drawing barcodes on page 250 for details on these barcode types.
Barcode group	Select Universal types or one of the 2D barcode groups to populate the Type list.
Type	If Universal types has been selected, this list is populated with standard barcodes, otherwise with the 2D barcodes available for the selected barcode group .
Scale	Density can be set to Low, Medium or High.
Presentation	These are cosmetic options that affect the presentation of the barcode when it is placed on the page. 90 degree orientation rotates the barcode 90 degrees clockwise. Use the Color option to change the color of the barcode from the default black.

Standard barcode settings

Specific barcode options

PDF417

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page. This is only available when the mode is set to **binary**.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread.

Barcode aspect is the height to width ratio of the number of code words per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode.

Start mode indicates the type of data most likely to occur at the start of the data to be encoded.

Font aspect controls the aspect of the font to be used to present the barcode.

Datamatrix

Note: Datamatrix barcodes comply to the ECC 200 standard.

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.

Mode determines the shape of the final barcode.

Security defines the amount of additional codewords in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Encoding allows the barcode data to be encoded using the **C40** scheme. If the **Automatic** option is selected then the encoding which creates the most compact symbol will be used.

QR Code

Code page – if you need to include non-Latin characters in the barcode, or the data is Base64 encoded this allows you to select an alternative code page. This is only available when the mode is set to **binary**.

Mode determines which type of data will be used with the barcode.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Standard barcode settings

Aztec

Mode determines the characteristics of the final barcode. Choose from:

Normal – allows you to set **Security** to any percentage in the range 1 – 99%.

Compact – use **Security** to specify the number of layers used to contain the symbol's information in the range 1-4. Error correction data fills unused space in the specified number of layers.

Full range – as for compact mode but in the range 1-32.

Security identifies the percentage of the symbol to contain error checking data. The range is determined by the mode. A value of 0% will produce the symbol with the default amount of error correction (i.e. 23% + 3 codewords). The higher the security level the greater the number of layers required to contain the symbol and thus its overall size.

Font scaling barcodes

Font scaling barcodes provide a number of scalability options and can be used with most output devices, except **Linedata** and **HTML**. Note that for drivers with raster fonts the original font is converted to a bitmap font.

Font scaling barcode settings

Label	Enter an optional name for the barcode object that will appear in the logic map. It is important to note that barcode values can be a maximum of 1024 characters. Longer values will cause Designer and Generate to issue warning messages and the value will be truncated when processed by Generate.
Value	Enter the value you want to use for the barcode. See Adding values to objects on page 171 for more information. It is important to note that barcode values can be a maximum of 1024 characters. Longer values will be truncated and a warning message displayed by Generate if this limit is exceeded.
Mode	Font scaling mode barcodes enable you to change the scale settings. Refer to Standard barcode on page 243 and Line drawing barcodes on page 250 for details on these barcode types.
Barcode group	Select Universal types or one of the 2D barcode groups to populate the Type list.
Type	If Universal types has been selected, this list is populated with font scaling barcodes, otherwise with the 2D barcodes available for the selected barcode group . If QR Code barcode has been selected you have the option to select the EngageOne Video type. See Linking to an EngageOne Video project on page 241 for further details.

Font scaling barcode settings

Options

These options are used with certain font scaling barcodes. The enabling of options is barcode type dependent.

Optional Check Digit automatically places a check digit, also known as a checksum, within the barcode. The check digit is used to verify that the value of the barcode has been entered correctly.

Wide bars can be used to increase the width of individual bars within a barcode.

Scale

These options are used to set the size of the barcode.

Height/width ratio is used to set the height to width ratio of the codewords used in each row of the barcode.

Weight reduction enables the font weight of the barcode to be changed.

The **Point Size** option enables the size of the barcode font to be set independently of the point size of the paragraph the barcode is associated with.

Presentation

These are cosmetic options that affect the presentation of the barcode when it is included on the page. Select **90 degree orientation** to rotate the barcode 90 degrees clockwise. Use the **Color** option to change the color of the barcode from the default black.

Specific barcode options

PDF417

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread.

Barcode aspect is the height to width ration of the number of codewords per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode.

Start mode indicates the type of data most likely to occur at the start of the data to be encoded.

Font aspect controls the aspect of the font to be used to present the barcode.

Font scaling barcode settings

Datamatrix

Note: Datamatrix barcodes comply to the ECC 200 standard.

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.

Mode determines the shape of the final barcode.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Encoding allows the barcode data to be encoded using the **C40** scheme. If the **Automatic** option is selected then the encoding which creates the most compact symbol will be used.

QR Code

Code page – if you need to include non-Latin characters in the barcode, or the data is **Base64** encoded this allows you to select an alternative code page. This is only available when the mode is set to **binary**.

Mode determines which type of data will be used with the barcode.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Weight reduction enables the font weight of the barcode to be changed.

GS1 Databar



Columns – is the number of barcode columns that will be placed on a single line in the barcode presentation area. Subsequent columns are wrapped onto the next lines. The example shows a barcode that is made up of six columns. When the **Columns** value is set to three the barcode is presented on two lines. This option is available for **Expanded Stacked** barcodes only.

Depending on the selected point size, it may be necessary to increase the magnification to view the barcode in the editor.

2 of 5

IATA 2 of 5 creates a variation of the 2 of 5 barcode which is commonly used in the airline industry

Font scaling barcode settings

Code 128 & GS1-128 (formerly known as UCC/EAN 128)

Trim spaces removes any extra spaces included in the value used for the barcode. **No control code** removes the ability for the barcode type to be set to either A, B or C dynamically based on the optimal mix of alpha numeric characters.

Code 39

Start/stop chars as asterisk enables you to use an '*' symbol instead of an alphanumeric character for the start and stop characters of the barcode.

Inter-cipher gap (1 unit) limits the gap between ciphers in the barcode to one unit. If this option is not selected the gap between ciphers may be inconsistent.

German Postal

Post check digit is only valid when the Auto Check Digit has also been selected. This option enables the barcode image to be used as Identcode and Leitcode symbols.

ISBN

ISBN-13 will prefix the barcode with '978' to conform to new standards being introduced by **ISBN** agencies.

Italian Postal 2/5

Insert Space places a space between the barcode value and the check digit; otherwise a dash is used when a check digit is included.

ITF

Vertical bearer bars provide an outline around the bars of the barcode. **H gauge** adds an additional 'h' character to the barcode to check for impression depth and ink spread. This is for use with older printers. Additional light margin space is also provided by this option.

MSI

Modulo 10 check digit adds another mod 10 check digit to the barcode in addition to the standard single mod 10 check digit.

Modulo 11 check digit places a mod 11 check digit before the normal mod 10 check digit.

Plessey

Separate data from check digit automatically places a space between the value of the barcode and the check digit.

Swiss Postal

B-Post symbol is selected if the barcode is to be used for **Swiss B-Post** instead of the default **A-Post**.

Font scaling barcode settings

Telepen

ESC character makes the first character in the barcode an **ASCII** escape character to conform with certain telepen numeric systems.

Even length will prefix the string of characters with a '0' where necessary to ensure the barcode value contains an even number of characters.

Aztec

Mode determines the characteristics of the final barcode. Choose from:

Normal – allows you to set **Security** to any percentage in the range 1-99.

Compact – use **Security** to specify the number of layers in the range 1-4 used to contain the symbol's information. Error correction data will fill unused space in the specified number of layers.

Full range – as for compact mode but in the range 1-32.

Security identifies the percentage of the symbol to contain error checking data. The range is determined by the mode. A value of 0% will produce the symbol with the default amount of error correction. The higher the security level the greater the number of layers required to contain the symbol and thus its overall size.

Line drawing barcodes

These barcodes are for use with AFP, eHTML, PCL, PDF and Postscript output devices.

Line drawing barcode options

Label

Enter an optional name for the barcode object that will appear in the logic map.

Value

Enter the value you want to use for the barcode. See [Adding values to objects](#) on page 171 for more information. It is important to note that barcode values can be a maximum of 1024 characters. Longer values will cause Designer and Generate to issue warning messages and the value will be truncated when processed by Generate.

Line drawing barcode options

Mode	<p>Line drawing mode barcodes enable you to size the height and width of the bars that form the barcode.</p> <p>Refer to Font scaling barcodes on page 246 and Standard barcode on page 243 and for details on these barcode types.</p>
Barcode group	<p>Select Universal types or one of the 2D barcode groups to populate the Type list.</p>
Type	<p>If Universal types has been selected, this list is populated with barcodes that are compatible with the output devices, otherwise with the 2D barcodes available for the selected barcode group.</p> <p>If QR Code barcode has been selected you have the option to select one of the following QR code types:</p> <ul style="list-style-type: none"> • EngageOne Video - see Linking to an EngageOne Video project on page 241 for further details. • QR Code - the standard QR code format. • Swiss QR Code - typically used to hold data for payment processing.
Scale	<p>These options are used to set the size of the barcode. Depending on which barcode you select, different scaling options will be available. Note that this setting is not available for Swiss QR Code barcodes as they are rendered at a fixed size.</p>
Presentation	<p>These are cosmetic options that affect the presentation of the barcode when it is included on the page. Select 90 degree orientation to rotate the barcode 90 degrees clockwise. Use the Color option to change the color of the barcode from the default black. Note that the color control is disabled for Swiss QR Code barcodes, these are always rendered as black.</p>
Specific barcode options	

Line drawing barcode options

PDF417

Barcode aspect is the height to width ratio of the number of code words per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread.

Height of cell and **Width of cell** define the size of the individual bits that make up the barcode.

Datamatrix

Note: Datamatrix barcodes comply to the ECC 200 standard.

Mode determines the shape of the final barcode.

Security defines the amount of additional codewords in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Width of single cell defines the height and the width of the individual bits that make up the barcode.

Encoding allows the barcode data to be encoded using the **C40** scheme. If the **Automatic** option is selected then the encoding which creates the most compact symbol will be used.

QR Code

Code page – if you need to include non-Latin characters in the barcode, or the data is **Base64** encoded this allows you to select an alternative code page. This is only available when the mode is set to **binary**.

Mode determines which type of data will be used with the barcode.

Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

Note: that when the Swiss QR Code type is selected the Codepage is fixed to UTF-8 and the **Mode** and **Security** options are fixed as *Binary* and *1* respectively.

Line drawing barcode options

Aztec

Mode determines the characteristics of the final barcode. Choose from:

Normal – allows you to set **Security** to any percentage in the range 1 – 99%.

Compact – use **Security** to specify the number of layers used to contain the symbol's information in the range 1-4. Error correction data fills unused space in the specified number of layers.

Full range – as for compact mode but in the range 1-32.

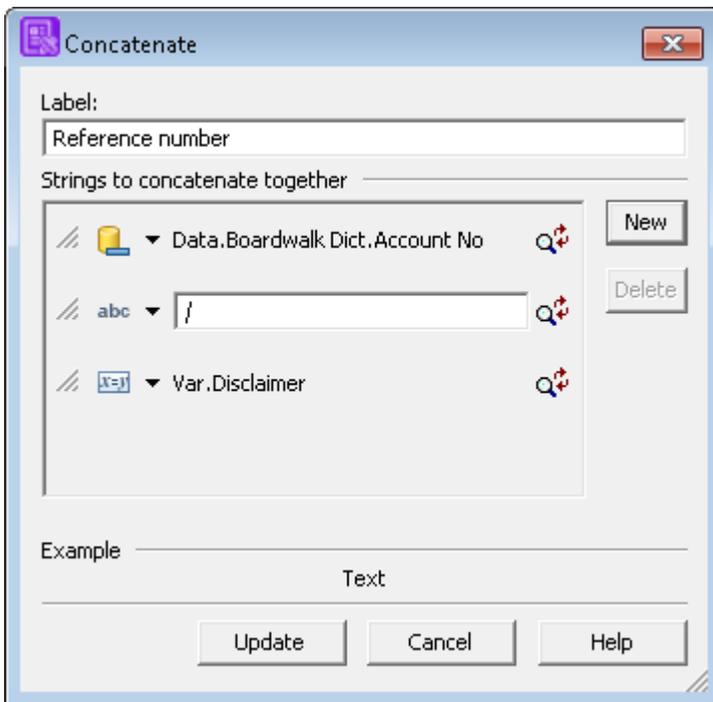
Security identifies the percentage of the symbol to contain error checking data. The range is determined by the mode. A value of 0% will produce the symbol with the default amount of error correction (i.e. 23% + 3 codewords). The higher the security level the greater the number of layers required to contain the symbol and thus its overall size.

Width of single cell defines the height and the width of the individual cells that make up the barcode.

Concatenation

The concatenation function enables you to join together text values from two or more objects to form a single string. The function will treat all values passed to it as text regardless of the actual object type supplying it.

To create a concatenation object: Select **Insert/Field/Concatenation** from the menu– or –when specifying a value for an object, select the **Value** button and select **Functions/Concatenate**. In the Concatenation dialog box click **New** to add the values to be concatenated. See [Adding values to objects](#) on page 171 for details.



Select **New** to add an object reference or constant to the list.

Delete the selected object from the list.

Text is added to the concatenation in the order of these strings. To adjust the order drag the handles at the beginning of each item line.

Lookup tables

A lookup table function allows you to replace predefined keywords with text strings or RTF fields at production run-time. The input keyword is checked against the entries in one or more lookup table files assigned to the job. If a match is found then the text in the lookup table is used to replace the keyword wherever this is used in publication logic

Note that if no match is found a special code is used to replace the string so that it can be identified within the output datastream if required.

A lookup table is normally a simple text file. The keywords are first, followed by a space, followed by the text to be inserted. The keywords must not contain spaces and are case sensitive.

In the example below, if the value assigned to the lookup table function is “sc2” then the string “Jackpot Bonus Scheme” will be inserted.

```
sc1-t Winter Winners Plansc4-mx Spring Specialssc2 Jackpot Bonus Scheme
```

For jobs intended to run under z/OS a lookup table can alternatively be created as a Key Sequenced VSAM dataset (KSDS). The keywords are provided as the VSAM key field and the substitution text is any other content in each record.

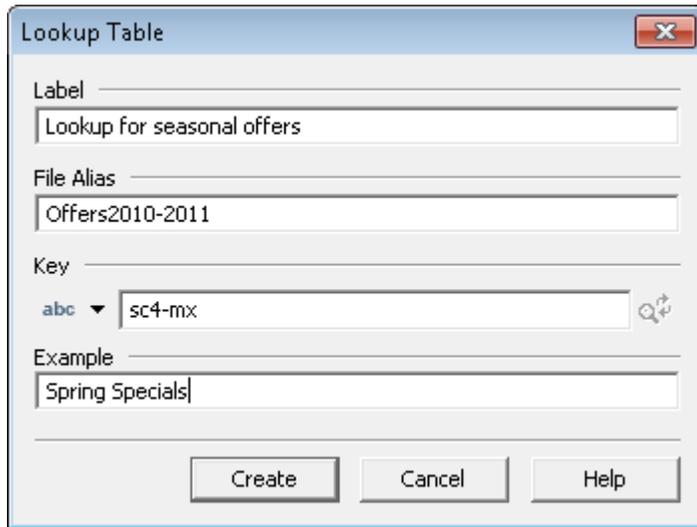
The files to be used as lookup tables for a production job are specified as part of a publishing task. You can override the lookup tables by using an Override Production Settings file (OPS). For more information, see “Using segmented resources” in the “Running Generate” section in the Production Guide..

To create a table lookup function:

1. On the menu click **Insert/Field/Lookup Table** and proceed to step 3– or –create an object to receive the output.

You can update a string variable with the function output or use it as input to any other object that requires a string value. See [Working with values](#) on page 170 for details.

2. When specifying a value for the object select **Functions/Lookup table** as the type.
3. In the **Lookup Table** dialog box:



- **Label** is used to describe the object in the logic map or when specifying a value.
- Enter a **File Alias** that indicates the lookup table that should be used with this function.
- The **Key** provides the value that will be looked up. This must be text or convertible to text.
- The **Example** string entered is used within the document design itself.

Substring

The substring function extracts one or more consecutive characters from within a text string.

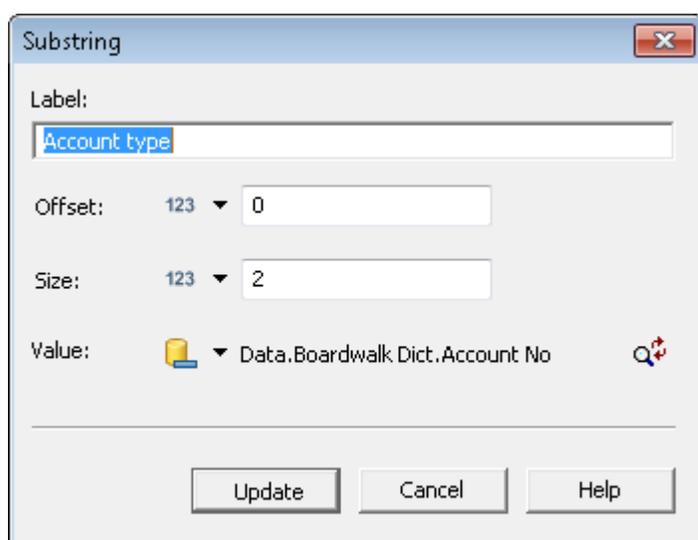
To create a substring:

1. From the **Insert** menu select **Field/Substring** and proceed to step 3– or –create an object to receive the output.

You can update a string variable with the function output or use it as input to any other object that requires a string value. See [Working with values](#) on page 170 for details.

2. When specifying a value for the object select **Functions/Substring** as the type.
3. In the **Substring** dialog box specify the input string and indicate the characters you want to extract.

Note that if the offset is larger than the input text, an empty string is returned.



- The **Offset** from the start of the input text where the substring begins, starting at '0'.
- Size is the (maximum) number of characters to be extracted.
- Select the object supplying the text **Value** from which the substring is to be extracted.

System variables

System variables allow you to use run-time statistics and information from the production environment within functions and other objects; for example, you can input the number of pages generated by the current publication into an arithmetic or use the date when the production job was run with a barcode object.

System variables are also often used as input to a journal to provide an index into the documents and pages within an output datastream generated by a production job. See also [System variables](#) on page 257 for inserting system variables directly into a paragraph.

To use a system variable:

1. Create an object to receive the value.

You can update a variable with the system variable value or use it as input to any other object that requires a value. See [Working with values](#) on page 170 for details.

2. When specifying a value for the object select **System Variable** as the type.
3. In the **System Variable** dialog box select the variable you want to use from the list.

(continued) System variables

Document page number	The current page number within the document being processed.
Document vector	The offset from the start of the output datastream being created by Generate to the start of the data that makes up the document currently being processed. The nature of this offset is system dependent – see “ Vector offsets ” for details. This option is only available for use within Journal objects. To ensure that the offset is the correct one for the document you must place the journal object that sets the vector offset either within the document logic, or before the document in the publication logic.
Driver publication page count	The total number of pages produced by the publication being processed for the supported output device. This option is only available for use within Journal objects.
Elapsed CPU ticks	This measures the elapsed time in CPU milliseconds the calling process – such as Generate or the Refresh option in the Document Editor – has taken up to the point in the logic map where the system variable is inserted. Placing the variable at multiple points in the logic map in conjunction with the Print Message action can be used to identify inefficient logic in the design of your Designer application. Note that while the time is measured in milliseconds it is output as a number.
Generation date/time	The system date or time when the publication HIP file being used by Generate was published.
Last presentation object	Only for use in remaining space, this indicates whether the last presentation object was placed or not. Values returned are: 1 when the last object was placed 0 when the last object was not placed -1 when its use is out of context, for example because of paragraph keep with next settings.
Job page number	The current page number of the entire production job, starting from 1.

(continued) System variables

Mandatory messages	This variable is only relevant for applications that include message objects created using Message1 and Content Author – see Interfacing with EngageOne Deliver on page 335 for more information. Returns 1 if there are messages that were marked as mandatory in the message application and are intended for the specified message area which have yet to be placed in the current document; otherwise returns 0.
Optional messages	As above except that 1 is returned if there are non-mandatory messages yet to be placed.
Page Vector	As for Document Vector but this will be the offset of the current page within the output datastream.
Publication page count	The total number of pages produced by the publication being processed. This option is only available for use within Journal objects.
Publication page number	The current page number within the publication being processed.
Publication sequence number	The number of publications that have been processed so far including the current publication.
Run date/time	The system date or time when Generate began processing the production job.
Table column overflow	Only available within table headers and footers. Returns a True (1) value if the contents of a table flow to another presentation column or page, otherwise it returns a False (0) value. For more details, see Using tables on page 262.
Table page overflow	As above but True or False is set only when a page overflow occurs.
Valid document flag	This value is set to 0 (zero) if there is an error with the publication currently being processed.

Note that some system variables are only available in certain instances, for example, the **Table** options can only be inserted in a table. For variables reporting page numbers be aware that these always relate to the number of 'logical pages' produced by the production job and not necessarily the number of sheets produced on the printer/browser. For example, you may have

chosen to place multiple logical pages on each sheet or have an output device that always adds separator pages to each job.

Vector offsets

The actual values generated for the Vector system variable depend on whether the output datastream is being generated as stream or record based data.

Record data is generated for AFP and line data output under z/OS. The vector value is the record count from the start of file and the first record is number 1.

Stream data is generated for all other output/host configurations. The vector value is the byte offset from the start of file. The first byte is number 0 (zero). Note that the values generated for document or page number '1' will not necessarily be the lowest possible count values. This is because protocols can contain header information related to the overall file but not directly relevant to a particular page.

Note that vector offset values are limited to ten digit numbers. As a result, where Generate needs to deal with file offsets >9,999,999,999 it will use a base-32 encoding scheme to present the value. While the offset value will only be machine readable in these circumstances, you will be able to identify when this occurs as all values will be prefixed with 'Z'. For post-processing requirements PCE is able to read such values as a normal vector offset.

16 - About tables

Designer uses tables for two distinct functions:

A static table is used to align text and graphics in rows and cells or generally to help with positioning objects on a page (as with most word processing tools).

A transaction table is used to control how repeating data is presented, i.e. how multiple iterations of the same repeating data structure are ordered and how they are positioned in relation to one another.

The layout of both types of table are defined in the same way but there are several additional settings that relate only to transaction tables.

In this section

Using tables.....	262
Table positioning.....	262
Basic editing.....	263
Borders and shading.....	264
Headers and footers.....	265
Pagination.....	266



Using tables

This section covers how tables are created and customized. See also [Repeating data and transaction tables](#) on page 120.

The Table Wizard allows you quickly and easily to create the basic structure and controls for any type of table. Once created you can modify the table layout and settings as required.

To create a table:

1. Position the cursor at the paragraph or other presentation object the table is to follow.
2. On the **Table** menu click **Table Wizard**.
3. Use the pages of the table wizard to specify the basic layout for your table.
4. Once the table has been created you can edit the row and column structure as required and specify border and shading settings for individual cells.
5. Add content to the table cells as required.

For advanced requirements such as including a table conditionally or as part of another object you may need to use the logic map when creating it. See [Using the logic map](#) on page 150 and the relevant topic in the section.

- A table is made up of one or many rows. By default, row height is automatically maintained to suit the contents but you may specify a minimum or fixed height if desired.
- To resize columns of aligned cells hover over a separating line then click and drag. To resize an individual cell press and hold Ctrl while dragging.
- To enter text, graphics or data into a table, click in a cell and proceed as normal. Row height adjusts automatically cell width must be adjusted manually if required.
- To select an entire row or column click near the outside edge when the arrow appears.
- Each row can have one or many cells. By default, rows have the same number of cells which are aligned to form a column but individual cells can be resized, added, deleted or within a row, merged together to form a custom grid.
- To resize an entire table, click on it and then use one of the drag handles. The left and right handles adjust cell widths. The bottom handle adjusts the minimum height of rows.

Table positioning

A table within the body area is always positioned in relation to the object selected when it was inserted. You cannot move a table except by cutting and pasting it so that it is relative to a different object.

If you want to position a table at a fixed position on a page or at fixed offset from an object you will need to include it in a text box. Note that tables within a text box cannot grow beyond the page on which they are started.

See [Text boxes](#) on page 101 for information about using text boxes.

As with other presentation objects you can edit the table's properties to change the amount of space to be left between the table and the objects that appear before and after it in the document flow. You can also specify that the table is to be kept together on a single page where possible.

By default, a table is positioned as close as possible to the left hand page margin. You can specify an indent to shift it to the right if required.

Note that widow and orphan controls are normally only useful when designing a transaction table. If you do specify them for a regular table however, the settings relate to the number of individual rows that are to be kept together when the table overflows onto a new page.

To customize table positioning properties: on the menu click **Table/ Properties**. Use the **General** page to select special positioning options.

Basic editing

Content is added to a table by clicking in a cell then typing text, adding a data alias or inserting an object as you would elsewhere.

You cannot insert the following objects within a table cell:

- Another table
- A new page action

Each table row starts off with a default minimum height. As content is added to the table, the height of individual rows will automatically grow to fit by default. You can specify that rows have a different minimum height or that they may not exceed a maximum height.

To adjust row height: select any part of the table and on the menu click **Table/Properties/Table**. Use the **Advanced** page to select the appropriate **Row Height** setting and specify a measurement if required.

When new rows or cells are added they adopt the attributes and dimensions of the currently selected element by default.

To add a row, column or cell: place the insertion point in an existing cell of the table, and from the **Table** menu click **Insert** and select the required row, column or cell option.

Most of the common resizing and editing operations can be performed using mouse operations. However, you may want to specify precise settings for some rows and cells.

When cells are deleted or resized, all remaining cells to their right retain their existing dimensions but close up to the left.

To resize cell widths: select the cells to be adjusted and from the **Table** menu select **Properties/Cell**. In the **Cell** dialog box enter a new **Width** and click **OK**.

Adjacent cells in a row can be merged together to form a single cell (existing content will also be merged). This can be used, for example, for a description that applies to more than one column.

To merge cells: select the relevant cells and from the **Table** menu click **Merge cells**.

To delete tables rows, columns or cells: click in or select what you want to delete, and from the **Table** menu click **Remove** and select the required table, row, column or cell option.

Borders and shading

You can customize the borders and shading used for cells, rows and tables. For borders you can specify the width and color of the line to be used for each edge of a cell, row or table. You can also specify rounded corners for the table. For shading, select the color to be used to fill the cell behind any contents it might have. As with all color issues, bear in mind the capabilities of your target output environment when setting these options.

The settings you specify for a selection of cells are applied individually to each selected cell.

To adjust borders and shading:

1. Select the table, row or cells to be amended.
2. From the **Table** menu, select **Properties** and **Table, Row**, or **Cell** as appropriate.
3. On the **Style** page of the **Table Properties** dialog box specify the required color and line settings.

For borders use the edge palette to select the edges you want to be changed. Note that any unselected edges will have existing settings removed. For shading, select a shading percentage for the color – 0 will be no color; 100 will be full color.

4. Click **Apply** to see the effect of your changes or click **OK** to apply the new settings and close the dialog box.

Tables can have rounded corners. They are applied to each corner of the table, on every page if the table overflows. The rounded corner is drawn in corner cells only; if the size specified is larger than the cell can accommodate, then that particular corner will not be rounded. Note that you may have to adjust the cell margin or format to prevent the contents of a cell overprinting the rounded corner.

Note that if you are using rounded corners on a shaded table and the output is to AFP, the shading may not be consistent with the same shading elsewhere in your publication. To ensure consistency specify CMYK or RGB color support for the AFP device refer to [AFP output device settings](#) on page 527

To specify rounded corners for a table: select the table and from the **Table** menu, select **Properties**. On the **Style** page of the **Table Properties** dialog box select **Enable rounded corners** and specify the required radius of the curve.

Headers and footers

Table headers and footers should not be confused with the headers and footer objects that can be applied in Page Setup options. Using a table header means that the contents of the first row (and, optionally, other consecutive rows) will be repeated at the top of the table whenever the table contents flow to a new page. Similarly, the contents of the table footer row will be repeated at the bottom of the table on all pages in which it appears.

For more information on variables, see [System variables](#) on page 257

When you specify the use of headers or footers it is assumed that only the first or last row is to be used for this purpose. You can add further rows to an existing header/footer if they are adjacent to these.

To enable or disable table headers or footers: select the Table Header or Table Footer object in the logic map and from the **Table** menu select **Insert/Row below** or **Row above**.

To add a row to a header or footer: click on a header or footer row and from the **Table** menu select **Insert/Row below** or **Row above**. By default the new row adopts all the properties of the current header /footer row.

To remove a specific row from headers or footers: select the row(s) and from the **Table** menu select **Remove/Row**.

Using overflow variables

The Designer provides two system variables specifically for use within table headers and footers. **Table page overflow** and **Table column overflow** return a **true** value when the content of the table flows onto another page or presentation column, otherwise a **false** value is returned.

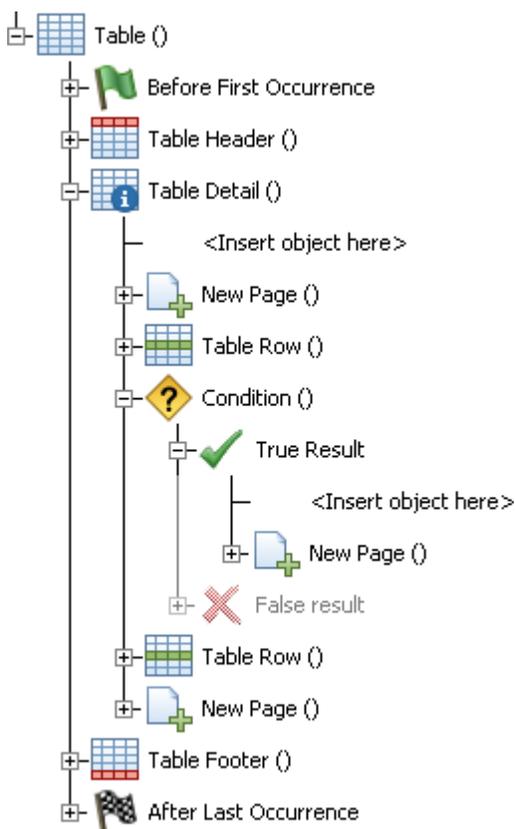
You can use the variables with conditional logic to set the content of your headers and footers based on the value they return. For instance, in a transaction statement you could use the table page overflow variable to display 'Sub-total' when True and 'Grand total' when False in the table footer.

Pagination

By default tables that span pages will automatically break where the last complete row can be fitted on the page. You can change this behavior using the Keep on same page, Widow and orphan processing, Keep transaction rows together table properties or, alternatively, you can manually insert page throws in your table logic where required.

Controlling pagination using table logic

Use the New Page object to specify where a page break is to appear in your table logic. New pages can be placed anywhere within a table detail group, except inside a table row. Note that the table properties described above are overridden when this type of pagination is used.



The New Page dialog box is invoked allowing identification and page header/footer properties to be entered. Refer to [Generating new pages](#) on page 184 for detailed information.

17 - Keyed objects

Designer allows you to select some types of presentation objects dynamically according to variable criteria or 'keys'. Images and Active Content are supported in this way.

In this section

Using keyed objects.....	268
About key maps.....	268
Creating the keyed object.....	270



Using keyed objects

Keyed objects can have a wide range of uses particularly where you would otherwise need to use many case or condition objects to select content. For example, you may want to use a branch number or similar to identify which of a large range of signatures or logo images needs to be placed. If you have more complex requirements you can use the same concept with Active Content objects that can themselves include presentation objects and associated logic.

Note that parameters and return values are not supported for keyed Active Content.

For keyed images, the simplest way to work is to import the files into the Designer so that their attributes are known and they can be referenced directly.

Where it is not possible to use integrated resources (i.e those directly available to the Designer) you can also reference external image resources – using one of two methods. You can create an external key map file that provides a reference to external images along with the relevant attributes. You can also create a user exit program that directly returns the image to be placed as a keyed object.

Note that an external key map is coded as an XML file with a predefined construct. The structure of this file and tools to help you create it are documented in the Production Guide. This guide also contains full details for creating and working with user exit programs.

For handling external keyed images in the EngageOne Server environment see [External Files management and EngageOne](#) on page 392

About key maps

The key map is central to using the keyed object feature. When using integrated resources the key map is a list of image or Active Content files that can be selected by a keyed object function. For external resources the key map allows you to include some sample content for use with the design itself and also provides a name by which the map is cross-referenced to the external control (file or user exit) at publication time.

Such key maps are then referenced when you actually place a keyed object within a design along with the key value itself which dictates which object is actually selected.

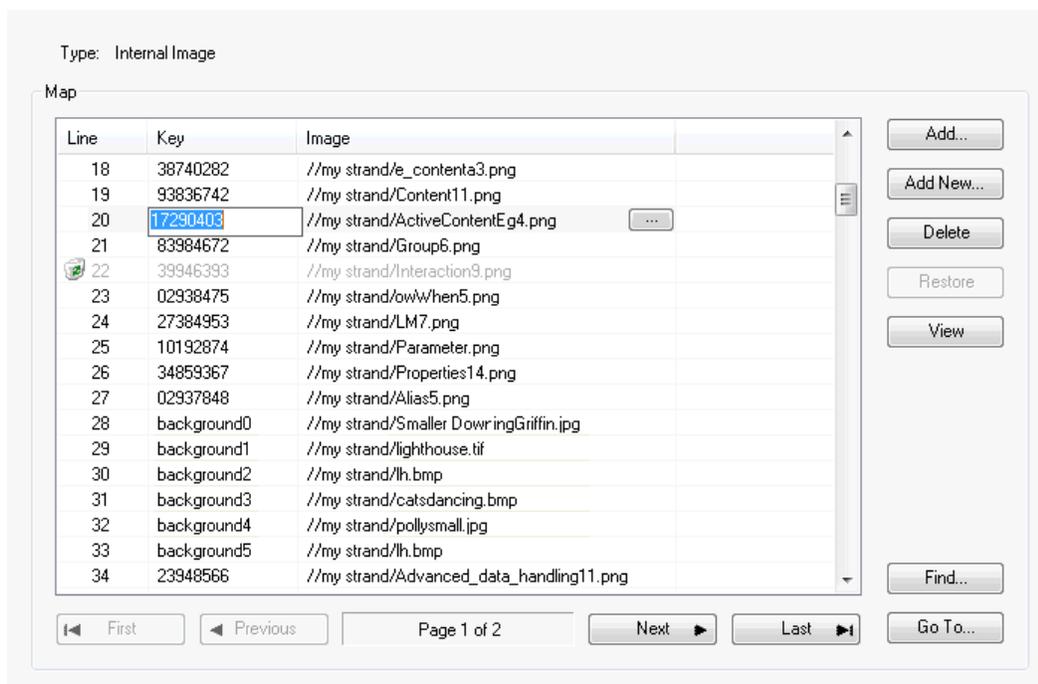
Key maps are maintained as individual files within projects in the navigator. They are available for use with any publication in that project. You cannot mix the type of object in a key map, but you can create any number of maps.

Objects do not have to exist already in the repository before being added to the key map; they can be created or imported as part of the add process.

To create a key map: on the **File** menu select **New** and then select the type of key map you require – **Internal Images Key Map**, **External Images Key Map** or **Active Content Key Map**. Enter a name and an optional description. You can now add entries to the key map.

To add entries to a key map: in the Key Map window select the **Add** button. In the **Add** dialog box, enter a key value (i.e. the value that must be matched for the resource to be used) and identify the actual object that will be used. Note that new entries are inserted at the end of the map.

NOTE: For external images, using a representative sample of the actual images that will be used will help when designing the layout of the page.



- Deleted entries are only removed when the map is saved. They can be restored any time before then.
- A suffix is added if the key is not unique, or when adding more than one object.
- The image file name can be used in the key.

To view the image in a key map: select it in the key map and use the **View** button. The image will be displayed in the workspace.

To create new Active Content files while working with a key map: in the key map select the **Add New** button. In the **Add New Active Content** dialog box, enter a key value and select a project for the new Active Content files. You can create up to 100 at a time. The new files are empty when created so you must edit them to create their actual content and logic. See [Active Content](#) on page 428 .

To import images while working with a key map: in the key map select the **Add New** button. In the **Add New Images** dialog box, enter a value or use the image filename for the key, select a target project into which the new images will be imported and browse for images.

Creating the keyed object

Within a publication design keyed objects can be inserted anywhere that other graphics objects can be used. The object actually placed can be of variable size – other content will flow around it according to the attributes of the keyed object. The object cannot, however, be rotated.

If a keyed object that references integrated resources is used within a text box then the text box will grow as required to fit the content. When referencing external images however, the eventual size of the image is not known and you will therefore need to ensure that text boxes are sized to accommodate the largest keyed object. In this situation a runtime error will be generated if the text box overflows.

You can monitor whether a keyed object has actually been placed at production run-time by specifying a variable into which a status code will be returned. This can then be used within conditional logic to deal with any problems.

The status values are:

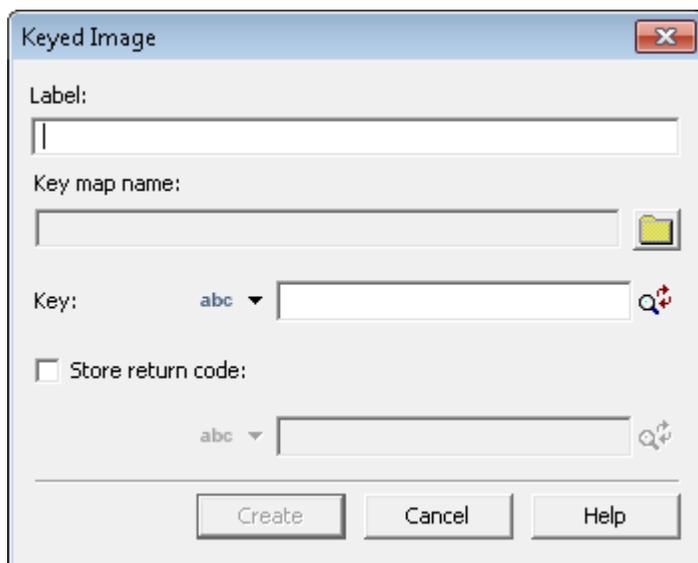
0 –	no keyed object has yet been encountered
1 –	object was placed successfully
2 –	no object was found
3 –	no object was found so the default or wildcard object was placed instead (external keyed objects only)

NOTE: Status values for external keyed objects report status of the lookup entries, not the referenced object files on disk.

To insert a keyed object:

1. In the document editor logic map mark the insertion point as required and from the Insert menu select **Presentation objects/Keyed image** or **Keyed Active Content**.
2. Select the key map. In the **Keyed object** dialog box select the key map that references the objects you want to use.
3. Select the key.

This provides the value that identifies which object is used. Select a data alias or other object that provides the variable data to match the keys specified in the key map. See [Working with values](#) on page 170.



- Browse for the required key map and enter the value that provides the key.
- If you wish to monitor the return code click on the **Store return code** option and enter the variable you want to use to hold the return status.

Once inserted the editor will select the keyed object for display according to the current value of the key (for instance, the field value from the current document in sample data). When referencing external images the editor will show the appropriate object according to any sample entries in the key map (if any).

You can move keyed images and change their properties as for any shape – see [Working with graphics](#) on page 95. Double-click on the object and use the **Keyed object reference** dialog box to change to a different object.

To use a different key or key map in a document: double-click on the object and in the resulting dialog box change the key or key map name.

18 - Charts

Designer has two ways of creating charts that visually represent data.

Native – the standard built in chart engine where the data is inserted into the chart logic map item.

Plug-in – an alternative chart engine supplied with either Designer, which allows you to configure a chart or a custom library, using a single dialog box and shows a preview of the chart as you configure it.

In this section

Which chart type to use.....	273
Using plug-in charts.....	274
Using native charts.....	279



Which chart type to use

Native charts

- Native charts support all color spaces.
- Native charts support all production platforms and drivers except eHTML.

Chart Director plug-in chart

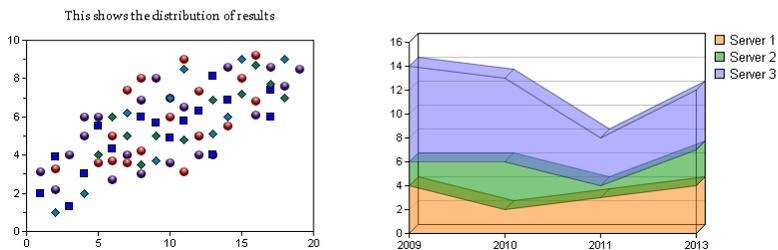
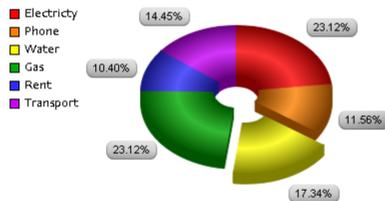
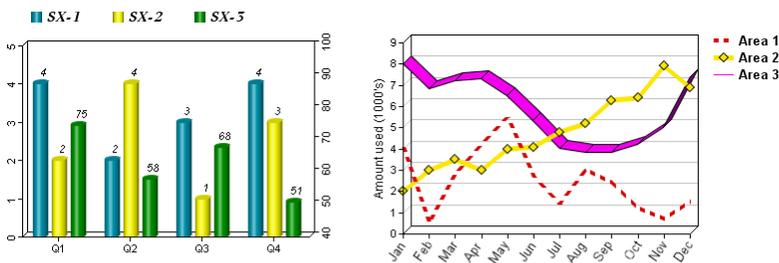
- They do not support CMYK coding – any CMYK values entered will be converted to RGB.
- They are not supported by the z/OS platform.
- Only these drivers are supported – AFP, PCL, Postscript, PDF and eHTML.
- Complex scripts are not supported.
- Although plug-in charts offer a greater range of chart types and styles with more flexibility they may be significantly slower in production.

Charts are treated like any other graphic in the document editor and can be moved, resized and deleted as required. For more information about editing in general see [Working with graphics](#) on page 95.

Using plug-in charts

Use the **Chart Director** to create various types of chart such as bar, line, pie, scatter, area charts.

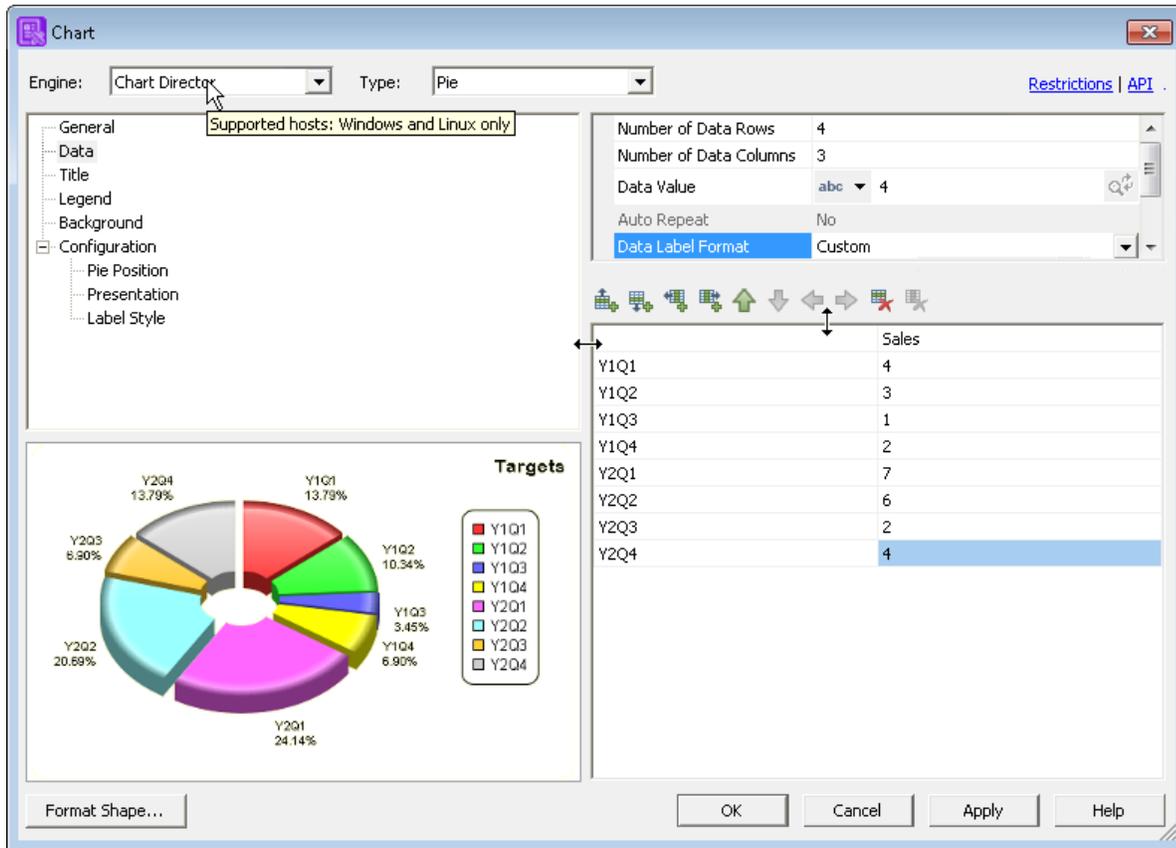
Each new chart is created using default settings, and as you customize it the changes are shown immediately in the automatic preview in the **Chart** dialog box.



To create a plug-in chart:

1. Position the cursor in the paragraph or presentation object that the chart is to follow.
2. On the **Insert** menu click **Presentation Objects/ Chart/ Plug-in**. The **Chart** dialog box is displayed.
3. Select the initial **Type** of chart you want from the drop-down list, for example **Bar, Pie, Area, Mixed** (bars, lines, areas). A chart with default settings is created inline which can then be customized as required. Use the tree options to navigate to and change the properties of the chart.

Bars and lines can be combined to make a mixed chart by overriding the data style for the relevant data columns.



- Click on **API** link to be directed to a public repository in GitHub which provides documentation, source code and an example of how to create your own plug-in chart engine for Designer/Generate.
- Hover over the **Engine** option to show which host platforms are supported.
- Use the tree options to customize the chart and preview your changes as they are applied. See [Configuring Data](#) on page 276
- The options available under **Configuration** in the treeview will depend on the type of chart selected.
- For pie charts – configure an **inner radius** to create the 'ring' effect.

To edit a plug-in chart: either right-click on the chart object and from the context menu select **Properties**

– or –

double-click on the chart object. The **Chart** dialog box will open.

To change the flow, wrapping, alignment etc. of a plug-in chart: select the chart object and from the **Format** menu select **Shape**

– or –

in the **Chart** dialog box (of an existing chart) click **Format Shape**. See [Using graphics](#) on page 96 for more information.

To resize a chart: use the **General** options to set the overall size of the chart object, i.e. the width and height of the background; use the **Configuration/Plot area** options to change the size of the chart itself. You can also use Format/Shape to set the overall size of the chart object.

Configuring Data

Data for the chart is presented in the form of a grid where the columns define sets of data and the rows define the elements within each set.

These charts are drawn with X (horizontal) and Y (vertical) axes. If required you can split the data to use two Y axes – primary and secondary. This would be used, for example, if there are two different sets of units. Pie charts only use the first data column. Initially default values are used so that you can see an instant representation of the chart. These must be changed to use your own data.

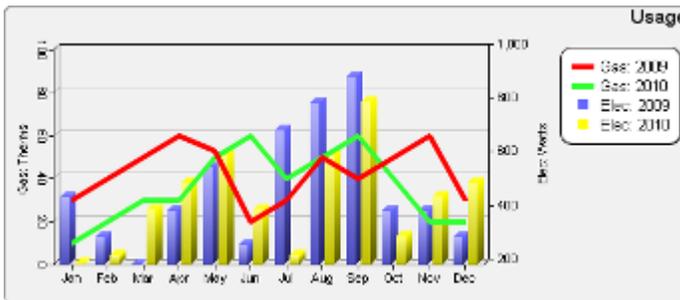
To enter data:

1. In the **Chart** dialog box select the **Data** option in the tree.
2. Configure the rows and columns.
 - a) Specify the required **Number of data rows** and **columns**.
 - b) In the grid select a row header (in the left column) or a column header (in the top row) and enter the required label – either using the **Data value** field or by typing text directly in the cell. Browse for and select the font to be used for the label. Labels for charts can be customized – see [Customizing Labels](#) on page 278 below.
 - c) For the column header you can use the color picker to specify the color to represent the data in the chart. Note that the column headers are used in the legend. Use the **Legend** tree option to configure this.
 - d) Values used for labels can be formatted by clicking the **Format** button – see [Formatting values](#) on page 173.
3. Specify the data.
 - a) Select a cell in the grid and use the Data value to provide input – see [Using data aliases within text](#) on page 117 for more information. The **Data Value** field is disabled if no cell (or the top-left cell of the grid) is selected.
 - b) In most charts the first column in the grid is used to define the categories. These will be evenly distributed on the X axis.
 - c) In a scatter chart, also known as an XY chart, the first column defines the values on the X axis that will be plotted against all the subsequent columns – which define the various Y axis values. Note that the labels on the X axis reflect these values.
 - d) When specifying repeating data or an array (via a variable) the selected data cell is used as the starting point for all associated data – i.e. repeated data and arrays will automatically populate the rest of the column; repeated data groups will automatically populate as many columns as required.
 - e) When the very top (legend) row is selected, the repeated data will automatically populate the row.

- f) Use the fields to configure the selected cell. Select the type of data – data alias, variable, constant etc or, you can enter constants directly in the cell.

Number of Data Rows	12
Number of Data Columns	4
Data Value	Data.Utility data dict.Amount
Auto Repeat	Yes
Data Label Format	
Custom Format	abc
Label Font	Arial 8 pt
Data Color	Red
Data Style	Line
Y Axis	Primary
Line Style	Solid
Line Width	0.127 cm
Data Symbol	No Symbol
Symbol Size	0.254 cm
Bar Shape	Square

- g) Each column can be assigned to the primary or secondary Y axis.



- h) You can add, move and delete rows and columns. The 'first' cell is used to specify repeating data and the light blue cells are populated automatically but can be formatted. Other rows and columns can be added as required.

	Gas: 2009	Gas: 2010	Elec: 2009	Elec: 2010
Jan	30	10	450	205
Feb	40	20	300	235
Mar	50	30	200	405
Apr	60	30	400	505
May	53	50	560	605
Jun	20	60	270	405
Jul	30	40	700	235
Aug	50	50	800	605
Sep	40	60	900	805
Oct	50	40	400	305
Nov	60	20	400	455
Dec	30	20	300	505

Customizing Labels

In addition to the standard label formats for charts you can customize labels using the following parameters, operator and text.

Parameters must be lower case and enclosed in curly brackets.

```
{label}
```

– the text label of the element

```
{value}
```

– the data value of the element

```
{percent}
```

– the percentage value of the element based on the total value

```
<*br*>
```

– a line break

For example:

<code>{value}<*br*>{percent} percent</code>	456.2413 percent
---	------------------

<code>\${value} ({{label}})</code>	\$456.24 (1st Qtr)
------------------------------------	--------------------

Using native charts

Several types of native charts are available in Designer. Two-dimensional vertical and horizontal bar charts are always available; other types are only available subject to your license keycodes.

The chart types are:

- 2-D bar chart - vertical and horizontal
- 3-D bar chart - vertical and horizontal
- Stack chart (2D & 3D)
- Line chart (with & without markers)
- Scatter chart
- 2-D pie chart
- 3-D pie chart

Chart terminology

The following types of native charts are available in the Designer. Two-dimensional vertical and horizontal bar charts are always available; other types are only available subject to your license keycodes.

2-D bar chart – vertical and horizontal

3-D bar chart – vertical and horizontal

Stack chart (2-D & 3-D)

Line chart (with & without markers)

Scatter chart

2-D pie chart

3-D pie chart

Axes

All charts except pie charts have two axes – the lines at the left (Y axis) and bottom (X axis) of the graphic that set the scale for the values presented. The axes can have tick marks at user defined intervals and optional labels.

Chart data

The values that are represented in the graphic.

Elements

Individual elements that make up a chart – a bar, a pie segment, a line, etc.

Grid lines

Optional lines drawn across the chart at intervals on the scale.

Index

Used to cross reference chart elements with data values in some chart types.

Legend

Provides a key for chart elements.

Sets and Groups

Used by some chart types to categorize data at the same level. For instance, a stacked bar chart may represent year-on-year statistics for a number of products. A set would represent the years and the products would be referenced in a group.

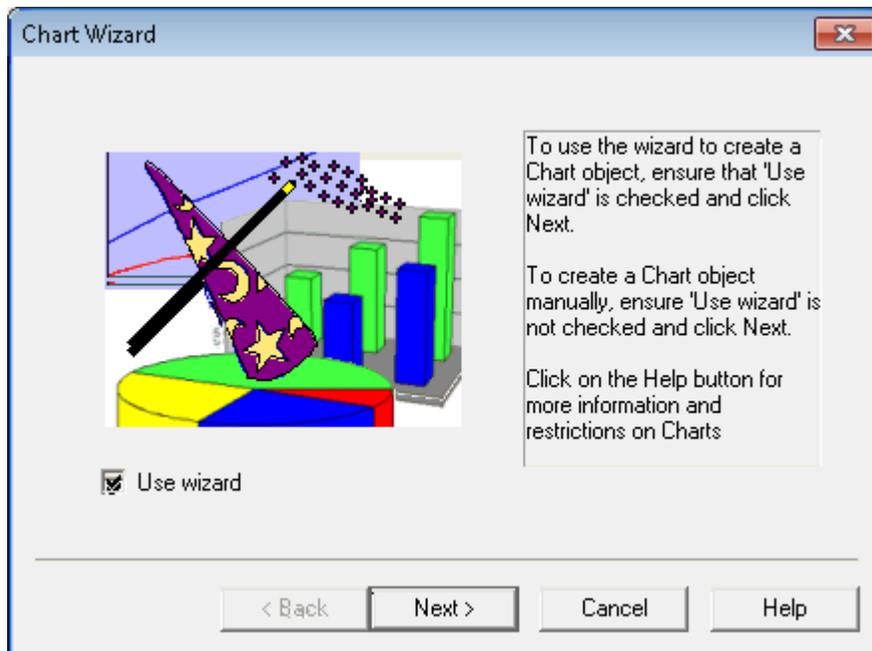
Units

Used by some chart types to specify the scale to be used when representing values. For instance, 64000 would show as a chart value of 64 if units are set to 1000.

Using chart wizard

Designer native charts are typically created using a wizard which assists with selecting the required chart configuration and which automatically creates the appropriate object tree in the logic map.

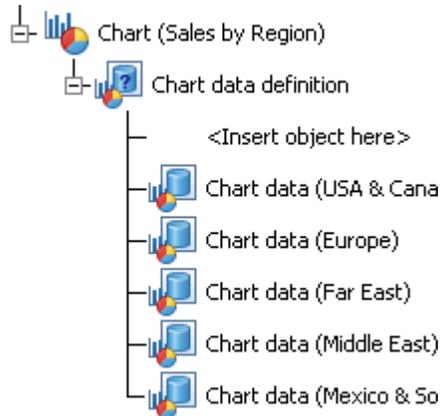
A new chart created using the wizard has dummy values assigned to it. You will need to edit the **Chart data** objects it contains to supply the actual values you want to represent.



To create a chart using the chart wizard:

1. Position the cursor in the paragraph or other presentation object the chart is to follow.
 - a) The chart wizard will create a basic chart layout using the options selected. You will often be able to use one of the **Market** options which will use default settings appropriate for the selected style.
 - b) Ensure that **Use wizard** is checked. If you want to manually generate all the objects required to build a chart create the main chart object as above but uncheck the **Use wizard** option.
2. On the **Insert** menu click **Presentation Objects/ Chart/ Native** button. The **Chart wizard** will be invoked.
3. Use the pages of the chart wizard to specify the initial setting for your chart. When you select **Finish** the new chart is generated and contains all the objects necessary to format the graphic according to your instructions.
4. Specify the values to be represented. In the logic map the main chart object will contain **Chart data** objects for each of the elements specified in the wizard. A dummy value (defined as a constant) will have been assigned to each.

- Specify the values to be represented in the logic map the main chart object will contain **Chart data** objects for each of the elements specified in the wizard. A dummy value (defined as a constant) will have been assigned to each.



- Double-click a **Chart data** object to show its properties dialog box. On the **Data** page use the **Value** control to select a data element or other object that will provide the chart input. See [Working with values](#) on page 170 for more information. You must only supply numeric values to a chart.
- Refine and customize the chart layout. Refer to the information about the chart type you are creating for details.

Native chart axes and scaling

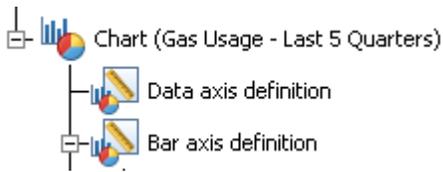
All charts except pie charts have two axes – the lines at the left and bottom of the graphic that set the scale for the values presented. With native charts the vertical line is always known as the **Data axis**. The name of the horizontal axis varies according to the chart type. The axes can have tick marks at user defined intervals, be labeled or unlabeled and you can restrict the data range to be displayed.

To enable the various tick mark settings, **Major Ticks** must be enabled. Once selected this allows you to choose the color for the ticks and whether you want to have minor ticks and labels in your chart.

Selecting the **Show grid lines** option will display grid lines in the chart and enable you to select the width and color of the lines.

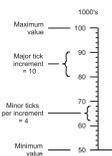
Units determine how the data presented in the chart is scaled to fit the axes. Bar and stack charts scale one axis; line and scatter charts scale both axes. For example, a bar chart with data axis units of 1000 will show a value of 83512 as a bar representing 84. Note that scaled values are always rounded up to the nearest whole number.

To customize axes and define units: double-click the relevant axis object in the logic map. The object dialog box that is invoked will allow you to define the axis and unit settings that are relevant for the chart type and axis you are working with.



Axes and data values

The settings, **Maximum value**, **Minimum value**, **Major tick increment** and **Minor ticks per increment** together determine the look of a chart axis. The first three of these are expressed in terms of the **Units** specified; for example, if units is set to 1000 then a **Maximum value** of 8 actually indicates a maximum value of 8000.



Maximum value and **Minimum value** set the upper and lower limits of the values to be displayed. Values above the maximum will be cropped; those below the minimum will produce an 'empty' element. If these settings both have a value of 0 (zero) the chart scale will automatically be adjusted to fit the maximum and minimum values to be presented.

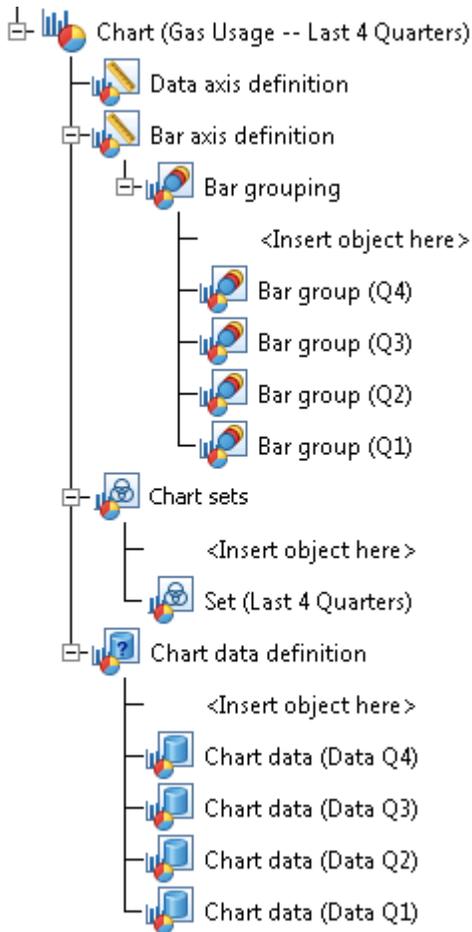
NOTE: If your chart displays the actual data values for each element the correct values will be shown even if they are outside the 'visual' range set by the Maximum/Minimum value settings.

Major tick increment is the number of units between tick marks on the scales axis. These marks have their scaled value next to them. The **Minor ticks per increment** is the number of marks between each major tick. Note that both major and minor ticks are optional.

Sequencing and indexes

For scatter and pie charts the order in which **chart data** objects appear in the logic map automatically determines the sequence in which the elements will appear in the graphic.

Bar, stack and line charts work with groups of values and you will need to use index numbers to associate data objects with the appropriate group. Line chart elements have a single index number – one per line in the chart. Bar and stack chart elements have two index numbers – one for the **Group** and one for the **Set** to which they belong. You will need to ensure that all **Chart data** objects reference the appropriate index number(s).



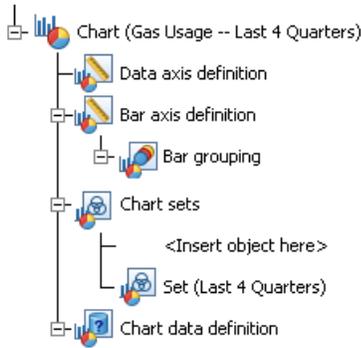
If you used the wizard to create your chart then the element and data objects will have been automatically created with the required index numbers to provide these references. You will need to use new indexes if you add elements to the chart.

In addition to providing cross references the index numbers also represent the sequence in which the elements will be added to the chart, starting from 1 and with a maximum of 32. You will normally want to ensure that the sequence has no gaps but note that, should a particular index number not be assigned a value, then no data will appear on the chart at that position.

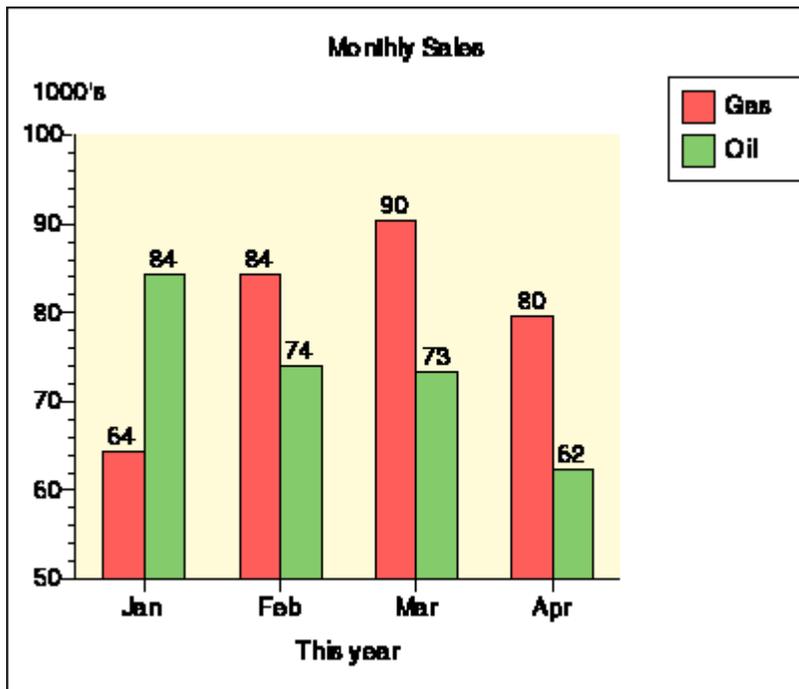
You should refer to the information about specific chart types for details of creating and referencing indexes.

Bar and stack charts

Values within bar or stack charts are always part of a **Set** and a **Bar group**.

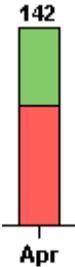


The set identifies values of the same type: for example, the same month in a number of years. The bar group places values within a comparison group: for example, a particular year. If your chart has only one comparison to make – for example, only months for a single year are to be shown – then the group will always be the same.



Definitions

Data axis title	Define in Data axis definition object.
Chart title	Define in Chart object (Label tab).
Legend	Define set names in Chart set objects. Define position & optional drop shadow in Chart object (Legend tab).

Width & spacing of bars	Define in Bar axis definition. The Width setting indicates the % of available space to be occupied by all bars. 100% indicates no gaps between sets. Note that only sets are spaced, not individual bars. To fill the space in a Set when an element has zero value open the Chart object and on the Element tab select Balance elements.
Ticks & labels	Define in Data axis definition. object. You can label the major ticks with the values they represent. Select a font & color as required.
Bar shading & color	A default style is applied when the chart is created. You can change these settings provided the Use default style setting is off (in the main Chart object). Open the appropriate Chart set object to specify new styles.
Frame	Define in Chart object (Frame tab). Set thickness to zero for no frame.
Grid lines	Define in Data axis definition when Major ticks have been selected. Horizontal lines will be drawn at major ticks.
Bar axis title	Define in Bar axis definition.
Bar groups	Define names in Bar group objects within Bar grouping section.
Backing color	You can color the backing box in the Chart object (Frame tab). The default is no color.
Element labels	Define in Chart object (Element tab). Choose Value as the Element label and select a font and color. Use the Chart Data dialog box to customize individual labels.
Elements in the same Set	In stack charts elements in each set are displayed one above the other as shown below: 

Sets and groups are defined as independent objects within the main chart object and must be assigned a unique index number along with their text description that is actually used on the chart. The chart data objects that define the values to be represented on the chart must reference these indexes. See [Sequencing and indexes](#) on page 283 for details.

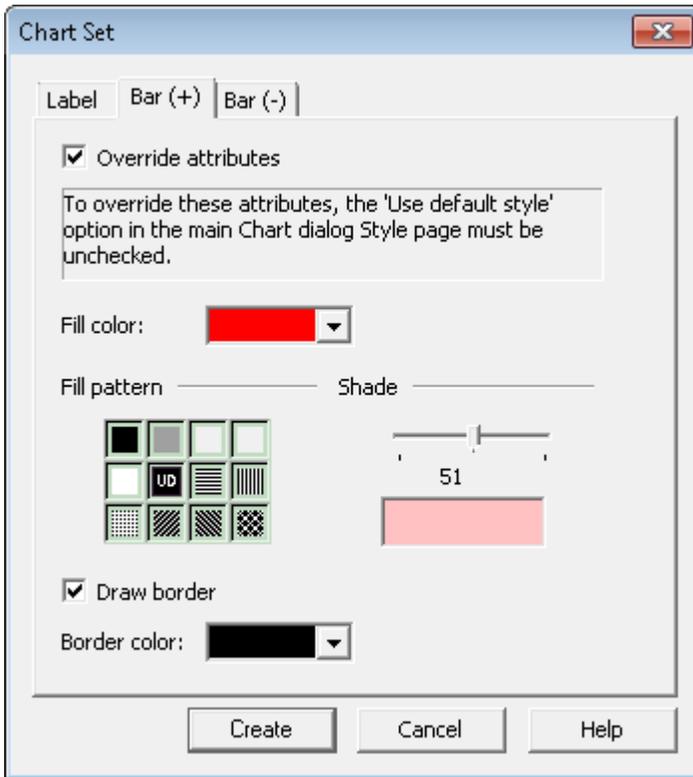
Creating a new Bar group

To create a new Bar group: mark the insertion point within the Bar grouping section of the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/Chart Elements/Group**.

- Assign a value to be used as the group title.
- Select a font & color for the title label.
- Format the label value if required.
- The next available index number is assigned automatically. Change with care if required – See [Sequencing and indexes](#) on page 283.

Creating a new set

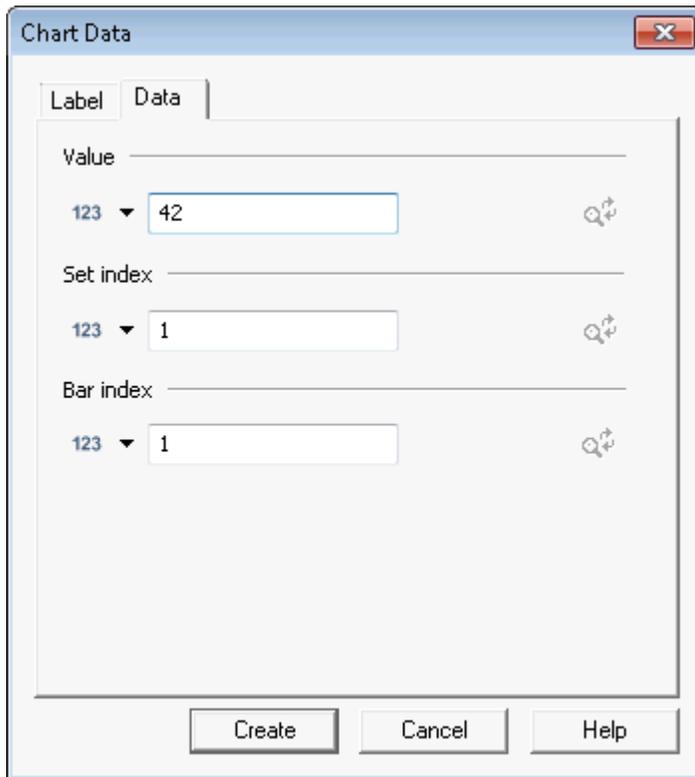
- **To create a new Set:** mark the insertion point within the Chart sets section of the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/Chart Elements/Set**.



- Label & index options are as for a Bar Group (see above).
- If the Chart is not set to use the default style you can use the **Bar** tabs to customize the format of the bar representing this set. You can specify different formats to represent positive (+) and negative (-) values.
- Check the User Defined (**UD**) option to enable custom shading.

Adding bars

- **To add a new bar:** open the Chart data definition section within the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/Chart Elements/Data**.



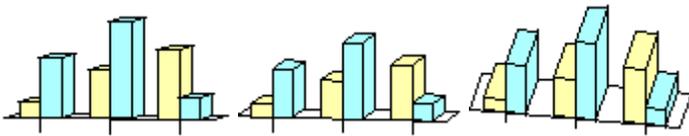
- Select a data element or other object that provides the number value to be represented by the bar. See [Working with values](#) on page 170 for more information.
- Provide number values (often constants) that identify the indexes of the set and group to which the new bar is to belong.
- Click on the Custom label option if you want to specify your own bar labels.
- Enter the value to be used as the bar label directly or select a data element. See [Working with values](#) on page 170 for more

Tilting

You can tilt a 3-dimensional chart to create a greater 3-D effect. You specify the angle that the chart is tilted from the horizontal. 3-D charts are created with a default tilt angle of 5°.

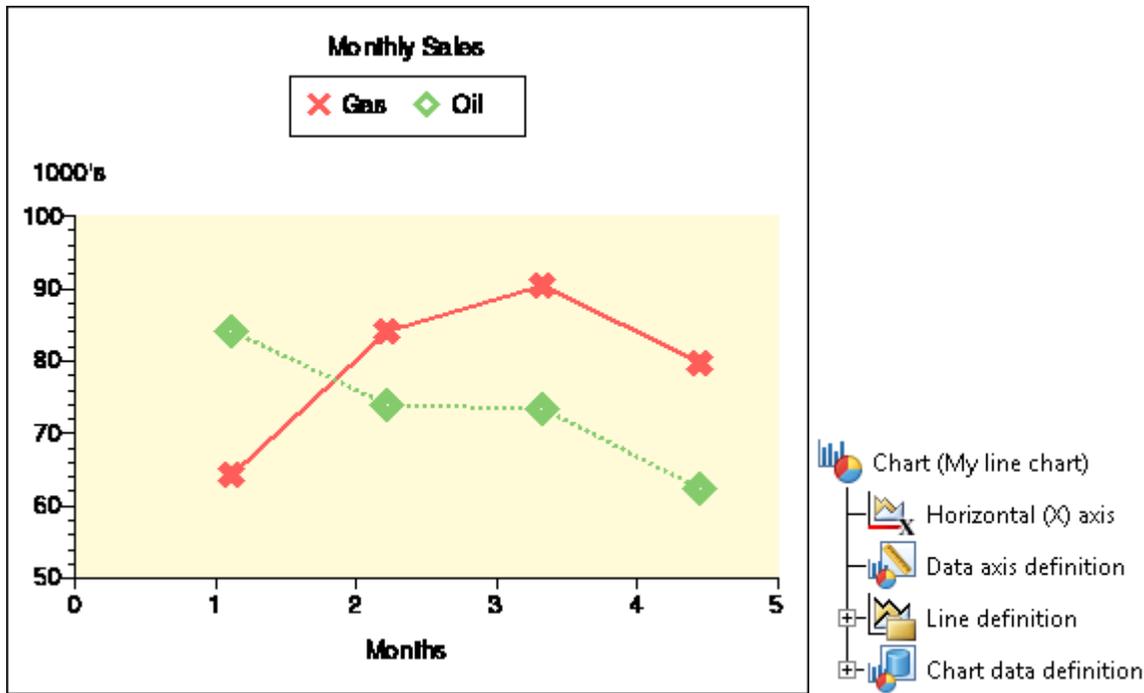
To tilt a 3-dimensional chart: open the Chart object in the logic map and in the **View** tab set the **tilt** to an angle in the range 5-45°.

The diagrams below illustrate various tilt angle settings.



Line charts

Each line is defined as an independent object within the line definition section and must be assigned a unique index number along with a text description to be used in the chart legend. The chart data objects that define the values to be represented on the chart must reference these indexes to be associated with a particular line.



Definitions

Define in **Data** axis.definition.

Chart title

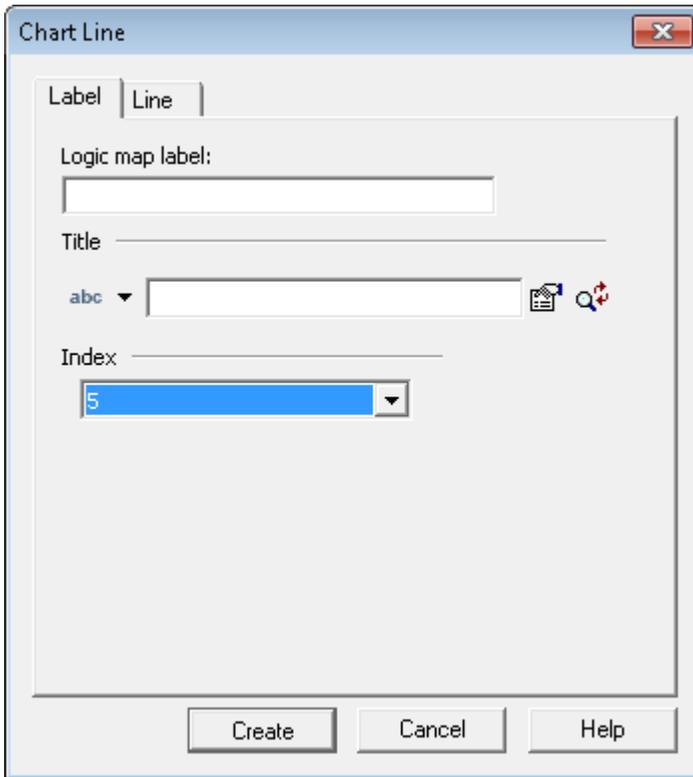
Define in **Chart** object (**Label** tab).

Legend

Define line names in **Line** group objects. Define position & optional drop shadow in **Chart** object (**Legend** tab).

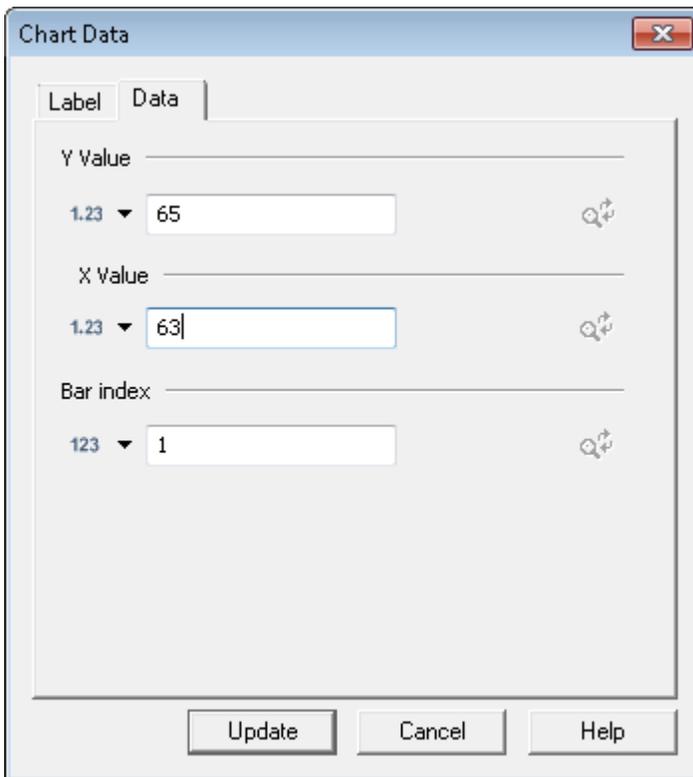
Grid lines	Define in Data axis definition when Major ticks have been selected. Horizontal lines will be drawn at major ticks.
Lines and Markers	A default style is applied to the lines and data markers (if used) when the chart is created. You can change these settings provided the Use default style setting is off (in the main Chart object). Open the appropriate the Line Group object to specify new styles.
Backing color	You can color the backing box in the Chart object (Frame tab). The default is no color.
Ticks & labels Y axis title	Define in the Horizontal (X) and Vertical (Y) axis xis object. You can label the major ticks with the values they represent. Select a font & color as required.
X axis title	Define in Horizontal (X) axis.
Frame	Define in Chart object (Frame tab). Set thickness to zero for no frame.

To create a new line: mark the insertion point within the Line definition section of the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/Chart Elements/Line**.



- The next available index number is assigned automatically. Change with care if required – see [Sequencing and indexes](#) on page 283 for details..

To add a new coordinate: open the Chart data definition section within the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/Chart Elements/Data**.

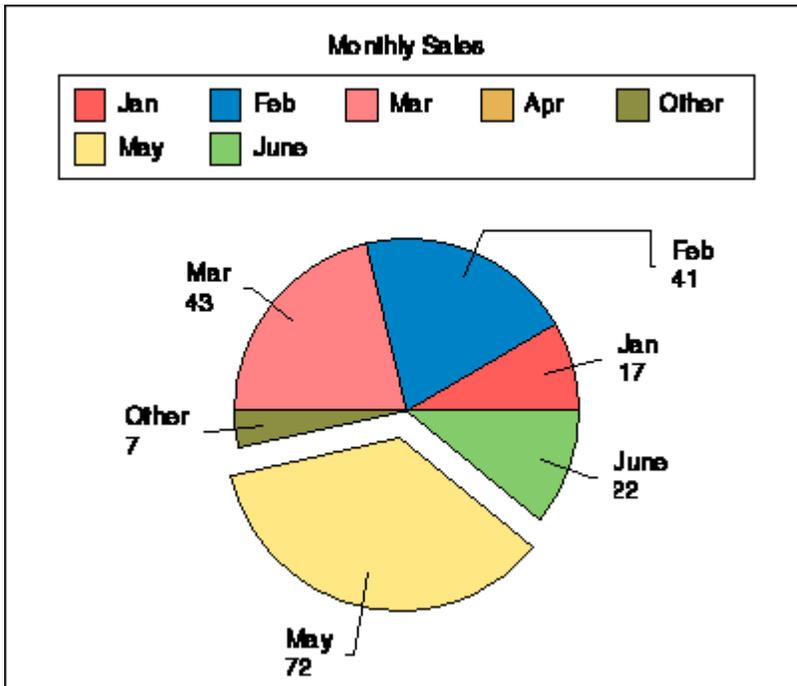


- Select data elements or other objects that provide the X & Y coordinate numbers to be represented on the line. See [Working with values](#) on page 170 for more information.
- Provide a number value (often a constant) that identifies the index of the line to which the new coordinate is to belong.

Pie charts

Each chart data object assigned to the chart equates to a segment of pie that can appear in the graphic but note that you may choose to ignore or group values below a user-defined threshold. Designer will calculate the percentage to be assigned to each segment depending on the contents of all the chart data objects assigned.

Pie charts do not use index numbers. The pie segments are displayed in the sequence they are placed with the chart data definition group.



Definitions

Segment shading & color

A default style is applied when the chart is created. You can change these settings provided the Use default style setting is off (in the main **Chart** object). Open the appropriate **Chart** data object to specify new styles.

Chart title

Define in **Chart** object (**Label** tab).

Legend

Define segment names in **Chart** data objects. Define position & optional drop shadow in **Chart** object (**Legend** tab).

Start position

By default the element relating to the first data object in the data definition group is placed with its bottom line at the 3 o'clock position.

Change this in the **Chart** object **View** tab by specifying a degree of rotation (0-360°).

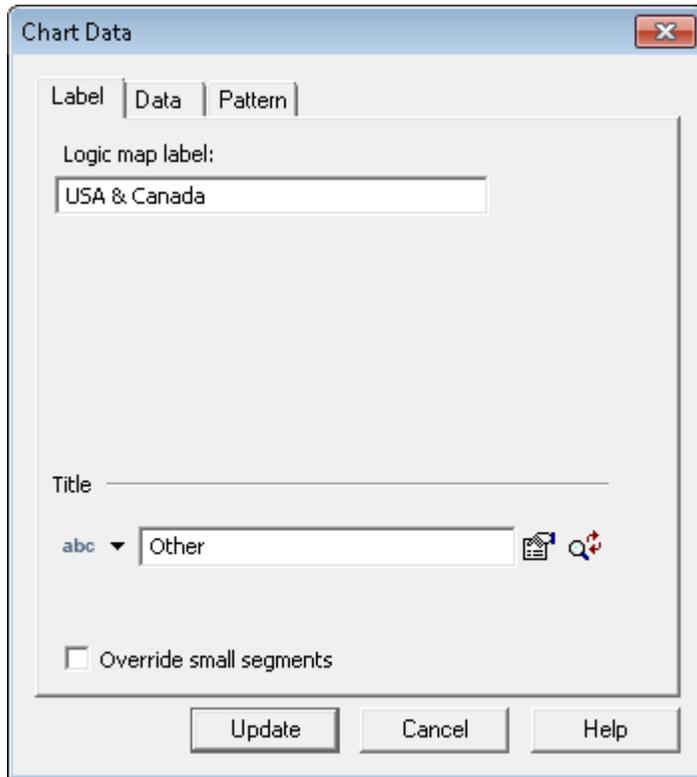
3 o'clock

20° rotation would place the bottom of the first segment here.

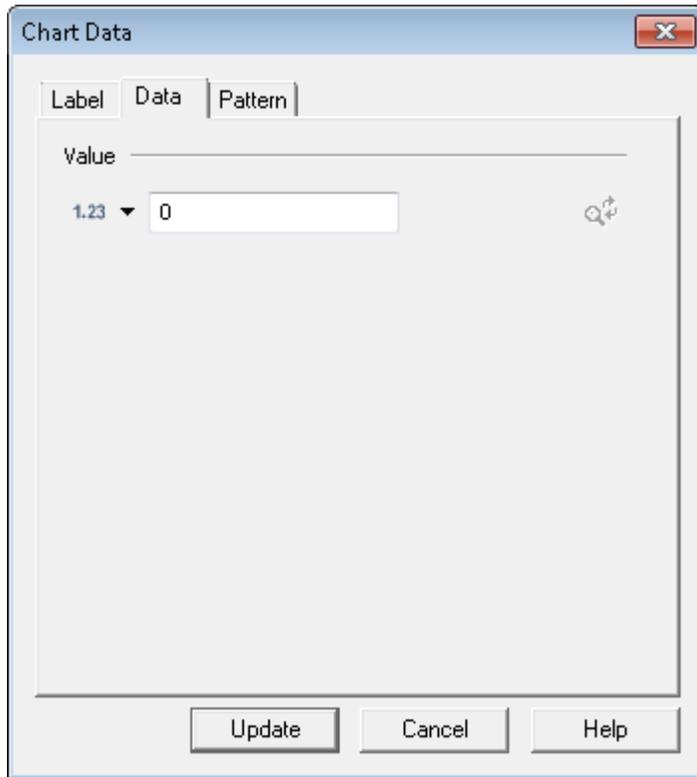
Position/size of the pie	<p>You can move and adjust the size of the pie relative to the overall chart frame. In the Chart dialog box (Frame tab), check <i>Move/resize the pie</i>. This enables the position and size options. Change as required.</p> <p>Note that Height is only enabled for a 3D pie chart, and the position is measured from the top left corner of the frame.</p>
Exploded segment	<p>Double-click on the Chart object and select the View tab. You can specify that all the segments are exploded or just the largest one.</p>
Frame	<p>Define in Chart object (Frame tab). Set thickness to zero for no frame.</p>
Small values	<p>You can group small values into one category with or without labels. See the <i>Customizing small segment display</i> section that follows.</p>
Element labels and pointers	<p>Define the type of label required in the Chart object (Element tab). You may optionally include the segment name and the data either as a percentage or the actual number.</p> <p>A line will join the label to the middle of the appropriate segment. You can choose to align all labels to the left & right of the chart in the Chart object (Type tab).</p>

To add a new segment:

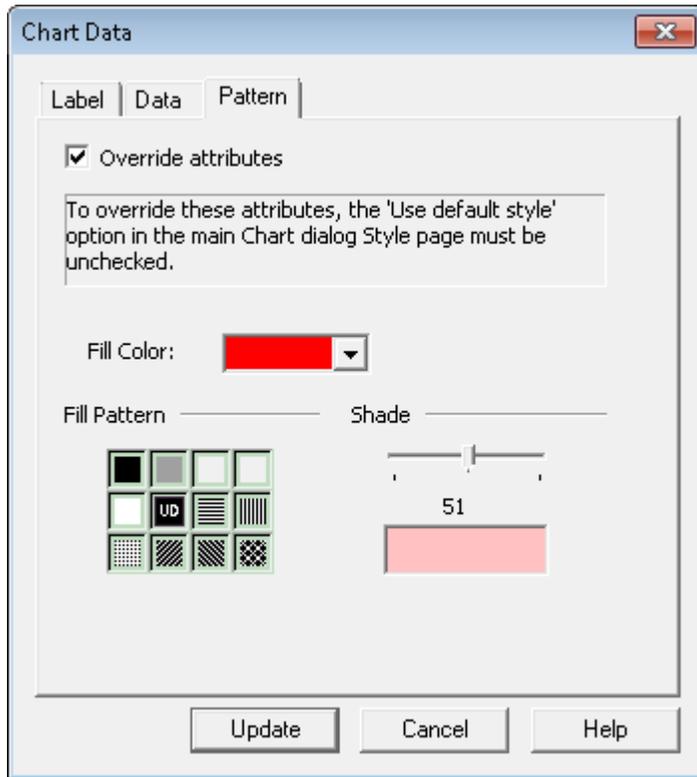
1. In the logic map open the **Chart** data definition section within the **Chart** object
2. Click on the existing chart data object that the new segment is to precede.
3. From the **Insert** menu click **Presentation Objects/Chart/Chart data**.
 - Provide text to act as the legend name for the segment.
 - Override the legend font if required.



- Select a data element or other object that provides the number value to be represented by the segment. See [Working with values](#) on page 170 for more information.



- If the Chart is not set to use the default style check this option if you want to specify an individual fill/color for the segment



Tilting

You can tilt a 3-dimensional chart to enhance the 3-D effect. You specify the angle that the chart is tilted from the horizontal. 3-D charts are created with a default tilt angle of 5°.

To tilt a 3-dimensional chart:

open the **Chart** object in the logic map and in the **View** tab set the **tilt** to an angle in the range 5-45°.

The diagrams below illustrate various tilt angle settings.



Labels

Element labels can be drawn either close to the pie segments or aligned on the right and left sides of the chart.

To change the position of labels on the chart: open the Chart object in the logic map and in the **Type** tab use the **Align pie segment labels** option.

NOTE: Label pointers may be combined where small segments overlap.

Color of 3-D pie chart base

The base of a three-dimensional pie chart can use either the same colors as the top of individual segments (this is the default), or you can specify a single color for the entire base.

To set shading for the pie base: open the Chart object in the logic map and in the **View** tab check **Use shading**. Select the **fill color**, **pattern** and **shade** as required.

Customizing small segment display

Your chart may be assigned values that would generate only a very small pie segment and which may not look very clear. For this reason there are several options that allow you to customize how such values are handled. These are all based on the threshold value initially set in the Chart wizard or Chart data object that determines when a value is to be treated as 'small'.

By default, the first Chart data object in the logic map – labeled **Other** – contains the small segment options.

To set the threshold for small segments: open the **Chart** data object in the logic map and in the **Data** tab use **Value** to set the number below which the small segment options will be applied. By default, the threshold value relates to actual chart data but if you select the **Treat value as percentage** option the value represents a percentage of the overall pie. If a value will produce a segment smaller than the percentage the small segment options will be applied.

To show all values regardless of size: open the Chart data object in the logic map and ensure that the **Override small segments** option is not checked. All segments will appear in the chart, regardless of size.

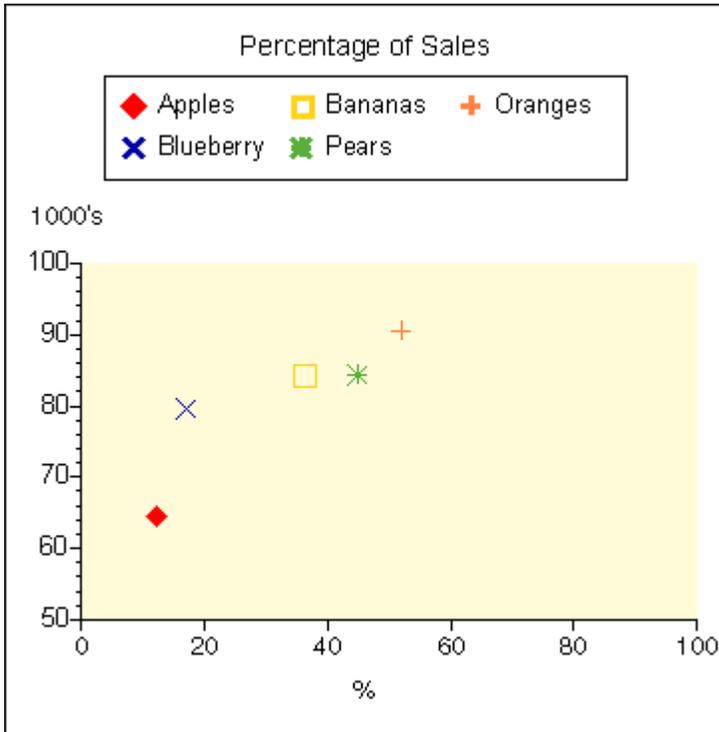
To remove labels from small segments: open the **Chart** data object in the logic map and ensure that **Override small segments** is checked and click **Drop labels**. The labels for segments with a value less than the threshold will not be displayed.

To group small values together in one segment: open the **Chart** data object in the logic map and ensure that **Override small segments** is checked and select **Group together**. In **Pattern**, specify the color for the grouped segment and use **Title** to select title text for the group segment.

Scatter charts

The coordinates are not connected to one another and therefore scatter charts do not use index numbers. However, you can use grid lines to assist with reading the relative scale position of the values provided.



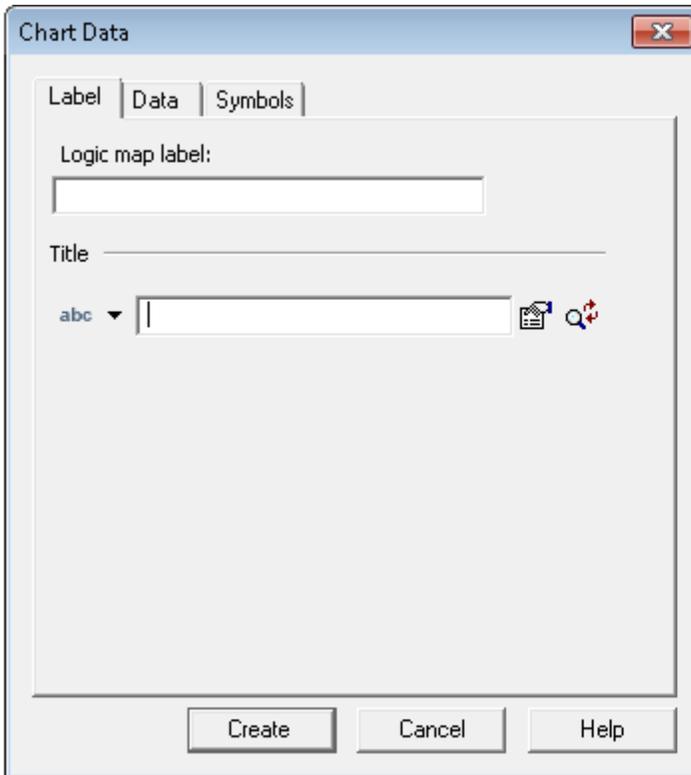


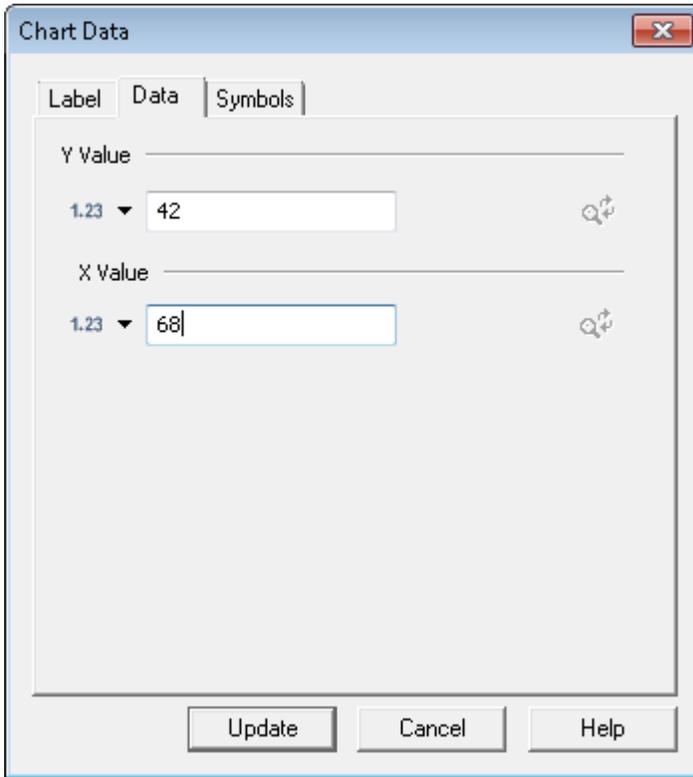
Definitions

Chart title	Define in Chart title
Legend	Define coordinate names in Chart data objects. Define position & optional drop shadow in Chart object (Legend tab).
Grid lines	Define in Data axis definition when Major ticks have been selected.
Markers	A default style is applied to the data markers when the chart is created. You can change these settings provided the Use default style setting is off (in the main Chart object). Open the appropriate Chart data object to specify new styles.
Backing color	You can color the backing box in the Chart object (Frame tab). The default is no color.
Ticks & labels	Define in the Horizontal (X) axis/ Vertical (Y) axis object. You can label the major ticks with the values they represent. Select a font & color as required.

X axis title	Define in Horizontal (X) axis
Frame	Define in Chart object (Frame tab). Set thickness to zero for no frame.

To add a new coordinate: open the Chart data definition section within the Chart object in the logic map and from the **Insert** menu click **Presentation Objects/ Chart Elements/Data**.





- Provide text to act as the legend name for the segment.
- Override the legend font if required.
- Select data elements or other objects that provide the X & Y coordinate numbers to be represented. See [Working with values](#) on page 170 for more information.

19 - Advanced data handling

The information in the chapter [Working with input data](#) on page 117 details most of the features you will need to use when using elements from your intended input data file in publication designs. This section covers any other topics related to the use of input data not covered elsewhere.

In this section

Sample data and data mapping.....	304
Associating and editing data maps.....	304
Using a repeating data control.....	305
Creating a repeating data control.....	306
Repeating data dialog.....	308



Sample data and data mapping

When you are designing publications, you work with indirect references to the records and fields that will be supplied by the input data file in the production environment. The actual data elements are defined in a data format file which uses a sample of the intended input data as a template. Before publishing the data aliases within a publication must be linked to the relevant elements in a data format using a data map object.

NOTE: You can create a data map as part of the publishing task itself. See [About publishing and deployment](#) on page 591 for details.

This allows a great deal of flexibility for creating and maintaining data references within designs but until a publication has been associated with a data format the document editor is unable to use variable examples to show data elements. Data aliases are displayed using the example data you specified when they were created.

Associating and editing data maps

If you want to see the contents of a sample data file used in the document editor you will need to create a data map object.

You can specifically create a new data map and associate it with a data format and your publication. You can then add individual links between the data elements defined in each.

Alternatively you can export both a data map and a data dictionary from an existing data format. This will allow designers to work with a list of known data elements and avoids the need to manually link individual elements.

Data maps and sample data can also be associated with components of a publication.

Data map and sample data assignments made at publication level are passed down to Active Content and public documents included in your publication. See [Sharing and synchronization](#) on page 162 for details.

While editing public documents and Active Content independently of a publication you can use data maps and sample data of your choice. This provides the flexibility to see how the design of a document or Active Content used in multiple publications is affected by different sample data. Note that the link between a data map and a publication is always the latest version. If you make any changes these will not take effect until saved.

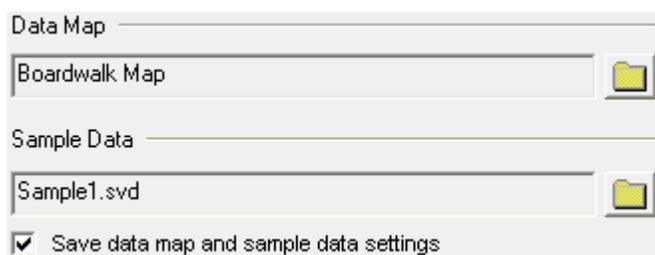
Associating and editing

1. Open the required Active Content or publication from the navigator. If you opened a publication, change to the Publication view.
2. Right-click on the Active Content or publication object in the logic map and select **Properties** from the context menu – or – double-click on the object in the logic map.
3. On the publication tab click the **Data Map** browse button and select a data map from the repository, click **Open**.
4. Click the **Sample Data** browse button and select a sample data file from the repository, click **Open**.

For Active Content, select the **Sample Data** tab and then click the **Data Map** browse button to browse for and select a data map from the repository.

NOTE: See [About data mapping](#) on page 495 for more information about data maps and how to create them using the data mapping wizard.

5. If necessary, click the **Sample Data** browse button, select a different sample data file from the repository and click **Open**.
 - **Save data map and sample data settings:** Selecting this option retains the data associations for the object, otherwise there will be no data associated next time the object is opened.
 - **Data map:** Selecting a data map will automatically include the sample data associated with it.
 - When you have mapped data references in your document design, use the navigation controls in the document editor to browse back and forth through the sample data sequentially or provide a record number to view a specific record in your document.

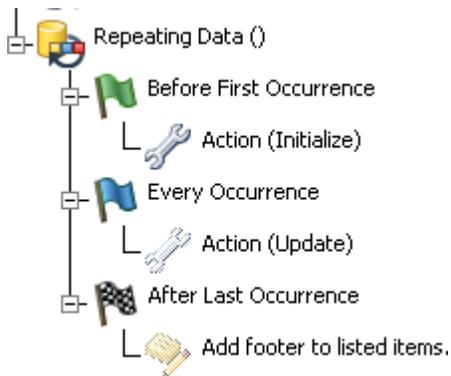


Using a repeating data control

The **repeating data** object allows you to process the multiple iterations of a repeating data element without using a transaction table.

As with a transaction table you can choose to sort the data within the control object before it is actually processed within the function. You can also access the number of the current repeat and the total number of repeats during processing.

You may need to use this feature if you want to process repeating data without actually placing fields in a document design, for instance if you want to total values from the data and store them for later use. You may also find it allows you more flexibility than a transaction table if you need to create complex logic in relation to placing repeating data.



In the logic map, the object has a main **Every Occurrence** group in which you specify the logic and, optionally, the presentation objects that will be processed iteratively according to the number of times the control element is repeated. It also has a **Before First Occurrence** group which is processed once when the control object is first encountered and an **After Last Occurrence** group which is processed once after all the repeating data has been handled. You can use presentation objects within the **Before** and **After** groups to create header and footer style sections around the repeating data if required – although presentation objects should not be used when the repeating data object is in a table.

Note that when processing a data record, the viability of other data records is governed by the structure in which they are defined in the data format. Only those records that are above or below in the same branch are viable. Records that are on the same level, i.e. from the same parent, are not viable. Nested repeating records are also not viable unless they are contained within the **Every Occurrence** of nested **Repeating Data** controls.

Creating a repeating data control

All presentation objects within the object are placed within a single column format which can occupy all the available page width.

Graphics and objects on the background may constrain the available space as during normal processing. Presentation objects within the **Every Occurrence** group are repeated for each iteration of the control object in the input data. These will automatically flow down the page and new pages will automatically be generated as required.

To create a repeating data control:

1. In the logic map, click on the object after which the repeating data control is to be inserted.
2. From the **Insert** menu select **Control Objects/ Repeating Data**.
3. Specify the control for the repeating data.

In the **Repeating Data** dialog box use the **Control Object** either to select the data record that is repeated or to specify the number of repeats in a variable.

Note that you can use the **New Alias** button to create a new record if a data alias for the required element has not yet been established.

4. Specify data sorting.

By default, repeating data will be presented according to the order in which it is provided in the input data file. If required, you can specify a sort that re-sequences the data prior to being placed in the document – this is available only when a data record has been selected as the repeat.

Sorting is based on the sequence of characters in the Unicode table. Results should be fully predictable when data contains standard ASCII-based characters but using sorting with other characters, particularly those from ideographic languages, may produce unexpected results.

For a **Sort** the contents of a field within the repeating data is used as the sort key. Select the name of the appropriate data alias in the **Field** control. You can also specify if the data is to be sorted ascending or descending.

5. Specify a repeat count.

In the Advanced page of the **Repeating Data** dialog box you can use the **Monitor status using these variables** option to specify variables (or Active Content parameters) as required in which to save the number of the current repeat and the total number of repeats. These are standard variables, so once set, are always available for use.

6. Populate the **Before First Occurrence** group.

Expand the Repeating Data object in the logic map. In the **Before First Occurrence** you can include one or more paragraphs that will be placed immediately above the repeating data itself. You may also want to add any logic to be executed before the repeating data is processed. Note that presentation objects should not be used when the repeating data object is in a table.

7. Add objects to be iterated in the body area.

Click **Every Occurrence** within the group and add one or more paragraphs that will be iterated for each repeat. When using data records to control the repeat, you can include data aliases from each record here. See [Using data aliases within text](#) on page 117 for details. Note that presentation objects should not be used when the repeating data object is in a table.

Consider using tabs within the paragraphs to line up fields if appropriate.

8. Populate the **After Last Occurrence** group.

Paragraphs in this group will be placed immediately after the repeating data. You may also want to add any logic to be executed after the repeating data is processed.

Repeating data dialog

This option controls how much example data for the repeating data control is shown in the document view.

Repeating data dialog box – General settings

Number of repeats to show in editor

Where data aliases have already been mapped to a data format the examples will be drawn from the sample data currently associated with the publication. Where data aliases have yet to be mapped the example data specified for each field will be used (repeatedly if necessary). These options determine how many such sample transactions are displayed. **IMPORTANT:** this has influence only during editing and has no effect on how data is handled in preview or production.

Until new page	Enough transactions will be displayed so that all available space on the first page is filled plus one transaction on the new page.
All	If the publication has already been mapped to a data format, data from the currently active sample data file is used to control the repeats that are displayed.
Limit to a maximum of	Only the number of transactions specified will be displayed.

Spacing

These values indicate the amount of space to be left between the complete set of presentation objects produced by the repeating data object (if any) and the objects that are placed before and after. They do not influence space within the objects produced by the element itself. If you want to make these kind of adjustments use paragraph settings.

Additional rules for including RTF fields in input data.

- Designer allows you to include RTF code within your input data and subsequently incorporate the formatted text directly into your design documents from the data format. The RTF can include simple text formatting but more advanced formatting is not supported –
- **RTF** on page 560 see for a list of all restrictions.
- XML input data must be used if you want to include hard returns in the RTF field. The RTF must be placed within a <CDATA> container for the hard returns to be handled correctly by Generate. For more details on XML, see [Working with Designer input data](#) on page 462

- To enable the RTF to be formatted correctly, when you import your sample data into a data format you need to ensure that the RTF field is defined as type **string**. While this will be done automatically for XML data, you must define it manually for other data types. [Creating a data format](#) on page 472 contains information on defining records and fields.
- For more information on RTF fields, see [RTF fields](#) on page 176.
- For information on including resources that are required by RTF fields in the HIP file see the Additional Resources section of the publish wizard, refer to [Running a publishing task](#) on page 602.

20 - Localization

Designer uses the concept of locales to allow control and customization of data and presentation objects to take account of regional variations. Such requirements may range from simply using regional date and currency formats to complete customization of document content including text translation and conditional resource selection.

In this section

About locales.....	311
Localizing document designs.....	312



About locales

Locales operate on several levels. Locale objects themselves contain default formatting information for most Designer data types. For example, you can specify the standard currency symbol, the character used to separate date components, the way in which positive and negative numbers are formatted and so on.

A locale is used as part of a data format to specify the default manner in which date and number values are evaluated when input data is parsed.

NOTE: See [Creating a data format](#) on page 472 for details.

Designer provides a default locale for publications and the design objects contained within them. By default, English (United States of America) is the locale that is used by publications until you choose to change the setting.

Document designs can reference many locales which are used to determine the default formatting required when data aliases, variables and other objects are output.

NOTE: You can override these defaults on a case-by-case basis. See [Working with values](#) on page 170 for details.

Each design object has a default locale that is active when processing of a document begins. An action in document logic allows you to switch to a different locale at any point during processing.

You can also create localized variations of entire paragraphs, normally so that the text they contain can be translated. The paragraph variants are associated with particular locales and, as with regional settings, are activated when the active locale is switched.

To set the default locale: in the navigator browse to and select the **Locale resources** and on the **Tasks** menu select **Set as Default Locale**. Note that you may need to use the filter to see locales.

Creating and customizing locales

Locales are stored as independent objects within the Designer and, permissions allowing, can be referenced by any document design or Active Content object. A default set of locale objects is provided with the product distribution material. You can customize these as required and create new locales to suit your needs.

To create a new locale:

1. In the navigator browse to the Locale Resources folder and on the **File** menu select **New/Locale**.
2. Enter a name for the locale and an optional description and click OK.
3. The **Locale Configuration** window will open. Specify the settings for each data type as required.

To edit an existing locale: double-click on a locale in the navigator and edit the settings as required.

Localizing document designs

Each document design has a default locale which sets the default formatting to be used when data aliases, variables and other objects are output. The selected locale is automatically activated whenever the document starts to process.

NOTE: Unless you have changed the default locale for the Designer, English (United States of America) will be used as the default locale for your design objects until specified otherwise.

If you are using sections within document designs you can override the locale specified for a document on a section-by-section basis.

To set the default locale for a document:

1. Double-click the appropriate Document icon in the logic map of the document or publication editor.
2. In the Document dialog box use the Locale button to select from a list of existing locale objects known to the Designer.

Overriding and changing the active locale

At any point within document or publication logic you can specify an action to activate an alternative locale. The new locale will then be used until a further Change locale action is encountered or until a new document (or a section with a locale specified) is loaded.

Note: Locale changes made in Active Content logic do not persist outside the scope of the Active Content.

To override the default locale for a section: double-click the appropriate Section icon in the document logic map and in the **Section** dialog box use the **Locale** button to select an alternative locale object.

You must ensure that the correct dictionary for the selected locale is used for the spell checker to be accurate.

To change the active locale:

1. In the document or publication editor logic map mark the insertion point at the required position.
2. From the Insert menu select **Control objects/Action**.

3. In the **Action** dialog box, select **Change locale** and using the browse button select a locale to be activated when the action object is encountered.

To ensure that the Change Locale action takes effect, you must remove any locales assigned to lower level documents or sections.

Localizing paragraphs

You can create variations of paragraph content which are automatically selected according to the active locale. The required locales must be specified before localized paragraphs can be created – and this determines the number of variants a paragraph can have. You must indicate which paragraphs are to be localized and the language (locale) required for each. Once defined, such paragraphs each have a master version and a variant for each selected language. The content of any existing paragraphs is automatically copied to all language variants when it is marked for localization (and when new languages are added). Once content has been applied to language variants it must be maintained independently of the master. You can create localized paragraphs within document designs and Active Content objects.

Enabling localized paragraphs

1. For a document, double-click the appropriate document icon in the logic map of the document or publication editor. Or, for Active Content, double-click the **Active Content** icon in the logic map of the **Active Content** editor.
2. Select the **Languages** tab.
3. Click the **Add** button and select from the locales (and hence languages) in the system.

Once localized paragraphs have been enabled you will be given the option to automatically set available languages for all paragraphs in the document or, you can make the assignment on a paragraph by paragraph basis, as detailed in the section that follows.

Setting and editing localized paragraphs

To set the available languages for a paragraph:

1. Double-click the paragraph icon in the logic map– or – highlight the paragraph and on the menu click **Format/Paragraph**.
2. In the **Paragraph** dialog box select the **Multiple Languages** check box.

The languages included in the **Languages** tab of the **Document** dialog box are now available to the paragraph.

Editing localized paragraphs

By default, the document editor will display (and allow you to edit) paragraph variants according to the locale that has been activated by the logic currently in force. For instance, if you have a

condition that sets the active locale for one of two languages, the condition that currently evaluates to true will determine the paragraph variants that are displayed.

You can override this default and show all the paragraph language variants for a specific locale if required.

To work with specific language variants: in the toolbar select the **Language** list and select the localized paragraph language you want to display.

When editing paragraph variants you can do general text editing, such as changing the fonts, line spacing, moving and deleting sections of text. These changes will apply to the current language variant only. You may need to adjust the content of other language variants manually to reflect any changes. Note: Only languages added via the **Languages** tab in the **Document** dialog box are available.

21 - Production journals

Journal files allow you to record the activity of a publication when it processes in the production environment. You may often want to use journals to make an index for the documents and pages produced by the application. These can help with things such as post-processing requirements, audit trail and archiving functions.

In this section

Working with Production journals.....	316
---------------------------------------	-----



Working with Production journals

You can write as many entries as required to one or many journal files. During processing in the production environment an entry will be written to the assigned file whenever a journal object is encountered in publication logic. You can use conditions and other logic to determine if and when an entry is actually written. Journals are normally generated within a publication, but you can also generate them at the start and end of the entire production job.

In the output file the entry itself is a new line or record containing the value(s) specified in the journal object. Each value can be separated with a user defined string if required or simply concatenated together.

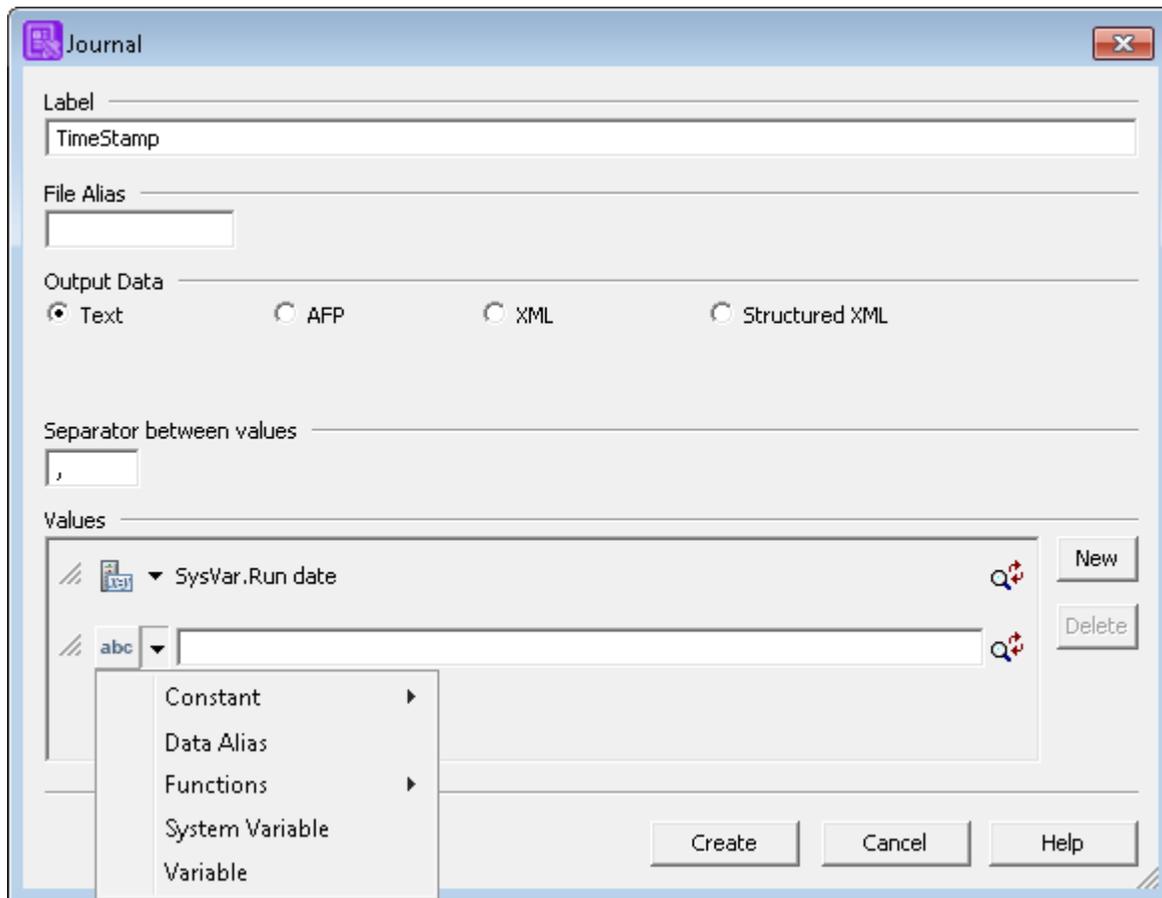
The target output files for journal entries are distinguished by the file alias you specify. Each different alias used in a publication design will produce a separate file in the production environment. You will need to specify the actual file references to receive the journal output when the publication is published. See [Publishing and deployment](#) on page 592 for details.

Indexes can be created in four formats.

- Text – a simple text record is written to the output file each time the journal is processed.
- AFP – the index is a structured file in IBM AFPDS format. There are four values per entry.
- XML – named elements are written together with their associated values to the journal in XML format.
- Structured XML – a predefined XML structure with a predefined schema. Where used this format automatically includes page offset and other key information for every output stream generated by the application. It can be supplemented with user defined values if required.

To create a journal object: mark the insertion point in the logic map and select Insert/Tools Objects/ Journal. The Journal dialog box is displayed. Specify the journal as required.

NOTE: Start and end of job journals must be specified in the Publication attributes – see the Journal Options in File Alias:



- This alias associates the entry with an actual file in the production environment. Make sure this matches aliases in any other journal object intended for the same file.
- The values specified are output to the journal file in one line or record, separated by this text (up to 8 characters).
- Change the order of values by dragging the handles up or down
- Select the type of journal required
- Create a New value and specify it. See [Working with values](#) on page 170 for details.

Working with XML Journals

When defining XML journal objects both element names and associated values are assigned. Note that the data structure is automatically defined in the Data Type Definition (DTD) embedded in the XML journal itself.

The screenshot shows a configuration window for XML journals. At the top, there are four radio buttons: 'Text', 'AFP', 'XML' (which is selected), and 'Structured XML'. Below these, there is a section for 'Element Name' with a dropdown menu showing 'ABC' and a text input field containing 'CustomerData'. To the right of the input field is a search icon. Below this is a 'Separator between values' section with a text input field containing a comma. Further down is a 'Values' section with a text input field containing 'Data.Boardwalk Dict.Account No'. To the right of this field is another search icon and a 'New' button.

Structured XML Journals

Unlike other types of journal, structured XML journals have a reserved design editor file alias of `SYS_XPJ`. This reserved alias is also used as the subsequent Publish and Generate OPS journal name. Only one structured XML file can be produced per application regardless of the number of journal entries defined in publication logic. The file automatically contains page offset and other production information for each output datastream generated by the application where appropriate. This information is supplemented with any values specified in journal entry objects associated with the structured XML journal. You can access the data stored in the journal by using the schema provided with the product distribution material. Note that HTML and eHTML output devices do not report Page Offset information, for these drivers this element is set to 0 in the journal. Refer to the Production Guide for detailed information on the structure of the journal.

A note about publishing to EngageOne Server - not applicable to Express Batch processing

The Names defined in the Designer Structured XML Journal must be placed in EngageOne Server Batch Journals as Output Variables. Refer to the EngageOne Server Administration guide for details.

When working with this type of journal you need to be aware of the following:

- Structured XML journals cannot be used with PCE.
- Duplicate names are allowed within a single and across multiple journal entries. However, these entries are invalid if used with an EngageOne application when:
 - duplicate names exist across multiple journal entries that do not share the same datatype.
 - duplicate names exist within the same journal entry.

- Multiple entries will appear in the journal when a journal object is placed inside a transaction table, loop or repeated data item. Only the first entry is processed if the journal is used with an EngageOne application.
- Journal data is cached to disk during the composition process based on memory options set in the publish wizard. Cached data must be processed at the end of the production job to create the final journal, this will have an impact on performance.

AFP Indexing

AFP indexes should only be specified in a Designer publication when at least one of the intended output datastream is AFP. The index information is in binary form and is intended for use only with third party software that can deal specifically with this indexing structure.

Format of AFP Indexes

When creating a journal that will be used as an AFP index, each entry must include four values that are used to provide parameters for the AFP index structure. These are:

Attribute	The name associated with entries of a particular type.
Attribute value	A value that distinguishes this entry from others with the same title.
Sequence number	A numeric value that indicates the sequential order of index entries.
Level	A numeric value that subdivides index entries with the same attribute title.

The values must be specified in the order shown and you may not omit any value. In the production environment Generate will use journals defined as AFP indexes in two ways:

- The journal file itself will be created as a self-contained AFP index containing all the entries generated by the job.
- For the AFP datastream itself an AFPDS Tag Logical Element (TLE) structured field containing the relevant index values will be included in each page generated by the appropriate publication.

Z/OS Note: When writing AFP Journal entries on Z/OS you must ensure that these are written to a Dataset with attributes of **U**.

22 - Interfacing with message applications

Content Author and Message1 are optional additions to EngageOne Compose that allow key parts of document content – messages – to be created, maintained and deployed independently of the main publication design. You can think of a message as a segment of page content that can contain text and graphics and make use of variable data available to the publications in which they are used.

In this section

Message areas and data fields.....	321
Importing the message environment file.....	321
Message boxes and areas.....	322
Message streaming.....	324
Resolving message data fields.....	325
Message signals.....	327
Production & preview	329
Customizing the audit trail.....	330



Message areas and data fields

Message applications use message areas to define the space available to particular groups of messages. In Designer you can map such areas to specific reserved space in document designs – message boxes – or alternatively message areas can be inserted directly as part of the main flow area of the document – message streaming. You will also need to map the data fields used in messages to specific values provided by your publication logic and – for Content Author only – provide a set of actions for signals if used. For Message1 only you must link the language settings used in the messages to the equivalent Designer locale.

Information about the message areas, data fields and signals is made available to Designer via a message environment file which is created by the message application.

Importing the message environment file

Ideally the administrator of the message creation environment will be the same person as the designer of the Designer publications intended to use the messages. Where this is not the case they will need to work closely to agree the available space for messages and to define the values that should be mapped to data fields.



Note: You may only include a single environment file in a Designer repository. This is assumed to contain all the data required in all publications that use messages. After the first import subsequent attempts are assumed to provide updates and will overwrite the information for areas and data fields defined in earlier imports.

To import a message environment file:

1. In the navigator select Import/Message Environment from the **Tasks** menu. Use the resulting dialog box to browse for the required environment file (.XML).
2. Confirm updates.

If you have previously imported an environment file any differences with the new file will be displayed. Click **OK** for the information to be updated.

Message boxes and areas

A **message box** reserves space in a document design for messages. You must assign one or more message areas to each message box to indicate the category of messages it can receive. Such **message areas** are created in the message application and specify the maximum dimensions of message content. The message box must always be large enough to accommodate the areas that have been assigned to it.

NOTE: Some types of message area allow individual messages to be repeated within the same document (multiple use). You cannot mix such areas with non-repeating areas within the same message box.

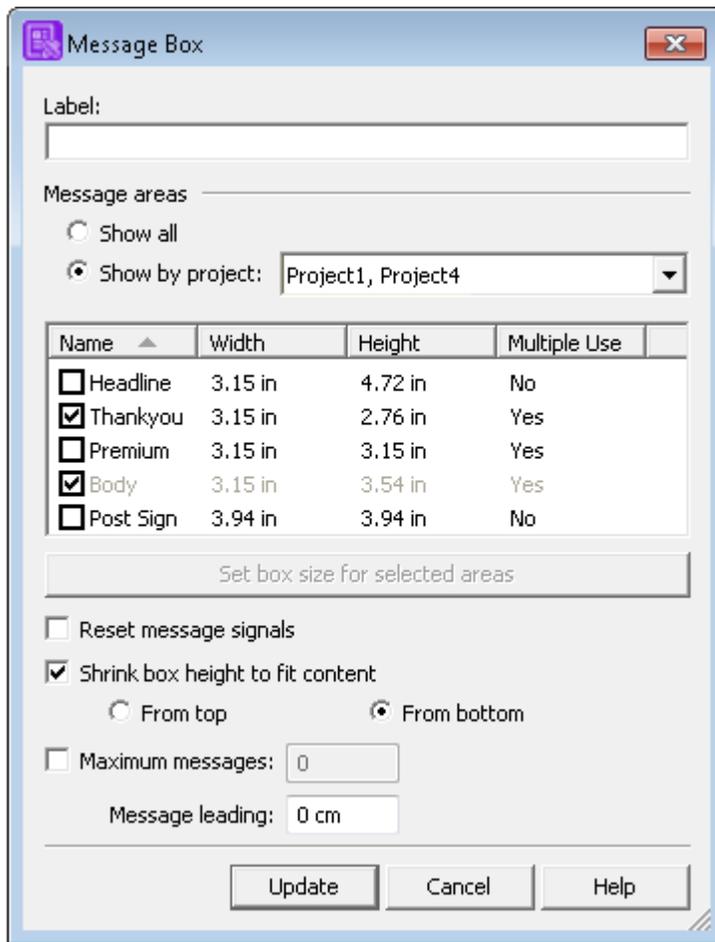
Message boxes behave similarly to text boxes. They can be inserted anywhere in a document design, for instance, in the body area, on the background, in headers and footers. Their attributes can be adjusted as with any other shape object. See [Working with graphics](#) on page 95 for details.

By default the width and the maximum height of a message box is fixed by the dimensions you specify. If required, you can use the **Shrink box height to fit content** options to specify that the height of the box should vary where the content is less than the specified height.



To create a message box:

1. Mark the required insertion point within the logic map.
2. From the **Insert/Presentation** Objects menu select **Message Box**
3. Create the box by clicking in the document design and drag out the box to approximately the required size. This can be adjusted later.
4. Specify message areas and customize. Double-click the message box object to show its properties
 - or –
 - right-click on the object and select **Properties** from the short cut menu.



- Select the message areas you want to include.
- You can filter available message areas by project.
- This automatically resizes the message box to fit the selected areas.
- Signal settings accumulate from previous message boxes and streams unless reset.
- You can limit the number of individual messages that will be included in the box.
- Specify the size of the gap to be left between each message.

Note that all previously selected areas are listed and any that are grayed out are from projects not in the current filter.

Message streaming

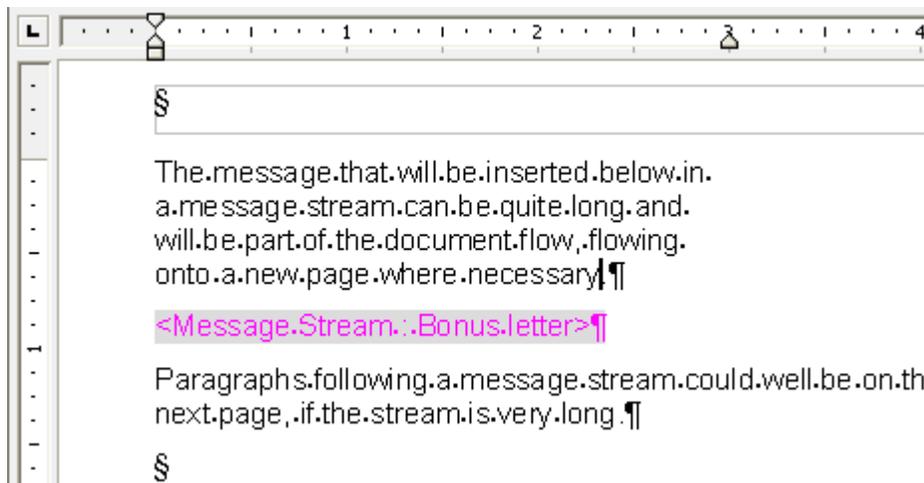
Message streams allow messages that have been created in Content Author to be inserted as part of the main flow area in a document. The content of a message area is inserted as a continuous, flow-able unit that will overflow onto new pages as necessary.

Formatting of the messages is as defined within Content Author, with the following exceptions:

- the width is that of the flow area, subject to any indents specified in Content Author
- there is no restriction on the height – the message stream will flow over as many pages as required.

Although the size of the message area is not relevant, the criteria for including an area is used.

Message streams can only be part of the main flow area; they cannot be inserted on the background, in Active Content or page setup, or in a text box or table.



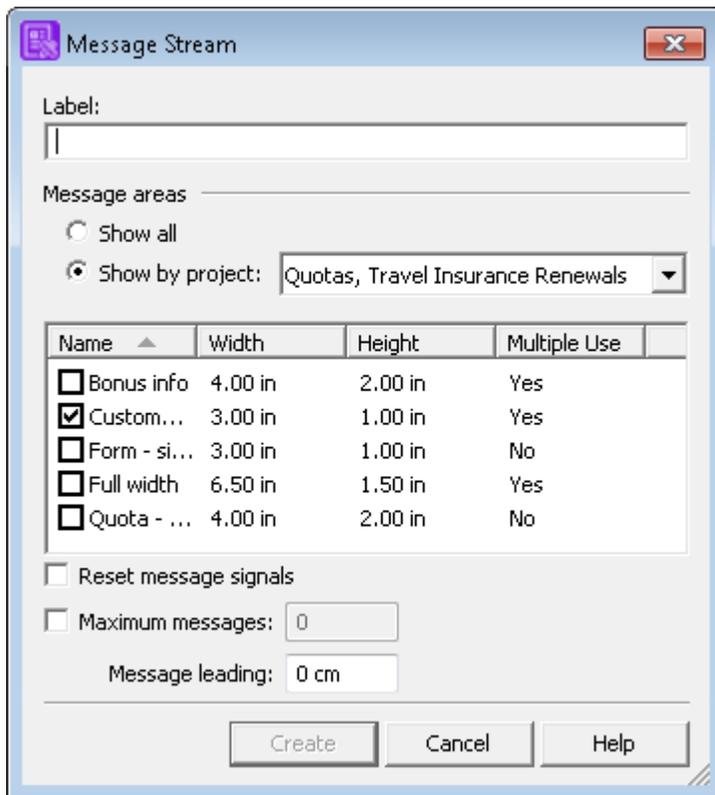
- Message streams can only be inserted between paragraphs, they cannot be part of one.
- The placeholder for the message stream does not reflect its eventual formatting. This will be as designed in Content.

To insert a message stream: from the **Insert** menu select **Presentation Objects/Message Stream** and use the **Message Stream** dialog box to configure it and select the required message areas.

Note that all previously selected areas are listed and any that are grayed out are from projects not in the current filter.

As with message boxes, you must map message data fields before messages are actually placed in the message stream. See [Resolving message data fields](#) on page 325 .

Note that if the cursor is at the start of a paragraph, then the message stream will be inserted before that paragraph, otherwise it will be inserted after.



- Reset message signals: Signal settings accumulate from previous message boxes and streams unless reset.
- Maximum messages: You can limit the number of messages that can be included in the message stream.
- Message leading: Specify the size of the gap to be left between each message.

Resolving message data fields

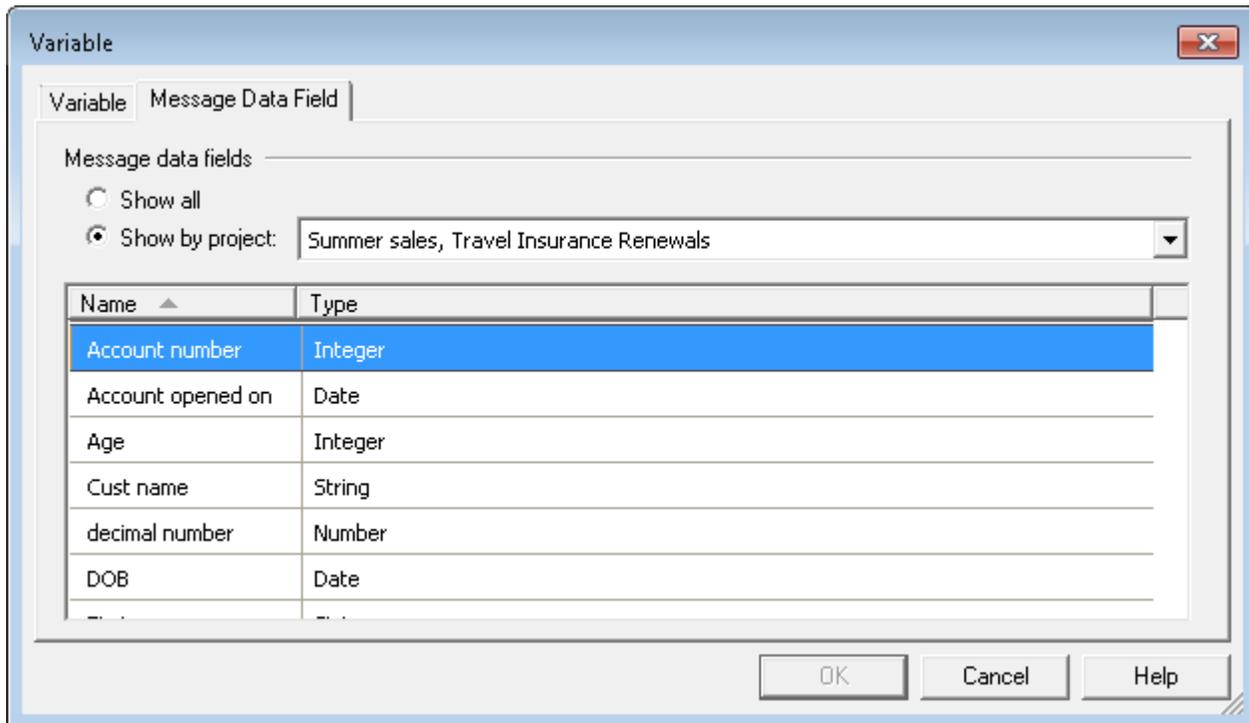
Messages often include references to values available to a publication at production run-time; these are known as message data fields. In a publication design you must ensure that each data alias likely to be used in messages is mapped to the required value using the Set variable action. In your publication logic you should ensure such mappings are defined before any messages are output but you can update them at any point in the logic (within a condition for example. Once the mapping is defined it will stay in force until a further action specifying the same data alias is encountered.

NOTE: See also the [Specifying files for Generate](#) on page 599 for Messaging options.

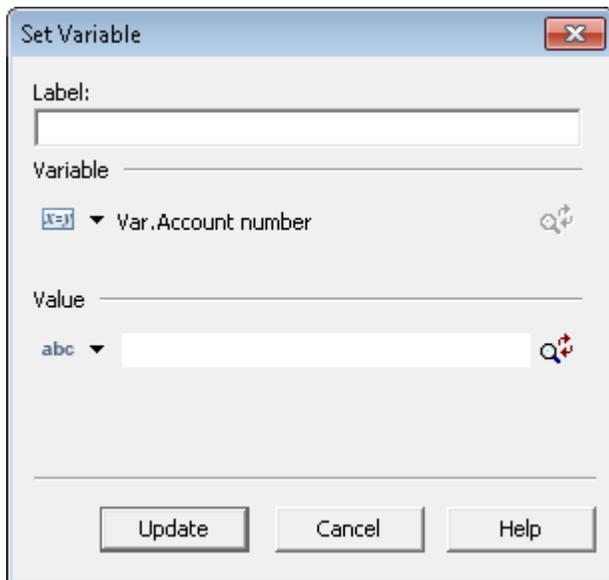
To map a message data field: mark the insertion point in your document logic and select **Insert/Set variable** from the menu:

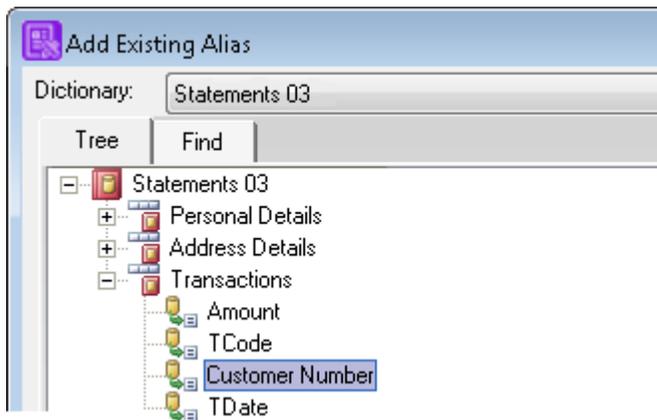
- In the Message data field tab, select the data field that you want to map and click OK.

- You can filter available data fields by project.



Specify the application data alias to be mapped to the message data field.





Message signals

Signals can be assigned to messages in the Content Author. A signal is an indication to the main Designer application that a particular message has been used within a publication and allows actions to be taken in response. This could be used for example, to pass information to inserter equipment (usually via an entry in a journal file) so that brochures, flyers etc. can be added to the same envelope as the printed document. A **message signal** object is used to test if messages with signals have been placed in the current publication and to perform the required actions.

Signals, which consist of a name and description, are passed to Designer in the **message environment file** and the **message signal** object can only be defined once the environment file has been imported into Designer see [Importing the message environment file](#) on page 321.

NOTE: As message processing is not performed by the Designer, only the **false** branch is enabled by normal logic flow. The **true** branch can be enabled by selecting the **Force/True** option from the context menu of the **message signal**.

A **message signal** group is similar to a **condition**. It has a true and a false path and can contain the same logic objects. Message signal objects should be included within publication logic after the appropriate **message box** or **message stream** have already been processed. Each message signal object queries if a particular signal has been set (or not set) in the current publication. You can include as many such queries as required within the logic.

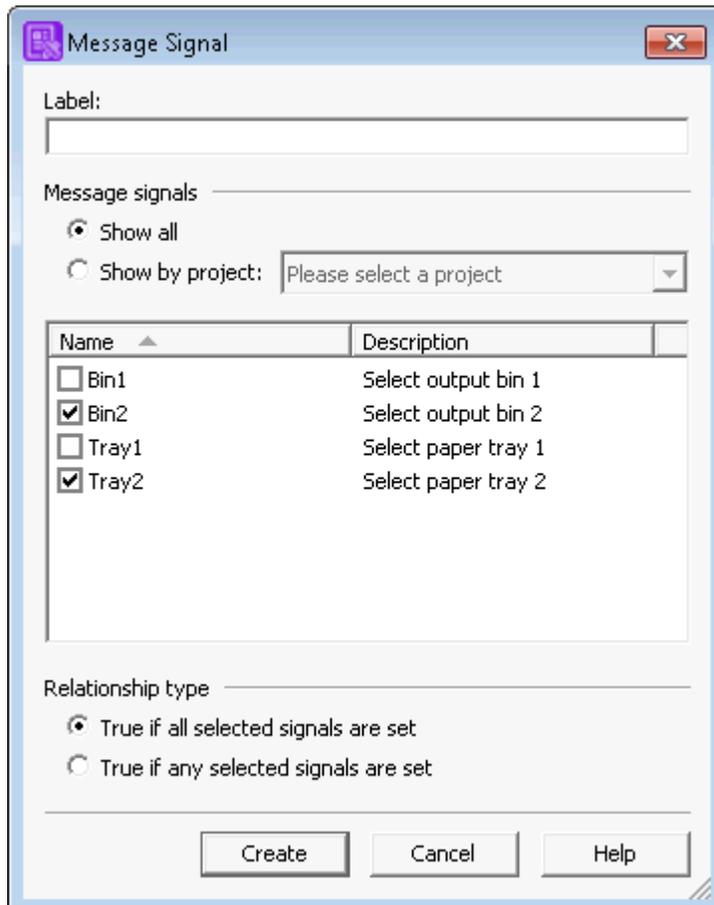
As message processing is not performed by the Designer, only the false branch is enabled by normal logic flow. The true branch can be enabled by selecting the Force/True option from the context menu of the message signal.

To define a message signal:

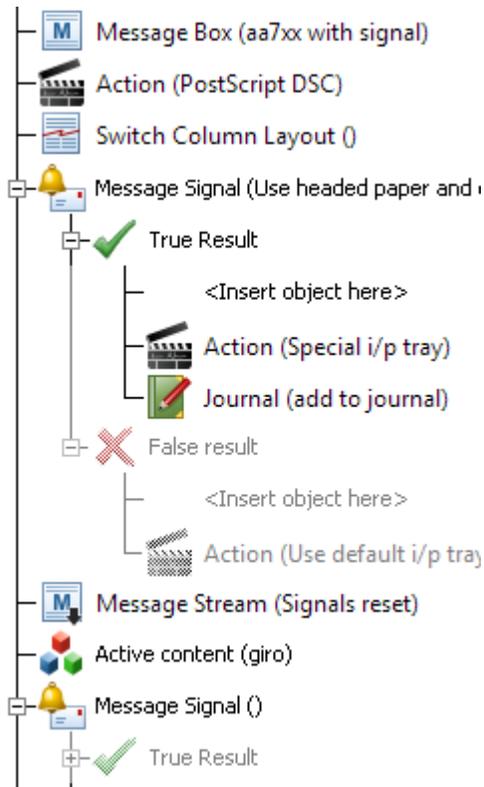
1. Ensure that the message environment file has been imported and mark the required insertion point in the logic map.
2. From the Insert/ Control Objects/ Condition menu select Message Signal.

3. Select the required signals for testing and choose whether all or any of them should be set.
4. Create the required logic in the True and the False paths.

As with conditions, to help when designing, you can force the True or False path to be active. This enables you to see the effect of each path in the condition. You can reset all paths in the document by selecting Insert/Control Objects/Condition/Reset All.



- If all or any of the selected signals are set in the placed messages, then the True path is taken.



- Message signals test messages already placed in a message box or stream.
- Signals are cumulative unless reset in a message box or stream.
- Signals can be tested any number of times or not at all. Testing can use preceding logic.

Production & preview

You can preview how the messages will look in your publication by using the advanced preview feature, see [Into production](#) on page 509. You can also preview messages from within the message application. You will need to generate a .HIP file (this must be a complete publication containing both design and resources) for the intended Designer publication in the normal way ensuring that the output file Output1 is set to PDF. Make both the .HIP file and the sample data file available to the message application. Note that for Message1 no other output devices should be specified.

When publishing and previewing, you will need to specify the Publish wizard options that are required for message applications including the file containing the latest messages and error handling options – see [Specifying files for Generate](#) on page 599.

Two system variables – **Mandatory messages and Optional messages** – allow you to monitor the status of messages that are available for placement within the document currently being processed and optionally activate appropriate logic. See [System variables](#) for details.

Customizing the audit trail

For publications that use messages Generate creates an additional log file known as the Message Audit Trail. This is similar to a regular journal file but is specifically and automatically used to log messages that have been included within documents. In the Designer you can assign a field that will be additionally included in the audit trail entries along with the automatically generated references to messages. This is intended to allow the user to generate a reference that can be used in the audit trail to identify documents uniquely – typically a field containing a customer reference or similar. The **Set message audit trail** option can be re-specified at different points in the logic if required.

To specify the unique reference for the audit trail: mark the insertion point in your document logic, (ensuring that this is before any messages are placed by the logic – i.e. before the first message box) and select **Insert/Control Objects/Action** from the menu. In the **Action** dialog box select the **Set customer ID** option and assign the required field or value.

23 - Interfacing with Vault

The Vault environment caters for high speed storage and retrieval of electronic documents. Vault itself is the main repository in which documents are stored and maintained. Various other Vault products provide access to the repository.

Liaison is an extension to the core Vault environment that allow web based display of billing data to end users, optionally with payment facilities. While it accesses the Vault repository, Liaison has its own management systems.

Vault can accept documents from several sources and has a variety of methods for loading the data and associated resources.

In this section

Working with Designer documents.....	332
Creating the index.....	332
DIJ settings for use with Vault.....	333
Table of contents.....	334



Working with Designer documents

For documents created by Designer applications Vault always requires the following resources:

- The actual documents to be archived as part of a standard output datastream file created by Generate
- A DIJ to act as an index for the documents also created by Generate
- The resources required to present the documents delivered within the appropriate HIP file (as created by a Designer publishing task).

For information about loading these files into Vault refer to the Vault Users Guide.

If the archived documents are also to be used within an Liaison environment the application also needs to provide vendor configuration data which is stored within the DIJ. Normally the data required can be created as part of the relevant publication object within the Designer.

Creating the index

The Vault repository has a very fast access method that relies on a well constructed index into the documents stored in the datastreams it holds. For documents generated by Generate this index is provided by a Document Interchange Journal (DIJ).

Each record within a DIJ specifies a unique key for each document by which it will be referenced in the Vault environment (usually an account number). A DIJ can also provide additional information about the customer account to which a document belongs, to provide search data for use with Vault client systems.

Custom presentation

If required, you can also write publication data to the DIJ for custom presentation by Liaison. When you use the **Store Data Set** option, the input data destined for each document is copied to the DIJ along with the regular index information you have requested. Contact your product supplier for further information if you want to use this option.

DIJ settings for use with Vault

This list shows how the information in a DIJ can be used by Vault.

Data tab	<p>These parameters are used to identify individual publications within the repository. Together they provide the primary index key within the Vault repository.</p> <p>Account Number – a unique reference number for the intended recipient of the publication (usually an account number).</p> <p>Name – a name that uniquely identifies the application generating the publication.</p> <p>Statement Date – a date that uniquely identifies the version of the publication</p>
Publication ID tab	<p>Use this section for unique identification of the company (vendor) and type of publication to which documents relate. Under normal circumstances the Default setting, using generic settings, will provide the information required.</p> <p>Master ID – you may need to specify this if your Liaison implementation uses a custom payment type.</p> <p>Read from the PSE file generated by Liaison that contains the relevant ID settings.</p>
Options tab	<p>City, State/Region, Zip/Post Code, Country & Phone – these provide additional search options within Vault client systems.</p> <p>Payable – select this option if the documents are to be used with Liaison.</p>
Address tab	<p>Use this to provide the address data related to the account to which each document belongs.</p>
Custom settings tab	<p>Use for additional document index information that may be required where custom versions of Liaison are in use.</p>

Table of contents

For extended document types you may want to include a table of contents that is available when documents are viewed from the repository.

For more information about creating bookmarks see [Bookmarks](#) on page 152.

Where used the structure and labels used in the **TOC** are specified by using **Bookmark** objects in the logic map of document designs (i.e. they are not configured in the DIJ dialog box). At production run-time, the **TOC** information will be automatically generated and stored in the DIJ if bookmarks have been used. **Bookmarks** can be nested to create the required structure for the table of contents.

24 - Interfacing with EngageOne Deliver

The EngageOne Deliver system manages **e-mail** and SMS customer communications. It allows organizations to design content for both these in a single design environment. It enables organizations to allow their customers to communicate at any time, maintaining an up-to-date customer-centric view at all times throughout the organization. It can also help organizations to maintain a brand consistency.

The DIJ is used as the main control file and contains information, such as customer account numbers, which references the corresponding publications in the output datastream via an EngageOne Deliver Identifier. See [Document Interchange Journal](#) for more details.

In this section

Datastreams available for EngageOne Deliver.....	336
DIJ settings for use with EngageOne Deliver.....	336



Datastreams available for EngageOne Deliver

The following output datastreams are available for EngageOne Deliver:

eHTML	For customers wishing to view their information as rich text. Use the eHTML output device to create output intended for email.
Linedata	For customers who prefer to view their information in plain text and on devices that can only support plain text. This driver can also output other text ready for insertion into other streams and web pages by EngageOne Deliver. Use the Linedata output device to create output intended for SMS

DIJ settings for use with EngageOne Deliver

This list shows what information in a DIJ can be used by EngageOne Deliver.

Refer to the EngageOne Deliver Reference Guide for details.

Data tab	Use all fields to identify the documents to be sent in the messages. Statement Date should be the date the message is to be sent, as it will appear in the e-mail header.
Publication ID tab	Not required – use the Default setting.
Options tab	Not required, but can be used for extra address fields.
Address tab	Address fields can be included as required.
Custom settings tab	Contact your product supplier for information on these settings.

25 - Publishing content for EngageOne Communicate

From Designer, you can:

- Publish documents to your EngageOne Communicate account, these documents can be included as email attachments to communications within EngageOne Communicate.
- Share content with EngageOne Communicate to be used in campaigns using mobile templates. Tagged content can be published for use in EngageOne Communicate.

To make use of the features described above:

- You must have an EngageOne Communicate account in place. Contact your support team for details on account registration.
- Use the Connections feature to allow for integration between Designer and Communicate; refer to [Connecting to your EngageOne Communicate account](#) on page 338 for detailed information.

In this section

Connecting to your EngageOne Communicate account.....	338
Dynamic attachments	342
Working with tagged content.....	343
Version Label and Audit Trail.....	346



Connecting to your EngageOne Communicate account

Before you can publish for Communicate, you need to define a connection to your EngageOne Communicate account. You can set up multiple Communicate connections, allowing you to select a specific connection for publishing.

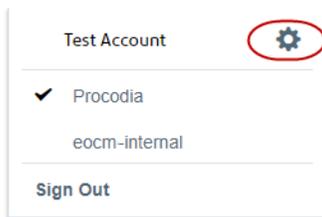
To create a connection, log into your Communicate account and copy the account credentials; use these credentials to create the connection in Designer.

Retrieve Communicate account credentials

Log into EngageOne Communicate to retrieve account settings required for defining your connection.

To create a connection :

1. Log into your EngageOne Communicate account.
2. Click your account profile and select the cog icon, as indicated below:



Note that the cog icon is only available to users who have team admin permissions.

3. From the Account Settings page, click on **Client Credentials**, copy **NAME**, **CUSTOMER ID**, **CLIENT ID**, and **SECRET KEY** values for subsequent use in Designer.

Teams Users **Client Credentials** Sender Email Addresses

Procodia
eocm-internal

Client Credentials

Refresh Add Credentials

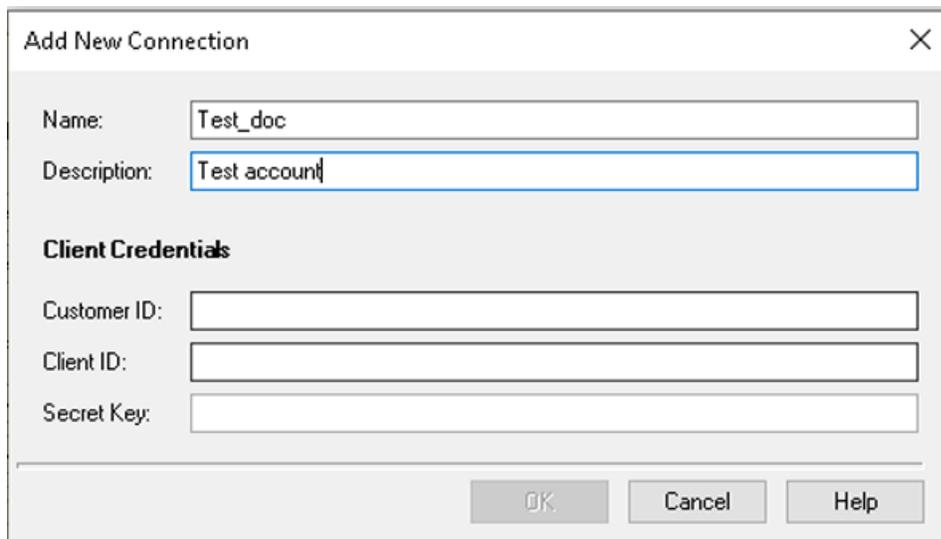
NAME	CUSTOMER ID	CLIENT ID	SECRET KEY
Designer_Doc	sa38fa9	*****	*****

Creating a connection

If you wish to publish to Communicate, you must first establish a connection to EngageOne Communicate in Designer. This section outlines the steps you need to follow to create connections between Communicate and Designer.

To define EngageOne Communicate account settings :

1. From the **Administration** menu, select **Connections**.
2. The Connections page is presented; click **Add**, you are presented with the Add Connections dialog:



The screenshot shows a dialog box titled "Add New Connection" with a close button (X) in the top right corner. The dialog contains the following fields and sections:

- Name:** A text input field containing "Test_doc".
- Description:** A text input field containing "Test account".
- Client Credentials:** A section header followed by three empty text input fields:
 - Customer ID:**
 - Client ID:**
 - Secret Key:**

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Name- mandatory; you can use the name defined in Communicate or enter a different value, if appropriate.

Description- optional.

Client Credentials - enter values obtained from your Communicate account as described in [Retrieve Communicate account credentials](#) on page 338.

3. Click OK. The Connections page is displayed. The connection is verified; a green tick symbol is displayed in the **Status** column for valid

The screenshot displays the 'Connections' and 'Connection Information' sections of the EngageOne Designer interface.

Connections

Name	Description	Customer ID	Client ID	Status
Test_Doc	Test account for docu...	sakal11	sakal11 moulbar@honyl@communicate.engageone.co	✓

Buttons: Add, Remove, Refresh

Connection Information

Details | Groups and Users

Name	Type
Administrators	Group

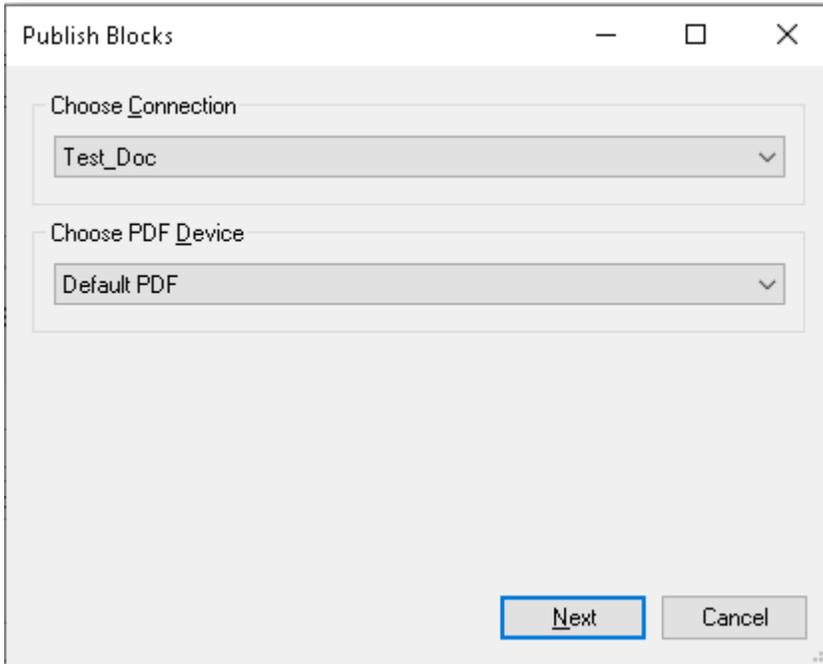
Buttons: < Add, Remove >

Name	Type
Administrator	User
Everyone	Group

When a new connection is created, it is automatically assigned to the Administrators group; you can change this in the **Groups and Users** tab.

Selecting your Communicate connection

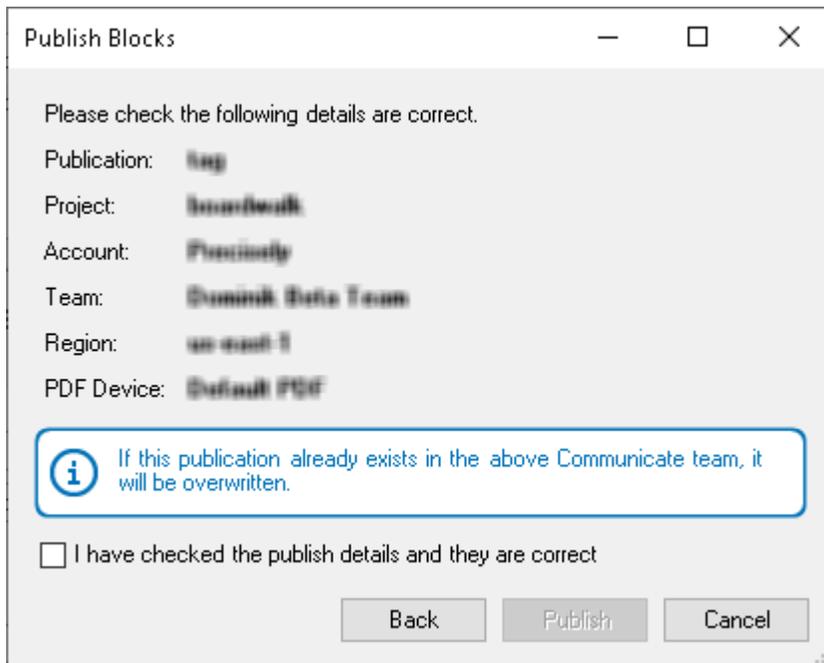
When publishing for Communicate, choose the connection and PDF device; note that Designer authenticates the connection during publishing:



The screenshot shows a dialog box titled "Publish Blocks". It has a title bar with a minus sign, a maximize button, and a close button. The dialog contains two dropdown menus. The first is labeled "Choose Connection" and has "Test_Doc" selected. The second is labeled "Choose PDF Device" and has "Default PDF" selected. At the bottom of the dialog, there are two buttons: "Next" and "Cancel". The "Next" button is highlighted with a blue border.

Note: When publishing blocks, the selected PDF Device is only used by the Preview method provided by the EngageOne API; it has no effect on the tagged blocks published to Communicate.

Click Next to authenticate using the credentials configured for the selected connection; this will also query the account and team information which will be displayed on the next page.



It is important to note that if a publication with the same details exists in Communicate, it will be overwritten by the latest publish.

Dynamic attachments

When publishing attachments to EngageOne Communicate, you will need to:

- have previously created a connection to EngageOne Communicate.
- be aware of non-supported design objects for this feature; these are:
 - External keyed images
 - Lookup tables
 - Custom plug-in chart implementations
 - Journals
 - Message areas & streams
 - Fields with the attribute **External RTF filename**.
- be aware that publication name is used as the attachment name in EngageOne Communicate.

Publishing attachments for EngageOne Communicate

Before you publish attachments you must have created a connection to EngageOne Communicate; refer to [Connecting to your EngageOne Communicate account](#) on page 338.

To publish attachments:

1. In the navigator, browse to and select the publication to be published.

From the **Tasks** menu, select **Publish Attachment for Communicate**.

2. Specify Publish Preferences.

- If you have multiple connections for your Communicate account, you are required to select the company you wish to work with.
- If you have multiple PDF output devices configured in the project, you must select the required output device.

Note: Where there is only one connection and a single PDF output device is configured, this step will be skipped.

3. Confirmation - click the confirmation checkbox.

If you are satisfied with the details presented; this will in turn enable the **Publish** button.

4. Click **Publish**.

This will publish a template package created from a single publication in a single ZIP file to the Communicate account.

Working with tagged content

You can share paragraph content associated with specific design objects in your publication with EngageOne Communicate using the **Tag Block for Communicate** logic map option. Use the **Preview Blocks for Communicate** option to preview the tagged content. When finalized, use the **Publish Blocks for Communicate** to push the shared content to your Communicate account for use in EngageOne Communicate.

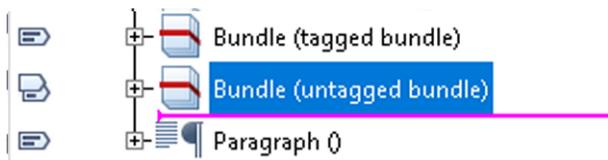
Tagging for content is supported for the following design object types:

- Paragraphs
- Bundles
- Conditions
- Case structures

- Active content references
- Active content
- Text Boxes
- Table cells
- Sections

Note: only paragraph content is shared not the design object itself.

Tagging is performed from your publication's logic map; refer to [Tagging content](#) on page 344 for detailed information. Icons identify content that has already been tagged in the logic map.



Where:

 - collapsed tagged item with tagged child items

 - collapsed untagged item with tagged child items

 - tagged item

Tagging content

Note that duplicate names for tagged blocks in the same publication, public document or active content are not allowed.

To tag publication content :

1. In the publication's logic map, hover over and right-click the design object to tag.
2. From the context menu, click the **Tag Block for Communicate** option.
3. From the **Tag Block for Communicate**, assign a name for the tagged block.

A list of existing tag blocks is displayed in a grid format.

Note: Where a tagged block has previously been defined, the **Tag Block for Communicate** option is replaced with **Edit Communicate Block Tag** and **Remove Tag** options.

Previewing tagged content

To preview tagged content:

1. In the navigator browse to and select the publication containing tagged blocks to be published.

Note: you can also perform the step that follows from within the publication.

2. From the **Tasks** menu select **Preview Blocks for Communicate**.

HTML preview files are created based on the sample data assigned to your publication.

Publishing tagged content

Before you publish tagged content you must have created a connection to EngageOne Communicate; refer to [Connecting to your EngageOne Communicate account](#) on page 338.

To publish tagged content :

1. In the navigator browse to and select the publication containing tagged blocks to be published. Note that you can also perform the steps that follow within the publication.
2. From the **Tasks** menu select **Publish Blocks for Communicate**. Enter your sign-in credentials to access your EngageOne Communicate account.
3. The content tagged in the Designer editor will be available for inclusion in EngageOne Communicate.

Version Label and Audit Trail

After publishing to EngageOne Communicate, a version label is created for the publication resources and the PDF output device used. You can retrieve historical published resources by branching the project based on the label. Note that branching on the publication will not include the output device, as there is no production job associated with this type of publishing.

The label description includes details such as the ID and publish time that match the corresponding asset in Communicate. These details are also recorded in the audit log entry when the Designer Audit Trail is enabled. Refer to [Version control](#) on page 132 for detailed information.

26 - Working with EngageOne Video design objects

EngageOne Video is an optional addition to EngageOne Compose that allows for the creation of personalized, data driven video content that can be included in the design of your document.

In this section

Workflow.....348



Workflow

In Designer

- Links to personalized video content, identified by Personalized URL (PURL) data aliases, can be added to a final customer communication using the following document editor objects.
 - EngageOne Video (QR Code) barcodes, refer to [Barcodes](#) on page 238.
 - Clickable images, refer to [Using images](#) on page 99.
 - URL links, refer to [URL links](#) on page 93.
- When the document design has been finalized the publication is published for EngageOne.

Note: the following recommendations:

- Default PURL values should be set to a specific URL. This value may be reviewed and changed during the development of the template. When working with EngageOne Video QR barcodes, you should ensure that the length of the default URL should be as representative as possible.
- Add document logic to cater for the default PURL not being changed, which may indicate an undetected production problem. This could be achieved with a simple Condition object that compares the PURL data alias with a constant containing the default PURL value. An alternative value can then be chosen by the Compose user or, an Action object used to abort the production run may be invoked.

In the EngageOne Compose

- The publication is imported into EngageOne Server
- For templates using Interactive Data, PURL data fields can be automatically populated using video and data mappings a template managers assigns to the publication.
- For templates not using Interactive Data, or those used in non-accumulated batch scenarios, the PURL data must be populated using your existing Extract, Transform and Load (ETL) mechanism. Refer to the EngageOne Video PURL generation section in the EngageOne Server Administration guide for further information.

27 - Creating EngageOne templates

EngageOne is a multi-channel solution for organizations who wish to create an enterprise communication service as an integral part of their software and operational infrastructure.

In this section

- EngageOne - operating modes.....350
- Working with interactive documents.....351
- Working with multiple languages.....354
- EngageOne data.....355
- Prompt groups.....386
- Building the EngageOne template.....391
- Previewing an EngageOne template.....397
- Publishing for EngageOne.....400
- Import and exporting interactive data definitions.....415



EngageOne - operating modes

EngageOne allows you to generate relevant communications through a mix of batch and real time processes delivered through your customer's preferred delivery channel using one of the following modes:

Interactive

Interactive allows for the creation, control and delivery of ad hoc customer correspondence, such as letters and offers, based on a standard Designer publication templates. In this scenario the designer creates a template which will be launched and interactively customized by a front-office user before delivering the document. Data can be supplied to your interactive communication from your existing business system on a push/pull basis and can be modified by the end-user via data prompting. These features are governed by the design of your template and associated data model.

Interactive is delivered as an "out of the box" application or as a collection of web services together with the editors to allow for data to be supplied to the interactive communication; this allows for flexible deployment. Loose integration and lower cost of ownership can be achieved with the standard application and the data push capability. More streamlined integration can be implemented using the web services and together with your organization's chosen editor type to tightly integrate the interactive features into your core business systems.

A note about editor types

Interactive can be configured to use multiple editor types, this is configured during the installation of EngageOne. For details about the various editor types, refer to the EngageOne Compose and EngageOne Server Installation guides.

EngageOne Batch

Your decision you use EngageOne Batch over the Designer/Generate batch solution will depend on your organization's flexibility and performance requirements.

EngageOne Batch allows you to vary template and delivery options for each customer record and includes the ability to use different template versions through the Effective date feature. EngageOne integrates with Vault and EngageOne Deliver and configuration is made using the Administration client. The Designer/Generate solution offers greater performance but is tied to a single template.

Applications that are more appropriate for EngageOne high-speed batch are for example, account statement applications with high-volume requirements and small Service Level Agreements.

EngageOne On-Demand

EngageOne On-Demand allows you to deliver communications in real-time for business processes that require this. Some examples include account activities through your portal or self-service options. On-Demand uses the standard delivery option features of EngageOne which caters for all

multi-recipient/channel output in forms other than realtime. For example, an archive copy may be queued up and processed later in a batch job.

Working with interactive documents

The layout and style of an ad hoc document may be very different from the designs used for batch applications but within Designer the files and objects used are fundamentally the same. It is only at publishing time that a publication is identified as an EngageOne resource and results in a template file being created for use with the EngageOne environment.

ANTERA CORPORATION
FINANCIAL SERVICES
Suite 300, Indianapolis, IN 46283
1-828-0615

January

Type the customer name
Fred Jones

Fred Jones
111 Some Street
Some City AZ 55555

Account Number: 11111

Dear Fred Jones:

We advised you in a letter dated 10/15/2007 that we have a reply from Some Merchant. Therefore the credit(s) for \$6.44 previously issued would remain on the account.

We have now received support of the charge(s). A copy of the supporting information is enclosed. We have therefore removed our existing credit and rebilled your account for \$6.44. The debit(s) will appear on your 09/15/2007 statement.

Finance charges will be adjusted accordingly.

Type free form text or delete.

It is a pleasure to be of service to you 24 hours a day, every day of the year. If we can be of further assistance, please call us at the phone number on your statement to speak with a Customer Service Representative.

Sincerely,

Valid Signature

Your Name
Your Title

At the design level the primary difference between a standard Designer batch publication and one intended for interactive communication is the use of an interactive data definition. This replaces the data format used to define input data for non-interactive publications. The interactive data definition defines each piece of variable data required by an interactive template and specifies whether the value should be provided directly by the user (prompted) or automatically from a linked external system. The dictionary and data map is created from this interactive data and linked to the template when it is created.

Design considerations

It is important to note that the design of the data model and template will have a significant affect on the end user experience. For this reason you need to be aware that the size of the interactive data model can have a significant impact on the performance of the application for both the design and interactive user. Best practice recommends the use of different data models for collections of templates so that the number of fields in a single data model is not excessive. Data models should not contain more than 500 fields.

Interactive variables

Interactive variables are another type of variable data that can be included in the design of your interactive document. This type of variable information allows the template designer to prompt for data that is not part of the data model and does not originate from your business system. Refer to [Interactive variables](#) on page 362 for further information.

Interactive Variables are added to the template by users with the appropriate role entitlement to cater for the additional client data without changing the data format. The use of Interactive Variables in this situation avoids the need to make changes to the data format which could have an impact on a large number of templates. Note that Interactive Variables can also be used in Public Documents and Active Content. However, they cannot be used within Active Content that is published independently or within the **Show when** condition of documents.

Editable text

Where more flexible content is required you may also specify that the user is directly able to amend or insert parts of paragraph text. To the user this is known as editable text. Refer to [Editable Text](#) on page 393 for further information.

Active Content

Active Content also plays an extended role in interactive applications. Active Content provides the ability to create pre-approved or optional content for user selection. The approach used by EngageOne is to allow the designer to define and organize this optional content in the context where it will be used. Active Content Groups are used to create the collection of optional content choices for the interactive user.

Active Content, when shared and independently published can increase agile changes into your template design process. When marked as publishable, you can deploy just the changed Active Content and not have to republish an entire template or set of templates. The benefits of using Active Content in this way are, reduced risk and testing effort due to the limiting the scope of change.

Additionally, organizations may often want to maintain many different EngageOne templates with similar content so using shared Active Content can allow changes to be easily distributed. For full details of working with Active Content see [Active Content](#) on page 428.

Deploying resources

Designer publishes all resources needed for delivery using the set of devices that have been previously identified for EngageOne. Such resources are placed within independent files that must be created and deployed in conjunction with requests from the EngageOne server. This architecture allows EngageOne to re-use previously deployed resources wherever possible. You can also publish resources for EngageOne Server from Designer into the same deployment package by selecting the

Publish for EngageOne menu option. For more information see [Publishing resources for EngageOne](#) on page 411.

Previewing

Before you deploy new or amended EngageOne resources to the server a Preview for EngageOne function allows you to see a template as it will appear to the user; i.e. loaded in a web browser and with interactive data prompts displayed.

Interactive data definitions

You can create a multi-lingual interactive data definition from your own files by importing your XSD together with language specific XFORM files. Refer to [“Import and exporting interactive data definitions”](#) for further information.

Working with multiple languages

To cater for users who work in different languages, you have the option to localize certain text that is presented to the front office user as follows:

Designer uses the concept of locales to allow control and customization of language variations at paragraph level in your document's design. The front office user can select the required language from the Document Language option in the interactive editor. Refer to [About locales](#) on page 311 for further information.

Localization options for data prompting

When creating your interactive data definition you have the option to configure localized prompt, help and choice details as follows:

- Prompt and help text for Active Content objects, fields in the interactive data definition and Interactive Variables. Refer to [Using Active Content with EngageOne](#) on page 443, [Using the Interactive Data Editor](#) on page 356 and [Previewing an EngageOne template](#) on page 397 for further information.
- Choice labels used when presenting specific options to the user for a particular data field. Refer to [Interactive data editor/Interactive Variable field properties](#) on page 363 for further information.
- Prompt and help text for free form editable text as defined in an editable container. Refer to [Editable Text](#) on page 393 for further information.

This allows the front office user to select the most suitable language to view prompts and also provides the ability to choose the language in which they wish to view localized paragraph content.

Using the Interactive Data Editor

The interactive data editor (IDE) is used to define all data that is required for the EngageOne document, including:

- data gathered from the user via on-screen prompts, either on an individual basis as they appear in the document logic, or grouped together in a single prompt dialog box.
- data supplied directly from external sources such as system supplied data, to be used directly in the publication.

Note that these attributes are only used to drive the interactive prompting behavior in the edit data screen in EngageOne Interactive and the interactive editor. These attributes change no behavior in Generate.

When you define fields these can be included in **groups** containing several fields of hierarchical data for defining prompted fields, or **repeating groups** of system supplied data. You can specify default values for fields within repeating groups for an unlimited number of repetitions of the group.

You can use the IDE to define annotations for any data field provided they comply with the W3C XML schema annotations standard. Annotations allow you to pass additional information through the EngageOne schema to your system data integration program. For more information about integrating external system data see the EngageOne Programmer's Reference Guide.

Creating an interactive data definition

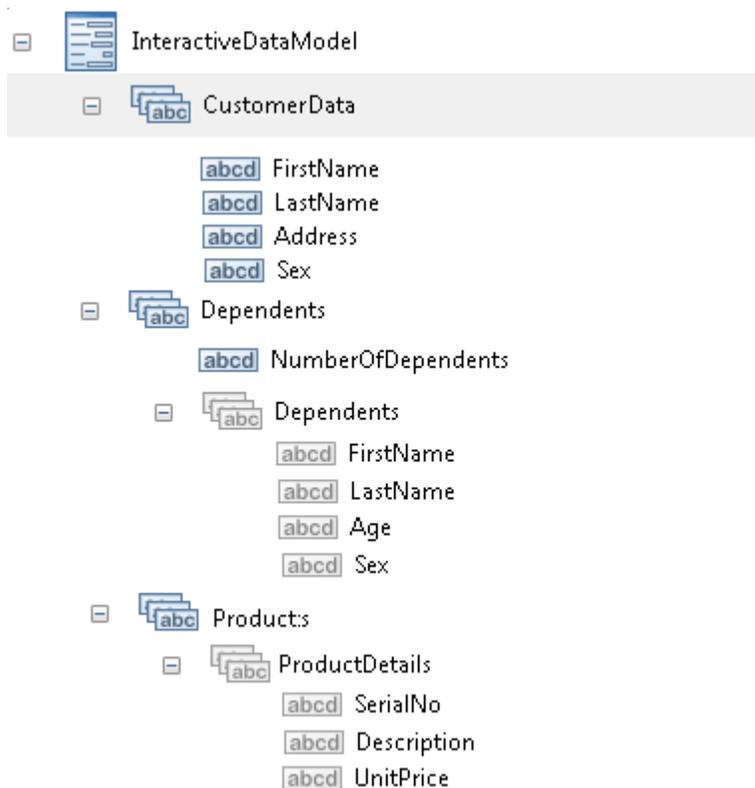
In the navigator browse to the required project or folder and on the **File** menu select **New Interactive Data**

-or-

right-click in the navigator and select the **New Interactive Data** option. The main window of the IDE is invoked.

Best practices around designing your data model recommend that you typically create parent data elements to organize your data.

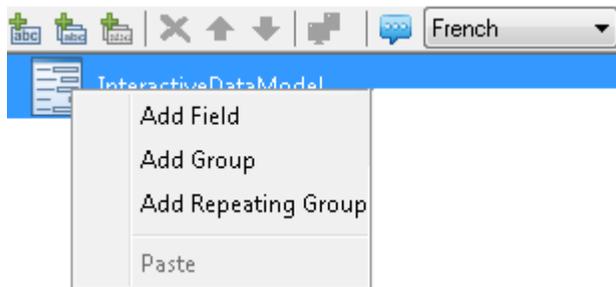
The example below illustrates a hierarchical method of organizing your data. CustomerData, Dependents and Products are parent data elements under which data fields and repeating fields have been created.



Adding fields and repeating groups

To add fields, groups and repeating groups:

1. Use the add buttons on the toolbar– or –right-click on the **Interactive Data Model** button to display the context menu. Add fields as required.

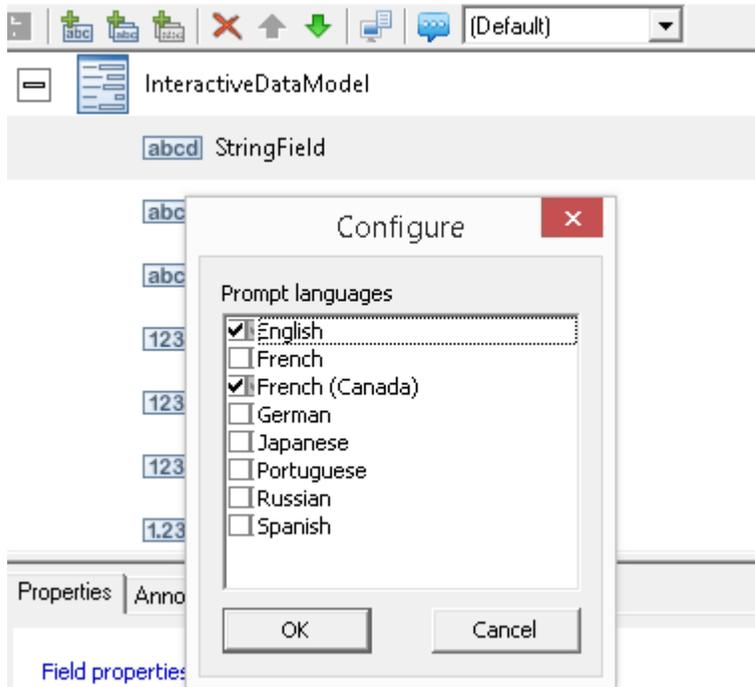


- Add a new interactive field for groups, or repeating groups.
 - Add a group.
 - Add a parent placeholder for a group of several fields.
 - Add a parent placeholder for a repeating group.
2. Use the **Field Properties** pane to define the properties and values of the currently selected field. For details see [Interactive data editor/Interactive Variable field properties](#) on page 363.
 3. Click **OK** after completing each field definition.

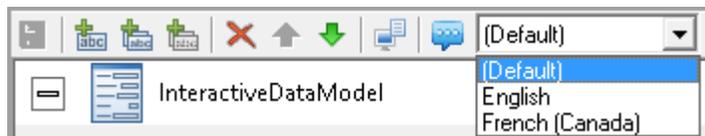
To define available languages, click on the Configure Languages toolbar button and choose the required languages for the interactive data definition. At this point the chosen languages will be available in the Prompt Language drop-down list.

Configuring prompt and help text

The IDE allows you to configure prompt and any associated help text for a selected field in your interactive data definition. You can optionally use the **Configure Prompt Languages** button , to select the languages you wish to use when working in a multi-language environment.



When you have chosen the required help and prompt languages, you can then select a specific prompt language to work with, as shown below:



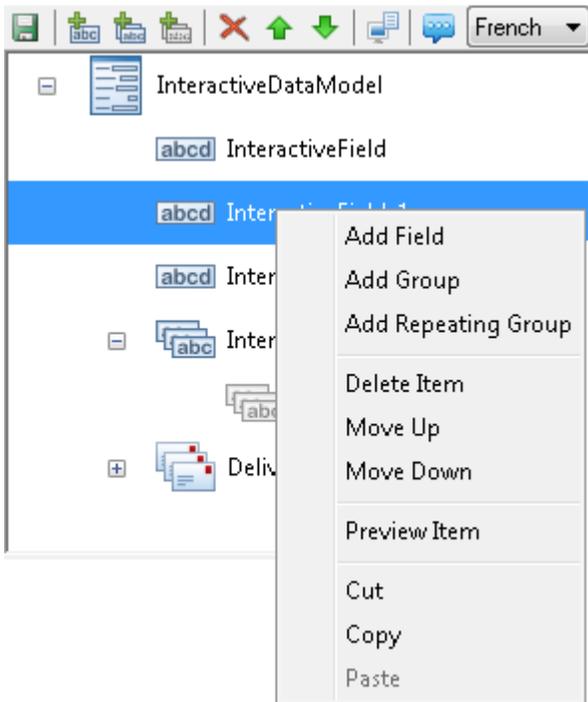
For the selected field, enter the prompt and help text in the **Properties** tab and click **OK**.

Note:

- If your language is not available from the drop-down list you should choose the **Default** option.
- If you have upgraded from an earlier version of Designer that does not support localized prompts, any existing text will be assigned to the **Default** language setting.

Re-ordering fields, groups and repeating groups

Adjust the field list as required by using the toolbar– or –use the context menu by right-clicking on any **Field**, **Group** or **Repeating Group**.



- Options include: **Cut** and **Paste** to move groups and fields as required on the list and **Copy** and **Paste** to copy items and add them either to a position within the same level, or to a different level or group. A copied item will take the original name plus an appended number. Rename this as required in the **Field Properties** pane. Use the **Move Up** and **Move Down** buttons or the context menu to move the selected field up and down the list view (within the same level in the tree). You can also drag and drop fields to move items one position up and down the tree.
- Note that the **Delivery Information** fields are predefined and can not be renamed or deleted, and the **Type** can not be changed.

Use the **Properties** tab to define the properties and values of the currently selected field from the field list. For detailed information about the fields available on the properties tab refer [Interactive variables](#) on page 362.

Once all interactive fields have been defined click the **Save** button on the toolbar to save the IDE and interactive data definition. Asterisks indicate required fields.

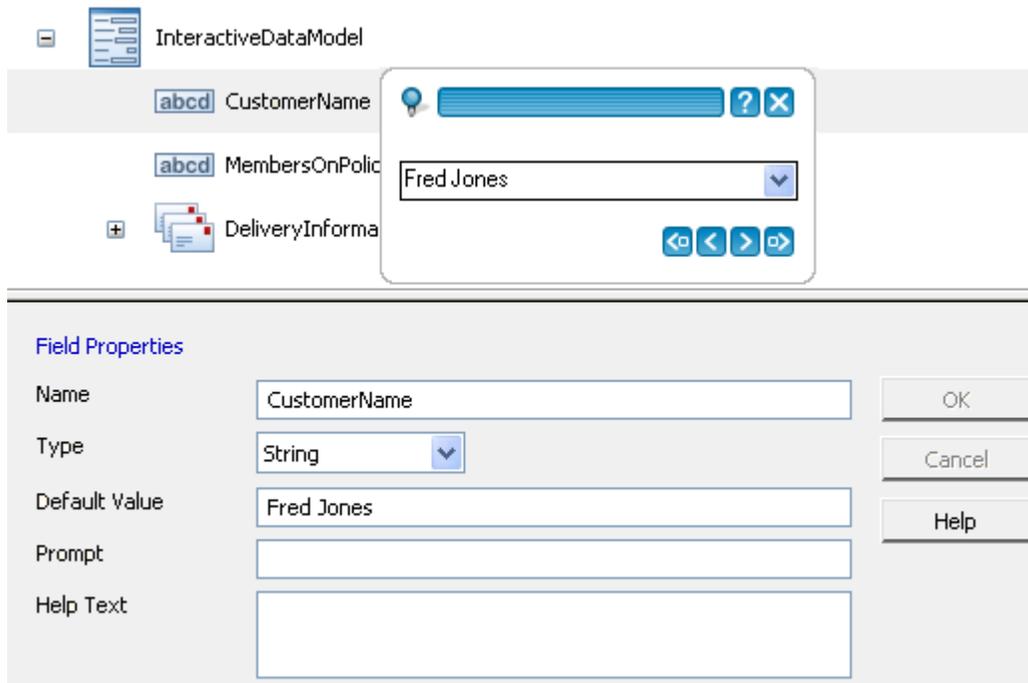
Editing and previewing IDE definitions

To edit IDE definitions: click on the field in the list you want to edit and change as required refer to [Interactive data editor/Interactive Variable field properties](#) on page 363.

To preview IDE definitions:, use the **Preview** button in the toolbar to display how your interactive data prompts will be seen by the user. You can move through the prompts using the capture control

-or-

double-click on the field.



- The horizontal pin indicates that the position of the window is fixed every time a prompt appears. You can toggle to unpinned (pin shown at a 45 degree angle).
- Use the "?" symbol to display help text entered for current prompt.
- The **Next Incomplete** button moves to the next field in the document which has not yet been completed.
- The **Previous Incomplete** button moves to the previous field in the document which has not yet been completed.
- Use the arrows to navigate forwards and backwards through the prompts or use the Tab key.

Note that if you want to view the help text window press the **F1** button and use the **SHIFT+F1** to close the window.

Interactive variables

Interactive Variables can be used independently of the document's associated data dictionary and can be placed in your document as you would any other type of variable field.

When using interactive variables note that:

- they cannot be shared across documents, or active content. They are only accessible within the document / Active content where they are created.
- they provide the template designer with a way to prompt for information for the specific document

To insert an interactive variable:

1. In the document editor mark the insertion point in the logic map – or –position the cursor in the paragraph to which the interactive variable is to be added
2. From the **Insert** menu click **Field/Interactive Variables**. Click on the Add button to define a new interactive variable.
3. Define the field properties and any validation rules you wish to apply to the interactive variable.

Interactive data editor/Interactive Variable field properties

Note: Most of the settings required for field definitions in the IDE and Interactive Variables are the same except for a few variations as described in the table that follows:

:

Prompt Language

Select a specific prompt language you wish to work with.

Note that this field is applicable to Interactive Variables only and is therefore not available when working with the IDE.

Name

A unique name to identify the field or group item. Must not contain spaces.

Type	Identifies the type of data the field contains: Select one of the following from the drop-down list. Integer – defines a whole number. Type the number in the Default value field or select the number using the drop-down list. Number – defines a decimal number. Type the number in the Default value field. String – defines a text field. Type the text in the Default value field. Date – defines a date. Click the drop-down arrow in the Default value field to display a calendar control, localized to the regional setting of the local computer. Select the date required. Time – defines a time of day value. Click the up and down arrows to change the time or enter the value directly in hours, minutes and seconds.
Prompt	Optionally defines the message to be displayed to describe the action required from the user during data capture.
Default value	Optionally represents the initial value that the interactive field will contain when the template is first loaded. Note that the default value is always overridden by pushed data. This field also dictates if a field displays on the edit data screen in EngageOne Interactive.
Help Text	Optionally defines an extended message for the Help button to provide more comprehensive instructions to the user.
System data	Identifies fields where values are supplied by the mainline business system.
Key field	<p>Indicates to mainline business systems that this is a primary, searchable field. Key field designation identifies to the Systems Data Integration program (SDI) fields that might be useful to look up information. Refer to the EngageOne Programmers reference Guide for detailed information.</p> <p>Fields marked as KEY are presented on the key field screen in Interactive if they are also marked as Visible on xForm. Fields marketed as KEY are also concatenated together and used as the task identifier for tasks in Interactive.</p> <p>Note that this field is not applicable to Interactive Variables and is therefore not available when working with Interactive Variables.</p>

Visible on XForm	<p>Determines if the field is presented for user interaction. When working with a key field and this option is set, the key field screen is presented and the field is displayed for user interaction. If the field is not defined as a key and this option is set, the field it is presented on the confirm data screen.</p>
Read-only	<p>Specifies whether field values are user-changeable or not. Fields marked as read-only are not prompted for entry. If Read-only is selected, Lock value, Mandatory, and Omit prompt are disabled.</p> <p>This property will be frequently set on fields containing system-supplied data from mainline business systems.</p> <p>If Read-only is selected and data is passed in with data push, the text entered in Default value is obsolete.</p>
Lock value	<p>Specifies that after a value is entered in the field and committed, the value cannot be edited. If Lock value is selected, Read-only and Omit prompt are disabled.</p> <p>If Lock value is selected and data is passed in with data push, the text entered in Default value is obsolete and the field is immediately locked.</p>
Mandatory	<p>Marks the interactive field as compulsory for completion. When filling out the template a value must be provided for this field in order to progress to the next interactive field or to generate the document.</p> <p>If Mandatory and Lock value are selected, the user is prompted to enter a value unless a value has already been entered.</p> <p>If Mandatory is selected, Read-only is disabled.</p>
Omit prompt	<p>Specifies that the interactive editor user is not automatically prompted to enter a value for the data element unless it forms part of a prompt group. If you wish to enter a value in the interactive editor, either click on the push pin in the main editor window or, choose from the Select Prompt drop-down list.</p>

Data validation

Select **Choices**, **Expression**, or **Max Length** from the drop-down list to open the appropriate fields required to define the values. Available options depend on the **Type** of field selected.

If the **Choice** option has been selected, you can set up validation choice properties as follows:

- Selection type - chose from **Drop Down List**, **Radio Button**, **Check Boxes**.
- Selection values are made using the **Define** button.

Check Boxes - points to note:

- A string with a space-separated list of the selected choice values is returned from a multi-selection check box prompt.
- The **Check Boxes** option is only available if:
 - the field is a string type
 - the field is a **Drop Down List** or **Radio Button** already and the choice values have no spaces in them.

Choices can be set in the Designer but are more often pushed into the data model through system data integration. An example of data being pushed into the data may be when populating choices with a list of drivers for an auto claims letter.

Validate expressions

Fields are opened to define data fields if **Data validation/Expression** was selected on non-numerical field **Type**. Select one of the following from the drop-down list.

Range of – opens the next two adjacent fields for entering the range values.

- The “from-to” values can be entered, or selected from the drop-down list. Greater than – opens the adjacent field. Type the value or select one from the drop-down list.
- Less than – opens the adjacent field. Type the value or select one from the drop-down list.

Note that choices can be set in the Designer but they are more often pushed into the data model through system data integration. Examples of when you might hard code them in Designer might be for yes/no selections. An example of pushed data would be populating choices with a list of drivers for an auto claims letter.

Validate maximum length

This field is opened if **Data validation/Max Length** was selected. Specify the number of characters allowed for a **String Type** field.

Validate choices

These fields are opened to define data fields if **Data validation/Choices** was selected in the numerical field **Type**. In the **Selection Type** drop-down list define how a single selection from a pre-defined list should be presented to the user by selecting:

Value	Label
1	Option 1
2	Option 2
3	Option 3

Drop Down List, Radio Buttons and Check Boxes

You can define a list of **Choice Labels** and associated values. The Choice Label will be made available to the user by clicking the **Define** button to display the **Choices** dialog box. One of these labels can be selected by the user during data capture

Choice labels can be defined in multiple languages depending on the Prompt Language chosen from the IDE toolbar, refer to [Working with multiple languages](#) on page 354 for details.

Points to note:

- If you update a choice value in a particular language the change to the value will be reflected across all languages.
- Choice labels are optional and if omitted will be set to the choice value.
- Choice values cannot contain spaces if the field is a check box type choice.

Using annotations

Use the annotations feature to create a self-documenting schema. You are not limited to a line length in Designer, only by the limits imposed by the XML standard. Annotations are definable on groups, fields, repeating groups and delivery nodes.

To add, or edit annotations:

1. Click on the item in the tree that you want to create annotations for.
2. Click the **Annotations** tab to display the **Field Annotations** pane.



- The default text can be edited as required.
 - Clicking OK saves the annotations entered. The XML text is validated to ensure it meets W3C standards.
3. Click the **Save** button on the toolbar to save the data definition.

Repeating groups - default data

If you have defined and selected a repeating group, a tab is enabled to allow you to add default values for each piece of repeating data within the repeating group. When the data definition is saved, the instance file containing the specified default values is automatically generated in the instance XML.

To present repeating values in your template use either the repeating data feature – see [Creating a repeating data control](#) on page 306, or set up a transaction table using the Control Data page to identify the repeating data element that will be displayed – see [Repeating data and transaction tables](#) on page 120. In either case use the data assigned to the publication – see [Generating a dictionary and data map for IDE](#) on page 390.

Be aware that if you try to change the current data type of a repeating group field with existing default data values, a warning message is issued. If you choose to accept the change, any existing values will be removed.

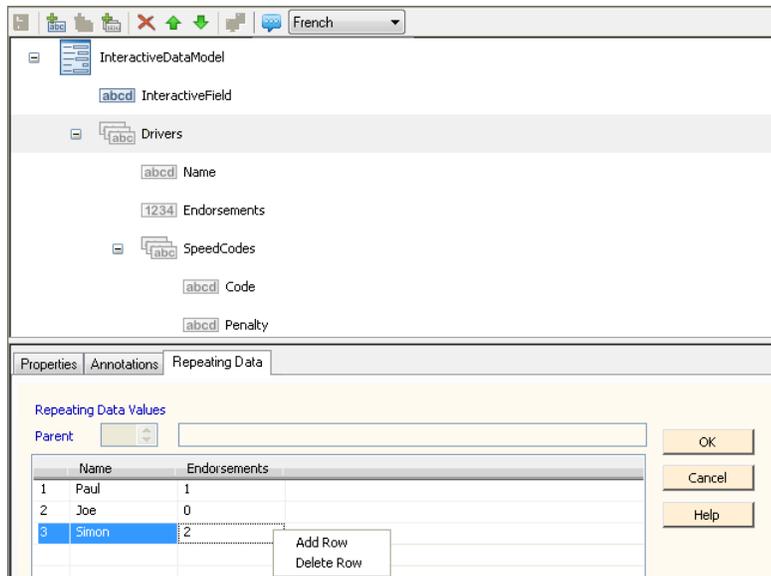
For more information about using repeating groups in EngageOne templates, see “XML instance file” in the EngageOne Programmer’s Reference Guide Generation 2.

For more information about using document selector values, see “Creating a delivery channel for EngageOne Deliver e-mail outbound profiles” in the EngageOne Administration Guide.

Setting default data

To set default data for repeating groups:

click on the repeating group for which you want to set default data and click the **Repeating Data** tab to display the **Repeating Data Values** pane.



- Right-click anywhere on the **Repeating Data Values** pane and select **Add Row** to add the values. The next index number is automatically supplied.
- Click on the field to add or change the values directly.

In the case of a nested, repeating group the entered values are assigned a read only, sequential index number. The values are matched to the corresponding index number of the parent node of repeating data. Note that if you set more nested, repeating data values than there are repeating data nodes, a nil record will be set up for those not matched. Refer to the following XML example.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<InteractiveDataModel version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="b271a7604aab4c4090686e40ad30daf5.xsd">
  <Publication>
    <InteractiveField xsi:nil="true"/>
    <Drivers>
      <Name>Paul</Name>
      <Endorsements>1</Endorsements>
      <SpeedCodes>
        <Code>SP30</Code>
        <Penalty>100</Penalty>
      </SpeedCodes>
    </Drivers>
  </Publication>
</InteractiveDataModel>
```

```
</Drivers>
<Drivers>
  <Name>Joe</Name>
  <Endorsements>0</Endorsements>
  <SpeedCodes>
    <Code>SP50</Code>
    <Penalty>70</Penalty>
  </SpeedCodes>
</Drivers>
<Drivers>
  <Name xsi:nil="true"/>
  <Endorsements xsi:nil="true"/>
  <SpeedCodes>
    <Code>SP70</Code>
    <Penalty>200</Penalty>
  </SpeedCodes>
</Drivers>
<DeliveryInformation>
  <Recipient xsi:nil="true"/>
  <EmailToAddress xsi:nil="true"/>
  <EmailFromAddress xsi:nil="true"/>
  <EmailSubject xsi:nil="true"/>
  <EmailBody xsi:nil="true"/>
  <FaxNumber xsi:nil="true"/>
  <DocumentSelector xsi:nil="true"/>
</DeliveryInformation>
<glprivate/>
</Publication>
</InteractiveDataModel>
```

Setting default data for nested repeating groups

The screenshot shows the EngageOne Designer interface. At the top, there is a tree view with the following structure:

- InteractiveField (abcd)
- Drivers (abc)
 - Name (abcd)
 - Endorsements (1234)
 - SpeedCodes (abc) - This group is highlighted in yellow.
 - Penalty (abcd)
- DeliveryInformation (+)

Below the tree view, the 'Repeating Data' tab is active. It displays the 'Repeating Data Values' pane. The 'Parent' dropdown is set to '1' and the path is '..\{Drivers\}Paul\{SpeedCodes'. Below this, there is a table with the following data:

	Code	Penalty
1	SP30	100
2	SP50	70
3	SP70	100

1. Click on the nested repeating group for which you want to set default data.
2. Click the **Repeating Data** tab to display the **Repeating Data Values** pane.
3. Select the **Parent** node that you want to add values to using the up and down arrows.
4. Add a row by right-clicking anywhere on the pane and selecting **Add Row**. The next sequential index number is automatically supplied.
5. Enter the values as required.
6. Click **OK** to save the repeating data values.
7. Click the **Save** button on the toolbar to save the data definition.

Dynamic prompting

In this section, we discuss more sophisticated prompting mechanisms available to you when designing your interactive communication.

A range of options are available to prompt for interactive data, you can:

- populate an interactive data/interactive variable (ID/IV) choice list from repeated data.
- repeatedly prompt for a specific value and act on it.
- prompt for multiple options in one prompt using check-boxes.

Modifying prompts

- The Modify Prompt object allows you to dynamically convert a non-choice validation type Interactive Data field, or Interactive Variable into a choice validation type field. Choices are added as the logic is executed in the ActiveX/Interactive Editor Application. Note that the changes to an Interactive Data field are visible across all documents in a Publication, whereas modifications to an Interactive Variable are only applied in the scope of a document.
- Any modifications are **not** saved to the XForm/XSD or PUBX files.
- Except at publication level, you can insert the Modify Prompt object anywhere in the logic map applicable to an action control object.

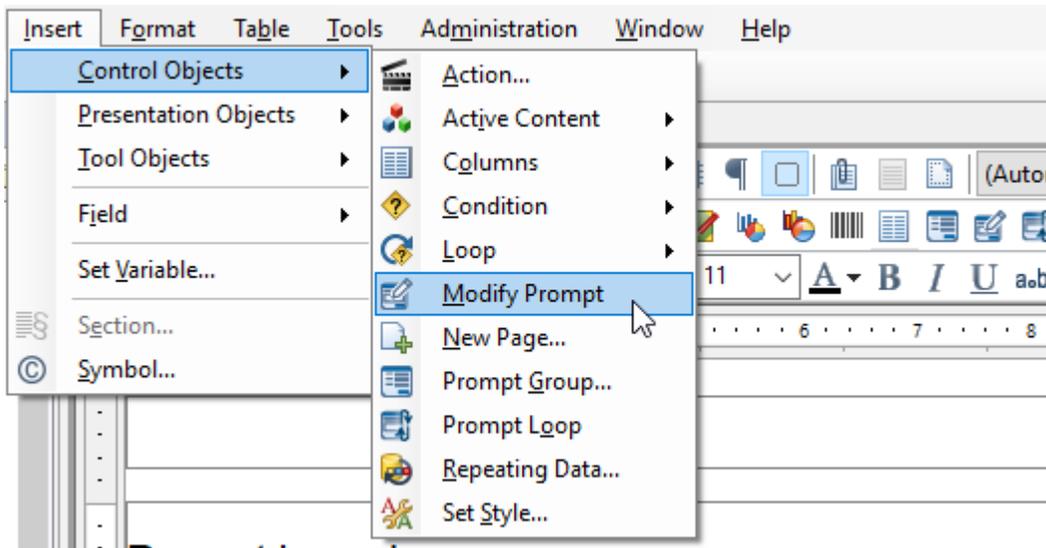
The Modify Prompt object cannot be used in:

- Publication level logic
- Page Setup logic
- Publishable Active Content

Note: The Modify Prompt object does not trigger a prompt in the Interactive Editor/ActiveX.

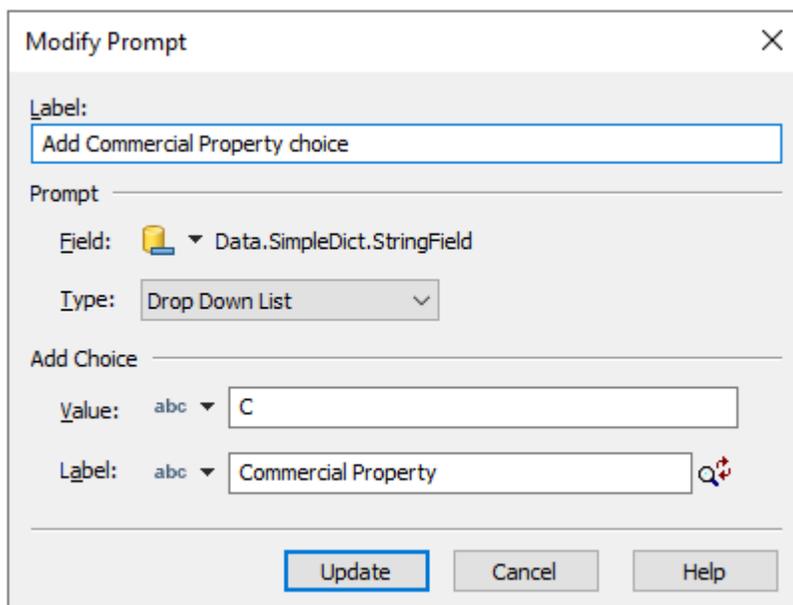
To use the Modify Prompt feature:

Select as shown below:



The Modify Prompt dialog is presented and allows you to configure the Modify Prompt logic map entry.

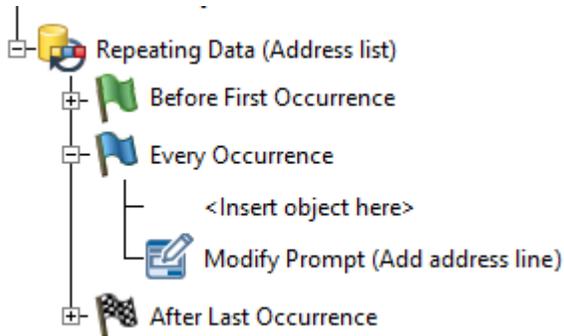
Modify Prompt settings



Field	The associated value picker allows you to choose the interactive data field or interactive variable whose choices you wish to change. The following controls are only enabled if the chosen field is not a choice type validation., refer to Interactive data editor/Interactive Variable field properties on page 363 for details.
Type	The ' Type ' combo box allows you to set the type of the selection; Drop Down List, Radio Buttons or, if the field is a string type, Check Boxes. You should typically choose the same type for all Modify Prompt objects that modify the same field to avoid confusion. Prompts are displayed according to the last type set when a Modify Prompt object is executed in the logic. If you choose different types, a warning is displayed in the editor log window.
Choice	<p>Best practice: The modified field's default value should be either blank, or set to a value you know will be in the choices added.</p>
Value	<p>This value picker allows you to set the choice value. For radio button and drop-down list selection types this can be any string value.</p> <p>Note the following:</p> <ul style="list-style-type: none">• If a dynamic value is specified that could lead to the supply of an invalid value, the choice is NOT added.• The value should be unique within the field choices.
Label	<p>This value picker allows you to set the choice label. The choice can be anything and can include spaces.</p> <p>Note the following:</p> <ul style="list-style-type: none">• Multi-language labels are not supported; the same label will be used regardless of the prompt language in effect.• The label should be unique within the field choices.

Logic map

The example below illustrates how the Modify Prompt object can be used within a repeating data structure where the object is used to add choices based on items in the repeating data group.



Restrictions

- Unlike a variable, an interactive data field can only have:
 - one value;
 - it cannot have a different value at different points in the logic;
 - this value persists until it is changed when prompted.

Best practice: Interactive field prompt choices should be set before the interactive field is first used in the logic.
- Care must be taken where choices are set in conditional logic controlled by an interactive value, either directly or indirectly.
 - Any changes made to the choice options are only visible the next time the field is prompted.
 - The current value of the field reflects the last choice made, it will not change until prompted.
- Duplicate Value and Label pairs are ignored. An error message is displayed if you attempt to create a selection with either a value or label that has already been used by another selection.
- The choice label is displayed for all prompt languages, you cannot set a different label for different languages
- Changing the choices of an IV/ID in a Prompt Loop would only modify the choices for the prompt when displayed in the loop instance and not globally.
- Once a field is changed, any previous field validation is lost.

Repeated prompting using prompt looping

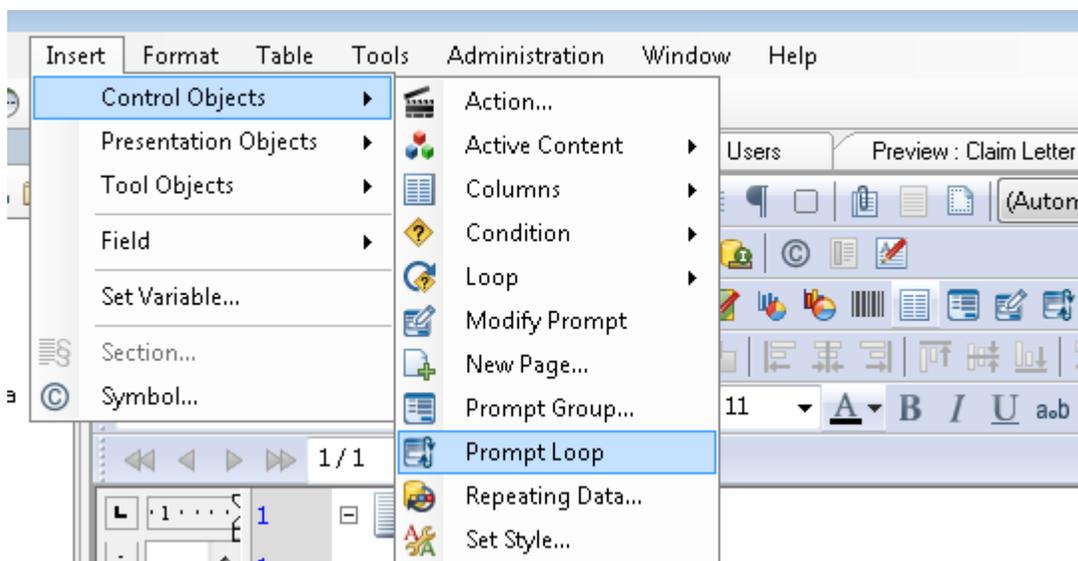
The Prompt Loop mechanism allows you to prompt from a drop-down list of items and perform further prompting associated with the selected item within the Prompt Loop. When all the prompt data has been entered, you are returned to the list and prompted to optionally select another item for further prompting.

The prompt data for each repetition of the Prompt Loop is stored as a distinct set of data in the Instance file in EngageOne Compose Interactive. The data for a Prompt Loop is distinct and stored separately from other Prompt Loops.

Note: The Prompt Loop feature is not supported in Publications with Cross-browser support enabled.

Adding a Prompt Loop

Select as shown below:



Restrictions

You can use Prompt Loops in Active Content logic and Document logic; however, they cannot be inserted within:

- Repeating Data, Transaction Tables, and Loops.
- another Prompt Loop.

It is also important to note that Prompt loops cannot be used in:

- Publication level logic
- Page Setup logic
- Publishable Active Content

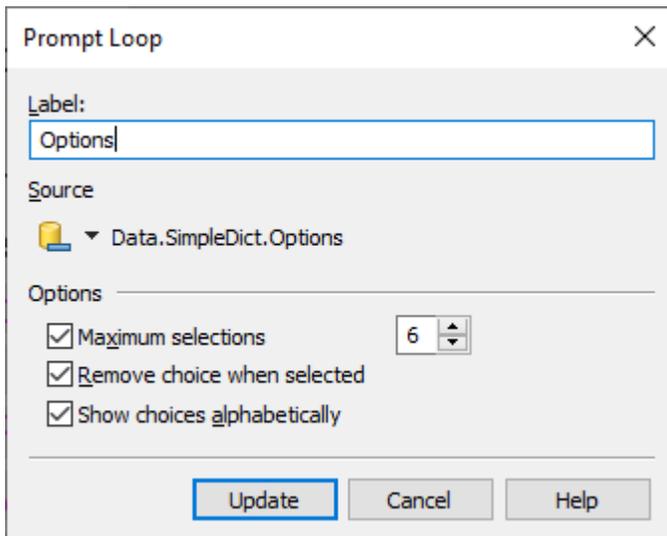
Prompt Loops cannot contain:

- another Prompt Loop
- Repeating Data
- Transaction table
- Active Content that contains Data Field or Interactive Variable
- Interactive Active Content Group

Editable text can be used in a Prompt Loop but the editable text will not have separate values for each loop iteration.

Configuration

The Prompt Loop dialog allows you to define the source interactive data/variable that will drive the Prompt Loop in addition to other properties.



Label	This entry field allows you to set the logic map label
Source	This value picker control allows you to select the source interactive variable or data alias for the loop. Only items with no data validation and items with choices validation can be used, selecting other types will not allow you to create or update the item. Choice type interactive variables or data aliases have a fixed list of choices determined when created. If a dynamic list of choices is required a non-choice type (for example, a string type), interactive variable or data alias can be modified to a choice type using the modify prompt dialog.

Options

The following options are available:

The '**Maximum selections**' option enables you to limit the number of selections that can be made when prompting. The minimum is 1, the maximum 99. An individual choice can only be selected once.

The '**Remove choice when selected**' option will remove the selected choice from the selection list during prompting. This may be useful if there are a large number of selections that could be made.

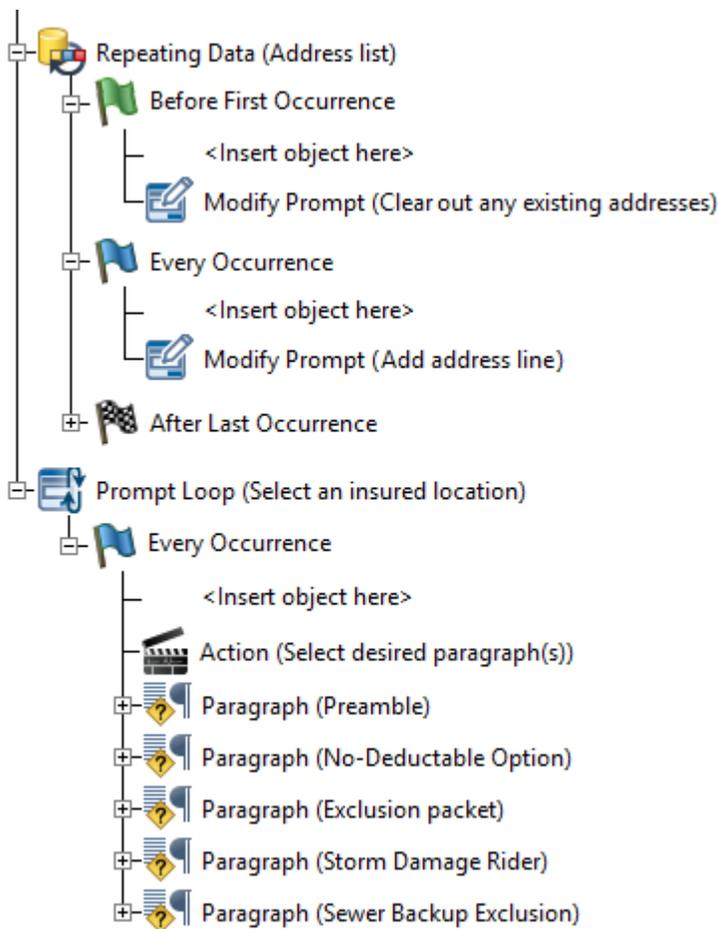
The '**Show choices alphabetically**' option determines if the list of choices in the prompt is shown in the order they were added or sorted alphabetically.

Logic map

The logic map example below shows how to set up the choices for an interactive variable or data field and then prompt until the user has completed data input. Each choice selection results in the 'Every Occurrence' logic being executed. Interactive variables and data fields within the "Every Occurrence" logic are distinct to the Prompt Loop iteration but are prompted for as expected in EngageOne Compose Interactive.

Note: Only the value of the interactive variable or data field related to the current iteration can be accessed within a Prompt Loop. Variables can be used to share values across iterations if needed.

In this example, the user is further prompted to make a choice which results in one or more following conditional paragraphs to be placed in the communication.



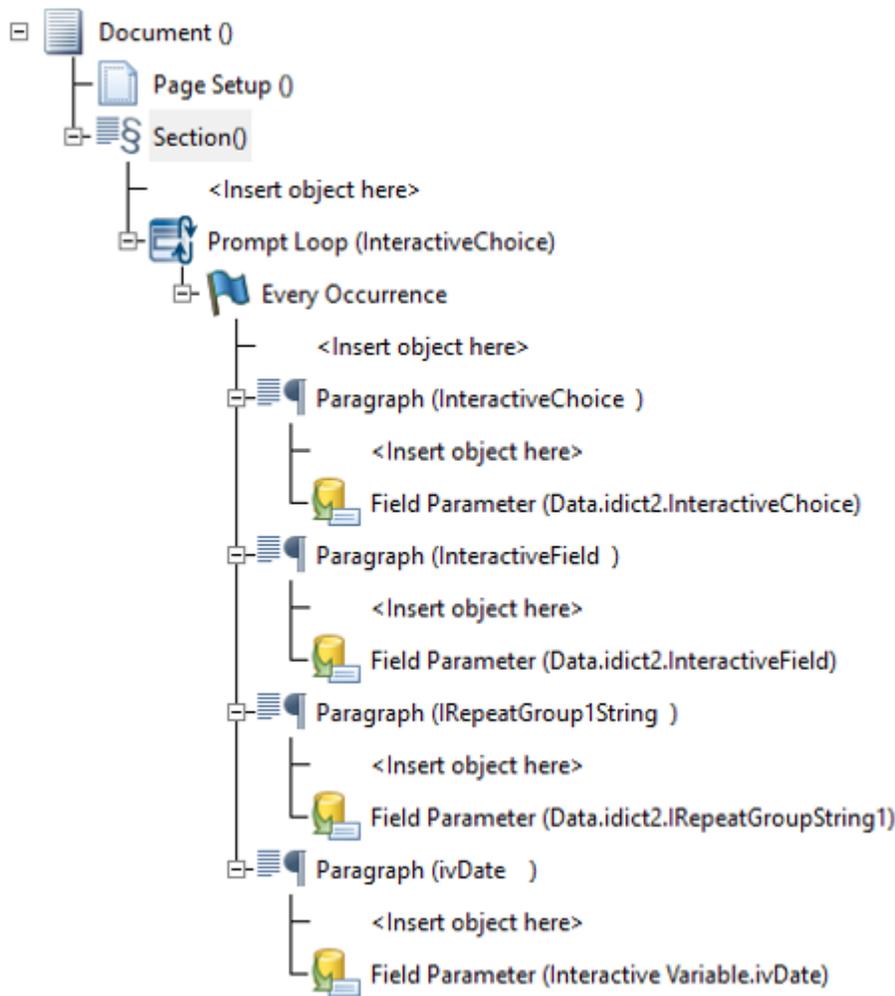
Note: The loop is driven from an interactive variable or data field that is configured with any type of 'Choices' data validation. No other types can be set as a source.

In EngageOne Compose Interactive, the Prompt Loop choice selection and prompt data from the Every Occurrence branch are stored within a repeating group structure created for the Prompt Loop. An instance of the repeating group is created for each Prompt Loop choice selection.

Prompt Loops and EngageOne Interactive

During Publish for EngageOne, a repeat group structure is automatically created in the interactive data model for each Prompt Loop item. The repeat group contains a data element for every interactive variable or data field used by the Prompt Loop, including the source data item. The name of the repeat group is derived from the name of the source data item, and the names of the data elements are derived from the original interactive variable or data field. Names are made unique if necessary.

The following example shows the Designer logic map and corresponding XML instance file produced by Publish for EngageOne.



```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<InteractiveDataModel version="1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="3792743AC5A3451FB641D7A97151CF6F.xsd">
  <Publication>
    <InteractiveField>InteractiveField default
value</InteractiveField>
    <InteractiveChoice>one</InteractiveChoice>
    <InteractiveRepeatGroup>
      <IRepeatGroupString1>repeat1_string</IRepeatGroupString1>
    </InteractiveRepeatGroup>
    <InteractiveRepeatGroup>
      <IRepeatGroupString1>repeat2_string</IRepeatGroupString1>
    </InteractiveRepeatGroup>
    <FLD_ivDate>2019-09-13</FLD_ivDate>
    <LPGRP_InteractiveChoice>
      <LPFLD_InteractiveChoice xsi:nil="true" />
      <LPFLD_InteractiveField>InteractiveField default
```

```

value</LPFLD_InteractiveField>

<LPFLD_IRepeatGroupString1>repeat1_string</LPFLD_IRepeatGroupString1>
  <LPFLD_ivDate>2019-09-13</LPFLD_ivDate>
  </LPGRP_InteractiveChoice>
  <DeliveryInformation>
    <Recipient xsi:nil="true" />
    <EmailToAddress xsi:nil="true" />
    <EmailFromAddress xsi:nil="true" />
    <EmailSubject xsi:nil="true" />
    <EmailBody xsi:nil="true" />
    <FaxNumber xsi:nil="true" />
    <DocumentSelector xsi:nil="true" />
  </DeliveryInformation>
  <glprivate />
</Publication>
</InteractiveDataModel>

```

In this example, the repeat group created for the Prompt Loop is called `LPGRP_InteractiveChoice` since the `InteractiveChoice` data field was selected as the source data item for the Prompt Loop in Designer. The elements in the repeat group represent the data items used by the Prompt Loop and are separate and distinct from the elements for interactive variables defined in the design logic and data fields defined in the Interactive Data editor. For instance, the `FLD_ivDate` element represents the `ivDate` interactive variable whereas `LPFLD_ivDate` represents the instance of the interactive variable used in the Prompt Loop. Similarly, although the `InteractiveChoice` data field was selected as the source data item, the actual source data value for the Prompt Loop is stored in `LPFLD_InteractiveChoice`.

In the interactive data model, the properties of the data elements in the Prompt Loop repeat group are copied from the original interactive variable or data field with the following exceptions:-

- The Prompt Loop source data element is marked as Read-only and Mandatory since it is not changeable inside the loop iteration and a value is required for each iteration.
- The Prompt Loop source data element is marked as System data if the Prompt loop contains any elements marked as System data. Data supplied for the Prompt Loop repeat group through EngageOne Interactive data push must contain values for the source data element.
- The Visible on XForm property is not supported for Prompt loop data elements and is automatically set to false.
- The Key field property is not supported for Prompt loop data elements and is automatically set to false.

For more details on the properties, refer to [Interactive data editor/Interactive Variable field properties](#) on page 363 for further information.

When supplying data for the Prompt Loop in EngageOne Interactive data push, the Prompt Loop repeat group should be mapped using a `RepeatMap` in the same way as other repeat groups. Refer to the **EngageOne Programmers Reference Guide** for further information.

- The Previous Selections list, located under the drop-down list is populated with any choices previously selected, initially it will be empty unless data has been added through system data integration.
- In the Prompt Loop dialog there is an extra button below the combo box.

Select address

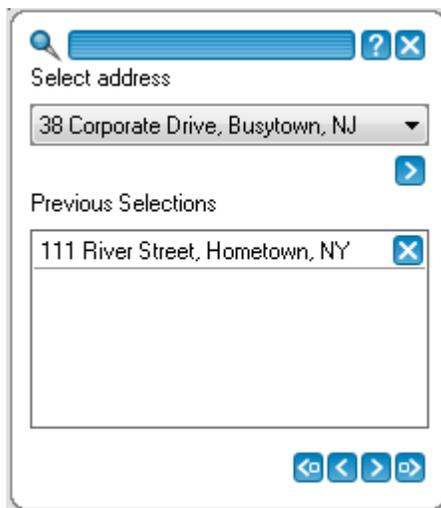


Pressing this selects the choice shown in the combo and starts prompting at the first prompt inside the Prompt Loop 'Every Occurrence' logic.

- A selection causes the prompting to move to the first prompt found in the Prompt Loop 'Every Occurrence' logic if a non-blank selection is made.
- If the blank selection is made the dialog closes and prompting moves to the next prompt **unless** the source field is mandatory and no selections have been made. In this case an error is displayed and the prompt remains open until a selection is made.

Note: pressing the next/next incomplete buttons will not take you to the first prompt inside the 'Every Occurrence' logic, it takes you to the next appropriate prompt after the Prompt Loop

Notes on previous selections



- Pressing the return key, or double clicking a row starts prompting again for that selection.
- Pressing the delete/backspace key or clicking the 'x' button removes the entry from the list.
- Pressing the Tab key while in the previous selection list starts any prompts for that selection and then takes you to the next entry in the list. If there are no more entries the dialog exits and prompting moves on to the next prompt
- Pressing the Shift-Tab keys while in the previous selection list takes you back to the previous entry in the list. If there is no previous entry the focus is put back to the combo box.
- The list order can be changed by dragging a row into the desired position or using the Ctrl-U or Ctrl-D key combination to move the selected row Up or Down.

Multiple checkbox prompting

You can include multiple checkbox selection in your in your publication design, where appropriate.

Refer to [Conditional expressions](#) on page 218, [Interactive data editor/Interactive Variable field properties](#) on page 363, for detailed information.

To configure multiple selection Check Boxes in the Interactive Data Editor:

1. Locate the required Interactive Data Model object from the Designer navigator.
2. Select the required field in the Interactive Data Editor. Note that the field must be of type String.
3. From the **Set data validation rules** frame select **Data validation, Choices** and a selection type of **Check Boxes**.

Set data validation rules

Data validation Choices ▾

Validate expressions:

▾

Validate maximum length:

Validate choices:

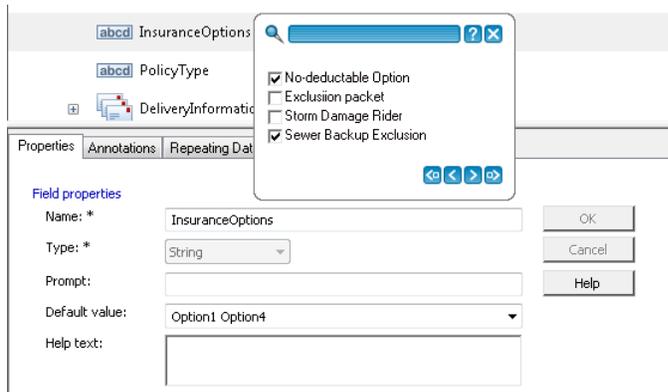
Selection type: Drop Down List ▾

Selection values: Check Boxes
Drop Down List
Radio Buttons

4. Click **Define** to display the **Choices** dialog box, here you can define Choice values and their associated labels.

Note: Choice values cannot contain spaces.

5. Click **OK** when you have entered all **Value** and **Label** pairs.
6. Click **Default value** to define default options when the check box is initially presented



Prompt groups

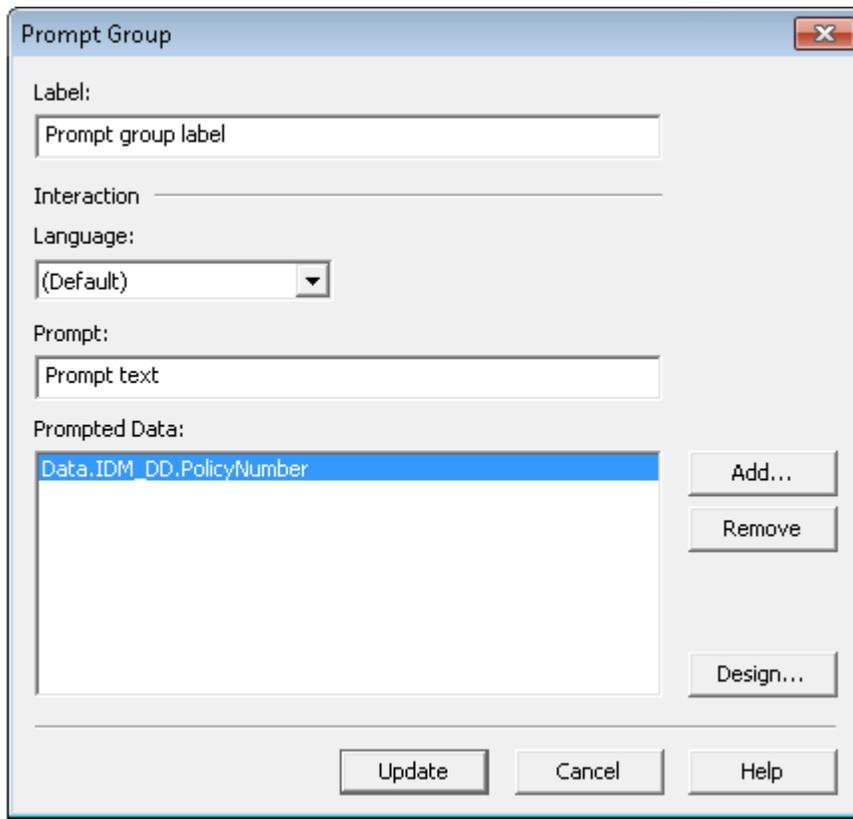
Prompt groups enable you to collect multiple data values in a single prompt dialog box. For example, you might have several single interactive data fields for lines of an address that you want to collect in one capture control. You can collect the data to be included in the prompt group by either selecting the interactive text on the page and creating the prompt group, or by inserting a prompt group in the required position in the logic map. You must then select the required data elements from an existing data dictionary. Once you have created the prompt group and defined the data, the prompt layout editor allows you to design the layout of the prompt that will be shown in the interactive editor.

Points to note:

- Overuse of prompt groups can make a poor user experience.
- Bear in mind screen sizes used by your users to ensure groups fit within all screen areas used.
- Do not use this feature to prompt for everything in a document, this eliminates all sense of context for the user.
- Interactive variables cannot form part of a prompt group.
- Prompt groups are only available within the interactive editor, they are not used on the xForm presentation for the key field or, confirm data screens.

To create a prompt group:

1. Select the position in the template document or its logic map where you want to place the prompt group and on the **Tools** menu select **Control Objects/Prompt Group**, – or – select the **Prompt Group** button on the **Standard** toolbar, – or – in the document select the data fields you want to group together on a prompt, right-click and from the context menu select the **Create Prompt Group** option. The **Prompt Group** dialog box is displayed.



- Set the logic map **Label** for the prompt group.
 - Set the **Language** if the template is designed to be used in more than one language, otherwise leave this option unchanged at the **Default** setting.
 - Set the **Prompt** text to be displayed for the prompt group. If you are working in more than one language ensure that the **Prompt** entered corresponds to the selected **Language** option.
 - **Prompted Data** - displays the currently associated interactive data if text was selected. This will be empty if created via the toolbar or **Insert** menus.
 - Click **Add** to invoke the **Add Existing Alias** dialog box where you can reference existing interactive data fields. See [Working with input data](#) on page 117 for more information about data aliases and data dictionaries.
 - Highlight a line of prompted data and click **Remove** to delete it from the prompted Data list.
 - Click **Design** to invoke the **Prompt Layout Editor** dialog box for designing your prompt box. Refer to [The Prompt Layout Editor](#) on page 389 for further information
2. Click **Create** to finalize your prompt group.

To update a prompt group:

1. In the template document double-click on the prompt group in the logic map – or – right-click and from the context menu select Properties to invoke the Prompt Group dialog box.

2. Click **Add** to invoke the **Add Existing Alias** dialog box where you can select the required interactive data fields. In this dialog box you can insert fields by either selecting them and clicking the **Insert** button or by double-clicking them

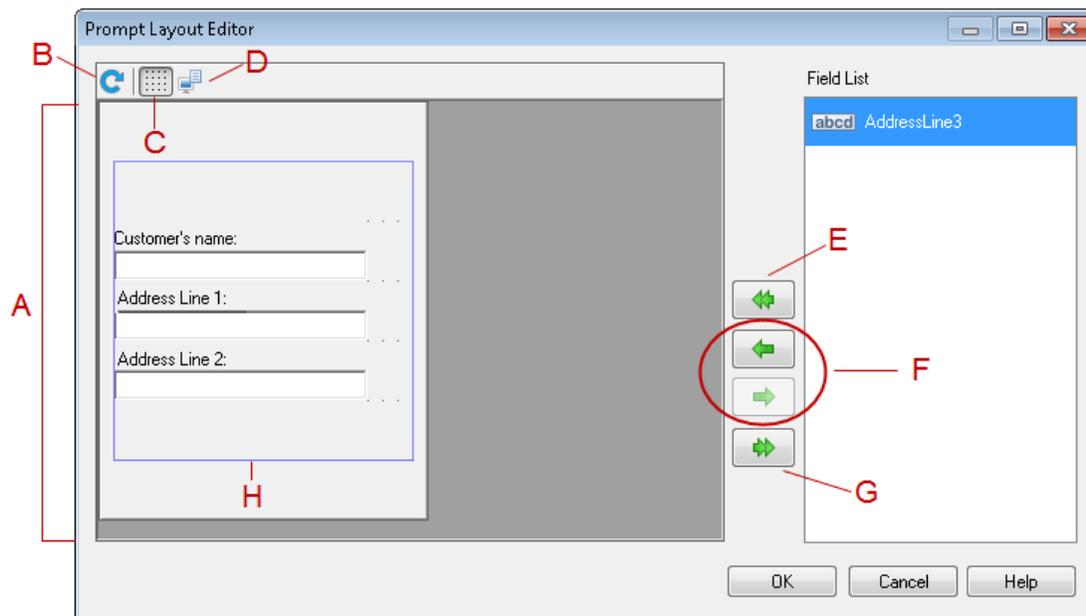
-or-

Highlight a line of prompted data and click **Remove** to delete it from the **Prompted Data** list.

3. Click **Update**.

The Prompt Layout Editor

The **Prompt Layout Editor** comprises a canvas area, which represents the capture control design and a field list that shows the fields available. You can move prompt fields from the field list to the canvas area and vice versa by using the left and right arrow keys. Once in the canvas area the prompt fields can be moved, removed and resized within the margin by dragging the selected field and dropping it in the required position. You can re-size the canvas area by clicking on it and dragging out the area as required.



Legend:

A	Canvas area
B	Resets the design to the default setting.
C	Enables/disables the Grid feature. When enabled this overlays the canvas making it easier to line up the selected field controls.

Building the EngageOne template

After defining the interactive fields applicable to your template design using the interactive data editor, you can build templates for your EngageOne document as you would a regular Designer publication.

Interactive fields are referenced as you would other data fields within Designer.

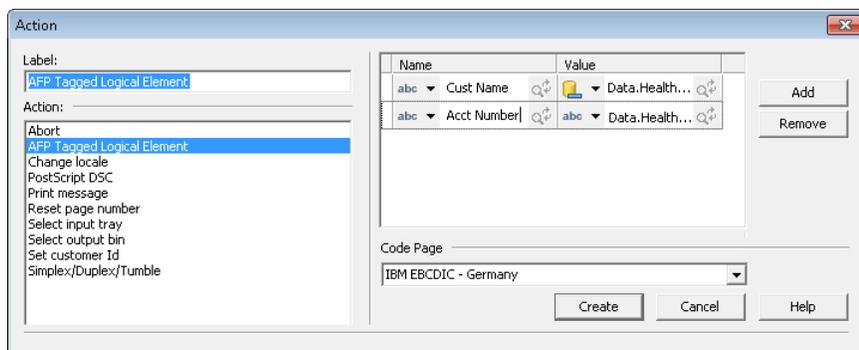
Note however, that the following features are not supported, or are restricted when working with EngageOne templates in Designer:

- Document Interchange Journals (DIJ) are not supported through Designer for EngageOne documents but can be created using the EngageOne Administration application. See the EngageOne Administration Guide for details.
- The ACIF journal mechanism, Tag Logical Element (TLE) is not supported on the EngageOne platform.
- Lookup tables for interactive documents and external keyed objects for interactive documents.
- A remaining space area can only contain static content.

Creating an AFP index

If you wish to create indexed output for AFP you can specify the relevant index values at the appropriate point in the logic map using the AFP Tagged Logical Element (TLE) action. An AFP structured field is created in the output based on the supplied name/value pair and code page which will form part of the AFP index.

To create AFP index values for EngageOne: From the Insert menu select Control Objects/Action. In the Action dialog box select an action type of AFP Tagged Logical Element.



As with Designer, EngageOne can make use of structured XML journals, which in addition to usual journal entries, automatically provide page count and page offset information. For more information about structured XML journals see [Structured XML Journals](#) on page 318. If XML journal entries in the form of a name and value pair are used, they must be given a unique name within the template. The names will be available in EngageOne Administration within Delivery Management for use as system variables. See the EngageOne Administration Guide for details.

Active Content and EngageOne

Unlike other uses of Active Content when working with groups you may want to create Active Content objects exclusively for use by a single document design.

Private Active Content is created and maintained within a document and is not visible outside that document.

Active Content plays an important role in defining optional content in EngageOne documents. Active Content objects are used within Active Content groups, which when marked as interactive allow content to be built dynamically according to selections made by the user from the optional content list.

In a typical EngageOne scenario, you could create an optional content list, with a variable set in one of the entries in the group. Should this variable be matched in the selection you could direct the user to another interactive list elsewhere in the document. Additionally you could automatically add other content later in the document according to a selection made by setting the **Show With** property in a non-interactive Active Content group.

For more detailed information about using Active Content in this way see [Using Active Content groups](#) on page 435.

External Files management and EngageOne

EngageOne templates reference certain resources that are not contained within the template. External resources include images (also referred to as key maps or keyed images), lookup tables, and RTF files. **IMPORTANT:** external files support is used during composition of your template and is therefore only available to EngageOne Batch and EngageOne On-Demand.

Images within the EngageOne environment can be handled in two ways.

- Images used directly in templates and managed within Designer. These are imported into the Designer environment and introduced into the EngageOne server via templates and HIP resource files. Once inside the EngageOne server there is no way of adding, removing or updating these images without editing and re-publishing the template.
- Images that are referenced in templates by external key maps. Key maps are used to map Designer keys dynamically to image resources so that images may be updated without having to update every template.

Lookup Tables are used to perform value substitution during composition of a document by using a file that contains key name and value pairs. Import lookup table files and index them on the EngageOne Server so that these files can be referenced and included in a Generate process for a template that will utilize these lookup tables.

RTF Files are used to include external content from RTF documents during composition of a document. The entry point for that content is marked within the template by defining special variables. The variable is then resolved as a reference to an external file. Import RTF files so that these files can be referenced and included in a Generate process for a template that will utilize them.

If you want to reference external image resources in an EngageOne template, or Active Content you can do so by first creating a regular external key map file – see [About key maps](#) on page 268. During **Publish for EngageOne** the external key map reference will be picked up automatically. Whereas internally referenced images are prepared for importing into EngageOne in **Publish Resources for EngageOne** (see [How to publish for EngageOne](#) on page 405), external keyed images are managed using the EngageOne Keymap Generator.

The EngageOne Key Map Generator is a standalone windows application provided with your EngageOne package. It allows you to select a target print stream and allocate the relevant image formats to convert source images into target image formats. The target image metrics are read and a Generate equivalent key map XML file is created, which is zipped up with image files ready to be imported into EngageOne Administration.

For more information about converting external keyed images for EngageOne see the EngageOne Key Map Generator User's Guide. For more information about defining key maps and image management in EngageOne see the EngageOne Administration Guide.

Editable Text

You can allow the user to edit or create free form text as required within an EngageOne document.

Such editable text is defined in an editable container object in Designer and can be limited to a specific string within a paragraph or sentence, or you can allow an entire paragraph to be editable. An editable container can contain editable text (highlighted in yellow), as well as protected, non-editable text. Empty editable paragraphs that are created within a container do not appear in the print stream.

You can create an editable container and insert or mark text as editable within it. Or you can drag the mouse over the required area of text or blank space in the document and mark that as editable, which will also create an editable container. You are also able to drag paragraph objects into and out of editable containers as well as between containers.

It is important to mark the end of paragraph marker as editable if you wish to allow the entire paragraph to be deleted. If it is not marked as editable, an empty paragraph will appear in the output even if all the other text in the paragraph is deleted. Formatting marks can be toggled on and off by pressing the **Toggle Formatting Marks** button on the editor toolbar.

Language specific prompt and help text can be assigned to an editable container. These will be available to the users based on the language chosen when creating an interactive communication.

Note: You can only mark one paragraph of text at a time as editable.

In preview mode, the editor toolbar provides standard text editing and formatting features and additional EngageOne related features. You can insert text into an editable container from an external

document using copy and paste and use the standard text editing features, bullets and numbering, as well as paragraph formatting (on highlighted text only). For more details see [Previewing an EngageOne template](#) on page 397. The user will be provided with the same kind of editing features in their front-office application.

Note that images and tables can not be inserted into editable content.

Note that you can only select fonts in the preview that have actually been used within the template design, but when you publish for **Interactive** you can specify extra fonts for the user to use with editable text – see [Running a publishing task](#) on page 602.

To create an editable container:

1. Select the required position in the template document or in the logic map and from the **Edit** menu select **Mark as Editable**. – or – select the **Mark as Editable** button on the **Standard** toolbar.
2. The **Editable Container** dialog box is displayed.
3. Assign an identification label and click **Create**.
4. Enter language specific prompt and help text as required and click **Create**.
5. Enter or copy text into the container as required. The text will automatically be marked as editable. To revert editable text back to protected text see [Editable Text](#) on page 393.

To mark text as editable or protected:

1. In the required template document drag the mouse over the required area of text or blank space in the document that you want to mark as editable and from the **Edit** menu select **Mark as Editable**. – or – select the **Mark as Editable** button on the **Standard** toolbar. – or – use CTRL+E.

Fred Jones
111 Some Street
Some City WI 55555

Certificate: 123456
Claimant: Wilma

Dear Fred Jones

We received Dr. James 11/18/2007 correspondence. The correspondence requests a review of services provided by Dr. James on 08/09/2007 for Wilma. Based upon this review, we find the following exclusions in the terms and conditions of the Certificate:

- Charges for treatment of Behavioral Health or Substance Abuse.
- Charges that are related to or a complication of a Pre-Existing Condition.
- Charges that are not specifically listed as a Covered Charge in the Medical Benefits section of the certificate.
- Charges that are due to complications of a non-covered service.
- Charges that are incurred before the Covered Person's Effective Date or after the termination date of the coverage.

Free form text area

Therefore under the terms and conditions of the Certificate, coverage is not provided.

2. Check that the interaction in your template design performs as required by using the **Preview for EngageOne** option. See [Previewing an EngageOne template](#) on page 397 for more details.

To unmark text as editable within an editable container :

Select the highlighted text you want to revert back to protected text. From the **Edit** menu uncheck the **Mark as Editable** option,

– or – select the **Mark as Editable** button on the **Standard** toolbar. – or – use CTRL+E.

Note that it is possible for an editable container to contain only non-editable text. If you delete an editable container, be aware that all content will be deleted within it, including protected, non editable text. If you want to delete the object in the logic map then you must move the content you still require outside the editable container and then delete the object.

Note that if you mark text as editable in a document without creating the editable container first, then the editable container will automatically be created for you.

Recipient processing

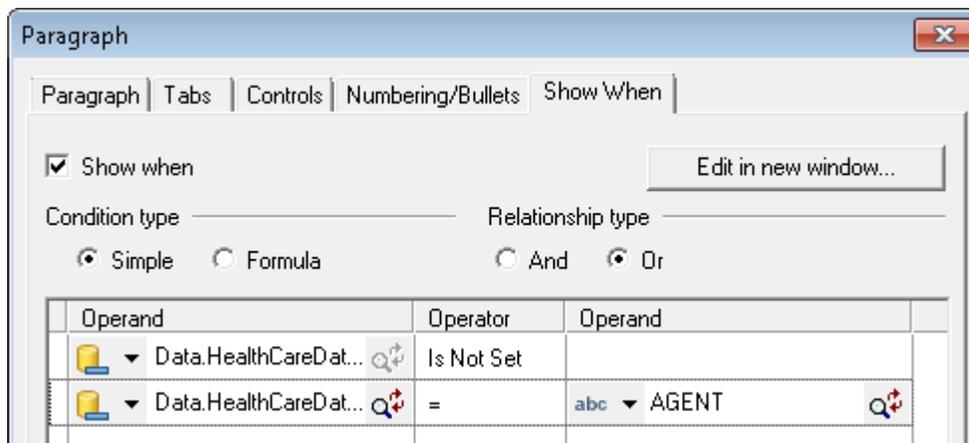
When a template is submitted for delivery by the user the delivery process is defined by the EngageOne Administrator, which involves defining and mapping devices and recipients to channels and grouping channels to form delivery options.

Provides automatic carbon copy processing using recipients, meaning that the user need only select one delivery choice and carbon copies will automatically be created and delivered.

Recipient conditions in the template are used to produce non-unique content. For example, if you do not want the user AGENT to receive the content designated to INSURED then you must check a condition against the recipient field for the content in your template design:if the recipient field **Is Not Set** (the user needs to see everything), **Or**, is equal to AGENT.

You must then use EngageOne Administration to set the **Inclusion Condition** in the delivery channel to match. See the Delivery Channels chapter in the EngageOne Administration Guide.

To set a recipient condition: on the **Format** menu option click **Paragraph**. On the **Show When** tab of the **Paragraph** dialog box check the **Show When** option and enter the condition as required.



- Select **Show when** to make the paragraph content conditional.
- Select the **Simple** condition together with the **Or** relationship type.
- Select the data alias **Delivery Information/ Recipient** from the data dictionary.
- Select the comparison **Is Not Set** (means that the user is to see everything in the document).
- Set the **Operator** equal to (=) and enter the name of the recipient in the **Operand** field.

For more details about conditional paragraphs and creating conditional expressions see [Conditional paragraphs](#) on page 88.

Previewing an EngageOne template

View a template containing one, or many documents, as it would be seen by a user in a front-office EngageOne environment so you can see how the user will interact with prompted input fields.

You must ensure that all data aliases are mapped to your template before you can run a preview for EngageOne. This will be the interactive data definition (data map) produced from defining fields in the interactive data editor and generating the data dictionary.

If your template contains localized prompts, you can view text in a particular language by selecting the appropriate Prompt Language from the drop-down in the EngageOne Compose Interactive Editor application preview; by default, the prompt language will be set to the locale of the Designer. Any prompts that have not been assigned for the selected language will be presented in the default language.

When previewing documents containing localized paragraph variations you can select a particular language option for presentation. Use the **Select Document Language** drop-down list to control the way in which localized paragraphs are presented. By default, the Customer Language setting is in effect and indicates that localized paragraphs are presented according to the locale set by the data processed in the document template. You also have the option to select the current prompt language to present localized paragraphs; note that this option will only be available if the document contains paragraphs configured in the current prompt language.

To run preview for EngageOne:

1. In the navigator browse to and select the publication you want to preview and from the **Tasks** menu select **Preview for EngageOne**.
2. A preview file is generated and the path is presented in the **Documents** window. Click on this path to review the output. Following is an example of a single template preview.

In interactive mode the document background is shaded and you can use the capture control to move through both the data prompts that have been introduced through the IDE and lists for selecting optional data defined using Active Content groups. Refer to [Using the Interactive Data Editor](#) on page 356 for details about general use of the capture control.

Note that any choices made will simultaneously update the document when you make a selection in a capture control that is defined using Active Content groups.

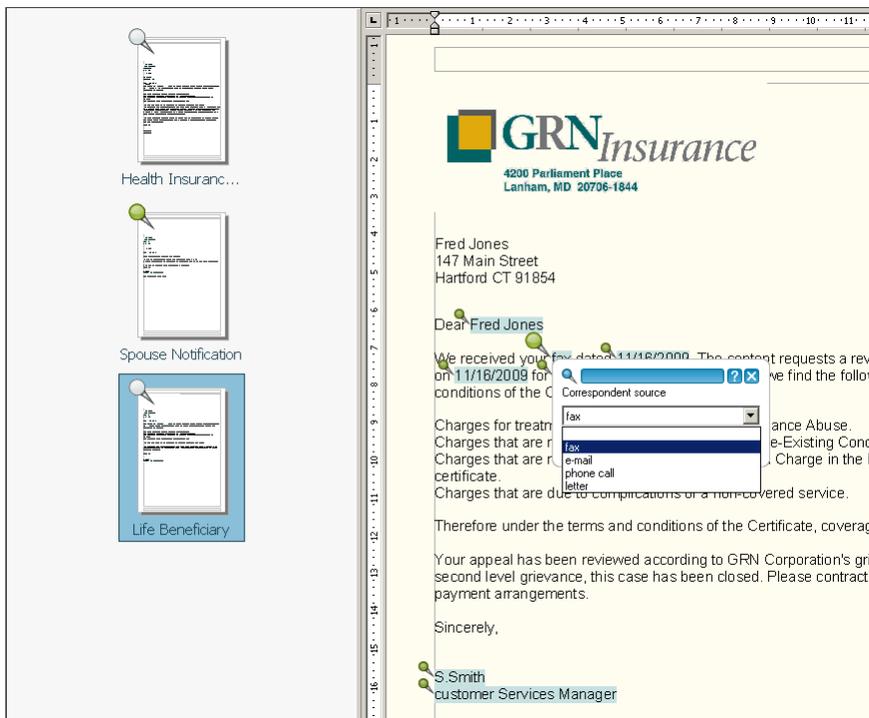
If a list of optional data has been defined with the **Allow Filtering** option checked in the Active Content Group dialog, then you can control the option list by typing your search criteria in the filter control box. Only options that contain the letter(s) you typed into the box will be displayed. For example, type in 'for' if you want to display only those options that have words containing the letters 'for'. Note that the filter is not case sensitive.

You can also tab through the prompts without entering data. In either case when you reach the end of all data prompts included in the template the document background turns to white and

turns back to gray again if you reach the first prompt in the document. The order in which you see the prompts is driven by the logic so if you want to change the sequence then you must change the positioning in the logic map. Any interactive data is identified on the document using pushpins.

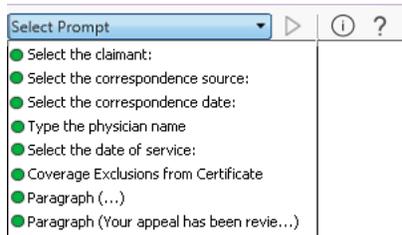
- Green pushpins indicate data that has already been entered.
- White pushpins show positions where data input is still required.
- The blue background indicates the currently active document.
- A larger version of pushpin indicates the current active prompt. This can be green or white depending on whether data has already been entered or not.

If your template contains multiple documents a navigation pane will be displayed that shows named thumbnails of the documents in the template. You can move between the different documents by clicking on the required thumbnail. Prompts are only displayed for the currently selected document and once all prompts have been completed the next document will be displayed automatically, until you reach the end of the series. Following is an example of a multiple document preview.



Editor toolbar

The editor toolbar provides standard text editing and formatting features for editable text and additional EngageOne related features. For details on how to use the editor toolbar features see [Editing features](#) on page 58 and [Creating paragraphs](#) on page 68. Pushpins can be toggled on and off using the Pushpin button (only when the prompt box is closed).



You can select an interactive location from the drop-down list to display it directly for editing. Or you can click on any pushpin to display the related capture control/ editable text. The Show Prompt button will take you to the last interactive location selected. If any font is used that is not installed on the user's machine, a warning message is issued when loading the template. The user can suppress this message by selecting the Do not show this again check box and the nearest font substitute will be used. The user can reactivate suppressed messages by selecting Tools/Enable suppressed messages from the editor toolbar.

Modifying sample data

If your sample data does not allow you to preview extensively enough, for example, if you want to include some sample data for repeating groups (previewing tables) then you can extract the sample data and modify it as follows:

To modify sample data in order to preview repeating group entries:

Open the required sample data file from the navigator and from the Tasks menu select Extract Sample Data.

Modify the repeating data in the XML file using a text editor such as Notepad.

Open the required publication from the navigator and on the View menu select Publication. Right-click on Publication in the logic map and select Properties from the short cut menu.

Click the Sample Data browse button, browse to the extracted sample data file and click OK.

Note that in a preview for EngageOne you can only find and replace text in an editable text section. Also the What to search options are not applicable. – spell settings are not applicable when using the interactive editor. For more information about spell check management see the EngageOne Administration Guide.

Publishing for EngageOne

Publishing for EngageOne Server differs from regular Designer publishing in that a template package is created and saved to a location of your choice.

This package is a combination of:

- Presentation elements
- Logic
- Interactive data placeholders
- A preview PDF
- A thumbnail image associated with the preview PDF presenting the first document in the template
- A manifest describing the attributes and contents of the template.

In the event that data aliases do not appear in your sample data they will be presented in the preview pdf but will be assigned default values. Refer to [Import Sample Data File options](#) on page 473 for details on the default values used.

Note that during publishing a preview PDF is automatically created and included in the template package. It is important to note that the preview is based on sample data. Care should therefore be taken when defining your sample data to avoid the propagation of sensitive information.

EngageOne Administration handles the import of these template packages into the EngageOne Server where they are made available to the user via a Web client. Note that templates do not contain any font and image resources for target output data streams. Such resources are placed within independent files that must be created and deployed in conjunction with requests from the EngageOne server. See [Publishing resources for EngageOne](#) on page 411.

Additionally Active Content (and associated resources) that has been enabled for publishing can be published as a group or individually. These files are also imported into the EngageOne environment. For more detailed information about this process and configuring the EngageOne production environment see the EngageOne Administration Guide. See “Using Active Content with EngageOne”. Special care must be taken when publishing Active Content on its own – see [Using Active Content to update publications in production](#) on page 441.

In some cases you might need to make a wider selection of fonts available to the user within editable sections of text, other than those contained within the template design. This is handled in the Publishing Wizard, where you can select additional fonts in the **Additional Resources** page. Note that the fonts specified will be available in preview mode and you must still use the **Publish Resources for EngageOne** option as usual.

All data aliases that have been used in the template must be linked with actual data elements. With EngageOne these references are resolved by referencing the data map produced from the interactive data file. See [Prompt groups](#) on page 386.

Express Batch

Selecting this option significantly improves the performance of the EngageOne Non Accumulated batch processing when output channel processing is not required.

Refer to the EngageOne Administration Guide for further information about configuring and running non-accumulated batch jobs.

Cross browser support

Cross browser support is available for templates containing interactive data allowing for your customer communications to be created within a non-ActiveX environment across all supported browsers. Cross browser support is available as an attribute setting for the publication. See [Working with publications and documents](#) on page 156 for further details.

Enabling cross-browser support on a publication generates additional prompt logic information in the EngageOne Template. This additional information is a sub-set of the logic, defined in a Publication, that allows for the creation of the appropriate Data Prompts in the browser. When designing EngageOne templates the restrictions listed below should be kept in mind.

Note: Existing EngageOne templates may need to be modified to make them suitable for deployment in the non-ActiveX environment.

The following restrictions will prevent publishing of templates into a Cross-Browser environment:

Scenario 1 where:

A variable assignment is made using one of the following:

- A system variable
- A function, for example, Arithmetic, Concatenation, Substring, etc.

- and -

The variable is subsequently used in conditional logic such as, Condition, Formula Condition, Case, etc.

In this scenario, cross browser support will not be available where the conditional logic contains branches with prompted data fields or interactive variables.

Note: When publishing prompt logic any conditional logic branches that do not contain prompted data fields or interactive variables will not be included in the prompt logic information added to the EngageOne template. These branches will still be executed by Generate during document composition.

Any prompted data fields or interactive variables that are referenced in the variable assignment will be generated as data prompts in the published prompt logic.

Scenario 2 where:

Comparisons in conditions, such as, Condition, Formula Condition, Case etc. are made with system variables or functions

- and -

the condition has logic branches that contain prompted data fields or interactive variables.

Note: If the logic branches of the condition do not contain any prompted data fields or interactive variables and the comparisons of the condition do not contain references to data fields (or interactive variables) then the whole condition will not be included in the prompt logic added to the EngageOne template. The branch will however be executed by Generate during document composition.

Scenario 3 where:

Comparisons in conditions, such as, Condition, Formula Condition, Case etc. are made with system variables or functions

- and -

when that condition has any logic branches that **do not** contain prompted data fields (or interactive variables), and set a variable. In this scenario the set variable action will not be published in the prompt logic. Unexpected behavior may result if this variable is subsequently used in a condition to evaluate whether interactive data should be prompted for.

Note: The whole condition will not be included in the prompt logic added to the EngageOne template. The branch will however be executed by Generate during document composition.

*Support is not available in **Active Content Groups** where:*

- the **Interactive** property is set
- the group contains Active Content which refers to prompted data fields or interactive variables
- the group contains properties that refer to prompted data fields or interactive variables

*Support is not available in **Message Signal Conditions** which contain:*

- logic that references prompted data fields or interactive variables
- any Set Variable actions that are unsupported (see Scenario 1 above) and the variable used in a condition (see Scenario 2 above)

Support is not available in templates containing:

- **Editable text:** this will not be published in the prompt logic and will not be available when a document is created in a non-ActiveX environment.

Note: If the same template is used in an ActiveX environment then this functionality will be available.

- **Page Setup logic:** this will not be published in the prompt logic. So, for example, any set variable actions in this logic will be ignored. Therefore, if this variable is subsequently used in a condition to evaluate whether interactive data should be prompted for, then this may result in unexpected behavior.

Note: These types of template can be published in the cross browser environment but the functionality will not be supported at the prompting stage of document creation:

- **Modify Prompt:** templates with these objects will not be publishable,
- **Prompt Loop:** templates with these objects will not be publishable,

Limited support is available for the following:

- **Keyed Image and Keyed Active Content** - where the return code is stored in a variable. Unexpected behavior may occur if this variable is subsequently used in a condition to evaluate if prompting is required for interactive data.
- The logic in **Publishable Active Content** will not be included in the prompt logic information added to the EngageOne template. This logic will still be executed by Generate during document composition. This may cause problems where the template logic uses variables set inside the Active Content logic.
- **Repeating Data** - data prompts will only be created for any prompted data field or interactive variable used in the repeat count. The logic in the 'before', 'every' and 'after' branches will not be included in the prompt logic information added to the EngageOne template. These branches will still be executed by Generate during document composition.
- **Fields in multi-lingual paragraphs** - all fields will be prompted for, not just those appropriate for the last changed locale action.

How to publish for EngageOne

To publish for EngageOne:

1. In the navigator browse to and select the publication or Active Content to be published.
From the **Tasks** menu select **Publish for EngageOne**. The **Publish Wizard** is initiated.

2. Specify version labeling.

You can specify the label you want to be applied to all the components of this publishing task – the publication or Active Content, resources etc. If you don't want the publishing task to be version controlled, for example if you are testing, then select **None**.

Note

- Selecting the **Create publish archive** option will create a branch of the production job.
- De-selecting the **Include thumbnail and PDF preview** option will not create a preview PDF and thumbnail image for the publication.

3. When publishing a non-interactive template the **Express Batch** option is available.
Selecting this allows you to specify additional file and configuration settings for non-accumulated batch processing in EngageOne and may provide a significant performance benefit.
4. Specify additional fonts.

When publishing a publication you can use the **Additional resources** page to add fonts that will be made available to the front office user – see “Additional Resources”.

5. Select additional Active Content to publish.

When publishing Active Content you can use the + and - buttons to add and remove Active Content as required. Note that if you select a publication, any publishable Active Content that it references will be added.

6. Publish.

Click **Finish**. This will publish either an EngageOne template package created from a single publication, or one or more Active Content objects in a single zip file for the EngageOne Interactive environment. Save the zip file in the required location.

General Publish Wizard options

Publish Options	
<i>Versions</i>	
Label type	<p>Specify a label for the associated production job:</p> <p>System – applies the date and time only;</p> <p>User – allows you to enter your own label name and description;</p> <p>None – this instance of the production job will not be version controlled so will not show in the history view and will not be available for the Publish by label option.</p>
Create publish archive	<p>This option branches the production job being used. The branch can be used as normal. Note that the User or System label previously supplied will used to identify the branch.</p>
Include thumbnail and PDF preview	<p>This is option controls the creation of a preview PDF and associated thumbnail image during publishing of an interactive template. Note that by default, this option is selected.</p>
Express Batch	<p>This option allows you to specify resources and configuration settings required by non-accumulated batch When this option is selected you can assign:</p> <ul style="list-style-type: none"> • EngageOne Compose compatible output devices. • Journals • Lookup tables • External key maps <p>Refer to Express Batch on page 408 below and to the EngageOne Server Administration Guide for further information on configuring and running non-accumulated batch processing.</p>

Include resources for advanced graphics ...

This includes resources that are required when Active Content is used with advanced graphics features, such as in a rotated text box.

Additional Resources

You can specify additional fonts that will be made available to a user when working with editable text in the front-office environment.

Active Content

This enables you to specify more than one Active Content for publishing. Use the + icon to browse the repository for publications. Any publishable Active Content objects found within a selected publication are added to the publishing task, as well as:

- any publishable Active Content objects referenced within other Active Content,
- publishable Active Content referenced by keyed Active Content maps.

Use the - icon to remove Active Content objects that you don't want to publish.

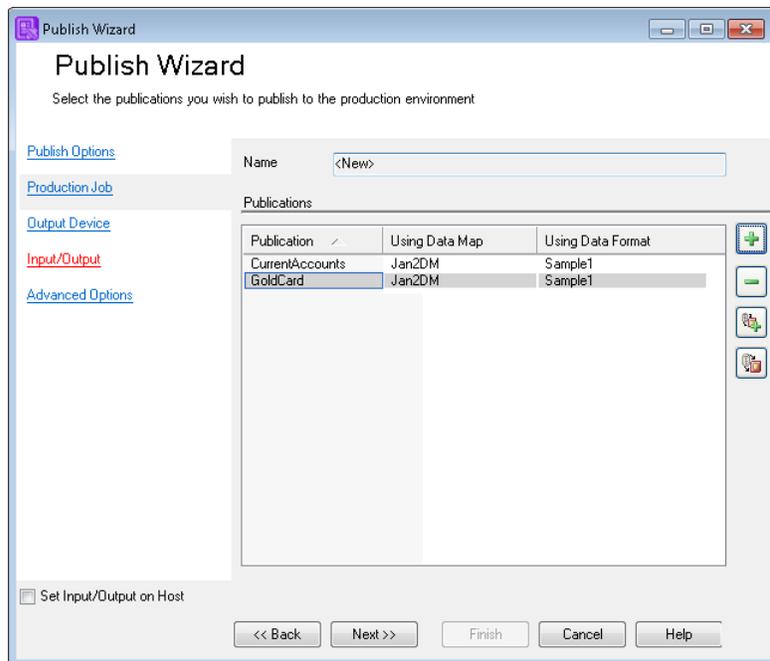
Express Batch

Additional **Express Batch** publish wizard options

Set Input/Output on Host - Select this option to use an OPS file for the path/filename of the production job's input and output files.

Production Job

Publications

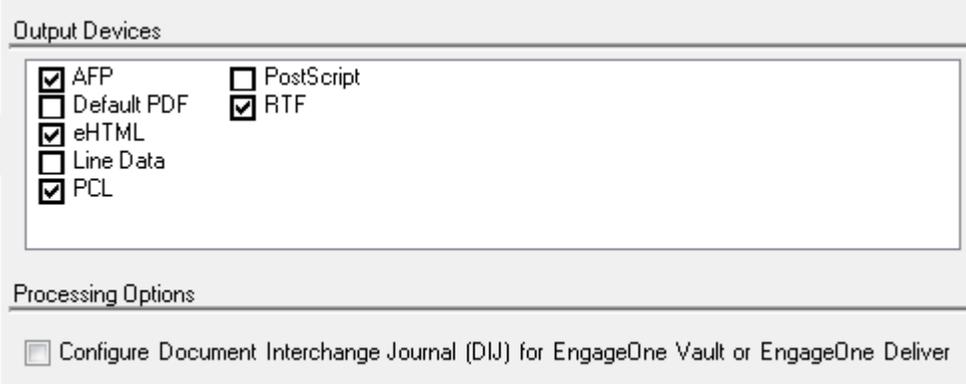


- Use the buttons to add or remove publications that will form part of the production job.
- Click **Assign Data Map** to associate the publication with a data map.
- Click **Edit Data Map** or double-click an entry to amend the data map.

Additional **Express Batch** publish wizard options

Output Device

Output devices



Output Devices

AFP PostScript
 Default PDF RTF
 eHTML
 Line Data
 PCL

Processing Options

Configure Document Interchange Journal (DIJ) for EngageOne Vault or EngageOne Deliver

- Choose from the list of available output device objects and modify its settings if required.
- You can publish to more than one output device.
- At least one device must be selected. You cannot therefore unselect a device if it is the only one in the list.
- Devices where the Enable as EngageOne is selected in the output device's settings will be enabled by default on this screen.
- All devices supported by EngageOne will be shown in this list for selection.

Processing options

Configure DIJ ... If selected a Document Interchange Journal (DIJ) will be created when the job is processed by Generate. This is required when the output datastream is to be used with the Vault, Liaison and EngageOne Deliver products. When this option is selected an additional page is provided in the wizard to allow you to specify the file to receive the DIJ and to provide related settings.

Input/Output

Journals - refer to [Specifying Journals](#) for further information.

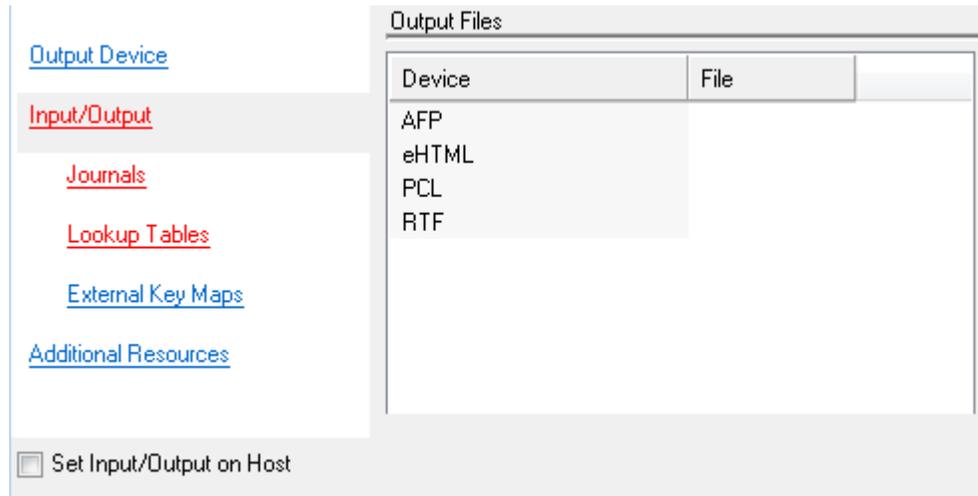
Note: Only the Structured XML Journal is supported in Publishable Active Content used with **Express Batch** published templates. **Any other journal type entries will not appear in the output.**

Lookup tables - refer to [Specifying Lookup Tables](#) for further information.

Additional **Express Batch** publish wizard options

External key maps - refer to [Specifying External Key Maps](#) for further information.

Note that if the **Set Input/Output on Host** is not selected, missing file assignments are indicated in red as shown below.



Publishing resources for EngageOne

EngageOne resources (except referenced keyed images, lookup tables, and RTF files) are published in conjunction with EngageOne templates in the EngageOne publishing process if the **Enable as EngageOne Device** check box is select within the supported EngageOne output devices.

When you publish resources for EngageOne this includes publishable Active Content as well as font and image resources. See [Creating output device objects](#) on page 524 for details.

Note that keyed images are handled differently in EngageOne. For details see [Prompt groups](#) on page 386.

When the completed template is imported into EngageOne, resources for that template are gathered by querying all the templates for the resources in the list. Using the **Delivery Management/Resources/Export Resource List** option in EngageOne Administration you can export template resources to a local file in XML format, which is passed manually to Designer.

The **Publish Resources for EngageOne** option in Designer allows the input file to be browsed and read and then prompts for the target output device definitions to publish to. A single ZIP file, which contains a ZIP archive for each selected device, is created that contains all the resources for the selected devices. This ZIP file is then imported into EngageOne Administration to create or update all devices installed on the server.

If publishing cannot complete because the requested file contains missing resources you must resolve them and then rerun the publishing process. Possible reasons could be that:

- Resources referenced by the templates were published from another Designer repository
- Resources are missing due to a database restore from an older backup
- Resources may exist, but on a different project

Additionally, a check is done to ensure that the Designer repository used to publish the resources matches that of the template resources.

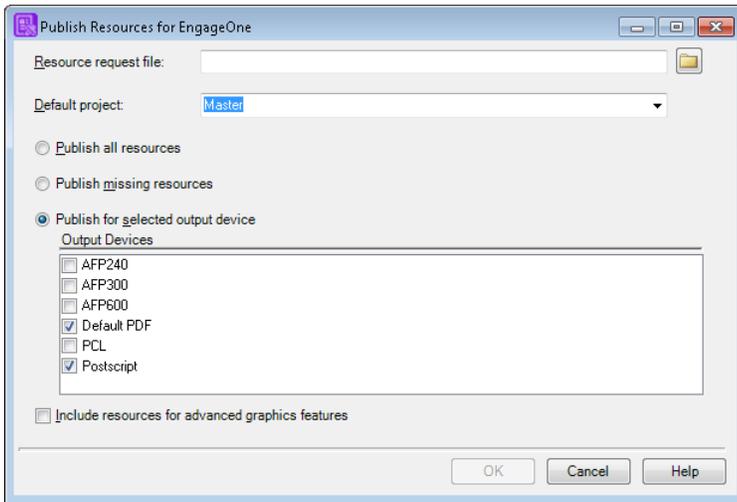
The basic process flow is:

- Import template from Designer into EngageOne.
- Export resources from EngageOne to Designer.
- Create devices for the resources in Designer.
- Import devices into EngageOne.

However, because resources can be published along with the template, resources for a new device can be published without having to publish every template.

For example, if you have an EngageOne environment with an APF device and a PDF device, you can introduce a Postscript device by exporting the resource request file from EngageOne Server and use **Publish resource for EngageOne** and select a Postscript device.

To publish resources for EngageOne:



Enter the name of the request XML file produced by the EngageOne application, or you can browse for it.

Select a default project to be searched for resources that can't otherwise be found.

Select one of the publishing options and select one or more EngageOne enabled output devices as required from the project.

If there are multiple output devices with the same name you will be asked to select one of them.

If advanced graphics – such as rotated text boxes – are used then you must select the Include resources for advanced graphics features option.

Select the **Publish Resources for EngageOne** menu option. This displays the dialog box for selecting the output devices required. Click **OK**. Using the Explorer dialog box provided, name and save the file to the location specified by your EngageOne Administrator. A file package is created ready for importing into the EngageOne Server. For more information about delivery management see the EngageOne Administration Guide.

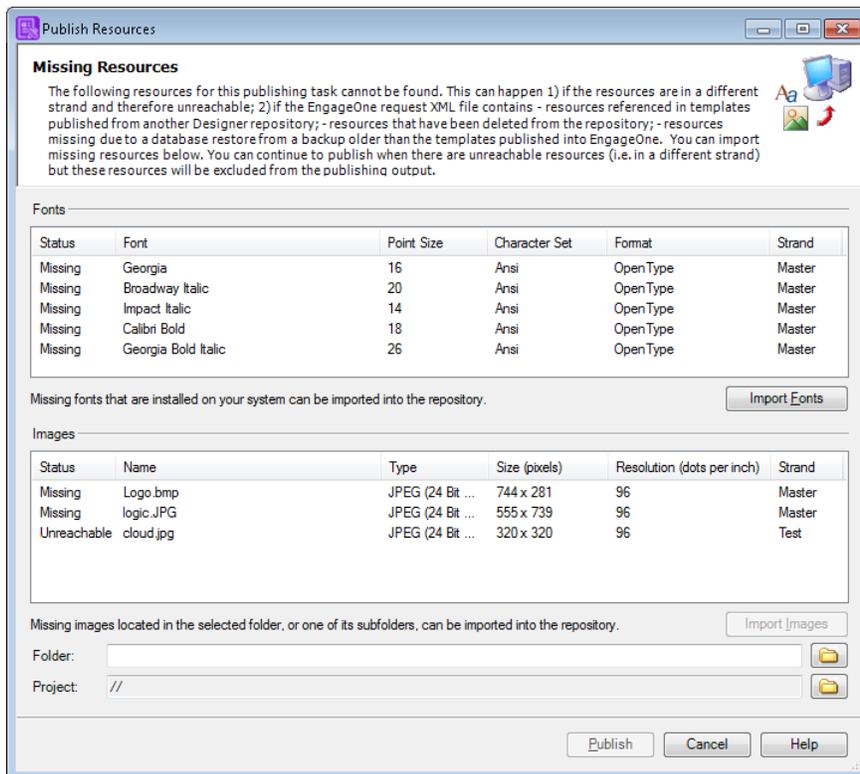
The three publishing options are:

Publish all resources: publish all the resources requested in the EngageOne resource list for all the output devices in the EngageOne resource list.

Publish missing resources: publish only the resources that EngageOne resource list states it does not have for certain output devices.

Publish for selected output device: publish all the resources requested in the EngageOne resource list for the selected output device. This option is useful when a new output device needs to be added to EngageOne Server. The device is created in Designer and then the resources are sent back to EngageOne Server, which then adds the new device.

If a requested resource cannot be found in the Designer repository the **Missing Resources** dialog box will list them all and can be used to reconcile the missing resources.



Example Publish Resources Dialog

- Strand - is the project in which the image was originally published.
- You must import all missing resources, but you can continue publishing without resources that are unreachable, or cancel the publishing process, resolve the unreachable resources and re-publish.

Resource cache management

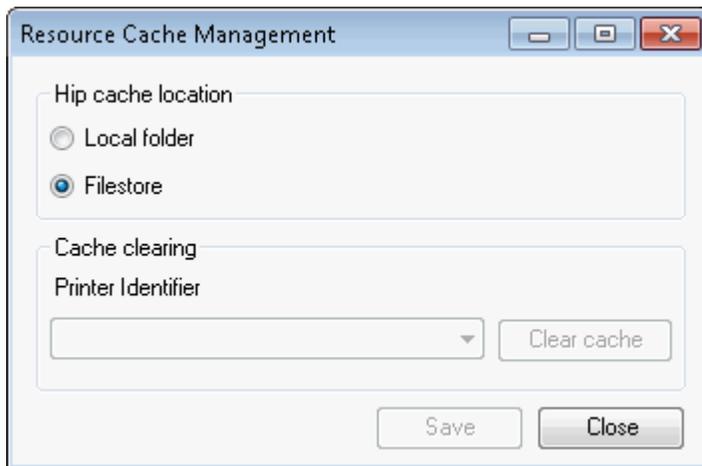
For most channels Designer converts the font and image resources you use in template designs to be compatible with the intended output environment. This can take a significant amount of time particularly where large numbers of resources are used or where fonts are complex.

You can choose to reduce this overhead by caching converted fonts and images (either locally or as part of the repository filestore) so they can be reused when subsequent publishing tasks require the same resources. Resources for each printer configuration (print stream, resolution, font and color settings) are cached together and you can clear the cache for any of these.

Only those with system management privileges can manage the resource cache and must be the only user logged in.

To set the resource cache location:

from the **Tools** menu click **Resource cache management** and set the location as required.



- A **local** cache can be faster as it does not need to use the network but a cache in the **filestore** is available to all users.

To clear resources in the cache:

from the **Tools** menu click **Resource cache management**, select the printer to be cleared and click **Save**.

Import and exporting interactive data definitions

You can use your own XML files to create an interactive data definition by importing your XSD schema and optional XML sample and XForm files into Designer. The XSD file is the only required file, but if you have XML sample files and XForm files, the import process can use these files to better detect the structure and transform it into a more complete interactive data definition that you can then use to generate EngageOne templates, data maps, and data dictionaries. The XSD schema contains the data structure. The XML file contains the sample data that corresponds to the data structure. See www.w3schools.com/schema/default.asp for detailed information about schema files.

When you import an XSD schema, the import process makes a “best effort” approach in transforming the document structure into a valid interactive data definition. It is important to understand how this works before you begin. Examples are included later in this section; see [Example file sets](#) on page 426

- The XSD schema must be well formed. If it is not, the import process will not proceed. In all other cases, the import process will proceed and handle each element on a case-by-case basis. If an irrecoverable error does occur, the element is skipped with a message indicating the reason for the error. The most common reasons for these types of error are incorrect XSD structure and unsupported data types.
- When parsing XML files, errors will be highlighted if the XML does not match the XSD structure. XForm files are also parsed and elements are matched to the XSD and XML sample, however, missing or surplus elements in these files are not considered as errors.
- Your XML files will most likely not contain interactive data definition-specific elements attributes. If these attributes are present, the import process reads them in and uses them. If not, a default value is assigned.
- When you initially import your XML files, the import process provides error messages, warnings, and information to help you understand how the elements in the XSD schema are being interpreted. After this process is completed, you can edit the interactive data definition in the Interactive Data Editor.
- If an XForm file is included that conforms to the rules and layout that the Interactive Data Editor requires, all prompts will have labels and help. If there is no XForm file included, all prompts will not have labels or help, or there will be default values based on the field names.
- If an XML sample file is included, the import process attempts to detect repeating groups from the same data. After the import process completes you can also mark groups as repeating data.
- When you export the interactive data definition the resulting XSD schema and optional files are not identical to the original files you imported because the import process only uses supported elements and data types. Unsupported elements and data types are ignored.

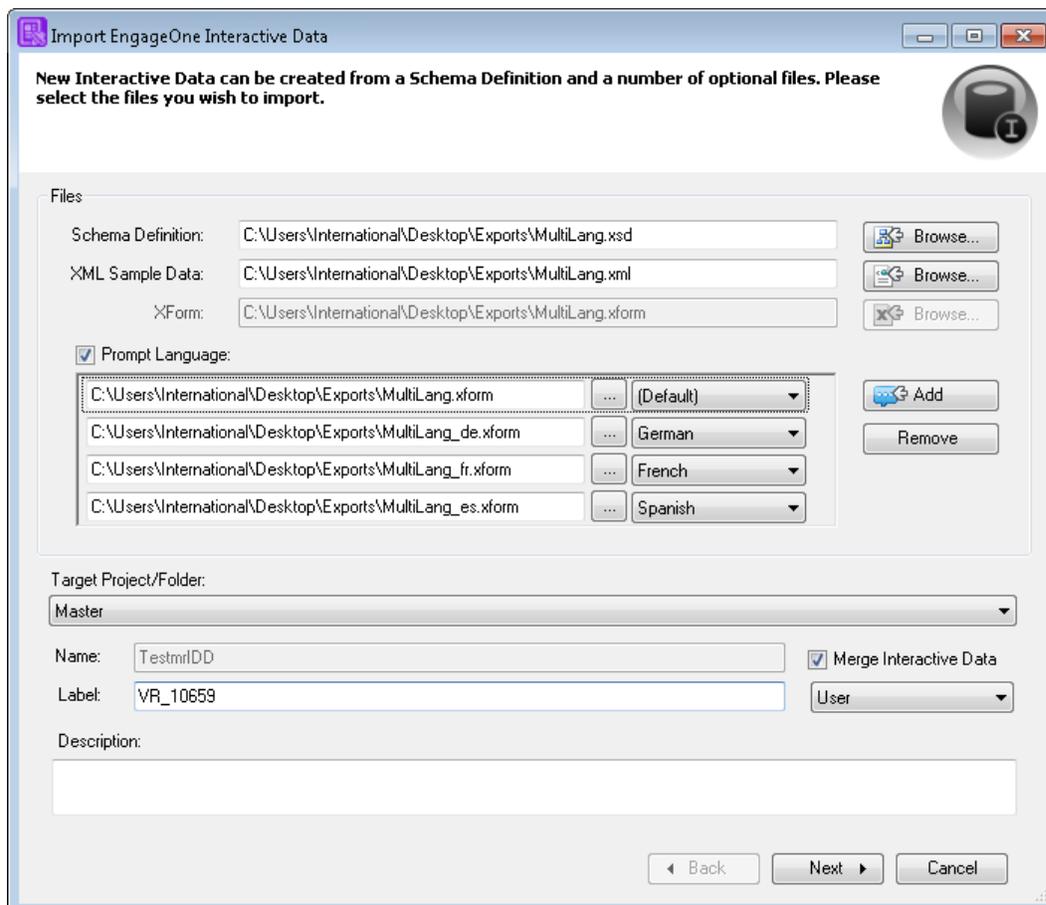
Importing an XSD file

An XSD file is required when importing an XML document structure to transform it into an interactive data definition. Optionally, you can include an XML sample file and an XForm. For more information, see [XML sample files](#) on page 422 and [XFORM files](#) on page 423.

To import an XSD file:

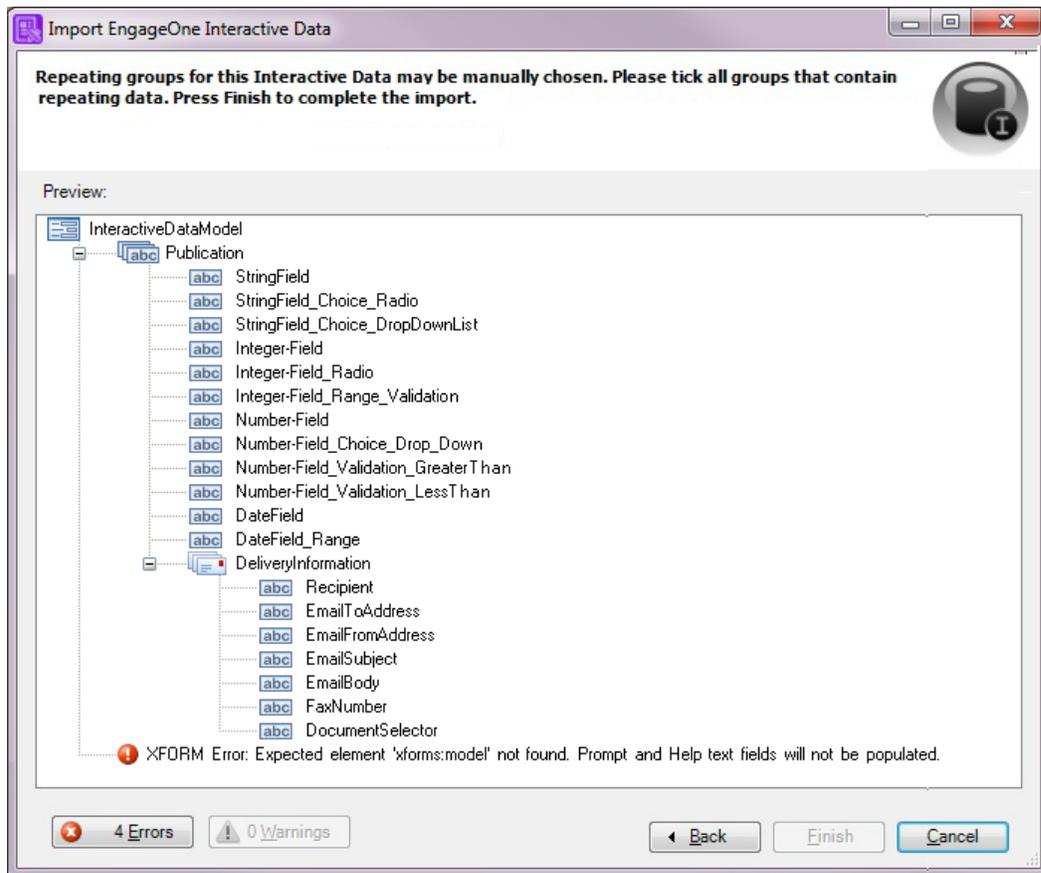
1. On the **Tasks** menu, select **Import > Interactive Data Definition**.

The **Import EngageOne Interactive Data wizard** is initiated.



- Use the Browse option to located your XSD file.
- Selecting the Prompt Language option allows you to add your multi-lingual prompt and help text XML files.
- Assign a Label for version control purposes. Note that this option is only available if you have chosen to Merge Interactive Data and the User option has been selected.

2. Select the files you want to import:
 - Click **Browse** to select the Schema Definition (XSD file). If there are matching XML and XForm file names in the folder with the selected XSD file, the wizard automatically adds those files.
 - Optionally, click **Browse** to select the matching XML Sample Data from a different folder.
 - Optionally, click **Browse** to select the matching XForm from a different folder.
3. If you are working in a multiple language environment, selecting the prompt language option allows you to associate language specific XForm files to your data definition. These files will contain localized help and prompt text. Typically the language is automatically detected by the import process, however, you can optionally override this setting as required.
4. In the **Target project/Folder** list, select the project or folder where you want to save this interactive data definition. Note that if you wish to save the interactive data definition to a specific folder, you must ensure it has first been selected in the Navigator before the Interactive Data Wizard is invoked.
5. In the **Name** field, enter a name for the interactive data definition or accept the file name as the default value.
6. Optionally, enter a description for the interactive data definition.
7. If you wish to merge new interactive data from an XSD file with an existing interactive data definition select the Merge Interactive Data option. Refer to [Merging interactive data](#) on page 420 for further information. **IMPORTANT:** The merge option is only enabled if you have selected a data definition in the Navigator prior to performing the import task and the Interactive Data Editor is not open in your Designer session.
8. Click **Next**.

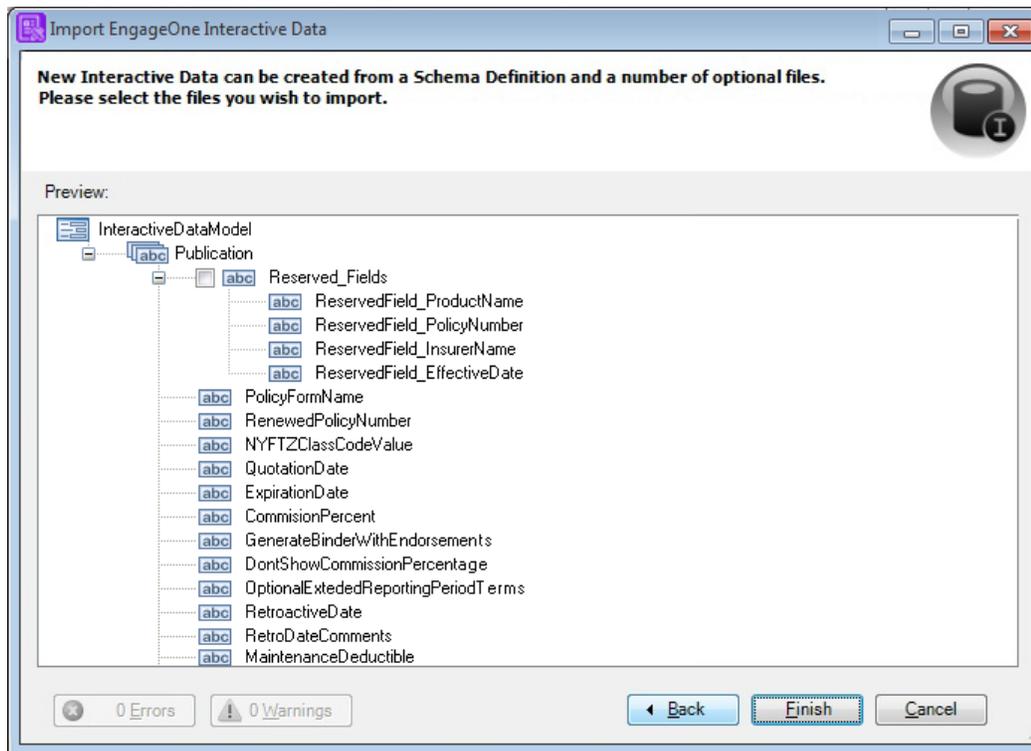


9. Click the **Errors** button to view each error. To correct the error, edit the XSD, XML, or XForm file in your XML editor.
10. Click the **Warnings** button to view each warning.
11. Click **Finish**.

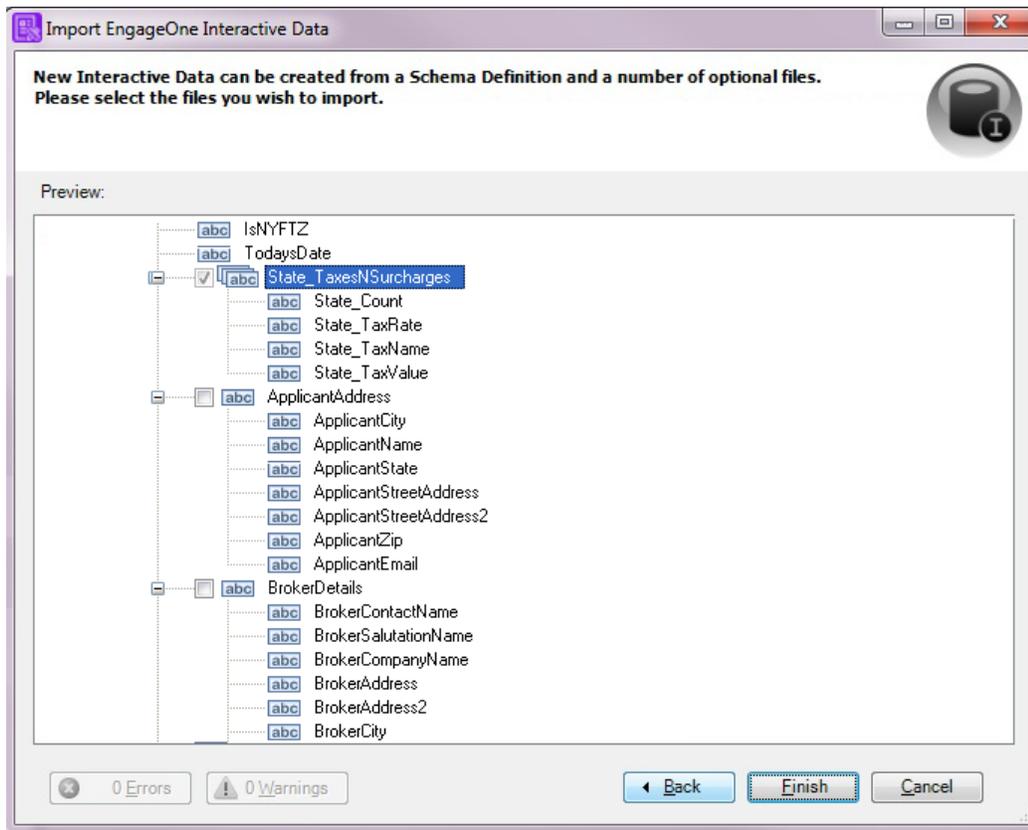
Working with repeating groups

If the file set includes repeating groups, the import process will automatically include a check box for those elements that are detected as repeating group. Only group elements have the repeating group check box enabled. Non-group elements do not have the check box available.

Select the check box if you want the element to be treated as a repeating group.



Clear the check box if you do not want the element to be treated as a repeating group.

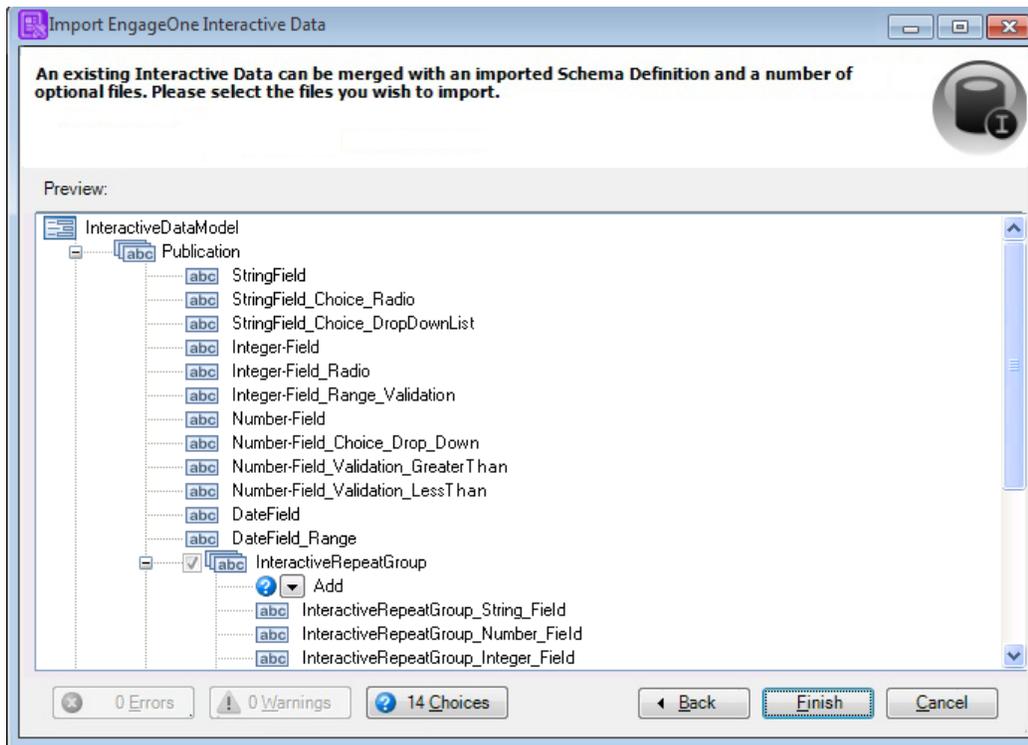


Merging interactive data

You can merge new interactive data from an XSD file with an existing interactive data definition.

Note: Adding a new repeating group by merging a new interactive data definition into an existing interactive data definition does not retain the repeating group data values.

To merge new data structures with an interactive data definition:



1. In the Navigator, select an existing interactive data definition.
2. On the Tasks menu, select Import > Interactive Data Definition.
3. Select the files you want to merge:
 - Click **Browse** to select the Schema Definition (XSD file). If there are matching XML and XForm file names in the folder with the selected XSD file, the wizard automatically adds those files.
 - Optionally, click **Browse** to select the matching XML Sample Data from a different folder.
 - Optionally, click **Browse** to select the matching XForm from a different folder.
4. In the **Target Project/Folder** list, select the project or folder where you want to save this interactive data definition.
5. Click Merge Interactive Data. The Name field is disabled when this check box is selected.
6. Click Next.
7. Click the **Errors** button to view each error. To correct the error, edit the XSD, XML, or XForm file in your XML editor.
8. Click the **Warnings** button to view each warning.
9. Click the Choices button to view each added element. Click the drop-down button to choose an action for the new element:
 - Add (default)
 - Skip

Choose an action for existing elements in the original definition:

 - Keep (default)

- Replace
10. Click **Finish**.

Exporting an interactive data definition

To export an interactive data definition:

1. On the **Tasks** menu, select **Export > Interactive Data Definition**.
2. Choose a location to save the file.
3. Click OK.

XML sample files

Use optional XML sample files to define the default values for elements and the number of repetitions for repeating groups. If an XML file is supplied to the import process, it has to conform with the XSD file.

Supported XML structure

The XML structure must conform to the XSD schema structure. Tags in the XML document must correspond to the values of the name attributes in the XSD schema. Tags representing simple elements can have a text value that will be treated as the default value for that element.

Repeating groups usage

The XML file represents the sample data part of the interactive data definition. As such, it can have values for repeating groups defined. Repeating groups are used to define a series of elements that need to be repeated a certain number of times. Each repetition, or instance, can have different values for the fields contained in the group. Also, a repeating group can contain other repeating groups and in those cases, instances of the nested groups can have different values for each instance of the parent group, which you can define in the interactive data definition.

The import process detects repeating groups by looking through the token tree and checking how many XML tags corresponding to the name of the token are present. Since it is not permitted to have two elements with the same name under one tag and treat them as separate, such occurrences is interpreted as representing a repeating group. Contents of that group are expected to be the same in each instance.

Repeating groups can be nested; in which case the inner group's instances will depend on the outer group's instances.

Here is an example of two nested repeating groups:

```
<OuterGroup>
  <FirstLevelField>3</FirstLevelField>
  <InnerGroup>
    <SecondLevelField1>A1</SecondLevelField1>
    <SecondLevelField2>B2</SecondLevelField2>
  </InnerGroup>
  <InnerGroup>
    <SecondLevelField1>A2</SecondLevelField1>
    <SecondLevelField2>B2</SecondLevelField2>
  </InnerGroup>
</OuterGroup>
<OuterGroup>
  <FirstLevelField>1</FirstLevelField>
  <InnerGroup>
    <SecondLevelField1>Z1</SecondLevelField1>
    <SecondLevelField2>Y1</SecondLevelField2>
  </InnerGroup>
</OuterGroup>
<OuterGroup>
  <FirstLevelField>5</FirstLevelField>
  <InnerGroup>
    <SecondLevelField1>001</SecondLevelField1>
    <SecondLevelField2/>
  </InnerGroup>
  <InnerGroup>
    <SecondLevelField1/>
    <SecondLevelField2>GG4</SecondLevelField2>
  </InnerGroup>
  <InnerGroup>
    <SecondLevelField1>003</SecondLevelField1>
    <SecondLevelField2>GG7</SecondLevelField2>
  </InnerGroup>
</OuterGroup>
```

The outer group has three instances. In the first instance, the field has a value of 3 and there are two instances of the inner group, each with two fields. The second instance of the outer group has a field with value of 1 and one instance of the inner group. The last instance of the outer groups has a field with the value of 5 and three instances of the inner group.

Fields are not required to have values (empty tags), however, the fields are expected to be there even if they don't define a value.

XFORM files

The optional XForm file defines the display properties for the elements. Its structure is considerably different from the XSD and XML files with the tag hierarchy only loosely referencing elements from XSD and XML files.

Supported XForm structure

The import process supports the standard XForm structure. While XForm has to be well formed, there are several important points:

- bind, input and select1 tags have a ref element that must have a value that defines the path to the element. This path must include the root element from the XML file.
- bind elements define a type attribute and while it should match the type defined in the XSD schema, the import process does not verify it, nor will it use this value. Type is defined by the XSD schema and that value is used by the import process.

Unsupported elements

Unsupported elements are ignored during the import process because the values are predetermined in the Interactive Data Editor:

- xhtml:title
- xforms:instance
- xforms:submission
- Elements in the model section are also ignored, except for xforms:bind.
- Elements in the body section will be ignored, except for span.

Supported data types

Data types supported natively by the editor:

- string (xs:string)
- integer (xs:integer)
- floating point a.k.a. Number (xs:float)
- date (xs:date)
- time (xs:time)

Data types that can be transformed to types supported by the editor:

- xs:normalizedString
- xs:decimal
- xs:byte
- xs:int
- xs:long
- xs:negativeInteger
- xs:nonNegativeInteger
- xs:nonPositiveInteger
- xs:positiveInteger
- xs:short
- xs:unsignedLong

- xs:unsignedInt
- xs:unsignedShort
- xs:unsignedByte
- xs:double

Unsupported XSD/XML constructs and data types

- XSD element tags
 - xs:extension
 - xs:attribute
 - xs:all
 - xs:choice
 - xs:anyAttribute
 - occurrence indicators: minOccurs, maxOccurs
- Restriction tags
 - xs:pattern
 - xs:whitespace
 - xs:length
- Data types
 - xs:token
 - xs:dateTime
 - xs:duration
 - xs:gDay
 - xs:gMonth
 - xs:gMonthDay
 - xs:gYear
 - xs:gYearMonth
 - xs:boolean
 - xs:hexBinary
 - xs:anyURI
 - xs:base64binary
 - xs:NOTATION
 - xs:QName

Example file sets

Here is an example of a matching XSD schema, XML sample, and XForm file set.

XSD schema example

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="GroupElement">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="StringElement" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="IntElement" type="xs:integer" />
  <xs:element name="AnotherString" type="xs:string" />
</xs:schema>
```

XML sample file example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Core>
  <GroupElement>
    <StringElement>String default value</StringElement>
  </GroupElement>
  <IntElement>0</IntElement>
  <AnotherString/>
</Core>
```

XForm example

```
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:ev="http://www.w3.org/2001/xml-events"
  xmlns:xforms="http://www.w3.org/2002/xforms"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xhtml:head>
    <xhtml:meta content="1.5" name="version"/>
    <xhtml:title>InteractiveDataModel</xhtml:title>
  <xforms:model>
    <xforms:instance src="3762494494e44df9a4576e17e0b64598.xml"/>
    <xforms:submission action="instance_GUID.xml" id="main_submission"
  method="put"/>
    <xforms:bind nodeset="/Core/GroupElement/StringElement"
  readonly="false()" required="false()" type="xs:string"/>
    <xforms:bind nodeset="/Core/IntElement" readonly="false()"
  required="false()" type="xs:string"/>
    <xforms:bind nodeset="/Core/AnotherString" readonly="false()"
```

```

required="false()" type="xs:time"/>
</xforms:model>
</xhtml:head>
<xhtml:body>
  <span style="display:none">
    <xforms:input ref="/Core/GroupElement/StringElement">
      <xforms:label>String element</xforms:label>
      <xforms:help>Dont need help</xforms:help>
    </xforms:input>
  </span>
  <xhtml:br/>
  <span style="display:none">
    <xforms:input ref="/Core/IntElement">
      <xforms:label>Int Element label</xforms:label>
      <xforms:help>Don need help either</xforms:help>
    </xforms:input>
  </span>
  <xhtml:br/>
  <span style="display:none">
    <xforms:input ref="/Core/AnotherString">
      <xforms:label>Last string element</xforms:label>
      <xforms:help></xforms:help>
    </xforms:input>
  </span>
  <xhtml:br/>
  <xforms:submit submission="main_submission">
    <xforms:label>Submit</xforms:label>
  </xforms:submit>
</xhtml:body>
</xhtml:html>

```

28 - Active Content

Active Content objects contain segments of document content and logic. They provide a highly flexible way to share and reuse such segments across multiple publication designs.

In this section

About Active Content.....	429
Creating a basic Active Content object.....	429
Using Active Content as a function.....	432
Using Active Content groups.....	435
Using Active Content to update publications in production.....	441
Using Active Content with EngageOne.....	443



About Active Content

In its simplest form Active Content provides a means to make the same segment of document design available to many different publications. By using parameters and return values Active Content can be used to create reusable functions. When used as part of an Active Content group these files can be conditionally linked via property settings to provide a mechanism for dynamically building frequently reused sequences of content.

If you are designing document templates for use with the EngageOne environment you can present a list of optional content for end user selection in the interactive editor using Active Content groups.

Active Content can be published independently and then used to dynamically update one or more publications that are already in production. Active Content can also be called as a keyed object. This allows specific Active Content objects to be called depending on the value of a key usually set by input data. See [Keyed objects](#) on page 267 for details.

It is important to be aware that changes to locale settings made in Active Content logic do not persist outside the scope of the Active Content.

Creating a basic Active Content object

By creating an Active Content object you are making document content and logic that can be inserted wherever it is needed in other design elements. This is sometimes called public Active Content. You could create an Active Content object that organizes customer details and presents them in a consistent fashion. You could reference that piece of Active Content wherever you wanted customer details to appear in a document layout.

Within the Active Content editor, paragraphs are created and manipulated exactly as you would within the main document editor and you can make use of all text related features. You can also include graphics (shapes or images) but note that these must be in-line or anchored to a paragraph; you cannot use a fixed anchor type.



Active Content can be referenced anywhere in other design elements but the content they provide may be automatically adjusted to fit the available space. To help with this you may want to use the **Page Setup** options in the Active Content object to adjust the available workspace to emulate the dimensions of a particular area such as a text box or table cell.

IMPORTANT: if you intend to make Active Content independently publishable, you cannot include data fields or interactive variables directly. For more information see [“Using Active Content to update publications in production”](#).

Otherwise, when working with Active Content you are able to select fields from any data dictionary known to the repository but if you intend to share Active Content across multiple publications you should consider carefully how the fields you use will be mapped to data elements in the actual Active Content input streams if these do not always share the same format. Note that where a data map and sample data, (or interactive data model in the case of EngageOne templates) have already been assigned to a publication these will also be made available to an Active Content object when it is launched from within a publication. See [About synchronization](#) on page 163 for more information.

Working with interactive variables

Interactive variables are another type of variable data that can be included in the design of your Active Content. This type of data is independent of the interactive data definition and is included in your Active Content as an object within the logic map. It is important to note that interactive variables cannot be used with **Active Content** that is published independently. For detailed information on

interactive variables and how to use them refer to [Creating EngageOne templates](#) on page 349 and [Interactive variables](#) on page 362.

Existing Active Content objects can be edited either independently using the Active Content editor in the navigator or within the context of a parent document.

To create an Active Content object:

1. In the navigator browse to the required project or folder and from the **File** menu select **New/Active Content**.
2. The **New Active Content** dialog box is displayed. Enter the Name and optional **Description** and click **OK**.
3. The Active Content editor is launched. Add logic and content in the normal way.
4. Save the new file. Note that only Active Content objects in the same project can be used within a document design.

What the Active Content dialog tabs are used for:

Properties

Properties are used in conjunction with Active Content groups to select the appropriate Active Content objects conditionally according to the values specified. For more information see [Using Active Content groups](#) on page 435. You can also enable Active Content to be published independently – see [“Using Active Content to update publications in production”](#).

Parameters

This tab allows you to specify the expected parameters when using **Active Content** as a function. See the following section [“Using Active Content as a function”](#) for more details.

Languages

If you intend to use localized paragraphs within the file use this tab to set the supported languages. See [Creating and customizing locales](#) on page 311 for details.

Page Setup

Use this tab to specify the size of the Active Content layout space when you want to emulate a specific area in which the content will need to fit.

Sample data

As with a regular publication design you may want to use sample data to see how it will affect the Active Content logic and layout. When editing Active Content from within a parent document such data assignments for the publication are automatically passed down to file being loaded. At other times you can use this tab to assign the required sample data and data map. See [Data mapping](#) on page 496 for more information about these resources.

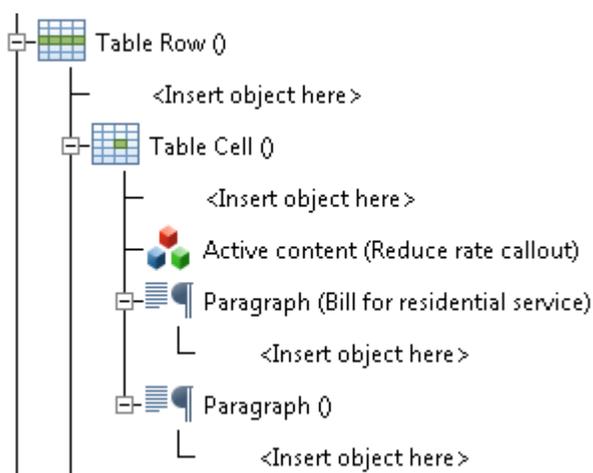
For details about the **System Properties** pane on the **Active Content** dialog box see [“Using Active Content to update publications in production”](#).

To update the attributes of an Active Content object: open the file from the navigator and double-click the main Active Content object.

To reference Active Content in a design element:

In the publication/document position the cursor in the paragraph that any new content is to follow or simply mark the appropriate insertion point in the logic map.

- From the Insert menu click Control Objects/Active Content.



- In the Active Content dialog box give the reference a Label and use the folder button to search for the required object.
- Click Create to add the reference and incorporate the logic and content into the active design.

Using Active Content as a function

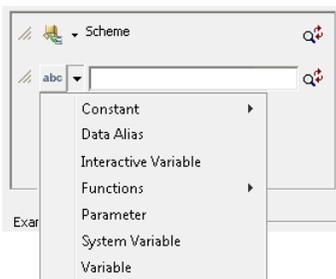
By making use of input parameters and the return value you can use Active Content objects to create repeatable logic functions that can be called as required by other design elements.

You may have an Active Content function that calculates the type of tariff applicable to various transaction amounts. The function would receive the amount as a parameter, evaluate it using a case object and return a string that indicates the type of the tariff applied. You could call the Active Content function wherever you needed to perform this evaluation. Note that Active Content may often be used as both a function and to provide content to the design element that references it.

Note that parameters and return values are not supported where Active Content is called as a keyed object. See [Keyed objects](#) on page 267 for details of this feature.

Any value type can be passed as a parameter to Active Content and these can be used as you would use other values in both logic and layout. Parameters can also include references to other Active Content objects effectively allowing you to nest function calls if required.

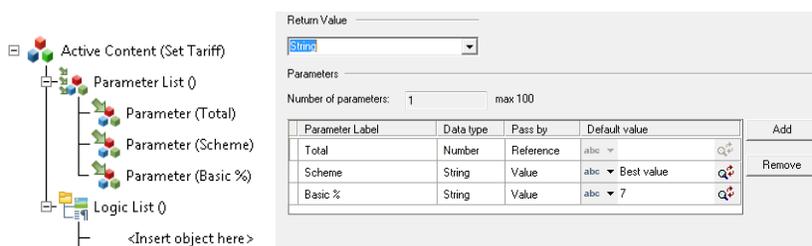
- To pass parameters to an Active Content function double-click the Active Content object within a design element. The **Active Content** dialog box will allow you to assign parameter values.
- To use parameters within Active Content logic: select **Parameter** from the value types when specifying input to any object such as a variable or standard function.



- To use parameters directly within a paragraph: select the appropriate insertion point within a paragraph and from the **Insert** menu click **Field/Parameter**.

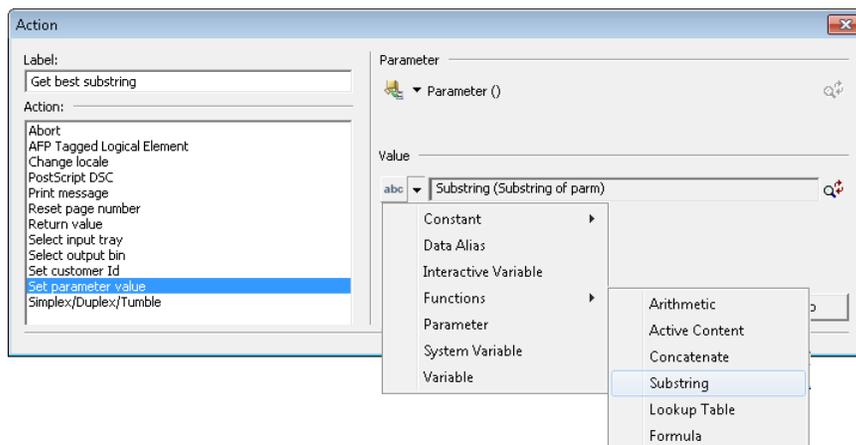
To specify a return value and parameters for an Active Content object:

1. Double-click on the main Active Content object in the logic map to open the **Active Content** dialog box.
2. Use the **Parameters** tab to specify the type of return value and each of the parameters that are expected.
 - a) Select a data type unless the function is not expected to return a value.
 - b) Select Reference if you intend to update a variable within the function.
 - c) Click to add new parameters to the list.
 - d) Enter labels you will recognize when assigning values to parameters.
 - e) The default value is used only when the Active Content is edited 'stand-alone' i.e. not from a publication.
 - f) Update parameter details from the main Active Content object or the Parameter List
 - g) Select **Pass by Reference** to dynamically update the values variables or other functions.



3. Within the Active Content object, mark the insertion point in the logic map where the value required to update the parameter is available.
4. From the **Insert** menu select **Control Objects/Action**.
5. In the **Action** dialog box select an action type of **Set parameter value**.

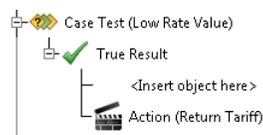
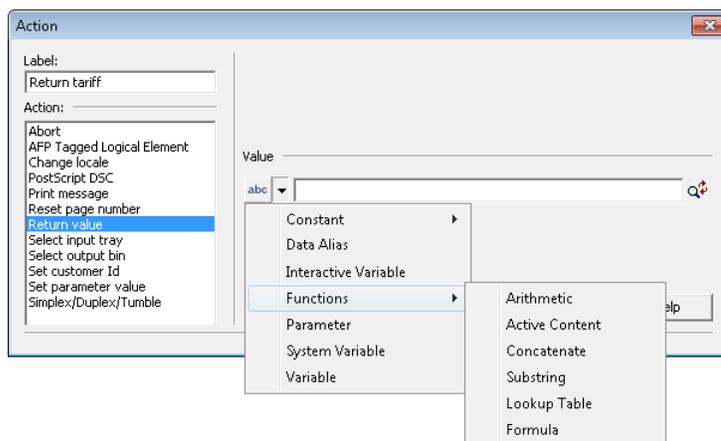
- a) Choose the parameter to be updated.



- b) Select the object to provide the return value. See [Working with values](#) on page 170 for details.

6. To return a value to the calling object.

- a) Within the Active Content object, mark the insertion point in the logic map where the return value will be available.



- b) From the **Insert** menu select **Control Objects/Action**.
 c) In the **Action** dialog box select an action type of **Return value**.
 d) Select the object to provide the return value. See [Working with values](#) on page 170 for details.

Calling an Active Content function that returns a value

Create the object that will receive the value (Variable, Action, Condition, etc.).

Parameter Label	Data type	Pass by	Value
stringval	String	Value	abc mystringval
stringref	String	Reference	abc mystringref

1. Choose from the available Active Content objects or select objects to provide parameter values. See [Working with values](#) on page 170
2. In the appropriate object dialog box select a value type of **Function/Active Content**. In the **Active Content** dialog box identify the required function and specify any parameters.

Using Active Content groups

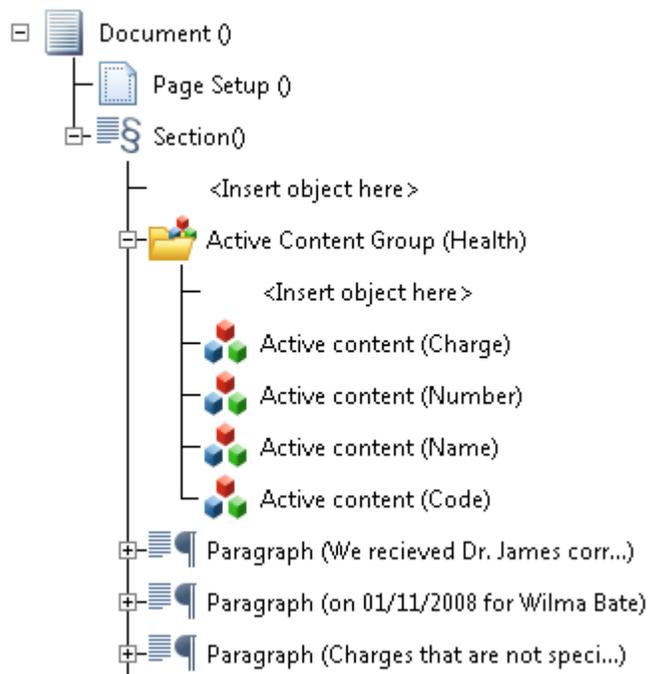
Active Content groups allow you to build linked but variable groups of pre-defined content.

For example, you may want to use an Active Content group to build the body of a customized letter from a range of standard paragraphs or you might want to create a variable 'terms and conditions' section within a document design.

When working with Active Content groups you need to ensure that each segment of content (usually one or more paragraphs) are included in separate Active Content objects. These files are then referenced in the Active Content in the required sequence. Variability is provided in relation to the

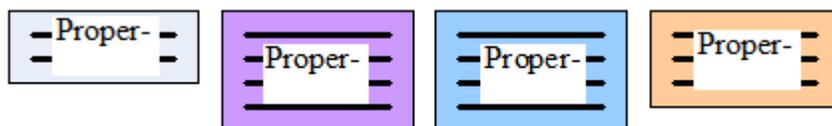
properties specified for each Active Content object: only those files that match the group's property map will be selected. If required you can make an entire Active Content group conditional.

Note: When designing for EngageOne applications you can also specify properties as interactive prompts. See [Active Content and EngageOne](#) on page 392 for details.



Property Map

Property	Operator	Data Source
Day	=	1.23 ▾ 1
Count	=	1.23 ▾ 3
	=	abc ▾

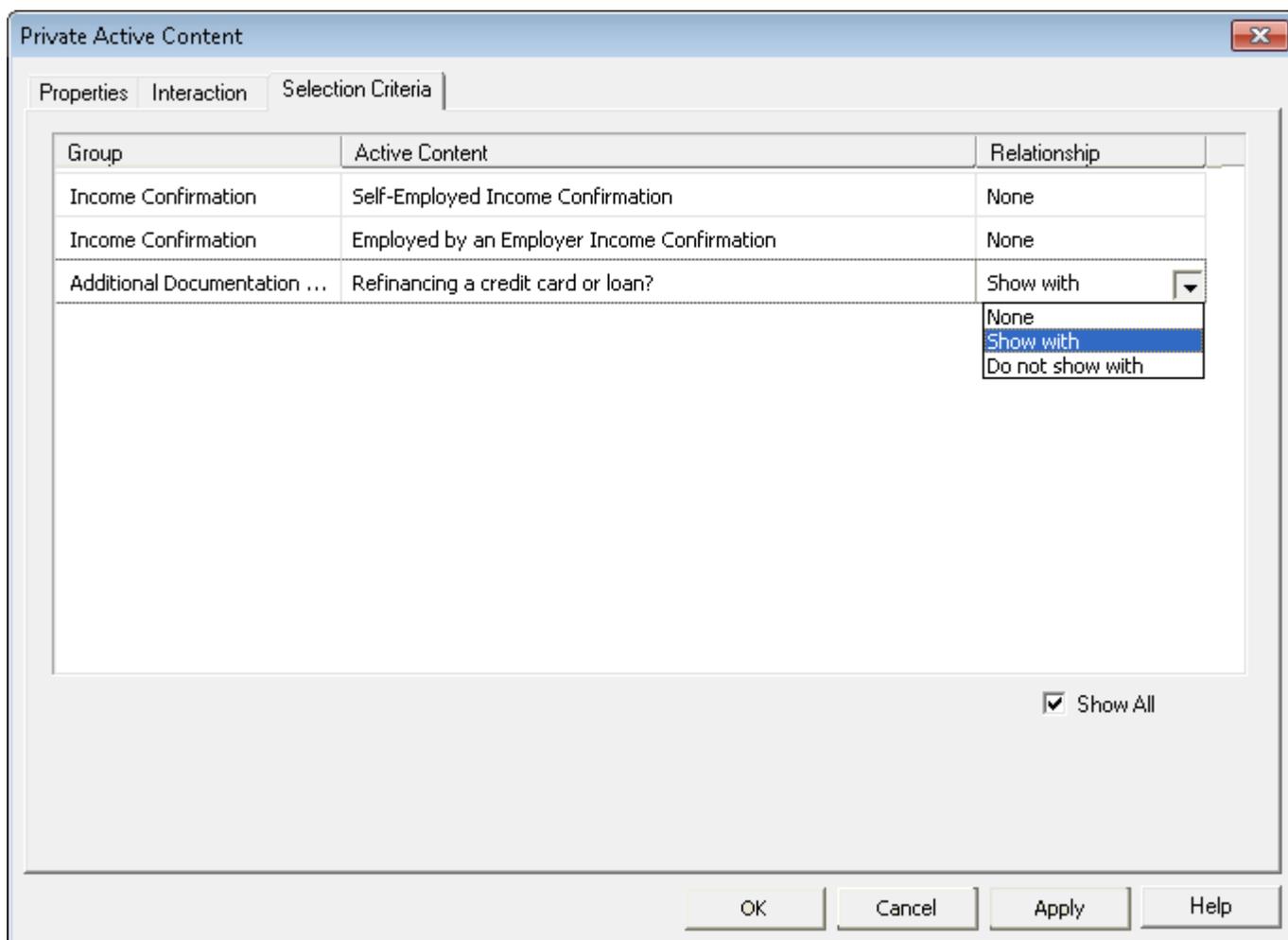


- Presentation objects in Active Content objects assigned to the group
- When the properties in the Active Content objects match those specified in the group **Property Map** the content from those files are included in the document.

Unlike other uses of Active Content when working with groups you may want to create Active Content objects exclusively for use by a single document design. These are known as private Active Content.

To create an Active Content group:

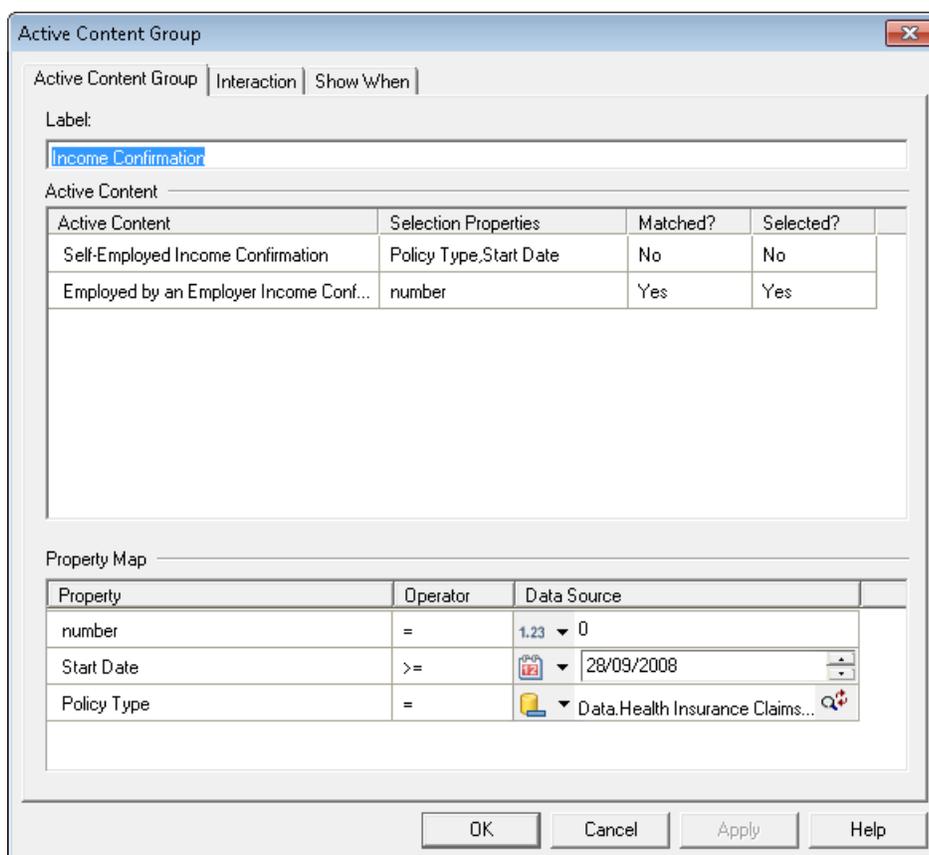
1. Create or update the public **Active Content** objects that will make up the group. Add suitable properties to each. If you are using custom properties ensure that they share the same name and have values assigned
2. Create the Active Content group. In the design element that will include the group, mark the insertion point as required and from the **Insert** menu select **Control Objects/Active Content Group**– or –click the **Active Content Group** button in the toolbar.
3. To add public **Active Content** object references to the group:
 - a) From the **Insert** menu click **Control Objects/Active Content/Active Content**– or –click the **Active Content** button in the toolbar.
 - b) Position the cursor at the insertion point under the appropriate Active Content group.
 - c) In the **Active Content** dialog box give the reference a **Label** and use the folder button to search for the required object.
4. To add private Active Content to the group
 - a) Position the cursor at the insertion point under the appropriate Active Content group.
 - b) From the **Insert** menu click **Control Objects/Active Content/Private Active Content**– or –click the **Private Active Content** button in the toolbar.
 - c) The **Private Active Content** dialog is displayed. If you want to define properties that should be matched, use the **Add** button from the **Properties** tab to define custom properties for private Active Content. If the properties do not match the content will not be included in the document and the object will be grayed out in the logic map.
 - d) The **Selection Criteria** tab shows all the Active Content groups and group entries that precede this active group entry. For each entry in the list one of three relationships can be established, which determines inclusion: **None** – the default value and there is no relationship. **Show with** – the Active Content is only placed in the document if the properties match AND the selected Active Content has been shown. **Do not show with** – the Active Content is only placed in the document if the properties match AND the selected Active Content has NOT been shown.



If checked all Active Content objects placed before this one are shown in the list. If this option is unchecked all Active Content objects with a relationship of **None** are hidden.

The **Active Content Group** dialog shows the Active Content entries within the group, the match status and whether the entry has been selected for inclusion. The associated properties are taken over from the Active Content group entry but can be changed in the **Property Map** for the group.

To map properties associated with the Active Content to a data source, double-click on the required Active Content group object in the logic map. The **Active Content Group** dialog is displayed.

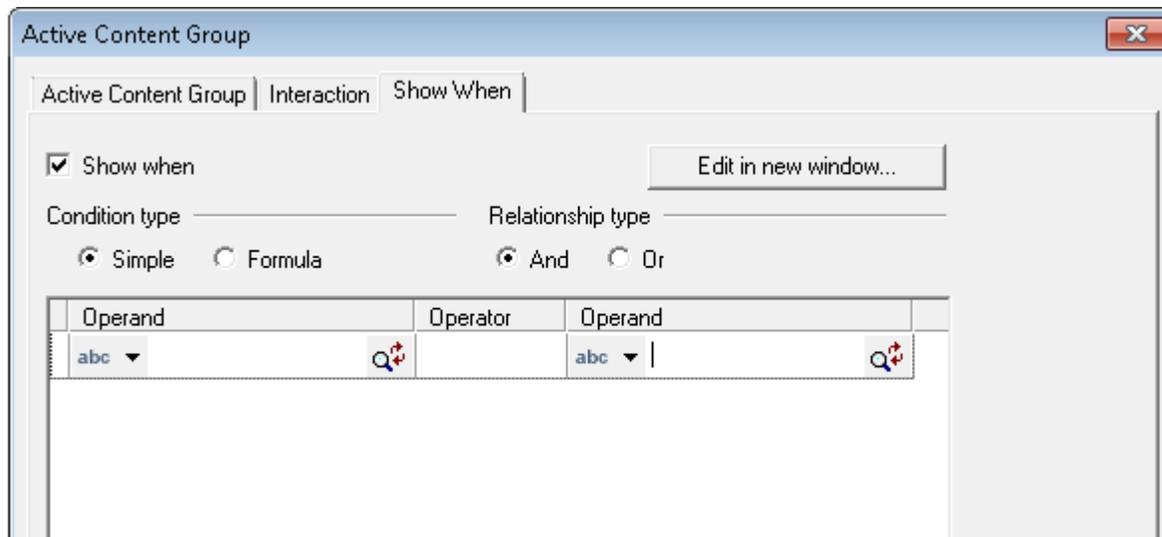


- Shows if the parameter is matched in the **Active Content** group property map.
- Determined by whether the properties and selection criteria have been matched.
- Properties from each of the **Active Content** objects assigned to the group will automatically be listed.
- Use the **Operator** and **Data Source** fields to specify the comparison that will be made for each property
- See [Appendix B Formula expressions](#) on page 716 for information about operator types
- The data source can be any available value but you will normally want to ensure it is variable data. See [Working with values](#) on page 170 for details.

Active Content groups can be made conditional so that they appear when certain criteria are met, for instance, you may want to show a group only if a customer has a certain type of insurance. You will typically want to use values from data aliases and variables that have been made available to the template as part of the condition.

Note that if you make an Active Content group conditional it will not be shown in the editor if the specified condition evaluates to false. To change the settings you will need to use the logic map to work with the relevant Active Content object. See [About the logic map](#) on page 149 for more information.

To make the entire Active Content group conditional, double-click the Active Content group object to display the **Active Content Group** dialog. On the **Show When** tab of the **Active Content Group** dialog box check the **Show When** option and enter the condition as required.



- Selecting makes the Active Content conditional.
- Choose the type of condition.
- When more than one condition is defined select either: **And** – all true **Or** – any one true.
- Enter conditions here. [Creating conditional logic](#) on page 217 for details.

Using Active Content to update publications in production

Active Content can also be published independently. This allows you to update publications dynamically that are already in production. You can enable and disable this feature for each Active Content object.

Note: use this option with caution as the impact on production jobs may be significant. For more information about publishing Active Content see [About publishing and deployment](#) on page 591.

Restrictions

You need to be aware of the following restrictions when working with independent publishable Active Content:

- Data aliases cannot be used directly in Active Content that is published independently. You can however pass them as parameters to the Active Content – see [“Using Active Content as a function”](#).
- Interactive variables cannot be used in publishable Active Content.
- The Active Content must be published from the same repository as the publications in production that refer to it.

To enable independent publishing of Active Content:

1. In the navigator browse to and open the required Active Content.
2. Open the Active Content properties.

From the logic map right-click the Active Content object and select **Properties**.

3. In **System Properties**, click on the **Publishable** value, set it to **Yes** and click **OK**. Note that you cannot set this publishable property to **Yes** if data associated with it is already being used.

Name	Value
Creation Date	18/09/2008
Last Modified	18/09/2008
Last Modified By	admin
Publishable	No <input type="button" value="v"/>
	No
	Yes

4. Save the Active Content.

Publishable Active Content will be included when you publish resources for EngageOne. For more information see [Publishing resources for EngageOne](#) on page 411.

Note that the Active Content HIP file is not created when you publish the publication. You must publish these separately – see [Running a publishing task](#) on page 602.

Using Active Content with EngageOne

Active Content objects play a special role in EngageOne applications. Typically an EngageOne document template will contain **Active Content** groups to present list(s) of optional content to **EngageOne Interactive**. Be aware to check the **Interactive** box on the **Interaction** tab of the **Active Content** group dialog – see [Using Active Content groups](#) on page 435 for more details. Depending on the choices made other content could be included later in the document or another list could be activated.

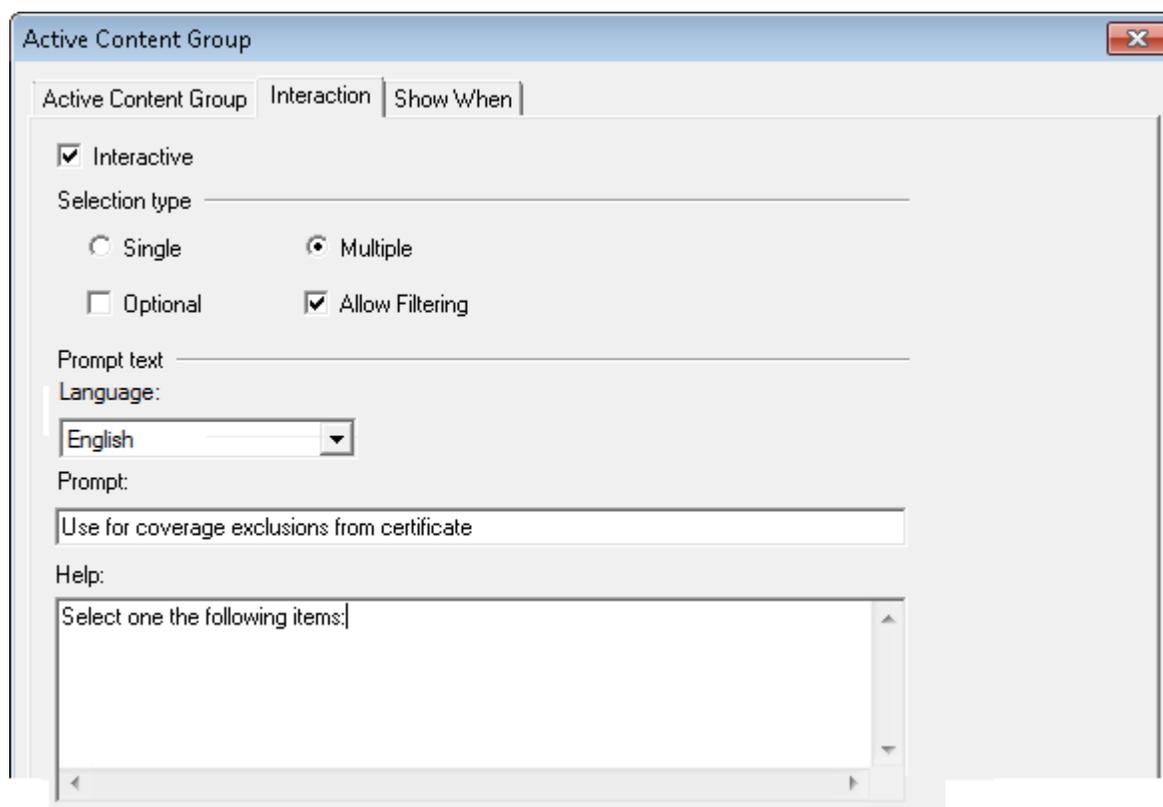
As an EngageOne solution commonly includes a large number of templates it will also often make use of independently publishable Active Content to avoid the need to republish all the templates when minor content changes are required.

For more detailed information about using interactive optional content using Active Content groups you should refer to the Active Content and EngageOne section in [Creating EngageOne templates](#) on page 349.

The **Interaction** tab of the **Active Content Group** dialog is used to specify that the template is to be used in interactive mode in the EngageOne front-office environment and allows you to set particular interactive characteristics for display in the interactive editor.

To make an Active Content group interactive:

1. Create the Active Content group as before – see [Using Active Content as a function](#) on page 432
2. Click on the **Interaction** tab of the **Active Content Group** dialog.



- Interactive will be prompted to select one of the Active Content objects for placement if there is more than one available.
- Interactive will be able to select multiple choices for placement
- Used in conjunction with the **Single** and **Multiple** options. If this option is selected all interactive elements will appear but do not have to be selected.
- Interactive will be allowed to filter the list of optional content presented in the interactive editor by typing into a text control box.
- Select the language to be associated with the prompt and help text. This will be presented in the language chosen by the front-office user when creating the customer communication.
- This help text will be displayed when using the **Help** button on the capture control in the interactive editor.
- This wording will show as a prompt on the capture control in the interactive editor.

Configuring localized text for interactive Active Content object

Localized Active Content text is used where front-office users work in more than one language. In this scenario, the template designer will define Active Content text for the default language. For other languages, the same text is localized and entered with the appropriate **Language** set. When creating an interactive communication the front-office user selects the language to work with, at which point any prompt/help/display text is presented in the chosen language.

In addition to the previously described localized prompt and help text options for the Active Content groups, you can also define localized display text for any public or private Active Content objects that make up your interactive template.

To localize private or public Active Content display text:

double-click an existing Active Content object and select the **Interaction** tab. Enter the **Display Text** for the Active Content object and select the appropriate **Language** from the drop-down list. If you are creating a new Active Content object within an interactive group, you also have the ability to localize display text using the same Interactive options.

Note that if you have upgraded from an earlier version of Designer that does not support localized Active Content text, any pre-existing text will be assigned to the default language setting.

29 - Review and approval

The Design Review and Approval feature provides a framework to ensure new and existing publications and active content are subject to quality control as part of a structured review and approval process driven from the EngageOne portal. To enable this feature, the appropriate project service must be configured; contact your EngageOne Server administrator for further information.

Note: The term **asset** is used throughout this section as a collective term to refer to Publications, Templates and Active Content resources.

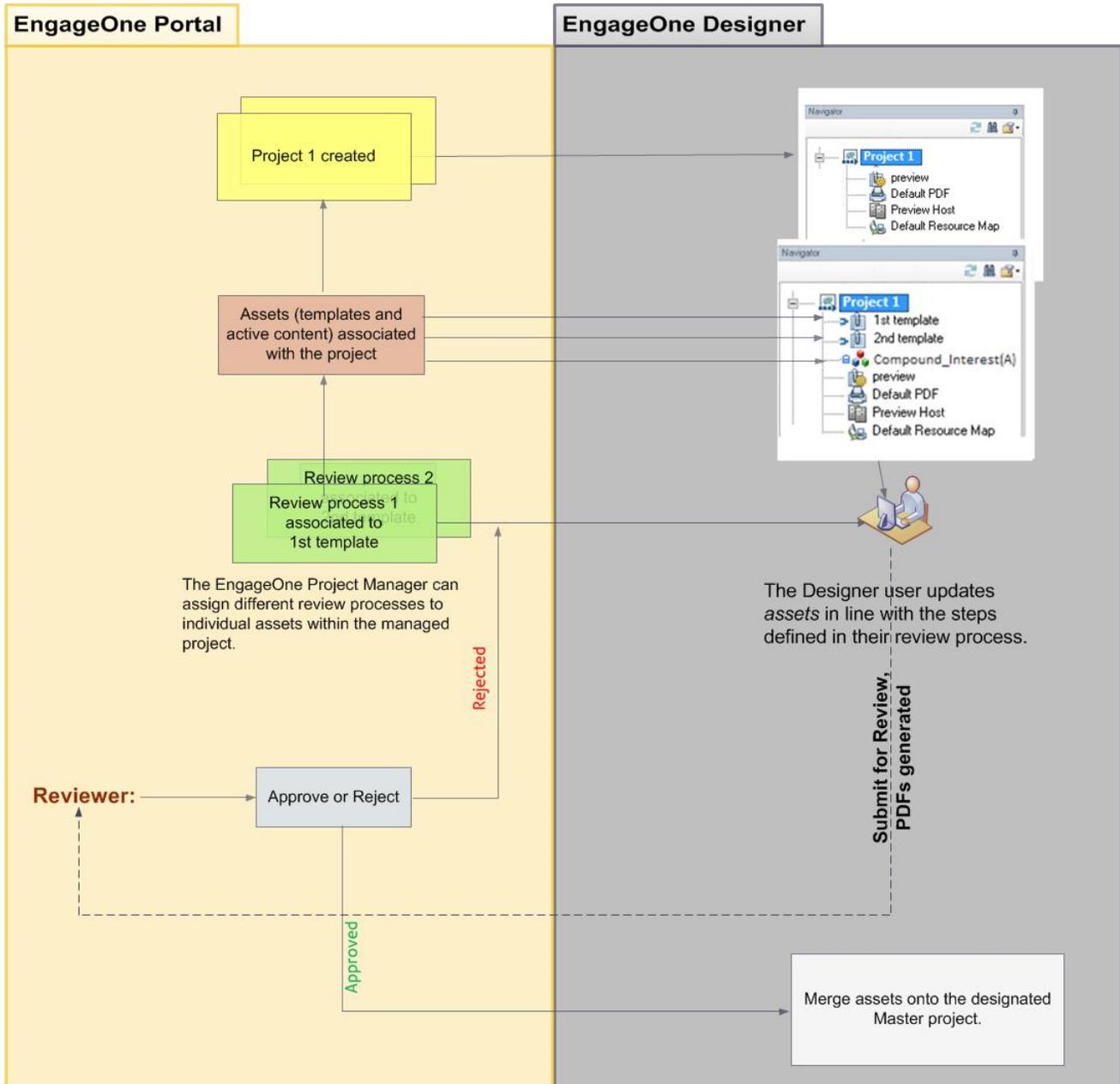
In this section

Workflow.....	447
Working with managed projects and publications in Designer.....	449
Submitting assets for review.....	451
Review comments.....	451
Merging managed projects.....	453



Workflow

The review and approval of Designer publications and Active Content is driven by Project Managers through projects created in the EngageOne portal.



A project manager will:

- Create new assets
- Assign existing assets from the Master project in Designer to the portal based project
- Assign newly created assets to the portal based project.

Note: Projects created in the portal are represented in Designer as managed projects branched from Master. They are visible together with their associated assets in the Designer navigator and visually identified as a managed projects.

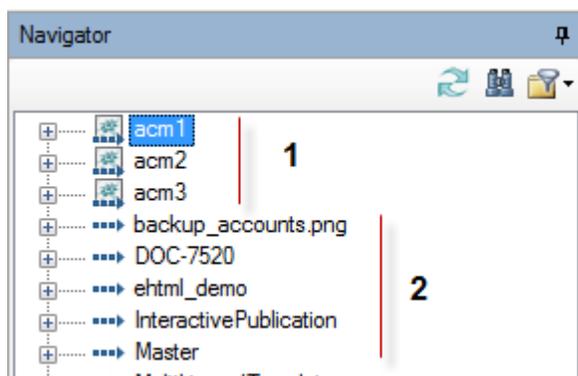
A Designer user making changes to assets managed by a review and approval workflow will:

- View managed projects as branched versions of their designated Master project.
- Update and submit the managed asset(s) for review.
- Receive review comments indicating changes to be made to the asset(s) following a review.
- Merge the changes to the asset(s) onto the Master project when all review processes have been approved.

Working with managed projects and publications in Designer

Managed projects and their associated assets are presented in Designer's navigator alongside projects that are not under review and approval control. Icons are used to distinguish these different types of projects and publications.

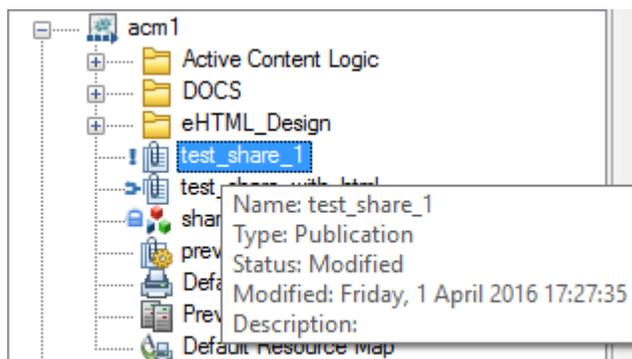
Projects presentation in the navigator



- 1 - Managed projects.
- 2 - Projects not under review and approval control.

Managed publications presented in the navigator

Icons and tooltip information are used to indicate the status of publications in the review/approval lifecycle.



Note: the **Modified** information is not presented in the tooltip when the publication is in an **Unmodified** state.

Table 5: Publication and Active Content status icons

Icon	Status	Description
	Unmodified	Status assigned to publication prior to the first edit.
	Modified	After edit of template or dependency* and prior to submission of publication.
	Submitted	After submission and prior to review.
	Approved	After successful review - indicated when the Project Reviewer portal user has approved the publication.
	Rejected	After unsuccessful review - indicated when the Project Reviewer portal user has rejected the publication.
	Submit modified	Template or dependency was changed after being submitted for review. After review is complete, status will change back to 'Modified'.
	Merged	After merge - prevents any further editing.
	Synchronization error	Displayed next to the project if any of its templates have been submitted for review but Designer could not contact the server to determine the exact state (approved or rejected).
	Read Only	Displayed next to the Active Content when it is being used by a template that has been brought into the project to be managed.

Note: A **Modified** status will be applied to your publication when changes are made to any of the publication's dependencies such as, sample data, interactive data, images, etc.

Submitting assets for review

Use the **Submit for Review** option when you are satisfied with your changes made to the asset you are working on. Your asset will then be sent to a designated reviewer or a set of reviewers in the EngageOne portal.

To submit asset changes for review:

1. Select the asset to be sent for review from your navigator pane.
2. From the **File** menu click **Submit for Review** or, right-click on the selected asset, and select **Review/Submit for Review**

Note:

- The **Submit for Review** option is only enabled when the asset is in a **Modified** or **Rejected** state.
- For existing assets, two PDF preview files will be created and made available to the EngageOne Interactive reviewer as follows:
 - A reference PDF representing the initial version of the publication.
 - A candidate PDF representing the current version of the publication.
- For new assets created by the project manager in the EngageOne portal, a single candidate PDF is created for review.
- The Submit for Review confirmation message may indicate there was a problem creating the reference PDF representing the initial version of the publication, even though the review has been submitted. Click on **Details** for information as to why the reference PDF was not created.

Review comments

Review comments are available when a Publication/Active Content has been reviewed by the appropriate reviewer in EngageOne portal.

To view comments:

1. Select the required Publication/Active Content from your navigator pane.
2. From the **File** menu click **View Comments** or, right-click on the selected publication/active content, and then click the **Review/View Comments**

Note:

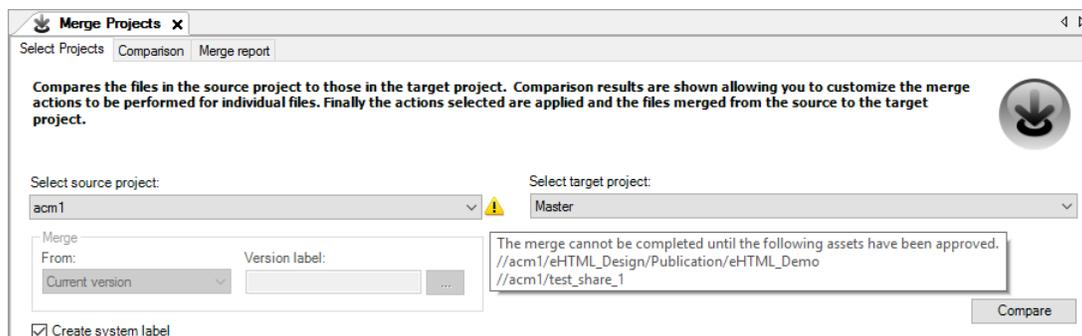
- The **View Comments** option is not available when the asset is in a **Unmodified** state.
- Review comments must be sent by the reviewer when a asset has been rejected and are optional on approval.
- You can, if required copy the comment text to your Clipboard for use in another application.

Merging managed projects

Assets and other managed project assets are merged in the usual way.

Note:

- Items can only be merged back to the designated master project. When a managed source project or item is selected, the target project is automatically set to **Master** and cannot be changed to another project.
- Only the latest version can be merged. When a managed source project or item is selected, the **Merge from** is not available.
- The merge operation can only be completed if all managed items are approved or, have already been merged.
- You will be presented with a warning icon if the managed project contains unapproved items. Hover over the warning icon to display a tooltip which lists the items that require approval. Note that you are not prevented from continuing at this point as you may wish to use the **Compare** option.



30 - Controlling the output environment

This section discusses features that allow you to issue specific commands to the intended output device within the datastream created by Generate.

These options are only used where dynamic changes to the output environment are required. General settings for the intended printer/browser are included in an output device object that is specified as part of the **Publish** task.

In this section

Spot color printers.....	455
Controlling printer features.....	457
PostScript DSC.....	458
Adding output comments.....	458
PDF Annotation.....	459



Spot color printers

Full color printers are able to mix toners to allow a full range of colors to be printed. However some printers only support the use of black plus one or more other single colors from fixed toner supplies. This is known as spot or highlight color.

When selecting colors in Designer for use with spot color printers you will need to be aware how your printer interprets color commands that are passed to it within an output datastream. If the printer supports only a single spot color this will typically be used whenever a color other than black is referenced within the datastream. Printers that support multiple spot colors will typically attempt to map colors within the datastream to the best match in their toner palette.

Within the Designer the use of spot color commands in the output datastream is specified by using a highlight color group in application logic. All presentation objects within such a group (other than charts and images) will be assigned the defined color. Existing color specifications in these objects will be overridden.

Objects not within a highlight color group will use their regular color settings but will be assumed to be black in a spot color environment.

The initial color to be used is specified as part of the highlight color object. If your intended printer supports multiple spot colors you can use an Action to change the active color. The selected color stays in force for objects within highlight color groups until a further change is requested.

If you intend to publish to an AFPDS printer that supports highlight color then you will need to specify the Highlight Color Space (HCS) options. Refer to the IBM MO:DCA specification for full details of the settings.

To create a highlight color group:

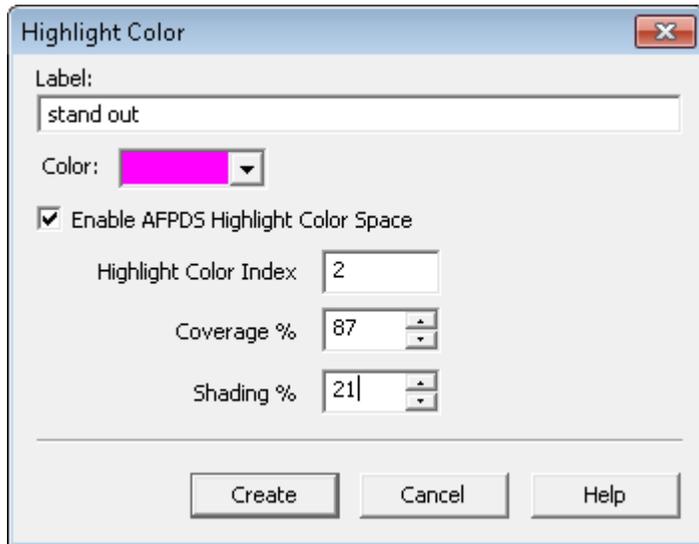
1. Mark the insertion point in the logic map.

Bear in mind that highlight color is a group object that contains the presentation objects to be colored.

2. On the menu click **Insert/Tool Objects/Highlight Color**.
3. In the **Highlight Color** dialog box select the spot color to be applied to the grouped objects.

A 16 color palette is used to represent the typical options of a spot color environment. For other requirements use the **Other Colors** option.

4. Enable AFPDS Highlight color printing if required.



Highlight color index is – either an index to a Color Mapping Table (CMT) when **coverage** and **shading** are both set to zero – or is a number associated with the printer’s highlight color.

Coverage defines the amount of coverage or density of ink that is used to render the output. The default value is 100%.

Shading defines the mixing of the default color (usually black) and the highlight color to render a mix of the two. The default value is 0%.

When AFPDS Highlight color is enabled this color is used only in the Designer

5. Add objects to the group.

Create logic map objects in the normal way. If the objects to be colored already exist you can drag them into the group.

To change the active spot color: create an action of type **Change Highlight Color** and specify the new color in the **Action** dialog box. This action is only available within a highlight color group.

Controlling printer features

Some printers use multiple paper trays to allow dynamic selection of the paper type to be used or multiple output bins to collate the pages generated into a custom sequence. Additionally, some printers also allow printing on both sides of the physical sheets of paper. This is known as duplex printing. When it is decided to print single-sided only, as simplex. It is possible to switch dynamically between simplex, duplex and tumble modes during the processing of a job but note that this is only supported where duplex or duplex tumble mode has been defined in the output device used when publishing the publication – see [Output, media and resources](#) on page 520 for more information.

Some printers also provide tumble printing – i.e. the back of each page is upside down compared to the front of the page. The top of one side of the sheet is at the same edge as the bottom of the other side.

Where these features are supported, Designer includes instructions for controlling the printer into the output datastream being generated.

The initial settings for these features are specified as part of the output device options that are used when publishing.

To switch the settings during the processing of a job you need to include action objects in the appropriate part of your document logic.

When creating an action to switch trays or bins you need to be aware how they are referenced on your target printer. Designer provides a set list of trays and bins that you can map using the appropriate format for the target printer. The mappings for the trays/bins are set in the Output Devices items in the navigator. You need to be aware of the mappings to select the correct tray/bin when you insert an Action object. While you can use an action to override which tray/bin is used when the document is published, you cannot override the tray/bin mappings created for the output device.

To create a printer control action:

1. Mark the insertion point in the logic map.

The change is made in relation to the page being generated when the action is called.

2. From the **Insert** menu select **Control Objects/Action**.
3. In the **Action** dialog box choose a type of:

- **Select input tray**
- **Select output bin**
- or **Simplex/Duplex/Tumble**

Points to note:

- Actions that change to **Duplex** or **Tumble** mode will only be applied to the output datastream when supported by the output device.

- Selecting the **Tumble** option for the following deprecated output devices results in:
 - **All page Tumble** printing for IJPDS
 - **Duplex** printing for MIBF
4. Specify when the change is actioned.
- By default, the change will take place immediately – i.e. the instruction to change the tray or bin will be included as part of the page currently being generated by the job. If you want to delay the change until the next new page select the **Active on next page** option.

PostScript DSC

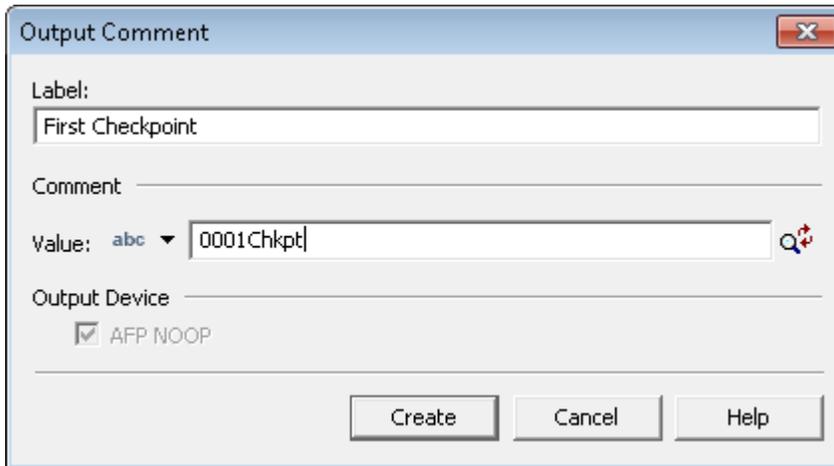
DSC comments are typically used to communicate with document manager systems in a PostScript environment. PostScript output has specific sections that provide controls related to pages, documents or the entire job. You will need to indicate what section your DSC comment is to appear in. The comment generated is not validated by Generate and the user must ensure that it conforms with the DSC guidelines published with your document manager. Invalid comments may result in unprintable PostScript files.

To program a DSC comment:

1. Mark the insertion point as required in the logic map and from the menu select **Insert/Control Objects/Action**.
2. In the **Action** dialog box choose a type of **PostScript DSC**.
3. Use the **DSC comment positioning** options to indicate the PostScript section to which the comment will be applied and use **Value** to specify the comment.

Adding output comments

Output comments are typically used as markers to control any post processing you may want to apply to your output datastream. When working with AFP output, these comments are inserted as No Operation (NOOP) instructions in the generated output. Comments can be applied at both publication and document levels. Note that output comment entries are ignored when generating output other than AFP.



1. In the logic map mark an insertion point where you want the output comment to appear.
2. Click the **Output Comments** toolbar button– or –from the **Insert** menu choose **Tool Objects** and click on the **Output Comment** option.
3. Enter details about the comment in the Output Comments dialog box as shown

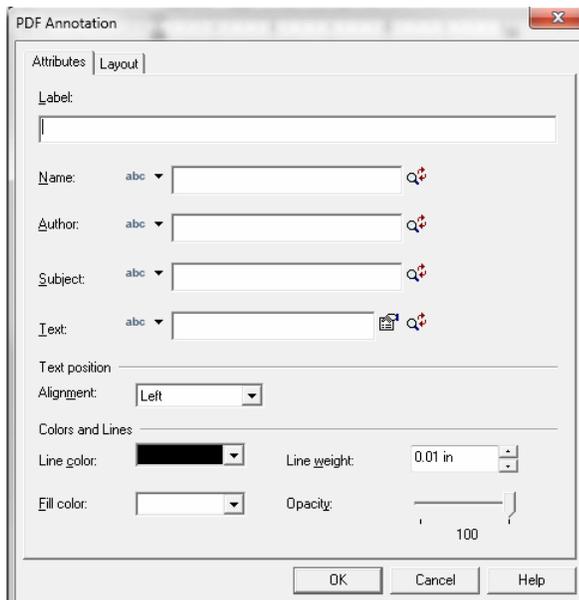
PDF Annotation

You can annotate a generated PDF and add notes and comments to it using PDF Annotations. PDF Annotations are objects designed to add additional content to PDF documents. Annotations are placed on a separate layer on a PDF page, above the normal content and do not interact with it. Support is limited to simple text annotations, used to mark a rectangular area of a PDF document page for later review or replacement in a post process.

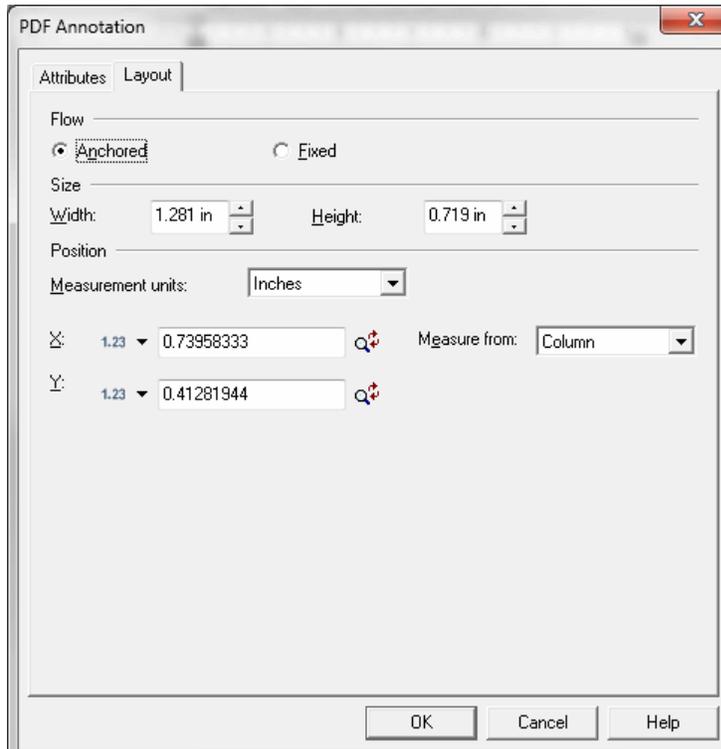
PDF Annotations are rectangular shapes and are treated like graphics for the purposes of positioning and customizing.

To create a PDF Annotation: Click in the paragraph where you want the PDF Annotation anchored and from the Insert/Presentation Objects menu select PDF Annotation– or –use the PDF Annotation button in the Objects toolbar. Create the box. If you are using a paragraph or fixed anchor click in the page layout where you want to start drawing and drag out the PDF Annotation to the required size.

To change PDF annotation attributes:



1. Right-click on the PDF Annotation object in the logic map and select Properties. The PDF Annotation dialog appears.
2. Use the Attributes tab to set values for Label, Name, Author, Subject and Text for the annotation. You can also set the required color, shading, alignment and line settings for the annotation. Fields to set color, shading, alignment, etc. behave in the same way as for other shape objects like text boxes, explained in the previous section. Additionally, there are five annotation related fields:
 - **Label** - The annotation label, as represented in the logic map.
 - **Name** - The annotation name, a text string uniquely identifying it among all the annotations on a page.
 - **Author** - Name of the author of the annotation.
 - **Subject** - Subject of the annotation.
 - **Text** - Text to be displayed for the annotation.
3. Click on the button adjacent to the magnifier button to modify field attributes such as font family, style, size, color, etc.
4. In the Name, Author, Subject and Text fields, you can either type in your inputs (with default data type as String) or map them with a data alias or a variable.
5. To map with a data alias or variable click the **abc** button adjacent to any of the fields and select Data Alias or Variable.
6. To change PDF Annotation layout use the **Layout** tab from the **PDF Annotation** dialog box to set the Flow (Anchored or Fixed), Size and Position of the PDF Annotation.



31 - Working with Designer input data

Input data is the variable information you use as part of your documents – typically an extract from your corporate databases. Designer supports input data in multiple formats.

In this section

Supported input data formats.....	463
About Designer input data	463



Supported input data formats

Keyed records – effectively a raw data extract where publication data is made up of one or more record types. Each record must be identifiable by the contents of a key field.

Delimited – a variation of keyed records where field locations are predefined using one or more consistent delimiter characters. This is sometimes known as CSV a (comma-separated values) file.

XML – any well formed XML file with an associated DTD or Schema. Note that XML support is an optional extra and is only enabled to users who install Designer with the appropriate license keycodes.

The type and structure of the input data file expected by a particular production job is described in a data format object. A data format is initially created by importing a sample of the expected input data which is then used as a template for identifying the expected records and fields. This is known as a sample data file.

Once you have imported your sample data into a new data format you can then use the DFE to specify and corroborate the details of the data elements it contains. The type of work you need to do in the DFE will depend on the type of input data you are working with.

About Designer input data

When you are designing documents you work with indirect references to fields and records. These references are stored as part of a data dictionary and you can use such dictionaries to represent the data available for a particular publication, document or any other logical grouping you want to establish.

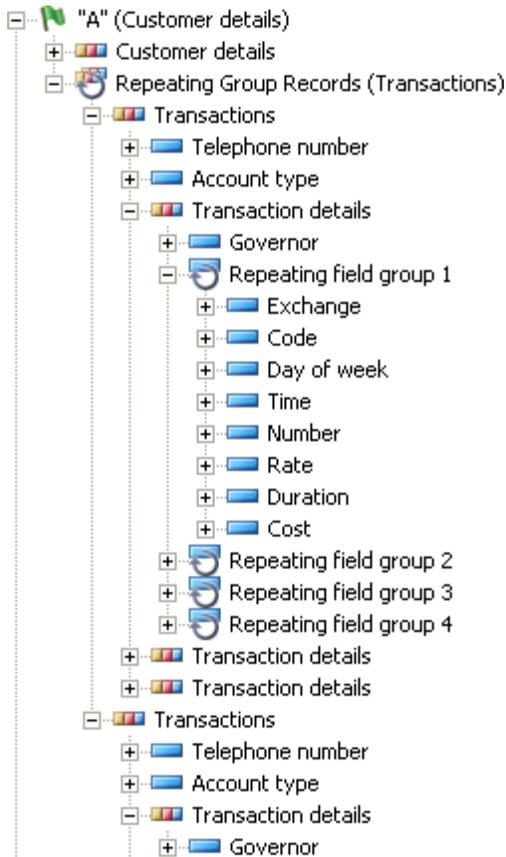
Before being published all the data aliases used by a publication must be linked with actual elements defined in the data format. This is done by creating a data map object. This can be done as part of the publication process or you can create one in advance.

Depending on your working practices you may want to create a data format first and then use it to export a data map and a data dictionary. This will allow document designers to work with a list of known data elements and avoids the need to manually link individual elements. If you are working with EngageOne, document input files are handled differently. For more details see [EngageOne data](#) on page 355.

Once you have associated a data map with a publication you will be able to use the sample data to provide extended example data for data aliases used in document designs. You will also be able to switch between sample data files providing they conform to the structure known to the data format referenced by the data map. You may want to do this for instance, if you may have a range of files that contain a variety of data samples.

Repeating data

Repeating data is the general term used for data elements or groups of elements that appear iteratively within the input file. These typically contain transactional information such as the details of an itemized phone bill or a list of purchases on a credit card.



If you are working with XMLinput, any repeating data structures will be distilled automatically from the sample data provided.

For keyed or delimited data you will need to specifically identify repeating data structures. Both records and fields can repeat both individually or as sequential groups of repeating elements.

See [Repeating records](#) on page 488 for more information.

Designer supports input data in the following formats:

Types of input data

Keyed records – effectively a raw data extract where publication data is made up of one or more record types. Each record must be identifiable by the contents of a key field.

Delimited – a variation of keyed records where field locations are predefined using one or more consistent delimiter characters. This is sometimes known as a CSV (comma-separated values) file.

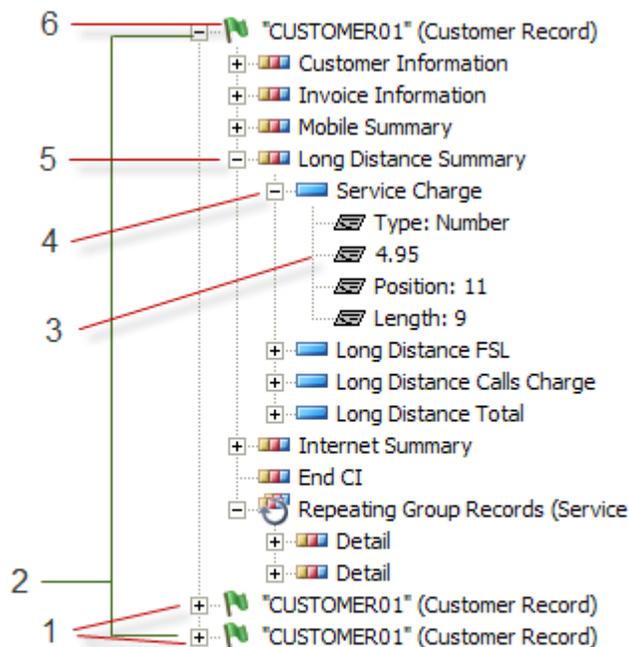
XML – any well formed XML file with an associated DTD or Schema. Note that XML support is an optional extra and is only enabled to users who install both the Designer and Generate with the appropriate license keycode.

Data structure

In the production environment Generate requires input data to be supplied in a single file for each job.

You may need to merge multiple input sources using the Data Flow product suite or similar.

Regardless of the type of input data being used Designer sees it as a grid of data elements consisting of one or more record types which themselves consist of one or more fields. A field is the data element that provides an actual value to Designer.



Legend:

1. Start of publication
2. Publication data set
3. value
4. field
5. record
6. Start of publication

The data intended for an individual publication must be provided in one or more sequential records within the input data file. This is known as a publication data set. It follows that the number of publications produced by a job depends on the number of publication data sets in the input data file.

The start of a new publication data set within the input data is indicated by the presence of a specific data element known as a start of publication record. Such records tell Designer to expect a new publication data set and to reload the appropriate logic. At least one element within a data format must be assigned this flag.

A job can produce multiple types of publication. Generate decides which design to apply to each publication depending on the start of publication record it encounters.

NOTE: See [Working with values](#) on page 170 for details of storing and passing values.

Each publication data set is discreet – i.e. you cannot directly share data between publications. However, document designs can include variables of job scope to pass values if required.

Keyed records

This type of input is effectively raw data although it must have a recognizable record structure. Each record type must be uniquely identifiable by the contents of a key field.

Keyed records

The key field can appear anywhere within a record but must be in the same position and have the same length in every record type. It typically contains a string that acts as a code for the relevant record type.

Various methods are supported for delineating the records within the input data file.

Any part of the remaining data within individual records can be defined as a field which is the data element normally referenced within publication designs.

All records and fields to be treated as repeating data must be sequential. Data intended as repeating fields must either always appear a fixed number of times or have a separate field within the same record that specifies the number of times the element/group appears (the 'governor field').

Keyed data record delineation methods

Record descriptor word (RDW) The RDW is a two-byte prefix that indicates the length of logical records within a stream. The pair can be either Most Significant Byte (MSB) or Least Significant Byte (LSB) ordered. MSB is commonly used on IBM operating systems. LSB is typical elsewhere.

For RDW input Designer can also allow for an 'eye-catcher' string that is sometimes used for visual identification of records in a binary stream.

Note that you must not use the same key for records in different parts of the data structure even if they are otherwise identical. The key field must be unique in each instance.

Byte measured – you can use this option to mark up single byte input data, for example, Latin-based text where each character is represented by a single byte.

Character measured – this option is necessary when marking up double-byte based text such as, Chinese, Japanese and Korean data.

XML

An XML file used as input to Designer must have a DTD or schema associated with it either within the XML or in an independent file.

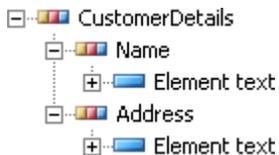
XML support is an optional extra and is only enabled to users who install both the Designer and Generate with the appropriate license keycodes.

By default, Designer maps XML data items to its internal data structure by converting elements to records and attributes to fields. You can choose to treat nested XML records as either individual child records or as fields within their parent record.

XML element structure

```
<Customer Details>
  <Name>TomGradin</Name>
  <Address>22 Main St.</Address>
</Customer Details>
```

Default handling of XML



Overriding default XML handling



Any 'well-formed' XML file should be acceptable as Designer input.

Limitations and restrictions when working with XML data

The following features are not currently supported:

- Model XML groups
- **XSL** transformations
- Boolean data type
- Abstract complexTypes
- The **<xs:choice>** element
- The **<xs:any>** element

Note: Elements with no defined type or defined as **xs:anyType** are not supported and may result in unexpected behavior. The **anyType** element is treated as a string by default when imported into Designer, you are allowed to customize the element definition further, but this means that it is given a fixed definition rather than **anyType** which is variable.

- Element substitution and blocking element substitutions

- The **Id** attribute used to identify a particular <xs:sequence> construct
- <xs:extension> with complex types.

Limitations:

- XML data files must always have a file extension of XML.
- Like-named elements in multiple schemas will be ignored.

xsi:type and <xs:extension> restrictions:

- A complexType must have an explicit element declaration.
- Element names greater than 253 characters in length will be undefined when imported.
- Do not use **xsi_type** as an attribute name as it can conflict with internally generated attribute names.
- Do not name elements as a combination of the base element and extension name separated by an underscore. Conflicts with internally generated element names.

Working with sample data

Normally you will want to import your sample data directly into a data format. See “[Creating a data format](#) on page 472” for details. This will result in the file being added to the appropriate project or folder in the navigator.

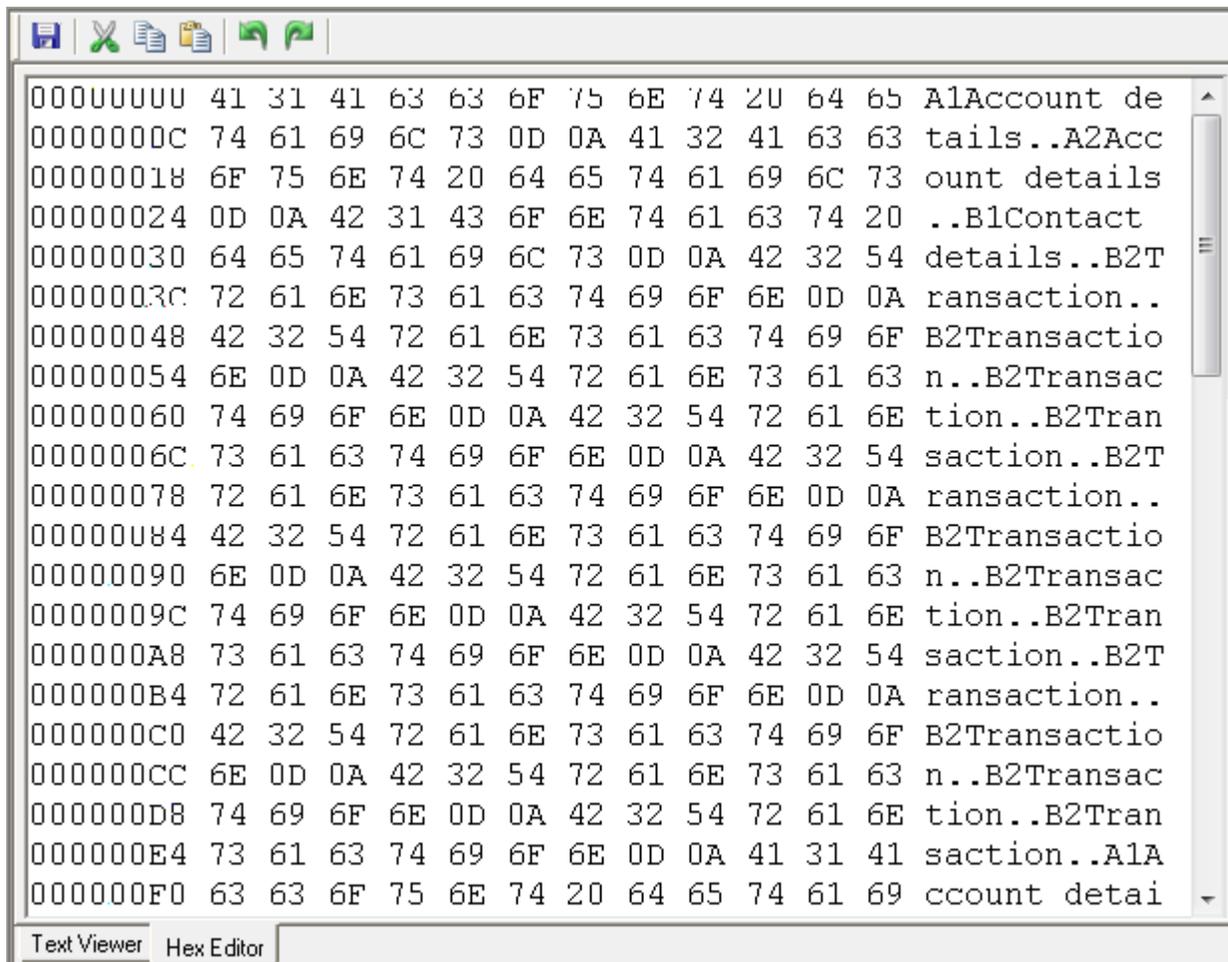
You can include additional sample data files in the repository as required and swap between files of the same type when working with the editors.

The navigator allows you to browse the content of sample data files within the repository and to make minor adjustments to them if required.

Note that care is required if you make structural changes to a sample data file it may no longer be suitable for the intended data format.

You can also export sample data files from the Designer. This enables you to take a sample data file from the repository and save it to a folder on your machine.

To browse or edit a sample data file: in the navigator browse to and open the required sample data file. This will be opened in the Sample Data Editor. Use the **Text Viewer** tab to browse or switch to the **Hex Editor** page to make adjustments as required.



The Hex editor presents the following information:

- save current changes
- cut, copy & paste data
- undo/redo changes
- The Toolbar is dock-able and allows quick access to editing functions.

and presents the following data in the following formats:

- byte offsets
- hex values
- text values

You can make changes to the hex or text values.

To export a sample data file:

1. In the navigator browse to and select the file you want to extract and select **Tasks/Export/Sample Data**.
2. Browse to the folder where you want to export the sample data to and click **OK**.
3. Click **OK** in the **Extraction Log** dialog box and the procedure is complete.

32 - About data formats

A data format is the Designer file that describes the type and structure of the input data file intended to be used with a particular job.

In this section

Creating a data format.....	472
-----------------------------	-----



Creating a data format

To create a data format you must import a sample of the data file that will be used in the production environment. This acts as the template on which the data format is built. The sample data file used for this purpose must contain examples of every type of data element that need to be referenced by the jobs that will use it.

The import process will automatically ascertain as much information as it can about the record and field structure of your input data. Once it is complete you can use the Data Format Editor to specify additional details about the structure. The amount of manual definition required will depend on the type of input data being used.

To import sample data into a new data format:

1. In the navigator browse to the required project or folder and on the **Tasks** menu select **Import/Sample Data**. Browse for the file to be imported and click **Open**.
2. In the **Sample Data File Properties** dialog box select the type of data being used and any related options.
3. Click **OK**. The Data Format Editor is invoked with the sample data file loaded ready for you to add specific definitions for records and fields within the structure. See “About the Data Format Editor” for details.

Importing a legacy data format file

You can import a legacy data format from DOC1 **Suite 4** into DOC1**Series 5**. When used with a sample data file imported into the repository, the data format will be predefined in accordance to the definitions set out in the DFE in DOC1 **Suite 4**. After it has been imported it can then be manipulated like a data format created in the Designer.

1. In the navigator browse to the required project or folder and on the **Tasks** menu select **Import/Legacy Data Format**. Browse to the required .ldf file and click **Open**.
2. As the data format is loading in the Data Format Editor another dialog box is displayed. Select the corresponding sample data file for the data format and click **Open**.
3. Select the **Define/Sample Data File Properties** menu option to set the default settings for the sample data and click **OK**. Proceed to define the data as described in [Working with keyed record and delimited data](#) on page 478.

If you are going to import a legacy data format containing EBCDIC characters you must ensure that you have selected **View/Undefined Data/As EBCDIC** before saving the ldf in the DFE in DOC1 **Suite 4**.

Configuring date options

Where your data contains date fields you may need to define how should handle incomplete or invalid values. In particular, you may want to customize how two digit year values are interpreted.

To customize date options: click the **Date Options** button on the Sample Data File Properties dialog box.

Date options - two digit year conversion boundary

Date fields that have the year component stored as only two digits are always converted to a full four digit number when used in Designer. Use this field to define the split between the 20th and 21st centuries for this conversion. The year is calculated to be in the range 19bv to 20(bv-1), where bv is the boundary value.

For example:

- A value of 75 gives a range 1975 – 2074.
- A value of 0 gives a range is 1900 – 1999.
- A value of 100 gives a range is 2000 – 2099.

Import Sample Data File options

Import Sample Data File options

Data Type	Indicate the format of the sample data file being imported from the list of supported types. Refer to Working with input data on page 117 for detailed information.
Delineation	For keyed record or delimited data, use this option to indicate how the records within the file are delineated. See Working with keyed record and delimited data on page 478 for details of supported data types and delineation methods.
Delimiter	For delimited data only. This option indicates the character(s) that are used to separate fields. Include spaces if these form part of the delimiter itself.
Quote	For delimited data use this option to indicate the quote mark character(s) used within the data. This will be used to identify the start and end of quoted string within the data. You may want to use quoted strings if your delimiter character(s) are also likely to appear as actual field data.

Import Sample Data File options

First record has column headings

For delimited data only. When selected the first record in the sample data is considered to be a column heading. Its values will be used to provide a default name for subsequent records and fields in the data format. The first record is ignored when the data is processed. Note that all instances of the 'header' record (i.e. with the same key) will also be ignored. It is important to note that the Column headings record must have a different record identifier to the other records.

RDW includes prefix length

For keyed record data using RDW delineation. Check this option if the record length indicated in the RDW includes the two bytes of the RDW itself. If this is not checked the length of the RDW is assumed to be excluded from this value.

Eye catcher size

For keyed record data using RDW delineation. An eye catcher is a string that is sometimes used as an additional prefix to allow visual identification of individual records in a binary stream. If your data contains eye catchers, use this option to indicate the number of bytes they occupy in each record (this position is then ignored by Generate when processing the input data).

Default encoding scheme

All data needs to be interpreted using a code page. System code pages included with Designer should be sufficient for most users, however, if you have imported a Custom code page that option will be enabled. This allows applications to understand the intended strings and values within the binary encoding that makes up the data itself. Use this option to identify the code page that will, by default, be assumed for interpreting the input data being imported. Note that parts of your data may use different encoding schemes (in particular, where text is based on multiple languages that use different code pages) so Designer allows you to override this default for each individual field if required.

Treat <xs:extension> as Record

This option is available for use with XML data. Records are created according to the <xs:extension> definitions in the schema. Refer to [Using xsi:type attributes](#) for further information.

Treat <xs:extension> as Union

This option is available for use with XML data. Records are combined, merging all the attributes of the elements that extend the base element. This is the default setting. Refer to [Using xsi:type attributes](#).

Create <xsi:type> field

Used with either of the above XML options for identifying different record types. An additional field called xsi:type is included in every record that has been extended.

Import Sample Data File options

Default locale	This option allows you to specify a default locale to be used to determine regional delimiters within number and date fields. It can be overridden for individual field definitions if required.
Default date handling	See Configuring date options on page 473 .
Default string options	This option enables you to have any spaces to the left or right of a data alias removed when it is inserted into a document. This setting is applied to all string fields in the sample data, however, it can be overridden for specific fields using either the Define Field or the Format dialog box.
Unavailable field defaults	<p>When an empty data field is encountered in a publication at production time this option enables you to choose how to handle it. You can select to Abort or Replace with constant value. This setting is applied to all fields in the sample data. However, it can be overridden for specific fields using the Define Field dialog box or by using the <code>IsSet</code> and <code>IsnotSet</code> operators in a condition. For more details, see <code>Concept_Appendix_B.dita</code>.</p> <p>Note that if Unavailable field defaults have not been set then the following default values are used:</p> <ul style="list-style-type: none"> • Date: 01/01/1970 • Time: 00:00:00 • Number/Decimal: 0 • String: Empty

Configuring date options

Where your data contains date fields you may need to define how Designer should handle incomplete or invalid values. In particular, you may want to customize how two digit year values are interpreted.

To customize date options:

click the **Date Options** button on the **Sample Data File Properties** dialog box.

Date options

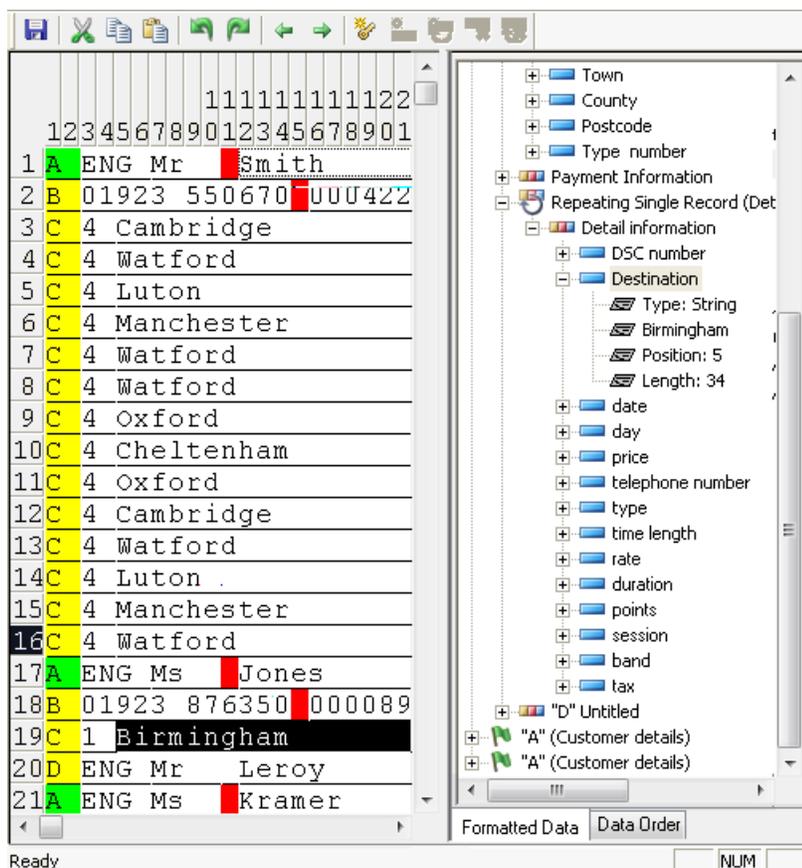
Two digit year conversion boundary

Date fields that have the year component stored as only two digits are always converted to a full four digit number when used in Designer. Use this field to define the split between the 20th and 21st centuries for this conversion. The year is calculated to be in the range 19bv to 20(bv-1), where bv is the boundary value. For example: A value of 75 gives a range 1975 – 2074. A value of 0 gives a range is 1900 – 1999. A value of 100 gives a range is 2000 – 2099.

About the Data Format Editor

The Data Format Editor DFE window always has two main panes. The left hand pane gives you a view on the sample data that was used to create the data format (or the latest version of the sample data to be loaded into the editor). The right hand pane shows you how the data will be interpreted—i.e. the structure of records and fields into which the data is placed – when it is used with a production job.

The actual views available in these two panes will differ depending on what type of input data you are working with. The methods used to carry out the additional definition work required once the sample data has been imported will also vary. See [Working with keyed record and delimited data](#) on page 478 and [Working with XML](#) on page 490 for details.



Data format changes

Some data format attributes can be changed in the DFE. You can find and replace the descriptive text for data fields and records, change the type of encoding for the data format, change the field terminator, override the default string options and change the empty field handling.

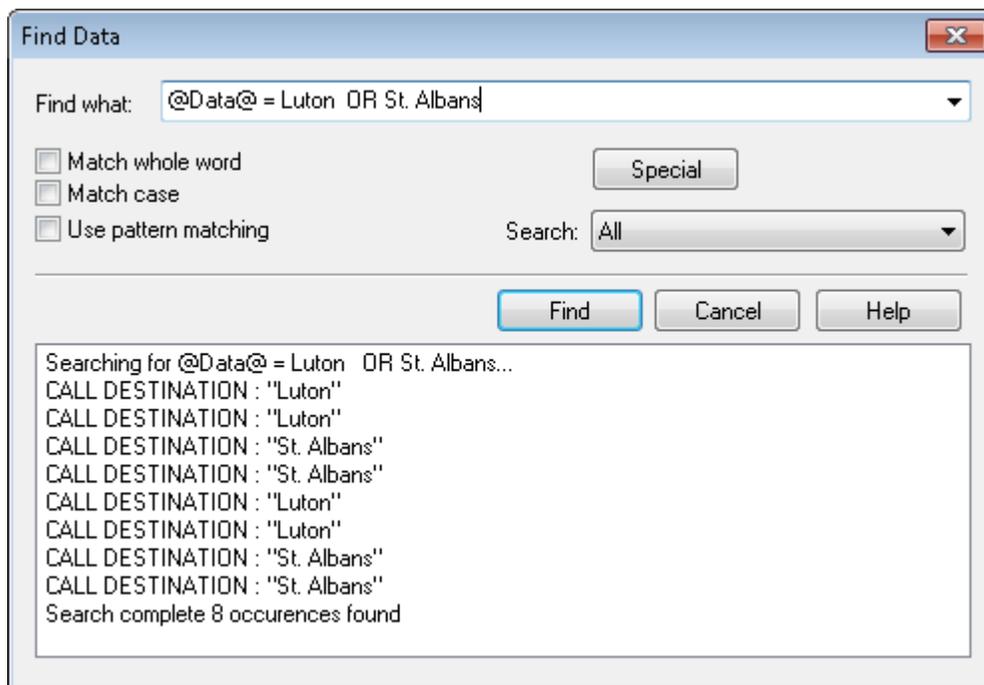
Note that any changes made to the data format in the Replace dialog box will be global

To make data format changes: select **Edit/Replace** from the menu and the Replace dialog box is displayed. Select the required option and make the necessary changes, then click **Replace All**.

Searching a data format

The find data feature enables you to search the data format for specific data records and fields. Using the options available in the Find Data dialog box enables you to narrow your search based on the type of data, as well as the labels of the records and fields. You can also use AND/OR to create more complicated search queries.

To search a data format: select **Edit/Find** from the menu and the Find Data dialog box is displayed. Choose the necessary options for the data you want to find and click **Find**.



- The Special button enables you to create AND/OR queries, as well as search specifically for labels or data
- The Search list includes the available field types for the data format

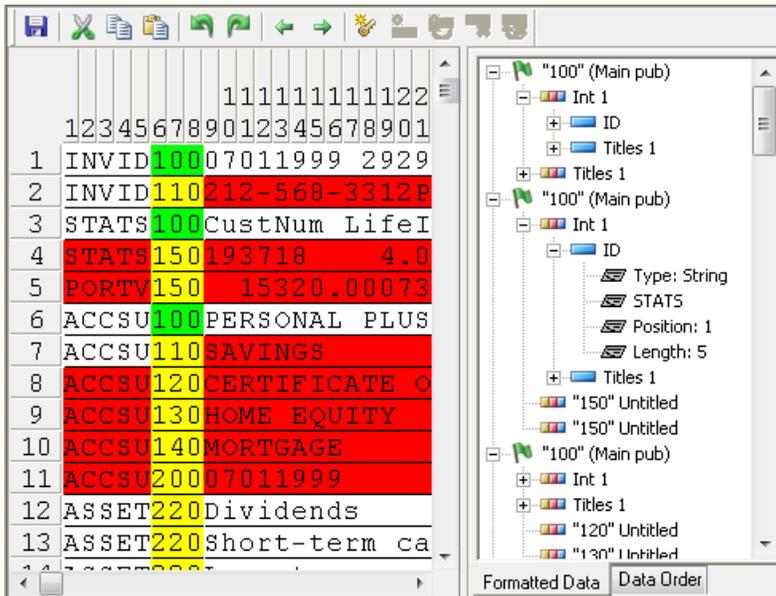
Data format options

The colors and fonts that are used in the DFE can be customized. In the Data view you can select from any of the fonts in your system and use the color picker to change the way its displayed.

To modify the Data view settings: in the Data Format Editor select Tools/Options from the menu to display the Options dialog box. Select the Grid or Data tab to make the required changes.

Working with keyed record and delimited data

With keyed record or delimited data loaded, the data view shows you the content of you sample data in a grid format. This is used to mark specific parts of the data when defining the extent of elements. The structure view shows you the records and fields you have defined and allows you to view and control their properties.



- You can customize the colors and font used in this view by selecting Tools/Options from the menu.

These data types essentially require you to use the same features in the DFE although with delimited data much of the information about the structure of the data can be assumed.

For keyed record data you will need to:

- identify the key field
- define the location of all individual fields that need to be referenced by a production job
- define controls for fields that have varying size.

For both keyed record and delimited data you will need to:

- indicate the record types that mark the start of a publication
- mark records that must always appear in a publication data set as **mandatory**
- define a reference name and data type of all fields that need to be referenced by a production job
- identify the record and field types that will appear iteratively – i.e. repeating data.

Defining the key field

Individual record types within keyed record data are identified according to the contents of the key field. The contents of this field must uniquely identify the record type and must be the same in each record of the same type.

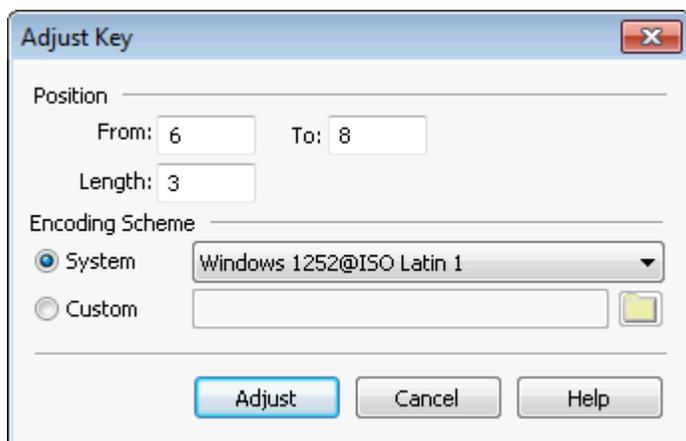
For delimited data the key field is automatically assumed to be the first field in each record and cannot be changed but it can have varying size.

For keyed record data the key field must appear at the same position within every record in the input data and can be 1-64 characters in length. The field length cannot vary in different records and its content must be padded with spaces if necessary.

When working with keyed record data no other data definitions can take place until the key field is defined.

To define or redefine the key field:

1. If required, select the columns to act as the key.
In the data view drag the mouse over the required positions in any record.
2. Select **Define/Key** from the menu or click the define key button on the toolbar and the **Define Key** dialog box is displayed.
3. Confirm or enter the key field offset and length and if necessary select a different encoding scheme to that chosen as the default scheme for the data format. Note that the option to change the encoding scheme is only available when working with keyed record (byte measured) input data.



4. Click **Define** to establish the key field. The appropriate columns are highlighted.

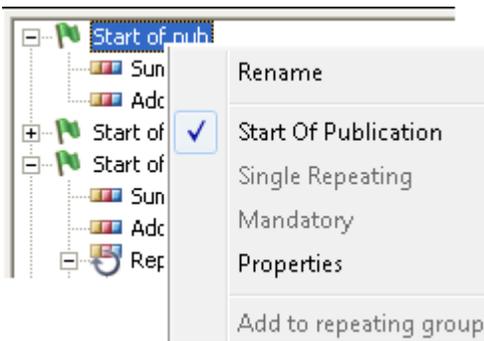
Working with records

Designer is automatically aware of the various record types within the sample data once the key field has been defined. Records assume a default label from the contents of the key field. You may often want to assign more meaningful names to the record types however.

Additionally, one or more records that indicate the start of a new publication data set will need to be identified.

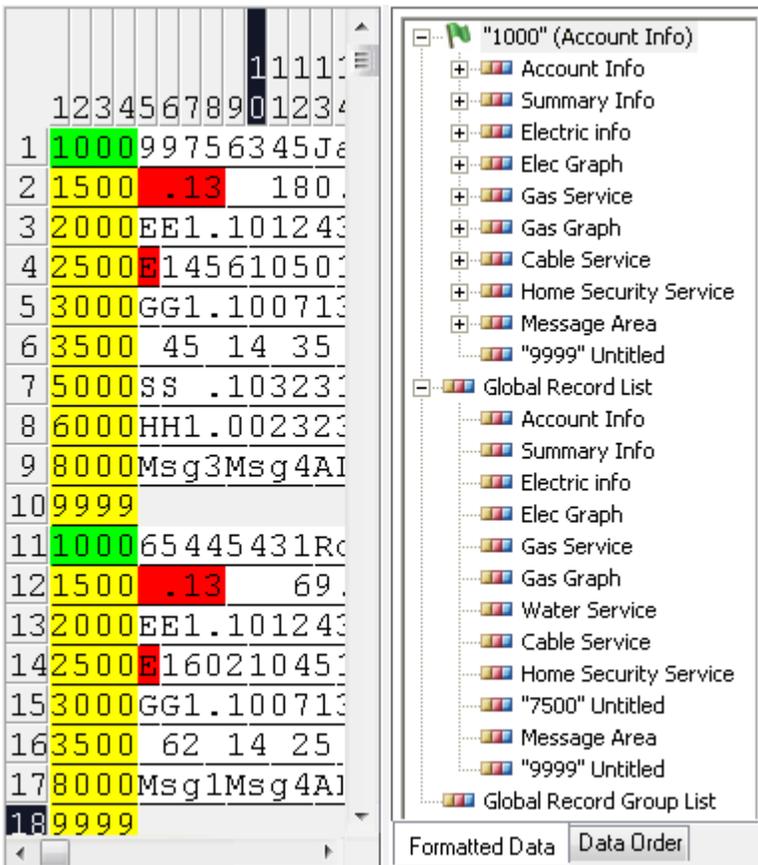
To define a start of publication record: right-click on it in the structure view and select **Start of Publication** from the context menu.

To rename a record: right-click on it in the structure view and select **Rename** from the context menu. Then overwrite the name within the view itself.



Managing records

The **Data Order** tab allows you to manage the record structure used in publications, for example, re-create the record order, exclude records or group them together.



Recreating record order

If necessary you can remove the data hierarchy of a publication data set that was previously defined. This allows you to redefine the data structure for a production job.

To recreate the record order: in the structure view select the **Data Order** tab. Right-click on the appropriate global record file and select **Recreate record order** from the context menu.

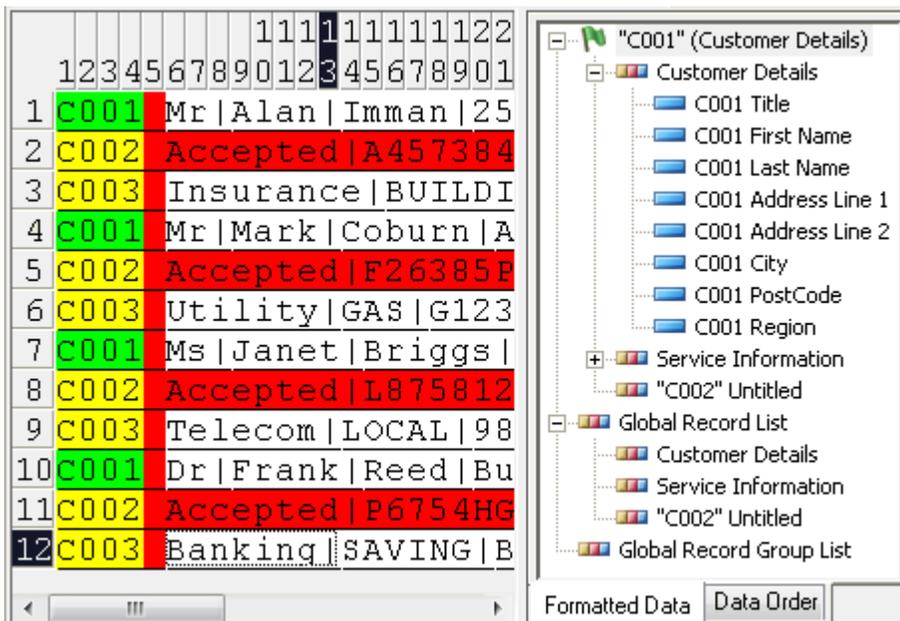
Use this option with care as it flattens the file and removes all previously defined hierarchy in the formatted data, such as repeating groups.

Excluding Records

You can **Remove** a record from the formatting but if the record is used in the current data it will be retained for future use.

The **Delete** option removes the record format in every data publication data set in which the record is used. Records that used it will be shown as unformatted.

- To delete or hide a record right-click the required record and select the appropriate option
- To remove, delete or hide a record click the Data Order tab.



If you **Hide** the record it will not be processed but the formatting remains. To reverse the hide status repeat the process and the tick is removed, reactivating the record.

After removing or deleting records the screen is refreshed and a new record format is created with the name “untitled” for each record in the current data that is unformatted.

Grouping records

If you want to use the same record at different locations within the data format, then you must use different key numbers for each record. However, you can group these records together, which will enable you to maintain them as one, i.e. changes made to any of the fields will be applied to all records in that group.

To group records together: in structure view select the **Data Order** tab. Highlight **Global Record Group List** and select **Create Record Group** from the context menu. In the **Record Group** dialog box, select the record from the record list that you want to use as the template, and **Add** it to the group. Repeat for all other records that you want in the group.

To modify a record group: in structure view select the **Data Order** tab and expand the **Global Record Group List**. Right-click the record group and select **Edit Record Group** to display the **Record Group** dialog box and make the required changes.

To delete a record group: in structure view select the **Data Order** tab and expand the **Global Record Group List**. Right-click the record and select **Delete Record Group**.

Mandatory records

By default, records other than those flagged as **Start of Publication** do not have to appear in every publication data set. Provided fields within the record are used conditionally within document designs they will simply be ignored when they are not present.

However, it may be important to your document design to make sure that certain records are always present before attempting to process a publication. Where this is the case you can flag such records as being **Mandatory**.

If a mandatory record does not appear in a publication data set where it was expected Generate will follow the on error with publication setting which is defined when publishing.

To define a record as being mandatory: right-click on it in the structure view and select **Mandatory** from the context menu.

Identifying and defining fields

You will need to define each field within the input data that you want to use as part of a document design.

- For keyed record data you will need to indicate the position and length of fields. (For delimited data this information can be assumed from the file format itself.)
- For both keyed record and delimited data you will need to give each field a reference name and specify the type of data it will provide.

Most keyed record fields will have a consistent offset and length within records. Variable length fields are supported providing they have a marker that indicates the end of the field contents. This marker must consist of a fixed length constant of one or more characters which are appended to the field contents proper whenever it appears.

To define a field:

1. If you are working with keyed record data first identify the field position. In the data view drag the mouse over the required positions in any occurrence of the record in which the field appears.
For delimited data or a field that has already been defined simply click within any occurrence of the appropriate field.
2. Select **Define/Field** from the menu or click the define field button on the toolbar.
3. In the **Define Field** dialog box you must specify the type of data the field contains and any related settings (see below).
4. Click **Define** to add the field definition to the data format.

Define Field dialog box

Define Field dialog box

Position	For keyed record data use these settings to adjust the field offset and length if required. Be aware of the effects changes might have on other field definitions if you make changes here.
Type	Identifies the type of data the field contains.
String	Can be either text data or RTF data with basic text formatting with a maximum size of 32k characters. With string types you can choose to override the default string options by selecting Trim Left and Trim Right .
Date	<p>A calendar date. If this type is selected you must use the Date Format Type field to identify the sequence and format of the date components. Note that any date separator characters within the input data are automatically allowed for and are not shown in the list. Month values provided as text are interpreted according to the settings in the default or field specific Locale instance. If the selected locale includes the full name of the first month of the year as 'January' the month component in the field must match this to be correctly interpreted. The meaning of the various date component options are:</p> <ul style="list-style-type: none"> • dd – day of the month • ddd – Julian day of the year (e.g. 032 = 1 February) • mm – month as a two-digit code (e.g. 01 = January) • MMM – month as a three character text abbreviation • MMMM – month as a full case sensitive string • yy – year as a two digit abbreviation • yyyy – year as a full four digit number • (P) – the date is stored as packed decimal. <p>Click Options if you want to override the default settings for interpreting date components for the field.</p> <p>See Working with keyed record and delimited data on page 478 for details of date interpretation settings. See Creating and customizing locales on page 311 for information about locales.</p>

Define Field dialog box

Number	A signed or unsigned number with an optional floating point. Values can have a maximum of 24 digits before the decimal place and 8 after. Where used, the decimal point is real – i.e. a byte containing a decimal point character is stored as part of the number. The character used as the decimal point is interpreted according to the settings in the default or field specific Locale .
Decimal	A signed or unsigned number with optional implied decimal places. The use of this option invokes the Precision entry field for the position of the decimal point to be defined. Maximum value is the same as type Number . Note that the binary and packed field options are not available when working with character measured keyed data.
Native Binary	A binary encoded number where the storage type (LSB or MSB as in the options above) is assumed from the operating system on which Generate processes a job. Decimal places are implied – use the Precision field as with type Decimal
LSB Binary	A binary encoded number where the most significant byte is the second in a two-byte pair (typical of applications from Windows and Unix systems). Decimal places are implied – use the Precision field as with type Decimal .
MSB Binary	A binary encoded number where the most significant byte is the first in a two-byte pair (typical of applications using IBM operating systems). Decimal places are implied – use the Precision field as with type Decimal .
Packed Decimal	A signed number represented by the hex values of the field bytes, two digits per byte. The second nibble of the final byte indicates the sign: hex 'D' indicates a negative value, all others a positive. Thus the hex string X'348961' represents the value +34896. Decimal places are implied – use the Precision field as with type Decimal .
Sign Overstruck	As for Packed Decimal except that the first nibble of the final byte indicates the sign.
Position relative to previous field	Fields that follow data structures of variable length within a record – i.e. fields that appear after variable size fields or repeating fields – should have this flag turned on. Under normal circumstances this option should be automatically checked where appropriate.

Define Field dialog box

Variable size field

Checking this option indicates the field can occupy a varying number of bytes. In order to use this format the field must always be terminated by the same delimiter characters. The length of the delimiter is specified as the **Byte count ending field** setting

Encoding scheme

This option allows you to specify the system code page to be used when interpreting the data within the field – i.e. the code page for which the field data was intended to be used with. A default code page is assigned when the data format is first created by importing sample data and you should not need to change this setting unless parts of your data are based on different encoding schemes (in particular, where text is based on multiple languages that use different code pages). **System** code pages included with Designer should be sufficient for most users. However if you have imported a **Custom** code page that option will be enabled. Note that the Encoding scheme options will be disabled for comma separated data (CSV).

It is important to note that if your input data is character measured, the option to define an encoding scheme at field level is not available. A single encoding scheme is set for the entire data format when working with this type of input data.

Locale

This option allows you to override the default locale specified for the data format and assign a specific locale to be used when interpreting the field. It is used to determine regional delimiters within number and date fields.

Unavailable field options

When an empty data field is encountered in a publication at production time this option enables you to choose how to handle it. You can select to **Abort** or **Replace with constant** value. This setting overrides the default settings for the data format. You can also use the **IsSet** and **IsnotSet** operators in a condition to set the handling options. For more details, see [Appendix B Formula expressions](#) on page 716.

Editing field definitions

When working with keyed record data you can copy or move the definition assigned to an existing field to create a new field definition at a different location.

You can redefine existing field definitions to change attributes or amend the defined position of the field. When moving field definitions, note that if the new location occupies any positions already

defined for other fields these will automatically be undefined. It is often cleaner to undefine the affected fields specifically before redefining their positions.

You may also want to shift a group of existing field definitions along the record offset range particularly if you are editing a data format to take account of additional data that has been added to a record.

To edit a field definition: double-click on it in either the data or structure view

– or –

with the field highlighted select **Define/Field** from the menu or click the toolbar button– or –right-click on it in the structure view and select **Properties** from the context menu.

To undefine a field definition: with the field highlighted select **Define/Remove Field** from the menu or click the toolbar button.

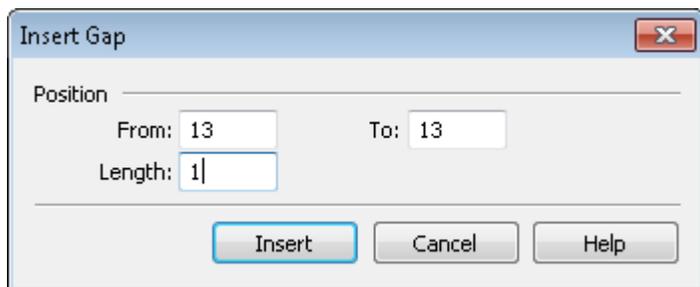
To copy or move a field definition:

1. Click on the field in either the data or structure view and select Copy or Cut from the Edit menu– or –right-click on the field and select the required option from the context menu.
2. Identify the start position for the new field by clicking in an undefined area in the data view.
3. Select **Edit/Paste** from the menu – or – right-click in the data view where the definition is to be inserted and **Paste** from the context menu.
4. Make any necessary adjustments in the **Define Field** dialog box and click Define to complete the new definition.

Shifting field definitions

You may also want to shift a group of existing field definitions along the record offset range particularly if you are editing a data format to take account of additional data that has been added to a record.

1. In the data view click on a field adjacent to where a gap is to be inserted or removed.
2. Select **Insert Gap** or **Close Gap** from the Define menu.
3. Use the dialog box to specify the positions to be inserted or removed from the record definition.



Defining repeating data structures

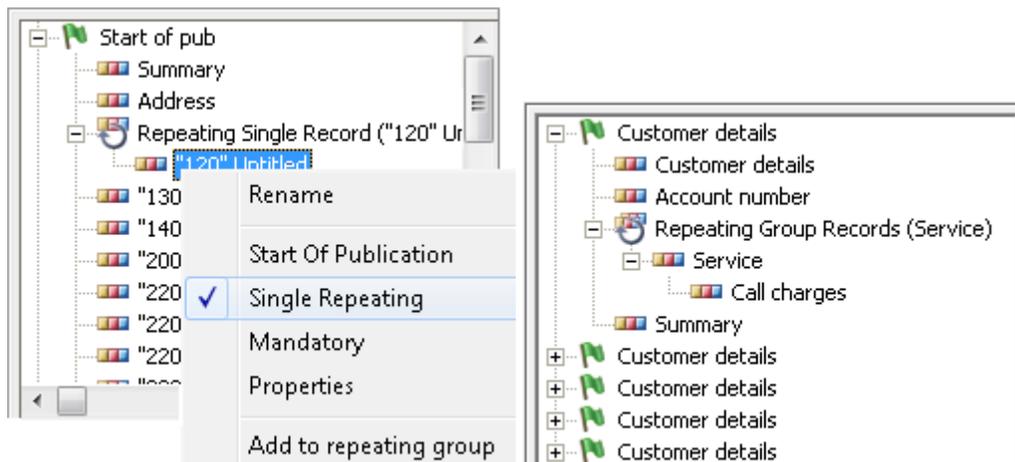
Repeating data are iterations of data elements of the same type within the input file. These typically contain transactional information such as the details of an itemized phone bill or a list of purchases on a credit card. For keyed record data, Designer supports two kinds of repeating data structure:

repeating records and repeating fields. Each type can either be an iteration of a single element or of a group of elements – e.g. a range of several elements which appears iteratively as a group. Groups of repeating elements must appear sequentially in the input data. It is possible to nest repeating records/groups and to have repeating fields/groups within repeating records. Nested repeating fields are not supported.

Repeating records

Repeating records are specified using properties of the record objects in the structure view.

To define or undefine a single repeating record: right-click on the appropriate record in the structure view and select **Single Repeating** from the short cut menu.



To define a repeating record group:

A repeating record group can consist of a mix of records that occur more than once, or a set of individual records that occur only once.

If you want to add records to the repeating group which occur more than once, right click on the first of these contiguous records and select the **Single Repeating** menu option.

IMPORTANT NOTE: Do **not** select this option for records you wish to add to the repeating group that occur only once.

1. Identify the parent record that will inherit the repeating group name.
2. Right-click on a subsequent record that is to be part of the group and select **Add to repeating group** from the context menu.
3. Repeat this for each of the records that make up the repeating group.

To remove a record from a repeating group: right-click on it in the structure view and select **Remove from repeating group** on the context menu.

Repeating fields

Unlike repeating records, fields cannot be iterated an unlimited number of times. Designer needs to be told how many times the field or group of fields are repeated each time it appears in the input data.

If the data is always repeated the same number of times you can specify the fixed count setting.

Alternatively, you can dynamically indicate the number of repeats by including a governor field within the input data. A Governor field must contain an integer value that indicates the number of times the group is repeated in the current record. It must appear in the same record and before the repeating fields to which it relates, but it need not be contiguous with the repeating data itself.

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1	KeyA																																																																																
2	KeyB	Rep1	Rep2	Rep3																																																																													
3	KeyC	RepA1	RepA2	RepA3	RepB1	RepB2	RepB3																																																																										
4	KeyD	03	Rep1	Rep2	Rep3																																																																												
5	KeyD	05	Rep1	Rep2	Rep3	Rep4	Rep5																																																																										
6	KeyE	03	RepA1	RepA2	RepB1	RepB2	RepC1	RepC2																																																																									
7	KeyE	02	RepA1	RepA2	RepB1	RepB2																																																																											

- KeyB: single field repeated a fixed number of times (3)
- KeyC: group of 3 fields repeated a fixed number of times (2)
- KeyD: single field with variable repeats specified in governor field (white)

To define a repeating field (single or group):

1. All the data elements to be included in the repeat must already be defined as individual fields. See [Working with keyed record and delimited data](#) on page 478 for details.
2. In the structure or data view click on the field that is to be the first (or only) in the repeating field group.

Note that all fields other than the first in the group must be defined as relatively positioned. If you did not specify this when defining the fields you will be asked to confirm that this setting should automatically be applied.

3. Select **Define/Repeating Field** from the menu– or –click the toolbar button.
4. In the **Define Repeating Field** dialog box adjust the value in Number of fields if more than one field is to be included in the group.
5. For a fixed number of repeats click the Fixed Count radio button and enter the number of times the group is repeated in the application data in the Number of repeats entry field.

For a variable number of repeats click the Governor Field radio button and select from the list of fields in the record (only those fields prior to the field being repeated are listed).



6. Select Define to establish the repeating field group.

Advanced publication properties

You can override the memory attributes for specific publication datasets within input data that will be processed by Generate. The memory settings for the input data are set for a production job in either the navigator or the Publication Wizard.

The following three options are available:

Use Host Settings – the memory settings used in the Production Job/Publish Wizard will be used by the publication dataset. This is the default setting.

Store All Publication Data – all the data for the publication dataset will be stored in memory until processing is complete. This should result in the fastest processing time.

Minimize Memory Requirements – Generate will hold the minimum amount of the publication dataset in memory. This will be either a single record or, if sorting of repeating data is required, the group of records.

To set the memory requirements for a publication data set: right-click the start of publication record and select **Properties** from the context menu. The Advanced Publication Properties dialog box is displayed. Select the required memory option and click **OK**.

Adjusting and undefining repeating fields

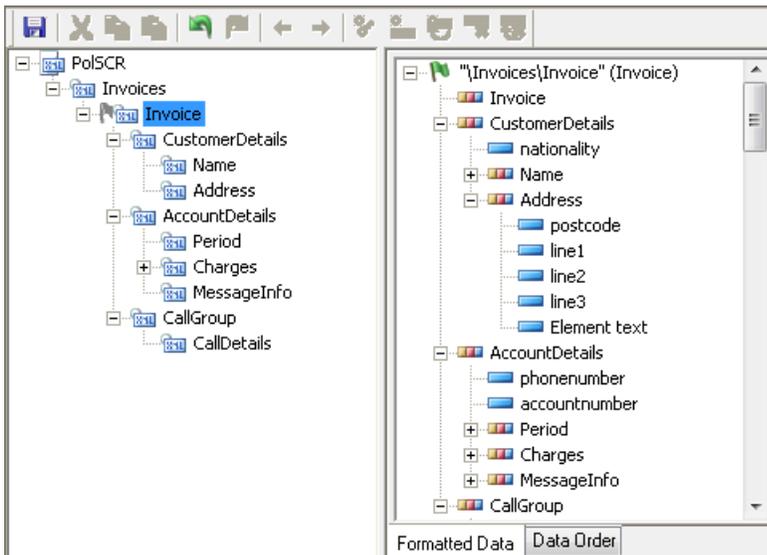
When undefining repeating fields it is important to be aware that you cannot undefine a repeating field group if it has existing field definitions following it in the record (these will normally be marked as relatively positioned and therefore rely on the repeating field for their offset information). Undefine the subsequent field definitions or unset their relative positioning flag before un-defining the repeating field.

1. Highlight the first field of the group.
2. From the **Define** menu select Remove Repeating Field from the menu– or –click the Remove Repeating Field toolbar button.
3. Recreate the repeating field with new settings if required.

Working with XML

XML files used as input to Designer must have an associated **document type definition** (DTD) or schema (XSD) either within the XML itself or in an independent file. Where these are stored externally, the sample data import task will attempt to find the DTD or XSL file at the location specified within the XML or at the same location as the XML file if no path is specified.

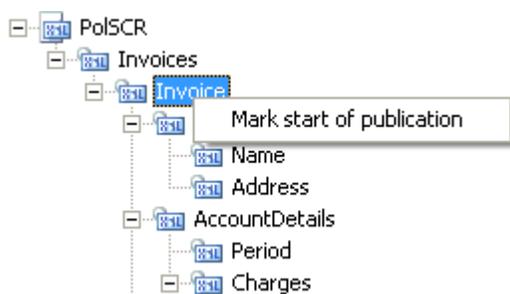
XML is self-describing so the sample data import task can automatically determine the data elements it contains including any repeating data structures. **“Task: Creating a data format”** describes how this task is performed. You will still need to use the DFE to indicate the record types that mark the start of publication data sets. In some circumstances you may also need to redefine some of the data types that have been assumed for individual fields.



With XML data loaded, the data view shows you the data elements defined in the schema. These equate to records in the Designer data model. One or more of these records need to be marked as **Start of Publication**.

See **About Designer input data** on page 463 for more information about the Designer data model and how publication data sets are identified.

Once this has been done Designer will evaluate all the elements within all the element declarations in the schema. In the structure view individual data fields will be created for each identifiable data element.



By default, when the schema is evaluated elements are automatically treated as records; their attributes and values are treated as fields of those records. You may choose to override this default setting and have child XML elements treated as fields within their parent record instead.

Only those data elements contained by the selected **Start of Publication** element will be available for use with a production job. In this example elements within 'AccountDetails' will not be available if 'Name' is selected as **Start of Publication**.

To define a start of publication record: right-click on it in the data view and select **Start of Publication** from the context menu.

To change the data type associated with an element: select the **Formatted Data** tab. Right-click the data alias you want to change and select **Properties** from the context menu. Choose the required data type from the **Type** list.

To override the default handling of XML elements/records: double-click the record in the Structured View and the **Advanced Record Properties** dialog box is displayed. Select the **Treat child elements as fields** option and click **Define**.

Sorting child elements

If you wish to present sorted XML elements in your output, you must ensure that the **Treat child elements as fields** option is set at the parent level. To access this option, right-click the parent record and select **Properties**, the **Advanced Record Properties** dialog is presented.



Using xsi:type attributes

The `xs:extension` construct allows you to take a base element structure, and add additional information. For example, generic address information could be enhanced by the addition of **State/ZIP code** values for **US** addresses, or **County/Postcode** values for **UK** addresses. To allow additional attributes in your data, you will need to include the appropriate `<xs:extension>` constructs in the schema to cater for the additional properties and provide `xsi:type` attributes in the data, for example:

```
<Address xsi:type="USAddress" ... \>
<Address xsi:type="UKAddress" ... \>
```

When importing your data you can create records based on the `xs:extension` definitions found in the schema by using the **Treat `<xs:extension>` as Record** option. Alternatively you can merge all attributes into a single record type Using the **Treat `<xs:extension>` as Union** option. The merge option allows you to make use of XML used in previous versions of Designer without the need to change your publication design. Note that for both import methods you can use the `Create <xsi:type>` field option to include an additional field in your record structure that contains the **`xsi:type`** value.

Treat `<xs:extension>` as Record import option



- `Address_UKAddress` and `Address_USAddress` records are created based on corresponding `<xs:extension>` constructs in the associated schema.

Treat `<xs:extension>` as Union import option



- When this option is used a single record type is created containing all possible attributes.

For a list of restrictions and limitations see [About input data](#) on page 116.

Exporting to other objects

If required you can export the definitions within a data format into the other main data related Designer objects, a data dictionary and a data map.

A data dictionary provides a list of data record and data field aliases that can be referenced when creating publication designs. A data map is used to link the aliases used by a publication to the elements within a data format and must be complete before it can be published.

See [Data mapping](#) on page 496 and [Working with data dictionaries](#) on page 505 for further information.

You may want to use this feature if you have a static input data environment and want to be confident that the aliases used in publication designs are represented in the actual input data file that will be available in the production environment.

To export a data dictionary and data map: in the navigator browse to and select the data format that you want to use for the data map. Select **Tasks/Generate Dictionary** to display the data dictionary and data map wizard. The wizard will guide you through the generation process.

Importing data formats from Data Flow™

The Data Flow data integration solution allows you to select and join data from multiple sources into a single data feed.

When working with keyed record or delimited data, you have the option in Data Flow to create a schema describing the structure of the data file (Sink). This schema can be associated with your input data in the DFE to automatically define the record and field structure of your input data.

To mark up data automatically using a Data Flow schema:

1. Create a new data format using your data from **Data Flow**, see to **“Task: Creating a data format”** for details.
2. From the **Define** menu select **Import Data Format Schema** – or – click the **Import Data Format Schema** toolbar button.
3. Locate and load the appropriate schema.
4. Assign a default locale to determine regional delimiters within number and date fields.

In addition to the import process, you can optionally export a schema from marked up data in the DFE , which can be used outside Designer if required.

To export an XML schema from marked up data in the DFE : from the **Define** menu select **Export Data Format Schema**

– or –

click the **Export Data Format Schema** toolbar button.

33 - About data mapping

When you are designing publications in the Designer you work with indirect references to fields and records. Before you can publish a design all such aliases must be linked with actual data elements that are defined in the data format to be used with the publication in the production environment. These cross-references are defined and maintained in a data map object.

In this section

Data mapping.....	496
-------------------	-----



Data mapping

If you do not already have a suitable data map for a publication you will need to create one before you can complete the publishing task. You can do this as part of the publishing task itself. Alternatively you can create a data map in advance and assign it to a publication. You will need to do this if you want to use a sample data file to show examples of 'real' data within a design before it is published. Refer to [Running a publishing task](#) on page 602 for further information.

Editing a data map

- Step 1: select the data format to be associated with the map.
- Step 2: select the appropriate publication data set from those defined in the data format. (The data format may support several publications.)
- Step 3: map each data record used in the publication design to a record definition in the data format
- Step 4: map each data alias used in the publication design to a field definition in the data format.
- Step 5: the summary page displays the mappings and enables the Finish button allowing you to create the data map.

You can also export both a data map and a data dictionary from a data format. This will allow document designers to work with a list of known data elements and avoids the need to link individual elements manually. Refer to [Exporting to other objects](#) on page 493 for further information.

Data maps can be edited using either the data mapping wizard or data map editor. The wizard allows you to define the required links using models of the data aliases used within a publication and the data elements defined in a data format.

Creating a data map

Data maps can be created and edited from the **Data Mapping Wizard** when configuring a publishing job, however, they can also be created from the **Data** tab. From the **Data** tab data maps can be created independently of a publication, either using the **File/New Data Map** option or by using the **Generate Dictionary** feature.

To create a data map:

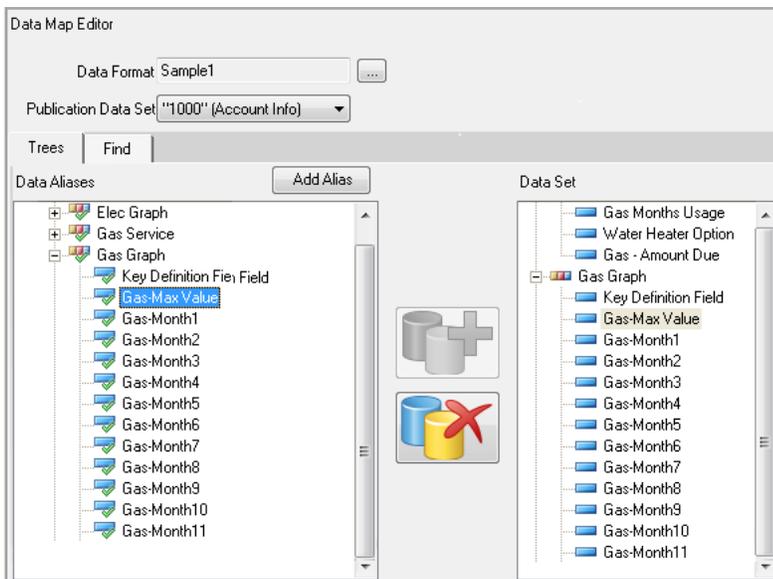
1. In the navigator browse to the required project or folder and on the **File** menu select **New/Data Map** or right-click in the navigator and click **New/Data Map**. A new data map object is displayed. Enter a name and optional description and press **OK**. The Data Map Editor is displayed.
2. In the **Tree** view click the **Data Format** browse button to select the data format to use with the data map.
3. From the **Publication Data Set** list, select the data set you want to use from the data format.

- Click **Add Alias** to select a record or alias from a data dictionary you want to map to a record or field in the data set.

Repeat the step for each record or field you want to include in the data map.

For more information on Data Dictionaries, see [Working with data dictionaries](#) on page 505.

- Select a field in the **Data Aliases** list and select a corresponding field in the **Data Set** list. Click the **Check** button to map the fields.



- Clicking the **Data Format** button enables you to change data formats. The **Data Set** list enables you to select from any data sets associated with the data format.
- Click **Add Alias** to add a alias from a data dictionary.
- Clicking the active button will create or break the link between the highlighted data aliases.
- Selecting a data alias will highlight what it is mapped to.

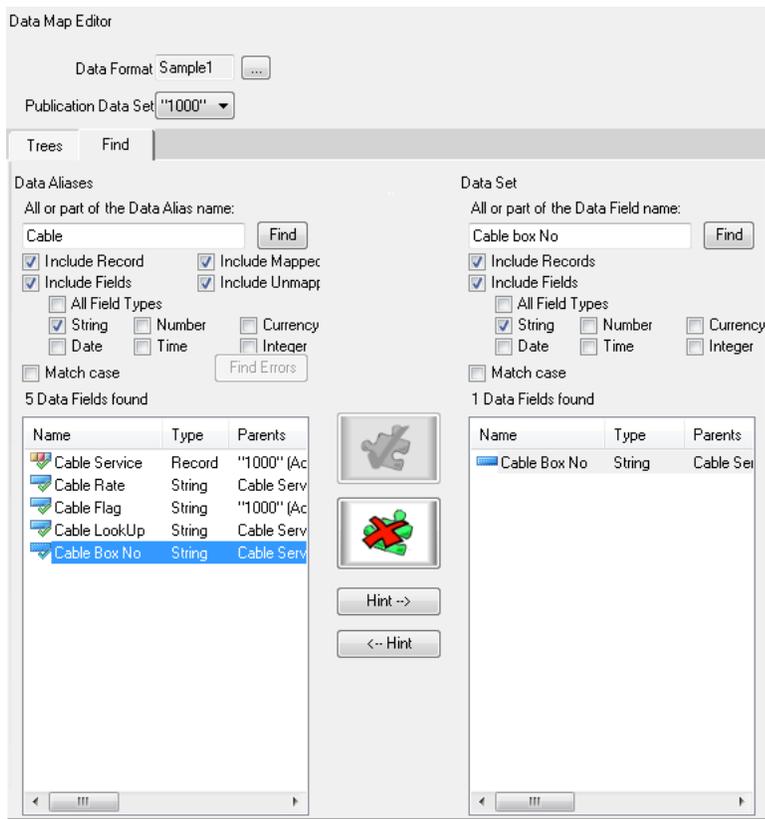
Editing a data map

Once a data map has been created you can change the data element mappings, as well as the data format used by the data map. These changes can be made in either the **Tree** or **Find** view.

The **Find** view enables you to search for data elements defined in either the data dictionary or the data format associated with the data map. This is useful when the data map contains a large number of elements. As well as being able to focus your search on specific types of records and fields, the **Hint** option can be used to suggest possible mappings.

To change the data format for a data map:

open the data map you want to modify and the Data Map Editor is displayed. Click the **Data Format** browse button and browse to a different data format. Now follow “To create a data map” from [step 3](#) onwards.



- The data aliases of the dictionary and data set are searched separately.
- Selecting a data alias and clicking **Hint** will search for the name of a data alias in the opposite list that matches the highlighted name.

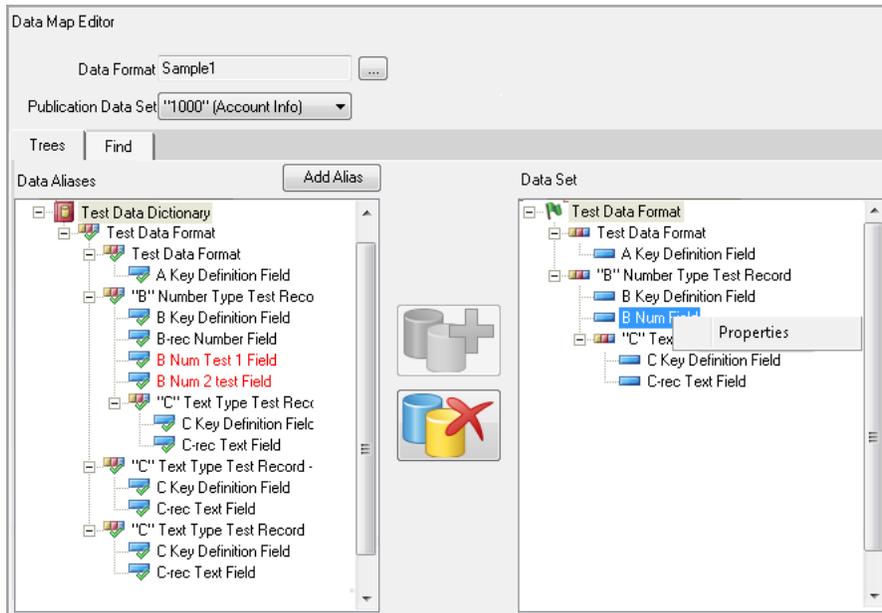
To change mappings: highlight the field you want to re-map in the **Data Alias** list and it highlights the field it is mapped to in the **Data Set** list. The button becomes active, enabling you to un-map the fields. You can now re-map the data alias.

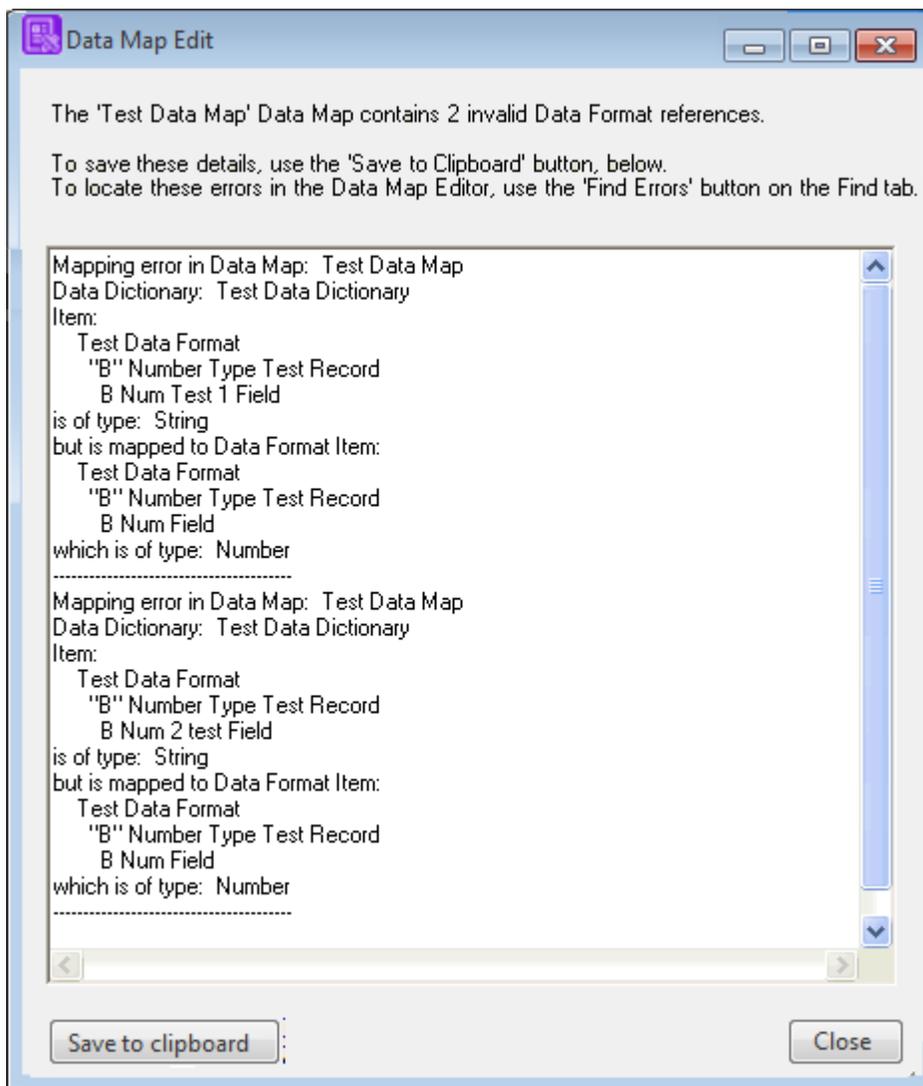
To view data element properties: right-click the element in the **Data Alias** or **Data Set** list and select **Properties**.

To search for data aliases: select the **Find** tab in the Data Map Editor. If required, enter the name of the alias in the **Data Alias name** field. Select the necessary parameters and click **Find**.

Invalid data map references

The data map can get out of step with the data format as the data evolves during the design – for instance if a field is renamed, its type changed, or if its position in the hierarchy is altered by moving it into a repeating group. This will cause invalid references in the data map.





- Use **Find Errors** in the **Find** tab to help locate & correct invalid mappings.
- Right-click on an element to display its properties.
- Invalid mappings detailed here are shown in red in the data map.
- Save this information and paste into a suitable text editor for reference.

These invalid references can be corrected automatically in the Designer by using the **Update mappings for amended data format elements** option in the Data Dictionary and Data Map Generation Wizard.

You can also correct invalid references when editing a data map. Details of the invalid references are listed in a separate dialog box when the data map is opened. The data map editor will then be opened when this dialog box is closed, and you can correct the invalid references.

Data Mapping Wizard

Use **Data Mapping Wizard** to map publication data set from a data format to data aliases used in a publication.

Open **Data Mapping Wizard** from the **Publish Wizard**, **Production Job** view or the **Publication** dialog box.

The **Data Mapping Wizard** provides a **Basic** and an **Advanced** view.

The **Basic** view simply lists the publication data elements and enables them to be associated with a data format.

The **Advanced** view provides the ability to search the lists. The same options are provided in the **Data Map Editor**, see [Data mapping](#) on page 496 for more details.

To map a publication's data aliases: invoke the **Data Mapping Wizard** by doing one of the following:

- In the **Document Editor** click **View/Publication** and double-click the publication in the logic map to display the **Publication** dialog box. Click on the **Data Map** button and select a data format.
- In the navigator browse to and select the required publication and click **Tasks/Publish...** to display the Publish wizard. Select the publication in the **Publications** list and click the **Assign Data Map** button.
- In the navigator browse to and open the required production job. Select a publication from the **Publications** list and click the **Assign Data Map** button.

Generating a dictionary and data map

Use **Generate Dictionary** to create a dictionary and data map from a data format. The data elements in the data format are replicated as records and fields in the new dictionary with the mappings already completed in the new data map. Use **Data Dictionary** and **Data Map Wizard** to create and update dictionaries and maps. It enables you to select which publication data set you want to use and what names you want to use for the new objects.

When updating a data dictionary and data map, additional options for handling changes to the data format are available. If changes are made to the data format, such as changing the data type of an element from a number to a decimal, you can use the **Update mappings for amended data format elements** option to rationalize the data mappings and ensure that spurious mappings between data aliases and data elements do not occur.

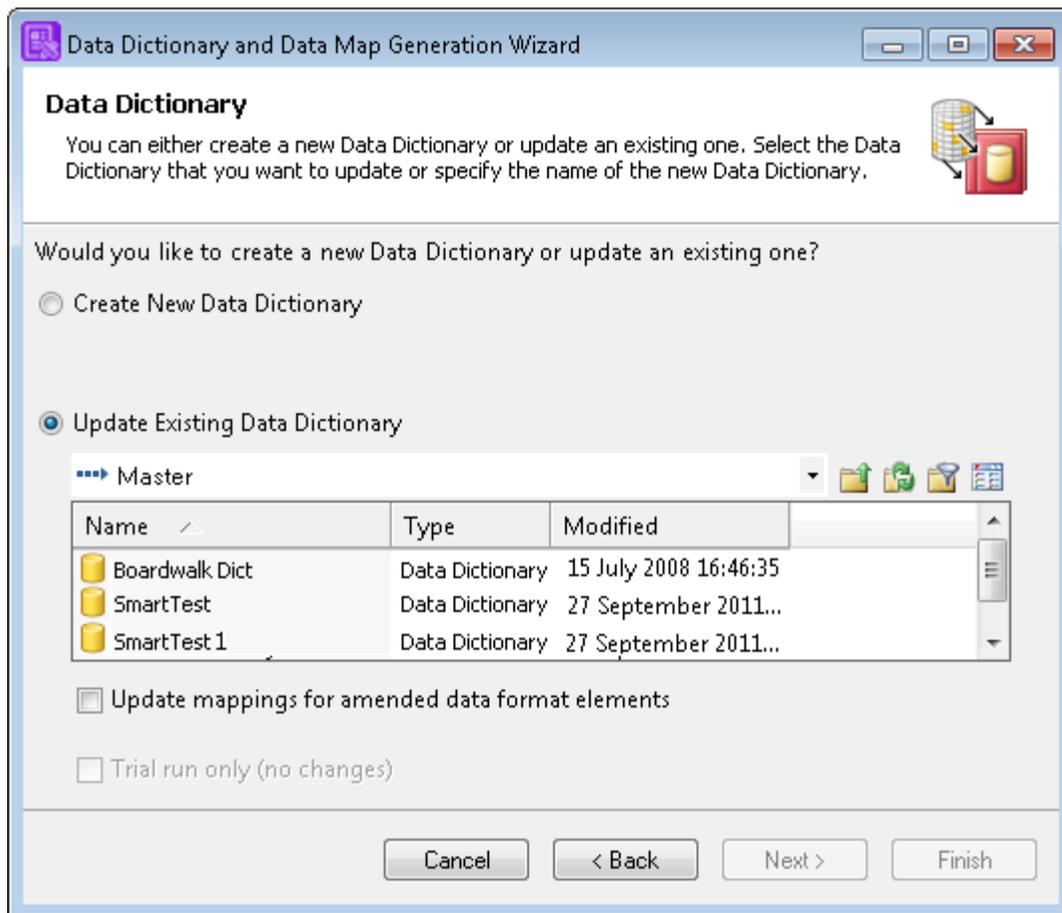
Using the **update mappings** option every time you make a change to the data format will keep the data map from having multiple data aliases mapped to the same data format element.

If you have a data dictionary and data map which have not made use of the **update mappings** option they may contain any number of extra data aliases. As this may have been unavoidable in previous

versions of the product, using the **update mappings** option on a legacy dictionary and data map will attempt to remove multiple data aliases to data element mappings.

To generate a data dictionary and data map: in the navigator browse to and select the data format that you want to use for the data map. Select **Tasks/Generate Dictionary** to display the **Data Dictionary** and **Data Map Wizard**. The wizard will guide you through the generation process.

To update a data dictionary and data map: select the data format used for the dictionary and data map to be updated and select **Tasks/Generate Dictionary**. After selecting the **Publication Data Set**, select the **update** options where applicable and the wizard will guide you through the update process. When it is complete a dialog box is displayed showing which mappings have been affected.



- **Update mappings** ensures that changes made to the data format do not lead to spurious mappings between data elements in the data map.
- The **Trial run only** option shows the impact that updating the mappings will have on the Data Map and Data Dictionary, without committing the changes.

Note: Wildcard name matching for is available with the filter -  option, refer to **Filtering by name using wildcard matching** on page 30 for further information.

Mapping data elements

Map between publication data elements and data format elements.

1. Click a publication data element.
2. Click the data format element.
3. Click the active icon to confirm or break a link.

34 - About data dictionaries

The data record and data alias references created within publication designs are always added to a data dictionary object. The purpose of these dictionaries is to allow you to group such data aliases into reusable categories.

The most obvious categorization is to list the data elements available to a particular production job – i.e. those defined in the data format that describes the contents of an input data file. To assist with this requirement you can export a data dictionary (along with a data map) from an existing data format. Refer to [Exporting to other objects](#) on page 493 for further information.

In this section

Working with data dictionaries.....	505
-------------------------------------	-----



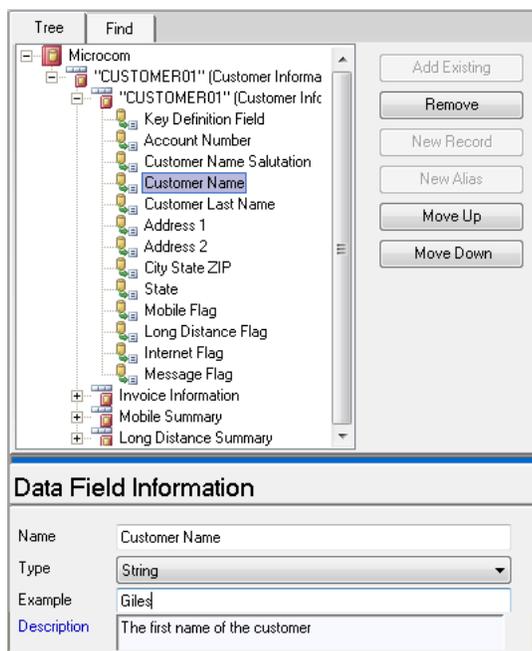
Working with data dictionaries

You may want to create other dictionaries to suit local requirements. Until you define custom data dictionaries a 'System dictionary will be assumed for all new data aliases that are created.

The data dictionary editor allows you to maintain existing dictionaries and to create new data aliases within a dictionary in advance of using them in publication designs.

In the editor the dictionary is presented as a tree structure and you can, for instance, place fields within records or nest records within other records. You may want to do so to reflect the structure of an actual input data file or to highlight other links between the data elements to publication designers.

NOTE: It is important to note that the structure shown within the data dictionary in no way influences how Generate sees its input data in the production environment. This is defined solely in the data format being used with a particular job



- Click and drag to move an element within the tree or use the buttons.
- The lower pane displays details of the selected alias. You can make changes as required.

To add a new data alias or record to a dictionary:

1. Open a data dictionary.
2. If the new alias is to be nested – e.g. if you want to add a data alias within a data record – click on the existing alias that will act as the parent.
3. Click the **New Record** or **New Field** button.

4. You can:
 - Use the tabs to toggle between **Tree** view and **Find**.
 - Click and drag to move an element within the tree or use the buttons.
 - The lower pane displays details of the selected alias. You can make changes as required.
5. In the **Create a New Alias** dialog box give the new field a reference **Name** (which will be shown in selection lists) and add an additional **Description** if required.
6. For data aliases:

- The **Dictionary** field is inactive when creating a new data alias.
 - Select a data **Type**; this is used to decide default output formatting for the field although it can be adjusted later. See [About data types](#) on page 172 for details.
 - Enter **Example** data; this is used as sample contents when the field is displayed in the editor.
7. Click **OK** to create the new alias.

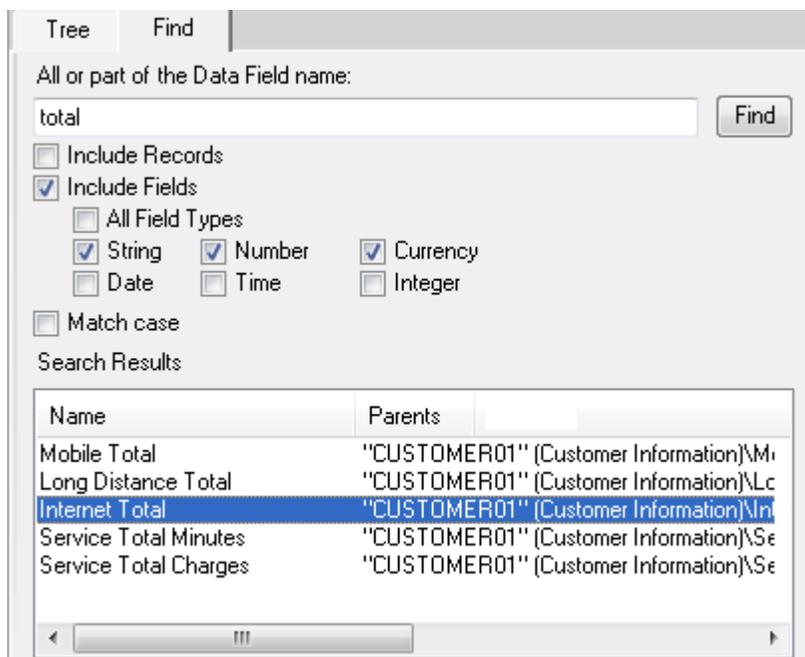
To copy an alias from another data dictionary: click on the appropriate alias if the new alias is to be nested and click the **Add Existing** button. Select the alias to be copied using the dialog.

To delete a data alias: select the alias and click the **Remove** button.

To edit a data alias: double-click on the alias and edit the settings as required.

Creating a data dictionary

- To create a new data dictionary in the navigator browse to the required project or folder and on the **File** menu select **New Data Dictionary**.
- To export a data dictionary from a data format, see [About data mapping](#) on page 495.
- To view or edit an existing data dictionary double-click on the object in the navigator.
- To search a data dictionary open a data dictionary and click the **Find** tab. Enter the name for the data alias, select the necessary search parameters and click **Find**.
- Select the necessary search criteria to focus your search.
- Select an alias from the search results pane to highlight it in the Tree view.



35 - About the production cycle

Generate is the batch program that processes production jobs on your chosen host system. Generate reads information about the job requirements from a Designer file, merges the input data file it is passed with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

In this section

Into production	509
-----------------------	-----



Into production

A production job is made up of one or more publication designs that share the same input data file. The appropriate publication is selected automatically when the control data with which it is associated (the **Start of publication** record as defined in a data format) is encountered in the input data.

Production jobs are prepared using a Publish task in the Designer. This brings together the selected publications and creates a single HIP file that will control the job in the batch environment. The HIP file contains all the information and resources required by a Generate to process the job and to print/present the output datastreams it creates on the actual output devices. The settings you specify when publishing can be saved in a Production Job object for reuse.

Before you can publish you will need to provide information about your intended printer/browsers in one or more output device objects. You will also need to create a host object that provides operating system and deployment information regarding the intended production platform.

If you have unresolved data aliases in your publication designs – i.e. references to data fields or records that have not yet been linked with actual data elements that will appear in the input data file to be used by Generate – you will be asked to map them as part of the publishing process. Such cross-references are stored in a Data Map object for future reference.

At the completion of a publishing task a HIP file is created and is deployed to the location indicated in the selected host object. If the necessary connectivity is available, deployment can automatically copy the HIP to your host platform using FTP or APPC connections (as defined in the Host object when the HIP was generated). Or you can deploy the file locally and manually copy to your production platform.

Once the production environment is prepared you can run the Generate program on the host system. You will need to specify the HIP file that controls the publication as a parameter when starting the program.

Previewing

Previewing allows you to run a publication through the production engine within the Designer itself without the need to go through the formal publication and deployment process. It produces a PDF version of the intended output and displays it in a Designer window for your review.

When you first launch the Preview function the sample data (if any) assigned to the publication is used to provide the variable data. Once the initial task is complete you can specify an alternative sample file and preview again. Sample data is an example of the data that will drive a Designer application in production and the sample provided must match the data format assigned to the publication or the task will fail. Refer to [Creating a data format](#) on page 472 for full details of creating and assigning a data format to a publication.

If you have unresolved data aliases in your publication design you will be asked to map them to actual data elements (as defined in a data format) just as you would with the regular publishing

process. As with regular publishing you must have a data format for your publication in place before you can run preview.

Previewing a publication

The regular preview feature uses default production settings to preview output as quickly as possible. While there is only one regular preview, each publication can have a specific advanced preview associated with it. The eHTML preview uses settings in a previously configured eHTML output device. It is therefore important to ensure that at least one output device has been configured. In a scenario in which multiple output devices are available, you will be prompted to select the output device to be used as the basis for the preview.

To run preview:

1. In the navigator browse to and select the publication you want to preview and on the **Tasks** menu select **Preview** for the regular preview.
2. Complete data mapping if required.
3. Review output. If the publication has unresolved data aliases you will be asked to update or create a data map that supplies the appropriate linkages. See [Data mapping](#) on page 496 for details. A PDF output datastream is composed and presented in a separate window. Use the **browse** button if you wish to use different sample data for a subsequent preview.

Advanced preview

If required, the advanced preview feature can be used to customize the preview process so that you can include runtime options such as supporting Vault, Message1, Content Author, lookup tables and technical support keywords. You can also edit the settings for the PDF output device. Advanced preview uses a wizard which is very similar to the Publish wizard. Refer to the Publish Wizard in [Publishing and deployment](#) on page 592 for more information.

To run an advanced preview:

1. In the navigator browse to and select the publication you want to preview and on the menu select **Tasks/Advanced Preview**.
The Advanced Preview Wizard is displayed.
2. Create a production job for the advanced preview.
Click **Select** and the **Select Production Job** dialog box is displayed. Select **Create New Production Job** and enter a name. Click **OK**.
3. Complete data mapping if required.

If the publication has unresolved data aliases you will be asked to update or create a data map that supplies the appropriate linkages. See [Data mapping](#) on page 496 for details.

4. Configure the Advanced Preview Wizard settings as necessary.

On the **Output Device** page, the only output device available is PDF.

On the **Input/Output** page, you can choose whether to use the sample data assigned to the publication, or browse for a different one.

For details of the other settings, refer to the Publish Wizard in [Publishing and deployment](#) on page 592.

5. When you have completed configuring the Advanced Preview click **Finish**
6. Review output.

PDF output datastream is composed and presented in a separate window.

To specify sample data: open the publication and select **View/Publication**. Double-click the Publication object within the logic map to display the **Publication** dialog box. Use the **Sample Data** browse button to choose a suitable input file.

36 - About Generate

Generate is the batch program that processes production jobs on your chosen host system. Generate reads information about the job requirements from a HIP file, merges the input data file it receives with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

In this section

Running Generate.....	513
-----------------------	-----



Running Generate

The program typically has the name doc1gen on all platforms. You will need to run the program from the batch environment appropriate to your host system and production process: command line, script, JCL, etc.

On some supported platforms memory resident versions of Generate are available (Server Mode and Started Task). These allow a Generate production environment to remain loaded and for batches of input data passed to the defined channel to be processed dynamically. You will need to configure the environment before running in these modes and the launch method will differ from that described in this section. For more details see “Running Generate in Server Mode” and “Running Generate as a Started Task” in the Production Guide.

The name and location of the HIP file that is to control a job is specified as a parameter to the doc1gen program when it is started. Normally, all other file references will have been specified in the production job settings that were used when the job was published in the Designer. If required however, you can create an Override Production Settings file (ops) in which you can specify additional/alternative file references and other settings. Where used an OPS file is specified as a second parameter to the DOC1GEN start-up command.

Segmented resources

A segmented resource is an independently published resource or Active Content.

- Generate needs to know where to find the segmented resources as they are not present in the design HIP file specified on the command line. The resource HIP is necessary for a job must be specified in the ops file using the `<Input>` section `ResourceHIP` and `ActiveContentLocation` keywords. One or more `ResourceHIP` keywords can be used, allowing device specific resources to be kept separate if desired. Segmented resources must be in the same repository as the publication design.
- Generate has no control over resource versions when working in this way. It is the user's responsibility to ensure that all HIP files are compatible and contain the appropriate resources.
- Care must be taken if resource HIP files specified by the `ResourceHIP` keyword contain conflicting device settings for the same device type. In this case the settings used by the last loaded HIP will be used, possibly resulting in unexpected output.
- The `<Output>` `Name` keyword is specified differently when resources are published independently. Refer to the `<Output>` section of the OPS file.

Code page support

In order to read input data and configuration settings specified on your production system DOC1GEN needs to be able to convert the data it receives into the Unicode format it expects internally.

To do this it uses a range of code page tables that define the required translations. For most Western applications these tables are contained within DOC1GEN itself and you need to take no specific action in the production environment. Due to their potentially large size, code page tables for non-Western applications are stored in a separate Extended Code Page (doc1ecp) file and you will need to ensure this is made available to DOC1GEN by referencing it when starting the program.

Note: doc1ecp is distributed with the Generate release material for all supported platforms.

Return codes

DOC1GEN always returns 0 (zero) for successful completion or where warning messages (only) have been issued. Return code 16 is issued where a failure has occurred – i.e. where an abort message has been issued.

Messages

Messages issued by Generate are always written to the standard output medium for the system on which the program is running. A list of possible messages and their explanations can be found on the [EngageOne Compose Support](#) web site.

Legacy Support

The DOC1 Series 5 production engine cannot process jobs created in a pre-Series 5 environment without modification.

You can use Generate to automatically launch the Series 4 production engine (known as EMFE). To do this you need to call the DOC1EMFE program with an OPS file (see below for details) that indicates the location of the DOC1EMFE program and the relevant EMFE initialization file. You should refer to your Suite 4 user documentation for details of the parameters and files that are required when using this method.

Debugging publications

When an error occurs in the logic of your publication design, you can use the debugging option in Generate or Preview to trace the problem.

Processing stops on the first publication that causes the error and a tree structure is generated that represents the path the program took when the error occurred. This can help in tracing the problem to the relevant object in the publication design. The tree structure shows logic map labels so it is advisable to ensure that your design is fully labeled.

Associated data is also shown, although this may include data from previous publications so you must be aware of which data the publication actually uses. Information on the internal opcodes being executed can also be shown if required. The debug information is output either to a file or to the standard output device.

Debug options can be specified in the:

- publishing task
- diagnostic section of the Publish Wizard
- OPS file

The **TraceLevel** option in the OPS file can be used to switch debugging mode on or off each time you run Generate.

OPS file reference

An override production settings (OPS) file allows you to specify supplements and alternatives to some of the job settings that were used when publishing a publication design. The use of an OPS file is optional unless you have not specified all file references in the publishing task. Refer to the EngageOne Generate Production Guide for detailed information on usage and OPS settings.

Using symbolic references

You can dynamically define parameters used in an OPS file by using symbols when starting a Generate job. The value assigned to a symbol is substituted wherever it is referenced in the OPS file and can be used to provide part or all of any parameter.

Where used, symbols must be defined after any other parameters in the start-up syntax. Under Windows for example, you could specify the following on the command line:

```
doclgen j1.hip ops=j1.ops ext=txt
```

where `ext` is a symbol name. When referenced in the OPS file the symbol names must be enclosed in percent (%) characters. For example:

```
<Journal>J1=\trace\docj1.%ext%J1=\trace\docj1.%ext%...
```

Symbol names are case sensitive. Where a symbol is referenced in the OPS but no value is assigned it is treated as an empty string.

Note that the following must not be used as symbol names: `ops`, `mode`, `ecp`, `mmgx`.

Symbols may also be defined within the OPS file itself by coding them in an `<OPS>` section prior to where they need to be referenced. The format is as follows:

```
<Symbols>Name =ParameterName =Parameter...
```

If the same symbol name is specified both in start-up parameters and in the OPS itself then the start-up parameter will override. For example; in the OPS:

```
<Symbols>
RunName=runBaseDir=\doc

<Journal>
J1=%BaseDir%\trace\docj1.txt
J2=%BaseDir%\trace\docj2.txt
J3=%BaseDir%\trace\%RunName%j3.txt

<Output>
Output1=%RunName%1.afp
Output2=%RunName%1.pdf
```

and on the Windows command line:

```
doclgen in.hip OPS=OFile RunName=tst
```

the 'J3' journal name will be specified as `\trace\tst.txt`.

Operating system considerations

You will need to ensure that all input files referenced in the productions settings and the OPS file are available at the locations specified when DOC1GEN is started. For jobs running under z/OS indirect file references using DD names are typically used in which case the start up JCL will need to include DD cards with these labels that indicate the actual datasets.

Output files will be created with the file location/names specified in the production settings and the OPS file. You will need to ensure that suitable disk space is available to receive the output at the defined location. Under z/OS you will need to allocate suitable datasets either in advance or as part of start-up JCL if DD references are used.

DOC1GEN under z/OS

Preparation:

A DOC1GEN job is typically submitted to the system via standard JCL.

The JOBLIB and STEPLIB concatenation must reference the main Generate load library and message library datasets. You may also need to include references to the IBM Language Environment (LE) run-time libraries if these are not known to system libraries.

The HIP file that controls the job plus an OPS file (where used) are specified as parameters on the EXEC card. However, these normally indicate DD references that are resolved to dataset names in subsequent DD cards.

The JCL must also include DD cards for all other files that have been identified using DD references in the HIP or OPS.

No two output files should be members of the same dataset.

EXEC card syntax:

```
EXEC PGM=DOC1GEN,PARM=('DD:HipRef [,OPS=DD:OpsRef ] [,ECP=DD:EcpRef ]
[,SCP=CodePage ] [,#restart] [,symbols]')
```

Parameters:

HipRef is the DD label indicating the HIP file that will control the job.

OpsRef	is the DD label indicating an override production settings file if appropriate.
EcpRef	is the DD label indicating the Extended Code Page file which will be required for most non-Western applications. The ecp file must be placed in a fixed block dataset.
CodePage	is the number of a host code page to be used instead of the default code page – US(37).
#restart	restart the job from the last checkpoint position. See “Checkpoint progress” the Publish Wizard checkpoint progress option in the Designer User’s Guide.
symbols	See Using symbolic references on page 515 .

Example JCL

```
//DOCJOB6 JOB '5438','JDOE',CLASS=F,REGION=2M//DOC1GEN EXEC
PGM=DOC1GEN,// PARM='DD:DOCHIP,OPS=DD:DOCOPS,ECP=DD:DOC1ECP'//*Generate
load libraries. You may need to add run-time libs//STEPLIB DD
DISP=SHR,DSN=PROD.DOC.LOAD// DD DISP=SHR,DSN=PROD.DOC.MSGS//*HIP
& OPS files//DOCHIP DD DISP=SHR,DSN=PROD.DOC.RUN(JOB6HIP)//DOCOPS DD
DISP=SHR,DSN=PROD.DOC.RUN(NEWFILES)//*Extended Code Page
file//DOC1ECP DD DISP=SHR,DSN=PROD.DOC.RUN(DOC1ECP)
```

```
//*Input data (as per DD ref in HIP or OPS)//DOCINPT DD
DISP=SHR,DSN=PROD.DOC.DATA//*Lookup tables (as per DD refs in HIP or
OPS)//DOCTL1 DD DISP=SHR,DSN=PROD.DOC.RUN(JOB6TL1)//DOCTL2 DD
DISP=SHR,DSN=PROD.DOC.RUN(JOB6TL2)//*Output datastreams (as per DD refs
in HIP or OPS)//AFPOUT1 DD SYSOUT=X,DCB=LRECL=8205//AFPOUT2 DD
SYSOUT=X,DCB=LRECL=8205//*Journals (as per DD refs in HIP or
OPS)//DOCJRN1 DD DISP=SHR,DSN=PROD.DOC.RUN(JOB6JRN1)//DOCJRN2 DD
DISP=SHR,DSN=PROD.DOC.RUN(JOB6JRN2)
```

DOC1GEN under UNIX, and Windows

Preparation:

DOC1GEN is executed from the command prompt.

The HIP file that controls the job plus an OPS file (where used) are identified as parameters to the start up command.

All other references to files to be used or created by Generate are defined within the HIP or OPS files. You will need to ensure that these are available (or creatable) at the locations indicated.

Syntax:

```
doc1gen HipRef [ops=OpsRef] [ecp=EcpRef] ["#restart"] [symbols]
```

Parameters:

HipRef	is the path/file name of the HIP file that will control the job.
OpsRef	if an override production settings file is being used this is the path/file name of the OPS.
EcpRef	is the path/file name of the Extended Code Page file which will be required for most non-Western applications.
#restart	restart the job from the last checkpoint position. See Running a publishing task on page 602 the Publish Wizard checkpoint progress option in the Designer User's Guide. Note that the quotes are only required when running under UNIX.
symbols	See Using symbolic references on page 515.

Examples

For Windows:

```
C:\doc\run\doc1gen job6.hip ops=C:\doctemp\newfiles.ops ecp=doc1ecp
```

For UNIX:

```
/doc/run/doc1gen job6.hip ops=C/doctemp/newfiles.ops ecp=doc1ecp
```

37 - Output, media and resources

In the Designer an output device object identifies the type of data stream to be generated by a publication and allows you to specify any custom settings related to the output environment.

Pre-configured output device objects are provided that reflect the most common requirements for the environment indicated by your product license. You can override these to provide custom settings as required. One or more output devices will need to be indicated when you use the **Publish** task to put a publication into production.

In this section

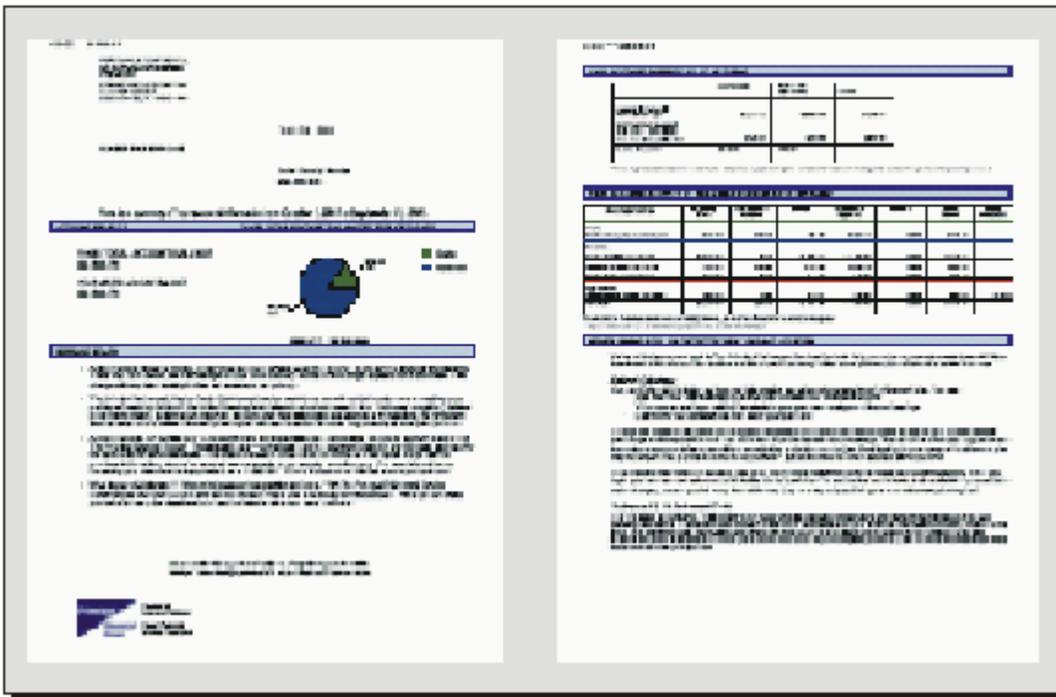
Logical pages and media layout.....	521
Font and image resources.....	523
Format of the output datastream.....	524
Support for ideographic text.....	524
Creating output device objects.....	524
AFP.....	526
PCL.....	536
Postscript.....	541
PDF.....	549
RTF.....	560
Line Data.....	563
HTML.....	566
Specifying a file template.....	585
Using resource maps.....	586



Logical pages and media layout

The page layouts you design in Designer are considered to be 'logical pages'. These are placed on the actual media pages – known as 'sheets' – using a defined offset for the top left corner. By default, a publication places a single logical page on each sheet and positions it as close as possible to the top left corner.

If you need to add multiple logical pages to a sheet you can use the **Layout** options of an output device object to customize the page layout. You can define how many logical pages are required per sheet and the sequence in which they should be used. Such media layouts are often known as '2-up', '4-up' or similar.



'2-up' layout printed on landscape dimension sheet size

- Page 1 **Offset**
- Page 2 **Offset**
- **Sheet**
- Logical **Page**

When working with multiple logical pages the intended sheet size becomes important. You will need to use the **Sheet size** output device options to let Designer know the area it has to work with when placing the logical pages.

For physical output media Designer also assumes that only one side of the paper will be printed on. This is known as 'simplex' mode. If your printer supports double sided printing you can use the **Duplex** or **Tumble** output device option, depending on which options are available on your printer hardware, to specify that both sides of the paper are to be used.

If you have specified duplex or duplex tumble printing, it is possible to switch between duplex, tumble and simplex modes by using an action in application logic. By default, printing of the first publication produced by a job will start with duplex mode where it is defined; if you want to start printing in simplex or tumble mode, you will need to specify an action before processing of the first document in the logic – see [Controlling the output environment](#) on page 454.

BookBill



- Sheet 1 of an example **BookBill** layout
- Logical Page
- Bookbill slip

An additional option allows you to work with the specialized 'BookBill' media layout. This format assumes that a payment slip or similar is part of the first physical page to be produced by a publication design. The area reserved for the slip is in addition to the regular media layout – i.e. the first sheet produced has a larger area than any subsequent sheets.

When working with a BookBill format your publication design will need to output an additional logical page in the appropriate sequence that contains the required content for the payment slip. When you are creating output for a duplex process you will need to have two additional logical pages – one per side.

Font and image resources

By default, a publishing task converts the Windows font and image resources used when designing a publication into the format required by the target output datastream environment. These are stored in the HIP file along with all the other controls required by the publication in production environment.

Such resources will be created and referenced using file names that adhere to the requirements of the target output datastream. For PostScript, PDF and variants this normally means the resources can have the same names as used in the design environment. For datastreams that support a limited file naming convention (for instance, AFP) the files will be created with alias names that primarily consist of a unique number reference. For example, an image resource of name 'Q3 results.bmp' may become 'S100014' when a HIP is created for AFP output. If required, you can use a resource map to override such defaults and indicate a specific name to be used for each individual resource file when it is being prepared for use with each type of output datastream.

You may also want to use a resource map to reference the names of existing resources in your output datastream environment. For example, if you already have a printer specific version of helvetica 10pt on your printer (but have not imported it into the Designer) you can indicate that all references to this font are replaced with the existing resource name.

Unless you specify otherwise, font and image resources are always embedded in each document within the output datastream where they are used. Output device settings allow you to turn off automatic embedding and this can provide a significant reduction in the size of the output file created by Generate. However, you will need to consider alternative methods of making the resources available to the printer/browser environment if this is not already the case. The DOC1RPU utility is provided with Generate distribution material to extract presentation resources from a HIP file if required.

NOTE: Refer to the Production Guide for details about configuring and running DOC1RPU.

Overlays

AFP overlays can be included in your document design. As with other native resources, overlays imported in Designer versions prior to 6.6 Service Pack 11 can be included in your document, and can only be inserted in the background layer. Overlays can be referenced in the print environment,

or if you are working with AFP, they can be embedded in the publication HIP file. Note that embedding is controlled via the associated output device setting.

Format of the output datastream

The default format of the logical records that make up an output datastream is set to cater for the most common requirements for each combination of production platform and output device type. However, you may need to adjust this format to cope with local circumstances, in particular if you need to transfer the datastream before it is actually made available to the output device. The **Format of logical records** setting in an output device object allows this customization.

Support for ideographic text

Designer & Generate support text using the extended character sets required for ideographic languages such as Chinese, Japanese and Korean (sometimes known as 'double-byte' text) for several output datastreams. It is not supported for the following datastreams:

- PCL
- RTF

Using these datastreams with publications that output ideographic text will produce unpredictable results.

Creating output device objects

An output device holds the production settings for a particular type of output datastream as it relates to a target printer or browser. For example, you may have a group of settings for AFP 240 dpi printers, another for 300 dpi machines using the same datastream and settings for a local PostScript environment. You would need to create an output device object for each.

NOTE: Detailed information about specific printers or browsers is outside the scope of this document. Refer to the documentation provided for your output device if you are unfamiliar with its requirements.

Every object has a range of general settings that apply to all output devices. These are supplemented by settings specific to the type of output datastream selected for the device.

Deprecated output devices

From Designer 6.6 Service Pack 9 the following output devices are deprecated:

- VIPP, VPS, PPML
- IJPDS
- MIBF
- Metacode

Detailed information on specific production settings for these output devices can be found in the documentation issued prior to the 6.6 Service Pack 9 release of Designer; refer to the [EngageOne Compose Technical Support website](#).

Creating an output device object:

1. Select the required project or folder in the navigator and on the **File** menu select **New/ Output Device**.

The **New Output Device** window is displayed.

2. Enter a name for the device and an optional description and press **OK**. An output device object will be created in the selected project.
3. Configure the settings as required – see the help or section for the respective output device.

To edit an existing output device: select the relevant output device in the navigator and either double-click on it, or from the **File** menu select **Open**. The **Output Device Configuration** window will be displayed.

AFP

An output device holds the production settings for a particular type of output datastream as it relates to a target printer or browser. For example, you may have a group of settings for AFP 240 dpi printers, another for 300 dpi machines using the same datastream and settings for a local PostScript environment. You would need to create an output device object for each.

Every object has a range of general settings that apply to all output devices. These are supplemented by settings specific to the type of output datastream selected for the device.

The level of AFP features related to color and graphics will often vary depending on your specific printer/browser environment. The output device options for an AFP device contain a number of settings that allow you to control specific aspects of these features to suit your installation. In an AFP environment, font, image and overlay resources are commonly stored in a structured file library for access by the software that handles the actual printing/presentation. If you choose not to embed the resources in the output datastream produced by Designer you will need to ensure that they are available in these libraries before the output can be sent to print.

Order of graphics

It is important to note that results may not be as expected when working with shaded boxes in your publication design. Shaded boxes will always be placed in front of other graphic objects regardless of their order in your design.

File structure

By default, AFP output is generated with variable length logical records. Each logical record contains a single AFPDS structured field. Under z/OS these are also separate records within the file structure itself. On all other platforms the logical records form a continuous stream.

The maximum logical record length is normally dictated by the textual data stored in the PTX structured fields for a particular page. Generate will attempt to place all text for a particular page within a single PTX structured field up to a default maximum of 8200 bytes. When publishing an application you can use an output device setting to specify a different limit if required.

Under z/OS

Ensure your output file has a record length suitable to receive the AFP datastream. If you are using the default **Limit record length** setting you should write to an output dataset with an LRECL of 8205.

You can optionally store AFP datastreams created by Generate in VSAM datasets of type **relative record** (RRDS) or **keyed sequential** (KSDS). When VSAM is used the individual composed pages are stored as entries in the VSAM table. This format may be useful if you intend to archive or post-process the pages.

To use this option you will need to specify the 'RRDSAFP' or 'KSDSAFP' keywords in the **Format of logical records** setting or by coding custom formatting parameters for this setting that include the \$BR or \$BK output descriptors. Refer to the Production Guide for further information about custom output formats. You will also need to allocate an appropriate VSAM dataset in your Generate JCL. Note that Generate always assumes VSAM datasets with fixed length records.

AFP output device settings

The section that follows provides detailed information about the output device settings that are applicable when generating AFP output datastreams.

The **Settings** tab contains options for the AFP output datastream being produced.

General

Resolution

This is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output. The default is 240 dpi

Printer carriage control

By default, the default printer carriage control (PCC) byte is not encoded. Use this option to select a coding system to be used for the PCC if required.

Format of logical records

This setting allows you to customize the structure of the logical records that make up the datastream. By clicking the ... button and using the

Predefined option of the **Format** dialog box you can select from a range of keywords that format for a range of known scenarios.

Or, by using the

Custom option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.

Limit record length

This setting indicates the largest permitted size of a PTX structured field within AFP output. PTXs contain the document text and typically produce the longest logical records in the output file. You may want to adjust the default if your production system is z/OS where the size of logical fields is relevant to dataset attributes.

The **Settings** tab contains options for the AFP output datastream being produced.

Graphics structured field size	If Level 2 GOCA is specified the amount of data stored in GAD structured fields may become very large. Use this option to specify the maximum size (in bytes) to be used for such fields before a new field is generated.
Max pels for thickest line	This setting indicates the maximum number of pels to be used when drawing lines and boxes. In AFPDS thick lines (rules) are actually drawn by using a number of lines at adjacent positions. The maximum thickness of any one line varies from printer to printer. Optimizing this setting for your printer and application can significantly reduce the size of generated AFP file.
Level 2 GOCA	When drawing charts or tables with rounded corners Designer uses GOCA instructions within GAD structured fields that conform to recent versions of the AFP architecture. This may cause printing difficulties in some AFP environments particularly where AFCCU controllers which have a version lower than 9.6 are in use. If required you can check this option to force Generate to use earlier versions of the GOCA instructions. Note that this may cause a significant increase in the output file size.
Rotate pages by 180 degrees	All pages will be rotated through 180° before being output.
Embed overlays	Selecting this option will embed the overlay in the publication HIP file. If you wish to reference overlays stored in the print environment then this option must not be selected. The default is not to embed overlays.
Medium Maps	
Default input tray	When working with printed media this setting specifies which input tray will be used to provide paper when the datastream is being printed until a different tray command is received. Note that the input tray and output bin being used can be dynamically modified during job processing. See Controlling the output environment on page 454 for details. The default is Continuous Feed , which means either no input trays are available or that the default tray should be used.
Default output bin	As above but relates to the bin to receive the pages produced by the output datastream (where applicable).

The **Settings** tab contains options for the AFP output datastream being produced.

None	Select if medium maps are not required. Creates output without a medium map, regardless of the settings in the Medium Map tab.
Allow	Allows medium maps to be embedded in the AFP datastream.
Use extended	The default option includes input tray and output bin settings with medium maps when they are embedded in the AFP datastream. When this option is de-selected the medium map's input tray and output bin settings will be ignored.
Color	<p>Use these options to modify the color commands used within the datastream.</p> <p>16 Colors – color references are restricted to the 16 color model used in AFPDS prior to the introduction of full color support. Color instructions within document designs will be mapped to the nearest equivalent color in this model.</p> <p>None – all color instructions within document designs are ignored. No color references are used in the output unless required by the datastream format in which case black is always used. Note – if your publication uses rounded corners on a shaded table, choosing 24bit RGB or 32bit CMYK will ensure consistency of color shades in your publication.</p> <p>24bit RGB – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color.</p> <p>32bit CMYK – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.</p>

The **Settings** tab contains options for the AFP output datastream being produced.

Use GOCA for shading

Use with objects such as Tables, Text Box, Native Barcharts, etc with a specified Shading value.

The Default is to use Avoid - Simple Images for AFP Shading. However, in certain circumstances, a finer level of control is required and AFP GOCA Shading can be specified if required, as follows:

Avoid (Default) – use Simple Image shading

Optimal – use either Image or GOCA Shading based on Object and Resolution

Always – always use GOCA Shading for all Objects

Graphic

Graphic manipulation quality

This will be used for graphics features such as rotated graphics and plug in charts as their use affects both the time taken to generate and the size of the output file, which can be considerably increased.

Default uses the resolution as specified for images.

Low gives the best speed for generating but the lowest graphics quality (96 dpi).

Medium graphics are 240 dpi.

High will take much longer to generate but will be a reasonable quality of graphic (300 dpi).

Image

Resolution

Select a value from the drop-down list if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specified as a dpi – dots per inch value. The setting of 0 (zero) indicates that the image resolution should use the general **Resolution** setting.

Type

This setting is used to ensure that images are printed correctly. The default option is **Black/White** (FS10). Use the **Color** (FS45) option only if your target printer has full color support (i.e. greater than 16 standard colors). For printers with the relevant object container support, the **Use originals (object container)** option enables you to use JPG, EPS, PDF Page Objects, TIF images.

The **Settings** tab contains options for the AFP output datastream being produced.

Color compression	If your publication uses many large images, you may wish to compress them to make the output datastream smaller. Use this option to specify the level of compression for color images. The default is best quality – None – i.e. the image is not compressed.
Do not create	Use this option if the required images are already available to the printer/browser and you do not wish to create them in the output data stream where they are referenced.
Create	Create the AFP images used in your publication. These are not used by Generate but are included in the HIP file for extracting manually to the output device.
Embed	By default, images are always embedded in each document within the output datastream where they are referenced. If the required images are already available to the printer/browser you may want to use Do not create to reduce the size of the datastream.
Minimize embed	Use this option to ensure that only the images used are included in the Generate output file.
Allow scaling up/down	These options determine what happens if the resolution of images used in the publication design does not match the Resolution or (Image) Resolution settings. If checked, Designer will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).
Rotation	By default, Designer uses the AFP Include Object (IOB) command, which allows images to be automatically rotated when landscape pages are being produced.
Color management	Selecting this will embed an image's ICC profile in the output datastream in CMR (Color Management Resource) format. This can improve the color quality of the image on printers that support it.
Font	

The **Settings** tab contains options for the AFP output datastream being produced.

Output format	<p>Determines the type of AFP font resource to be created. Options are:</p> <p>AFP Bitmap (FOCA) – fonts are created as native AFP bitmap fonts (non-scalable) in accordance with the AFP FOCA definition. This option produces fonts suitable for most AFP print environment but note that file sizes will be very large where DBCS output is required.</p> <p>TrueType (DOF) – TrueType or OpenType fonts used with the publication design are included directly in the output datastream within an object container. While this will produce much smaller file sizes than the bitmap option, note that your AFP printer must specifically support such Data Object Fonts (DOF).</p> <p>AFP Outline (FOCA Type1) – fonts are created as native AFP FOCA Type1 fonts (Embedded Type1 Outline fonts) in accordance with the AFP FOCA definition. This option produces fonts suitable for most AFP print environments.</p> <p>Note that imported AFP bitmap fonts will not be converted to outline format.</p>
Method for underlining	<p>Where text is to be underlined this setting specifies the drawing method that will be used.</p> <p>Native – indicates that the underlining method within the output datastream protocol will be used if possible. This is the default.</p> <p>Emulate – when underlining is required an independent line is drawn under the appropriate text. Note that if the protocol cannot support underlining in a reasonable manner (as with LINEDATA output), then this setting is ignored.</p>
Do not create	<p>Use this option if the required fonts are already available to the printer/browser and you do not wish to create them in the output data stream where they are referenced.</p>
Create	<p>Create the AFP fonts from the Windows fonts used in your publication. These are not used by Generate but are included in the HIP file for extracting manually to the output device.</p>
Embed	<p>By default, fonts are always embedded in each document within the output datastream where they are referenced. If the required fonts are already available to the printer/browser you may want to use Do Not Create to reduce the size of the datastream.</p>

The **Settings** tab contains options for the AFP output datastream being produced.

Minimize embed	Use this option to ensure that only the fonts used are included in the Generate output file.
----------------	--

Build for IBM OnDemand	If checked, the output datastream can be viewed in the IBM OnDemand environment. You need to Run DOC1ACU to extract the necessary control information for integration with IBM OnDemand, see the Production Guide for further information.
------------------------	--

EngageOne

Enable as EngageOne device	This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).
----------------------------	--

Media

This page enables you to edit the medium maps for the default **Formdef** file provided with Designer – **F1G1DFLT**, as well as create and delete medium maps for new formdef files. This AFPDS printer file contains information about the printer and a collection of medium maps. The medium maps contain information on input trays, output bins and printer modes. The **F1G1DFLT** formdef file is automatically generated and placed inline to the print stream. If another name is specified in the **Formdef** field then the formdef is not generated.

General

FormDef	<p>Contains the name of the default formdef file provided with Designer.</p> <p>Double-click on an existing mapping and use the Edit Mapping dialog box to edit the Input Tray, Output Bin and Printer Mode details that will be used with it. Click the + icon for a new mapping, provide a name for the map and fill out the details required. Use the - icon to delete a highlighted medium map.</p> <p>NOTE: Medium map names are uppercase and have a maximum length of eight characters.</p>
---------	--

Layout

The Layout tab enables you to configure settings necessary for placing multiple logical pages on a physical sheet of paper. The layout settings are dependent on knowing the size of the physical page, so if the output device's **Sheet Size** option is set to **AUTO** the layout options are disabled.

The **Settings** tab contains options for the AFP output datastream being produced.

Sheet size

By default, the size of the physical sheets of paper used on the printer is assumed to be the same as the logical pages within your document designs. This is the **AUTO** option. Where this is not the case use this setting to specify the actual sheet size being used. You can choose from a range of standard sizes or select **CUSTOM** and enter specific dimensions.

Duplex processing

This setting determines if the datastream should be presented in simplex (**single-sided**) or duplex (**double-sided**) mode. Note that the duplex options can be modified dynamically during job processing. For printer drivers that support 'tumble' mode you can choose **double-sided tumble**. This causes the reverse side of the page – i.e. even pages (2,4,6 etc.) – to be output the opposite way round. See [Controlling the output environment](#) on page 454 for more information.

Pages per sheet

Select how many logical pages from your publication you want to output per side of sheet. The **Pages across** and **Pages down** options control the total number of pages for each sheet.

Book Bill remittance slip – this option enables you to include a remittance slip with each publication. The remittance slip is added by Generate to the end of the first sheet of the publication and does not affect the Sheet Size setting or any other output device option.

NOTE: For the BookBill option to function correctly you must include a remittance slip specific document in your publication. The page orientation for the document must be landscape and the page size cannot be larger than that specified by the Slip width and height settings. See [Using multiple documents](#) on page 160 for more details.

The **Settings** tab contains options for the AFP output datastream being produced.

Placement order

This setting controls the flow of the page placement on each sheet. The placement options are:

- **Pages flow left to right** or **Pages flow right then left**
- **Pages flow across and then down** or **Pages flow down and then across**

To ensure the publication prints correctly when using duplex printers the back side of each sheet is printed in the opposite order. **Place multiple publications on same sheet** – this enables different publications to be placed on the same sheet of paper. A new publication will not be printed on the back side of a sheet of another publication. This option will not be available if **BookBill remittance slip** is selected.

Placement details

These settings list the logical pages and on which **side** of the sheet they are placed. It also displays the **offset** for each page which is automatically set in the Layout tab. It is assumed that each logical page is the same size. If you need to change the page size, double-click on the corresponding page and the **Page Details** dialog box enables you to change the settings.

NOTE: When a change is made that causes the Placement details to be regenerated all the offset values are reset. For instance, if the sheet size, width, height or printer mode is changed in the Settings tab.

Example

This displays the layout of the publication's logical pages on the physical sheets after Layout tab options have been set.

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

PCL

If you need to transfer PCL files between operating systems be aware that, by default, Designer applications generate such output as a true stream (as normally expected at the printer). For instance, if receiving a PCL file from z/OS onto a PC workstation you should not specify the CRLF attribute.

Positioning of the text can vary from printer to printer when non-zero orientations are used. You may need to adjust the position of text in relation to the bottom/right side of the logical page to cope with such variances.

PCL output device settings

The **Settings** tab contains options for the PCL output

General	
Resolution	This is set at 300 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.
Format of logical records	This setting allows you to customize the structure of the logical records that make up the datastream. By clicking ... and using the Predefined option of the Format dialog box you can select from a range of keywords that format for a range of known scenarios. Or, by using the Custom option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.
Bottom anchor	Specify the number of pels by which orientated text is to be adjusted in relation to the bottom of the logical page.
Right anchor	As for Bottom anchor position.
Top pels offset	The number of pels by which printing is to be offset from the top of the physical page.

The **Settings** tab contains options for the PCL output

Left pels offset	The number of pels by which printing is to be offset from the left side of the physical page.
Use non-HPGL/2 commands	The PCL commands used to draw charts are, by default, in HPGL/2 format as supported by PCL5 printers. Check this option if your printer does not support HP/GL2 and your publication does not use charts (other than 2D bar charts). Note that if your publication uses advanced chart features and this option is set, Generate will abort.
Media	
Default input tray	Input trays are used by the printer to provide paper for the output datastream. Configure the necessary input trays by selecting them from the available input trays in the list. The default is Continuous Feed which means either no input trays are available or that the default tray should be used.
Default output bin	The output bins provide the location where the pages are sent after being printed. Configure the required output bins by selecting them from the list. The default is Continuous Feed .
Graphic	
Graphic manipulation quality	This will be used for graphics features such as rotated graphics and plug in charts as their use affects both the time taken to generate and the size of the output file, which can be considerably increased. Default uses the resolution as specified for images. Low gives the best speed for generating but the lowest graphics quality (96 dpi). Medium graphics are 240 dpi. High will take much longer to generate but will be a reasonable quality of graphic (300 dpi).
Image	
Resolution	Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a Dpi – dots per inch value. The default setting of 0 (zero) indicates that the image resolution should use the general Resolution setting.

The **Settings** tab contains options for the PCL output

Mode	The default Color option should be used if you are using a color printer. If a monochrome printer is being used you must select the Black/White option.
Embed	By default, images are always embedded in each document within the output datastream where they are referenced. If the required images are already available to the printer/browser you may want to uncheck this option to reduce the size of the datastream.
Allow scaling up/down	These options determine what happens if the resolution of images used in the publication design does not match the Resolution or (Image) Resolution settings. If checked, Designer will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).
Font	
Method for underlining	Where text is to be underlined this option specifies the drawing method that will be used. Native – indicates that the underlining method within the output datastream protocol will be used if possible. This is the default. Emulate – when underlining is required an independent line is drawn under the appropriate text. Note that if the protocol cannot support underlining in a reasonable manner (as with LINEDATA output), then this option is ignored.
Output format	The default Bitmap will include a separate font for each different point size used. TrueType will include a single font to represent multiple point sizes of the same font, which can reduce the size of the PCL output file.
Embed	By default, fonts are always embedded in each document within the output datastream where they are referenced. If the required fonts are already available to the printer/browser you may want to uncheck this option to reduce the size of the datastream.
EngageOne	
Enable as EngageOne device	This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).

The **Settings** tab contains options for the PCL output

Layout

The Layout tab enables you to configure settings necessary for placing multiple logical pages on a physical sheet of paper. The layout settings are dependent on knowing the size of the physical page, so if the output device's **Sheet Size** option is set to **AUTO** the layout options are disabled.

Sheet size	By default, the size of the physical sheets of paper used on the printer is assumed to be the same as the logical pages within your document designs. This is the AUTO option. Where this is not the case use this setting to specify the actual sheet size being used. You can choose from a range of standard sizes or select CUSTOM and enter specific dimensions.
------------	---

Duplex processing	This option determines if the datastream should be presented in simplex (single-sided) or duplex (double-sided) mode. Note that the duplex options can be modified dynamically during job processing. For printer drivers that support 'tumble' mode you can choose double-sided tumble . This causes the reverse side of the page – i.e. even pages (2,4,6 etc.) – to be output the opposite way round. See Controlling the output environment on page 454 for more information.
-------------------	--

Pages per sheet	<p>Select how many logical pages from your publication you want to output per side of sheet. The Pages across and Pages down options control the total number of pages for each sheet.</p> <p>Book Bill remittance slip – this option enables you to include a remittance slip with each publication. The remittance slip is added by Generate to the end of the first sheet of the publication and does not affect the Sheet Size setting or any other output device option.</p> <p>NOTE: For the BookBill option to function correctly you must include a remittance slip specific document in your publication. The page orientation for the document must be landscape and the page size cannot be larger than that specified by the Slip width and height settings. See Using multiple documents on page 160 for more details.</p>
-----------------	--

The **Settings** tab contains options for the PCL output

Placement order

This setting controls the flow of the page placement on each sheet. The placement options are:

- **Pages flow left to right** or **Pages flow right then left**
- **Pages flow across and then down** or **Pages flow down and then across**

To ensure the publication prints correctly when using duplex printers the back side of each sheet is printed in the opposite order. **Place multiple publications on same sheet** – this enables different publications to be placed on the same sheet of paper. A new publication will not be printed on the back side of a sheet of another publication. This option will not be available if **BookBill remittance slip** is selected.

Placement details

These settings list the logical pages and on which **side** of the sheet they are placed. It also displays the **offset** for each page which is automatically set in the Layout tab. It is assumed that each logical page is the same size. If you need to change the page size, double-click on the corresponding page and the **Page Details** dialog box enables you to change the settings.

NOTE: When a change is made that causes the Placement details to be regenerated all the offset values are reset. For instance, if the sheet size, width, height or printer mode is changed in the Settings tab.

Example

This displays the layout of the publication's logical pages on the physical sheets after Layout tab options have been set.

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

Postscript

PostScript is a text language and by default, Generate produces the PostScript commands in a format suitable for reading on the production system. If you are generating PostScript output under z/OS this means that the PostScript commands would be in EBCDIC-based text and any referenced resources in ASCII-based text. To avoid having multiple character codes in the same file, the default code page for PostScript is the ASCII-based `Windows1252@ISO Latin1`.

Where this is not the case or in other non-standard scenarios you can override the default setting and indicate that a specific code page be used for the PostScript command text. Additionally, to cope with variations in the way systems interpret non-alpha numeric characters, you can indicate specific code points to be used for the PostScript control characters. Both of these options are defined in output device settings when publishing.

On some platforms you may also want to restrict the size of the PostScript records being generated.

Some external print management systems require a page number to be used within the PostScript page descriptor command. Output device settings allow you to choose between using a job or document page number for this feature.

Postscript DSC comments

DSC comments are typically used to communicate with PostScript document management software. You can add such comments to the output datastream using an action object within your publication logic.

NOTE: See [PostScript DSC](#) on page 458 for details of creating a DSC comment action.

Note that DSCs are not validated by Generate – you must ensure that the comment syntax you enter for the action conforms to the requirements of your document management environment.

Resource embedding

Normally Generate will place resources to be embedded in a PostScript file within the file header or 'Form Space'. The exception to this is EPS images which, by default, are embedded directly in each page where they are required; this is known as inlining. This is done to reduce the memory overhead on the printer which can be significant when EPS files are included in the PostScript file header. However, you can choose to embed some or all EPS files in the Form Space if required.

Do this by running the DOC1RPU utility against the HIP file that controls the application. Use the **MarkForFormSpace** option to specify the resources that are to be included in the header. For further information, refer to the information about RPU in the Production Guide.

Image caching

To cache images:

1. In the navigator browse to and open the required PostScript output device.
2. Click on the **Resource mapping** tab to view the images defined for the current resource map.
3. Select the image you wish to cache and click on the Edit button.
4. Use the **Mappings Details** dialog box to enable the **Cache image** option.

NOTE: You can set all images to be cached or not cached by clicking the Cache All/Cache None buttons on the Images tab.

See [Using resource maps](#) on page 586 for further information on working with resource maps.

Printer resident images

Images that are used in your publication may also be resident on the target printer and can be used from there. The image would then be referenced by name, rather than the actual file being included in the resource pack. This can result in a smaller HIPHIP file.

NOTE: You must ensure that the image is available on the printer in the same format and size. This is outside the scope of the product.

1. In the navigator browse to and open the required PostScript output device.
2. Click on the **Resource mapping** tab, select the image that you want to use from the printer and click on the Edit button.
3. Use the **Mappings Details** dialog box to **Map to a printer resident image**.

Output device options for PostScript

Output device options for PostScript

The **Settings** tab contains options for the PostScript output datastream being produced.

General

Resolution

This is set at 72 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.

Printer carriage control

By default, the default printer carriage control (PCC) byte is not encoded. Use the drop-down list to select a coding system to be used for the PCC if required.

Format of logical records

This setting allows you to customize the structure of the logical records that make up the datastream. By clicking the ... button and using the **Predefined** option of the **Format** dialog box you can select from a range of keywords that format for a range of known scenarios. Or, by using the **Custom** option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.

Record length

Specify a maximum record length for the PostScript commands in the output datastream. This should be an integer in the range 64-255.

Page label

Selecting JOB means that the current page number within the complete output stream will be used as the parameter in the PostScript page descriptor command (%%Page). This is the default setting. Selecting DOCUMENT means that the current page number within the active document will be used as the parameter.

Customizations

The default is **None**, while selecting Truepress Jet520 makes the PostScript output compatible with that specific printer.

Output device options for PostScript

Printer resident path	The location of resources on a Truepress Jet520 (if this printer is specified under the Customizations section above).
Output codepage	If the default formatting of the PostScript commands is unsuitable for your environment select a specific code page to which PostScript will be translated before being output.
Hex Encode output strings	When the codepage of the text to be output does not match the configuration setting (see above) the text in the PostScript output is hex encoded by default, for example where non-Western languages are being output and Windows 1252 PostScript commands are being generated. You can use this option to override these defaults and force Generate always to output regular text. This may be required where a post-composition process needs to manipulate the output, but note that it is the users responsibility to ensure that this option is suitable for use with the production environment.
Media	
Default input tray	When working with printed media this setting specifies which input tray will be used to provide paper when the datastream is being printed until a different tray command is received. Note that the input tray and output bin being used can be dynamically modified during job processing. See Controlling the output environment on page 454 for details. Available trays are defined using the Media tab. The default is Continuous Feed which means either no input trays are available or that the default tray should be used.
Default output bin	As above but relates to the bin to receive the pages produced by the output datastream (where applicable).
Color	
Support	Use these options to modify the color commands for text and shapes used within the datastream. 24bit RGB – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color. 32bit CMYK – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.

Output device options for PostScript

Graphics

Graphic manipulation quality

This will be used for graphics features such as rotated graphics and plug in charts as their use affects both the time taken to generate and the size of the output file, which can be considerably increased. **Default** uses the resolution as specified for images. **Low** gives the best speed for generating but the lowest graphics quality (96 dpi). **Medium** graphics are 240 dpi. **High** will take much longer to generate but will be a reasonable quality of graphic (300 dpi).

Image

Resolution

Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a Dpi – dots per inch value. The default option of 0 (zero) indicates that the image resolution should use the general **Resolution** option.

Do not embed

Select this option if the required images are already available to the printer/browser and you do not wish to embed them in the output data stream where they are referenced.

Embed

Embeds images into each document within the output datastream where they are referenced. If the required images are already available to the printer/browser you may want to use **Do Not Embed** to reduce the size of the datastream.

Minimize embed

Use this option to ensure that only the images used are included in the Generate output file.

Allow scaling up/down

These options determine what happens if the resolution of images used in the publication design does not match the **Resolution** or (Image) **Resolution** settings. If checked, Designer will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).

Preserve color space

This applies to TIFF and JPEG images only. The default is off – which will convert images to BMP with RGB color space. Selecting the option will convert images to EPS2 and retain the original color space, e.g. CMYK, RGB.

Font

Output device options for PostScript

Output font format	This option is used to specify which font format you want use for the production job when it is sent to the PostScript printer. Automatic – fonts are converted to Type42 fonts. Type1 – fonts are converted to Type1 fonts
Do not embed	Select this option if the required fonts are already available to the printer/browser and you do not wish to embed them in the output data stream where they are referenced.
Embed	Embeds fonts into each document within the output datastream where they are referenced. If the required fonts are already available to the printer/browser you may want to use Do Not Embed to reduce the size of the datastream.
Subsetting	Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by Generate, particularly when DBCS" fonts are used Note that subset fonts are not suitable for use with PCE or other post-processing tools.
EngageOne	
Enable as EngageOne device	This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).
Media	
Input tray	Enter the necessary Job DSC and Reference DSC for the input tray by double-clicking the tray to invoke the Edit Mapping dialog box.
Output bin	Enter the necessary Job DSC and Reference DSC for the output bin by double-clicking the bin to invoke the Edit Mapping dialog box.
Layout	
The Layout tab enables you to configure settings necessary for placing multiple logical pages on a physical sheet of paper. The layout settings are dependent on knowing the size of the physical page, so if the output device's Sheet Size option is set to AUTO the layout options are disabled.	

Output device options for PostScript

Sheet size

By default, the size of the physical sheets of paper used on the printer is assumed to be the same as the logical pages within your document designs. This is the **AUTO** option. Where this is not the case use this setting to specify the actual sheet size being used. You can choose from a range of standard sizes or select **CUSTOM** and enter specific dimensions.

Duplex processing

This setting determines if the datastream should be presented in simplex (**single-sided**) or duplex (**double-sided**) mode. Note that the duplex options can be modified dynamically during job processing. See [Controlling the output environment](#) on page 454 for more information.

Pages per sheet

Select how many logical pages from your publication you want to output per side of sheet. The **Pages across** and **Pages down** options control the total number of pages for each sheet.

Book Bill remittance slip – this option enables you to include a remittance slip with each publication. The remittance slip is added by Generate to the end of the first sheet of the publication and does not affect the Sheet Size setting or any other output device option.

Note that for the BookBill option to function correctly you must include a remittance slip specific document in your publication. The page orientation for the document must be landscape and the page size cannot be larger than that specified by the Slip width and height settings. See [Using multiple documents](#) on page 160 for more details.

Placement order

This setting controls the flow of the page placement on each sheet. The placement options are:

- **Pages flow left to right** or **Pages flow right then left**
- **Pages flow across and then down** or **Pages flow down and then across**

To ensure the publication prints correctly when using duplex printers the back side of each sheet is printed in the opposite order.

Place multiple publications on same sheet – this enables different publications to be placed on the same sheet of paper. A new publication will not be printed on the back side of a sheet of another publication. This option will not be available if **BookBill remittance slip** is selected.

Output device options for PostScript

Placement details

These settings list the logical pages and on which **side** of the sheet they are placed. It also displays the **offset** for each page which is automatically set in the Layout tab. It is assumed that each logical page is the same size. If you need to change the page size, double-click on the corresponding page and the **Page Details** dialog box enables you to change the settings.

Note that when a change is made that causes the Placement details to be regenerated all the offset values are reset. For instance, if the sheet size, width, height or printer mode is changed in the Settings tab.

Example

This displays the layout of the publication's logical pages on the physical sheets after Layout tab options have been set.

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

PDF

Generate produces PDF that can be viewed in a PDF browser either as one file containing all documents generated by the application, or as a series of PDF files each containing a single document. You can also produce PDF as an output datastream which can be passed directly to the Document Repository in a Vault Server environment, or for processing by the DOC1DIME utility for e-mail distribution. The Document processing output device option determines the type of PDF output produced.

PDF documents have a number of identification attributes that can be applied such as password, e-mail address, title, author, subject and so on. They can also have permission attributes that determine what actions the reader is allowed to carry out with regard to a particular document such as the ability to print or edit. In the Designer environment these are defined as attributes of a publication object. Any attribute that is specified is automatically associated with every document produced by the publication in the production environment. Note that in order for permission attributes to be enabled within PDF documents you must also specify an owner password in the appropriate output device settings (see below).

To specify publication attributes: open a publication design and switch to the publication view. Double-click on the publication object to display and amend its attributes.

NOTE: For more information about publication attributes see [Working with publications and documents](#) on page 156

PDF document security

PDF documents produced by Generate can optionally be password protected and encrypted to protect them from unauthorized access. Two types of password can be applied: user and owner.

User passwords protect documents from unauthorized opening in a PDF viewer. When using this feature the passwords to be applied are specified as part of publication attributes. You should resolve the **Password** attribute with a value that provides the required password setting. The password can be varied from document to document if the value used is based on input data or other objects providing dynamic values.

Owner passwords protect the PDF documents from unauthorized modification after they have been created. This option is assigned globally to the documents produced by a publication and is specified in output device settings when it is published. For fully secure documents you can produce a randomized password to prevent any amendments to the PDF documents after they have been created.

Encryption is automatically applied to all documents that have either type of password specified – see [PDF output device settings](#) on page 552.

Additional options in output device settings can prevent various specific activities being carried out against the documents when they are opened in a viewer or editing tool.

Resource handling

By default, text and graphics within PDF output are compressed using the Flate algorithm (as used by Adobe Acrobat). Alternatively you can set the output device settings to use the LZW compression algorithm. Note that the Flate setting typically results in a better compression but requires more processing time. You can also turn off compression altogether, or choose to exclude embedded images from compression if required by selecting the Compress option in the image section.

Controlling file size by adjusting image quality

One of the major factors that can affect the size of PDF output is the quality of image resources. Very often, a significant decrease in file size can be achieved by reducing the overall quality of images included in PDF output.

NOTE: You should consider using JPEG where color fidelity or control of file sizes is important to your application.

The **Optimize output file size** and **Image Quality** options can be used together to convert all images to JPEG at a specified image quality and will normally result in a reduction in output file size.

JPEGs are passed to the output stream unchanged and therefore it is the user's responsibility to make sure that the appropriate compression is applied before the files are used in Designer. This differs from the Image Compress option previously described, which applies a predefined compression algorithm (Flate/LZW) to all embedded images in the output datastream without any control over image quality.

NOTE: that keyed images are not processed by the Optimize output file size settings. You must ensure images of this type are manually optimized.

As PDF output is likely to be distributed electronically it is a common requirement not to embed fonts so that the file size is kept as small as possible. PDF viewers (in common with PostScript printers) will normally have a basic set of fonts available – known as the Base 14 group – and will attempt to match missing fonts using the font references and attributes within the PDF. Use the **Embed (Fonts)** setting as appropriate – see [PDF output device settings](#) on page 552 for details.

E-mail distribution

The DOC1DIME program is used to e-mail documents from within a PDF output datastream using a local e-mail client. It can only be used with PDF output produced with the Compound document processing option.

NOTE: The mailing function of DIME is only available under Windows. It interfaces to the e-mail client using either CMC (Common Messaging Calls) or MAPI (Messaging Application Programming Interface) APIs.

The e-mail addresses to be used must be specified as part of publication attributes. You should resolve the **Email address** attribute with a value that provides the address to be used for each document. Normally this value will need to be based on input data to provide individual addresses for each document.

DOC1DIME uses an initialization file (INI) to specify the output file to be processed, where the output documents are to be created and any other processing options including e-mail settings. The program is then run as a batch job on the production system specifying the INI file as a parameter. other govern. Full information about using DIME including all INI settings, e-mail API details and platform specific guidance is provided in the Production Guide.

PDF output device settings

The **Settings** tab contains options for the PDF output

General

Resolution

This is set at 72 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.

Printer carriage control

By default, the default printer carriage control (PCC) byte is not encoded. Use the drop-down list to select a coding system to be used for the PCC if required.

Format of logical records

This setting allows you to customize the structure of the logical records that make up the datastream. By clicking the ... button and using the **Predefined** option of the **Format** dialog box you can select from a range of keywords that format for a range of known scenarios. Or, by using the **Custom** option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.

Compression

This setting determines how the output is to be compressed. Options are:

FLATE – use the Flate compression algorithm. In most situations this results in a smaller output file than the **Faster** option. This is the default setting.

LZW – use the LZW compression algorithm. In most situations this results in a faster compression than the **Smaller** option.

NONE – do not compress the output.

The **Settings** tab contains options for the PDF output

Document processing	<p>Use this setting to determine the type of the PDF output produced by Generate. Options are:</p> <p>COMPOUND – a single file containing all documents produced by the application. This can be passed to directly to a Vault Server or be processed by the DOC1DIME utility to create individual PDF documents for e-mail distribution. Note that the file produced by this option cannot be viewed directly using a PDF reader.</p> <p>MERGED – a single viewable PDF file containing all documents produced by the application.</p> <p>SPLIT – viewable PDF files created for each document Generated by the application. If you choose this option you will need to specify a filename template for the output produced. Refer Specifying a file template on page 585 for further information.</p>
Create start of document bookmarks	<p>Automatically inserts a bookmark at the start of each document when you use the MERGED option.</p>
PDF/A compatibility	<p>This ensures that the output is compliant with the PDF archiving standard PDF/A-1b.</p> <p>Images in the publication will be converted to the same color space as the Color Support option, when necessary. e.g. if 24bit RGB is selected then a CMYK TIFF image is converted to an RGB JPEG image.</p> <p>Note: that the color space of an external keyed image will not be converted, which may result in incompatible PDF/A output.</p> <p>The Embed Fonts option must be selected; the Compression option is automatically set to NONE and the Document Security option is automatically disabled. When the Document processing option is set to COMPOUND, DOC1DIME must also be run to make the output PDF/A compatible.</p> <p>It is important to note that hyperlinks do not function when this option is enabled.</p>
Translate to 7 bit	<p>If this option is checked any image data and compressed text in the output will be translated to standard ASCII values to ensure the maximum amount of readable data. Otherwise such objects are output as true 8-bit values.</p>

The **Settings** tab contains options for the PDF output

Use Identity-H	<p>If this option is checked and any DBCS / Arabic / Thai / Unicode Design Language is used then all True Type and Open Type Fonts will use PDF CID Font Identity-H encoding instead of default CMap. Typically this would be used when publishing PDF, targeted for PDF viewers that do not support embedded CMap encodings, such as Microsoft Edge.</p> <p>Restriction: This option is not recommended when mixing True Type and Open Type Double byte fonts.</p>
Document security	<p>Allows various security options to be set to protect documents from unauthorized access – see PDF document security on page 549 for more information.</p>
Owner password	<p>If the owner password feature is required use this setting to specify the password string or use the Random keyword to instruct Generate to produce the password using randomized characters. If you specify the password you must enter a 32 character alphanumeric string. If you do not provide an owner password, any permission attributes specified for a publication will be ignored.</p>
Encryption key length	<p>Encryption is enabled on all documents that have either type of password specified. The key length can be 40 (default) or 128-bit. Note that 128-bit encryption is not supported by Acrobat Reader versions earlier than 5.0.</p>
Allow notes	<p>If this option is checked notes may be added to the PDF documents using a suitable PDF tool after they have been created.</p>
Allow modifications	<p>If this option is checked the PDF documents may be modified by a suitable PDF tool after they have been generated.</p>
Allow copying	<p>If this option is checked the contents of the PDF documents may be selected and copied to the clipboard in the viewer.</p>
Allow printing	<p>If this option is checked the PDF documents may be printed from the viewer.</p>
Color	

The **Settings** tab contains options for the PDF output

Support

Use these options to modify the color commands for text and shapes used within the datastream. **24bit RGB** – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color. **32bit CMYK** – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.

Graphics

Graphic manipulation quality

This will be used for graphics features such as rotated graphics and plug in charts as their use affects both the time taken to generate and the size of the output file, which can be considerably increased.

Default uses the resolution as specified for images.

Low gives the best speed for generating but the lowest graphics quality (96 dpi).

Medium graphics are 240 dpi.

High will take much longer to generate but will be a reasonable quality of graphic (300 dpi).

Image

Resolution

Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a Dpi – dots per inch value. The default setting of 0 (zero) indicates that the image resolution should use the general **Resolution** setting.

Embed

By default, images are always embedded in each document within the output datastream where they are referenced. If the required images are already available to the printer/browser you may want to uncheck this option to reduce the size of the datastream

Allow scaling up/down

These options determine what happens if the resolution of images used in the publication design does not match the **Resolution** or (Image) **Resolution** settings. If checked, Designer will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).

The **Settings** tab contains options for the PDF output

Preserve color space	<p>This option allows JPEG images to be included directly in the PDF output and will convert TIFF or EPS images to JPEG.</p> <p>This Preserve color space option may be important if the images used by your application have been created with a CMYK color space rather than RGB.</p> <p>If this option is not checked all images will be converted to BMP format.</p> <p>Note that this option is ignored when the PDF/A compatibility option is selected.</p>
Compress	<p>If this option is unchecked all embedded images are excluded from compression. Note that if 'Compression' is NONE then this option is ignored.</p>
Optimize output file size	<p>Select this option to automatically convert all images to JPEG format. Note that if this setting is not checked all images other than JPEG resources will be converted to bitmap format, unless you are using the Preserve color space option.</p>
Image quality	<p>This option is enabled if the Optimize output file size option has been checked; it is used to control the JPEG quality and thus the final file size. 1 represents lowest image quality at the highest compression and 100, the default setting, represents the best quality at the least effective compression. If you are working with publications that include optimized JPEG image resources you need to consider carefully the effect of using this setting.</p>
Font	<p>Do not embed</p> <p>Select this option if the required fonts are already available to the printer/browser and you do not wish to embed them in the output data stream where they are referenced.</p> <p>Note that this option is not recommended for complex script text, e.g. Arabic/Thai/Unicode.</p>
Embed	<p>Embeds fonts into each document within the output datastream where they are referenced. If the required fonts are already available to the printer/browser you may want to use Do Not Embed to reduce the size of the datastream.</p>

The **Settings** tab contains options for the PDF output

Subsetting	Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by Generate, particularly when DBCS fonts are used.
<hr/>	
EngageOne	
<hr/>	
Enable as EngageOne device	This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).
<hr/>	
EngageOne Accessibility Optimization	<p>Select this option if you are using the optional EngageOne Accessibility feature. This option adds an extra space to the end of Generate PDF BT..ET (text) commands, allowing PDF Accessibility readers to parse individual lines. Generate PDF output produced using this option can also be post-processed using separate PDF transformation tools such as EngageOne Accessibility to produce PDF/UA.</p> <p>Note the following restrictions:</p> <ul style="list-style-type: none"> • The use of Unicode fonts is not advised as there are some incompatibility issues. PDF output using Unicode fonts will be displayed by the PDF reader but may not be consumable by EngageOne Accessibility. • Complex Script and DBCS text, for example, Arabic and Simplified Chinese, is not supported. Hence the output for such text is unaffected when this option is selected.

Layout

The Layout tab enables you to configure settings necessary for placing multiple logical pages on a physical sheet of paper. The layout settings are dependent on knowing the size of the physical page, so if the output device's **Sheet Size** option is set to **AUTO** the layout options are disabled.

Sheet size	By default, the size of the physical sheets of paper used on the printer is assumed to be the same as the logical pages within your document designs. This is the AUTO option. Where this is not the case use this setting to specify the actual sheet size being used. You can choose from a range of standard sizes or select CUSTOM and enter specific dimensions.
------------	---

The **Settings** tab contains options for the PDF output

Duplex processing	<p>This setting determines if the datastream should be presented in simplex (single-sided) or duplex (double-sided) mode. Note that the duplex options can be modified dynamically during job processing. See Controlling the output environment on page 454 for more information.</p>
Pages per sheet	<p>Select how many logical pages from your publication you want to output per side of sheet. The Pages across and Pages down options control the total number of pages for each sheet.</p> <p>Book Bill remittance slip – this option enables you to include a remittance slip with each publication. The remittance slip is added by Generate to the end of the first sheet of the publication and does not affect the Sheet Size setting or any other output device option.</p> <p>NOTE: For the BookBill option to function correctly you must include a remittance slip specific document in your publication. The page orientation for the document must be landscape and the page size cannot be larger than that specified by the Slip width and height settings. See Using multiple documents on page 160 for more details.</p>
Placement order	<p>This setting controls the flow of the page placement on each sheet. The placement options are:</p> <ul style="list-style-type: none"> – Pages flow left to right or Pages flow right then left – Pages flow across and then down or Pages flow down and then across. <p>To ensure the publication prints correctly when using duplex printers the back side of each sheet is printed in the opposite order. Place multiple publications on same sheet – this enables different publications to be placed on the same sheet of paper. A new publication will not be printed on the back side of a sheet of another publication. This option will not be available if BookBill remittance slip is selected.</p>

The **Settings** tab contains options for the PDF output

Placement details

These settings list the logical pages and on which **side** of the sheet they are placed. It also displays the **offset** for each page which is automatically set in the Layout tab. It is assumed that each logical page is the same size. If you need to change the page size, double-click on the corresponding page and the **Page Details** dialog box enables you to change the settings.

NOTE: When a change is made that causes the Placement details to be regenerated all the offset values are reset. For instance, if the sheet size, width, height or printer mode is changed in the Settings tab.

Example

This displays the layout of the publication's logical pages on the physical sheets after Layout tab options have been set.

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

RTF

Generate can produce RTF output which can be viewed and edited using an RTF editor, such as Microsoft Word. The output will be split into separate files for each publication produced by a job. You will need to specify a filename template – used to provide unique names for each file. See [Specifying a file template](#) on page 585 for more information.

The layout of an RTF document may not look exactly as designed in Designer. This is because:

- RTF may be interpreted slightly differently by different editors;
- The RTF format does not support all the features offered by Designer. Where possible, an alternative method of presenting the feature has been used.

Other restrictions to be considered when generating RTF output:

- Inline objects, other than in-line images, are not supported and will cause Generate to stop processing.
- Any graphic with content, for example, text box, header, footer, address block, reserved area, barcode etc. will be converted to a table with a single cell without borders.
- Designer inserts physical page breaks into RTF output where a new page is required. As a result, if you subsequently edit the RTF, the content will not flow automatically between pages.
- Fully justified text will be changed to justified text.
- Character alignment on anything other than a decimal point is ignored.
- The RTF driver is not supported for applications using message applications.
- No security settings will be applied to the generated RTF documents.
- Charts are created as a non-editable image;
- Multiple columns are not supported.
- Fonts using extended character sets (as typically used in China, Japan etc.) are not supported.
- URLs will be converted to plain text.
- Anchored shapes will be placed in the correct position, but the anchor will be changed to a fixed anchor.
- Bullets, paragraph numbering and page numbering will be converted to plain text.
- Output is always ASCII encoded regardless of the platform used for Generate.
- Non-printing characters such as tab characters are not supported.

RTF output device settings

The **Settings** tab contains options for the RTF output

General	
Resolution	This is set at 300 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.
Document processing	Use this setting to determine the type of the PDF output produced by Generate. Options are: MERGED – a single viewable PDF file containing all documents produced by the application. SPLIT – viewable PDF files created for each document generated by the application. If you choose this option you will need to specify a filename template for the output produced. Refer Specifying a file template on page 585 for further information.
Word 97 compatible	RTF that is compatible with Microsoft Word 2000 is generated by default. Setting this option will generate RTF that is compatible with Microsoft Word 97.
Image	
Resolution	Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a Dpi – dots per inch value. The default setting of 0 (zero) indicates that the image resolution should use the general Resolution setting.
EngageOne	
Enable as EngageOne device	This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).

The **Settings** tab contains options for the RTF output

Resource mapping

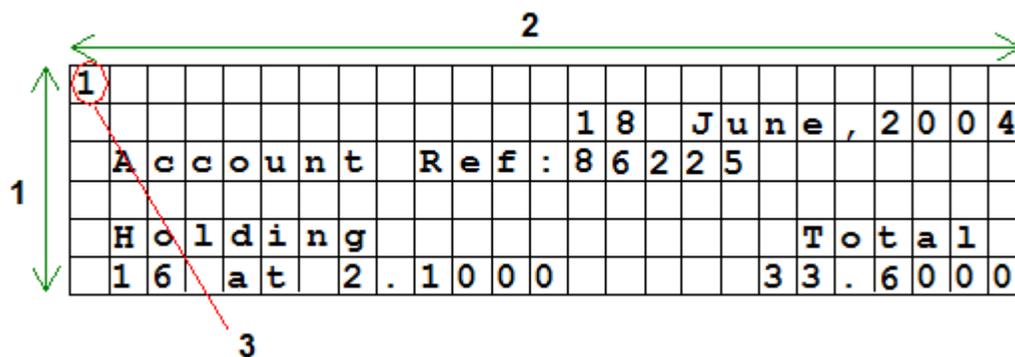
Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

Line Data

Line Data output is a textual representation of output from your application. It is intended for viewing via character based terminals or, as a feed for archiving. Note that graphical objects present in your design will be ignored in output generated by this driver, and that right-to-left text is not supported for Line Data.

Formatting

Output from your application is placed into a predefined grid providing an approximate representation of the original page design in terms of character positioning and justification.



Legend:

1 - grid height

2 - grid width

3 - carriage control information

The dimensions of the presentation grid are set in the publishing phase as part of the output device settings for the Line Data driver. However, where lines exceed the grid width the height setting for the grid is automatically adjusted and extra lines are added to cater for the overrun.

Note: A continuous string of text intended for Line Data output must not exceed 1024 characters.

Printer carriage control

The first column of the grid is set aside for printer carriage control (PCC) information. Note that the only control produced by Generate is the new page command on the first line of data for each page. The format of the PCC byte can be specified to conform to one of the following standard types: ANSI, ASCII, MACHINE or CRLF.

Line Data output device settings

The **Settings** tab contains options for Line Data output being produced.

General

Resolution	This is set at 300 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.
Printer carriage control	By default, the default printer carriage control (PCC) byte is ASCII encoded. Use the drop-down list to select a different coding system to be used for the PCC if required.
Format of logical records	This setting allows you to customize the structure of the logical records that make up the datastream. By clicking the ... button and using the Predefined option of the Format dialog box you can select from a range of keywords that format for a range of known scenarios. Or, by using the Custom option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.
Height of character grid	This value indicates the height of the grid within which characters will be placed (i.e. the number of 'lines' available).
Width of character grid	The width of the grid within which characters will be placed (the number of columns available).
Output codepage	If the default formatting of the Line Data output is unsuitable for your environment use the drop-down list to select an option to indicate a specific code page to which Line Data will be translated before being output.
Force full width characters	If your application uses double byte character sets (DBCS) select this option to ensure that the spacing of characters within the grid is correct.

Configure for EngageOne Deliver

Select this if the output datastream is to be used by EngageOne Deliver. This will:– suppress new page commands from all but the first page of a document (these are normally generated for each page in a document); – include a publication ID after the new page command. Note that setting this option can make the Linedata output unsuitable for other uses.

EngageOne

Enable as EngageOne device

This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

HTML

Generate supports two forms of HTML output.

The HTML client/server format typically supports applications that require interaction between web clients and a server on which pages and associated resources are stored. The output for each publication is created as one or more independent web pages. These pages and their associated image resources are output in an XML structure so they can be identified, extracted and deployed to the server environment as required. Cascading style sheets are used for presentation.

The HTML e-mail format provides a simplified version of HTML and is intended to be included as part of an e-mail communication with the end user. The output for each publication is created as a single HTML page (regardless of any page breaks in the design) and all images are simply referenced using the base name of the imported resource with a user defined URL. The HTML commands do not include cascading style sheets.

For both HTML formats you should bear in mind that only font references are included in the output and not the fonts themselves. We therefore strongly recommend that applications generating HTML use only fonts that are known to be generally available to web browser environments. Consider using a resource map if your Designer application does not use such standard font references (see [Using resource maps](#) on page 586).

NOTE: Unlike other output devices Generate has no direct control over the font resources used for HTML output. Therefore when working with Arabic, Indian dialects or similar languages it is important to note that the browser or e-mail client is responsible for handling such things as inserting ligatures and formatting for right-to-left text.

HTML client/server

The following restrictions apply to Designer applications producing HTML client/server output:

- The use of the Designer barcode feature is not supported
- Rotated text – i.e. text elements presented other than at 0° – is not supported.
- The output is only compatible with browsers supporting cascading style sheets, such as the 4.0 or later versions of Internet Explorer and Netscape Communicator

The output device settings used when publishing are particularly important when HTML is being produced. They allow you to specify:

- Default browser presentation settings, including text colors and background.

NOTE: that the color settings may be overridden by the end user using browser customization options.

- Location references for associated resource files that need to be deployed to a web server. Such references are embedded within the HTML documents and represent the locations where resources will need to be made available.
- Browser optimization options.

The HTML pages produced by a production job are created within a single XML file that also contains any other resources required to present the pages. It can be parsed to extract the pages as individual HTML documents or passed directly to the Document Repository if you are using Vault or Liaison to manage the output.

By default, the DTD describing the XML is stored as an integral part of the file but you can optionally request that it be extracted to a separate file if required. On record based systems (z/OS) you should allow a logical record size of 4k to receive the XML file.

The character set specified within the HTML pages is set to ISO-8859-1 by default which is suitable for most Latin-based text. You can specify a different character set within output device settings but note that the same character set is defined for all pages produced by a single job.

Images used in your design are converted to png format. By default, converted images are identified by their original base name together with a generated identifier. For example, if you have included an image called welcome.bmp it will be converted to welcome_g0001.png. If required, this name can be overridden using the resource mapping facility, refer to [Using resource maps](#) on page 586 for further information.

Output device settings for HTML

Settings

The **Settings** tab contains options for HTML output being produced.

Output device settings for HTML

General

Resolution	This is set at 96 Dpi – dots per inch – and is used where the datastream requires a specific resolution be indicated within the protocol itself or where Generate needs to produce or reference bitmap images as part of the output.
Printer carriage control	By default, the default printer carriage control (PCC) byte is not encoded. Use the drop-down list to select a coding system to be used for the PCC if required.
Format of logical records	This setting allows you to customize the structure of the logical records that make up the datastream. By clicking ... and using the Predefined option of the Format dialog box you can select from a range of keywords that format for a range of known scenarios. Or, by using the Custom option you can program your own logical record structure using formatting codes. These options are detailed in the Production Guide.
Optimize for rendering	The HTML will be optimized for the most efficient download speed such that it renders as quickly as possible on the browser type selected but the file size of pages will be increased.
Document Interchange XML	If checked, the output datastream can be loaded directly into Vault.
URLs	

Output device settings for HTML

Resource

By default, the vector graphic used for lines and boxes are assumed to be available at the same location as the HTML pages themselves once the resources have been extracted from the XML file. If a different location is used this option allows you to specify the URL where the resources are located and will be used to reference them within the pages. The following codes can be used to include attributes from the XML structure as part of the path if required (you can use the same information when extracting files from the XML): &pakUID – unique ID for the current job &groupUID – unique reference to account group &groupID – contains the keyword parameter from the attributes of the first document in the publication (if coded). &docUID – unique reference to the current document &docID – a reference indicating the sequence of the document within the group &pageID – a reference indicating the sequence of the page within the document &jdUID – ID specific to a chart data object if present.

Examples:

`http://pubs/html/resource``http://pubs/html/resource/&pakUID``http://pubs/&pakUID/&docUID/&pageUID`

Image

By default, vendor supplied graphics are assumed to be available at the location indicated by Resource location or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the image files are to be referenced within the HTML pages. The URL syntax is the same as for ResourceURL.

Java

By default, the DOC1 graphics applet required for chart/shape drawing is assumed to be available at the location indicated by Resource location or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the applet is to be referenced within the HTML pages. The URL syntax is the same as for Resource URL.

Output device settings for HTML

Chart data

By default, files containing chart drawing parameters are assumed to be available at the location indicated by Resource location or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the chart data files are to be referenced within the HTML pages. The URL syntax is the same as for Resource URL. If **Locate Data/ Query** is selected, specify a server side query script. For example:

```
http://scripts/getchart.cgi?&jdUID
```

Charts

No applet message

If specified, this message is displayed if charts are used within a page but the DOC1 graphics applet is unavailable to the browser.

Locate data

By default, the Java data location setting is expected to indicate a URL for locating chart/shape data files. Select **QUERY** if you want to locate the files using a server side script (CGI, ASP, PERL, etc.).

Threshold

This option specifies the maximum inline size (in bytes) for parameters related to chart drawing. Where the parameters required for chart drawing fall under this limit they will be included as meta data within the actual HTML pages. Parameter lists exceeding this limit are created as independent objects within the XML file.

Java applet record length

By default, the record length for data to be sent to the DOC1 graphics applet used for chart/shape drawing is set to 128 bytes. If required use this option to define a different record length.

Styles / Schemas

Style sheet

By default the style sheet is embedded in the HTML page. Selecting EXTERNAL means that the HTML page references an external file for its style settings.

Comment out style sheet tags

Check this option if you wish to comment out style sheets tags. This may be required if your application specifically needs to support older versions of some web browsers. Note that by default, style sheets are not commented out.

Output device settings for HTML

CSSP	<p>Specifies the method to be used for positioning objects within the HTML. This can have a significant effect on download or presentation efficiency. Options are:</p> <p>DIV – use HTML DIV tag. The line breaks used with DIV tags will minimize the file size and thus the bandwidth required to download it. The client browser may take longer to render the page compared to where the SPAN option is used. This option is set as default.</p> <p>SPAN – use the HTML SPAN tag. This will optimize the commands so that they can be rendered by the client browser as quickly as possible. The file size will be bigger by two characters per positioning object required by the page as compared to using the DIV option.</p>
Inline CSSP	<p>By default, the CSSP structure is placed inline within the HTML. Uncheck this option to have it embedded within a style sheet.</p>
External DTD	<p>If unchecked, the DTD that describes the HTML PAK file remains part of the PAK file. In this scenario DOC1EDU is expected to perform the HTML extraction process. If checked, the DTD is externalized to the location specified in the Dtd File keyword.</p>
DtdFile	<p>This setting must be specified if the External DTD option is checked. Specify a path and file name that will receive the DTD.</p>
DtdReference	<p>By default, the HTML PAK file will reference its externalized DTD at the location indicated by DtdFile. If you need to relocate the DTD during deployment – e.g where OS file naming conventions prevent a valid reference or where the web server is not directly available to the system running Generate. You can use this keyword to specify the location to be used to reference the DTD within the PAK file.</p>
Color	
Background	<p>This is the color to be used for the background of all pages when displayed in a browser. The default color is white. If Background image is also specified the background color will only be used while the image is loading.</p>

Output device settings for HTML

Link	This is the color to be used to highlight all hyperlinks. The default color is blue. IMPORTANT: this setting will override the colors used for such links within document designs.
Active link	This is the color used to highlight active hyperlinks. The default setting is red. Note that this setting may be overridden in some browsers.
Visited	This is the color used to highlight hyperlinks that have been visited. The default color is gray. Note that this setting may be overridden in some browsers.
Shade using white	Presentation objects that have a shading value of less than 100% use the HTML background color in conjunction with the object color to produce the shade effect (sometimes known as 'true color'). Checking this option means that the background will be treated as 'paper' and white will be used in conjunction with the object color to produce the shade effect.
Image	
Resolution	Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a Dpi – dots per inch value. The default setting of 0 (zero) indicates that the image resolution should use the general Resolution setting.
Background	This is a URL indicating an image to be used as the background for all pages. No image is used by default. For example: <code>http://pubs/webres/logo.jpg</code>
Path	If coded, defines the location of graphics to be read in at the end of the job and embedded in the output datastream.
Allow scaling up/down	These options determine what happens if the resolution of images used in the publication design does not match the Resolution or (Image) Resolution settings. If checked, Designer will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).

Output device settings for HTML

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

Working with the XML file

The output from a production job generating HTML output is an XML file that contains both the pages and any associated resources. If the output is to be managed by Vault or Liaison you can pass the XML directly to the Document Repository without modification. Otherwise you will need to use a method of extracting the material from the XML into individual files before it can be browsed.

If you intend to use the HTML in a web server environment you will need to parse the XML file, identify the relevant resources and extract them to the appropriate server locations. Files other than the actual pages will need to be made available at the resource locations specified within the HTML itself (and which can be customized using output device settings). Detailed information about the XML file format and guidelines for creating an extract routine can be found in the Production Guide.

If you simply want to extract the files from the XML for local browsing or onward processing you can use the EDU (Extract and Deploy Utility) program which is provided as part of the product distribution material in the production environment.

The DOC1EDU program uses an initialization file (INI) to identify the XML input file and to specify the locations to be used for the output files. An index file can be created which provides information about the document and page sequence which produced the HTML and cross-references for resource files.

Full information about using EDU including all INI settings and platform specific guidance is provided in the Production Guide. However, the following is a simple example of how you might configure and run EDU to extract HTML pages and resources on a Windows host.

Simple EDU example

Create an INI file:

```
<Files>;HTML output from Generate (XML)Input=pubhtml.xml;Location to
receive HTML pagesPageFileMask=\html\%1.htm;Location to receive resource
filesResourceFileMask=\htmlres\%1;File to receive index
fileIndexFile=\pubs\htmlindx.txt
```

- <Files> is a required section name.
- %1 is a place holder for the file names that will be automatically created by EDU. These will sequentially number the page and resource files in the order they are created.
- Comment lines start with a semi-colon.
- Save the new file as `extract.ini` (in this example).

Run the DOC1EDU utility:

```
doc1edu ini=extract.ini
```

HTML for e-mail

The HTML for e-mail (eHTML) driver provides a simplified version of HTML which is primarily intended to be incorporated into e-mail for desktop and mobile devices.

The output for each publication is created as a single HTML page (regardless of any page breaks in the design). These can be output into one single compound file or, as individual files according to a path template. See [Specifying a file template](#) on page 585 for more information.

Layout options

The eHTML driver provides options for Fixed and Fluid layout in your device's reading pane.

The **Fixed** option presents page content positioned as defined in the template design. If you choose this option, horizontal/vertical scrollbars are presented as necessary to allow the user to view the page content as it was originally designed.

The **Fluid** option will re-flow page content according to the size of your device's reading pane.

A Legacy layout option is also provided to cater for eHTML output designed before the availability of the **Fixed** and **Fluid** layout options.

Note that the Fixed and Fluid layout options provide greatly enhanced support of paragraph and table content and should be used in preference to the Legacy option.

When designing an application suitable for HTML for e-mail it is recommended that the editor page is defined to have a height large enough to accommodate all expected output on a single page.

Images other than those created for barcodes are assumed to be server resident and are simply referenced within the HTML using the base name of the imported resource file at a user defined URL. At publishing time image resources are not created for this device.

Barcodes are supported for this device and rendered as GIF images output as independent files according to a further template path name. Unlike other images the barcode GIFs are assumed to require physical packaging with the HTML itself as part of the e-mail system.

Once a production job has completed it is the user's responsibility to check the validity of any links in the HTML pages, package barcode GIFs with the pages if required and then actually include the pages in the appropriate e-mail messages. Generate itself does not support direct e-mailing for this type of output.

Previewing your eHTML output

Previewing allows you to run a publication through the production engine within the Designer itself without the need to go through the formal publication and production process. The preview can either be run for a selected publication in the navigator, or for a particular publication in the editor. Note that the last saved version of the selected publication will be previewed from the navigator. However, if you invoke the preview for a publication within the editor, any modifications you may have made to the publication will be included in the preview.

The preview process uses settings for a previously configured eHTML output device; you will therefore need to ensure that at least one output device has been configured for your project. In a scenario where multiple output devices are available, you will be prompted to select the output device to be used as the basis for the preview.

To run the eHTML preview:

1. In the navigator browse to and select the publication you want to preview and on the **Tasks** menu select **Preview for eHTML**.
2. Complete data mapping if required. If the publication has unresolved data aliases you will be asked to update or create a data map that supplies the appropriate linkages. See “About data mapping” for details.
3. Review output. The eHTML output datastream is composed and HTML links are presented in a separate window. Use the browse button if you wish to use different sample data for a subsequent preview.

eHTML restrictions

The table that follows details the restrictions you need to be aware of when using the three supported layout options.

Designer feature	Legacy	Fixed	Fluid	Comments/recommendations where supported
Page Size				Create a page size large enough to accommodate all content in your document design.
Page Margins				<p>In a scenario where your document design runs over multiple pages, page margins are handled in Fixed and Fluid layouts as follows:</p> <ul style="list-style-type: none"> • The top of the eHTML output will use the top margin of the first page in the document design. • The bottom of the eHTML output will use the bottom margin of the last page of the document design. • Inter-page margins are ignored. • Left and right margins set in the document’s design are honored.
Background				Background content is presented. The use of the page setup to place presentation objects in background is not recommended as placement cannot be guaranteed.

Designer feature	Legacy	Fixed	Fluid	Comments/recommendations where supported
Header				When working with Fixed and Fluid layout options, header content is ignored and not included in the output datastream. Note: inconsistent header placement will occur in Legacy layout mode.
Footer				When working with Fixed and Fluid layout options, footer content is ignored and not included in the output datastream. Note: inconsistent footer placement will occur in Legacy layout mode.
Page Breaks				
Page orientation				
Columns				
Lines				Table borders can be used as a workaround.
Shapes				
Text Boxes				Layout issues may be observed in older versions of Outlook when text boxes are placed side by side. Note that although this feature can be used with Legacy layouts, it is not fully supported.
Internal Margins				
External Margins				

Designer feature	Legacy	Fixed	Fluid	Comments/recommendations where supported
Tables				
Internal Margins				
External Margins				
Spacing Before				
Spacing After				
Line thicknesses				
Rounded Corners				The recommendation is to use images placed at the appropriate positions in the publication design to provide the required effect.
Fonts	Restrictions remain in force.	No restriction	No restriction	Refer to the section that follows for details
Paragraph Formatting				For the most predictable results across the majority of e-mail clients it is recommended to keep to the default paragraph settings. However, for Fixed and Fluid layout options paragraph and line spacing settings are supported and should be applied as required. Note that the hanging indent feature is not widely supported across e-mail clients.
Justified Alignment				Fully justified formatting is not supported.
Tab stops				For simple use, not all e-mail clients support this feature, spacing may be unpredictable.

Designer feature	Legacy	Fixed	Fluid	Comments/recommendations where supported
Rotated text				Although this feature is supported with Fixed and Fluid layouts, it is not recommended. Viewing issues may be encountered particularly with mobile devices.
Bullets lists				For simple use as not all e-mail clients will display these correctly. The space between the bullet character and text may be unpredictable if many levels are used. If multiple levels are required, the use of static tables is recommended.
Numbered lists				As for bulleted lists.
Indents				
Line Spacing				Results may be variable across e-mail clients. Note that care should be taken when using large font point size changes in multi-line text as this can result in overlapping text in the HTML output. The best practice in this scenario is to keep font point sizes the same or very close in each paragraph and use exact line spacing in the paragraph to ensure the output reflects your design. If large font size changes are required then tables should be used to ensure each line has the necessary spacing.
Widow /Orphan				Not required, pagination is not applicable in e-mail output.
Charts Native				
Charts Plug-in				
Anchored positioning				
Fixed positioning				

Designer feature	Legacy	Fixed	Fluid	Comments/recommendations where supported
------------------	--------	-------	-------	--

Inline positioning				
--------------------	---	---	---	--



Points to note when using the Legacy layout option to generate eHTML:

- The <Title> element for HTML pages is provided by the Title option found in the Attributes tab of the **Publication** dialog box. See [Working with publications and documents](#) on page 156 for more details.
- When your document design includes tables a separate table is created in the HTML output for each table row in your document. Line thickness of all such tables is set according to the properties of the first cell in the design table.
- When limited to HTML v3.2 fonts are referenced by size only. Without this restriction **font-family** and **font-face** attributes will be included. Font sizes are mapped according to the following table:

Designer font size	HTML font size
<8 pt	1 xx-small
8<10 pt	2 x-small
10<12 pt	3 small
12<14 pt	4 medium
14<18 pt	5 large
18<24 pt	6 x-large
24 pt +	7 xx-large

Output device options for eHTML

Settings

The **Settings** tab contains options for e-HTML output being produced.

Output device options for eHTML

General

Resolution This is set at 96 Dpi – dots per inch. This setting is not editable and is included for information only.

Barcode file template This allows you to define a base file name to be used for barcode GIF images. It must be specified if your publication uses barcodes and must be in a format suitable for the host platform. Two parameters are available to help make the filename unique – %1 is the generated filename and %2 is an index number.

For example:

Publication output filename = Out%1.htm Barcode file template = Bar%1-%2.gif

will produce the following GIF image names:

BarOut0000001.htm-001.gif BarOut0000001.htm-002.gif BarOut0000002.htm-001.gif BarOut0000002.htm-002.gif

Layout Choose the type of e-mail HTML you would like to generate. Options are: **Fluid**, **Fixed** or **Legacy**.

Fluid - the page content will automatically flow according to the size of reading pane.

Fixed - the page content is of a fixed size as defined in your template design. When the page is re-sized horizontal/vertical scroll bars are presented as necessary to allow the user to view the page content as it was originally designed.

Legacy - this option is provides legacy support for the eHTML driver prior to the implementation of Fixed and Fluid options.

Note that by default when creating a new eHTML output device, the **Layout** option is set to **Fixed**.

Compound file for EngageOne Deliver Check this option to generate a single file containing ALL documents. This makes it more efficient for processing by EngageOne Deliver. Note that a compound file is not supported by HTML browsers.

Legacy

Note that the following options are only visible if you have chosen the **Legacy** layout option.

Output device options for eHTML

Use table to position output

This forces each page to be output in an HTML table element, which preserves the designed width of the page. The table itself will not be visible.

Alignment of table (where used)

If a table is used to position the output, this allows you to specify its alignment. The effect of this setting will only be apparent if the window used to view the output is re-sized beyond the width of the table. This option would normally be used where the direction of text is right to left, to preserve an overall look. Default is **left**.

Generate HTML v3.2

Selecting this option generates output that strictly adheres to the standards for HTML version 3.2 as specified by the World Wide Web Consortium (W3C). Use this option to ensure that your output will be displayed correctly in e-mail clients, such as Lotus Notes and Microsoft Outlook. If you do not check this option then the HTML output will conform to HTML version 4.01. Note that it is not recommended that this option is used when a publication contains Arabic text, as HTML v3.2 does not fully support right-to-left text. For more information on HTML version 3.2, see <http://www.w3.org/TR/REC-html32>.

Fluid

Reduce image size to fit

When selected, images are reduced in size in line with the display used to view the page. Note, this frame is only visible when the Fluid layout option is used.

URLs

Image

Use this to specify a URL where image resource files are to be referenced within the HTML pages.

Background image

Use this to specify a URL that indicates an image to be used as background for the publication in the browser. No image is used by default. The image is either placed repeatedly as necessary (i.e. tiled) in the body section of the output or, when the **Use table to position output** option is used, is placed once in the HTML table element of the output. Note that the background image will not be shown in the Designer. For example: <http://pubs/webres/logo.jpg>

Output device options for eHTML

Generate barcode

Check this option to output barcode images from your application to the destination specified in the **Barcode Image URL** setting. If this option is not selected, references to barcode images are placed in the HTML page included in an e-mail with the 'cid:' command, using the Multipurpose Internet Mailing Extensions (MIME) standard.

Barcode image

This is only applicable when the **Generate Barcode** for URL option is selected. By default, barcode images will be placed in the same location as the HTML output files and are referenced within the HTML pages using the MIME standard. When selected barcode images will be placed in the location specified for embedding in the HTML pages. For example:

`http://pubs/html/resources//g1abc/output/bc/Images/`

Color

Background

This is the color to be used for the background of all pages when displayed in a browser. The default color is white. If **Background image** is also specified the background color will only be used while the image is loading.

Link

This is the color used to highlight all hyperlinks. The default color is blue. **IMPORTANT:** this setting will override the colors used for such links within document designs.

Active link

This is the color used to highlight active hyperlinks. The default color is red. Note that this setting may be overridden in some browsers.

Visited

This is the color used to highlight hyperlinks that have been visited. The default color is gray. Note that this setting may be overridden in some browsers.

Graphics

Graphic manipulation quality

This will be used for graphics features such as rotated graphics and plug in charts as their use affects both the time taken to generate and the size of the output file, which can be considerably increased. **Default** uses the resolution as specified for images. **Low** gives the best speed for generating but the lowest graphics quality (96 dpi). **Medium** graphics are 240 dpi. **High** will take much longer to generate but will be a reasonable quality of graphic (300 dpi).

Output device options for eHTML

Graphic file template

This allows you to define a base file name to be used for graphics JPG images. It must be specified if your publication uses graphics features such as rotated graphics and plug in charts and must be in a format suitable for the host platform. Two parameters are available to help make the filename unique – %1 is the generated filename and %2 is an index number.

For example:

```
Publication output filename = Out%1.htm
Graphic file template = Pic%1-%2.jpg
```

will produce the following JPG image names:

```
PicOut0000001.htm-001.jpg PicOut0000001.htm-002.jpg
PicOut0000002.htm-001.jpg PicOut0000002.htm-002.jpg
```

Generate graphic for URL

Check this option to output graphics features such as rotated graphics and plug in charts from your application to the destination specified in the **Graphic Image URL** setting. If this option is not selected then references to these graphics are placed in the HTML page included in an e-mail with the 'cid:' command, using the Multipurpose Internet Mailing Extensions (MIME) standard.

Graphic image URL

This is only applicable when the **Generate graphic for URL** option is selected. The graphics images will be placed in the location specified for embedding in the HTML pages. For example:

```
http://pubs/html/resources///g1abc/output/pics/Images/
```

EngageOne

Enable as EngageOne device

This must be selected if resources – fonts and images – are to be created when publishing for EngageOne (templates, Active Content and resources).

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see [Using resource maps](#) on page 586.

Specifying a file template

Some Generate output formats can create documents as individual files rather than as part of a datastream. This applies to:

- PDF output using the Split option,
- RTF output
- HTML for e-mail output.

In these situations each file must be assigned a unique name. This is achieved by embedding a variable in the appropriate output file name assignment using the Publish Wizard. During production Generate will automatically resolve the variable with a unique value for each file produced.

Choose from one of following variables when making file name assignments:

- %1 – a 7 digit sequential hex number based on order of processing.
- %2 – a 7 digit sequential decimal number based on order of processing.
- %3 – the value in the Title field which is specified as an attribute of the publication (see [Working with publications and documents](#) on page 156). If Title is missing, %1 will be substituted.

NOTE: It is important to ensure that the resultant file names are compliant with naming conventions of your host operating system.

z/OS example:

```
'G1UKJB.TSTPCOM.PDF (D%2) '
```

may produce:

```
G1UKJB.TSTPCOM.PDF (D0000001) G1UKJB.TSTPCOM.PDF (D0000002) G1UKJB.TSTPCOM.PDF (D0000003)...
```

Windows example:

```
TSTPCOM%3.RTF
```

may produce:

```
TSTPCOM1012831.RTFTSTPCOM5598179.RTF...
```

In the example above the Title publication attribute has been set to customer account number.

Using resource maps

A resource map allows you to override the default names given to the font and image resources created by a publishing task (and the references used to identify the resources within an output datastream). These resources are added to the Designer HIP file that controls a publication in the production environment.

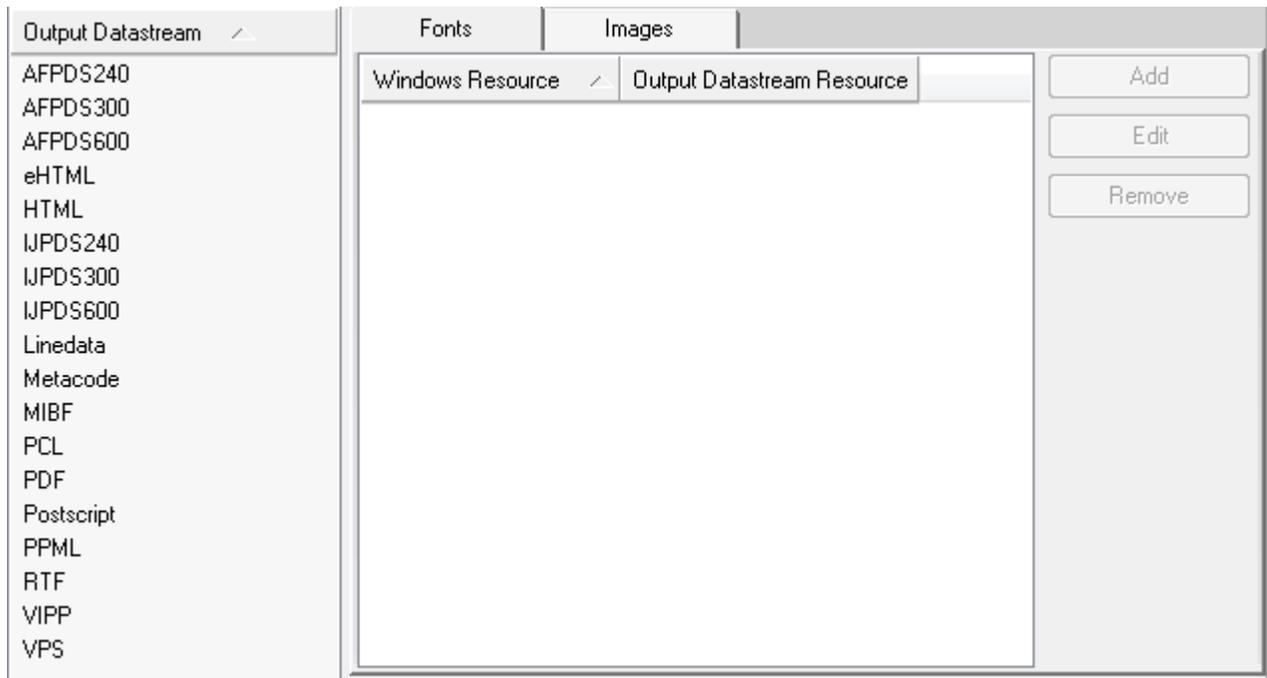
You may want to use a resource map to make the resource names more identifiable or if you want to use existing resources in your output datastream environment rather than those created by Designer.

NOTE: By default, font and image resources produced by Generate are embedded in the output datastream created by a publication. If you want to reference existing external resources you must disable the embed fonts or embed images options for the appropriate output datastream objects.

A Default resource map is always included in the repository and contains references to the fonts and images used in any publication that has been published. This cannot be edited but you can override the default mappings by creating a Custom resource map or by using the User Defined Resource Map, which contains override mappings for existing publications. Override resource maps are used in conjunction with the Default map when you publish.

To create a custom resource map:

1. In the navigator browse to the required project or folder and on the **File** menu select **New/Resource Map**.
2. Enter a name and optional description and click **OK**. The resource map editor window is opened.
3. Create new mapping entries as required.

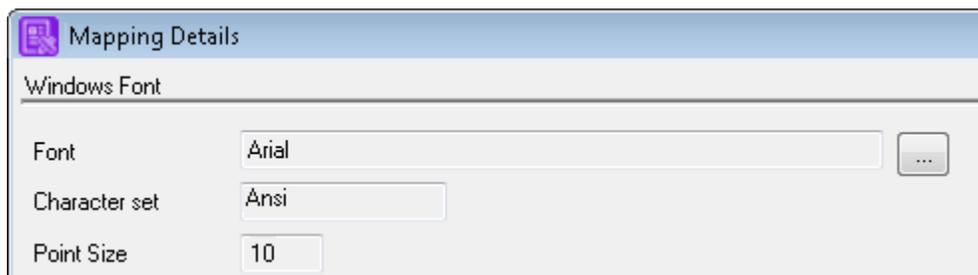


- Click on the output datastream for which the resource mapping is intended
- Select **Fonts** or **Images** as required. Note that AFP also has **Bitmap**, **Outline** and **DOF** font tabs.
- Click **Add** to create a new mapping.

To use font mappings:

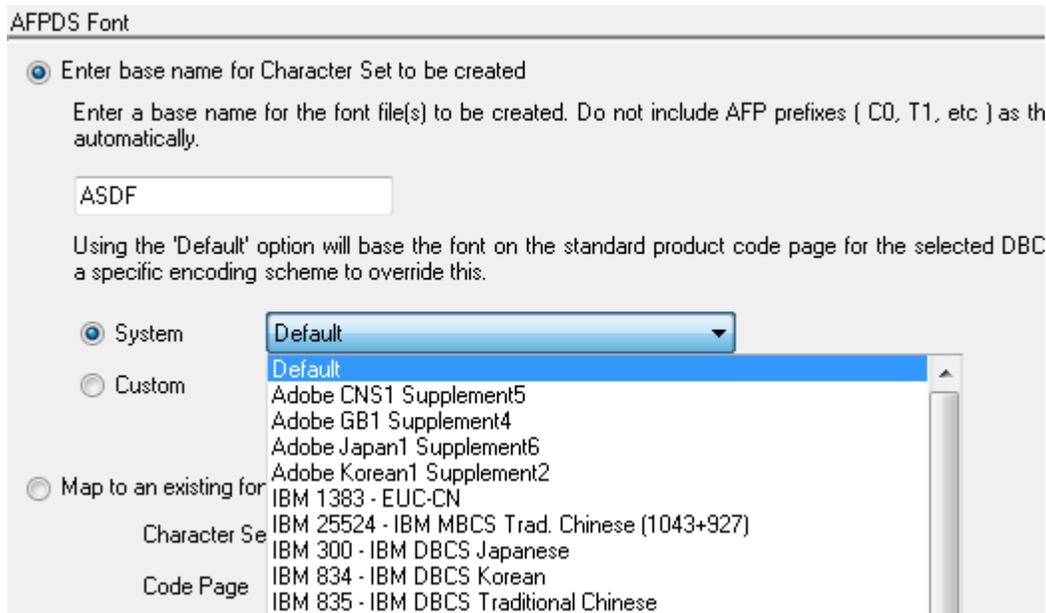
use the **Mapping Details** dialog box for **Fonts** to specify either a new resource name or a cross reference to an output datastream resource known to the repository. The resulting dialog box depends on the selected output datastream. Note that for AFP and PostScript you have the option to reference fonts from your printer environment.

For AFP bitmaps, each combination of font-face, point size and character set counts as an individual font that can be mapped. You should open the appropriate font folder and select the specific point size you want to map.



Bitmap AFP font encoding can be changed from the default settings to indicate the source character set or encoding from which the AFP font is to be generated. This option is used to

change the source character set for creating fonts of non western languages, such as Chinese, Japanese and Korean. Only applicable encoding schemes supporting these languages are available for selection. Single-byte code pages cannot be selected.



- This section has slight variations for **Bitmap**, **Outline** and **DOF** fonts.
- Only 'double-byte' and 'multi-byte' character sets are available for selection.

Outline AFP font encoding can be changed from the default settings to indicate the encoding of the AFP font to be created. This option is used with single-byte fonts used for western languages. Only applicable encoding schemes supporting these languages are available for selection. Fonts used with non western languages, such as Chinese, Japanese and Korean are generated using the UCS2 encoding scheme. For complex scripts, fonts are created based on the original Windows fonts. Note that the encoding scheme for DOF AFP outline fonts is set to UCS2 and cannot be changed.

Existing AFP bitmap and outline fonts are a combination of up to three file types: – a character set containing the actual font bitmaps;– a code page that maps code points to characters;– a coded font that references a character set and a code page.

It is possible to reference a font using a coded font, a character set/code page combination, an outline font family or a printer resident font directly. If you specify a new name for a font it will be used as the base name for all required files. Note that for bitmap and outline fonts, the name can be no more than six characters.

If you want to **Map to a ... font** you should specify either a coded font or character set/code page combination, or an outline font family. Note that for AFP outline fonts you can only map to fonts that have previously been imported – see [Using fonts and images](#) on page 141 for information on importing fonts. Note that if you select a double-byte font you can only map to a coded font.

Windows Font

Font: Arial

Character Set: Ansi

Output Datastream Font

Automatically generate an Output Device font called:

BookT

Map to an existing font:

For PostScript, PDF and variants all point sizes of a font-face are mapped together. You need only select the appropriate font folder and character set when mapping. For certain types of PostScript multi-byte and double-byte fonts you can optionally specify a mapping file known as a CMAP. The CMAP file name is entered directly after the font details in the **Printer Font** field and must be preceded with two hyphens, as follows: **Mincho-Light--90ms-RKSJ-HA**. In this example 90ms-RKSJ-HA is the CMAP file associated with the Mincho-Light font.

To use image mappings: from the resource map editor window, click on the **Images** tab and use the **Mapping Details** dialog box to select the image to be mapped. You can specify either a new resource name or a cross reference to an output datastream resource known to the repository.

Output Datastream Image

Automatically generate an Output Datastream image called:

Map to an existing image:

Map to a printer resident image:

Details

Type: _____

Resolution: _____

Color Depth: _____

Cache Image in Printer Memory

Cache Image

- The Map to a printer resident image option can only be used where the target output device supports printer resident images, such as Truepress Jet520.

- Use the Cache Image option for multi-use PostScript images. Beware of memory limitations if you select for multiple images

Note that PostScript images can be cached in printer memory as required to increase performance and reduce the size of the output datastream. See [Postscript](#) on page 541 for further information.

To delete entries from a resource map: open the appropriate custom resource map in the resource map editor. Select the mapping to be deleted and click **Remove**.

To edit an override resource map entry:

1. Double-click the resource map object you want to edit and it opens in the resource map editor.
2. Select an output datastream and the mapping that you want to change and click **Edit**.
3. Use the **Mappings Details** dialog box to specify alternate attributes for the resource mapping.

To associate a resource map with an output device:

1. Open an Output Device object and select **Resource Mapping**.

Note: Select the **Resource Mapping** tab to access the **Mapping** field

Note: You can access an output device from the navigator, a Production Job or the Publish Wizard.

2. Click the Browse (...) button on the **Mapping** field and the **Select Resource Map** dialog box is displayed.
3. Choose the required resource map and click **Open**.

38 - About publishing and deployment

Running a publishing task in the Designer prepares the material required for Generate to process a production job. You can either publish for Generate or for the EngageOne Server environment.

In this section

Publishing and deployment.....592



Publishing and deployment

Publishing for Generate

A production job is made up of one or more publications that share the same input data file. You can publish the complete publication, which brings together the selected publications and creates a single HIP file containing the publication design and resources for use with Generate. In certain circumstances you may want to publish the resources and publication design separately; for example, in a large scale production environment this could save time if you just need to update the resources. You can produce any number of separate resource HIP files for use with Generate.

Publishing for EngageOne

In the EngageOne Interactive environment, a template package is created from one publication. This comprises a manifest that describes all the files that make up the template. See [How to publish for EngageOne](#) on page 405 for details.

Publishing Active Content

Active Content can be published independently and then used dynamically to update one or more publications that are already in production. This may be useful where you have several publication designs that reference the same Active Content object and where frequent updates to the content are required. However, you should exercise care when using this option as changes made to Active Content may have a significant impact on production jobs. Note that when publishing for Generate, the use of extra fonts or images will require the resources to be regenerated. This includes the use of additional point sizes of fonts already used and new keyed image maps.

Publishing for Communicate

Designer content can be included in a communication originating from EngageOne Communicate. You have the following publishing options in Designer for this feature:

- Publish Attachment for Communicate - this option allows you to publish content that will be included as an email attachment in a communication designed in EngageOne Communicate.
- Publish Blocks for Communicate - you can mark blocks in your publication that can subsequently be used in campaigns using mobile templates designed in EngageOne Communicate.

Refer to [Publishing content for EngageOne Communicate](#) on page 337 for detailed information.

About publishing

Publishing is performed on the latest versions of Active Content and publications and the other files that they reference. However, for Publish types that use a production job, you can use the **Publish by label** option to publish earlier versions.

Before you can complete a publishing task all data aliases and records that have been used in the publications or Active Content must be linked with actual data elements that will appear in the input file to be used by Generate. These references are resolved by creating, updating or referencing a data map.

Resources are automatically private if they are created within another object (e.g. a document in a publication) although they can be changed to public as required. Resources created independently are automatically public. Refer to [Managing Resources](#) on page 631 for more information about public vs. private assignments.

Publications

When you run a complete publishing task for a publication you will need to specify:

- one or more output device objects that identify the type of datastream to be produced. For detailed information about customizing output devices see [Output, media and resources](#) on page 520.
- a host object that provides information about the operating system on which Generate will process the job and any settings required to communicate with the production machine when deploying the HIP file.

You can retain information about the publishing options you use in a Production Job object. If required, production jobs can be created in advance and can then be made available to users responsible for publishing.

For detailed information about customizing output device objects see [Output, media and resources](#) on page 520.

When the publishing task completes, the HIP file is created and automatically deployed to the location indicated in the assigned Host object. Normally this will be a file location on the actual server where Generate will run but if you do not have direct connectivity with the intended production machine the host object can indicate a local directory in which the HIP will be created.

Separate design and resources

When publishing the design only, a single design HIP file will be produced – with no resource information. Resources must be published separately. You can either publish all the required output devices to a single resource HIP file or publish them separately in multiple resource HIP files.

Note that when publishing the design and resources separately you will need to make the resource HIP files and output file names known to Generate in the OPS file; see “Using segmented resources” in the “Running Generate” section in the Production Guide.

Active Content

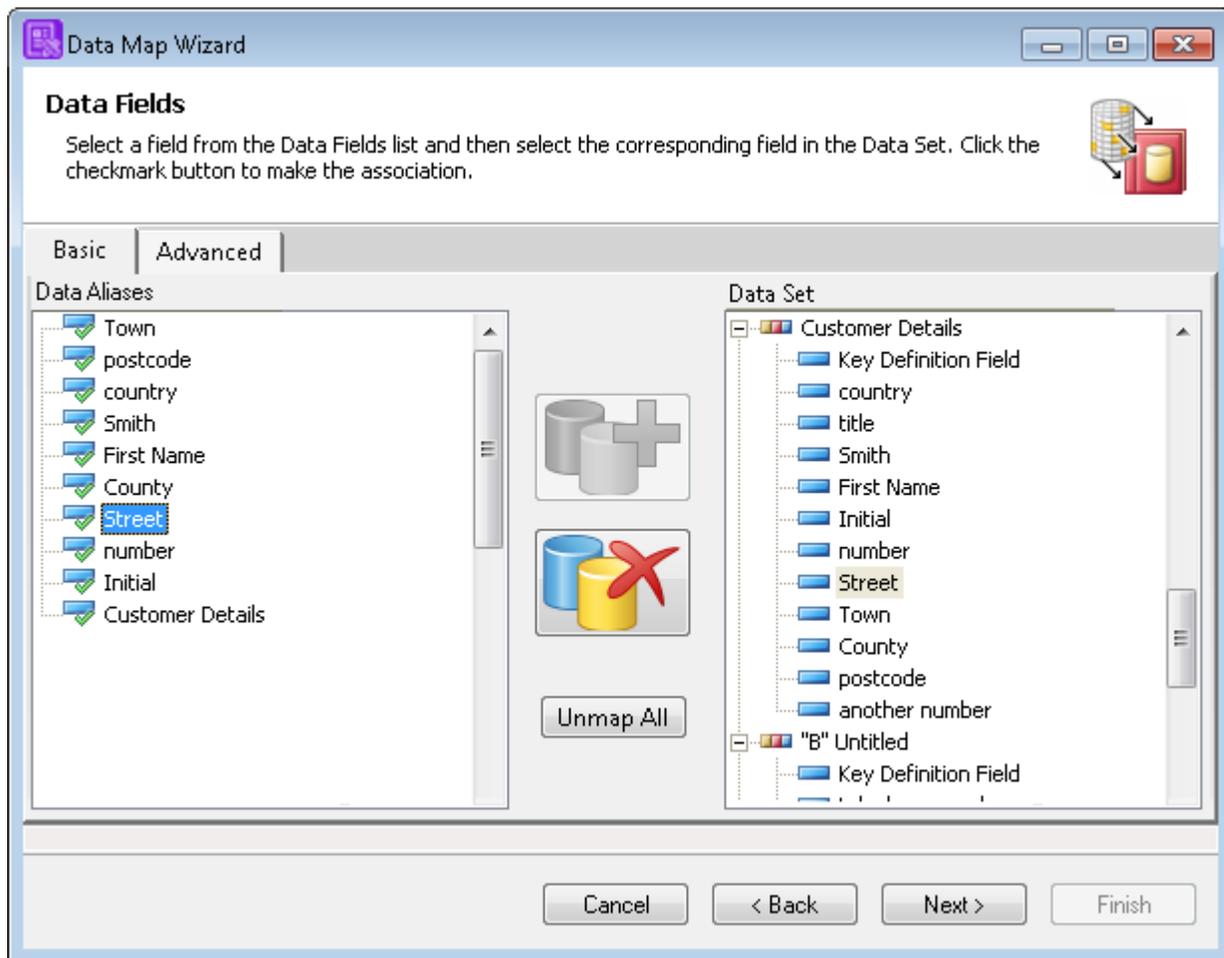
If you reference publishable Active Content in your publication you must ensure that the Active Content has already been published. For details of how to specify the location of the Active Content HIP files in Generate see “Using segmented resources” in the “Running Generate” section in the Production Guide.

About data mapping

A data map links the data aliases (fields and records) you have used in your document designs to actual data elements in the input file to be used by Generate. The definition of the input file is stored in a data format and it follows that a data format must have been created for your intended input data before you can publish.

As part of a publishing task you will need to associate each type of publication with a data map or create a new data map if required. Where your publication design contains data aliases that are not resolved in the selected data map, a dialog box is displayed listing the unmapped aliases and then the data map wizard will be initiated to allow you to provide the required links to the data format.

For details about creating a data format and exporting a data map see [Creating a data map](#) on page 496.



- Click the active icon to create or break a data alias's link to the data set.

- All the data aliases in the publication are listed in the wizard and must be associated to a field in the data set before the wizard can create the data map

The links you specify using the wizard are stored in a data map object and this can be reused in subsequent tasks and can be shared by other publications if appropriate.

You can create a data map in advance (along with a data dictionary to be used in document designs) by using the **Generate Dictionary** option when editing a data format.

Creating a host object

Host objects are used to specify the type of operating system on which Generate will run plus the settings to be used when deploying HIP files to the host machine.

Generate creates temporary work files in the host environment for processes that involve subsetting, disk caching, keyed images etc. You can allow the host operating system to create these files automatically or specify a file template which must conform to the filename conventions required by the host platform. If you are working in a memory constrained host environment you may want to use the memory handling options in the Publish Wizard to control the use of these work files.

To create a host object: in the navigator browse to the required project or folder and on the **File** menu select **New/Host**. The new host object is loaded in the workspace ready for editing.

To edit an existing host: in the navigator browse to and open the required host object and it will open in the workspace for editing.

Host object settings

Type	Indicate the type of Operating system in use on the host.
Connection	<p>This section indicates how the Designer will communicate with the host system when deploying the HIP file as part of a publishing task. If you do not have direct connectivity with the host or the host is the local machine or a computer directly available on the LAN, you should select the Local option. Otherwise, select a file transfer option to indicate the protocol to communicate with the host system.</p> <p>You have two options:</p> <p>FTP - provides a non-encrypted file transfer function. Options in this section allow you to specify the account and password when communicating with the host.</p> <p>SFTP - provides an encrypted file transfer function using the Windows OpenSSH component. To use this file transfer protocol, you must ensure Windows Open SSH has been configured to use public/private key authentication required for this type of transfer protocol. For detailed information on installing Windows OpenSSH and managing OpenSSH keys. Click here for the relevant Microsoft documentation.</p> <p>Note: The Server name used for SFTP/FTP hosts can be specified either as a computer name or a TCP/IP address.</p>

Host object settings

Location

This allows you to indicate the location on the host system where HIP files will be deployed. Always use the notation appropriate for the intended host system: for Windows or UNIX specify a folder using UNC syntax or equivalent; for z/OS specify a fully qualified dataset name and if required indicate that the dataset will be partitioned.

The **Hip Filename** option allows you to provide a specific name for the HIP file. If no name is specified the HIP will be given the name of the production job. The following parameters can be used in the creation of the HIP filename:

%B – Generate build version

%D – deployment date, the date format is yyyyymmdd

%I – production job short name (first 8 characters)

%J – production job name

%M – data map name

%N – publication name; for multiple publications the name of the first one is used

%P – project name; for multiple projects, the first publication's project is used

%R – revision of the publication; for multiple publications the highest revision is used

%T – deployment time; the time format is hhmmss

%U – the full name of the user who created the HIP

Example:

```
%D_%I = 20040215_CurrentA.hip
```

Workspace

This section allows you manage temporary files used by Generate at runtime. You can opt to have the host operating system allocate temporary files, or define a file template using %1 or %2 place holders. This option is not for use on z/OS. However you can define a temporary file explicitly when specifying the output file in a production job (see [Specifying files for Generate](#) on page 599) . Note that the file template must conform to the convention required for the host operating system. Refer to [Specifying a file template](#) on page 585 for further information.

Specifying files for Generate

At publishing time you will need to indicate the files that will provide input to a production run and those that will receive the output created by the job.

Normally you will want to assign specific references for such files, i.e. you will specify the location and file name where Generate will look for them or create them.

If required, you can choose the **Set input/output on host** option to allow the actual file references to be specified in an Override Production Settings file (OPS) that is passed to Generate when a production job is actually started. This allows you to amend file references for individual production runs. You will need to take a note of the **Name** assigned to each file by the publication task as you will need to specify this as part of the OPS format.

Information about creating an OPS file for use with Generate can be found in the “Running Generate” section in the Production Guide.

Additionally, where the intended host system is z/OS you can specify file references as DD labels that can then be resolved to actual dataset names in the JCL used to initiate Generate.

Where you are using specific file references you should always use the notation appropriate for the intended host system. For example, if Generate is going to run on a Windows or UNIX platform you might use a server, folder and file name reference; for z/OS you should use dataset/member references.

Note that if you specify only a file name (i.e. no other server/path references) Generate will normally search for the file in the location from where it was started.

Note that the HIP file that is created by the publishing process is always passed to Generate as part of the batch start-up command.

Production job file references

Input

Input files

These can comprise:

- the primary input data file for the production job that contains the variable information presented in the documents created.
 - one or more references to resource HIPs generated when you publish resources separately from the publication.
-

Production job file references

Lookup tables

If a publication uses the table lookup feature to map text references to alternative strings you will need to specify one or more files that contain the lookup information. Refer to [Lookup tables](#) on page 255 for information about the expected structure of a lookup table. **Table** references the File Alias used in the lookup table object in the publication. **File** is the location of the lookup table on disk. **Terminator** enables a character to be used to designate the end of a record entry when using a lookup table with z/OS. **Code page** enables a system or custom code page to be associated with the text in a lookup table.

Note: that the default code page setting is **Host Default**, this will be the host's code page on which Generate executes. Examples: 1252 codepage for WesternEurope/Americas Windows versions, UTF-8 codepage for Linux versions.

External key maps

If a publication uses keyed images that are not already in the Designer, then you must specify the process that will supply them. This can be either a third party file that provides the key map along with image details, or a user exit that performs the mapping and supplies the actual image. A third party file is just specified by its filename; a user exit is specified by using the format `USEREXIT (function)`, where function is the registered user exit name. For more information see the "User Exit" section in the Production Guide.

If you are using a third party file to identify the images, then this file must be available on the host machine and the images themselves must be available to the printer (or on the host machine if you are generating PDF). Images provided by a user exit must similarly be made available, for example, retrieved from a database.

The Mode option specifies how the key map is read at run time: **Cached keys in memory** will copy the key map into memory, making access much faster. This is the default setting. **Run time lookup** will access the key map externally each time. This does not use any extra memory, but will not be as fast. You may want to use this option if there are a great number of entries in the map and memory is limited.

Output

Production job file references

Output files

These are the files that will receive the output datastream from the job; i.e. the composed pages ready to be sent to the target output device and any associated work files. The number of files expected depends on the number of output devices specified for the production job and if outsourcing is being used. If required, you can override the name of the temporary file used by Generate for a specific output device using `<output file>,<temporary file>`. (This is used in preference to the **Workspace** option in the **Host Object** settings – see [Creating a host object](#) on page 597.) For information about the structure of the output file refer to the information for your intended output datastream in [Output, media and resources](#) on page 520.

Journals

If a publication writes data to one or more journal objects you will need to provide a file reference for each. You can also assign a specific code page to a journal. By default, the journal's code page is set to **AUTO** and its definition may differ depending on the type of journal object selected, as follows:

- AFP Journal: AUTO = **IBM 500**
- Text Journal: AUTO = **Host Default**
- XML Journal: AUTO = **Host Default**

The setting used for Text and XML journals above will be the host's code page on which Generate executes. Examples: 1252 codepage for WesternEurope/Americas Windows versions, UTF-8 codepage for Linux versions.

Note: that if your publication contains Structured XML journal entries the journal alias name and code page assignment are automatically set and cannot be changed.

Running a publishing task

The publishing task runs a wizard that steps you through the options required to prepare a production job. This can be to publish the complete publication, to publish the publication design or resources separately, or to publish Active Content.

Note that before you can publish Active Content you must enable it for publishing – see [Using Active Content to update publications in production](#) on page 441

When publishing publications you can identify the intended host system and output devices, specify the file references to be used by Generate on the host system and set any other run-time options relevant to the job.

You can select an existing production job object to load the settings it contains into the wizard. If you are entering new settings or updating those loaded from an existing object you will be able to save them to a production job when the wizard finishes.

Note that Publishing can also be run from a command line using DOC1Publish. This allows you to publish existing production jobs without the need to run the Designer. See the Designer Utilities guide for full details.

Earlier versions of a production job can also be run using the **Publish by label** option – see [Working with production jobs](#) on page 618.

To publish for Generate:

1. In the navigator browse to and select the publication or Active Content to be published.
From the **Tasks** menu select **Publish for Generate**. The **Publish** dialog box is invoked where you can create a new production job, select an existing one, or choose not to use a production job – i.e. a one-off publishing task. The Publish Wizard is then initiated.
2. Specify version labeling.
You can specify the label you want to be applied to all the components of this publishing task – the production job if used, the publication or Active Content, resources etc. If you don't want the publishing task to be version controlled, for example if you are testing, then select **None**. Selecting the **Create publish archive** option will create a branch of the production job.
3. Specify what to publish.
You can publish the complete publication or Active Content, just the design or just the resources.
4. Select additional publications /Active Content to publish.
Use the + and - buttons to add and remove publications/Active Content from the publish task as required. Note that if you add a publication that references publishable Active Content then they will all be included.

Note: When adding publications to the production job, wildcard matching by name is available using the filter -  option from the **Select Publication** dialog. Refer to [Filtering](#)

[by name using wildcard matching](#) on page 30 for further information on wildcard matching.

5. Complete data mapping if required.

Each publication must be associated with a data map. In the **Production job** page double-click a publication to assign, amend or create a data map – or – click the **Assign Data Map** button to associate a data map with a publication or create one from a data format using the Data Mapping Wizard.

You will be asked to update the selected data map if it does not contain references for all data links within the publication.

See [Data mapping](#) on page 496 for details.

6. In the **Output device** page identify the types of output datastreams to be produced by the job by selecting one or more Output Device objects.

See [Output, media and resources](#) on page 520 for details.

7. Use the **Input/Output** page to identify the type of host system intended to run the job and to specify references to all files be used by Generate on the host.

See [Creating a host object](#) on page 597 and [Specifying files for Generate](#) on page 599 for details.

8. Select other publishing options required and click **Finish**.

If all publication options are valid, a HIP file is created and deployed to the location specified in the assigned Host object.

If you have updated the settings from an existing production job object you will be prompted to save them.

If a mandatory option in the Publish Wizard has not been resolved the corresponding page title is displayed in Red and the Finish button is disabled.

Working with the publish wizard

Publish Wizard options

Publish Options

Versions

Label type

The screenshot shows a dialog box titled 'Versions'. It has two main sections: 'Label type' and 'Label name'. The 'Label type' section has a dropdown menu with four options: 'System', 'None', 'System', and 'User'. The 'Label name' section has a text input field containing '2011-10-04 10:35:38Z'. Below these is a 'Description' section with a large empty text area. At the bottom left, there is a checkbox labeled 'Create publish archive'.

- A system label.
- The **description** is only for a user label
- When a label is specified you can create a branch of the production job.

Specify a label for the associated production job:

System – applies the date and time only;

User – allows you to enter your own label name and description;

None – this instance of the production job will not be version controlled so will not show in the history view and will not be available for the **Publish by label** option.

Create publish archive

This option branches the production job being used. The branch can be used as normal.

Mode

Publish Wizard options

Publications:

This will publish the files required by your publication to run Generate on the host system and to print/present the resulting output datastream on your intended output devices. You can publish all the publication design and resources at the same time, or you can choose to publish them separately.

Choose from:

Complete publication produces a single HIP file that supplies all the publication design and resources.

Publication design only produces a single HIP file containing the rules for constructing the publication.

Publication resources only produces a single HIP with all the resources (fonts, output device settings, images and overlays) to support the publications you have selected. You can produce any number of resource HIP files for a publication. They can be for any combination of output devices required for the publication – one, some or all. One or more can then be used when running Generate. Note that when publishing design and resources separately the resource HIP files must be specified as input files to Generate. See the ResourceHIP option in the <Input> section of the OPS file in the Production Guide for more information.

Publish Wizard options

Active Content

This will publish an individual HIP file for the Active Content selected. If you want to publish more than one Active Content object click **Next**. Use the + icon and browse through the folders and publications in the repository using the window provided. Any publishable Active Content objects found within a selected publication are added to the publishing task, as well as:

- any publishable Active Content objects referenced within other Active Content,
- publishable Active Content referenced by keyed Active Content maps.

You can use the - icon to remove Active Content objects.

Click **Finish** to activate the publishing task. Use the window provided to browse to a destination folder, or create one for saving the HIP files.

Choose from:

Complete Active Content produces a single HIP file that supplies all the Active Content design and resources.

Active Content design only produces a single HIP file containing the rules for constructing the Active Content. Note that this is the default option for Active Content publishing.

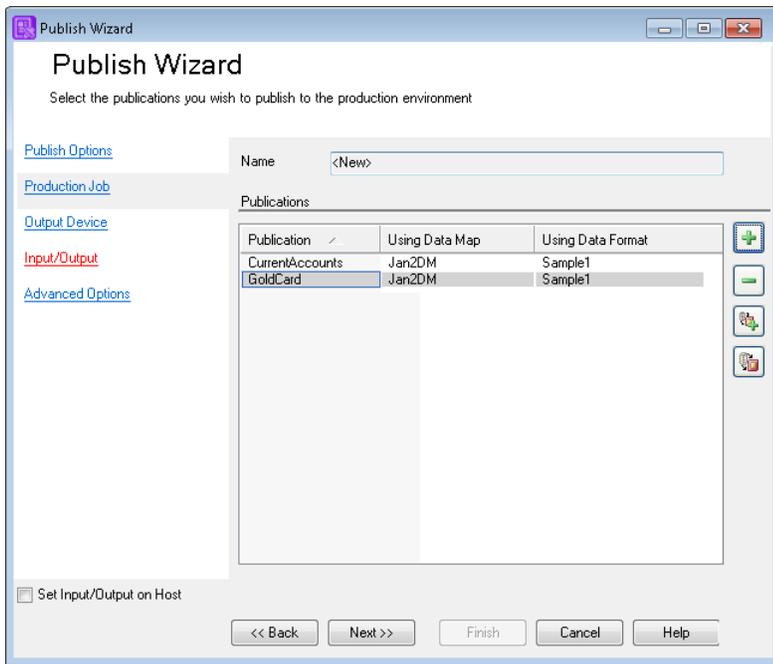
Active Content and selected resources produces a single HIP file containing the rules for constructing the Active Content together with selected font and/or image resources. This option allows you to publish specific resources used by your Active Content that may have been added or changed.

Note: It is important to note that all Active Content that use the resources being published must also be included in the production job.

Production job

Note that the options available on this page depend on whether you are publishing a complete publication, the design only or the resources only.

Publish Wizard options



- **Select** invokes the **Select Production Job** dialog box. Either create a new job or select an existing job to load its settings.
- Use these buttons to add or remove publications that will form part of the production job.
- Click **Assign Data Map** to associate the publication with a data map.
- Click **Edit Data Map** or double-click an entry to amend the data map.
- Select **Set Input/Output on Host** to use an OPS file for the path/filename of the production job's input and output files.

Output Device

Processing options

Note that the options in this section are made available depending on the user profile being used.

Use multiple output devices

If selected multiple output datastreams are created in relation to the same set of input data.

Publish Wizard options

Configure DIJ ...

If selected a Document Interchange Journal (DIJ) will be created when the job is processed by Generate. This is required when the output datastream is to be used with the Vault, Liaison and EngageOne Deliver products. When this option is selected an additional page is provided in the wizard to allow you to specify the file to receive the DIJ and to provide related settings.

- Choose from the list of available output device objects and modify its settings if required.
- You can publish to more than one output device.

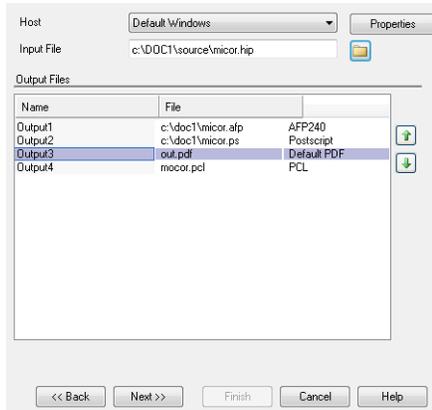
Cache AFP DBCS fonts for re-use

If selected AFP DBCS fonts are automatically stored in your local file cache. Selecting this option can result in an increase in publication time, however these fonts will be re-used by subsequent publications.

Input/Output

These pages allow you to specify the files that will provide input to a production run and those that will receive the output created by the job. Refer to [Specifying files for Generate](#) on page 599 for details. Note that the Input File browse button is disabled if a non-Windows host is selected.

Publish Wizard options

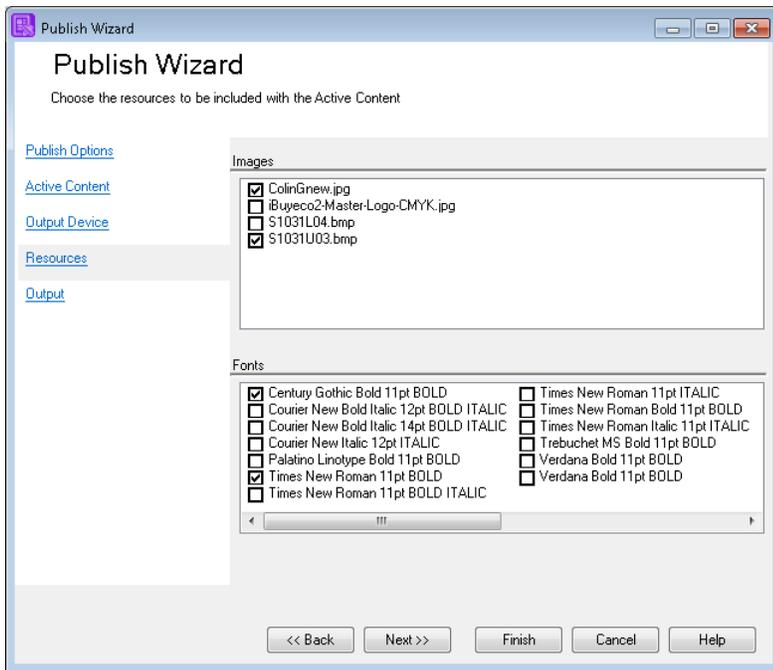


- Choose from the list of available **Host** objects for which the production job will be prepared and modify its settings if required.
- Double-click on a file to invoke the **Edit Output File Entry** dialog box and amend its settings.
- If you use the Content Author **Preview in context** option you must move **Default PDF** up to the top to be **Output1**.

Resources

Note that this page is available if you have selected the **Active Content and selected resources** mode. From this page you can select specific font and image resources for publishing with your Active Content rules.

Publish Wizard options



- Select one or more font/image resources associated with the Active Content object to be included in the publishing task.
- Click **Next** to define Output options or **Finish** to finalize the publishing task.

Output

These pages are used when publishing resources separately and allow you to specify the files that will receive the output created by the job. Refer to [Specifying files for Generate](#) on page 599 for details.

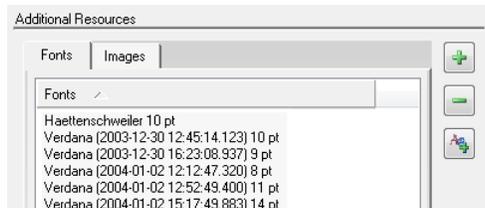
Additional Resources

Publish Wizard options

Additional resources can be included in your publications HIP file. These can either be added from the repository or imported from your local machine.

This option can be used to add extra resources referenced by RTF fields included in your input data. Refer to [Advanced data handling](#) on page 303 for further information.

Note that when including native printer fonts as additional resources you must ensure these match your intended output datastream type.



- Use these buttons to add or remove resources from the repository.
- If you wish to import resources from your local machine click on this button. Refer to [Working with design files and resources](#) on page 131 for further information.

Advanced Options

These pages allow you to fine-tune the Generate run-time environment.

Publications to process

Select Range of publications if you want Generate to process only a subset of the publication data sets available in the input data file. You may want to do this if you need to rerun portions of a production job without creating a new input data file. Use the entry field to indicate the sequential numbers of the publication data sets to be processed. For example:

- 27, 280, 674 – specific publications
- 100-1000 – publications between 100 and 1000
- 1000+ – all publications after the first 1000
- 366, 500-1000, 2000+ – combinations.

Publish Wizard options

Checkpoint progress

Use these options to indicate when checkpoints are to be created by Generate. Checkpoints output a message at user-defined intervals indicating which publication data set is currently processing. This information is used if you want to restart processing after a failure.

If you want to restart processing from the last checkpoint position you will need to specify a **checkpoint file**. You must also specify **#restart** when restarting Generate, either in the OPS file (see **OPS file reference** on page 515) or on the command line (see **DOC1GEN under z/OS** and **DOC1GEN under UNIX and Windows**

Use the **Every n publications** setting to indicate the frequency with which checkpoints are issued. This is the number of publication data sets to be processed from the input data file before the next checkpoint message is raised. If required, you can also use **Starting after** setting to indicate that checkpointing messages should not be issued until the specified number of publication data sets have been processed.

Error Handling

On error with publication data set This setting relates to the handling of keyed record data only. Use it to indicate what action should be taken if Generate encounters a publication data set that cannot be processed. This may occur in the following circumstances:

1. A record is defined as Mandatory but is missing in a publication data set
2. An undefined record key is encountered
3. A child record of a repeating group is present without its parent record.

Options are:

Abort job – no further publication data sets are processed and any output files that have been created by the job are deleted (if this is permitted by the host operating system).

Skip publication – the publication data set in error is ignored and no output is created from it. Processing continues with the next publications data set in the input file.

Record data and skip publication – as above but the publication data set is also copied to the file indicated by the Data record file field before proceeding. Note that this file specification must conform to the same naming conventions as all other Generate I/O files; refer to **Specifying files for Generate** on page 599 for details.

Publish Wizard options

When shape is positioned off page

These options indicate what action Generate should take if a graphic object (including text boxes) have offset values that mean they are positioned all or in part outside the active logical page area. Different actions are possible for graphics that are placed using constant values and those that use variable data (i.e. when using any value object other than a constant to provide an offset).

Options are:

Abort – Generate aborts immediately. Any output files that have been created by the job are deleted (if this is permitted by the host operating system). This is the default for objects placed using variable data.

Warn – a warning message is issued for each object that is found to be positioned outside the logical page area. Processing of the job continues as normal. The off page object is included in the output datastream; the effect of this in the printer/browser environment will depend on the device type.

Ignore – as above but no warning message is issued. This is the default for objects placed using constant values.

Suppress messages

Use this setting to indicate any Generate diagnostic messages that you do not want to be issued for this production job. You can either indicate the category of messages to be suppressed or use the **Only these messages** option to indicate specific messages that you do not want to be reported. When using this option you must specify only the identifier that appears before each message when issued.

Memory Handling

Publish Wizard options

The default setting of **Allocate memory as required** allows Generate to call for system memory in the amounts most suitable for its processing needs. If you need control over such allocations select the **Pre-allocate memory at start up** option and then use the **Allocate memory blocks of size** setting to specify the amount of additional memory to be reserved each time a call is made. You can also use the **Start up with** setting to indicate an initial memory allocation prior to any additional blocks being called.

If you need to set a maximum limit for the total amount of concurrent memory that can be allocated by Generate, select the **Abort if memory exceeds** option and specify the limit in the entry field.

By default, Generate will attempt to hold all the pages of output datastream created for an individual publication in system memory until it is complete. This means that Generate can perform output functions as efficiently as possible. You can use the **Limit composed pages** option to specify an upper limit to the amount of memory to be allocated for this purpose. If this option is enabled a default of 4mb will be used as the limit although you can adjust this to any value above 512k. When the memory requirement of a publication exceeds this limit the existing pages in memory will be output to a local disk file which is used as temporary work space. When the publication is ready for output the pages in the work space will be retrieved and merged with those still in memory. Use the **Overflow...** option to indicate a specific file location for temporary workspace or, if this option is left blank Generate will create a filename based on the **Workspace** configuration in the host object. Note that this file specification must conform to the same naming conventions as all other Generate I/O files; refer to [Specifying files for Generate](#) on page 599 for details.

Note that all system memory settings are, by default, expressed as kilobytes. You can alternatively use the M suffix to indicate a number of Megabytes if required. For example: 64 = 64 kilobytes 100M = 100 megabytes 128K = 128 kilobytes

Diagnostics

Issue report of memory usage

If this option is enabled Generate will issue a summary of the amount of memory allocated by the job on completion. You may want to use this option during testing to optimize memory handling settings.

Log file

This option is used to specify the file that will receive any error or warning messages issued by Generate. The filename and path must be in the required format for the operating system.

Publish Wizard options

Produce trace information	This provides a method of tracing publications that cause errors in Generate or Preview. A tree structure representing the path that Generate took that caused the error is displayed. This enables the error to be tracked down to a particular object in the publication design. Note that this can also be specified in the OPS file. See Debugging publications on page 515 for more information.
Trace output file	Specifies a file for the output from the trace. The filename and path must be in the required format for the operating system. If no file is specified the output is sent to the standard output medium of the Generate host environment (e.g. command prompt window, system log, etc.). If you specify %d in the file name, for example: <pre>OutputFile=traceoutput%d.txt</pre> then %d in the file name will be replaced with the date and time the file was generated.
Memory limit	This is the maximum amount of memory in bytes that can be used for buffering trace information. A value of '0' means no limit.
Output file codepage	This allows you to change the code page for the output file from the default host code page to one that is more suitable.
Verbose trace information	Set this to include GUID information and the internal names of the opcodes that are being executed.
Messages	
If your publication design contains messages created using the Message1 or Content Author products, use this page to customize the production side of such message application environments. For more information see Importing the message environment file on page 321.	
Messages	Click the + icon to display the Add Message dialog box. You can search for one or more message files (.HIM) that are to be used with the publication by using the Browse button.
Audit trail file	Specify the file to be used for audit information for each message that is included in the published documents.
Mandatory not placed	Select the action you want Generate to take when a mandatory message cannot be included in a document for which it was intended. Stop – the default is to stop processing Continue – ignore the error and carry on Warn – issue a warning and continue.

Publish Wizard options

Mandatory message error	Select the action you want Generate to take when a mandatory message is found to have unresolved links – typically when a font used by the message is not included in the resource pack or when a data alias used in a message has not been mapped. Stop – the default is to stop processing Continue – ignore the error and carry on.
Optional message error	As above but specifies the action to be taken for non-mandatory messages.
Use HIP font mapping	Select this to ensure that any mappings that have been applied to fonts used in a publication in the Designer (i.e. are in the HIP file) will also be applied to the same fonts used in messages created in Content Author or Message1.
Server Mode	
This section is used if you want your publication to be processed using the Server Mode method of running Generate under UNIX and Windows. For more details on Server Mode, see the Production Guide.	
Run Generate in Server Mode	Select this option to activate Server Mode and enable its related settings. Abort on failure defines the behavior of Generate if a print job fails. Selecting this option will abort Server Mode, otherwise Server Mode attempts to discard the failing print job and await further commands.
Command Queue	<p>Server Mode uses a command queue to receive commands and application data. The command queue can be specified either as a named pipe or a TCP/IP socket. The format for named pipe under NT should be</p> <pre>\\.\pipe\cmdqueue</pre> <p>and under UNIX it should follow a path/filename format such as</p> <pre>/temp/cmdqueue</pre> <p>For TCP/IP the format for Socket is an IP address such as</p> <pre>10.133.54.202</pre> <p>and for Host it would be a host name, such as</p> <pre>spa02</pre> <p>Both use the Port field which is a number corresponding to the required port.</p>

Publish Wizard options

System Commands

These commands can be specified to be executed before Server Mode has processed application data, if it completes successfully or if it fails to complete successfully. They can be any commands that can be passed to the appropriate operating system from the command line.

Custom

This page allows you to enter any special settings that may be required as part of problem resolution. Only enter values here at the request of Precisely Support.

Working with production jobs

A production job object holds settings to be used with publishing tasks in the Designer. You can create a production job at the time when the Publish task is run to gather the options you specify. Alternatively you can create a new object in the navigator for future use. Production jobs will always use the latest versions of resources, but you can run earlier versions of a production job that uses older resources.

Note that a production job can also be run from a command line using DOC1Publish. This allows you to publish production jobs without the need to run the Designer. See the Production Guide for details.

To create a production job object in advance:

1. In the navigator browse to the required project or folder and on the **File** menu select **New/Production Job**.
2. Enter a name and optional description for the production job and click **OK**.
3. Configure the Production Job, Output Device, Input/Output and Advanced Options settings as necessary.

To edit an existing production job: in the navigator browse to and open the required production job.

To publish using an older version of a production job: In the navigator browse to and select the required production job.

- From the **Tasks** menu select **Publish by label**. The **Find version label** dialog box is invoked – use this to select the version you want to publish.
- Click **OK** to start the publish task. Alternatively you can right-click the required production job in the history view and select **Publish by label** from the context menu. The publish task will run. Note that progress of the pub

lish task will be shown in the status bar.

39 - Access control and security

Access to Designer and its design objects is governed by an access control schema. Each project, folder and object has its own set of access rights.

All users must have a user account before they can log in to Designer. Roles are used to specify the functions that users can access. Groups of users can be created to make access control easier to manage.

Design objects are stored in projects and folders. Each project, folder and object has its own set of access rights – or privileges – which can be assigned to users and groups to determine which users have access and the level of access they should have. By default no security is assigned – i.e. all design objects are available to all users.

In this section

User accounts.....	620
User roles.....	622
User groups.....	625
Security.....	626
Securing data transfer.....	629
Sessions.....	630



User accounts

Before any user other than the initial Administrator can log in to the Designer they must be assigned a user account. Each account is given specific access rights to resources, which are assigned using roles. A user account can have any number of role assignments. Users can also be assigned specific project, folder and object privileges which will override those available via the role.

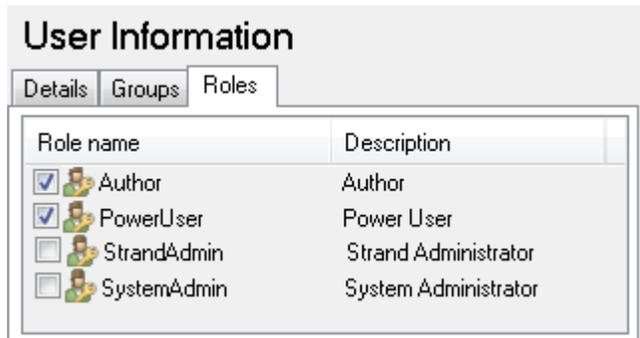
Additionally you can assign access rights by including user accounts in groups which also have roles assigned. The roles defined for the group are conferred on each user belonging to it in addition to the roles that the individual user has.

There is one built-in user account called the admin user account. This is a permanent member of the Administrators user group and has full administrative rights and full access to all projects and folders; it cannot be removed.

NOTE: The system administrator does not need to be added to groups or roles as they already have full access rights.

Only a system administrator has full control over creating and deleting user accounts, assigning groups and roles, and changing user names. Other users are restricted to editing their own full name and password.

To work with user accounts: on the menu click **Administration/Users**. The **Users** view will be shown in the workspace. Click on an account to view and edit its details. Note that editing functionality may be restricted depending on the roles you are assigned.



To create a new user account: access the **Users** view and click **Add**.

The screenshot shows two parts of the user management interface. The top part is a table titled "Users" with columns for Full name, User name, and Status. The bottom part is a "User Information" panel for the selected user, Robin Goodfellow, with tabs for Details, Groups, and Roles.

Full name	User name	Status
Administrator	admin	Logged On
Andrew Aguecheek	Andy	
John Falstaff	JohnF	
Juliet Capulet	Julie	
Robin Goodfellow	Puck	

User Information

Details | Groups | Roles

Full name: Robin Goodfellow

User name: Puck

Password: *****

Save details

In the **Add New User** dialog box enter information for the new account and click **OK**. The account is created with the default role of Power User. Note that all fields are mandatory.

The screenshot shows the "Add New User" dialog box with the following fields and buttons:

Full name: Product manager

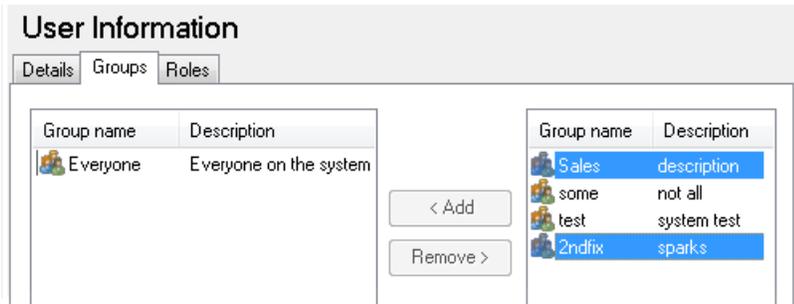
User name: prodman

Password: *****

Buttons: OK, Cancel, Help

To delete a user account: in the **Users** view select the accounts you want to delete and click **Remove**.

To associate a user with groups: in the **Users** view select the relevant user account and click the **Groups** tab. Select one or more groups to **add** the user to, or **remove** the user from, or double-click on one to move it from one list to the other. See [User groups](#) on page 625 for more details.



To associate a user with roles: access the relevant user account details and click the **Roles** tab. Select the required roles for the user. See [User roles](#) on page 622 for more details.

To modify your user details: on the menu click **Administration/Edit user details**. The **User details** dialog box is displayed where you can change the **full name** and the **password**.

User roles

A role is a set of privileges that gives authority to perform actions and to access specific objects within Designer. They are assigned to user accounts and user groups and represent a user's job function. Roles are cumulative so users with multiple roles will have the capabilities of all the assigned roles.

There are four built-in roles:

System Administrator – allows all actions to be performed and gives permanent full access to all areas of the Designer. It includes tasks such as management of users, groups, projects and roles, including access control and security. It also includes all tasks available to the other three built in roles, so user accounts with this role do not need to be assigned any other roles or privileges.

Project Administrator – the same as a system administrator but for specified projects only. It allows the user to perform all actions within a project and gives permanent full access to all areas within the project. User accounts with this role need to be assigned to projects in order to have access to them.

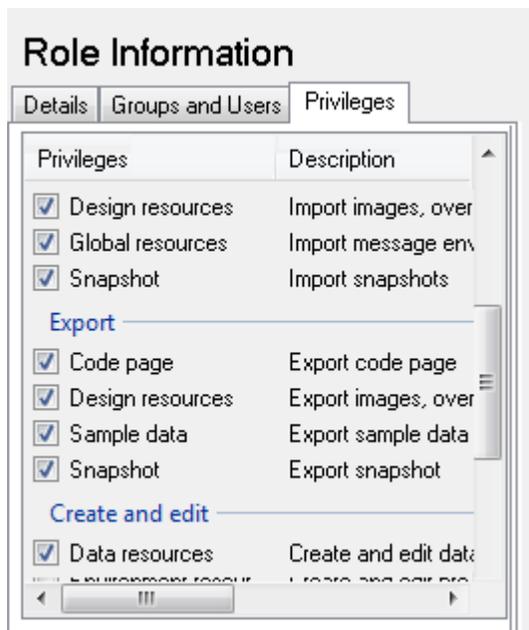
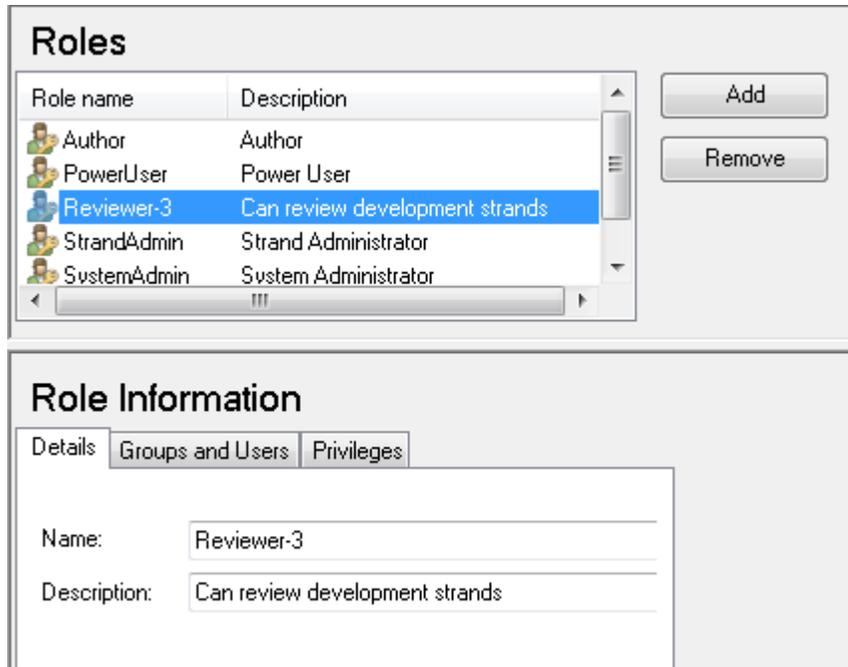
Power User – the default role for new user accounts. It provides the ability to perform actions related to all aspects of design in the Designer. This includes publishing, editing of locales, data setup and snapshot functionality. It does not allow administrative functions such as management of user accounts, user groups or projects.

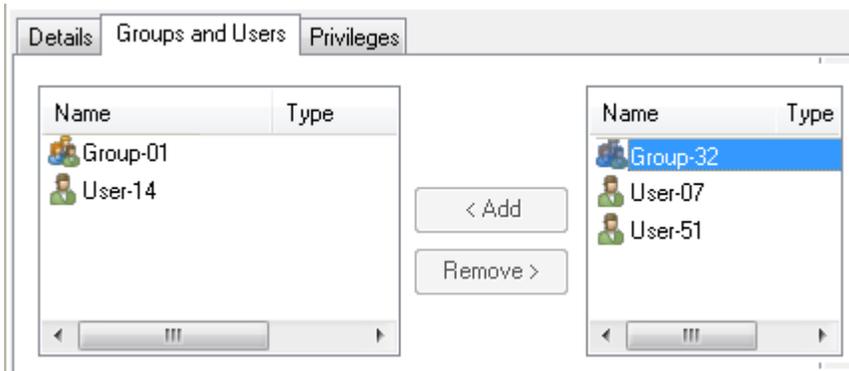
Author – provides the ability to edit publications, documents and Active Content. Access to other aspects of design, such as publishing and snapshots, are not permitted.

You can create new roles as required, but the built-in roles cannot be deleted and only their description can be modified.

To work with roles: on the menu click **Administration/Roles**. The **Roles** view is shown in the workspace where you can view details and, as a system administrator, add, edit and remove roles.

To create a custom role: access the **Roles** view and click **Add**. In the **Add new role** dialog box enter a unique name for the role and an optional description and click **OK**. The role is created. You can copy privileges from an existing role or specify them explicitly.





To delete a role: access the **Roles** view and select the accounts you want to delete and click **Remove**. Note that you will not be allowed to remove the built-in roles.

NOTE: Menu options in the navigator for which a user does not have the required privileges are indicated by **no access icon** in the menu.

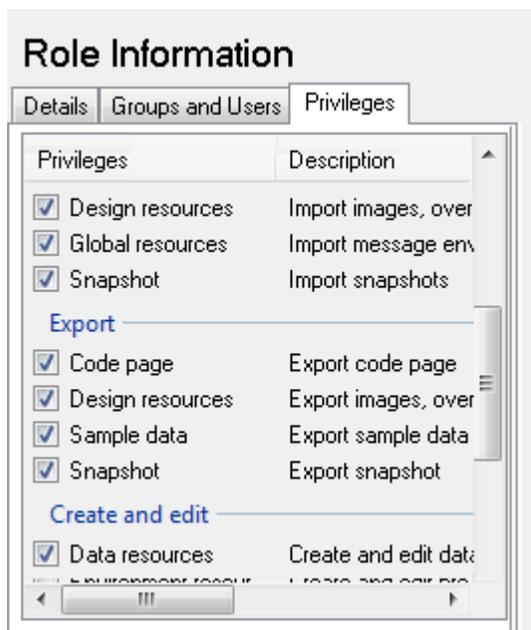
Privileges

A role is defined by specifying the various functions – or privileges – that the role is allowed to perform. These include administration, import/export, preview, publishing, and version control. Roles can be assigned any number and combination of these. Only users with the system management privilege can edit and create roles.

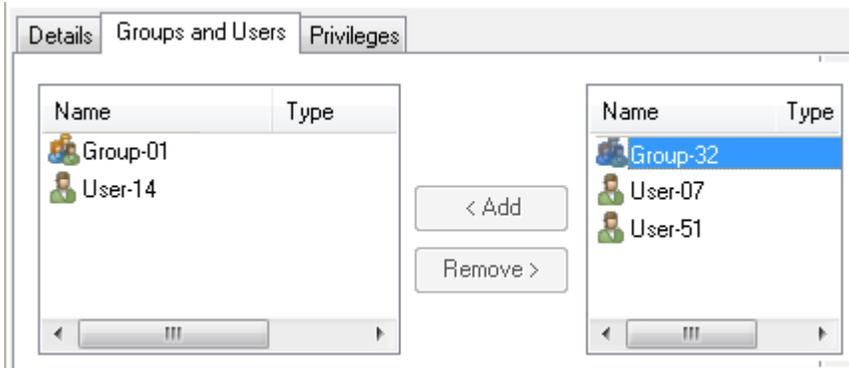
Note that the absence of a privilege does not mean that access is always denied as privileges are cumulative when a user is assigned more than one role.

Privileges should be used in conjunction with security rights – see [Security](#) on page 626.

To specify the privileges of a role: access the **Roles** view, select the required role and then select the **Privileges** tab. Select and deselect privileges as required.



To assign a role to a group or user: access the **Roles** view, select the required role and then select the **Groups and users** tab. Populate the list as required. Use **Ctrl + click** to select more than one.

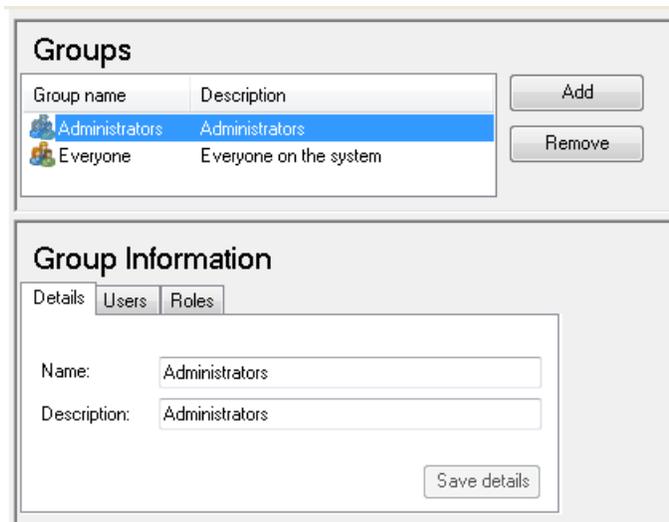


Note that you can also assign roles to groups and users using the **Groups** view or the **Users** view.

User groups

User groups provide the same access control functions as individual user accounts but apply to all users assigned to a group. If user accounts have individual roles these will apply in addition to the group roles.

There are two built-in groups:



System Administrators – by default users in this group are assigned the system administrator role so can perform all repository administration tasks and all design tasks. The admin user account is by default a member of this group.

Everyone – users are automatically added to this group.

You can create new groups as required but note that the built-in Everyone group cannot be modified or deleted. As many users as required can be added to a group but you cannot add one group to another (i.e. create nested groups).

To work with user groups: on the menu click **Administration/Groups**. The **Groups** view is shown in the workspace where you can view details and, as an administrator, add, edit and remove groups.

To create a new user group: access the **Groups** view and click **Add**. In the **Add New Group** dialog box enter a unique name for the group and an optional description.

To delete a user group: in the **Groups** view select the groups you want to delete and click **Remove**.

Security

Every item in the navigator – project, folder, object – can have security applied to it. This controls which users can access the item and what level of access – or privileges – they have e.g. create, publish, etc. By default no security is assigned – i.e. all design objects are available to all users. Only those with project administrator privileges can apply security. Note that project administrators must be added to a project before they can manage it.

When a project is first created every user is given full access to the objects contained within it – folders and resources – although the functions that they can perform on these will depend on their user roles.

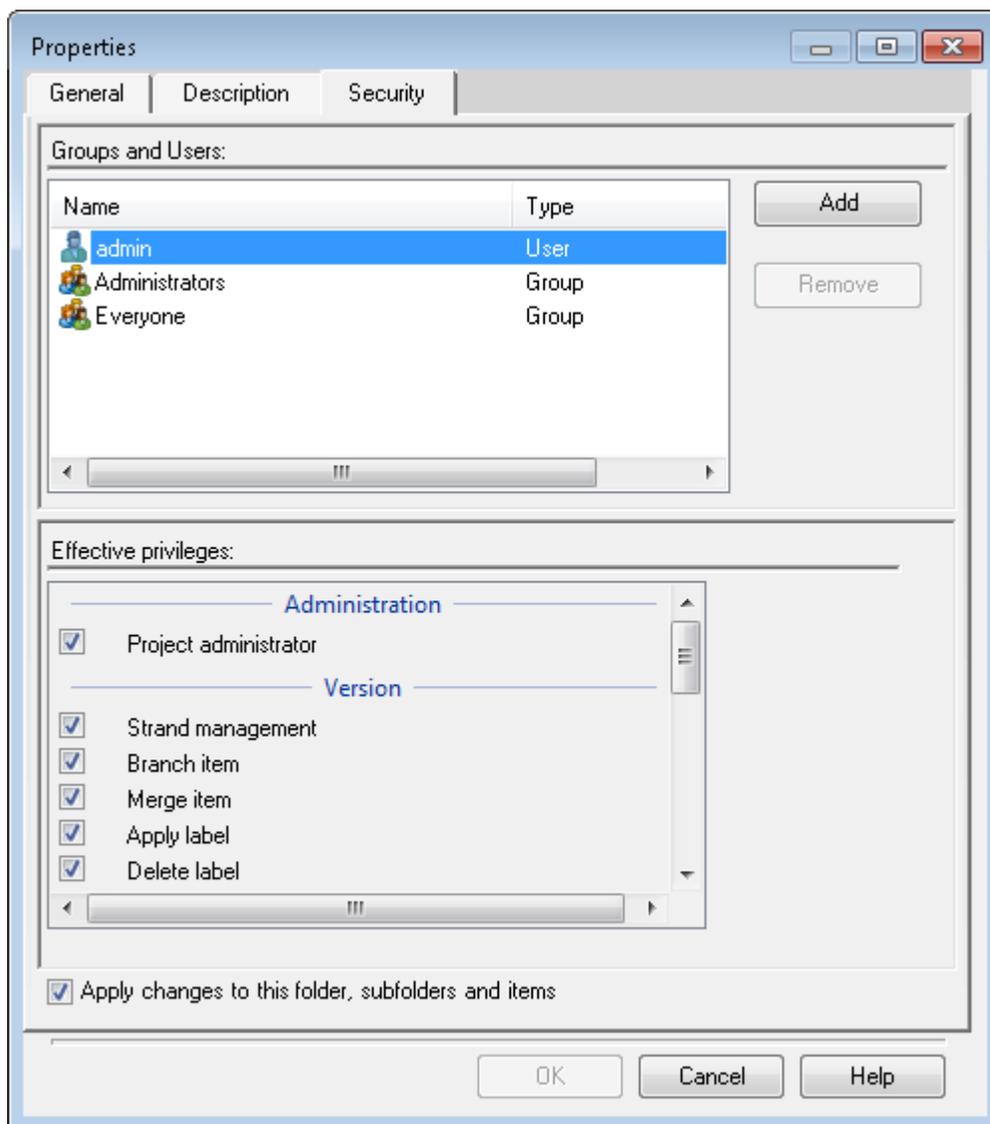
It is recommended that folders are used to control security, and that all objects within the folder have the same access rights as the folder. If objects within a folder have different access rights to the folder and there is a conflict when accessing an object, then the security on the object is used. For example, if a user only has read access to a folder but has create/edit access to one of its objects then they will be able to edit that object.

When a folder or object is first created it will inherit access rights from the folder (or project) in which it is created. The table below shows how security settings are applied when items are generated in other ways.

Action on an item	Access rights
create new	projects – everyone has access subject to their user roles; otherwise inherit rights from the containing folder or project;
save as, make public, copy & paste	inherit access rights from the target folder or project;

Action on an item	Access rights
cut & paste	preserve existing access rights;
branch	everyone has access subject to their user roles;
merge	new items inherit access rights from the target folder or project;existing items will preserve existing access rights from target project;
import	inherit access rights from the target folder or project;

To control access to a project, folder or object:



1. Specify which users and groups can access the item.

In the navigator select the required item and from the menu and click **File/Properties**. In the **Properties** dialog box switch to the **Security** tab. **Add** and **remove** users and groups as required.

2. Set the **privileges** for a user or group.

Select a user or group to view and modify their **privileges**.

Note that the 'admin' user and the 'Administrators' group have full access to everything, and their privileges cannot be changed.

3. Choose the scope of the security settings.

When setting the security for a project or folder you can bulk-apply the settings to the project/folder and to all the items it contains by setting the **Apply changes to this folder, sub-folders ...** option before clicking **OK**.

Securing data transfer

To ensure client/server data is securely transferred you can configure all data communications to use secure channels. This mode of data transfer contains a set of security protocols that provide identity authentication and secure communication through encryption.

To activate this method of data transfer, set `secure="true"` in the following files:

- `BusObj.config` on the server machine at `C:\Program Files (x86)\EngageOne\EngageOne Compose\Designer\Server`.
- `Processor.dll.config` on all client machines at `C:\Program Files (x86)\EngageOne\EngageOne Compose\Designer\Client`.
- `ResourceAccessService.exe.config` on all client machines at `C:\Program Files (x86)\EngageOne\EngageOne Compose\Designer\Client`.
- These configuration files are shipped with Secure Channel deactivated.

Here is an example of the `secure` attribute of the `channel` element in `BusObj.config`.

```
<system.runtime.remoting>
  <customErrors mode="off"/>
  <application>
    <channels>
      <channel ref="tcp" secure="true" port="37719">
        <serverProviders>
          <formatter ref="binary" />
        </serverProviders>
      </channel>
    </channels>
  </application>
```

1. On the server, edit `BusObj.config` and set the channel tag `secure` attribute to `true`.
2. On all client machines, edit `Processor.dll.config` and set the channel tag `secure` attribute to `true`.
3. On all client machines, edit `ResourceAccessService.exe.config` and set the channel tag `secure` attribute to `true`.
4. On the server machine restart EngageOne Template Designer Application Service using the Windows Start button and then run `services.msc`.
5. Locate EngageOne Template Designer Application Service and restart the service.
6. Restart Designer on the client machines.

Sessions

SUPPORT: Sessions in Designer, Application Import and DOC1 Publish can be terminated.

Occasionally in a multi-user environment a user's login session may need to be terminated by the administrator. This could be necessary, for instance, if a session has been left running for some time and another user wants to access the same object. The user is logged out with immediate effect (although the Designer will remain open) and when they next access their session will be notified that they must log back in. Once logged in again, they will be given the option to save their changes with a different name for objects such as publication, Active Content, data format etc. Otherwise they will only be able to view their previous edits in the Designer, but must apply changes again. Only those with the system management privilege can terminate a session.

To terminate a user's login session: on Administration click **Sessions**. In the **Sessions** window select the session to be terminated and click **Logout**.

40 - Managing Resources

Resources stored in the Designer repository are considered to be one of two types: composite or non-composite. Composite resources are objects such as publications and documents that consist of and make reference to other resources – composite and non-composite – stored in the repository.

Designer provides a version control system for tracking and maintaining these resources. This enables you to branch objects into a separate project. You can then work on these objects independently of the original project, and then merge changes back into the original. This allows the integrity of the original resources to be preserved whilst you are developing new versions. For details see [Version control](#) on page 132.

In this section

Private/public resources.....	632
Working with resource objects.....	632



Private/public resources

Most resources are considered public – which means that they can be used by any composite object in the same project. The exceptions to this are document designs that are created as part of a publication. These documents are considered to be private to an individual publication unless specifically made public.

Note that this should not be confused with allowing access rights for a particular resource to other users. Access rights can be conferred in object properties regardless of the public/private status.

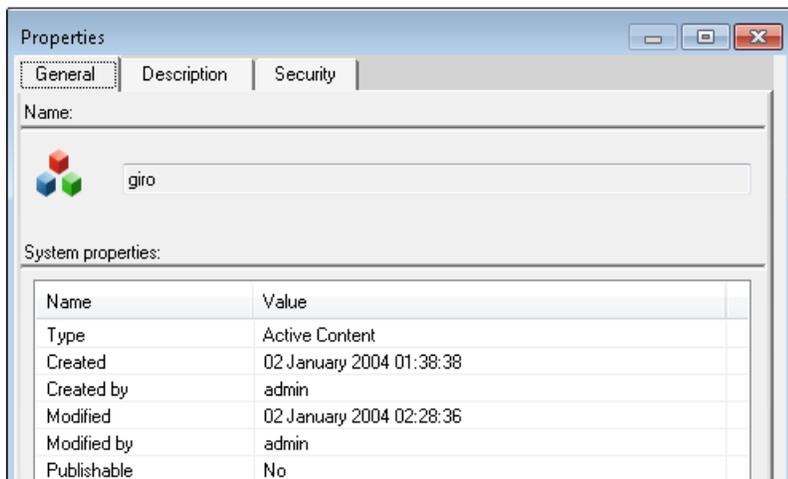
To make a document public: from the Document Editor select **View/Publication** or right-click in the logic map and click **View Publication**. Highlight the document and select **Edit/Make Public**.

Note that once a document has been made public it can no longer be edited from within a publication. To edit a public document, you must open it directly from the navigator.

Working with resource objects

Resources are shown in the form of a tree structure in the navigator. They are grouped in one or more projects which may be further divided into folders. The navigator is always available in the Designer, although it may be hidden from view. You can use the filter to show the resource types that you want to see in the navigator. For more information see [Using the navigator](#) on page 20.

Resource properties



- The **General** tab shows details of the resource such as name, type, dates etc.
- The **Description** tab includes any user comments.
- **Security** shows which users have access to an object and what rights they have, such as read and write.

Each object in the repository has properties that provide detailed information about the resource. The properties can also be used to set the access rights for a resource – see [Security](#) on page 626

To view resource properties: highlight the resource that you want to view and select **File/Properties**.—or—right-click the resource and select **Properties** from the context menu. The **Properties** dialog box is displayed.

Copying

Copying a resource creates a new instance of the object that is independent of the original – it is not a branch. Resources can only be copied within the same project. To copy between projects you must branch the object and merge it back into the required project – see [Version control](#) on page 132. Note that the access rights of the new copy of the resource will be the same as the original resource. You can change them as require, for example, by setting the access rights for the containing folder or project.

To copy resources within a project: in the navigator highlight the resources you want to copy and select **Copy** from the **Edit** menu or the context menu. Select the location in the same project where you want to copy the objects and from the **Edit** menu select **Paste**.

Saving

If you open a publication and modify it and then use the **Save** command it will save and over write the original publication stored at the original location. If on the other hand you open an existing file, modify it and use the **Save As** command you can keep both the modified and original versions of the publication, as long as you use a different file name from the original.

Deleting

The latest version of a resource can be deleted from the navigator provided that it is not 'used by' any other resource. You must have delete access rights for that resource. The resource remains in the navigator marked as deleted. If required you can choose to show items that have been deleted from the navigator,. Note that you cannot undo the deletion of resources.

Use the **Show Dependency** option to see what a resource 'uses' and is 'used by' – see [Displaying object associations](#) on page 635 .

Complete projects and all their contents can also be deleted from the navigator. Once deleted you cannot undo the action, so care must be exercised when using this option. You must have project management privilege for the relevant project.

To delete a resource: highlight the resource to be deleted in the navigator and on the **File** menu click **Delete**. – or –right-click the file to be deleted in the navigator and select **Delete** from the context menu. The resource will be marked as deleted.

To show deleted resources: in the navigator on the **View** menu select **Show deleted items**. All deleted resources except projects will be shown.

To delete a project: in the navigator highlight the project to be deleted and from the **File** menu select **Delete**. – or –right-click the project to be deleted in the navigator and select **Delete** from the context menu.

Renaming

Changing resource names does not affect the relationship between files as Designer keeps track of file associations via unique identifiers embedded in each file.

To rename a resource: highlight the resource in the navigator and select **Rename** from the **File** menu or the context menu. Edit the name and press Enter.

Viewing and editing

Resources can be opened for viewing or editing from the navigator. The latest version of the resource will be opened in the appropriate editor. If the resource is in use and locked by another user you will only be able to open it for viewing (read-only). As well as viewing the actual resource, you can view its properties, history and use of resources.

To view or edit a resource: in the navigator browse to and select the relevant resource and from the **File** menu select **Open** or **Open Read-only**– or –right-click on the resource and select **Open**. The file will be opened with the appropriate editor in the workspace. You will not be able to edit or save when opening read-only. For more information on Editing and Creating resources, see [Working with design files and resources](#) on page 131.

Displaying object associations

The navigator provides the ability to view the dependencies that a resource has with other objects in the repository – such as which objects use this resource or which objects are used by this resource. Note that only public objects will be shown.

To show the dependencies of a resource: select the resource in the navigator and from the **File** menu select **Show dependency**– or –right-click on the resource and select **Show dependency** from the context menu.

Note: The **Show dependency** option is also available from the advanced search feature for a selected search result.

A window with two views is displayed – a Tree view and a Details view.

Tree view Details view

Resources directly used by this item:

Name	Type	Creation date	Created by	Last modified	Last mo...
Boardwalk PostScript	Production Job	11/6/2006 8:11:37 PM	admin	11/6/2006 8:11:37 PM	admin
Boardwalk PDF	Production Job	11/6/2006 8:06:27 PM	admin	11/6/2006 8:06:27 PM	admin

Resources this item directly uses:

Name	Type	Creation date	Created by	Last modified	Last mo...
Boardwalk Map	Data Map	3/10/2004 12:03:31 PM	admin	3/10/2004 12:03:33 PM	admin
blackcat.bmp	Image File	3/10/2004 12:05:44 PM	admin	3/10/2004 12:05:49 AM	admin
Gas.bmp	Image File	3/10/2004 12:05:49 PM	admin	3/10/2004 12:05:49 AM	admin
Arial 9 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 11 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 8 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 5 pt	Font	3/10/2004 3:36:08 AM	admin	3/10/2004 3:36:08 AM	admin
Arial 6 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 11 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 10 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin
Arial 8 pt	Font	12/17/2002 3:31:39 AM	admin	12/17/2002 3:31:39 AM	admin

Indirect dependencies of this item (expanded objects in the tree view only)

Name	Type	Creation date	Created by	Last modified	Last mo...
User Defined Resource...	Resource Map	4/10/2007 11:45:39 AM	admin	4/7/2011 10:40:02 AM	admin
blackcat.bmp	Image File	3/10/2004 12:05:44 PM	admin	3/10/2004 12:05:44 PM	admin
Cable.bmp	Image File	3/10/2004 12:05:45 PM	admin	3/10/2004 12:05:45 PM	admin
Details.bmp	Image File	1/8/2004 3:41:42 AM	admin	1/8/2004 3:41:42 AM	admin
dynamite.bmp	Image File	3/10/2004 12:05:47 PM	admin	3/10/2004 12:05:47 PM	admin
Elec.bmp	Image File	3/10/2004 12:05:48 PM	admin	3/10/2004 12:05:48 PM	admin
Gas.bmp	Image File	3/10/2004 12:05:49 PM	admin	3/10/2004 12:05:49 PM	admin
logo.bmp	Image File	1/8/2004 3:19:01 AM	admin	1/8/2004 3:19:01 AM	admin
Innn? hmn	Image File	3/10/2004 5:52:51 AM	admin	3/10/2004 5:52:52 AM	admin

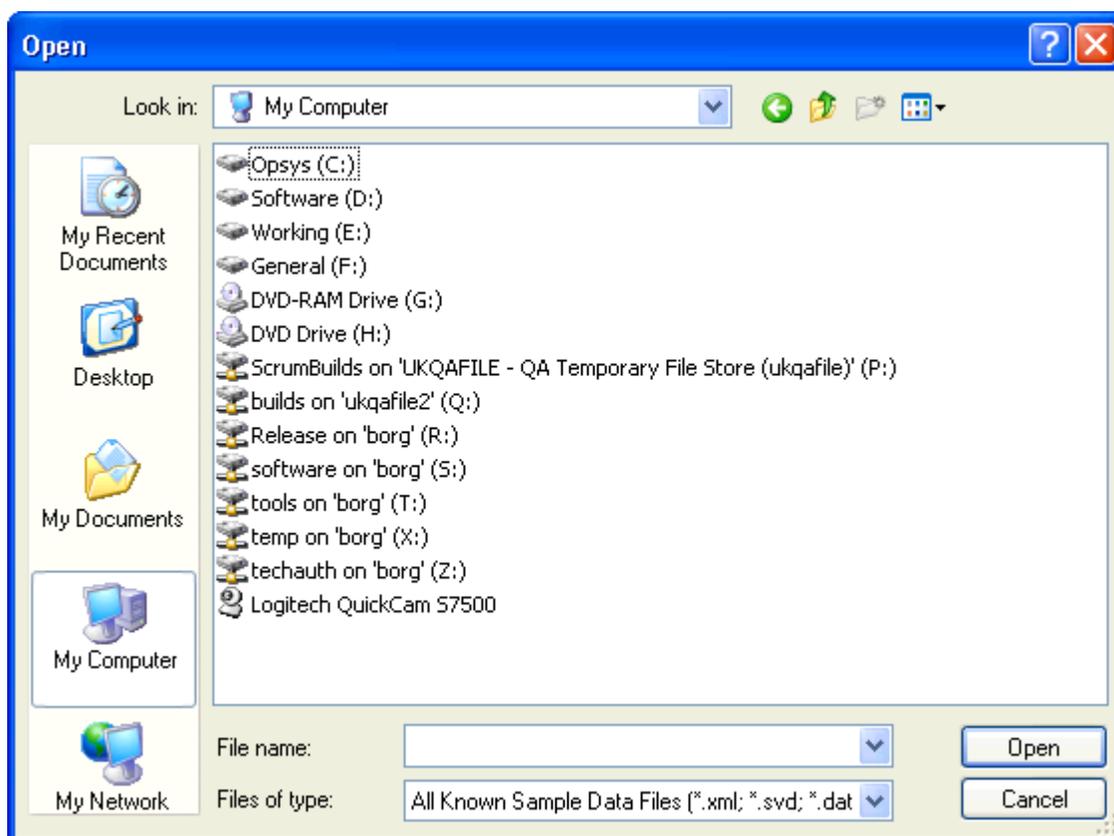
Export to file Close Help



- The tree view shows the hierarchical relationships of the resource.
- The order will be the same as in the navigator
- The first level under each resource shows its direct dependencies, subsequent levels show indirect dependencies.
- You can export the current view to a comma-separated text file (.csv).
- This can then be viewed, for example, in a spreadsheet program.

- This shows the details of the resources in the **tree view**.
- It shows the direct dependencies and their dependencies (indirect) that are currently showing in the **tree view**.
- Click in the header to sort on a column.

Opening files Designer provides standard Windows functionality to browse for and open files. What you can browse will depend on the type of file you want to open or import into the navigator.



- When importing files into Designer you can browse local folders and select the type of files that will be displayed.

41 - Repository management

The Designer Repository stores and controls revisions of all the design resources and other objects that are used in the Designer. These include resources that are created within the Designer itself or have been imported into it from an external source.

For details of version control see [Version control](#) on page 132.

The repository consists of a database and a filestore. The database is used to provide revision control for the repository and will be in the default location as specified in the SQL Server. The filestore is where Designer resources and design files are stored, and is specified as a folder location.

In this section

Designer Repository Configuration.....	639
Snapshots.....	647



Designer Repository Configuration

Administrative tasks can be performed on the repository using the Designer Repository Configuration Tool. For instance you can install a new repository, upgrade, backup and restore existing repositories, and enable different repositories to be used.

The tool is supplied with the product distribution material and is run separately.

To run the Repository Configuration Tool:

1. Ensure that there are no users using the repository.
2. From the **Start** menu select **Programs\EngageOne\Repository Configuration Tool** (or **Programs\Group1 ...**)
3. In the **Connect to SQL Server** dialog box click the **Browse** button to select the server that is being used for the repository from the **SQL Server** list – or **local** if it is on your machine. If you are connecting to an SQL server that requires its own login, i.e. not the Windows user name and password, clear the **Use Windows authentication** check box and enter the **SQL Login Name** and **Password**. Click **Connect**.
4. Select the required task from the tabs along the top of the window. More detailed instructions for each task are given below.

Setting up the repository

A repository is configured and set up as part of the Designer installation process. The Repository Configuration Tool also allows you to install a new repository, and upgrade or change the settings of an existing repository.

You can install any number of repositories, selecting which one to use when logging in to Designer – see [Repository selection configuration](#) on page 641.

Installing a new repository

You can install any number of repositories, selecting which one to use when logging in to Designer – see [Repository selection configuration](#) on page 641.

1. In the Repository Setup page, click **Install a new repository**.
2. Enter the **Database details**.

If you specify an existing database name, you will be given the option to overwrite the database. The database itself will be in the location that was specified in the SQL Server. You can specify another location but it is not recommended.

Note that the backup locations for the database and the filestore used with the backup/restore options must be accessible by the database server. For Client/Server environments the accessible location is determined by the SQL Server account. The default locations are based on the Designer installation folder. Note that any existing backup files will not be overwritten.

3. Enter the **Filestore details**.

Use a different filestore location for each repository.

4. Click **Create**. If a list of repositories has been set up (see below) you will be given the option of adding this new repository to it. The log window will show the various stages of the installation.

Upgrading an existing repository

You can use the Repository Configuration Tool to upgrade an existing repository.

1. In the Repository Setup page, click **Upgrade an existing repository**.
2. Backup or compact the database if required.

Setting the option to **compact the database** will defragment the database and rebuild its index. This will reduce the size of the database, but can take some time, especially for large repositories.

Note: If you have large amounts of data use the compact option with caution as compacting the database may slow down performance when using Designer.

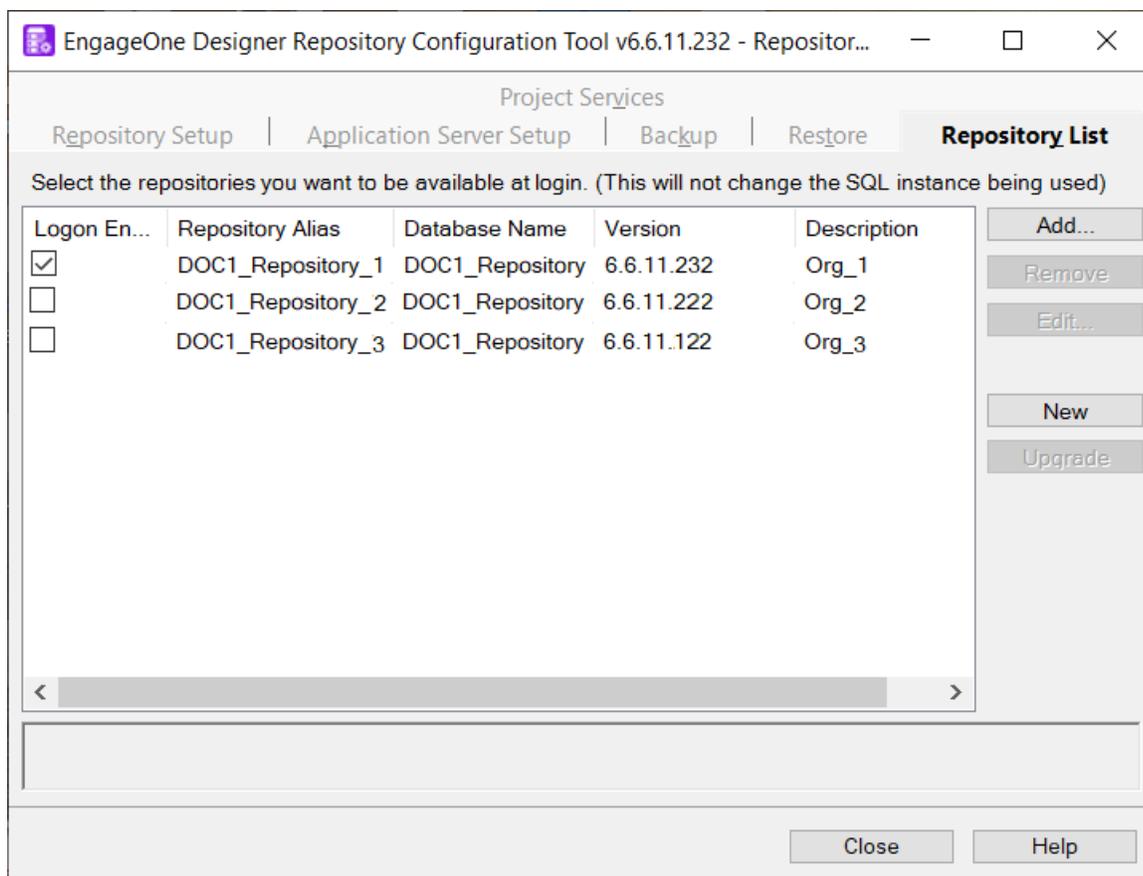
3. Select a database to be upgraded from the drop-down list. Use the browse buttons to change the backup folder locations.
4. Click **Upgrade**. The log window will show the various stages of the installation.

Repository selection configuration

Designer can be run with different repositories – for instance, you could have separate ones for production, training and evaluation. The repositories must all be on the same SQL server that is accessible from Designer. You can select which repository to use when logging on to Designer.

To create a list of repositories to be available at login: select the **Repository list** tab. Add any relevant repositories to the list, and select only those that are to be available at login.

Note that repositories appear shaded when they are out of date. If you want to select one of these you must first upgrade to the latest version. See [Setting up the repository](#) on page 639 for details.



- The repositories selected will be available when logging in to Designer.
- Add repositories giving them a unique **repository alias** name.
- Removing a repository does not remove the database itself.
- You can edit the **alias** name and **description**.
- Create a **new** repository, which can be added to this list, or **upgrade** the highlighted repository to the current version.
- The **repository alias** name and **description** will be shown at login.

Running repository backup

Backing up the repository will take a snapshot of the entire repository. This can provide protection in case of data loss or corruption.

Note that you must ensure that no users are logged onto the repository when the backup is run particularly in a Client/Server environment.

The backup process can create the following files:

.dbk – this file contains the contents of the database

.zip – this file contains the contents of the filestore and when the archive option is selected it contains both the database and filestore backups

The filenames for the backup files are controlled by Designer and cannot be set by the user. A unique sequential number is added to the name to differentiate between backups.

The filenames use the following structure: **RepositoryName_Backup#.%1**

For example: DOCRepository_3.zip

To backup the repository:

1. Ensure that the Repository Configuration Tool is connected to the relevant server.
2. Select the **Backup** tab and from the **Database** list select the database that you want to backup.
3. Click the browse button to select a location where the database and filestore contents will be placed for backup.
4. To create a separate zip file containing the database and filestore contents select **Copy Backup Files to Archive** and then click the browse button to select a location for the separate compressed archive of the database.
5. Click **Backup** and the designated backup files are created.

Restoring the repository

You must ensure that no users are logged onto the design repository when the database is being restored – particularly in a Client/Server environment. Some of the restore options will only be available if you use SQL Server for your database.

To restore the repository:

1. Ensure that the Repository Configuration Tool is connected to the relevant server.
2. Select the **Restore** tab and from the **Database** list select the database that you want to restore– or –Enter the name of a new database that you want to restore.

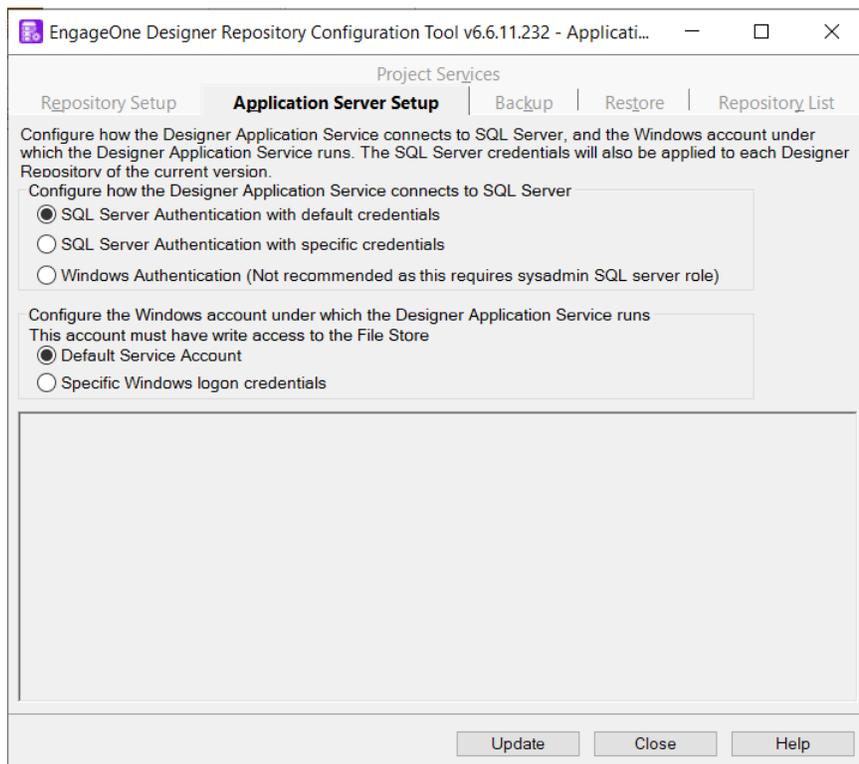
Note that you cannot use a universal naming convention (UNC) filepath with the backup files when restoring a repository.

3. Select the required files for the restore. Click the browse button to select the files for the following:
 - **Database Location** – the location where you want to restore the database
 - **Database Backup File** – the database (.dbk) file that you want to restore
 - **Filestore Location** – the location where you want to restore the filestore
 - **Filestore Backup File** – the filestore (.zip) file that you want to restore
4. Compact the database if required. Use with caution – see [Setting up the repository](#) on page 639 for details.
5. Click **Restore** and the database and filestore will be restored.

Application Server setup

This allows you to configure how the Designer Application Service connects to SQL Server, and the Windows account under which the Designer Application Service runs.

To configure server connection: select the **Application Server Setup** tab. Please note that any changes made here are applied to all repositories that have been upgraded to the current version. To upgrade a repository to the latest version see [Setting up the repository](#) on page 639.



Configure SQL Server Connection

SQL Server Authentication with default credentials

Configures the Designer Application Service to connect to SQL Server using the default SQL Server login name and password.

SQL Server Authentication with specific credentials

Configures the Designer Application Service to connect to SQL Server using a SQL Server login name and password of your choice.

To configure the SQL Server login name and password, click Update and enter the credentials in the subsequent dialog.

Windows Authentication

This enables Designer to connect to the SQL Server with Windows user credentials. This must be a valid Windows user that has permissions to access the SQL Server. You cannot use this option in connection with **Default Service Account** explained below. This type of authentication is not recommended as it requires that the trusted connection has the sysadmin SQL Server role.

Note that if SQL Server is not running in Mixed Authentication mode, enter here the Windows user credentials under which SQL Server service is running.

Configure Designer Application Service

Configures the Windows account under which the Designer Application Service runs. Note that this account must have write access to the file store.

Default Service Account

Sets the Designer service to run with the Network Service account that Windows uses.

Specific Windows logon credentials

Use a valid Windows user account that will be used to run the Application service. This can be used to set the Application Service to have the same logon credentials as the filestore.

Project services setup

To enable the Design Review and Approval feature you need to configure **Project Services** settings during installation of Designer. The Project services configuration enables communication between Designer and Interactive.

To configure project services: select the **Project Services** tab.

1. If you are working within a multiple repository environment, select the required **Repository**.
2. Click in the **Master project name** field and select from the list of projects in your repository. This will define the master project from which managed projects will be branched.
3. Click on **Import** to browse for the properties file supplied by your EngageOne Server system administrator.

EngageOne Designer Repository Configuration Tool v6.6.11.232 - Project S...

Repository Setup | Application Server Setup | Backup | Restore | Repository List

Project Services

Repository Settings

Repository:

Master project name:

Service Settings

Local service URL:

Key store:

Trust store:

Alias:

Project services URL:

Security services URL:

Services user name:

Server settings imported successfully.

Note: That the properties file will usually be available from your EngageOne Administrator. If you have imported a valid properties file, the necessary **Service Settings** will be populated automatically. Click **Update** to finalize the configuration. When an invalid properties file is imported, information to help you identify invalid settings is presented in the message area under the **Service Settings** entry fields.

The table that follows cross-references the fields presented in the Project Services tab with the settings in the properties file

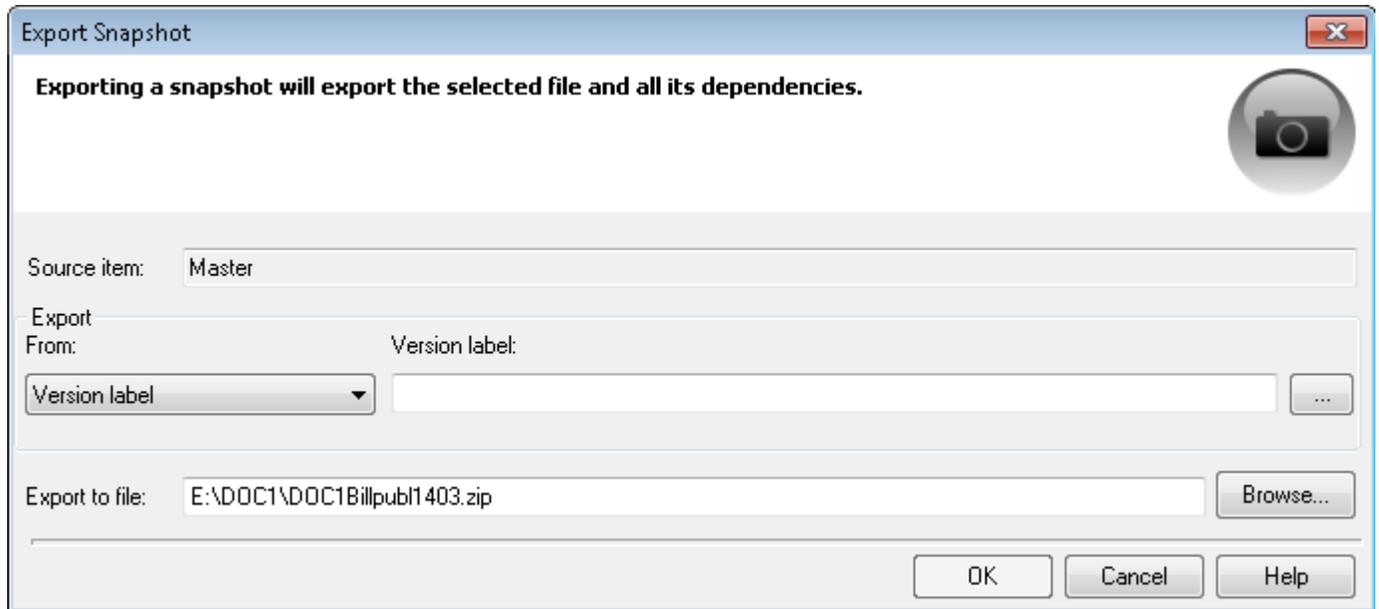
Project Services tab	deploy.properties field	Comments
Local service URL	designer.services.url	
Key store	tls.key.store.location tls.key.store.password	Optional, configured if your organization uses TLS security. If configured an encrypted password is set in the properties file but is not visible on the Project Services tab in Designer.
Trust store	tls.trust.store.location tls.trust.store.password	Optional, configured when using TLS security. If configured an encrypted password is set in the properties file but is not visible on the Project Services tab in Designer.
Alias	designer.tls.key.alias	Optional, configured when using TLS security
Project services URL	core.services.url	
Security services URL	security.services.url	
Service user name	system.user.username system.user.password	An encrypted password is set in the properties file but is not visible on the Project Services tab in Designer.

Snapshots

The snapshot feature allows you to capture and export a project or any of the objects within it such as a folder, a publication or Active Content. You can capture the current state or any previously labeled version. The snapshot can then be imported into a Designer repository as a new project. This enables you to copy objects from one Designer repository to another; for instance when you need to transfer resources to Precisely Support for analysis. All resources that are used by the object are also included.

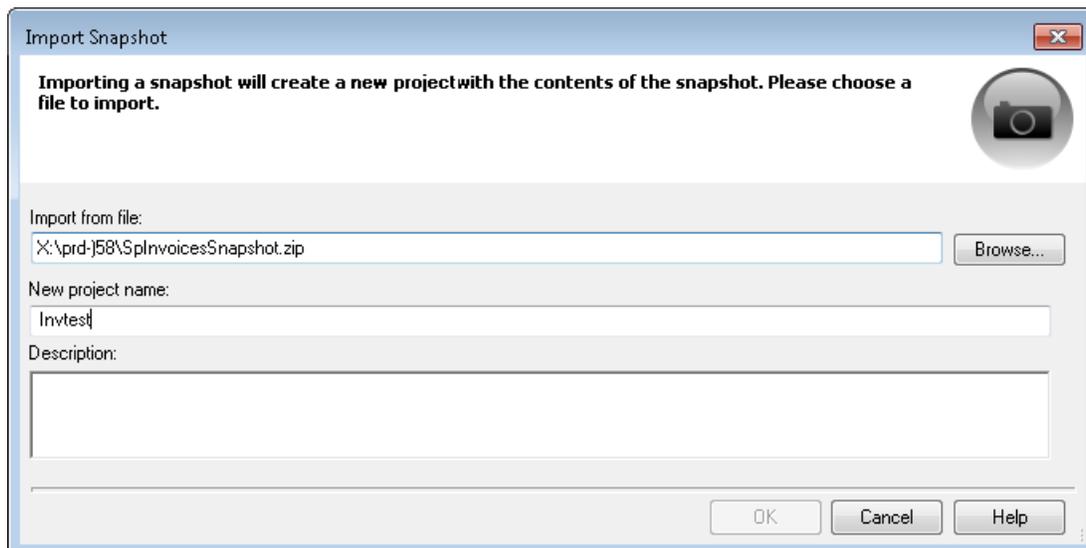
When importing data dictionary objects that already exist in your repository, it is important to note that in order to preserve data integrity, deleted data aliases and records are never removed from data dictionary. Note that import can only be run when there are no other users logged on, and that users will not be allowed to logon during the import process.

To export a snapshot: in the navigator browse to the required project or folder and select the object that you want to export. From the **Tasks** menu select **Export/Snapshot**. Use the **Export Snapshot** dialog box to specify and export the selected object.



- This object and all its dependencies will be zipped up and exported to here.
- You can export the latest version or browse for a previous version.
- When exporting the latest version you can use the default system label or specify your own label.

To import a snapshot: on the **Tasks** menu select **Import/Snapshot**. Use the **Import Snapshot** dialog box to locate and import a snapshot. Note that you can only import a snapshot created by Designer. **SUPPORT:** If you are using a filestore on a different machine – such as in a distributed installation environment – then you must ensure that the Application Service can access the filestore – see [Application Server setup](#) on page 644



- A new project is created for each imported snapshot.

Care must be taken when importing Active Content to ensure that all dependent publications are published. If the Active Content is independently published then it also must be published.

42 - Appendix A Glossary

ABCDEFGHIJKLMNOPQRSTUVWXYZ

In this section

2of5.....	657
3of9.....	657
A.....	657
Action.....	657
Active Content.....	657
ACU.....	658
Address block.....	658
Administration.....	658
Adobe font metrics.....	658
Advanced preview.....	658
AFP (Advanced Function Presentation).....	659
AFPDS (Advanced Function Presentation Data Stream).....	659
Anchor.....	659
Anchored.....	659
Application Import utility.....	659
Arithmetic.....	660
ASCII.....	660
Asset.....	660
Audit trail.....	660
Axes, axis.....	660
<i>B</i>	661
Background.....	661
Bar.....	661
Barcode.....	661
Base 14 fonts.....	661
Batch publishing.....	662
Bidirectional text.....	662
Bitmap or BMP.....	662
Body/body area.....	662
BookBill.....	662
Bookmark.....	663
Bounding box.....	663
Branch.....	663
Browser.....	663
Bundle.....	663



C.....	664
Carriage return.....	664
Case.....	664
Casting.....	664
Cell.....	664
Centralized printer.....	665
Character set.....	665
Chart.....	665
Chart element.....	665
Checkpoint.....	665
Child object.....	666
Clickable image.....	666
CMYK.....	666
Code128.....	666
Coded font.....	666
Code page.....	667
Column.....	667
Column layout.....	667
Communicate account.....	667
Composite resource.....	667
Concatenation.....	668
Condition.....	668
Conditional paragraph.....	668
Constant.....	668
Content Author.....	668
Control object.....	669
Cross browser.....	669
Currency.....	669
<i>D</i>	669
Data alias.....	669
Database.....	670
Data dictionary.....	670
Data field.....	670
Data Flow.....	670
Data format & Data Format Editor.....	670
Data map.....	671
Data input user exit.....	671
Data record.....	671
Date.....	671
DBCS.....	671
Default page header/footer.....	672
Deployment.....	672
Design review and approval.....	672
Delimited data.....	672
DFE.....	672
Dictionaries.....	673
DIJ - Document Interchange Journal.....	673

DIME.....	673
DOC1ACU.....	673
DOC1DIME.....	673
DOC1GEN.....	674
DOC1Publish.....	674
DOC1RPU.....	674
Document.....	674
Document area.....	674
Document Editor.....	675
Document logic.....	675
Document section.....	675
Dot leader.....	675
Double byte fonts.....	675
Dpi.....	676
Draft.....	676
DSC comments.....	676
<i>E</i>	676
EBCDIC.....	676
Editable text.....	677
eHTML.....	677
Embedded resources.....	677
EMFE (Extract Merge and Format Engine).....	677
EngageOne.....	677
EngageOne Communicate.....	678
EngageOne Compose ActiveX Editor.....	678
EngageOne Compose Interactive Editor.....	678
EngageOne Deliver.....	678
EngageOne Video.....	678
EPS.....	679
Eye-catcher.....	679
External document.....	679
Express Batch.....	679
<i>F</i>	679
Field.....	680
Filestore.....	680
Fixed anchor.....	680
Folder.....	680
Font.....	680
Font scaling barcode.....	681
Font subsetting.....	681
Footer.....	681
Form.....	681
Formatting marks.....	681
Formula.....	682
<i>G</i>	682
Generate.....	682
Governor field.....	682

<i>H</i>	682
Hard return.....	683
Header.....	683
HIP.....	683
Host.....	683
HTML.....	683
<i>I</i>	684
Image.....	684
Indentation.....	684
Indexes.....	684
In-line anchor.....	684
Input data.....	685
Insertion point.....	685
Integer.....	685
Interactive Data Editor (IDE).....	685
Interactive variable.....	685
<i>J</i>	686
Journal.....	686
<i>K</i>	686
Keycodes.....	686
Key field.....	686
Key map.....	687
Keyed object.....	687
Keyed record data.....	687
Keyed user exit.....	687
<i>L</i>	687
Legend.....	688
Lines.....	688
Line data.....	688
Line drawing barcode.....	688
Locale.....	688
Lock/unlock.....	689
Logic map.....	689
Logical page.....	689
Logo.....	689
Lookup table.....	689
Lookup table user exit.....	690
Loop.....	690
Label.....	690
<i>M</i>	690
Managed project.....	690
Mandatory records.....	691
Margin.....	691
Margin note.....	691
Merge.....	691
Message.....	691
Message area.....	692

Messages file.....	692
Message box.....	692
Message stream.....	692
Message audit trail.....	692
Message environment file.....	693
MVS.....	693
<i>N</i>	693
Navigator.....	693
Nested table.....	693
Non-repeating data.....	694
Note.....	694
Number.....	694
<i>O</i>	694
OpenType font.....	694
OPS (override production settings).....	695
Orphan.....	695
Output datastream.....	695
Output device.....	695
Outsorting.....	695
Overlay.....	696
Owner password.....	696
<i>P</i>	696
Page check.....	696
Page setup.....	696
Paragraph.....	697
Parameter.....	697
PCE.....	697
PCL.....	697
PDF, PDF/A.....	697
Pel (picture element).....	698
Physical page.....	698
Pica.....	698
Point.....	698
Post Composition Engine.....	698
PostNet.....	699
Postscript.....	699
Preview.....	699
Privilege.....	699
Production job.....	699
Project.....	700
Project services.....	700
Prompt group.....	700
Prompt layer editor.....	700
Prompt loop.....	700
Public.....	701
Publication.....	701
Publication data set.....	701

Publication Editor.....	701
Publish.....	701
PURL.....	702
<i>R</i>	702
Rasterize.....	702
RDW (record descriptor word).....	702
Record.....	702
Remaining space.....	703
Repeating data.....	703
Repository.....	703
Repository Configuration Tool.....	703
Reserved area.....	703
Reserved area entry.....	704
Resource map.....	704
Return code.....	704
Revert.....	704
Revision.....	704
RGB.....	705
RPU (Resource Pack Utility).....	705
RTF (Rich Text Format).....	705
Rollback.....	705
Row.....	705
Rulers.....	706
<i>S</i>	706
Sample data.....	706
Scope.....	706
Section.....	706
Segmented publishing.....	707
Set.....	707
Shape.....	707
Sheet.....	707
Short Message Service (SMS).....	707
Snapshot.....	708
Soft return.....	708
Sort.....	708
Spell checking.....	708
Start of publication record.....	708
Static table.....	709
String.....	709
Substring.....	709
Symbol.....	709
Symbolic reference.....	709
System variable.....	710
<i>T</i>	710
Tag logical element (TLE).....	710
Template.....	710
Text box.....	711

Time.....	711
Transaction table.....	711
TrueType font.....	711
<i>U</i>	711
Unlock.....	712
URL.....	712
User.....	712
User exit.....	712
User group.....	712
User password.....	713
<i>V</i>	713
Variable.....	713
Value.....	713
Vault.....	713
Vector offset.....	714
VPS.....	714
VSAM.....	714
<i>W</i>	714
Widow.....	714
<i>X</i>	715
XML.....	715
<i>A</i>	715
Zoom.....	715

2of5

Type of barcode.

3of9

Type of barcode.

A

[Back to top](#)

Action

Designer functions that change the way an application is being processed at run time.

Active Content

Active Content files contain segments of document content and logic and provide a highly flexible way to share and reuse such segments across multiple publication designs.

ACU

This provides the ability to extract AFP codepage information within a HIP file.

Address block

Design object that automatically formats name and address data into a style that is the standard for a named country.

Administration

The view where the administrator can create and maintain users and administer **repository** tasks.

Adobe font metrics

The font metrics used with PostScript fonts.

Advanced preview

Runs the current publication through Generate and displays the resulting output into a window within the Designer workspace. This task differs from the regular preview in that the preview process can be customized include runtime options such as supporting Vault, Message1, Content Author, lookup tables, technical support keywords and the ability to edit PDF output device settings .

AFP (Advanced Function Presentation)

AFP is IBM's print and presentation architecture. Also the name sometimes given to a printer that this architecture supports.

AFPDS (Advanced Function Presentation Data Stream)

The output datastream of the AFP environment.

Anchor

The method of position objects within a publication design.

Anchored

Anchor method where graphics are positioned relative to paragraphs within the document design.

Application Import utility

Allows publication designs to be imported from other applications into the Designer. The external design must be defined in XSL (Extensible Stylesheet Language) format. See the Designer Utilities guide.

Arithmetic

Designer function that performs calculations within publication designs.

ASCII

The character coding system used in Windows and UNIX operating systems.

Asset

In EngageOne, assets contain data on design, layout and logic.

- Assets can be part of a publication, and independently publishable active content.
- Assets can be used by the Interactive editor, and can be part of a non-accumulated batch.

Audit trail

See [Message audit trail](#)

Axes, axis

The lines on a chart that provide a scale references.

B

[Back to top](#)

Background

A design area whose contents are placed behind the main document layer. The background is often used for a watermark or similar.

Bar

An element on a bar chart

Barcode

A code normally consisting of a series of vertical lines or bars of varying thickness which can be read by an optical scanner.

Base 14 fonts

A set of standard fonts that is available with most PostScript and PDF printers/ browsers.

Batch publishing

Publishes Designer resources ready for production from a command prompt.

Bidirectional text

A combination of both left to right and right to left text in a paragraph.

Bitmap or BMP

A computer image format that is the standard working format within Designer. You can convert a wide range of other image formats to BMP.

Body/body area

The primary area in a document design where content is added.

BookBill

A physical media layout that includes a payment slip or similar as an extension to the first physical page of a publication design.

Bookmark

A Designer design object used for generating bookmarks within PDF output.

Bounding box

A rectangular shape area that encompasses all elements of a presentation object.

Branch

A copy of a project, folder or object in the navigator plus all related objects, making a new project that is completely independent of the original.

Browser

A program that allows the on-line viewing of documents such as Acrobat Reader for PDF files.

Bundle

Design object that allows you to temporarily withdraw other objects from the design scope.

C

[Back to top](#)

Carriage return

See [Hard return](#) on page 683.

Case

Design object that selects one of a range of conditional paths when a comparison value is matched.

Casting

Converting one data type to another, for instance, using a string containing 121205 as a date.

Cell

The basic element of a table. Combines to make rows and columns.

Centralized printer

The name sometimes given to a type of Xerox printer.

Character set

One of three font files used in the IBM AFP architecture, a character set contains the actual raster patterns for the characters.

Chart

A graphical representation of numeric application data. Designer supports bar, pie, line and scatter charts.

Chart element

Used to represent a single numeric value within a chart, e.g. a single bar in a bar chart.

Checkpoint

A means by which Generate can report, at specific intervals, which publication data set is currently processing.

Child object

An object that is within another object in a logic map.

Clickable image

An image defined as a URL link (for PDF, HTML and eHTML datastreams only).

CMYK

Color model expressed as cyan, magenta, yellow and key (black) values.

Code128

Type of barcode.

Coded font

One of three font files used in the IBM AFP architecture, a coded font combines a character set and code page.

Code page

One of three font files used in the IBM AFP architecture, a code page maps code points used in output data with actual characters in a character set.

Column

1. Consists of any number of **cells** aligned vertically and is used with **rows** to create a table. Also used to specify the layout of a table on the page, for example, a three column table will split the table into three, side by side.
2. A rectangular section within the body area of a page for positioning main document content. Can have up to a maximum of eight. Defined using a **column layout**.

Column layout

Used to define the layout of **columns** on a page.

Communicate account

Cloud storage for EngageOne Compose products. In the context of EngageOne Designer, publications can be uploaded to your Communicate account for onward processing by EngageOne Communicate.

Composite resource

Any design object that contains other design resources, for instance a publication that contains document designs.

Concatenation

Joining of two or more text values together to form a single string.

Condition

Design object that queries an expression and executes either a True or False logic map path depending on the result.

Conditional paragraph

A paragraph that has criteria associated with it defining when it should be included in the document output.

Constant

Provides non-variable data to objects.

Content Author

Content Author is an optional module of EngageOne Compose that allows key parts of publication content – messages – to be created and maintained independently of the main publication design.

It consists of a web browser client which provides an e-mail based workflow environment for creating and reviewing the messages, and a separate Windows desktop Administrator program for managing the Content Author environment and interfacing with Designer publications.

Control object

The data element that is used within a repeating data object.

Cross browser

Applies to interactive communications, where the final output is available for viewing in any supported browser. Refer to the EngageOne Server release notes for supported browsers.

Currency

A Designer data type used for monetary values.

D

[Back to top](#)

Data alias

A reference to a field or record in a publication design. Data aliases enable you to create a publication design without having sample data available.

Database

This is the part of the Designer **repository** that provides revision control.

Data dictionary

A group of existing data aliases.

Data field

A field from input data used in a publication design. This is an alias to the actual field and is mapped to the appropriate data element before the publication is put into production.

Data Flow

An integration solution that allows the selection and joining of data from multiple sources into a single data feed.

Data format & Data Format Editor

A data format describes the type and structure of the input data intended to be used with a particular production job. It is created using the Data Format Editor (DFE) in Designer.

Data map

Used to map data aliases from a particular data dictionary to real data in a data format.

Data input user exit

A type of user exit that takes control over locating, parsing and delivering data to Generate.

Data record

A record from input data used in a publication design. This is an alias to the actual record and is mapped to the appropriate data element before the publication is put into production.

Date

A Designer data type for calendar dates.

DBCS

Double byte character set. This term is used to describe fonts that have more than 255 characters and which therefore cannot be uniquely identified with a single byte – typically the case with Chinese, Japanese, Korean and other pictorial languages.

Default page header/footer

Used to define the height of headers and footers in the current document and to format how these areas appear on the page.

Deployment

The process of sending publication resources to a production system for subsequent processing by Generate.

Design review and approval

A Designer feature that provides a framework to ensure new and existing publications are subject to quality control as part of a structured review and approval process driven from the EngageOne portal.

Delimited data

A structure of input data supported by Designer. This is a variation of keyed record data where field locations are automatically identifiable to Designer by the presence of one or more consistent delimiter characters between each field.

DFE

See Data Format Editor.

Dictionaries

1. The reference files used by the spell checking and automatic hyphenation features.2. See [data dictionary](#).

DIJ - Document Interchange Journal

Provides an index of documents within an output datastream for use with Vault and EngageOne Deliver.

DIME

Dynamic Internet Mailing Engine. This utility is provided to extract the individual documents from a PDF stream generated by Generate and, optionally, to e-mail them.

DOCTACU

See ACU (AFP Codepage Utility).

DOCTDIME

See [DIME](#).

DOC1GEN

See [Generate](#).

DOC1Publish

See [batch publishing](#).

DOC1RPU

See [RPU \(Resource Pack Utility\)](#).

Document

The basic design unit of a publication. Typically, publications have only a single document design which is made available automatically when the publication is created. However, you can create additional document designs for conditional selection or reuse within a range of similar publications.

Document area

A logical area that defines the dimensions within which messages are designed in Message1. Messages associated with a document area can appear multiple times in the same document.

Document Editor

The main Designer document layout tool.

Document logic

The sequence of objects and associated logic that determines processing of a publication in the production environment.

Document section

See [section](#).

Dot leader

A row of dots between the end of preceding text and a tab stop, typical of table of contents and similar styles.

Double byte fonts

See [DBCS](#).

Dpi

Dots per inch – used to indicate the resolution of a printer, display device or image.

Draft

The status of a Designer design resource before it is saved.

DSC comments

Optionally included in the output datastream to communicate with PostScript document management software.

E

[Back to top](#)

EBCDIC

The character coding system used on IBM z/OS operating systems.

Editable text

Applies to interactive communications, allowing users to amend or insert parts of paragraph text.

eHTML

HTML for e-mail, this driver provides a simplified version of HTML, typically used for desktop and mobile device e-mailing.

Embedded resources

Fonts and image resources that are actually contained within an output datastream rather than being referenced.

EMFE (Extract Merge and Format Engine)

The main production engine program of the DOC1 Suite 4 (and earlier) environment.

EngageOne

EngageOne is an interactive document production system, which enables the creation and delivery of ad-hoc customer correspondence.

EngageOne Communicate

Organizations use EngageOne Communicate to create, send and analyze the success of email or SMS campaigns. Communications can be created from Designer publications, and can then send multiple campaigns from one or more communications.

EngageOne Compose ActiveX Editor

A fully featured editor available for editing interactive data in the EngageOne environment. This editor provides an accurate view of the final printed document. An ActiveX component must be installed for this editor. Note that it can only run in Internet Explorer.

EngageOne Compose Interactive Editor

A desktop program that is installed on your computer. This fully featured editor is available for editing interactive data in the EngageOne environment. This editor is not ActiveX based and can run on a variety of browsers.

EngageOne Deliver

The EngageOne Deliver system manages e-mail and SMS (Short Message Service) customer communications.

EngageOne Video

An optional addition to EngageOne Compose that allows for the creation of personalized, data driven video content to be included in the design of a document.

EPS

Encapsulated PostScript file that can be imported into a publication. EPS2 is a Level II Encapsulated PostScript file which includes support for JPEGs in their original format.

Eye-catcher

A sequence of bytes used to provide a human-readable reference in input data records.

External document

A pre-composed document in a supported external format that can be inserted into a publication.

Express Batch

Selecting this option significantly improves the performance of the EngageOne Non Accumulated batch processing when output channel processing is not required.

F

[Back to top](#)

Field

General term used for the various values that can be used in a document, including data field, variable and system variable.

Filestore

This is the part of the Designer **repository** that contains Designer files and other resources used when designing **publications**.

Fixed anchor

Anchor method where a graphic is positioned at a fixed offset on the page where it is inserted (or multiple pages if used on the **background**).

Folder

Groups together objects within a project in the navigator, that are used and controlled for a common purpose. Can contain other folders but not projects.

Font

A set of printable or displayable text characters in a specific style. In Designer you design publications using Windows fonts which are then converted (if necessary) to a format suitable for your intended output device.

Font scaling barcode

A type of barcode that provides scalability options and can be used with most output devices, except RTF, Linedata and HTML variants.

Font subsetting

Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by Generate, particularly when DBCS fonts are used.

Footer

An area at the bottom of a page or table containing any type of content. Can be specified to appear only in certain conditions, for example, only on the first page.

Form

The Xerox term for an overlay.

Formatting marks

Non-printing characters that indicate controls within paragraphs such as end markers, spaces, tab characters and so on.

Formula

Can be used as part of some design objects to specify more complex parameters than can be provided by dialog options.

G

[Back to top](#)

Generate

The batch program that processes production jobs on your chosen host system. It reads information about the job requirements from a HIP file, merges the input data file it receives with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

Governor field

Indicates the number of times data is repeated to a repeated data object.

H

[Back to top](#)

Hard return

Used to create a new paragraph by pressing the ENTER key.

Header

An area at the top of a page or table containing any type of content. Can be specified to appear only in certain conditions, for example, only on the first page.

HIP

Created by a publishing task, a HIP file contains all the information and resources required for a particular publication in the production environment. It also typically contains the resources required to print/present the resulting output datastream on your intended output devices.

Host

Provides properties for the computer where the Generate program will be run.

HTML

Hypertext Mark-Up Language is used for displaying information on the web.

/

[Back to top](#)

Image

A graphics file used by design files.

Indentation

The position of a paragraph in relation to the page margins.

Indexes

The references used to assign data to chart elements in some types of charts.

In-line anchor

Anchor method that positions graphics as a character within paragraph text.

Input data

The data file that is provided as input to a production job in the production environment.

Insertion point

In the logic map, this indicates the object after which the next object will be added.

Integer

A Designer data type. An integer with up to 9 digits.

Interactive Data Editor (IDE)

The IDE is a Windows application which allows prompts, help text and other attributes (metadata) of interactive data fields to be defined for EngageOne.

Interactive variable

A type of variable data included in the design of an interactive document. This type of variable information allows the document designer to prompt for data that is not part of the data model and does not originate from any business system.

J

[Back to top](#)

Journal

A file that records the activity of a publication when it processes in the production environment.

K

[Back to top](#)

Keycodes

The codes used to protect product software from unauthorized use. Various keycodes will be issued to the user depending on their software and hardware configuration.

Key field

In keyed record data this is the field that uniquely identifies each record type.

Key map

Maps keys for **keyed objects** to the actual image or Active Content when using resources that are in Designer. If images are supplied externally, it maps keys to sample images for use when designing, with the actual images being provided via an external control.

Keyed object

Keyed objects provide a means of allowing content (Active Content) and images to be selected dynamically at production time based on variable criteria or 'keys'.

Keyed record data

A structure of input data supported by Designer. This is effectively raw data although it must have a recognizable record structure. Each record type must be uniquely identifiable by the contents of a key field.

Keyed user exit

Returns the image to be placed as a **keyed object**.

L

[Back to top](#)

Legend

The key to the elements shown in a chart.

Lines

1. Drawing objects that can be used in document designs.2. The lines that connects two data markers in a line chart3. Optional rules on some chart types that provides an additional scale reference.

Line data

Organized text data with an implied structure that can be used by suitable input/output software but which does not contain any actual controls within the data itself.

Line drawing barcode

A type of barcode this is drawn without the use of fonts; native drawing methods are used instead, such as lines or rectangles.

Locale

Designer object that provides settings related to the customization of data and presentation objects to take account of regional variations.

Lock/unlock

In the Designer **repository** resources that are being edited are automatically locked. They are normally unlocked automatically but the user can choose to keep the lock when the resource is not open.

Logic map

An icon based view of the instructions you are generating as you add objects to a design.

Logical page

Generate produces logical pages within output datastreams. You can configure a job to place multiple logical pages on a single physical page.

Logo

A format of graphic used on Xerox printers where the image is stored as one or more font characters.

Lookup table

Keywords passed to a lookup table are replaced with the appropriate text string if a match is found.

Lookup table user exit

A user exit that overrides any existing lookup table functionality allowing an application to access reference tables not normally supported by Designer.

Loop

Design object that queries an expression and continuously executes a logic map path until it is no longer True or until a Break Loop is encountered.

Label

A unique identifier assigned to a project, folder or object in the navigator at a specific point in time – for example on a branch or merge, or when specifically applied by the user.

M

[Back to top](#)

Managed project

A project that is under review and approval control.

Mandatory records

Records in input data that must be included in every publication data set.

Margin

The space around the edge of a page that is not available as part of the body area.

Margin note

Typically used to present related feature information adjacent to content in the main document flow but not interrupting it.

Merge

Combination of changes in one project in the navigator into another project, with all differences being reconciled.

Message

An object created and controlled by Message1 or Content Author.

Message area

A logical area that defines the dimensions within which messages are designed in the message applications.

Messages file

The file (.HIM) containing Message1 or Content Author objects associated with a particular campaign. It contains all the information and resources required for messages in a production run.

Message box

Reserves fixed space in a document for messages from a message application environment.

Message stream

Allows messages from Content Author to be inserted directly into the flow of the page in the Designer publication.

Message audit trail

This is similar to a regular journal file but is specifically and automatically used to log messages that have been included within documents.

Message environment file

This contains information about the available message areas and message data fields. It is an XML file generated in the message application and is imported into Designer.

MVS

The previous name used for IBM's z/OS mainframe operating system.

N

[Back to top](#)

Navigator

A tree based view in Designer that shows all the items that you can access and work with to create publications.

Nested table

Used to include a table between major row elements in the detail section of another table.

Non-repeating data

Fields that typically appear only once in the input data for a single publication.

Note

An object that allows you to provide commentary within a publication design.

Number

A Designer data type, this is a decimal number with a maximum of 24 digits before the decimal place and 8 after.



[Back to top](#)

OpenType font

The OpenType font format is an extension of the TrueType font format, adding support for PostScript font data.

OPS (override production settings)

Used in the production environment, an OPS file allows you to specify additional or alternative file references and other settings to those use when publishing a publication design.

Orphan

A few lines at the end of a paragraph that are separated from the rest of the paragraph by a page break.

Output datastream

The main output of Designer applications. An output datastream contains the composed pages in the low-level format suitable for your target printer/browser.

Output device

In Designer an output device object is used to indicate the output datastream required by a particular job plus any device specific settings you may need.

Outsourcing

Splits the output from a single production job into multiple stream to support the needs of post-printing equipment such as enveloping machines or similar.

Overlay

An overlay is the term used by IBM for an electronic form. The Xerox equivalent of an overlay is a form (FRM) file.

Owner password

Protects the attributes assigned to a PDF file from unauthorized modification.

P

[Back to top](#)

Page check

A specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified.

Page setup

A document design function that defines the page size, headers, footers, **background** and new page logic.

Paragraph

A continuous piece of text which can include graphics, images, text boxes anchored to it. Has properties such as alignment, spacing, tabs.

Parameter

A value passed into Active Content.

PCE

The Post Composition Engine (PCE) handles the requirements of some applications for additional processing once document pages have been composed.

PCL

A control language used to control printers or other output devices based on the Hewlett Packard family of laser printers.

PDF, PDF/A

Portable Document Format is a file based document format suitable for Internet distribution and on-line viewing with a PDF viewer such as Adobe Acrobat Reader. PDF/A is a subset of PDF that is suitable for long-term archiving.

Pel (picture element)

In printing, the smallest area that can be individually toned by the printer.

Physical page

An actual piece of paper on a printer.

Pica

Unit of measure (UOM) used by the editor

Point

Unit of measure (UOM) used by the editor

Post Composition Engine

See [PCE](#).

PostNet

A type of barcode.

Postscript

A common page description language acceptable to many desktop printers and also, increasingly, for high volume production printers.

Preview

Runs the current publication through Generate and displays the resulting output into a window. This can then be sent to the local printer.

Privilege

Controls the level of access to items in the navigator for users and groups.

Production job

The production settings associated with a publication as saved in Designer.

Project

Groups together objects in the navigator that are used and controlled for a common purpose. Can contain folders but not other projects.

Project services

A service that enables communication between Designer and Interactive for the Review and Approval feature. Project Services settings are configured using the Repository Configuration Tool

Prompt group

The collection of multiple data values within a single prompt dialog box within an interactive document.

Prompt layer editor

Allows for the definition of data values used in a prompt group.

Prompt loop

An interactive template logic construct that can repeatedly prompt an interactive user to make a selection. Each selection made may provide additional prompts specific to the selection made.

Public

Property of some design resources indicating that they can be shared by multiple publications.

Publication

An assembly of one or more related **documents**.

Publication data set

The input data intended for an individual publication.

Publication Editor

Where you specify the documents that make up a publication and define publication properties.

Publish

A task that prepares a Designer job for the production environment. It creates the **HIP** file used by Generate to control processing.

PURL

A Personalized URL data field.

R

[Back to top](#)

Rasterize

A process used to convert a graphic into pixels for output.

RDW (record descriptor word)

A method of delineating records in input data.

Record

A segment of input data. A record is typically a line of data in the input file containing one or more fields.

Remaining space

The space available to content such as advertisements or promotional material once regular content placed in the body area.

Repeating data

A data element that is iterated within a single publication data set. Repeating data typically contains transaction type information

Repository

This stores and provides revision control for all the design resources and other objects that are used in Designer. These include resources that are created within Designer itself or have been imported into it from an external source. The repository consists of a **filestore** and a **database**.

Repository Configuration Tool

A stand-alone tool for configuring and managing repositories

Reserved area

Only available on the **background**, it defines space on the page that cannot be overwritten by content within the main document body which will flow around the reserved area as necessary.

Reserved area entry

A means of dynamically updating the content of a reserved area.

Resource map

A tailored mapping of resources created by a publishing task. A resource map overrides the default names given to the font and image files (and the references used to identify the resources within an output datastream).

Return code

Indicates the status of a completed instance of Generate. For example:

- 0 (zero) for successful completion or where warning messages (only) have been issued.
- 16 where a failure has occurred – i.e. where an abort message has been issued.

Revert

This function allows you to restore to an earlier revision of a non-composite design resource.

Revision

A version of a design resource.

RGB

Color model expressed as red, green and blue values.

RPU (Resource Pack Utility)

This provides the ability to extract and manipulate resources within a HIP file.

RTF (Rich Text Format)

A document mark-up language commonly used to exchange data and graphics between different applications. Can be imported into documents and is an output format supported by Generate.

Rollback

This function allows you to restore to an earlier revision of a composite design resource.

Row

Cells aligned horizontally within a table.

Rulers

Measurement guides that can be shown at the top and side of the Document Editor.

S

[Back to top](#)

Sample data

An extract from intended job input data. This is used as a template for defining a data format.

Scope

The level at which variable names are reserved. For example, a value in a variable with **Publication** scope is available throughout an entire publication.

Section

A sub-set of logic map objects within a document design. You can reuse sections within the same document.

Segmented publishing

A term used where the publication design is published separately from the output device configuration and resources. Any number of separate resource HIP files can be produced for use with Generate.

Set

Provides categorization in some chart types.

Shape

Lines, rectangles, rounded rectangles and ovals that can be drawn in a document.

Sheet

An actual media page as used on a printer or produced by an electronic document format. A sheet is the area on which the logical pages produced by Designer are placed.

Short Message Service (SMS)

A means of sending text messages to and from mobile phones.

Snapshot

Allows you to capture any version of any object – for example a folder, a publication, Active Content. This is exported to an external file which can then be imported into a Designer repository.

Soft return

Used to create a new line within the same paragraph by pressing the SHIFT + ENTER keys.

Sort

In Designer terms, the reordering of repeating data prior to its use with a publication design.

Spell checking

Verification of paragraph text using the dictionary of the assigned language.

Start of publication record

Data format property that identifies the data element that indicates the start of a new publication data set.

Static table

A regular table in a document design. It cannot contain repeating data.

String

A Designer data type that can contain any alphanumeric characters can be used up to a limit of 32k characters.

Substring

The substring function extracts one or more consecutive characters from within a text string.

Symbol

A character that cannot be accessed directly from the keyboard.

Symbolic reference

Parameters in an OPS file are dynamically resolved at run-time using symbolic values.

System variable

Information from the production environment that can be included in publication designs, for example current page number or date.

T

[Back to top](#)

Tag logical element (TLE)

An AFPDS structured field that is the functional part of an AFP index. TLEs can form part of an independent index file or be included directly in the composed pages to which they relate.

Template

Templates can be interactive or non-interactive.

- Interactive templates use the Interactive data model, and can be used with the EngageOne Interactive web application.
- Non-interactive templates employ the Designer data model. They are used with the EngageOne batch processes, as well as the interactive data model.

Both interactive and non-interactive templates can contain active-content, see [Active Content](#) on page 657

Text box

Acts as a container for specific page content and can contain both paragraph text and graphics. Can be used to highlight particular content or as a reserved area which other paragraph text can flow around.

Time

A Designer data type for time values.

Transaction table

Controls how repeating data is presented, i.e. how multiple iterations of the same repeating data structure are ordered and how they are positioned in relation to one another.

TrueType font

An Windows outline font where the characters are scalable.

U

[Back to top](#)

Unlock

See [lock/unlock](#).

URL

Uniform Resource Locator – the global address of a document or file on the Internet.

User

An individual given a user name and password, enabling them to work in the Designer environment.

User exit

User exits allow an application to call a user defined program function. Generate supports user exit programs coded in C. There are four types of user exit: data input, file, lookup table and keyed.

User group

A list of users who share similar responsibilities. Groups are used for logical groupings of multiple users and enable the administrator to manage a number of users at one time.

User password

Used to protect PDF documents from unauthorized opening in a PDF browser.

V

[Back to top](#)

Variable

Variables store values for subsequent use. They can hold values of any supported data type, but each individual variable can only accept data of a single type.

Value

Values in document designs can be provided by data aliases, variables, system variables, the output from functions or you can simply enter a constant.

Vault

The Vault environment caters for high speed storage and retrieval of electronic documents. Vault Server itself is the main **repository** in which documents are stored and maintained. Various other Vault products provide access to the **repository**.

Vector offset

A reference to the start of individual composed documents within production job output.

VPS

Variable Print Specification. This is a PostScript based page description language developed by Scitex for use with VPS enabled printers.

VSAM

A data set structure used under z/OS.

W

[Back to top](#)

Widow

A few lines at the beginning of a paragraph that are separated from the rest of the paragraph by a page break.

X

[Back to top](#)

XML

XML – Extensible Markup Language – provides a common method of describing data for internet and other applications. Designer supports XML as a format for input data.

A

[Back to top](#)

Zoom

Magnification used in an editor.

[Back to top](#)

43 - Appendix B Formula expressions

Formulas can be used as part of some design objects to specify parameters more complex than can be provided by simple value controls or as a more convenient way of coding compound functions.

In this section

Working with formula conditions.....717



Working with formula conditions

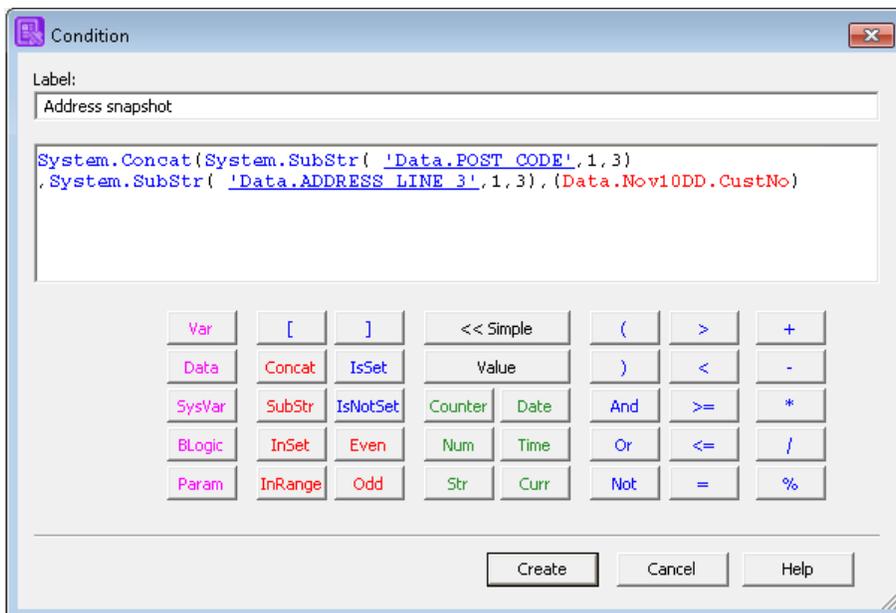
You can create a complex statement for a condition or use a formula code to code a calculation rather than using nested arithmetic objects.

Creating a complex statement:

```
SubStr(CN,0,1) = "V" And Pay > 35000
```

Using a formula to code a calculation rather than using nested arithmetic objects:

```
(GrossPay - Deduction) * TaxRate
```



In the **Advanced** view of the **Formula** dialog box you can use the various function buttons to help with coding the syntax.

Formula syntax

A Formula expression consists of a single statement. Compound statements are allowed but assignments and program logic are not supported. All keywords plus data and variable names are case sensitive.

Objects and functions are referenced using their name plus their type and a varying number of other qualifiers. You may exclude some or all qualifiers (always working left-to-right) where there is no possibility of duplication or ambiguity. For a data alias reference for example, you may have a fully qualified name of:

`Data.MainDic.CustomerAddress.CityName` Provided `CityName` is not used as a field or variable name elsewhere in the publication design you may simply code: `CityName`

If the field name is used in another record of the same data dictionary you can code: `Data.MainDic.CityName` You may not code: `CustomerAddress.CityName`

If a name contains spaces or characters used as part of the formula syntax itself the entire name must be contained within single quotes. For example: `'Var.Pub.Current Rate'`

Constants	Defines a value and an associated data type.
Counter	an integer 362436
Currency	C'\$9.99' or Currency'\$9.99'
Date	D'13/10/2003' or Date'13/10/2003'
Numbers	18.67
Strings	"Account balance"
Time	T'11:22:33' or Time'11:22:33'
Data Aliases	<p><code>Data.Dictionary.Record.[Record...] Field</code></p> <p>Returns the value of Field. Dictionary indicates the name of a data dictionary object. Record and Field are data elements within the dictionary. You may need to specify multiple record names where these are nested within the dictionary. All qualifiers other than Field are optional.</p>

Examples:

```
Var.Temp=Data.AnexDic.R1.R12.TaxesCurrency(Data.Taxes) IsNotSet Ref001Taxes
IsNotSet Ref001Taxes
```

System Variables

```
SysVar.Name
```

Returns the current value of a system variable. Name is one of:

RunDate

RunTime

GenerationDate

GenerationTime

PublicationSequenceNumber

DocumentPageNumber

PublicationPageNumber

JobPageNumber

See [System variables](#) on page 257 for details.

Example:

```
Var.Temp=SysVar.JobPageNumberSubStr(RunDate,1,4)
```

Variables

```
Var.Scope.Type.Name [index]
```

Returns the current value held in variable Name. If Name is an array index must be specified to indicate the element required. Scope and Type may be used to identify the variable uniquely if duplication is possible:

Scope – indicates the defined scope of the variable.

- Options are Job, Pub, Doc or Loc.

Type – indicates the data type of the variable.

- Options are Counter, Num, Str, Date, Time or Currency.

Examples:

```
Var.Temp=Var.Job.Str.Addr[2]Date(CDateString)
```

```
IsSet Var.Time.Tdate
```

Conditions

Comparison operators:

When using a formula within a condition object the expression is evaluated using one of the following operators and TRUE or FALSE is returned.

><>=<==<>	Greater than Less than Greater than or equal to Less than or equal to Equal to Not equal to
IsSet object	returns TRUE if object is a data alias that is available in the current data set or is a variable currently in scope.
IsNotSet object	returns TRUE if object is a data alias that is unavailable in the current data set or is a variable that is not currently in scope. Examples: <code>Var.DayCode = 7</code> <code>Data.Received > D'13/10/2003'</code> <code>Var.Option <> "Gold"</code> <code>IsNotSet Var.CountR</code>
Logical Operators:	You may also use these operators within conditions to join expressions and return the result. Note that any function that returns TRUE as its result (such as InRange, InSet, Even, Odd) must specifically test for "= TRUE" when the function is used as part of a formula using the logical operators.
ANDORNOT	returns TRUE if both expressions are TRUE returns TRUE if either expression is TRUE returns TRUE if an expression is FALSE. Examples: <code>Data.Tax > 10 AND IsNotSet Var.CountR</code> <code>(System.InSet (Var.Code, 312, 315, 317) = TRUE)</code> OR <code>(System.InRange (Var.Code, 320, 350) = TRUE)</code>
Functional Operators	Use these operators to perform calculations of input values. You may calculate dates within the restrictions of the regular Designer arithmetic function. See Date Calculations for details.

+	addition
-	subtraction
*	multiplication
/	division
%	modulo (remainder)
	Examples:
	<code>Var.Job.Str.Addr[2] + 10 SysVar.RunDate -"2M"</code>

Functions The following functions can act on values supplied to a formula and return a value. Use the qualifier `System` if necessary to ensure their name is uniquely identified.

Concat Concatenate multiple strings

```
System.Concat(string1, string2...)
```

Returns the input strings as a single concatenated string.

Example:

```
Concat("Today is ", SysVar.RunDate)
System.Concat(Var.T1, Var.T2, Var.T3)
```

Even `System.Even(counter)`

Returns TRUE if counter is even.

Example:

```
Even(Var.MonthCode)
```

InRange Search a range of values for a match.

```
System.InRange(target, low, high)
```

Returns TRUE if the target is between low and high. Any data type can be used with `InRange` except strings but all values must be of the same type.

Example:

```
InRange(Var.Tcode, 1, 20)
System.InRange(Rating, 5, MaxRating)
```

InSet

Search input values for a match.

```
System.InSet(input, comp1, comp2 ...)
```

Returns TRUE if input matches any of the comparison values. You can have as many comparison values as necessary. All parameters must be of the same data type.

Example:

```
InSet(Var.Tcode,1,2,7,10)
```

```
System.InSet(Rating,6,9,MaxRating)
```

Odd

```
System.Odd(counter)
```

Returns as TRUE if counter is odd.

Example:

```
Odd(Var.Pub.Counter.Weekday)
```

SubStr**Substring**

```
System.SubStr(source, offset, length)
```

Returns a substring of string source starting at position offset and continuing for length bytes.

Example:

```
SubStr(Var.FullAddress,1,20)
```

```
System.SubStr(RunDate,6,8)
```

Casting

```
System.type (input)
```

Returns input converted to the specified data type where possible. Type can be:

Currency

Date

Counter – integer, a whole number up to 9 digits

Num – number

Str – string

Time

Note that, wherever possible, data types will automatically be converted to meet the requirements of a function. You need only code explicit casting where there is ambiguity. The format of the return value (from both explicit and implicit casting) is based on the Designer locale that is active when the function is called. See [About locales](#) on page 311 for more information. Use the qualifier System if necessary to ensure the function is uniquely identified.

Examples:

```
System.Date("06/07/2005")
```

```
Num(Var.TCode)
```

Active Content

Blogic

Call Active Content function

```
BLogic.Name (parm1, parm2, ...)
```

Returns the return value of Active Content function Name which is called with any required parameters. See [Active Content](#) on page 428 for more information.

Example:

```
BLogic.GetRate(3) = Var.MaxRating
```

Param

Accesses Active Content parameter.

`Param.Name`

Use only within an Active Content function itself. Returns the value of parameter Name which must be a parameter passed to the function.

Example:

`InRange(Param.Rating, 1, 5)`

44 - Appendix C

Interchange Journals

A **Document Interchange Journal** DIJ is an XML file that records information about the documents within a Generate output datastream. Each record within a DIJ specifies a unique key for each document, such as an account number. This can be used by other products, such as Vault or EngageOne Deliver, as an index into the output datastream. Note that each product will make use of a different set of DIJ options.

In this section

Creating a DIJ.....726



Creating a DIJ

The DIJ data is created as part of the attributes of a publication.

To create a DIJ:

1. In the Document view, from the **View** menu select **Publication** or right-click in the logic map and select **View Publication** from the context menu.
2. Double-click the Publication icon in the logic map and the **Publication** dialog box is displayed. Select the **Journals** tab.
3. Select **Create Document Interchange Journal** and click **Edit DIJ**. The **Document Interchange Journal** dialog box is displayed.
4. Supply values for the required parameters. Note that if you wish to use the **Address** or **Pay Details** tabs you must set parameters in the **Options** tab first.

NOTE: All values are added using the regular document editor mechanism. See [Working with values](#) on page 170 for details.

DIJ dialog box

Data

These parameters are used to identify individual publications.

Account Number	A unique reference number for the intended recipient of the publication (usually an account number).
Name	A name that uniquely identifies the publication.
Statement Date	A date associated with the publication, such as creation date, date of usage.

Publication ID – for use by Vault

These options uniquely identify the company (vendor) and type of publication to which documents relate. Unless specifically required, use the **Default** setting. Options are:

Default	The references are automatically created. If you use this setting and are working with documents that relate to multiple vendors you must ensure the publication is stored within a project otherwise the vendor ID generated will not be unique.
---------	---

DIJ dialog box

Specify	<p>Selecting this option allows you to enter the ID references manually. You will need to supply data that uniquely identifies each category:</p> <p>Vendor ID – a reference that identifies a specific company;</p> <p>Publication ID – a reference that uniquely identifies the type of publication;</p> <p>Master ID – any additional unique references required for the application.</p>
Read from .PSE file	<p>Use this option if the relevant ID settings are available elsewhere. This will enable a Locate option where you can browse for the file.</p>

Options

You can specify which parameters will be available in other tabs in the DIJ dialog box.

Skipped Pages	<p>If your publication design produces banner pages that you do not want to be included use this parameter to indicate the page numbers that can be skipped in each document processed.</p>
Additional street address lines (0–5)	<p>If you want to include more than one line of main address information in the index use this option to set the number of additional lines required. The address is specified in the Address tab.</p>
City, State/Region, Zip/Post Code, Country, Phone	<p>Use these options to enable additional address fields in the Address tab.</p>
Payable	<p>This enables the parameters in the Pay Details tab – for use by Vault.</p>
Use Secondary Currency	<p>This option enables a second currency in the Pay Details tab if required – for use by Vault.</p>
Store Data Set	<p>Use this option to include the entire data set for each publication within the DIJ – for use by Vault.</p>

Address

Use this tab to provide the address data related to the account to which each document belongs. The options available in this tab are specified in the **Options** tab.

DIJ dialog box

Pay Details – for use by Vault

These parameters are enabled by the **Payable** and **Use Secondary Currency** settings in the **Options** tab.

Due Date	The date that the bill being generated must be paid by
Primary Currency	<p>A group of parameters:</p> <p>Currency – the currency in which the amount values are expressed. The currency parameter must be resolved with a three character BSI Currency Code (ISO 4217 Maintenance Agency).</p> <p>Due – the total amount due.</p> <p>Min Due – the minimum amount due if partial payment is acceptable.</p>
Secondary Currency	If your application supports two payment currencies use this parameter group to specify the currency and values of the secondary currency type. Parameters are as for Primary Currency .

Custom Settings

You can use custom settings to pass additional document information that may be required. Each entry provides you with a name and a value field for you to enter a user specified value to be included in the DIJ.

You must specify that the DIJ file is to be generated when the document is published.

To generate a DIJ: in the Publish wizard **Output Device** page, select the **Configure DIJ for Vault or EngageOne Deliver** option. This will enable you to specify the file name and path for the DIJ.

NOTE See [About publishing](#) on page 593 for more information.

Notices



Copyright © 1993, 2021 Precisely. All rights reserved

This publication and the software described in it is supplied under license and may only be used or copied in accordance with the terms of such license. The information in this publication is provided for information only, is subject to change without notice, and should not be construed as a commitment by Precisely. To the fullest extent permitted by applicable laws Precisely excludes all warranties, representations and undertakings (express or implied) in relation to this publication and assumes no liability or responsibility for any errors or inaccuracies that may appear in this publication and shall not be liable for loss or damage of any kind arising from its use.

Except as permitted by such license, reproduction of any part of this publication by mechanical, electronic, recording means or otherwise, including fax transmission, without the express permission of Precisely is prohibited to the fullest extent permitted by applicable laws.

Nothing in this notice shall limit or exclude Precisely liability in respect of fraud or for death or personal injury arising from its negligence. Statutory rights of the user, if any, are unaffected.

*TALO Hyphenators and Spellers are used. Developed by TALO B.V., Bussum, Netherlands Copyright © 1998 *TALO B.V., Bussum, NL *TALO is a registered trademark ®

Encryption algorithms licensed from Unisys Corp. under U.S. Patent No. 4,558,302 and foreign counterparts.

Security algorithms Copyright © 1991-1992 RSA Data Security Inc

Datamatrix and PDF417 encoding, fonts and derivations - Copyright © DL Technology Ltd 1992-2010

Barcode fonts Copyright © 1997 Terrapin Solutions Ltd. with NRB Systems Ltd.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Artifex and the Ghostscript logo are registered trademarks and the Artifex logo and Ghostscript are trademarks of Artifex Software, Inc.

This product contains the Regex++ library Copyright © 1998-2000 Dr. John Maddock

PostScript is a trademark of Adobe Systems Incorporated.

PCL is a trademark of Hewlett Packard Company.

Copyright (c) 2000 - 2015 The Legion of the Bouncy Castle Inc. (<http://www.bouncycastle.org>)

PStillDll (c) Dipl.-Ing. Frank Siegert, 2005-2018

PStill is a trademarked term, registered with the German patent and trademark office

This product contains RestSharp, version number 105.2.3, which is licensed under the Apache License, version number 2.0. The license can be downloaded from

<http://www.apache.org/licenses/LICENSE-2.0>. The source code for this software is available from <http://restsharp.org>.

This product contains Json.NET, version number 9.0.1, which is licensed under the MIT License. The license can be downloaded from

<http://github.com/JamesNK/Newtonsoft.Json/blob/master/LICENSE.md>. The source code for this software is available from <http://www.newtonsoft.com/json>.

ICU License - ICU 1.8.1 and later Copyright (c) 1995-2006 International Business Machines Corporation and others All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

This product contains Elasticsearch, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://www.apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch>.

This product contains Elasticsearch.Net, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch-net>.

This product contains NEST, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch-net>.

This product contains Antlr4cs Runtime, which is licensed under the BSD-3-Clause. The license can be downloaded from <https://raw.githubusercontent.com/tunnelvisionlabs/antlr4cs/master/LICENSE.txt>. The source code for this software is available from <http://www.antlr.org>.

This product contains Log4net. The license for log4net can be downloaded from <https://www.apache.org/licenses/LICENSE-2.0>. The source code for this software is available from https://logging.apache.org/log4net/download_log4net.cgi.

Otherwise all product names are trademarks or registered trademarks of their respective holders.

Support

Click [here](#) for full EngageOne Compose documentation and access to your peers and subject matter experts on the Knowledge community.



1700 District Ave Ste 300
Burlington MA 01803-5231
USA

www.precisely.com

© 1993, 2021 Precisely. All rights reserved.