

**precisely**

# Finalist

## Finalist Users Guide

Version 9.22.0



# Table of Contents

## 1 - Introduction to Finalist

---

What Is Finalist?.....	5
Finalist Benefits.....	5
Finalist Files.....	6
Input.....	10
Output.....	10
Finalist Auxiliary Processing.....	11

## 2 - Configuring Finalist

---

Using the pbfncfg File to Configure Finalist.....	14
Using the Workbench or Lookup Tool to Configure Finalist.....	35

## 3 - Interactive Address Lookup

---

Using the Windows Lookup Tool.....	50
Using Finalist CICS.....	56
Using Finalist IMS.....	79

## 4 - Defining a Job

---

Building Batch Jobs.....	88
Building Batch Jobs with the Workbench.....	122

## 5 - Running a Job

---

Running a Batch Job on z/OS.....	134
Running a Batch Job with the Workbench.....	137
Running a Batch Job via the Windows Command Line.....	137

Running a Batch Job via the Unix/Linux Command Line.....	139
--	-----

## 6 - Using the Finalist Reports

---

Address Detail Report.....	142
Finalist Batch Report.....	152
USPS Form 3553 (CASS Summary Report)....	176

## 7 - Finalist Error Codes

---

Error Codes (Returned Strings).....	180
Finalist Batch Driver Return Codes.....	182
DPV Error Messages.....	184
LACS <sup>Link</sup> Error Messages.....	185
Suite <sup>Link</sup> Error Messages.....	186
EWS Error Messages.....	186
eLOT Error Message.....	187

## 8 - AddrScan Option

---

What is the AddrScan Option?.....	189
Why Should I Use the AddrScan Option?.....	190
How Do I Install AddrScan?.....	191
How Does AddrScan Processing Work?.....	191
Understanding AddrScan Output.....	196
Calling AddrScan.....	197
Setting Returned Line Options.....	198
Defining Processing Options.....	203
Activating AddrScan Advanced Options.....	203
Using the Lookup Tool to Define AddrScan Processing Options.....	205
CALLAREA Structure Definition.....	208
COBOL Copybook Version.....	212
Using ADDRTBLS for Exceptions Processing...	214

Calling AddrScan With the Finalist Driver.....	215
Calling AddrScan With A User-Written Driver....	215
AddrScan Examples.....	216
C Driver Example.....	218
COBOL Driver Example.....	219

## 9 - Exceptions Table Option

---

What is the Exceptions Table Option?.....	224
Can I Use the Exceptions Table Option?.....	224
Does the Exceptions Table Option Affect CASS?.....	224
Why Should I Use the Exceptions Table Option?.....	224
How Do I Install the Exceptions Table Option?..	225
How Do I Activate Exceptions Table Processing?.....	225
How Does Exceptions Table Processing Work?.	230
Exceptions Table Input.....	231
Structures Containing Exceptions Table Information.....	241
Exceptions Table Output.....	241

## 10 - CASS Certification

---

What is CASS Certification?.....	243
Why is CASS Certification Important for Finalist?.....	244
Types of Certification.....	244
What is the Address Management System II (AMS II)?.....	245
National Customer Support Center (NCSC)....	245

# 1 - Introduction to Finalist

## In this section

---

What Is Finalist?.....	5
Finalist Benefits.....	5
Finalist Files.....	6
Input.....	10
Output.....	10
Finalist Auxiliary Processing.....	11



# What Is Finalist?

Finalist is a USPS CASS-certified software application that helps you manage the address information in your customer file(s). During processing, Finalist:

1. Corrects misspellings in firm, street, and city names.
2. Standardizes address elements including street names, directionals, suffixes, city names, states, ZIP Codes, and ZIP + 4 Codes and then adds the carrier route codes and delivery point barcodes you need to maximize your postal discounts in compliance with USPS Coding Accuracy Support System (CASS) regulations.
3. Compares each address to the Finalist data files to verify the accuracy of each address.
4. Corrects errors in the ZIP Code, ZIP + 4 Code, and carrier route codes. These codes are the key to USPS deliverability and minimizing your undeliverable-as-addressed mail. Precisely supplies customers with updated data files on a regular basis.
5. Performs additional CASS-required processing on your address file including Delivery Point Validation (DPV), SuiteLink, and LACS<sup>Link</sup> processing.
6. Perform additional Finalist auxiliary processing on your address file including AddrScan, Exceptions Table, Early Warning System (EWS), Residential Delivery Indicator (RDI), Enhanced Line of Travel (eLOT), and PreciselyID processing.

## Finalist Benefits

Some of the benefits available to you when processing with Finalist include:

- Eliminates the expense of developing and maintaining an in-house software program and comprehensive data file
- Reduces your non-deliverable as addressed mail
- Reduces fraud by ensuring accurate and deliverable addresses
- Minimizes "lost" mail - and customers - resulting from incorrect or improperly formatted addresses
- Provides an option to update your address files with corrected information from Finalist processing
- Reduces postage costs by adding ZIP + 4 codes and associated delivery point barcodes to address records to qualify for postage discounts
- Ensures address list accuracy
- Improved demographic information. Adding ZIP + 4 Codes allows you to identify a particular block, post office box, building, apartment, or business location.

# Finalist Files

There are multiple input and output files associated with Finalist. Some files are addressed directly by the Finalist coding engine and other files are addressed by driver programs.

- For Windows, Unix, and Linux systems, the files are accessed as direct file names including the path.
- For z/OS, the files are accessed as a DDNAME pointing to the DSN.

Precisely distributes all database files monthly with the exception of the Early Warning System (EWS) data. EWS data is available directly from the USPS and is usually updated on a bi-weekly (every other week) basis. Finalist distributes a current copy of the EWS file with the monthly database distribution (current as of the time the Finalist database is manufactured).

## Configuration File

The default name of the Finalist configuration file on the Windows, Unix, and Linux platforms is `pbfn.cfg`. On the z/OS platform, the DD name `PBFNCFG` represents the same information and can be used interchangeably in this section.

**Note:** All FILENAMEs in the `pbfn.cfg` file represent DD names on the z/OS platform.

### *Defining Data Files in the Configuration File*

The Finalist data file names are entered in to the Finalist configuration file (`pbfn.cfg`).

Input File	Configuration File Entry Field
Finalist CITY database	City Directory Filename
Finalist ZIP4 database	ZIP+4 Directory Filename 1
Exceptions Table	Exception Table Filename Controlled by the Number Exception Table Entries parameter - also defined in the Finalist configuration file.
Early Warning System (EWS) file	EWS Filename Controlled by the Early Warning System parameter - also defined in the Finalist configuration file.

Input File	Configuration File Entry Field
Enhanced Line of Travel (eLOT) database	<p data-bbox="678 325 829 350">LOT Filename</p> <p data-bbox="678 371 1393 422">Controlled by the Assign Lot parameter - also defined in the Finalist configuration file.</p>
Delivery Point Validation (DPV) database	<p data-bbox="678 493 821 518">DPV Filepath</p> <p data-bbox="678 539 1414 590">Controlled by the Delivery Point Validation parameter - also defined in the Finalist configuration file.</p> <p data-bbox="764 611 1414 695"><b>Note:</b> This is a file path - not a file name. The z/OS platform does not use this parameter. The z/OS file name is implied as DPVDB, DPVSDB, or DPVHDB.</p>
Residential Delivery Indicator (RDI) database	<p data-bbox="678 777 813 802">RDI Filepath</p> <p data-bbox="678 823 1425 873">Controlled by the Residential Delivery Indicator parameter - also defined in the Finalist configuration file.</p> <p data-bbox="764 894 1414 978"><b>Note:</b> This is a file path - not a file name. The z/OS platform does not use this parameter. The z/OS file name is implied as RDIDB.</p>
Locatable Address Conversion System (LACS or LACS <sup>Link</sup> ) database	<p data-bbox="678 1060 865 1085">LACS<sup>Link</sup> Filepath</p> <p data-bbox="678 1115 1377 1165">Controlled by the LACS<sup>Link</sup> parameter - also defined in the Finalist configuration file.</p> <p data-bbox="764 1186 1398 1278"><b>Note:</b> This is a file path - not a file name. The z/OS platform does not use this parameter. The z/OS name is implied as LLKDB.</p>
Suite <sup>Link</sup> (SLK) database	<p data-bbox="678 1360 857 1386">Suite<sup>Link</sup> Filepath</p> <p data-bbox="678 1415 1370 1478">Controlled by the Suite<sup>Link</sup> parameter - also defined in the Finalist configuration file.</p> <p data-bbox="764 1499 1398 1579"><b>Note:</b> This is a file path - not a file name. The z/OS platform does not use this parameter. The z/OS name is implied as SLKDB.</p>

## Defining Output Files in the Configuration File

The Finalist output file names are entered in to the Finalist configuration file (pbfncfg).

Output File	Configuration File Entry Field
Finalist Batch (Summary) Report	Batch Report Filename
USPS 3553 Report	3553 Report Filename This output file is only produced when running in CASS- certified mode.
Finalist Detail Report	Report Filename
Finalist Log (Error Log)	Log Filename

## Finalist Batch Driver Files

This section describes the Finalist batch driver:

- Windows — Finalist.exe
- z/OS — PGM=FINALIST
- Unix/Linux — ./finalist

The input JOB file controls the Finalist batch driver.

- Windows, Unix, and Linux — The JOB file is specified by the -j execution option (finalist.exe -j verimove.job).
- z/OS — The PBFNJOB file (DD) name points to the JOB file.

The Finalist batch driver produces an output job log file.

- Windows, Unix and Linux — The driver job log file can be overridden with the -l execution option (finalist.exe -l verimove.log). The default file is the JOB filename with a suffix of .log.
- z/OS — The JOBLOG output uses the JOBLOG file (DD). This JOBLOG file should not be confused with the Finalist LOG file.

For more information, please refer to [Running a Job](#) on page 133.



## Inside the JOB File

The Job File parameters are described in the following table.

Parameter	Description
fInput	Points to the input file: <ul style="list-style-type: none"> <li>• Windows, Unix, and Linux — This is the full file name (including path).</li> <li>• z/OS — This is a file (DD) pointing to the input file.</li> </ul>
fConfigFile	Overrides the location of the default Finalist configuration (pbfncfg or PBFNCFG) file (see above).
fDefinition	Points to the Finalist definition (DEF) file. The specified value can be anything but is typically the jobname followed by .def (verimove.def) or PBFNDEF file (DD) name.
fOutAllFile	Points to an output file that contains all records (good and bad): <ul style="list-style-type: none"> <li>• Windows, Unix, and Linux — This is the full file name (including path).</li> <li>• z/OS — This is a file (DD) pointing to the output file.</li> </ul>
fOutErrorsFile	Points to an output file that only contains records that did not address validate: <ul style="list-style-type: none"> <li>• Windows, Unix, and Linux — This is the full file name (including path).</li> <li>• z/OS — This is a file (DD) pointing to the output file.</li> </ul>
fOutValidFile	Points to an output file that only contains records that address validated: <ul style="list-style-type: none"> <li>• Windows, Unix, and Linux — This is the full file name (including path).</li> <li>• z/OS — This is a file (DD) pointing to the file.</li> </ul>
fReport	Overrides the location specified in the configuration file for the driver program to use to write the Finalist Batch Report (Batch Report Filename=).
fDetailReport	Overrides the location in the configuration file for the driver program to use to write the Address Detail Report (Report Filename=).
fpbfncfgLog	Overrides the location in the configuration file for the driver program to use to write the Finalist engine log messages (Log Filename=).

## User Drivers

A user driver can use any files. A user driver can override the default Finalist configuration files or allow the files to process as documented above. For more information, please refer to Chapter 4, Programming Tips and Techniques in your Finalist Reference Guide.

## Input

Finalist processing input consists of the following items:

- Your Name-and-Address Input File
- Finalist ZIP + 4 Data File
- Finalist City/State File
- Additional databases (for example, Delivery Point Validation (DPV), LACS<sup>Link</sup>, etc.)
- Job control and formatting information

## Output

Finalist processing generates the following output:

<b>Corrected and standardized addresses</b>	You determine whether the processing results are written to your input file or to additional output files.
<b>Return codes</b>	You can use the Finalist return codes to analyze addresses that Finalist could not match to USPS data to determine any corrective action.
<b>Reports</b>	<p>These Finalist reports provide information for the processing performed on your input file during the processing run:</p> <ul style="list-style-type: none"> <li>• Address Detail Report — The Address Detail Report consists of four sections providing processing information for each input address.</li> <li>• Finalist Batch Report — The Finalist Batch Report displays all configurable options and statistics generated at the time of the processing run.</li> <li>• USPS Form 3553 (CASS Summary Report) — With each mailing submitted at an automation-based rate, a mailer must submit a completed USPS Form 3553 (CASS Summary Report). You can submit the original USPS Form or</li> </ul>

the facsimile generated by your program. Finalist will produce a facsimile of USPS Form 3553 (CASS Summary Report) and partially complete it for you.

**Error messages** Finalist error messages assist in troubleshooting any errors you may encounter during processing.

## Finalist Auxiliary Processing

Finalist includes the following options for performing additional processing on your input file.

Option	Description
Delivery Point Validation (DPV)	DPV processing uses DPV data from the USPS to ensure your input file addresses are actual physical addresses to which the USPS delivers mail.
LACSLink	LACSLink processing provides you access to address conversion data resulting from 911 emergency response implementation.
Enhanced Line of Travel (eLOT)	Enhanced Line of Travel (eLOT) processing ensures that Enhanced Carrier Route mailings are sorted more closely to the actual delivery sequence.
Early Warning System (EWS)	Early Warning System (EWS) processing provides access to new address information that is in use, but not yet available on the ZIP + 4 File.
SuiteLink	SuiteLink processing improves your business address information by adding secondary (suite) information to business addresses that CASS processing identified as highrise default records.
Residential Delivery Indicator (RDI)	Residential Delivery Indicator (RDI) processing determines whether an address is a residential or business address.
Exceptions Table	Exceptions Table processing allows you to alter input address fields to enhance coding of your address file.
AddrScan	AddrScan processing can: <ul style="list-style-type: none"> <li>• Identify individual components in multi-line addresses</li> <li>• Format and standardize the address information</li> <li>• Return processed address lines in a specified order</li> <li>• Identify firm, street, and city lines</li> <li>• Preprocess multi-line and/or "floating" address lines</li> </ul>

Option	Description
PreciselyID	<p>For each addressable location, PreciselyID processing provides a unique and persistent identifier to reference the address without storing the whole address string. The PreciselyID is further verification an address exists, regardless of DPV matching.</p> <p>Additionally, to link it to other data (i.e., geoenrichment), Precisely has done the complex matching for you. Use the PreciselyID as the gateway to the rest of the Precisely data portfolio. Contact your Sales Representative for more information, based on your specific needs.</p>

For detailed information on Finalist auxiliary processing, please refer to your Finalist Installation Guide.

# 2 - Configuring Finalist

## In this section

---

Using the pbfm.cfg File to Configure Finalist.....	14
Using the Workbench or Lookup Tool to Configure Finalist.....	35



# Using the pbfncfg File to Configure Finalist

This section provides guidelines for using the pbfncfg file to configure your Finalist installation. On the z/OS platform, the DD name PBFNCFG represents the same information and can be used interchangeably in this section.

**Note:** All FILENAMEs in the pbfncfg file represent DD names on the z/OS platform.

## What is the pbfncfg Configuration File?

The settings in the Finalist pbfncfg configuration file determine how Finalist processes your files. During installation, the pbfncfg file is placed in the bin directory with default settings. You can process with Finalist using those default settings or you can change the settings in the pbfncfg file for your specific installation site.

For example, the default file name for the Finalist Batch Report file is batch.txt. You can edit the Batch Report Filename field in pbfncfg to use a different name for the Finalist Batch Report file.

## Altering the pbfncfg Configuration File

You can alter the pbfncfg configuration file using your application or any text editor. After you modify the pbfncfg file, the PBFNInit function loads the desired parameters. If you do not wish to use the pbfncfg configuration file to save setup parameters for your user-written application, you can store these values elsewhere. In that case, instead of using the PBFNInit call, you would allocate your own PBFNSetupDef structure, fill in the appropriate information, and pass that PBFNSetupDef structure directly to the PBFNInit call. The following is a sample pbfncfg configuration file for the Windows and Unix platforms.

```
File Section:
-----
City Directory Filename =
ZIP+4 Directory Filename 1 =
CASS Section:
-----
Batch Report Filename = batch.txt
3553 Report Filename =
CASS Company Name =
CASS Product Name =
CASS Product Version =
Z4 Change Certified Company Name =
```

```

Z4 Change Product Name =
Z4 Change Product Version =
LOT Certified Company Name =
LOT Certified Product Name =
LOT Certified Product Version =
DPC Certified Company Name =
DPC Certified Product Name =
DPC Certified Product Version =
CASS Mailer Name =
CASS Mailer Address =
CASS Mailer Address2 =
CASS Mailer Address3 =
CASS Mailer Address4 =
CASS Mailer City Line =
CASS Configuration = AAA
Batch Report = ON
3553 Report = ON
Process Section:
-----
CASS Flag = ON
Assign Carrier Routes = ON
Assign Unassign Records =
Assign Abbreviated City = OFF
CASS Standardize Case = U
Cache Size = 12
Beginning Frame Char = !
End Frame Char = !
Return DPBC = ON
Dual Address Switch = P
Return Alias Street Name =
Assign LOT = OFF
Padded String Data =
Process LOT Only =
Uppercase Input =
Remove Noise Characters =
Process Firms = ON
Return Input Firm =
All Street Matching =
R777 Deliverable =
Convert Secondary to PMB =
Number Exception Table Entries =
Suggestion Count = 10
Product Section:
-----
Exception Table Filename =
EWS Filename =
LOT Filename =
DPV Filepath =
RDI Filepath =
LACSLink Filepath =
SuiteLink Filepath =
PreciselyID Processing =
Early Warning System =

```

```

Delivery Point Validation =
DPV Shutdown Indicator =
Delivery Point Validation Tie Break =
DPV No-Stat Table =
DPV Vacant Table =
DPV PBSA Table =
DPV DNA Table =
DPV Throwback Table =
DPV NSL Table =
DPV Buffer Size =
Residential Delivery Indicator =
Commercial Mail Validation =
LACSLink =
LACSLink Processing =
SuiteLink =
SuiteLink Small Memory Flag =
SuiteLink Shutdown Indicator =
Return SLK Input Secondary =
SOFTWARE KEY =
DPVKey =
LACSLink Key =
Report Section:
-----
Report Filename = report.txt
Report Title = Finalist Report File From PBFN.CFG
Address Detail Report PAGE LEN = 0
Address Detail Report MAX REC = 0
Address Detail Report NTH REC = 0
Address Detail Report Type =
Address Detail Report Isol =
Address Detail Report Sugg =
Address Detail Report Info =
Log Msg Section:
-----
Log Filename = log.txt
Log Level = 3

```

The following table provides a description for each field in the pbfncfg configuration file.

**Table 1: Finalist pbfncfg Configuration File**

Configuration Setting	Description
File Section	
City Directory Filename	Identifies the City Database file name.
ZIP + 4 Directory Filename 1	Identifies the ZIP + 4 Database file name.



Configuration Setting	Description
CASS Section	
Batch Report Filename	The Output file name for the Finalist Batch Report. The default file name is batch.txt.
3553 Report Filename	The Output file name for the USPS Form 3553 (CASS Summary Report). The default file name is cass3553.txt.
CASS Company Name	The name of the CASS-certified company displays in section A box 1 on the USPS Form 3553 (CASS Summary Report).
CASS Product Name	The name of the CASS-certified product displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
CASS Product Version	The version number for the CASS-certified product displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
Z4 Change Certified Company Name	The Z4Change-certified company name displays in section A box 4 on the USPS Form 3553 (CASS Summary Report).
Z4 Change Product Name	The Z4Change product name displays in section A box 5 on the USPS Form 3553 (CASS Summary Report).
Z4 Change Product Version	The Z4Change product version number displays in section A box 5 on the USPS Form 3553 (CASS Summary Report).
LOT Certified Company Name	The Line of Travel (LOT)-certified company name displays in section A box 10 on the USPS Form 3553 (CASS Summary Report).
LOT Certified Product Name	The Line of Travel (LOT)-certified product name displays in section A box 11 on the USPS Form 3553 (CASS Summary Report).
LOT Certified Product Version	The Line of Travel (LOT)-certified product version displays in section A box 11 on the USPS Form 3553 (CASS Summary Report).
DPC Certified Company Name	The DirectDPV company name displays in section A box 7 on the USPS Form 3553 (CASS Summary Report).
DPC Certified Product Name	The DirectDPV product name displays in section A box 8 on the USPS Form 3553 (CASS Summary Report).
DPC Certified Product Version	The DirectDPV product version displays in section A box 8 on the USPS Form 3553 (CASS Summary Report).

Configuration Setting	Description
CASS Mailer Name	The Mailer's name displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
CASS Mailer Address	Mailer's address. This information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
CASS Mailer Address 2-4	Additional address lines for Mailer's address display in section D box 3 on the USPS Form 3553 (CASS Summary Report).
CASS Mailer City Line	The Mailer's city, state, and ZIP Code information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
CASS Configuration	The CASS configuration value displays in section A box 3 on the USPS Form 3553 (CASS Summary Report).
Batch Report	Indicate whether to print the Finalist Batch Report: <b>OFF</b> Do not print the Finalist Batch Report <b>ON</b> Print the Finalist Batch Report <b>Blank</b> Defaults to OFF.
3553 Report	Indicate whether to print the USPS Form 3553 (CASS Summary Report). <b>OFF</b> Do not print the USPS Form 3553 (CASS Summary Report). <b>ON</b> Print the USPS Form 3553 (CASS Summary Report). <b>Blank</b> Defaults to OFF.
Process Section	
CASS Flag	Indicate whether to process in a non-CASS or CASS mode. <b>OFF</b> Process in non-CASS mode. Do not check databases and engines for expiration dates. <b>ON</b> Process in CASS mode. Check databases and engines for expiration dates. <b>Blank</b> Defaults to ON. <b>Note:</b> If CASS Flag = ON and a conflicting option is encountered (Configuration, Assign CR, Return DPBC, LACS <sup>Link</sup> =OFF, Suite <sup>Link</sup> =OFF, or DPV=OFF), a warning message is written to the log file indicating the reason for CASS being forced off. The message is similar to: Warning Message; CASS forced off: CASS Configuration, Return DPBC, Assign CR, Assign SuiteLink, Assign LACSLink, Assign DPV

Configuration Setting	Description
Assign Carrier Routes	<p>Indicate whether to assign carrier route codes to your file.</p> <p><b>OFF</b> Do not assign carrier route codes.</p> <p><b>ON</b> Assign carrier route codes.</p> <p><b>Blank</b> Defaults to ON.</p>
Assign Unassign Records	<p>Indicate how to code previously unassigned or expired addresses. For more information on the assign/unassign feature, refer to the description of the cProcessUnassigned field of the PBFNSetupDef structure.</p> <p><b>OFF</b> Process all supplied addresses.</p> <p><b>ON</b> Process only previously unassigned or expired addresses. If an encrypted address assignment date (cProcessDate) in the PBFNAddressDataDef structure is passed to Finalist and that assignment date meets USPS regulations for valid address records, Finalist does not process that record but includes that record as valid for CASS record counts. If a record has not been previously coded and a date assigned, Finalist will also process that record. Records bypassed and counted return with a 5200 error code. This date must be stored and passed on subsequent processing runs.</p> <p><b>XXX</b> Number of elapsed days since address was assigned. This number can be any value from 1 to the USPS determined number of days an address is valid. That value is currently set to 180. If a number greater than 180 is entered, the USPS expiration days are used.</p> <p><b>Blank</b> Defaults to OFF.</p>
Assign Abbreviated City	<p>Indicate whether to return abbreviated city names in the label lines.</p> <p><b>OFF</b> Do not return abbreviated city names.</p> <p><b>ON</b> Return abbreviated city names.</p> <p><b>Blank</b> Defaults to OFF.</p>
CASS Standardize Case	<p>Indicate whether to return addresses in all upper, all lower, or mixed upper and lower case. This field affects the address coding output only.</p> <p><b>U</b> Return addresses in all upper case.</p> <p><b>L</b> Return addresses in all lower case.</p> <p><b>M</b> Return addresses in upper and lower case.</p> <p><b>Blank</b> Defaults to U.</p>

Configuration Setting	Description
Cache Size	<p>Indicate whether Finalist should use internal caching.</p> <p><b>OFF</b> Turn off internal caching.</p> <p><b>XX</b> Replace "XX" with a value between 12-99 for the number of buffers to use for internal caching.</p> <p><b>Blank</b> The default value for z/OS is 30. The default value for all other operating systems is 12.</p>
Beginning Frame Char	<p>Specify the character to use for the front framing character for advanced barcodes. If you do not specify a character for BegFrame or EndFrame, Finalist uses the exclamation point (!) as the default value.</p>
End Frame Char	<p>Specify the character to use for the end framing character for advanced barcodes. If you do not specify a value here, the BegFrame character is used as the default value. If you do not specify a character for BegFrame or EndFrame, Finalist uses the exclamation point (!) as the default value.</p>
Return DPBC	<p>Indicate whether to return delivery point barcodes.</p> <p><b>OFF</b> Do not return delivery point barcodes.</p> <p><b>ON</b> Return delivery point barcodes.</p> <p><b>Blank</b> Defaults to ON.</p>

Configuration Setting	Description
Dual Address Switch	<p>Specify the order for processing and matching addresses when the input file contains dual addresses (one a conventional address, a second containing a PO Box address). If the selected address is valid, processing stops. If the selected address does not validate, Finalist attempts to code the secondary address.</p>
<b>ACZ</b>	<p>Above city and ZIP Code line. The address line closest to the last line in the address is given the highest priority in the match process. Any address line above the last line is not used for matching.</p>
<b>L12</b>	<p>Line 1 in the dual address is given the highest priority in the match process.</p>
<b>L21</b>	<p>Line 2 in the dual address is given the highest priority in the match process.</p>
<b>A</b>	<p>The conventional address is given the highest priority in the match process.</p>
<b>P</b>	<p>The PO Box address is given the highest priority in the match process.</p>
<b>1ST</b>	<p>The first valid address (in the order Address line 1 and then Address line 2) is given the highest priority in the match process.</p>
<b>Blank</b>	<p>Defaults to ACZ.</p> <p>When dual addresses are contained on a single line and cCASSFlag is set to ON, the USPS address type priority is used in the following order:</p> <ol style="list-style-type: none"> <li>1. PO Box</li> <li>2. Firm</li> <li>3. Highrise</li> <li>4. Street</li> <li>5. Rural Route</li> <li>6. General Delivery</li> </ol>

Configuration Setting	Description
Return Alias Street Name	<p>Indicates whether to return alias street names in label lines. An alias street name is an alternate name for a street, maintained at the range Plus4 ZIP Code level. An alias street name can be a nickname alias street name of a preferred alias street name.</p> <ul style="list-style-type: none"> <li>• <b>ON</b> - If the input address matches to an alias, return the alias. If the input address matches to a base address, but a Preferred alias exists, return the Preferred alias. If the input address matches to a base address and no Preferred alias exists, return the base address.</li> <li>• <b>PXO</b> - If a Preferred alias exists, return the Preferred alias. If no Preferred alias exists, but an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated or Preferred alias exists, but some other alias type was input, return the input alias. If none of these scenarios apply, return the base street name.</li> </ul> <p><b>OFF   P</b> If a Preferred alias exists, return the Preferred alias. Otherwise, return the base street.</p> <p><b>PX</b> If a Preferred alias exists, return the Preferred alias. If no Preferred alias exists, but an Abbreviated alias exists, return the Abbreviated alias. If neither exists, return the base street name.</p> <p><b>XPO</b> If an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated alias exists, but a Preferred alias exists, return the Preferred alias. If no Abbreviated or Preferred alias exists, but some other alias does exist, return the alias. If no alias of any kind exists, return the base street name.</p> <p><b>XP   X</b> If an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated alias exists, but a Preferred alias exists, return the Preferred alias. If neither exists, return only the base street name.</p> <p><b>Blank</b> Defaults to OFF.</p>
Assign LOT	<p>Indicate whether to assign Line of Travel (LOT) codes.</p> <p><b>OFF</b> Do not assign Line of Travel (LOT) codes.</p> <p><b>ON</b> Assign Line of Travel (LOT) codes.</p> <p><b>Blank</b> Defaults to OFF.</p>
Padded String Data	<p>If you use a platform or coding language (for example, COBOL) that does not support null terminated strings, pad all fields with blanks. Indicate whether you use input/output fields that are padded with blanks or null terminated:</p> <p><b>OFF</b> Use null terminated strings.</p> <p><b>ON</b> Use blank filled fields.</p> <p><b>Blank</b> Defaults to OFF.</p>

Configuration Setting	Description
Process LOT Only	<p>Indicate whether to perform only eLOT processing.</p> <p><b>OFF</b> Perform normal Finalist processing.</p> <p><b>ON</b> Perform eLOT processing only. If you specify "ON" for this field, Finalist does not perform address cleansing. A fully-coded address record will be passed to the engine. Finalist performs an eLOT lookup and returns the eLOT code.</p> <p><b>Blank</b> Defaults to OFF.</p>
Uppercase Input	<p>Indicate if your input data is in an all uppercase format.</p> <p><b>OFF</b> Input data may not be in an all uppercase format. Convert input data to an all uppercase format.</p> <p><b>ON</b> All input data is in an all uppercase format. To improve performance, do not perform the all uppercase format conversion.</p> <p><b>Blank</b> Defaults to OFF.</p> <p><b>Note:</b> Mainframe data tends to be all uppercase.</p>
Remove Noise Characters	<p>Indicate whether to remove noise characters (unnecessary punctuation and blanks).</p> <p><b>OFF</b> Input data does not contain unnecessary punctuation and/or blanks. Bypass the Finalist routine that removes noise characters. To improve performance, bypass the routine that checks each line for noise characters.</p> <p><b>ON</b> Input data may contain unnecessary punctuation and blanks. Remove noise characters.</p> <p><b>Blank</b> Defaults to ON.</p>
Process Firms	<p>Indicate whether to perform firm processing.</p> <p><b>OFF</b> Do not process firms. If you do not store firm names, or are not interested in matching to a firm level (for non-CASS mode processing only), specify "OFF" for this field to bypass firm processing and improve performance. Finalist will not load or use the firm portion of the database to improve performance.</p> <p><b>ON</b> Finalist performs normal CASS processing for firms.</p> <p><b>Blank</b> Defaults to ON.</p>
Return Input Firm	<p>Indicate whether to return the input firm.</p> <p><b>OFF</b> Do not return the input firm.</p> <p><b>ON</b> Return the input firm.</p> <p><b>Blank</b> Defaults to OFF.</p>

Configuration Setting	Description
All Street Matching	<p>Indicate whether to perform All Street Matching (ASM) processing to apply additional matching logic to correct errors in street names and obtain a match. For example, when the first letter of a street is misspelled or missing on input, this feature searches all street names in a locality to find an input address. ASM provides the best address validation but may reduce performance. ASM processing is available for U.S. addresses only.</p> <p><b>OFF</b> Do not perform ASM processing.</p> <p><b>ON</b> Perform ASM processing.</p> <p><b>Blank</b> Defaults to OFF.</p>
R777 Deliverable	<p>Addresses with Carrier Route R777 are phantom routes that are not eligible for street delivery. Since these addresses are assigned a ZIP + 4 code by the USPS, Finalist marks these addresses as deliverable. If you do not want the addresses with Carrier Route R777 marked as deliverable, set this option to OFF and the following actions are performed for the address:</p> <ul style="list-style-type: none"> <li>• No ZIP + 4 code is assigned</li> <li>• Address is not counted on the USPS Form 3553 (CASS Summary Report)</li> <li>• DPV Footnote of R7 is returned</li> </ul> <p>Valid values are:</p> <p><b>OFF</b> R777 addresses are not deliverable.</p> <p><b>ON</b> R777 addresses are deliverable.</p> <p><b>Blank</b> Defaults to ON.</p>
Convert Secondary to PMB	<p>Indicate whether to convert secondary information to “PMB” under the following conditions:</p> <ul style="list-style-type: none"> <li>• A secondary number is present in the returned ZIP + 4 address.</li> <li>• The secondary number does not DPV confirm.</li> <li>• The primary number (and/or other secondary number) confirms as a CMRA.</li> <li>• The unconfirmed unit designator is not a pound sign (#).</li> </ul> <p>This processing only applies if the primary address codes to a CMRA.</p> <p>Valid values are:</p> <p><b>OFF</b> Do not perform Secondary to PMB conversion processing.</p> <p><b>ON</b> Perform Secondary to PMB conversion processing.</p> <p><b>Blank</b> Defaults to OFF.</p>
Number Exception Table Entries	The maximum number of exception table entries. Defaults to 1000.
Suggestion Count	The maximum number of returned suggestions per address based on the specified suggestion criteria.



Configuration Setting	Description
Product Section	
Exception Table Filename	The Exceptions Table file name and path.
EWS Filename	The Early Warning System (EWS) file name and path.
LOT Filename	The Line of Travel (LOT) file name and path.
DPV Filepath	The Delivery Point Validation (DPV) file path.
RDI Filepath	The Residential Delivery Indicator (RDI) File path.
LACSLink Filepath	The LACS <sup>Link</sup> File path.
SuiteLink Filepath	The Suite <sup>Link</sup> File path.
PreciselyID Processing	Indicate whether to perform PreciselyID processing. <b>OFF</b> Do not perform PreciselyID processing. <b>ON</b> Perform PreciselyID processing. <b>Blank</b> Defaults to OFF.
Early Warning System	Indicate whether to perform Early Warning System (EWS) processing. <b>OFF</b> Do not perform Early Warning System (EWS) processing. <b>ON</b> Perform Early Warning System (EWS) processing <b>Blank</b> Defaults to OFF.

Configuration Setting	Description
Delivery Point Validation	<p>Indicate whether to perform Delivery Point Validation (DPV) processing.</p> <p><b>Note:</b> The USPS CASS regulations require DPV processing to generate the USPS Form 3553 (USPS CASS Summary Report).</p> <p><b>OFF</b> Do not perform DPV processing.</p> <p><b>ON</b> Perform DPV processing using the DPV Full database (dpvh.db).</p> <p><b>MEM</b> Perform DPV processing with the DPV Full database (dpvh.db) fully in memory for improved performance. NOTE: MEM is converted to ON with DPV Buffer Size=H.</p> <p><b>SPL</b> Perform DPV processing using the DPV Split database (dpvs.db). Split File processing separates the DPV Data File into 100 smaller files based on the first two digits of the ZIP Code. The Split File segment associated with the first two digits of the ZIP Code is loaded into memory. If you sort your mailing file by ZIP Code, you can bring the relevant portion of the DPV file into memory. This process reduces the number of I/O requests that normally occurs when you use the full DPV Data File. Use this option if your file is sorted by ZIP Code. This is the optimal solution for Finalist customers having at least 100MB, but no more than 650MB, of RAM/Memory.</p> <p><b>FLT</b> Perform DPV processing using the DPV Flat database (dpv.db). Use the DPV Flat File for improved batch performance. This file is in excess of 2.3GB and not dependent on RAM/Memory capacity. Input files should be sorted by ZIP Code. This is an optimal solution if you have less than 100MB of RAM/Memory available.</p> <p><b>Blank</b> Defaults to OFF.</p>

Configuration Setting	Description
DPV Shutdown Indicator	<p data-bbox="573 327 1422 380">Indicate the action to take when encountering a DPV Seed during the processing run.</p> <p data-bbox="573 396 1349 449"><b>W</b> Issue warning message when a DPV Seed has been encountered. Processing continues but DPV processing is disabled.</p> <p data-bbox="573 470 1422 522"><b>S</b> Stop processing when encountering a DPV Seed. Finalist issues an error message, stops processing succeeding addresses, and exits the program.</p> <p data-bbox="573 543 792 569"><b>Blank</b> Defaults to W.</p> <p data-bbox="729 585 1333 638"><b>Note:</b> If a DPV seed is encountered and W is set, CASS processing is turned off.</p> <p data-bbox="644 674 1422 726">CASS processing is turned off for any jobs submitted after the DPV False Positive (Seed) violation was encountered.</p> <p data-bbox="644 747 1422 894">For the job that encountered the DPV False Positive (Seed) violation, the CASS statement is provided showing the number of records that were DPV confirmed up until the point of the DPV False Positive (Seed) violation. No records will be DPV confirmed after the DPV False Positive (Seed) violation occurred.</p> <p data-bbox="644 915 1422 1083">Any jobs submitted before the DPV False Positive (Seed) violation was encountered will run through to completion, including DPV processing for each file. This occurs even if the jobs do not complete processing until after the time the DPV False Positive (Seed) violation was found in the job that first encountered the DPV False Positive (Seed) violation. Finalist generates the USPS Form 3553 (CASS Summary Report) for these jobs.</p> <p data-bbox="644 1104 1422 1188">For any job set to perform CASS processing and submitted after a DPV False Positive (Seed) violation occurs, Finalist generates an initialization error and stops processing.</p> <p data-bbox="729 1209 1406 1262"><b>Note:</b> The Shutdown Indicator applies to batch processing only. If your DPV key is NCOA or “No-Stop”, this option is ignored.</p>

Configuration Setting	Description
Delivery Point Validation Tie Break	<p>The USPS allows DPV processing to be used as a tie breaker for matching inexact street records. If only one of the records in a tie is delivery point validated, a match is allowed to the inexact record. When processing results in an inexact match due to the input address directional, DPV processing can be used as a tie breaker if only one of the records is found to be delivery point validated and the delivery point validated record does not violate the cardinal direction rule.</p> <p><b>Note:</b> The USPS CASS regulations require DPV Tie Break processing to generate the USPS Form 3553 (USPS CASS Summary Report).</p> <p>Indicate whether to perform DPV Tie Breaker processing.</p> <p><b>OFF</b> Do not perform DPV Tie Breaker processing.</p> <p><b>ON</b> Perform DPV Tie Breaker processing. The DPV Tie Breaker Option can increase your matching percentages but can also negatively impact performance. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>Blank</b> Defaults to ON.</p> <p><b>Note:</b> The default value "ON" only applies when "Delivery Point Validation" is defined as ON, MEM, SPL, or FLT.</p>
DPV No-Stat Table	<p>DPV processing uses the No-Stat Table to identify deliveries that are not valid for Computerized Delivery Sequence (CDS) pre-processing. Indicate whether to use the No-Stat Table and return the proper No-Stat code to the output.</p> <p><b>OFF</b> Do not perform No-Stat Table processing.</p> <p><b>ON</b> Perform No-Stat Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform No-Stat Table processing. Load the No-Stat Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform No-Stat Table processing. Use the No-Stat Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>

Configuration Setting	Description
DPV Vacant Table	<p>DPV processing uses the Vacant Table to identify delivery addresses that have been active in the past but, according to USPS data, have not been occupied within the last 90 days. Indicate whether to use the Vacant Table and return the proper Vacant code to the output.</p> <p><b>OFF</b> Do not perform Vacant Table processing.</p> <p><b>ON</b> Perform Vacant Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform Vacant Table processing. Load the Vacant Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform Vacant Table processing. Use the Vacant Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>
DPV PBSA Table	<p>DPV processing uses the PBSA Table to identify PO Box Street Addresses (PBSA). A PBSA address is a street address that represents a USPS PO Box. Indicate whether to use the PBSA Table and return the PBSA result to the output.</p> <p><b>OFF</b> Do not perform PBSA Table processing.</p> <p><b>ON</b> Perform PBSA Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform PBSA Table processing. Load the PBSA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform PBSA Table processing. Use the PBSA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>
DPV DNA Table	<p>DPV processing uses the Door Not Accessible (DNA) Table to identify delivery addresses where carriers cannot knock on the door for mail delivery or where carriers cannot physically access a residence/building such as rural/highway contact route (HCR), long driveway, or gated community. Indicate whether to use the DNA Table and return the proper DNA code to the output.</p> <p><b>OFF</b> Do not perform DNA Table processing.</p> <p><b>ON</b> Perform DNA Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform DNA Table processing. Load the DNA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform DNA Table processing. Use the DNA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>

Configuration Setting	Description
DPV Throwback Table	<p>DPV processing uses the P.O. Box Throwback Table to identify a delivery point that is a street address where mail is not delivered. Instead, delivery is made to the customer's P.O. Box address. Indicate whether to use the P.O. Box Throwback Table and return the proper Throwback code to the output.</p> <p><b>OFF</b> Do not perform P.O. Box Throwback Table processing.</p> <p><b>ON</b> Perform P.O. Box Throwback Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform P.O. Box Throwback Table processing. Load the DNA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform P.O. Box Throwback Table processing. Use the DNA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>
DPV NSL Table	<p>DPV processing uses the No Secure Location (NSL) table to identify delivery locations that are not secure. For example, a carrier can access a door but cannot leave a package due to security concerns. The NSL designation alerts mailers to locations where businesses are closed on certain days and locations without mail receptacles (for example, a storefront). Indicate whether to use the NSL Table and return the proper NSL code to the output.</p> <p><b>OFF</b> Do not perform NSL Table processing.</p> <p><b>ON</b> Perform NSL Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.</p> <p><b>IN</b> Perform NSL Table processing. Load the NSL Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>OUT</b> Perform NSL Table processing. Use the NSL Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).</p> <p><b>Blank</b> Defaults to OFF.</p>
DPV Buffer Size	<p>Indicates the memory model for DPV processing.</p> <p><b>P</b> Pico. Stores no data in memory. No tables or indexes are loaded.</p> <p><b>U</b> Ultra-small. Stores no data in memory. Partial indexes are loaded.</p> <p><b>S</b> Small</p> <p><b>M</b> Medium</p> <p><b>L</b> Large</p> <p><b>H</b> Huge. Stores all data in memory.</p> <p><b>Blank</b> Defaults to M.</p>

Configuration Setting	Description
Residential Delivery Indicator	Indicate whether to perform Residential Delivery Indicator (RDI) processing.
	<b>OFF</b> Do not perform Residential Delivery Indicator (RDI) processing.
	<b>ON</b> Perform Residential Delivery Indicator (RDI) processing.
	<b>Blank</b> Defaults to OFF.
Commercial Mail Validation	Indicate whether to perform CMRA processing. Private companies offering mailbox rental services to individuals and businesses are Commercial Mail Receiving Agents (CMRA).
	<b>OFF</b> Do not perform CMRA processing.
	<b>ON</b> Perform CMRA processing.
	<b>IN</b> Perform CMRA Table processing. Load the CMRA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
	<b>OUT</b> Perform CMRA Table processing. Use the CMRA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
<b>Blank</b> Defaults to OFF.	
LACSLink	Indicate whether to perform LACS <sup>Link</sup> processing. USPS CASS regulations require LACS <sup>Link</sup> processing. If you do not perform LACS <sup>Link</sup> processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).
	<b>OFF</b> Do not perform LACS <sup>Link</sup> processing.
	<b>ON</b> Perform LACS <sup>Link</sup> processing.
	<b>Blank</b> Defaults to OFF.
LACSLink Processing	Indicates the memory model for LACS <sup>Link</sup> processing.
	<b>P</b> Pico. Stores no data in memory. No tables or indexes are loaded.
	<b>U</b> Ultra-small. Stores no data in memory. Partial indexes are loaded.
	<b>S</b> Small
	<b>M</b> Medium
	<b>L</b> Large
	<b>H</b> Huge. Stores all data in memory.
	<b>Blank</b> Defaults to S.

Configuration Setting	Description
SuiteLink	<p>Indicate whether to perform Suite<sup>Link</sup> processing. USPS CASS regulations require Suite<sup>Link</sup> processing. If you do not perform Suite<sup>Link</sup> processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).</p> <p><b>OFF</b> Do not perform Suite<sup>Link</sup> processing.</p> <p><b>ON</b> Perform Suite<sup>Link</sup> processing.</p> <p><b>Blank</b> Defaults to OFF.</p>
SuiteLink Small Memory Flag	<p>Indicates the memory model for Suite<sup>Link</sup> processing.</p> <p><b>P</b> Pico. Stores no data in memory. No tables or indexes are loaded.</p> <p><b>U</b> Ultra-small. Stores no data in memory. Partial indexes are loaded.</p> <p><b>S</b> Small</p> <p><b>M</b> Medium</p> <p><b>L</b> Large</p> <p><b>H</b> Huge. Stores all data in memory.</p> <p><b>Blank</b> Defaults to L.</p>
SuiteLink Shutdown Indicator	<p>Indicate the action to take when encountering a Suite<sup>Link</sup> processing error during the processing run.</p> <p><b>W</b> Issue warning and turn off Suite<sup>Link</sup> processing.</p> <p><b>S</b> Stop Finalist processing when encountering a Suite<sup>Link</sup> error.</p> <p><b>I</b> Ignore errors and continue Suite<sup>Link</sup> processing.</p> <p><b>Blank</b> Defaults to W.</p> <p><b>Note:</b> USPS regulations require Suite<sup>Link</sup> processing for CASS certification and to generate a USPS Form 3553 (CASS Summary Report). Per the USPS regulations, any job that does not include Suite<sup>Link</sup> processing, must not generate the USPS Form 3553 (CASS Summary Report). If you are processing in a CASS-certified mode (CASS=ON), Finalist sets the Suite<sup>Link</sup> Shutdown Indicator to "S" in order to stop processing if the Suite<sup>Link</sup> initialization fails and to prevent generation of an invalid USPS Form 3553 (CASS Summary Report).</p>



Configuration Setting	Description
Return SLK Input Secondary	<p>Indicates how to return secondary information when Suite<sup>Link</sup> secondary information is available.</p> <p><b>B</b> Both. Return both Suite<sup>Link</sup> and input secondary information. (This is the Default).</p> <p><b>S</b> Suite<sup>Link</sup>. Return Suite<sup>Link</sup> secondary only. Do not return input secondary.</p> <p><b>I</b> Input. Return input secondary only. Do not return Suite<sup>Link</sup> secondary.</p> <p><b>N</b> None. Do not return Suite<sup>Link</sup> secondary or input secondary.</p> <ul style="list-style-type: none"> <li>Blank — Defaults to B.</li> </ul> <p><b>Note:</b> If Suite<sup>Link</sup> processing does not result in a match, Finalist ignores this option and returns the normal address output. Regardless of the option specified, the output ZIP + 4 is based on the match made using the Suite<sup>Link</sup> secondary. Input secondary information includes the “#” designator for purposes of this option.</p>
Software Key	Enter your Finalist activation key.
DPVKey	Enter the Delivery Point Validation (DPV) activation key.
LACSLink Key	Enter the LACS <sup>Link</sup> activation key.
Report Section	
Report Filename	Specify the Address Detail Report output file name. Defaults to rpt.txt.
Report Title	Specify the title of the Address Detail Report.
Address Detail Report PAGE LEN	Specify the maximum page length for the Address Detail Isolation Report.
Address Detail Report MAX REC	Specify the maximum number of records to output to the Address Detail Isolation Report.
Address Detail Report NTH REC	Specify the records to print on the Address Detail Isolation Report (for example, print every 8th record).

Configuration Setting	Description
Address Detail Report Type	Indicate whether to print the Address Detail Report.
	<b>OFF</b> Do not print the Address Detail Report.
	<b>ON</b> Print the Address Detail Report.
	<b>ERR</b> Print only failed addresses on the Address Detail Report.
	<b>Blank</b> Defaults to OFF.
Address Detail Report Isol	Indicate whether to print the Address Detail Isolation Report.
	<b>OFF</b> Do not print the Address Detail Isolation Report.
	<b>ON</b> Print the Address Detail Isolation Report.
	<b>Blank</b> Defaults to OFF.
Address Detail Report Sugg	Indicate whether to print the Address Detail Suggestion Report.
	<b>OFF</b> Do not print the Address Detail Suggestion Report.
	<b>ON</b> Print the Address Detail Suggestion Report.
	<b>Blank</b> Defaults to OFF.
Address Detail Report Info	Indicate whether to print the Address Detail Information Report.
	<b>OFF</b> Do not print the Address Detail Information Report.
	<b>ON</b> Print the Address Detail Information Report.
	<b>Blank</b> Defaults to OFF.
Log Message Section	
Log Filename	Specify the Log file name and path.
Log Level	Indicate the level of the message to log:
	<b>0</b> No messages.
	<b>1</b> Critical messages.
	<b>2</b> Error and critical messages.
	<b>3</b> Warning, error, and critical messages.
	<b>4</b> Information, warning, error, and critical messages.
	<b>5</b> Debugging, information, warning, error, and critical messages.

# Using the Workbench or Lookup Tool to Configure Finalist

You can use the PBFN Setup Option to configure your Finalist application. To use the PBFN Setup Option to configure your Finalist application, use one of the following methods to access the PBFN Config Setting dialog box:

- From the Workbench, click Tools > PBFN Setup.
- From the Lookup Tool, click Edit > Config.

## Defining Your Files

To define the files for Finalist processing, click the Files tab.

### ***Defining Your Data Files***

To define the data files for Finalist processing, click the Files tab.

1. In the City Directory File field, enter the name and path to the City database file. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
2. In the Zip 4 Directory File field, enter the name and path to the ZIP + 4 database file. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.

### ***Defining Log Messages***

To define the log messages for Finalist processing, click the Files tab.

1. In the Log Filename field, enter the name and path to the Log file. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
2. In the Log Level field, from the drop down list, select the level of the messages to log:
  - 0 — No messages.
  - 1 — Critical messages.
  - 2 — Error and critical messages.
  - 3 — Warning, error, and critical messages.
  - 4 — Information, warning, error, and critical messages.
  - 5 — Debugging, information, warning, error, and critical messages.

## Defining CASS Options

To define the USPS CASS options for Finalist processing, click the CASS tab.

### ***Defining Your CASS-Certified Software***

To define your USPS CASS-certified software for Finalist processing, click the CASS tab.

1. In the CASS Company Name field, enter the name of the CASS-certified company. This information displays in section A1 box 1 on the USPS Form 3553 (CASS Summary Report).
2. In the CASS Product Name field, enter the name of the CASS-certified product. This information displays in section A1 box 2 on the USPS Form 3553 (CASS Summary Report).
3. In the CASS Product Version field, enter the version number for the CASS-certified product. This information displays in section A1 box 2 on the USPS Form 3553 (CASS Summary Report).

### ***Defining Your Z4Change-Certified Software***

To define your Z4Change-Certified software for Finalist processing, click the CASS tab.

1. In the Company Name field, enter the Z4Change-certified company name. This information displays in section A1 box 4 on the USPS Form 3553 (CASS Summary Report).
2. In the Product Name field, enter the Z4Change-certified product name. This information displays in section A1 box 5 on the USPS Form 3553 (CASS Summary Report).
3. In the Product Version field, enter the Z4Change-certified product version number. This information displays in section A1 box 5 on the USPS Form 3553 (CASS Summary Report).

### ***Defining LOT-Certified Software***

To define the LOT-Certified Software for Finalist processing, click the CASS tab.

1. In the Company Name field, enter the Line of Travel (LOT)-certified company name. This information displays in section A1 box 10 on the USPS Form 3553 (CASS Summary Report).
2. In the Product Name field, enter the Line of Travel (LOT)-certified product name. This information displays in section A1 box 11 on the USPS Form 3553 (CASS Summary Report).
3. In the Product Version field, enter the version number for the Line of Travel (LOT)-certified product. This information displays in section A1 box 11 on the USPS Form 3553 (CASS Summary Report).

## Defining Additional CASS Options

To define additional CASS options for Finalist processing, click the CASS Cont'd tab.

### **Defining DPC Changed Certified Options**

To define DPC Changed Certified options for Finalist processing, click the CASS Cont'd tab.

1. In the Company Name field, enter the DirectDPV company name.
2. In the Product Name field, enter the DirectDPV product name.
3. In the Product Version field, enter the version number for the DirectDPV product.

### **Defining CASS Mailer Options**

To define CASS Mailer options for Finalist processing, click the CASS Cont'd tab.

1. In the Name field, enter the mailer's name. This information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
2. In the Address field, enter the mailer's address. This information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
3. In the Address2, Address3, and Address4 fields, enter additional mailer's address information. This information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
4. In the City Line field, enter the mailer's city, state, and ZIP Code information. This information displays in section D box 3 on the USPS Form 3553 (CASS Summary Report).
5. In the Configuration field, enter the CASS configuration value. This information displays in section 1 box 3 on the USPS Form 3553 (CASS Summary Report).

### **Defining CASS Report Options**

To define CASS report options for Finalist processing, click the CASS Cont'd tab.

1. In the Batch Report Filename field, enter the output file name for the Finalist Batch Report or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files. The default file name is batch.txt.
2. In the 3553 Report Filename field, enter the output file name for the USPS Form 3553 (CASS Summary Report) or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files. The default file name is cass3553.txt.
3. In the Batch Report field, click Batch Report to generate the Finalist Batch Report.
4. In the 3553 Report field, click 3553 Report to generate the USPS Form 3553 (CASS Summary Report).

## Defining Process Options

To define your Finalist processing options, click the Process tab.

1. To process in a CASS-certified mode, click CASS Flag. If you process in a CASS-certified mode, Finalist checks expiration dates for bases and engines. The cCASSMode field in the Job Definition File (.job file) should also be set to YES in order to generate the USPS Form 3553 (CASS Summary Report).

NOTE: If CASS Flag = ON and a conflicting option is encountered (Configuration, Assign CR, Return DPBC, LACS<sup>Link</sup>=OFF, Suite<sup>Link</sup>=OFF, or DPV=OFF), a warning message is written to the log file indicating that CASS has been forced off and the reason for CASS being forced off. The message is similar to:

```
Warning Message;CASS forced off: CASS Configuration, Return DPBC,
Assign CR, Assign SuiteLink, Assign LACSLink, Assign DPV
```

2. In the Assign Carrier Routes field, click Assign Carrier Routes.
3. If you are using a platform or coding language that does not support null terminated strings and you need to pad all fields with blanks, click Padded String Data. This field indicates whether you will be using input/output fields that are padded with blanks or null terminated.
4. To assign Line of Travel (LOT) codes to your file, click Assign LOT.
5. To indicate your input data is in an all uppercase format, click Uppercase Input.
6. To performs only eLOT processing, click Process Lot Only. If you choose to perform only eLOT processing, Finalist does not perform address cleansing. A fully-coded address record is passed to the engine and Finalist performs an eLOT lookup and returns the eLOT code.
7. To remove noise characters (unnecessary punctuation and blanks) during Finalist processing, click Remove Noise Chars.
8. To indicate that only previously unassigned or expired addresses should be postal coded in batch mode, click Assign Unassign Records. For more information on the process unassigned records option, see the section "PBFNSetupDef" in your Finalist Reference Guide.
9. To perform normal CASS-certified processing of firms, click Process Firms. If you do not store firm names, or are not interested in matching to a firm level (for non-CASS mode processing only), do not select this option. Finalist will not load or use the firm portion of the database to improve performance.
10. To return abbreviated city names in label lines, click Assign Abbreviated City.
11. To return the input firm, click Return Input Firm.
12. To return delivery point barcodes, click Return DPBC.
13. To perform All Street Matching (ASM) processing, click All Street Matching. All Street Matching (ASM) applies additional matching logic to correct errors in street names and obtain a match. ASM provides the best address validation but may reduce performance. ASM processing is available for U.S. addresses only.

14. To count Carrier Route R777 addresses as deliverable, click R777 Deliverable.
15. To convert secondary information to PMB where appropriate, click Convert Sec to PMB.
16. In the Return SLK Input Secondary field, from the drop down list, select the method to use for returning secondary information when Suite<sup>Link</sup> secondary information is available.
  - **B** — Both. Return both SuiteLink and input secondary information. (This is the Default).
  - **S** — SuiteLink. Return SuiteLink secondary only. Do not return input secondary.
  - **I** — Input. Return input secondary only. Do not return SuiteLink secondary.
  - **N** — None. Do not return SuiteLink secondary or input secondary.
  - Blank — Defaults to B.

**Note:** If Suite<sup>Link</sup> processing does not result in a match, Finalist ignores this option and returns the normal address output. Regardless of the option specified, the output ZIP + 4 is based on the match made using the Suite<sup>Link</sup> secondary. Input secondary information includes the “#” designator for purposes of this option.
17. In the Return Alias Street field, from the drop down list, select the method to use for returning alias street names. This field indicates whether Finalist returns alias street names. An alias street name is an alternate name for a street, maintained at the range Plus4 ZIP Code level.
  - **ON** - If the input address matches to an alias, return the alias. If the input address matches to a base address, but a Preferred alias exists, return the Preferred alias. If the input address matches to a base address and no Preferred alias exists, return the base address.
  - **PXO** - If a Preferred alias exists, return the Preferred alias. If no Preferred alias exists, but an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated or Preferred alias exists, but some other alias type was input, return the input alias. If none of these scenarios apply, return the base street name.
  - **OFF | P** — If a Preferred alias exists, return the Preferred alias. Otherwise, return the base street.
  - **PX** — If a Preferred alias exists, return the Preferred alias. If no Preferred alias exists, but an Abbreviated alias exists, return the Abbreviated alias. If neither exists, return the base street name.
  - **XPO** — If an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated alias exists, but a Preferred alias exists, return the Preferred alias. If no Abbreviated or Preferred alias exists, but some other alias does exist, return the alias. If no alias of any kind exists, return the base street name.
  - **XP | X** — If an Abbreviated alias exists, return the Abbreviated alias. If no Abbreviated alias exists, but a Preferred alias exists, return the Preferred alias. If neither exists, return only the base street name.
  - **Blank** — Defaults to OFF.
18. In the Beginning Frame Char field, enter the character to use for the front framing character for advanced barcodes. If you do not specify a character for Beginning Frame Char or End Frame Char, Finalist uses the exclamation point (!).

19. In the End Frame Char field, enter the character to use for the end framing character for advanced barcodes. If you do not specify a value here, the Beginning Frame character is used. If you do not specify a character for Beginning Frame Char or End Frame Char, Finalist uses the exclamation point (!).
20. In the Suggestion Count field, enter the maximum number of returned suggestions allowed per address based on the specified suggestion criteria.
21. In the Cache Size field, enter the number of buffers to use for internal caching.
  - **0** — Turn off internal caching for Finalist.
  - **XX** — Enter a value between 12-99 for the number of buffers used for internal caching. The default value for z/OS is 30. The default value for all other operating systems is 12.
22. In the Dual Address field, from the drop down list, select the option to use when the input file contains dual addresses (one a conventional address, a second containing a PO Box address). This field determines what order to use to process and match the addresses. If the selected address is valid, processing stops. If the selected address does not validate, Finalist will attempt to code the secondary address. Valid values are:
  - **ACZ** — Above city and ZIP Code line. The address line closest to the last line in the address is given the highest priority in the match process. Any address line above the last line is not used for matching.
  - **L12** — Line 1 in the dual address is given the highest priority in the match process.
  - **L21** — Line 2 in the dual address is given the highest priority in the match process.
  - **A** — The conventional address is given the highest priority in the match process.
  - **P** — The PO Box address is given the highest priority in the match process.
  - **1ST** — The first valid address (in the order Address line 1 and then Address line 2) is given the highest priority in the match process.

When dual addresses are contained on a single line and you select CASS Flag, the USPS address type priority is used in the following order: 1. PO Box 2. Firm 3. Highrise 4. Street 5. Rural Route 6. General Delivery
23. In the CASS Standardize Case field, select an option to indicate whether to return addresses in all upper, all lower, or mixed upper and lower case. This field affects the address coding output only. It does not affect the database APIs (APIs prefixed with PBCS). The database APIs always return upper case data.
  - **U** — Return addresses in all upper case.
  - **L** — Return addresses in all lower case.
  - **Mixed** — Return addresses in upper and lower case.
24. In the Number Exception Table Entries field, enter the maximum number of exception table entries.
25. Click Assign PreciselyID to perform PreciselyID processing.



## Defining Product Options

To define Product options for Finalist processing, click the Product tab.

### Defining Product Files

To define Product files for processing, follow these steps:

1. In the Exception Table File field, enter the Exceptions Table file name and path or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
2. In the EWS Filename field, enter the Early Warning System (EWS) file name and path or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
3. In the LOT Filename field, enter the Line of Travel (LOT) file name and path or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
4. In the DPV Filepath field, enter the Delivery Point Validation (DPV) file path or click on the Browse button (located to the right of the field) to open a dialog box and point to the folder (path) that contains the database(s). This field is not used for z/OS.
5. In the RDI Filepath field, enter the Residential Delivery Indicator (RDI) File path or click on the Browse button (located to the right of the field) to open a dialog box and point to the folder (path) that contains the database(s). This field is not used for z/OS.
6. In the LACS Filepath field, enter the LACS<sup>Link</sup> File path or click on the Browse button (located to the right of the field) to open a dialog box and point to the folder (path) that contains the database(s). This field is not used for z/OS.
7. In the Suite<sup>Link</sup> Filepath field, enter the Suite<sup>Link</sup> File path or click on the Browse button (located to the right of the field) to open a dialog box and point to the folder (path) that contains the database(s). This field is not used for z/OS.

### Defining Additional Processing

To define additional processing, follow these steps.

1. To perform Early Warning System (EWS) processing, click Early Warning System.
2. To perform Residential Delivery Indicator (RDI) processing, click Residential Delivery Indicator.

### Defining DPV Processing

To define your DPV processing options, follow these steps:

**Note:** The USPS CASS regulations require Delivery Point Validation (DPV) processing for CASS certification. If you do not perform DPV processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).

1. From the Mode drop down list, select a value to indicate whether Finalist performs DPV processing and the DPV mode to use for processing your file.
  - OFF — Do not perform Delivery Point Validation (DPV) processing.
  - ON — Perform Delivery Point Validation (DPV) processing with the DPV Full database (dpvh.db).
  - MEM — Perform Delivery Point Validation (DPV) processing with the DPV Full database (dpvh.db) fully in memory for improved performance. You will need a minimum of 650MB of RAM/Memory if you choose this option. MEM is converted to ON with a buffersize (cDPVBufSize) of H (huge).
  - SPL — Perform Delivery Point Validation (DPV) processing using the DPV Split database (dpvs.db). Split File processing separates the DPV Data File into 100 smaller files based on the first two digits of the ZIP Code. The Split File segment associated with the first two digits of the ZIP Code is loaded into memory. If you sort your mailing file by ZIP Code, you can bring the relevant portion of the DPV file into memory. This process reduces the number of I/O requests that normally occurs when you use the full DPV Data File. Use this option if your file is sorted by ZIP Code. This is the optimal solution for Finalist customers having at least 100MB, but no more than 650MB, of RAM/Memory.
  - FLT — Perform Delivery Point Validation (DPV) processing with the DPV Flat database (dpv.db). Use the DPV Flat File for improved batch performance. This file is in excess of 2.3GB and not dependent on RAM/Memory capacity. Input files should be sorted by ZIP Code. This is an optimal solution if you have less than 100MB of RAM/Memory available.
  - Blank — Defaults to OFF.
2. The USPS allows DPV processing to be used as a tie breaker for matching inexact street records. If only one of the records in a tie is delivery point validated, a match is allowed to the inexact record. When processing results in an inexact match due to the input address directional, DPV processing can be used as a tie breaker if only one of the records is found to be delivery point validated and the delivery point validated record does not violate the cardinal direction rule. To perform DPV Tie Breaker processing, click Tie Break. Defaults to ON.

**Note:** The default is ON; however, the default value only applies when DPV Mode is defined as ON, MEM, SPL, or FLT.

**Note:** The USPS CASS regulations require DPV Tie Break processing to generate the USPS Form 3553 (USPS CASS Summary Report).

3. From the Shutdown Indicator drop down list, select a value to specify the action to take during batch processing when a False Positive (Seed) record is encountered.
  - WARNING — Continue batch processing when a False Positive (Seed) violation record is encountered. Finalist displays a warning message indicating a violation occurred. Finalist processing continues but DPV processing is disabled.

If a DPV False Positive (Seed) violation is encountered and WARNING is set, CASS processing is turned off.

CASS processing is turned off for any jobs submitted after the DPV False Positive (Seed) violation was encountered. Any jobs submitted after the DPV False Positive (Seed) violation, fail with an initialization error.

For the job that encountered the False Positive (Seed) violation, the CASS statement will be provided showing the number of records that were DPV confirmed up until the point of the False Positive (Seed) violation. No records will be DPV confirmed after the False Positive (Seed) violation occurred.

Any jobs submitted before the False Positive (Seed) violation was encountered run to completion, including DPV processing for each file. This occurs even if the jobs do not complete processing until after the time the DPV False Positive (Seed) violation was found in the job that first encountered the False Positive (Seed) violation.

- **SHUTDOWN** — Stop batch processing when a False Positive (Seed) violation record is encountered. Finalist does not process subsequent records and automatically exits the program.

The Shutdown Indicator applies to batch processing only. If your DPV key is NCOA or “No-Stop”, this option is ignored.

4. From the Buffer Size drop down list, select a value to indicate the memory model for DPV processing:

- **Pico** — Pico. Stores no data in memory. No tables or indexes are loaded.
- **Ultra** — Ultra-small. Stores no data in memory. Partial indexes are loaded.
- **Small** — Small
- **Medium** — Medium
- **Large** — Large
- **Huge** — Huge. Stores all data in memory.
- **Blank** — Defaults to Medium.

## ***Defining DPV Table Processing***

To define DPV table processing, follow these steps:

1. Private companies offering mailbox rental services to individuals and businesses are Commercial Mail Receiving Agents (CMRA). Select the CMRA processing option:
  - **OFF** — Do not perform CMRA processing.
  - **ON** — Perform CMRA processing.
  - **IN** — Perform CMRA processing. Load the CMRA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **OUT** — Perform CMRA processing. Use the CMRA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.

2. DPV processing uses the Vacant Table to identify delivery addresses that have been active in the past but, according to USPS data, have not been occupied within the last 90 days. Indicate whether to use the Vacant Table and return the proper Vacant code to the output.
  - **OFF** — Do not perform Vacant Table processing.
  - **ON** — Perform Vacant Table processing. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.
  - **IN** — Perform Vacant Table processing. Load the Vacant Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **OUT** — Perform Vacant Table processing. Use the Vacant Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.
  
3. DPV processing uses the Door Not Accessible (DNA) Table to identify delivery addresses where carriers cannot knock on the door for mail delivery or where carriers cannot physically access a residence/building such as rural/highway contact route (HCR), long driveway, or gated community. Indicate whether to use the DNA table and return the result:
  - **OFF** — Do not perform DNA Table processing.
  - **ON** — Perform DNA Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the DNA Table is used in memory or outside of memory. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.
  - **IN** — Perform DNA Table processing. Load the DNA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **OUT** — Perform DNA Table processing. Use the DNA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.
  
4. DPV processing uses the P.O. Box Throwback Table to identify a delivery point that is a street address where mail is not delivered. Instead, delivery is made to the customer's P.O. Box address. Indicate whether to use the P.O. Box Throwback Table and return the result:
  - **OFF** — Do not perform P.O. Box Throwback Table processing.
  - **ON** — Perform P.O. Box Throwback Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the P.O. Box Throwback Table is used in memory or outside of memory. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.
  - **IN** — Perform P.O. Box Throwback Table processing. Load the DNA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).

- **OUT** — Perform P.O. Box Throwback Table processing. Use the DNA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.
5. DPV processing uses the PBSA Table to identify PO Box Street Addresses (PBSA). A PBSA address is a street address that represents a USPS PO Box. This field indicates whether Finalist uses the PBSA Table and returns the result.
- **OFF** — Do not perform PBSA Table processing.
  - **ON** — Perform PBSA Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the PBSA Table is used in memory or outside of memory. For memory loading options, refer to the section "Delivery Point Validation (DPV) Option" in your Finalist Installation Guide.
  - **IN** — Perform PBSA Table processing. Load the PBSA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **OUT** — Perform PBSA Table processing. Use the PBSA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.
6. The DPV No-Stat Table identifies deliveries that are not valid for Computerized Delivery Sequence (CDS) pre-processing. This field indicates whether Finalist uses the No-Stat Table and returns the result.
- **OFF** — Do not perform No-Stat Table processing. This is the default.
  - **ON** — Perform No-Stat Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the No-Stat Table is used in memory or outside of memory. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.
  - **IN** — Perform No-Stat Table processing. Load the No-Stat Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **OUT** — Perform No-Stat Table processing. Use the No-Stat Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
  - **Blank** — Defaults to OFF.
7. DPV processing uses the No Secure Location (NSL) table to identify delivery locations that are not secure. For example, a carrier can access a door but cannot leave a package due to security concerns. The NSL designation alerts mailers to locations where businesses are closed on certain days and locations without mail receptacles (for example, a storefront). Indicate whether to use the NSL Table and return the result:
- **OFF** — Do not perform NSL Table processing.

- **ON** — Perform NSL Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the NSL Table is used in memory or outside of memory. For memory loading options, refer to the section "Maximizing Performance" in your Finalist Installation Guide.
- **IN** — Perform NSL Table processing. Load the NSL Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
- **OUT** — Perform NSL Table processing. Use the NSL Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize).
- **Blank** — Defaults to OFF.

## Defining LACSLink Processing

To define LACSLink processing, follow these steps:

1. To perform LACSLink processing, click LACSLink. The USPS CASS regulations require LACSLink processing. If you do not perform LACSLink processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).
2. From the LACSLink Processing drop down list, select the LACSLink memory model for processing your file:
  - Pico — Pico. Stores no data in memory. No tables or indexes are loaded.
  - Ultra — Ultra-small. Stores no data in memory. Partial indexes are loaded.
  - Small — Small.
  - Medium — Medium.
  - Large — Large.
  - Huge — Huge. Stores all data in memory.
  - Blank — Defaults to Small.

## Defining SuiteLink Processing

To define SuiteLink processing, follow these steps:

1. To perform SuiteLink processing, click SuiteLink. The USPS CASS regulations require SuiteLink processing. If you do not perform SuiteLink processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).
2. From the SuiteLink drop down list, select the SuiteLink memory model for processing your file:
  - Pico — Pico. Stores no data in memory. No tables or indexes are loaded.
  - Ultra — Ultra-small. Stores no data in memory. Partial indexes are loaded.
  - Small — Small
  - Medium — Medium
  - Large — Large
  - Huge — Huge. Stores all data in memory.

- Blank — Defaults to Large.
3. From the Shutdown Indicator drop down list, select a value to specify the action to take when encountering a Suite<sup>Link</sup> error during the processing run:
    - WARNING — Issue warning and turn off SuiteLink processing.
    - SHUTDOWN — Stop Finalist processing when encountering a SuiteLink error.
    - IGNORE — Ignore errors and continue SuiteLink processing.
    - Blank — Defaults to WARNING.

**Note:** USPS regulations require Suite<sup>Link</sup> processing for CASS certification and to generate a USPS Form 3553 (CASS Summary Report). Per the USPS regulations, any job that does not include Suite<sup>Link</sup> processing, must not generate the USPS Form 3553 (CASS Summary Report). If you are processing in a CASS-certified mode (CASS=ON), Finalist sets the Suite<sup>Link</sup> Shutdown Indicator to “S” in order to stop processing if the Suite<sup>Link</sup> initialization fails and to prevent generation of an invalid USPS Form 3553 (CASS Summary Report).

## Defining Your Software Keys

To define your software keys, follow these steps:

1. In the Software Key field, enter your Finalist activation key.
2. In the DPV Key field, enter the Delivery Point Validation (DPV) activation key.
3. In the LACS<sup>Link</sup> Key field, enter the LACS<sup>Link</sup> activation key.

## Defining Report Options

To define your Finalist report options, click the Report tab.

1. In the Report Filename field, enter the Address Detail Report output file name. Defaults to rpt.txt or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
2. In the Report Title field, enter the title of the Address Detail Report.
3. In the Page Length field, enter the maximum page length for the Address Detail Isolation Report.
4. In the Max Output Records field, enter the maximum number of records to output to the Address Detail Isolation Report.
5. From the Report Type drop down list, select the method for printing the Address Detail Report:
  - OFF — Do not print the Address Detail Report.
  - ON — Print the Address Detail Report.
  - ERR — Print only failed addresses on the Address Detail Report.

- Blank — Defaults to OFF.
6. To print the Address Detail Isolation Report, click Report Isol.
  7. To print the Address Detail Suggestion Report, click Report Suggestion.
  8. To print the Address Detail Information Report, click Info Code.



# 3 - Interactive Address Lookup

## In this section

---

Using the Windows Lookup Tool.....	50
Using Finalist CICS.....	56
Using Finalist IMS.....	79



## Using the Windows Lookup Tool

This section describes how to use the Finalist Lookup Tool. The Finalist Lookup Tool runs in a Microsoft Windows environment. The Lookup Tool lets you search the contents of the postal data files for a specific entry. You can use the Lookup Tool to:

- Display a list of streets within a ZIP Code
- Display a list of cities, and the city's ZIP Code, within a state
- Postal code an individual address
- Display last line information for city, state, or ZIP Code

You can use this tool if you cannot find a particular address during regular postal coding. You can also use this tool to obtain the highest-quality address. For example, you may want to determine a directional of N, S, E, or W for an address.

To open the Lookup Tool, follow these steps:

1. Click on the Start button.
2. From the Start menu, select Programs.
3. Select the folder in which Finalist was installed (by default, Precisely Finalist XXX).
4. Click Lookup. The Lookup Tool window displays. You can also access the Lookup Tool from the Workbench by selecting Run Lookup from the Tools menu.

## Using the Database Viewer

You can use the Database Viewer to look at the ZIP + 4 and City Files to determine whether an address is in the database.

### ***Looking for An Address Within a ZIP Code***

To search for an address in the Finalist database using a ZIP Code, follow these steps:

1. Click Database. The Database Viewer dialog box displays.
2. Click the Mode for your search. You can select to choose information for a street, rural route/highway contract, or P. O. Box address.
3. In the ZIP Code field, enter a ZIP Code to view a list of city names within that ZIP Code. After typing at least three numeric characters, a list of city names associated with the first three ZIP Code numeric characters displays. The list of possible city names gets smaller as you key in more of the ZIP Code.
4. Click on a city name to see a list of streets for that city.

- Click on a street name to view detailed information for the street specified. The Primary tab displays high and low range, predirectional, postdirectional, suffix, carrier route, EOB (even/odd/both/consolidated) range indicator, and high and low ZIP Code range, and ZIP Code.

The icons in the first column provide a visual representation of the address type.

- Firm



- Highrise



- Highrise with firm on first floor



- Alias street address



- P. O. Box



- Rural route address



- To view secondary information for a highrise, click on a highrise address. The Secondary Tab displays the secondary information for the specified highrise address. The Secondary Tab displays secondary range, unit, carrier route, EOB (even/odd/both/consolidated) range indicator, and high and low ZIP Code range. If available and applicable, the firm name displays in the Firm Name section. If the street includes a preferred or non-preferred alias street name, the Alias icon displays in the first column.
- To view base street information for an alias, click on an address with an alias icon. The Alias tab displays detailed information for the base address of the alias.

## Looking for An Address Within a City

To search for an address in the Finalist database using a city name, follow these steps:

- Click Database. The Database Viewer displays.
- Click on the City Search Tab.
- Identify the mode for your search. You can select to choose information for a street, rural route/highway contract/military, or P. O. box address. The Mode Indicator is located at the top right corner of the Database Viewer.

4. Enter the city name in the City field to view a list of city names for the specified state. The list of possible city names will get smaller as you key in more of the city name. The list narrows to one city name when the full city name is keyed in.
5. Click on the city name displayed in the Street Name box to see a list of streets for the specified city name.
6. Click on a street name to view detailed information for the street. The Primary Tab on the Database Viewer displays high and low range, predirectional, postdirectional, suffix, carrier route, EOB (even/odd/both/consolidated) range indicator, and high and low +4 range. If "m" displays next to a number in the first column, multiple +4 codes exist for the primary range.
7. To view secondary information for a highrise, click on a highrise address. The Secondary Tab displays the secondary information for the specified highrise address including secondary range, unit, carrier route, EOB (even/odd/both/consolidated) range indicator, and high and low ZIP Code range. If available and applicable, the firm name displays in the Firm Name pane and the multiple +4 codes display in the Mult+4 Type pane.
8. To view base street information for an alias, click on an address with the alias icon. The Alias Tab displays detailed information for the base address of the alias.

## Copying Address Label Information

You can copy the address label information shown in the envelope at the top right area of the Database Viewer dialog box to your selected destination. To copy the address label information from the envelope, right click on the envelope. Click Copy Label to copy the address information to the clipboard. Click Paste at your selected destination to paste the address label information from the clipboard.

## Last Line Lookup

You can use the Lookup Tool Last Line feature to lookup ZIP Code information based on the city and state for an address. To perform a last line lookup, follow these steps:

1. From the Lookup Tool, click Last Line.
2. To find the ZIP Code for an address, type the city in the Last Line box.
3. Click Look Up. The city, state, and ZIP Code information displays.

## Coding an Address

You can use the Lookup Tool Postal Code feature to postal code an address. You must enter at least the range, street name, and either the city/state or ZIP Code address elements.

### ***Coding an Address Using Standard Address Data***

To code an address using standard address data, follow these steps:

1. From the Lookup Tool, click Postal Code. The Code an Address dialog box displays.
2. On the Address Data tab, enter the range, street name and either city/state or ZIP Code. Click Code. Your coded address displays in the Output Label box. If the following options are enabled, additional information displays if applicable.
  - If Delivery Point Validation (DPV) Option is enabled, the delivery point and commercial mail information displays.
  - If the Residential Delivery Indicator (RDI) Option is enabled, the residential delivery indicator information displays.
  - If the Suite<sup>Link</sup> Option is enabled, the match code, fidelity code, and return code displays.
3. To clear all fields on the Code an Address dialog box, click Clear.

### ***Coding an Address Using Parsed Address Data***

To enter parsed address data for coding, follow these steps:

1. From the Lookup Tool, click Postal Code. The Code an Address dialog box displays.
2. From the Parsed Addr tab, enter the address elements in the appropriate fields.
3. Click Code. Your coded address displays in the Output Label box.

### ***Coding an Address Using AddrScan***

To enter address data for coding in a free-form format, follow these steps:

1. From the Lookup Tool, click Postal Code. The Code an Address dialog box displays.
2. From the AddrScan tab, enter the address elements.
3. Use the following buttons to set AddrScan options.
  - Adv — Use Adv to set AddrScan advanced options.
  - Rtn — Use Rtn to set AddrScan return line options.
  - Ord — Use Ord to set the AddrScan return line order.
4. Click Code. Your coded address displays in the Output Label box.
5. If your entered address data does not result in a coded address, you may receive a message that coding suggestions are available. For more information on coding suggestions, refer to the discussion on the Suggestion List Dialog Box.

6. To view detailed AddrScan return information, click AS Rtn Info. The AddrScan Return Information dialog displays the standard output, label output, and return information.

### ***Coding an Address Using Address Components***

To enter address data for coding using city, state, ZIP Code, and ZIP + 4 components, follow these steps:

1. From the Lookup Tool, click Postal Code. The Code an Address dialog box displays.
2. From the Address Data2 tab, enter the secondary information separately from the address line.
3. Click Code. Your coded address displays in the Output Label box.

### ***Analyzing Coding Results***

For information on how your address was coded, follow these steps:

1. After coding an address, from the Code an Address dialog box, click Address Info.
2. The Address Information screen displays address coding statistics including match level, location settings, address type, and components.

### ***Viewing a Coded Address in a Parsed Format***

For information on how your address was coded, follow these steps:

1. After coding an address, from the Code an Address dialog box, click Parsed Info.
2. The Parsed Information dialog box displays your address separated into individual address elements. The More tab displays additional information for your address. The Alternate tab displays alternate address information.

### ***Viewing Address Coding Suggestions***

To view address coding suggestions if your first attempt does not result in a coded address, follow these steps:

1. After attempting to code an address, from the Code an Address dialog box, click Suggestions.
2. The Suggestion List dialog box displays.
3. Highlight the correct address and click Select. The Code an Address dialog box displays your selected address.
4. Click Code and the postal coded address now displays in the envelope. If you do not see your correct address, click Cancel to exit this screen.

### ***Viewing Return Address Information***

To view return address data information, follow these steps:

1. Click Process Info.
2. The Standard Address Data Returned screen displays detailed address information for your coded address.

## Using Phonetics to Lookup an Address

You can use the Lookup Tool Phonetics feature to look up street names based on the phonetic sound of the information you have. To use the Lookup Tool Phonetics feature, follow these steps:

1. From the Lookup Tool, click Phonetics. The Phonetic Search dialog box displays.
2. Enter the street name. As you type the street name, the Phonetic Search dialog box displays the phonetic name in the Phonetic Name field and suggestions for a street name you can use for a postal coding attempt.

## Using the Lookup Tool to Define Configuration Settings

You can use the Lookup Tool to configure your Finalist application.

1. Start the Lookup Tool using one of these methods.
  - a) To start the Lookup Tool from the Finalist Workbench, from the Tools menu, click Lookup Tool.
  - b) Click the Lookup Tool icon on the Workbench toolbar.
  - c) Click Start > Programs > Precisely Finalist XXX > Lookup.  
In "FinalistXXX", "XXX" represents the current version.
2. To define or edit configuration settings, go to Edit, and select Config. The PBFN Config Setting dialog box displays.
3. For information on using the tabs on the PBFN Config Setting dialog box to configure your Finalist application, refer to [Using the Workbench or Lookup Tool to Configure Finalist](#) on page 35.

## Accessing the Finalist Help

To access the entire set of Finalist documentation from a single access point in an online Help format, from the Lookup Tool, go to Help, and then select Contents.

You can also go to Help, and then select Installation Help, User's Help, or Reference Help to access a more specific type of help.

## Accessing Version Information

You can use the About Lookup Option to access current version information. From Help, click About Lookup. The About Lookup dialog box displays the current version information for the Finalist Lookup

Tool including whether you are processing in a 32-bit or 64-bit mode and the expiration date for your software key. Click OK to return to the Lookup Tool.

## Using Finalist CICS

Finalist CICS performs Finalist processing in an on-line, real-time environment. After you enter address data for processing, CICS returns the same information as Finalist batch processing.

**Table 2: Finalist CICS Transactions**

Transaction	Description
LPCT	Simple transaction to do basic address validation using ZIP + 4 data.
PBFN	Simple transaction to do basic address validation using ZIP + 4 data and the Finalist native interface. Member PBFNDRVC in the CISOURCE installation PDS contains the source for this transaction.
LPCF	Complex transaction that lets you use various address coding options when processing an address. LPCF also provides advanced return codes and additional lookup options.

## Using the LPCT Transaction

1. Enter the LPCT transaction name.
2. Press Enter. The Inquiry Input screen displays.



3. Follow the information on the screen.

```
FINALIST ON-LINE N.N

PLEASE ENTER THE FOLLOWING INFORMATION:

EXCEPTION TABLE ==> (Y/N) N

FIRM LINE ==>
URB LINE ==>
ADDR LINE 1 ==>
ADDR LINE 2 ==>
CITY/STATE ==>
ZIPCODE ==>
SECSG ==>
CR ==>

PRESS ENTER TO CONTINUE - CLEAR TO CANCEL
```

## Using the PBFN Transaction

1. Enter the transaction name PBFN. The Inquiry Input screen displays.
2. Follow the information on the screen.

```

FINALIST ON-LINE N.N

Option ==>   (Select one of the following)

      1 Finalist Address Lookup          2 Product Information

Processing Options (Y or N):
Exception Table      ==> N             Delivery Point Validation ==> N
Early Warning System ==> N             LACSLink                 ==> N
RDI                  ==> N             SuiteLink                 ==> N

Firm Line   ==>
URB Line    ==>
Addr Line 1 ==>
Addr Line 2 ==>
City/State  ==>
ZIPCode     ==>          SecSg ==>          CarRte ==>

=====
PF1: Opt 1   PF2: Opt 2   PF3:         PF4 :         PF5 :         PF6 :
PF7:         PF8:         PF9: Reset   PF10: Clear   PF11:         PF12: Exit

```

## Using the LPCF Transaction

1. Type the transaction name **LPCF**.
2. The **Main Menu** displays.

```

FINALIST ON-LINE N.N
Option Code: 1 (Valid options listed below)                HH:MM:SS
1-Address Lookup      3-ZIPCODE Information      5-Street Information
2-City Information    4-Street Name List      6-Product Information

Exception Table: N (Y/N)          ALSLBL : N (Y/N/1/2/3/4/5/6)
EWS: N (Y/N)                    FIRMLBL: D (I/D)
RDI: N (Y/N)                     ASM : N (Y/N)
LACSLink: N (Y/N)                DUALADR: F (F/A/1/2/P/C)
SuiteLink: N (Y/N)               CTYLONG: Y (Y/N)
SLK Secondary: B (B/S/I/N)       R777 Deliverable: Y (Y/N)
DPV: N (Y/N/S/F)                 Cnvrt Sec to PMB: N (Y/N)
NoStat: Y CMRA: Y Tie Break: Y (Y/N)
Vacant: Y PBSA: Y DNA: Y (Y/N)
THRWBK: Y      NSL: Y (Y/N)

PF1: OPT 1   PF2: OPT 2   PF3: OPT 3   PF4 : OPT 4   PF5 : OPT 5   PF6 : OPT 6
PF7: N/A     PF8: N/A     PF9: RESET   PF10: N/A    PF11: N/A    PF12: EXIT

```



- Use **PF10** to toggle between the Return and Reason Codes screen and the Address Information Codes screen.

```

FINALIST ON-LINE N.N

                                RETURN AND REASON CODES

TYPE   CODE   VALUE   MEANING                                HH:MM:SS

RETURN  1   :   0   = ZIP, CARRIER ROUTE AND ZIP+4 ASSIGNED
REASON  1   :   0   = ZIP CODE VERIFIED
        2   :   0   = CITY VERIFIED AS INPUT
        3   :   1   = CARRIER ROUTE RETURNED
        4   :   1   = ZIP+4 RETURNED
        5   :   1   = STREET STANDARDIZED FROM INPUT
        6   :   0   = STREET RANGE CORRECT
        7   :   0   = SUFFIX AND DIRECTIONAL CORRECT
        8   :   0   = INPUT STREET IS NOT AN ALIAS
        9   :   2   = FIRM NAME CHANGED FOR MATCHING ONLY
       10  :   2   = UNIT DESIGNATOR ABBREVIATED
       11  :   0   = UNIT NUMBER (IF ANY) CORRECT
       12  :   0   = NON-CONVENTIONAL ADDR. GOOD (IF PRESENT)

PF KEYS =====
PF1: N/A  PF2: N/A  PF3: N/A  PF4 : N/A  PF5 : N/A  PF6 : N/A
PF7: N/A  PF8: N/A  PF9: N/A  PF10: AICODES PF11: RETINFO PF12: EXIT

```

- Press **PF12** to exit.

### Address Information Codes Screen

- From the Reason Codes screen, press **PF10**.
- The Address Information Codes screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N

                                ADDRESS INFORMATION CODES

TYPE   CODE   VALUE   MEANING                                HH:MM:SS

INFO   1   :   0   = SINGLE ZIP CODE CITY
        2   :   5   = FIRM ADDRESS
        3   :   0   = CITY DELIVERY
        4   :   7   = FIRM ON FIRM LN; DELIVERY ADDR ON LINE 1
        5   :   0   = ADDRESS FOUND ON PRIMARY LOOKUP
       6-8 : 200 = BUILDING HOUSING ONE OR MORE FIRMS
        9   :   7   = FIRM AT HIGHRISE ADDRESS CODED
       10  :   9   = ZIPMOVE FORWARDING DID NOT OCCUR

PF KEYS =====
PF1: N/A  PF2: N/A  PF3: N/A  PF4 : N/A  PF5 : N/A  PF6 : N/A
PF7: N/A  PF8: N/A  PF9: N/A  PF10: RETCODE PF11: N/A  PF12: EXIT

```

3. Press **PF10** to toggle between the Return and Reason Codes screen and the Address Information Codes screen.
4. To exit, press **PF12**.

For more information on available options, see "Using the Compatibility Interface", in your Finalist Reference Guide.

### Return Area Screen

1. From the Reason and Return Code screen, press **PF11**.
2. The Return Area screen displays. This screen does not accept input. Error information is not available on this screen.
3. To return to the Reason and Return Code screen, press **PF11**.
4. To exit from the Return Area screen and return to the Address Lookup screen, press **PF12**.

A failed LINK, LOAD, or XCTL may be caused by a missing RDO entry, a missing program, or a missing library. Missing or wrong-length COMMAREAs may result from mixed releases.

### City Information Screen

1. From the **Address Lookup** screen, press **PF2**.
2. The **City Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N
                                CITY NAME LIST
                                HH:MM:SS
INPUT CITY : XXXXXXXXXXXXXXXX INPUT STATE: XX (2 CHAR ABRV)

  CITY NAMES      ZIPCODES      CITY NAMES      ZIP CODES
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN      XXXXXXXXXXXXXXX  NNNNN

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6:  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT
    
```

3. Press **PF12** to exit.

### ZIP Code Information Screen

1. From the **Address Lookup** screen, press **PF3**.
2. The **ZIP Code Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N
                                ZIPCODE LIST
                                HH:MM:SS

STARTING ZIPCODE:  NNNNN

  ZIPCODE      CITY NAME      STATE  COUNTY  MOTHER  MULTIZIP  MULTICITY
                CITY          ZIPCODE  CITY      ZIP

  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X
  NNNNN      XXXXXXXXXXXXXXXX  AA     NNNNN  NNNNN    X         X

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6:  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT
    
```

3. Press **PF12** to exit.

### Street Name List

1. From the **Address Lookup** screen, press **PF4**.

- The **Street Name Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N

                                STREET NAME LIST
                                HH:MM:SS

INPUT ZIPCODE:  NNNNN
INPUT CITY   :  XXXXXXXXXXXXXXXXXXXX INPUT STATE: XX
INPUT STREET :  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

      STREET NAME                                STREET NAME

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6 :  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT
    
```

- Press **PF12** to exit.

## Street Information Screen

FINALIST ON-LINE N.N						STREET RANGE LIST				
						HH:MM:SS				
INPUT ZIPCODE: NNNNN										
INPUT CITY : XXXXXXXXXXXXXXXX						INPUT STATE: XX				
INPUT STREET : XXXXXXXXXXXXXXXXXXXXXXXX										
LOW RANGE	HIGH RANGE	PRE DIR	POST DIR	SUF	ALS IND	SEC/SEG EVEN	SEC/SEG ODD	CAR EVEN	RTE ODD	ZIP
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
XXXXXXXXXX	XXXXXXXXXX	AA	AA	AA	A	NNNN	NNNN	XXXXX	XXXXX	NNNNN
*****ERROR MESSAGE*****										
PF KEYS=====										
PF1: N/A	PF2: N/A	PF3: N/A	PF4: N/A	PF5: N/A	PF6: N/A					
PF7: UP	PF8: DOWN	PF9: N/A	PF10: N/A	PF11: N/A	PF12: EXIT					

## CICS Product Information

The Product Information Screen provides information for the Finalist CICS product.

**Note:** The EWSDATA File Copyright Date does not display on the Product Information Screen until you process a record with the EWS Option turned on.



```

FINALIST ON-LINE N.N.N
                                     PRODUCT INFO   HH:MM:SS

FINALIST STREET DATA FILE
VERSION          : N.NN.NN.N.NN
CREATED DATE     : MM-DD-YYYY
EXPIRATION DATE  : MM-DD-YYYY
TYPE             : FULL (ZIP, ZIP+4 AND CARRIER ROUTES)
COVERAGE        : NATIONAL

FINALIST CITY FILE                      USPS EWS DATA FILE
VERSION          : N.NN.NN.N.NN        COPYRIGHT DATE: MM-DD-YYYY
CREATED DATE     : MM-DD-YYYY
EXPIRATION DATE  : MM-DD-YYYY

SOFTWARE VERSIONS
ENGINE           : N.NN.NN.N.NN
EXPIRATION DATE  : MM-DD-YYYY

PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6 : N/A
PF7:  N/A   PF8:  N/A   PF9:  N/A   PF10: N/A  PF11: N/A  PF12: EXIT

```

## Troubleshooting

The LPCF transaction handles errors by displaying a return code and a window message overlay on any screen where an error occurs.

## LPCF Extended Capabilities

The Finalist LPCF transaction does not have input/output (I/O) capabilities other than the use of the Finalist master files. System exit points help update a file's addresses. These exit points can call I/O routines. One exit point is called to read an address into the Address Lookup Screen (input exit). The other is used to write addresses from the Address Lookup Screen (output exit).

Use these exits to read an input file (PF7) and ensure that an address is correct. After validating and/or correcting an address, the output exit (PF8) can write the corrected address to a new file or the file used as input. This is a highly effective way of ensuring address files are correct and contain deliverable addresses.

You can call the input exit using the PF7 key from the address lookup display. You can call the output exit using the PF8 key.

### ***Using the Exit Programs***

The exit points are coded as CICS links to the exit programs. These programs exist as small stubs when Finalist CICS is delivered. Each stub can be replaced in the load library by a user-written

program of the same name. The programs are called by the system via a CICS link command to the following programs:

- LPWN509C program for the input exit.
- LPWN510C program for the output exit.

Both programs are passed a COMMAREA of the same format. The COMMAREA passed to the exit routines is defined in the source member LPWN020D (LPWN021D for COBOL) on the Finalist distribution. LPWN020D contains a copy statement for LPFNCL04 (copybooks LPFNCL01 for COBOL and LPFNCL04 for Assembler) that is a source member on the Finalist distribution.

**Table 3: Exit COMMAREA Fields**

Field	Description
EXITRC (BYTES 01-02)	Exit program return code
EXITDSL (BYTES 03-03)	<p>Display Security Level:</p> <p>Forty bytes of the area passed between the input and the output exits are displayed for possible update on the address lookup screen in a field labeled "ADDR KEY". Display security level allows you to protect those 40 bytes as follows:</p> <p>Code U-Allow update of the field</p> <p>Code V-Only allow users to view the field</p> <p>Code H-Hide the field entirely</p> <p>(Any value other than "V" or "H" defaults to "U". See EXITPASS below.)</p>
EXITREQW (BYTES 04-04)	<p>Require rewrite switch</p> <p>Do not allow the user to execute the input exit a second time until the output exit has been called if the value of this field is "Y". This is accomplished by disabling the 'read' and 'clear' PF keys until the "save" PF key has been pressed.</p>
EXITPAS4 (BYTES 05-08)	<p>User defined data - not updateable</p> <p>This field is initialized to binary zeros when the output exit is called on an address not read from the input exit. Note that an address is considered to have been read if the input exit was executed and you have not hit the "write" or "clear" PF key. This is true even if the address is altered beyond recognition.</p>

Field	Description
EXITPASS (BYTES 09-48)	<p>User defined data - passed from the input exit to the output exit.</p> <p>This field also displays on the address lookup screen as "ADDR KEY" for update if it is not locked by the input exit.</p> <p>If this field is not locked, it is updated from the address lookup screen. In this case, any bytes containing low values (X'00') are compressed out by CICS mapping. Significant characters to the right of the low value bytes shift left during compression. This causes a potential for inadvertent change.</p> <p>If this field is locked against update, it is passed unchanged and no compression occurs.</p>
FINALIST CALLAREA	<p>This field immediately follows the exit information defined above. This field's length and format are described in the Finalist Reference Guide. Distributed source members LPFNCL01 and LPFNCL04 map this area in COBOL and Assembler respectively. This same area is passed to Finalist and should be filled in by the input exit as for a call to Finalist. This area should be interpreted by the output exit as if it was returned directly from Finalist.</p>

### ***Input Exit Point***

This section describes the input exit point including the data passed to the exit routine on a call and how the data from the exit routine is processed when returned to the Finalist CICS system. The list shown next describes the steps for a call to the input exit.

1. Press the PF7 key.
2. The ADDR key is taken from the Address Lookup screen and placed in the COMMAREA.
3. The Finalist CICS system links to the exit program (LPWN509C) and passes the COMMAREA.
4. The LPWN509C program processing starts.
5. The LPWN509C program returns control to the Windows programs.
6. The input address and the 40-byte passthrough area in the COMMAREA are formatted onto the Address Lookup screen.
7. The Finalist function and tailoring options (set on the main menu) are restored. The function and tailoring options have priority over those in the COMMAREA.
8. If the value for the require rewrite switch in the COMMAREA is "Y", the read and clear PF keys are disabled on address lookup.
9. If the return code from the exit is not zero, the address lookup screen displays at this point without any further processing. The exit return code displays. Otherwise, Finalist is called with the address passed back by the exit as if the address had been entered on the address lookup screen. The results display on the Address Lookup screen.

## Output Exit Point

The exit points are coded as CICS links to the exit programs. These programs exist as small stubs when Finalist CICS is delivered. Each stub can be replaced in the load library by a user-written program of the same name. The programs are called by the system via a CICS link command to the following programs:

This section describes the output exit point including how the output exit point works, the data passed to the exit routine on a call, and the action taking place after the exit routine returns control to the Finalist program. The list shown next describes the steps for a call to the output exit point.

1. You must process the address on the Address Lookup screen before actually executing the exit program. This includes a coding attempt through Finalist. A coding attempt is defined as either of the following:

- An address is keyed into the Address Lookup screen and the Enter key is pressed.
- The input exit routine returns an address and a return code of "00".

The coding attempt builds the FINALIST CALLAREA. This CALLAREA is part of the COMMAREA passed to the exit routine. If the exit display security level (EXITDSL) is set for update, the field EXITPASS is updated from the address key field on the address lookup screen at this time.

2. Press PF8 to execute the output exit routine.
3. The Finalist system links to the LPWN510C exit program and passes the COMMAREA described in [Using the Exit Programs](#) on page 65.
4. The LPWN510C program returns control to the Finalist programs.
5. A message displays the exit return code on the Address Lookup screen.
6. The read and clear PF keys, PF7 and PF10 respectively, are enabled. This occurs in case these keys were disabled by the require rewrite switch from the input exit.
7. The exit return code initializes to character zeros.
8. The four-byte, hidden pass-through area initializes to binary zeros.
9. The Addr Key field on the Address Lookup screen is unlocked. This occurs in case the key was locked by the exit display security level from the input exit routine.
10. The Address Lookup screen displays with the original address and Addr Key data left intact.

## Programming Notes and Tips

The exit points were developed to provide a means of reading and writing addresses. These exits can eliminate keying the addresses as input and manually transferring the results of Finalist CICS processing to the proper location for use. The following subsections provide several scenarios for using the exits. These examples do not represent a complete list of uses for the exits. The examples provide a few helpful tips in using the exit points.

**Note:** There may be times when it is desirable to disable the exits. For details on disabling (locking) the exits, refer to [Pass Control Block](#) on page 73.

### Figure 1: Sequentially Processed Files

The first scenario involves using a sequentially processed file of addresses through Finalist CICS. Each address is read using the input exit program and written using the output exit program. The exit COMMAREA is mapped by distributed source members LPWN020D (Assembler) and LPWN021D (COBOL).

The exit common area contains a safety feature called EXITREQW (the require rewrite switch). If this switch is set to "Y" during the input exit, you are not able to process the next record until the output exit has been called for the current record. This setting prevents records from being accidentally skipped. For more information, see [Using the Exit Programs](#) on page 65. Once the record is processed, the output routine may need key information to update the record. The two ways to pass this information are described next.

- A four-byte PASSAREA (EXITPAS4): The four-byte PASSAREA is never seen on the screen. This is an ideal use for passing data such as relative record numbers or database keys which only have meaning to a computer.
- A 40-byte PASSTHROUGH area (EXITPASS): The PASSTHROUGH area is shown on the Address Lookup screen. The input routine may set a flag to determine whether this data should be:
  - Hidden from the user entirely
  - Shown but locked against updating
  - Shown with update capability

See source member LPWN020D (Assembler) and/or LPWN021D (COBOL) for more information.

### **Figure 2: Using CICS as the Original Entry Screen**

The second scenario involves the ability to enter addresses using the Finalist CICS product as the original entry screen. It may be helpful when entering a new address to track whose address is being entered. Key information relating the address to the addressee is keyed into the Addr Key field on the Address Lookup Screen.

Records going to the output exit may have been manually entered or read with the input exit. It may be necessary to distinguish between the two. One way to accomplish this is to take advantage of the four-byte invisible PASSTHROUGH field. This field is initialized to binary zeros after each call to the output exit routine or when PF10 is pressed to clear the Address Lookup Screen. If the input exit routine never passes binary zeros to this field, manually entered addresses are distinguishable from records read using the input exit.

### **Figure 3: Reading Input Records Directly**

In the third scenario, the input exit reads input records directly rather than sequentially. A record key is needed to find the desired record. The Addr Key field on the Address Lookup Screen is passed to the input exit in the 40-byte PASSTHROUGH and can be used to provide this information to the input exit routine.

### **Figure 4: Deleting Records**

You may want the output routine to delete rather than update a record. Write the output routine to recognize a specific value in the Addr Key field or one of the address line fields as a delete key.

### **Figure 5: Reading Input Without an Automatic Coding Attempt**

Input from the input exit routine with an exit return code of "00" results in a system attempt to code the incoming address through Finalist. Any other return code value displays on the Address Lookup screen with the returned address data. No attempt is made to code the address through Finalist. If you do not want immediate Finalist processing, return a non-zero code.

## ***Calling Finalist CICS From Another Application***

You can call Finalist CICS from another application to:

- Access Finalist CICS system from another application
- Process an address
- Return to the calling application directly from Finalist CICS system

Address information in the same format as a call to Finalist can be carried to and from the Finalist CICS application. Any application used in this way must meet the requirements specified in this chapter. The list below contains items you should know when using an application to call the Finalist CICS product.

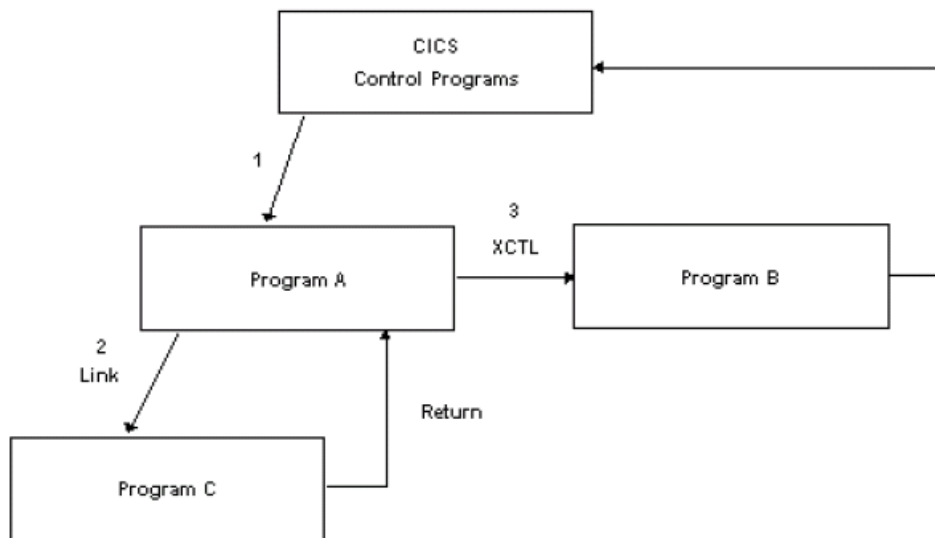
1. The Address Lookup Screen displays first when you access the Finalist CICS system. This screen will look as if you just executed the input exit routine using the PF7 "read" key. This is because the data passed from the calling application closely resembles the data passed back from the input exit routine.
2. Values usually entered on the Main Menu Screen are passed to the Windows programs by the calling application. These values are viewed and updated by accessing the Main Menu Screen from the Address Lookup Screen.
3. Press Clear from any screen or PF12 from the Main Menu Screen to return to the calling application. The Clear key here refers to the Clear key on your terminal and not the clear-screen function PF10 key.
4. The address data returned to the calling application comes from a call to Finalist. This call to Finalist occurs when you press Enter on the Address Lookup Screen. This call can also occur when Finalist CICS receives control from the calling application if the call area's exit return code field contains zeros. Address data entered after the last call to Finalist is not passed back to the calling application.

## **Technical Background Information**

### ***Finalist Program Linkage***

Finalist CICS is a pseudo-conversational dialog. This means control returns to CICS after a screen displays, and returns to the application each time you press an attention key (for example, Enter,

Clear, etc.). CICS applications are broken down into hierarchical levels according to the manner of linkage between programs. This illustration shows how the program linkage works.



**Figure 6: Program Linkage**

1. CICS passes control to program A. Program A is on the first level of hierarchy below CICS.
2. Program A links to Program C. This link establishes a lower hierarchical level. When program C returns, control passes to program A rather than to CICS.
3. Program A performs an XCTL to program B. XCTL does not establish a new hierarchical level. When program B returns, control passes to CICS.

The CICS return command only returns control to the next higher level of hierarchy above the returning program. A pseudo-conversational program must be at the hierarchical level immediately below CICS to return control to CICS. This means you cannot link to Finalist CICS as you would link to FINALOL.

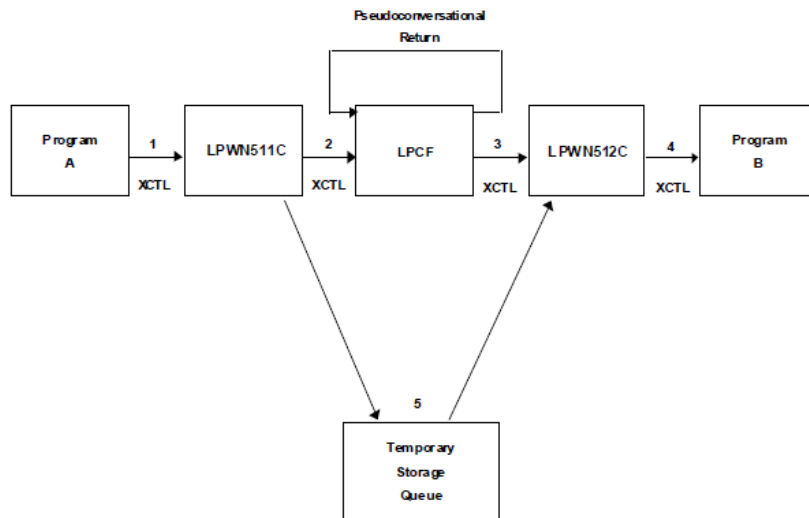
A program on the calling application's top hierarchical level must XCTL to Finalist CICS. In turn, CICS must XCTL back to the calling application to preserve the current hierarchical level. This can lead to complications as all dynamic storage obtained by the program doing the XCTL command is lost after the XCTL unless passed in a COMMAREA.

The calling application must pass all data it will need after CICS processing. This data is passed through CICS via the COMMAREA. The size of the data in the calling application's COMMAREA is not available to CICS. Therefore, it can not carry all the data in its own COMMAREA.

Finalist CICS uses a temporary storage queue to hold the calling application's COMMAREA. When CICS processing is complete, the calling application's COMMAREA is rebuilt. The temporary storage queue is deleted.

## Callable CICS Linkage

This illustration shows how the callable Finalist CICS linkage works.



**Figure 7: Callable Linkage**

1. Your program COMMAREA contains user data with additional data added to the end for use by Finalist CICS.
2. XCTL 1 passes the full COMMAREA to LPWN511C.
3. LPWN511C formats a Windows COMMAREA to pass to Windows (LPCF). The rest of the COMMAREA is saved in a temporary storage queue. The TS Queue is named LPCFxxxx where “xxxx”=EIBTRMID. LPWN511C saves the transaction code and changes it to LPCF. The Windows COMMAREA is passed to Windows in XCTL 2.
4. Finalist CICS (LPCF in the previous figure) runs as usual until you press Clear ending CICS processing. Prompted by a flag passed by LPWN511C, CICS XCTLs to LPWN512C instead of returning to CICS. The CICS COMMAREA is passed in the XCTL.
5. LPWN512C performs the following:
  - Reads and deletes the temporary storage queue
  - Merges the CICS CALLAREA into your COMMAREA
  - Restores your transaction code
  - XCTL's to a program named by you
    - This program can be the same one that passed control to CICS or a different program. This is a user decision. The user program (Program B in the previous figure) is responsible for knowing control is coming from CICS.
    - If the field PASSPGM in the Pass Control Block is not filled in by your application prior to calling LPW511C, abend LP71 occurs when Finalist CICS ends and attempts the XCTL.



## Technical Specifications for Calling Finalist CICS

Finalist CICS is called externally using the following CICS command:

```
EXEC CICS XCTL PROGRAM('LPWN511C') COMMAREA(dataname) LENGTH(datalen)
```

The "dataname" above refers to the name of the COMMAREA to pass. The "datalen" refers to the length of the COMMAREA to pass.

**Note:** This must always be an XCTL and not a LINK. Never include this in a program called via a CICS LINK command. The Finalist CICS system must be at the highest level of hierarchy below CICS.

## COMMAREA Passed to Finalist CICS

The COMMAREA passed to CICS in the XCTL command includes:

- Data important to the calling application
- A pass control block defining the call to CICS
- Exit COMMAREA image

Data is not allowed between the pass control block and the exit COMMAREA image. Also, data is not allowed after the exit COMMAREA image. These areas are located by subtracting their lengths from the end of the whole COMMAREA.

## Pass Control Block

The Pass Control block is defined in the source member LPWN030D in the Finalist distribution. The CICS programs assume the pass control block begins on a doubleword boundary. This table describes the pass control block fields.

Field	Description
PASSPGM	Bytes 01-08. Program to XCTL to when CICS processing is finished.
PASSTRAN	Bytes 09-12. Alternate Finalist CICS transaction code. On blank or low-values, this code defaults to LPCF. This code can be used if you renamed LPCF in your own CICS PCT entry.
PASSCNFG	Bytes 13-15. Finalist configuration value. Valid values are those that are currently CASS-certified.
PASSUNIQ	Byte 16. Finalist tailoring option for unique ZIP Codes. (One byte. Value "Y" or "N").
PASSSTPH	Byte 17. Finalist tailoring option for street phonetics. (One byte. Value "Y" or "N").
PASSFRMC	Byte 18. Finalist tailoring option for firm correction. (One byte. Value "Y" or "N").

Field	Description
PASSCTPH	Byte 19. Finalist tailoring option for city phonetics. (One byte. Value "Y" or "N").
PASSCWGT	No longer supported.
PASSSWGTT	No longer supported.
PASSZPCR	Byte 22. Finalist tailoring option for ZIP correct. (One byte. Value "Y" or "N").
PASSCTCR	Byte 23. Finalist tailoring option for city correct. (One byte. Value "Y" or "N").
PASSSTRC	Byte 24. Finalist tailoring option for street cosmetics. (One byte. Value "Y" or "N").
PASSFIRM	Byte 25. Finalist tailoring option for firm parse. (One byte. Value "Y" or "N").
PASSUNIT	Byte 26. Finalist tailoring option for unit designator. (One byte. Value "Y" or "N").
PASSCTYL	Byte 27. Finalist tailoring option for long city name. (One byte. Value "Y" or "N").
PASSALSL	Byte 28. Finalist tailoring option for alias street name in the Finalist label format area. (One byte. Value "Y" or "N" or '1' through '6').
PASSFRML	Byte 29. Finalist tailoring option for firm name in the Finalist label format area. (One byte. Value "D" or "I").
PASSEXPS	Bytes 30-32. Finalist future expansion space - leave blank. (three bytes).
PASSEXCP	Byte 33. Finalist CICS control switch. Exceptions processing. Y = Use exceptions table in Finalist processing.
FILLER	Byte 34. Reserved for future use.
PASSEX1L	Byte 35. Finalist CICS control switch. Lock input exit. L = Input exit (LPWN509C) can not be accessed while executing Finalist CICS.
PASSEX2L	Byte 36. Finalist CICS control switch. Lock output exit. L = Output exit (LPWN510C) can not be accessed while executing Finalist CICS.
PASSWCRI	Byte 37. Finalist CICS control switch. Finalist CICS call return indicator. Finalist CICS returns a "Y" in this field indicating a return from callable Finalist CICS.
Bytes 38-48	Reserved for future use.

## Exit COMMAREA Image

The exit COMMAREA image consists of an exit control block followed by the FINALIST CALLAREA. This is the same format as the COMMAREA passed to the user exit routines.

## Sample Program for Calling Finalist CICS

The program shown next illustrates one way to call Finalist CICS from another application. In this case, one program passes control to Finalist CICS and instructs Finalist CICS to pass control back to the same program calling it. You can also set up Finalist CICS to pass control to any other program. When finished, the Finalist CICS system passes control to any program per your instructions.

```

EXAMPLEX TITLE '- CICS WINDOWING APPLICATION DRIVER'
      DFHEIGBL ,
COMMDSCT DSECT
COMMDATA DS 0D
*     COPY MAPDS (MAPSET DSECT)
*     :
*     (MISCELLANEOUS DATA)
*     :
      DS 2048C
      COPY LPWN030D (PASS CONTROL BLOCK)
      COPY LPWN020D (EXIT COMMAREA IMAGE)
PASSLEN EQU *-PASSCNTL
COMMLen EQU *-COMMDATA
      EJECT
DFHEISTG DSECT
      DFHEISTG
*     :
*     (WORK FIELDS USED ONLY WITHIN 1 PASS THRU THIS PROGRAM)
*     :
ABCODE  DS      CL4                HOLD ABEND CODE FOR ABEND COMMAND
      DS 0D
      DFHEIEND
      EJECT
      REGEQU
      EJECT
EXAMPLEX DFHEIENT CODEREG=(3),DATAREG=(13),EIBREG=(11)
      USING COMMDSCT,R5
*     :
*     (USER APPLICATION LOGIC)
*     :
      CLC EIBCALEN,=H'0' IF NO COMMON AREA PRESENT,
      BNE NOTFIRST GET SPACE TO PUT IT IN.
      MVC  ABCODE,USR1              SET ABEND CODE JUST IN CASE
      EXEC CICS HANDLE CONDITION NOSTG(ABENDIT)
      EXEC CICS GETMAIN SET(R5) FLENGTH(=A(COMMLen)) INITIMG('X'00')
      MVI PASSWCRI,C'N' RESET WINDOWS RETURN SWITCH
      B NOTRETN
NOTFIRST DS 0H
      L R5,DFHEICAP A(COMMAREA)
      CLI PASSWCRI,C'Y' IF RETURNING FROM WINDOWS

```

```

        BE PROCRETN PROCESS RETURNED DATA
NOTRETN DS 0H
*      :
*      (USER APPLICATION LOGIC)
*      :
*      (FILL IN ADDRESS DATA TO PASS TO WINDOWS)
*      :
*      (FILL IN EXIT CONTROL BLOCK TO PASS TO WINDOWS)
*      :
MVC PASSPGM,=C'EXAMPLEX' RETURN TO EXAMPLEX PROGRAM AFTER WINDOWS
MVC PASSTRAN,=C'LPCF' WINDOWS TRAN CODE IS LPCF
MVI PASSEX1L,C'L' LOCK INPUT EXIT
MVI PASSEX2L,C'L' LOCK OUTPUT EXIT
MVI PASSEXP,C'N' NO EXCEPTION TABLE
MVC PASSTALR,=CL13' ' BLANK TAILORING GETS DEFAULTS
SPACE
EXEC CICS XCTL PROGRAM(=C'LPWN511C') COMMAREA(COMMDSCT)
X
      LENGTH(=Y(COMMLEN))
      EJECT
PROCRETN DS 0H PROCESS RETURN FROM WINDOWS
*      :
*      (PROCESS DATA RETURNED FROM WINDOWS)
*      :
      MVI PASSWCRI,C'N' RESET WINDOWS RETURN SWITCH
*      :
*      (PROCEED WITH USER APPLICATION LOGIC)
*      :
* USUAL RETURN FOR A PSEUDO-CONVERSATIONAL PROGRAM
* EXEC CICS RETURN TRANSID(EIBTRNID) COMMAREA(COMMDSCT) X
* LENGTH(=Y(COMMLEN))
      EXEC CICS SEND FROM(UNLOCK) LENGTH(1) ERASE WAIT
      EXEC CICS RETURN
*      :
*      (THE REST OF THE PROGRAM)
*      :
      DFHEIRET
ABENDIT DS 0H ABEND DELIBERATELY
      EXEC CICS ABEND ABCODE(ABCODE)
USR1 DC CL4'USR1' ABEND CODE LITERAL
UNLOCK DC C' '
      END

```

## Programming Tips for Calling Finalist CICS LPCF

This list provides programming tips you may find useful when calling the Finalist CICS LPCF transaction:

- The exit COMMAREA image passed in the called Finalist CICS COMMAREA is placed in the same storage area as the exit routine COMMAREA. This means data such as record keys can be passed from the application that calls Finalist CICS to the exit routines. Also, any of the switches set by the exit routines to affect Finalist CICS processing can be set in the calling application.

- In many cases, you will not want to use the exit routines when calling Finalist CICS and passing it an address. Also, it is desirable to protect users from accidentally erasing the passed address. With PASSEX1L and PASSEX2L set to "L" and EXITREQW set to "Y", both the clear address PF10 key and the exit routine calls are locked. All changes made to a passed address must be made deliberately.
- There are separate switches for locking the input and output exits. These exits can be disabled separately allowing you to only read or only write from a called Windows dialog.

## System Error Codes

Abend Code	Abending Module	Description
LP00	LPWN500C	Transfer control attempted to program LPWN512C. The transfer control has failed. The program name might be missing from the CICS program Table (PPT). Check the installation again.
LP01	LPWN500C	GETMAIN failure - required storage not available.
LP02	LPWN500C	Link attempted to program LPWN502C, LPWN503C, LPWN504C, LPWN510C or FINALOL. The link has failed. The program name might be missing from the CICS Program Table (PPT). Check the installation again.
LP03	LPWN500C	The load has failed for LPWN501C (the Finalist messages table). Same probable reason and correction needed as described above for Abend Code LP02.
LP04	LPWN500C	Pseudo-conversational return failed - possibly because LPWN511C was LINKed to or XCTL'd to by a program that was LINKed to.
LP10	LPWN502C	The caller (LPWN500C) did not pass a COMMAREA on the CICS link command.
LP11	LPWN502C	The caller (LPWN500C) passed an invalid screen type indicator. Valid values are 2, 3, 4, and 5.
LP20	LPWN503C	The caller (LPWN500C) did not pass a COMMAREA on the CICS link command.
LP21	LPWN503C	The caller (LPWN500C) passed an invalid screen type indicator. Valid values are 2, 3, 4, and 5.
LP30	LPWN504C	The caller (LPWN500C) did not pass a COMMAREA on the CICS link command.

Abend Code	Abending Module	Description
LP32	LPWN504C	Unexpected, fatal error from a CICS command.
LP40	LPWN505C, LPWN508C	The caller (LPWN500C) did not pass a COMMAREA on the CICS link command.
LP41	LPWN505C, LPWN508C	Load failure for LPWN501C.
LP42	LPWN505C, LPWN508C	Link to LPWN507C failed.
LP50	LPWN506C	The caller (LPWN500C) did not pass a COMMAREA on the CICS link command.
LP60	LPWN511C	COMMAREA passed in XCTL to LPWN511C was too short to contain required passdata.
LP61	LPWN511C	Link to tailoring code translator failed.
LP62	LPWN511C	XCTL to LPWN500C failed.
LP70	LPWN512C	COMMAREA passed from Windows to LPWN512C was not the expected length.
LP71	LPWN512C	XCTL back to user program failed. Field PASSPGM is invalid.
LP80	LPWN507C	COMMAREA not passed to LPWN507C during link.
LP81	LPWN507C	Load failure on LPWN501C.
LP90	LPWN509C	Wrong length COMMAREA passed to input exit stub.
LPA0	LPWN510C	Wrong length COMMAREA passed to output exit stub.

## CICS Basic Mapping Support (BMS)

Basic mapping support (BMS) is an interface between CICS and application programs. Input and output data is formatted in response to BMS commands in the application programs. To do this, BMS uses device information from CICS system tables and formatting information from maps that have been prepared for the application program. A single BMS command in an application program can address more than one kind of device, since the terminal information is retrieved by BMS from

a systems table. Since formatting information is stored separately, writing applications is simpler and less susceptible to changes in the system or peripheral devices. To learn more about the use of BMS mapping in an application program, refer to your IBM CICS system manuals.

## Finalist CICS Sample Code

COBOL source for CICS transaction PBFN is included as a sample COBOL program. See PBFNFC00 in the CISOURCE library.

## Using Finalist IMS

This table provides information on using the S56LPCH and S56LPWNH transactions for Finalist IMS processing.

**Table 4: Finalist IMS Transactions**

Transaction	Description
S56LPCH	Simple address validation transaction to verify your Finalist IMS On-Line system installation using just the ZIP + 4 data.
S56LPWNH	Configurable address validation transaction using ZIP + 4 plus auxiliary Finalist databases. Includes database inquiry functionality.

## Using the S56LPCH Transaction

1. Enter the **S56LPCH** transaction name on any IMS terminal.
2. Press **Enter**. The Input Inquiry Screen displays.
3. Follow the information on the screen.

## Using the S56LPWNH Transaction

1. To access Finalist IMS, enter the transaction name **S56LPWNH** at any terminal signed on to IMS/VS.
2. Press **Enter**. The **Main Menu** displays.
3. Follow the information on the screen.

For more information on available options, see "Using the Compatibility Interface", in your Finalist Reference Guide.

### **Address Lookup Screen**

The Address Lookup screen displays when you select option code **1** (PF1) from the Main Menu. This is the primary input screen. Enter the data that defines the targeted information in the input fields and press Enter. The return data displays on the screen along with the selection data you entered in the input fields. You may enter new data or type over the existing data in the input fields. If an error window appears, you must press PF9 to clear the error window before you can enter new information.

### **Return and Reason Codes Screen**

1. From the **Address Lookup** screen, press **PF11**.
2. The **Return and Reason Codes** screen displays. This screen does not accept input.
3. Use **PF10** to toggle between the Return and Reason Codes screen and the Address Information Codes screen.
4. Press **PF24** to exit.

### **Address Information Codes Screen**

1. From the Reason Codes screen, press **PF10**.
2. The Address Information Codes screen displays. This screen does not accept input.
3. Press **PF10** to toggle between the Return and Reason Codes screen and the Address Information Codes screen.
4. To exit, press **PF24**.

For more information on available options, see "Using the Compatibility Interface", in your Finalist Reference Guide.

### **Return Area Screen**

1. From the Return and Reason Codes screen, press **PF11**.



- The Return Area screen displays. This screen does not accept input. Error information is not available on this screen.
- To return to the Return and Return Codes screen, press **PF11**.
- To exit from the Return Area screen and return to the Address Lookup screen, press **PF12**.

### City Information Screen

- From the **Address Lookup** screen, press **PF2**.
- The **City Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N
                                CITY NAME LIST
                                HH:MM:SS
INPUT CITY : XXXXXXXXXXXXXXXX INPUT STATE: XX (2 CHAR ABRV)

  CITY NAMES      ZIPCODES          CITY NAMES      ZIP CODES
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN
XXXXXXXXXXXXXXXX  NNNNN          XXXXXXXXXXXXXXX  NNNNN

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6:  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT

```

- Press **PF12** to exit.

### ZIP Code Information Screen

- From the **Address Lookup** screen, press **PF3**.

- The **ZIP Code Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N
                                ZIPCODE LIST
                                HH:MM:SS

STARTING ZIPCODE:  NNNNN

    ZIPCODE      CITY NAME      STATE  COUNTY  MOTHER  MULTIZIP  MULTICITY
                   CITY          ZIPCODE  CITY      ZIP

    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X
    NNNNN      XXXXXXXXXXXXXXXX  AA   NNNNN  NNNNN    X         X

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6:  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT
    
```

- Press **PF12** to exit.

### Street Name List

- From the **Address Lookup** screen, press **PF4**.
- The **Street Name Information Screen** screen displays. This screen does not accept input.

```

FINALIST ON-LINE N.N
                                STREET NAME LIST
                                HH:MM:SS

INPUT ZIPCODE:  NNNNN
INPUT CITY   :  XXXXXXXXXXXXXXXX  INPUT STATE:  XX
INPUT STREET :  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    STREET NAME                                STREET NAME

    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6 : N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT
    
```

- Press **PF12** to exit.

## Street Information Screen

```

FINALIST ON-LINE N.N

                                STREET RANGE LIST
                                HH:MM:SS

INPUT ZIPCODE:  NNNNN
INPUT CITY   :  XXXXXXXXXXXXXXXX  INPUT STATE:  XX
INPUT STREET :  XXXXXXXXXXXXXXXXXXXXXXXX

    LOW RANGE      HIGH RANGE  PRE DIR  POST DIR  ALS  SEC/SEG  CAR  RTE  ZIP
    RANGE          RANGE      DIR     DIR     SUFFIX IND  EVEN  ODD  EVEN  ODD

XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN
XXXXXXXXXX XXXXXXXXXXXX AA  AA  AA  A  NNNN NNNN XXXXX XXXXX NNNNN

*****ERROR MESSAGE*****
PF KEYS=====
PF1:  N/A   PF2:  N/A   PF3:  N/A   PF4:  N/A   PF5:  N/A   PF6 :  N/A
PF7:  UP    PF8:  DOWN  PF9:  N/A   PF10: N/A   PF11: N/A   PF12: EXIT

```

## Product Information Screen

1. From the **Address Lookup** screen, press **PF6**.
2. The **Product Information Screen** screen displays. This screen does not accept input.
3. Press **PF12** to exit.

## S56LPCH Extended Capabilities

Finalist IMS does not have I/O (Input/Output) capabilities of its own other than its use of the Finalist master files. To facilitate the update of addresses on a file, there are exit points in the system which can call I/O routines. Call one exit point to read an address into the Windows address lookup screen. Use the other exit point to write addresses from the Windows address lookup. You can call the input exit by pressing the PF7 key from the address lookup display. You can call the output exit by pressing the PF8 key.

## Using the Exit Programs

The programs exist as small stubs. Each stub can be replaced in the load library by a user-written program of the same name. The Finalist IMS system calls the programs through the PF7 and PF8 keys:

- LPWNEX1 program for the input exit

- LPWNEX2 program for the output exit

Both programs pass a call area of the same format. The source member LPCFEXT, on the On-Line Windows distribution tape, defines the call area passed to the exit routines. LPWNEX1 contains a copy statement for LPCFEXT which is a source member on the distribution tape. The table below describes the WNDWS-EXIT-PASS-AREA fields.

**Table 5: WNDWS-EXIT-PASS-AREA Fields**

Field	Description
FILLER	Fourteen-character reserved area used by On-Line Windows
WNDWS-SPA-EXIT-RC	Two-character return code set by the user to indicate the success or failure of the function of the user's exit.
WNDWS-SPA-EXIT-MESSAGE	A 40-character message filled by the user exit that displays upon completion of the exit.
FILLER	A 256-character reserved area used by the windows transaction.
WNDWS-FINAL-CALL-AREA	The Finalist call area.
FILLER	Reserved for use by the WINDOWS transaction.

### ***Input Exit Point***

This section describes input exit point processing including:

- Data passed to the exit routine on a call
- How the exit routine data is processed when returned to the Finalist IMS system

The steps for a call to an input exit point are:

1. Press the PF7 key.
2. The Finalist IMS system calls to the exit program, LPWNEX1, and passes the WNDWS-EXIT-PASS-AREA.
3. The LPWNEX1 program processing starts.
4. The LPWNEX1 program returns control to the Windows programs.
5. The WNDWS-EXIT-PASS-AREA is formatted onto the address lookup screen.
6. The Finalist function and tailoring options (set on the main menu) are restored. The function and tailoring options are the same as those in the WNDWS-EXIT-PASS-AREA.
7. The Address Lookup Screen displays the information passed to from the exit.

## **Output Exit Point**

This section describes output exit processing including:

- Data passed to the exit routine on a call
- Processing after the exit routine returns control to the Finalist IMS program

The steps for a call to an output exit point are:

1. You must process the desired address on the address lookup screen before actually executing the exit program. This includes a coding attempt through Finalist. A coding attempt is defined as an address displays on the address lookup screen, and the Enter key is pressed or the input exit routine returns an address and a return code of "00". The coding attempt builds the Finalist CALLAREA. This CALLAREA is part of the SPAAREA passed to the exit routine.
2. Press the PF8 key to execute the output exit routine.
3. The Finalist IMS system calls to the LPWNEX2 exit program and passes the WNDWS-EXIT-PASS-AREA described earlier.
4. The LPWNEX2 program returns control to the On-Line Windows programs.
5. A message displays the exit return code on the address lookup screen.
6. The Address Lookup Screen displays with the original address and ADDR key data field left intact.

## Testing Methods

You may choose to test Finalist processing in batch, on-line, and/or windows mode. This appendix provides information for two testing methods.

- Batch Message Processing (BMP)
- Batch Terminal Simulation (BTS)

### ***Batch Message Processing (BMP)***

You may test Finalist processing through BMP. You do not need to modify the S56LPWNH PSBs as the parameter setting CMPAT=YES provides them.

### ***Batch Terminal Simulation (BTS)***

You may test Finalist processing through BTS. The following are sample BTS control cards.

```
./D TYPE=3270-A2 SIZE=(24,80) LIMIT=1
./T TC=S56LPWNH MBR=S56LPWNH LANG=CBL PLC=20 SPA=3500
./D LTERM=IOPCB EOM=$ DDOF=3270-2 TPBUF=2275 FORMBUF=22510
S56LPWN  ENTER $
ENTER $
L05C13 '1200 RO RD #200'
L07C13 'GLEN ELLYN IL'  ENTER $
/EXIT
```

# 4 - Defining a Job

## In this section

---

Building Batch Jobs.....	88
Building Batch Jobs with the Workbench.....	122



# Building Batch Jobs

This section provides information on creating the Job File (.job) and Definition File (.def) that are required to run the Finalist batch driver.

## Batch Driver Features

The batch driver program includes the following features.

- Interfaces with the Finalist CASS-certified APIs. This eliminates the need for a custom-written application. You only need to decide how to process the returned address information.
- Interfaces with the AddrScan to handle a wide range of block-type input file data formats.
- On Windows and Unix/Linux, accepts as input straight ASCII text or ASCII text delimited by characters (for example, a tab or comma).
- Can dynamically update your original address file during processing.
- Can optionally create output files containing the addresses that coded successfully, the addresses that did not code successfully, or both. These output files provide you with information on the success or failure of the postal coding process on your addresses.
- Generates the USPS Form 3553 (CASS Summary Report) and the Finalist Batch Report when you run in a CASS-certified mode.
- Processes an entire file or a range of records within a file. This can be especially helpful when you are performing testing on a very large data file.

## Setting Up Batch Driver Processing

To set up batch driver processing, follow these steps to identify the format of the data to process.

1. If a file contains binary data, you must specify an input record size (cRecSz) and an output record size (cRecSzOut).

**Note:** ASCII files with binary data can only be processed if they are processed with a file type of ASCIIFIXED. In this case, all records must be the same length (not variable defined by CRLF). If an ASCII file contains a CR or LF or both, but is processed as ASCIIFIXED, the input and output record sizes must include the CR or LF fields in the length specification.

2. If each record terminates with a carriage return/line feed (CR/LF) pair, set the Job file setting cInputFileType to ASCIICRLF.
3. If each record contains a fixed number of bytes (no CR/LF at the end of each record):



- a) For z/OS, use the Job file setting `clnputFileType=MFFIXED`.
  - b) For all other platforms, use the Job file setting `clnputFileType=ASCIIFIXED`.
  - c) Set the Job file keyword `cRecsz` to be the number of bytes that each address record occupies.
4. Are the fields within each record delimited (separated) by a specific character (for example, a tab)?
- a) Use the Job file setting `clnputFileType=DELIMITED`.
  - b) Use the Job file setting for `cDELIMITER` based on your delimiter character as shown next.

`cDelimiter=TAB`

`cDelimiter=COMMA`

`cDelimiter=X` (where X is any single character value)

`cDelimiter=XX` (where XX is the hexadecimal value of the character)

`cDelimiter = 0xXX` (where XX is the hexadecimal value of the character)

`cDelimiter = 'XX'` (where XX is the hexadecimal value of the character)

Refer to the sample files named `DELIMITED.*` for an example of a data file where the fields are delimited by the tab character. Note that for delimited data files, overlaying the original data in place is not allowed, so you must specify one or more output files in the Job file via the `fOutAllFile`, `fOutValidFile`, and `fOutErrorsFile` keywords in order to obtain the results of postal coding your data.

#### EXAMPLE

The Windows sample data file `DELIMITD.TXT` is delimited by the tab character. The sample Job file `DELIMITD.JOB` uses the setting `DELIMITER=09` (09 is the hexadecimal value for the ASCII tab character).

5. Are the records in a dbase file? If yes, use the Job file setting `clnputFileType=DBASE`
6. Specify the post processing options for the Finalist return data.
  - a) To indicate whether to overlay the input record with the Finalist return data, specify the appropriate `OVERLAY` Job file setting option (`jOverlay=YES` or `jOverlay=NO`).  
 You will receive an error message and your job will fail if you try to use the `jOverlay` option on the z/OS platform. The `jOverlay` option is not supported on the z/OS platform.
  - b) To indicate whether to add the Finalist return data to the input record, specify the appropriate `ATTACH` Job file setting option (`jAttach=YES` or `jAttach=NO`).

## Batch Driver Input

Before processing an address file, you must provide the Finalist driver with the following information:

- Layout of each record in your input file.

- Processing to perform.
- Location of your input and output files.

The driver looks for this information in the Finalist Job File (.job) and Definition File (.def).

## Job File

The Job File contains keyword entries specifying the path and file names to use for the original data file and the output file(s). The Job File also includes information on the processing options to be performed, as well as a keyword that specifies where the Definition File is located. For the z/OS platforms, the job file is specified as PBFNJOB. For other platforms, the file ends with the .job extension.

## Definition File

The Definition File contains keyword entries specifying the location and length of the input fields to be passed to the postal coding engine by the driver. The driver determines the location of the Definition File using a keyword in the Job File. For z/OS platforms, the default definition file is specified as PBFNDEF. For other platforms, the file ends with the .def extension.

## How Do I Create and Edit Job and Definition Files?

The following methods are available for creating Job and Definition Files.

<b>Finalist Workbench</b>	Windows users can use the Workbench to create and edit Job and Definition Files. For more information on the Finalist Workbench, refer to <a href="#">Building Batch Jobs with the Workbench</a> on page 122.
<b>Standard Text Editor</b>	Advanced users can use a standard text editor to create and edit the Job and Definition Files.

## Creating and Editing Job and Definition Files

Before building the Job and Definition Files:

1. Assess the format and level of detail in your address file. You must know the format of your address file in order to create or edit a Definition File. You will need to identify the position and length of each address line or component that is used as input to the driver. For information on address component fields, refer to "Using the Structures and Constants" in your Finalist Reference Guide. Refer to [Definition File](#) on page 103 for a description of keywords that can be used within the Definition File.
2. Use the information in "Using the Structures and Constants" in your Finalist Reference Guide as a checklist. Write down the position and length of address fields on the input record.

Ideally, your data file will contain all of the fields listed in "Using the Structures and Constants" in your Finalist Reference Guide. To optimize the benefits of OVERLAY processing, it is important to have as many address components available as possible. Detailed address information enables

the driver to process and update address fields in the file and eliminates the need for post-processing of the output. Note that the driver can only update the fields that are defined in the Definition File when using the jOverlay option. If your file does not contain the necessary level of detail, you can overlay the components that you have identified in the Definition File, and direct returned address data to a separate file.

If your input file does not contain detailed address components at specific offsets and instead has blocks of address data (for example, city, state and ZIP Code at a specific offset), you can still set up a Definition File to handle the data using block-style keywords in the Definition File.

Consider reformatting your input file to attain a higher level of detail. You can achieve this by using block-style addressing for a single execution of the driver, building new, more detailed data records using the output from the previous run, and then running the driver a second time on the more detailed data. After reformatting the data, a new Job and Definition file should be created to use the new fields. You can then use overlay processing option to update the data in place.

**Note:** It is strongly recommended that you create a backup copy of your data before performing OVERLAY processing on your file for the first time.

## Job File

The Job File contains the two sections of information described next.

<b>{FILES} section</b>	Contains complete path names for files associated with the job (either used for input or created as output)
<b>{OPTIONS} section</b>	Contains processing options

### **{FILES} Section**

The {FILES} section identifies each file to be used during processing. For z/OS platforms, provide the DD name for each file to be used during processing. For all other platforms, provide the complete path for each file to be used during processing.

#### **Figure 8: Keywords**

The following table describes the required and optional keywords that can be defined in the {FILES} section. In the table descriptions, note that (file spec) should be replaced with the complete path of the appropriate files without the parenthesis (for example, C:\Precisely\MYDATA.TXT).

**Table 6: {FILES} Section Keywords**

Keyword	Required/ Optional	Description
fInput=(file spec)	Required	Defines the input address file to be processed by the driver program.  The maximum allowed record size for input data is 32,767 bytes (32K).
fDefinition=(file spec)	Required	Defines the Definition File that describes the layout of address information for the file appearing in the fInput keyword.
fOutAllFile=(file spec)	Optional	Defines the file to write every record that is processed, regardless of whether or not the record was processed successfully. To see the exact layout of the output record, refer to the section <a href="#">Contents of jAttach Record</a> on page 119.
fOutValidFile=(file spec)	Optional	Defines the file to write every record that is coded successfully. The layout of each record in this file is identical to the layout of the records written to the fOutAllFile file.
fOutErrorsFile=(file spec)	Optional	Defines the file to write every record that is NOT coded successfully. Like the fOutValidFile file, the layout of each record in this file is identical to the layout of the records written to the fOutAllFile file.
f3553FileName=(file spec)	Optional	Defines the USPS Form 3553 (CASS Summary Report) file name. The value specified overrides the value specified in the configuration file.
fReport=(file spec)	Optional	Defines the location for the driver program to write the Finalist Batch Report. If omitted, the report file path, "Batch Report Filename = ", contained in the pbfm.cfg will be used. We recommend that you specify a unique report file name for each job file using this keyword in order to associate a particular report file with a specific batch run.
fDetailReport=(file spec)	Optional	Defines the location for the driver program to write the Address Detail Report. If omitted, the report file path, "Report Filename = ", contained in the pbfm.cfg will be used. We recommend that you specify a unique report file name for each job file using this keyword in order to associate a particular report file with a specific batch run.

Keyword	Required/ Optional	Description
fpbfnLog=(file spec)	Optional	Defines the location for the driver program to write the Finalist engine log messages. If omitted, the report file path, "Log Filename = ", contained in the pbfm.cfg will be used. We recommend that you specify a unique report file name for each job file using this keyword in order to associate a particular report file with a specific batch run.
fConfigFile=(file spec)	Optional	Defines the configuration file to use. Note that certain keywords in the [OPTIONS] section override values in the pbfm.cfg. If you want to define individual configuration files for each Job File that you create, you can use the fConfigFile keyword to point to a specific, user-modified copy. Note that the configuration file(s) you create and use can be named something other than "pbfm.cfg". For more information on the pbfm.cfg file, refer to <a href="#">Configuring Finalist</a> on page 13.

### **{OPTIONS} Section**

The {OPTIONS} section contains keywords that are used to indicate the actions required during processing. Note that in the Keyword column below, a vertical bar (|) is used to separate possible value for each keyword.

#### **Figure 9: Keywords**

The following table describes keywords that can be defined in the {OPTIONS} section.

Table 7: {OPTIONS} Section Keywords

Keyword	Description
cAddrScan=YES   NO   AddrScan Return Line Option	<p>Indicates whether to perform AddrScan processing. For more information on AddrScan processing, refer to <a href="#">AddrScan Option</a> on page 188. This keyword is generally only valid when block-style address keywords are specified in the Definition file (iL1, iL2, iL3, etc.). This keyword is generally only valid when block-style address keywords are specified in the Definition file (iL1, iL2, iL3, etc.). However:</p> <p>If you specify iFirm, AddrScan processes with the SkipFirm option.</p> <p>If you specify iCity, AddrScan processes with the SkipCSZ option.</p> <p>If you specify iURB, AddrScan processes with the SkipURB option.</p> <p>Valid values are:</p> <p><b>YES</b> Perform AddrScan processing. The value YES implies the default option of 95N for returned line setting.</p> <p><b>NO</b> Do not perform AddrScan processing.</p> <p><b>Three-character returned line setting</b> Use the three character AddrScan returned line setting described in <a href="#">Setting Returned Line Options</a> on page 198.</p> <p><b>Blank</b> Defaults to 95N.</p>
cCase=U   L   M	<p>Indicates whether to return addresses in all upper, all lower, or mixed upper and lower case.</p> <p><b>U</b> Return addresses in all upper case.</p> <p><b>L</b> Return addresses in all lower case.</p> <p><b>M</b> Return addresses in upper and lower case.</p>
cCassMode=YES   NO	<p>Indicates whether you want to process in a CASS-certified mode.</p> <p><b>YES</b> Generate the Batch Processing Report and the USPS Form 3553 (CASS Summary Report).</p> <p><b>NO</b> Only generate the Batch Processing Report.</p>
cOverlayCheck=ON   OFF	<p>Indicates whether you want to prevent overlapping field specifications.</p> <p><b>ON</b> Prevent output fields from overlaying input fields unless the output and input field formats are identical. The Finalist driver ensures that output fields do not overlay input fields.</p> <p><b>OFF</b> Create an output file with a format that is not identical to the format of the input file. If set to NO, the output file is initially blank and then only populated by output fields specified in the corresponding job definition (DEF) file.</p> <ul style="list-style-type: none"> <li>blank — Defaults to ON.</li> </ul>

Keyword	Description
cStartAft=(start after value)	Indicates how many records in the input file to bypass before beginning address processing. The driver will skip the number of records specified here prior to processing. Valid values are 0 through 99,999,999. The default value is <b>ZERO</b> .
cStopAft=(stop after value)	Indicates the number of the last record to process. Valid values are 1 through 99,999,999 and EOF. If you specify EOF, the driver will process through the last record in the file. The default value is <b>EOF</b> .
cVerboseCount=[count]	<p>Indicates the number of records to process before generating progress status messages. For example, if you specify "100", every 100th record, starting with 1, produces the message:</p> <p>Processing record XXX ...DDD MMM DD HH:MM:SS YYYY</p> <p>For Windows, the message overlays each preceding instance as processing progresses. If you specify "0", no messages are produced. Windows, Linux, and Unix systems can also prevent the message with the command line option -q. The default value for Windows, Linux, and Unix systems is <b>1</b>. The default value on the mainframe is <b>0</b>.</p>
jAddrScanAO=(AddrScan addressing options)	<p>Defines the AddrScan addressing options:</p> <ul style="list-style-type: none"> <li><b>#</b> To prevent AddrScan from concatenating lines beginning with # to existing address lines, enter # in the AddrScanAO field. This field is only available when AddrScan processing is selected (see previous step).</li> <li><b>P</b> To prevent AddrScan from recognizing periods as valid characters and to specify that periods should be removed before scanning address lines, enter a P in the AddrScanAO field. This field is only available when AddrScan processing is selected (see previous step).</li> </ul> <p>The values # and P can populate either position for the jAddrScanAO keyword definition.</p>
jAttach=YES   NO	<p>Defines whether or not the processed and corrected data will be attached to your original data. Valid values are described next.</p> <ul style="list-style-type: none"> <li><b>YES</b> Attach corrected data to my original data. The results of the address lookup will be attached to the address fields in each input record that you process.</li> <li><b>NO</b> Do not attach corrected data to my original data. In order to receive returned address information, you must define the jOverlay keyword to overlay your original input record with the corrected results or specify a new output file using fOutAllFile, fOutValidFile, or fOutErrorsFile in the {FILES} section of the Job File. For a description of the output record, refer to <a href="#">Contents of jAttach Record</a> on page 119.</li> </ul>

Keyword	Description
jOverlay=YES   NO	<p>Defines whether or not your original data will be overlaid with the corrected results. This selection updates a file in place.</p> <p>You will receive an error message and your job will fail if you try to use the jOverlay option on the z/OS platform. The jOverlay option is not supported on the z/OS platform.</p> <p><b>YES</b> Overlay original data with corrected results. Address fields in each input record that you process will be overlaid with the results of the address lookup. Note that only fields that have been defined via keyword entries will be overlaid. For example, if the ZIP Code field is not specified by defining an iZip keyword and an oZip=o keyword within the Definition File, the ZIP Code field will not be overlaid.</p> <p><b>NO</b> Do not overlay original data with corrected results. In order to receive returned address information, you must define the jAttach keyword to attach results to your original input record or specify a new output file using fOutAllFile, fOutValidFile, or fOutErrorsFile in the {FILES} section of the Job File. You can overlay specific input fields with the processing output on a field level basis by specifying jOverlay=NO and defining an input keyword and output keyword. For example, you do not want to overlay your entire input record with the processing output but only want to overlay the input ZIP Code field with the output ZIP Code resulting from Finalist processing. To accomplish this you would specify jOverlay=NO and define an iZip keyword and an oZip=o keyword. The input ZIP Code field will be overlaid with the output ZIP Code. No other input fields will be overlaid with output information.</p> <p><b>EXAMPLE</b></p> <p>Your input data contains ZIP Code information in the fields you defined as iCity or iState. The driver uses the ZIP Code information to process the address. The driver overlays the city information at the location indicated by the iCity keyword and the state information at the location indicated by the iState keyword. However, since you did not define an iZIP field, the driver has no location for overlaying the ZIP Code information. The returned ZIP Code information is lost. We recommend you use jOverlay=NO if your input file does not contain each address component at a consistent starting position within each record. This will ensure that the driver will return complete address information.</p>
rListProcessorName=(list processor name)	<p>If you want to enter a list processor's name, use this keyword to enter a list processor's name to appear on the USPS Form 3553 (CASS Summary Report).</p>
rMailerAddress=(mailer address)	<p>If you want to CASS certify under your own company name, use this keyword to enter the street address to appear on the USPS Form 3553 (CASS Summary Report).</p>
rMailerAddress2=(mailer address)	<p>If you want to CASS certify under your own company name, use this keyword to enter additional street address information to appear on the USPS Form 3553 (CASS Summary Report).</p>



Keyword	Description
rMailerAddress3=(mailer address)	If you want to CASS certify under your own company name, use this keyword to enter additional street address information to appear on the USPS Form 3553 (CASS Summary Report).
rMailerAddress4=(mailer address)	If you want to CASS certify under your own company name, use this keyword to enter additional street address information to appear on the USPS Form 3553 (CASS Summary Report).
rMailerCity=(mailer city)	If you want to CASS certify under your own company name, use this keyword to enter the mailer city, state, and ZIP Code to appear on the USPS Form 3553 (CASS Summary Report).
rMailerName=(mailer name)	If you want to CASS certify under your own company name, use this keyword to enter the mailer name to appear on the USPS Form 3553 (CASS Summary Report).
cRecsz=(record size value)	Allows you to indicate the size (in bytes) of each input record within your input file if each record within your input file is not followed by a carriage return/line feed (CR/LF) pair. This field defaults to a value of CR/LF, indicating that each record is followed by a carriage return/line feed pair. If your records do not end with a CR/LF, you must specify a number in the range of 1 to 32,767 inclusive. If you put an incorrect value here, it will result in the improper reading of input records.
cRecszOut=(record size value)	Allows you to indicate the size (in bytes) of each output record. This keyword is only valid for fixed length records.

Keyword	Description
cInputFileType=ASCIICRLF   ASCIIFIXED   DELIMITED   DBASE   MFFIXED	<p>This keyword is required, and indicates what type of input data you have.</p> <p><b>ASCIICRLF</b> Each record in your data file is terminated with a carriage return/line feed (CR/LF) pair. The default value is ASCIICRLF if cRecsz is zero.</p> <p><b>ASCIIFIXED</b> Each record in your data file is a fixed number of bytes in length, and is not terminated by a CR/LF pair. The default value is ASCIIFIXED if cRecsz is non-zero. Set the Job file keyword cRecsz to the number of bytes that each record occupies.</p> <p><b>DELIMITED</b> The fields within each record are delimited by a specific character (for example, comma, etc.), and each record is terminated with a CR/LF pair. When using this option, you are required to specify what the delimiting character is via the DELIMITER keyword.</p> <p><b>DBASE</b> The driver program supports dBASE files. dBASE is a popular PC-based file format. This value indicates your input file is in a dBASE format. NOTE: dBase does not support attached fields.</p> <p><b>MFFIXED</b> Each record in your data file is a fixed number of bytes in length, and is not terminated by a CR/LF pair (mainframe fixed input file). Set the Job file keyword cRecsz to the number of bytes that each record occupies.</p>
cDelimiter=COMMA   TAB   X   xx (where xx is the hexadecimal value of the character)	<p>Indicates the delimiting character when the fields within each of your data records are delimited by a specific character. There is no default value for this keyword. Valid values are described next.</p> <p><b>COMMA</b> The delimiter is a comma (.).</p> <p><b>TAB</b> The delimiter is the tab character.</p> <p><b>X</b> The delimiter is any single character.</p> <p><b>xx</b> The delimiter is any character whose hexadecimal value can be represented using two hexadecimal digits (in the range of 00 to FF).</p> <p><b>0xXX</b> The delimiter is any character whose hexadecimal value can be represented using two hexadecimal digits (in the range of 00 to FF).</p> <p><b>x'XX</b> The delimiter is any character whose hexadecimal value can be represented using two hexadecimal digits (in the range of 00 to FF).</p>

Keyword	Description
cXTern=YES   NO	<p>Defines whether to attach 250 bytes of additional information codes to your output file. This parameter was designed for use with the Precisely VeriMove move update product. For more information, refer to <a href="#">Precisely VeriMove Move Update Product Information Codes</a> on page 99.</p> <p><b>YES</b> Return the additional information codes to the output files. To attach the 250 bytes of additional information codes to your output file to use as input to the Precisely VeriMove move update software, specify "YES".</p> <p><b>NO</b> Do not return the additional information codes to the output files. In order to receive returned address information, you must specify a new output file using fOutAllFile, fOutValidFile, or fOutErrorsFile in the [FILES] section of the Job File. For a description of the output record, refer to <a href="#">Contents of jAttach Record</a> on page 119.</p>

### Precisely VeriMove Move Update Product Information Codes

The table below describes the information codes for the VeriMove move update product External CASS processing option.

**Table 8: VeriMove Move Update Product Information Codes**

Information Code	Length (bytes)	Description
ZIP + 4 Software Name	30	CASS-certified software product name.
ZIP + 4 Software Version	12	CASS-certified software product version number.
ZIP + 4 Database Date	8	CASS database date in a CCYYMMDD format.
Address Coded Flag	1	ZIP + 4 match status.
		<p><b>Blank</b> No ZIP + 4 match for address.</p> <p><b>4</b> Address did ZIP + 4 match.</p>

Information Code	Length (bytes)	Description
Address Match Level Indicator	1	Level of matching achieved for an address. <b>Blank</b> No ZIP + 4 match found for address. <b>1</b> Street level match. <b>2</b> Rural route default match. <b>3</b> Highway contract default match. <b>4</b> Rural route with secondary match. <b>5</b> Highway contract default match. <b>6</b> Firm match. <b>7</b> General delivery match. <b>8</b> Highrise default match. <b>9</b> Highrise with secondary match. <b>A</b> PO Box match. <b>B</b> Nondeliverable match. <b>C-Y</b> Reserved for future use. <b>Z</b> Unknown match.
ZIP Type	1	ZIP Code type. <b>Blank</b> No information available. <b>M</b> Military ZIP Code.
LACS Indicator	1	LACS processing indicator. <b>Blank</b> LACS processing not available <b>L</b> LACS indicator.
Early Warning System (EWS) Flag	1	EWS status. <b>Blank</b> No ZIP + 4 match found for address. <b>Y</b> EWS match caused address failure. <b>N</b> Did not match an EWS record.
CMRA Present Flag	1	CMRA status. <b>Blank</b> No ZIP + 4 match found for address. <b>Y</b> CMRA information present on input. <b>N</b> CMRA information not present on input.

Information Code	Length (bytes)	Description
DPV Software Name	30	DPV software product name.
DPV Software Version	12	DPV software product version number.
DPV Database Date	8	DPV database date in a CCYYMMDD format.
DPV Footnotes	24	DPV footnotes.
DPV Delivery Indicator	1	DPV status. <b>Blank</b> No DPV results. <b>Y</b> Match primary and secondary. <b>N</b> No DPV match. <b>D</b> Primary match but missing secondary. <b>S</b> Primary match but no match on secondary.
DPV CMRA Indicator	1	DPV CMRA status. <b>Blank</b> No DPV results. <b>Y</b> Match to CMRA. <b>N</b> No match to CMRA.
LACS <sup>Link</sup> Software Name	30	LACS <sup>Link</sup> software product name.
LACS <sup>Link</sup> Software Version	12	LACS <sup>Link</sup> software product version number.
LACS <sup>Link</sup> Database Date	8	LACS <sup>Link</sup> database date in a CCYYMMDD format.

Information Code	Length (bytes)	Description
LACS <sup>Link</sup> Return Code	2	LACS <sup>Link</sup> processing return code. <b>A</b> Record matched through LACS <sup>Link</sup> processing. <b>00</b> No matching record found during LACS <sup>Link</sup> processing. <b>09</b> LACS <sup>Link</sup> processing matched the input address to an older highrise default address. The address has been converted. However, rather than provide an imprecise address, LACS <sup>Link</sup> processing does not provide a new address. <b>14</b> Match found during LACS <sup>Link</sup> processing but conversion did not occur due to other USPS regulations. <b>92</b> Record matched through LACS <sup>Link</sup> processing. Unit number dropped on input.
Suite <sup>Link</sup> Software Name	30	Suite <sup>Link</sup> software product name.
Suite <sup>Link</sup> Software Version	12	Suite <sup>Link</sup> software product version number.
Suite <sup>Link</sup> Database Date	8	Suite <sup>Link</sup> database date in a CCYYMMDD format.
Suite <sup>Link</sup> Footnote	2	This is the Suite <sup>Link</sup> footnote code encountered while CASS processing the input record. <b>A1</b> Exact Match - All words in the business name matched. <b>A2</b> Good Match - All but one word in the business name matched. <b>A3</b> Poor Match - At least one word in the business name matched. <b>B</b> No match. <b>C</b> The business name consisted only of "noise" words. <b>D</b> The ZIP + 4 provided was not a high rise default. <b>E</b> The database was expired.
DPV ZIP4	4	ZIP + 4 used by DPV. <b>Blank</b> No ZIP + 4 sent to DPV.

Information Code	Length (bytes)	Description
DPV Vacant Flag	1	DPV Vacant status.
		<b>Y</b> Match to Vacant Table.
		<b>N</b> No match to Vacant Table.
		<b>Blank</b> The DPV Vacant Table was not queried.
DPV No-Stat Flag	1	DPV No-Stat status.
		<b>Y</b> Match to No-Stat Table.
		<b>N</b> No match to No-Stat Table.
		<b>Blank</b> The DPV No-Stat Table was not queried.
DPV PBSA Flag	1	DPV PBSA status.
		<b>Y</b> Match to PBSA Table.
		<b>N</b> No match to PBSA Table.
		<b>Blank</b> The DPV PBSA Table was not queried.
DPV DNA Flag	1	DPV Door Not Accessible (DNA) Table status.
		<b>Y</b> Match to DPV DNA Table.
		<b>N</b> No match to DPV DNA Table.
		<b>Blank</b> The DPV DNA Table was not queried.
Reserved	6	Reserved for future use.

## Definition File

The Definition File contains keyword entries specifying the location and length of the input fields to be passed to the postal coding engine by the driver. The driver determines the location of the Definition File using the fDefinition keyword in the Job File. Finalist provides the following Definition File formats, and you can use these formats to accommodate almost any record layout (for example, location of address data within the record).

- **Format One: Block-Style Address Line Keyword Definitions** on page 104
- **Format Two: Address Component Keyword Definitions** on page 107

## Format One: Block-Style Address Line Keyword Definitions

Block style address lines are used when your data file does not contain address information at a fine level of detail.

### EXAMPLE

Your file does not contain a ZIP Code that starts at a specific offset in each record. Your file contains a ZIP Code that floats in an area containing city, state, and ZIP Code information. You can use block-style keywords to specify that city, state, and/or ZIP Code information is located within a specific area of the record using block-style addresses.

Without use of the AddrScan, the Finalist driver can process three lines of block-style addresses. You can incorporate AddrScan to enable the Finalist driver to process up to six lines of block-style address information prior to calling the postal coding engine for validation and correction. The Finalist driver makes the following assumptions about the contents of block-style address lines prior calling the Finalist API.

**Line 1 (iL1)** Contains street address information, including primary and secondary street range, predirectional, post-directional, and street name. It may also contain unit designator information.

**Line 2 (iL2)** Contains unit designator (apt./suite) information, secondary address information, or is blank.

**Line 3 (iL3)** Contains city, state, and/or ZIP Code information.

The Definition File should be set up to extract the address information from the proper position within the input record. When defining block-style addresses, you may specify up to ten individual positions and lengths for each line that is defined in the Definition File.

If you cannot extract address information as stated in the previous paragraph, you can use AddrScan. AddrScan makes it possible for the driver to extract up to six lines of address information from the input record. These lines are passed to AddrScan and assembled into three lines of address information conforming to the assumptions listed above.

When AddrScan is used, the input definitions of the block-style address lines are described next.

- Line 1 (iL1) — Unknown. Requires AddrScan.
- Line 2 (iL2) — Unknown. Requires AddrScan.
- Line 3 (iL3) — Unknown. Requires AddrScan.
- Line 4 (iL4) — Unknown. Requires AddrScan.
- Line 5 (iL5) — Unknown. Requires AddrScan.
- Line 6 (iL6) — Unknown. Requires AddrScan.

The following processing steps apply when a Definition File contains definitions for any of iL4, iL5, or iL6.

1. The driver calls AddrScan to examine the lines of data.



2. AddrScan determines which lines best fit the definitions for iL1, iL2, iL3 and iL4, and identifies values for the ZIP Code and ZIP + 4 fields.
3. AddrScan returns these isolated address components to the driver.
4. The driver moves line 1 to the iAddress1 field, line 2 to the iAddress2 field, line 3 to the iCity field, line 4 to the iFirm field, the identified ZIP Code data to the iZip field, and the identified ZIP + 4 data to the iZip4 field.
5. If AddrScan is requested and the DEF file specifies iZIP or iZIP4 fields, the following occurs:
  - If AddrScan identifies a ZIP or ZIP4 inside the data fields (iL1-iL6), AddrScan passes the identified ZIP or ZIP4 fields to Finalist.
  - If AddrScan is not able to identify a ZIP or ZIP4 field, AddrScan passes the information from the iZIP or iZIP4 fields to Finalist.

The following table shows a sample Definition File using block-style address keywords. Note that you can specify up to 10 positions and lengths to extract information from multiple locations within the data record for each of the lines. This feature accommodates a wide range of record layouts.

**Table 9: Definition File Layout Using Block Style Address Keywords**

Keyword	Description	Recommended Length
;	Comment lines are indicated by a semicolon (;) in column one.	Semicolon must begin in column one for comment lines.
; Finalist Definition File.	Comment. Enter a description of the file.	Fewer than 80 bytes
; Created by:	Comment. Enter your name and date created.	Fewer than 80 bytes
; Input file=Inputfile.txt	Comment. Enter the name of the input file being defined.	Fewer than 80 bytes
iL1=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Required.</b> Identifies positions (x) and lengths (y) of data to be treated as block address input line one. Note that up to 10 starting positions and lengths may be specified.	70 bytes
iL2=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Optional:</b> . Identifies positions (x) and lengths (y) of data to be treated as block address input line two. Note that up to 10 starting positions and lengths may be specified.	70 bytes

Keyword	Description	Recommended Length
iL3=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Optional:</b> . Identifies positions (x) and lengths (y) of data to be treated as block address input line three. Required if IZIP is not specified. Note that multiple starting positions and lengths may be specified.	70 bytes
iL4=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Optional:</b> . Identifies positions (x) and lengths (y) of data to be treated as block address input line four. Note that up to 10 starting positions and lengths may be specified.	70 bytes
iL5=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Optional:</b> Identifies positions (x) and lengths (y) of data to be treated as block address input line five. Note that up to 10 starting positions and lengths may be specified.	70 bytes
iL6=x,y[,x,y,x,y,x,y,x,y,x,y,x,y,x,y,x,y]	<b>Optional:</b> Identifies positions (x) and lengths (y) of data to be treated as block address input line six. Note that up to 10 starting positions and lengths may be specified.	70 bytes
iCrrt=x,y	<b>Optional:</b> Identifies the position (x) and length (y) of the input carrier route.	4 bytes
iFirm=x,y	<b>Optional:</b> Identifies the position (x) and length (y) of the input high-rise name or a firm name.	40 bytes
iZip=x,y	<b>Optional:</b> Identifies the position (x) and length (y) of the input ZIP Code field. <b>Required</b> if iL3 is not specified.	5 bytes
iZip4=x,y	<b>Optional:</b> Identifies the position (x) and length (y) of the input ZIP + 4 (sector segment) field.	4 bytes

## Format Two: Address Component Keyword Definitions

If the address information in your data records contains a high level of detail, format two is the preferred method of processing. Having component level address data gives you the option of directly replacing data within the input record with corrected address data via the jOverlay option. This eliminates the need for customized post-processing applications to analyze the output file(s) in order to determine how the data file should be updated.

**Note:** When passing information to the batch driver, you must provide enough information for Finalist to code addresses. This information can consist of a single combined address line or a parsed range and street name (additional components may also be necessary). Last line requirements are a combined city state line, or the City and State, or the ZIP Code.

If the batch driver detects a field that begins with iPars, the batch driver will process the entire address with only parsed address components. Combined address components like iAddress1 will be ignored. Conversely, if no fields are found that begin iPars, the batch driver will process the entire address using non-parsed address components like iAddress1.

If the batch driver is processing non-parsed addresses and no secondary field information is given (iUnit, iUnit2, iPMUnit), the output information will be combined with the oAddress1 field.

The following table describes the keywords for a Definition File Layout using component keywords for processing. Keywords are not case-sensitive. The maximum length for all allocated fields is 255 bytes.

**Table 10: Definition File Layout Using Component Keywords**

Definition Keywords	Description	Recommended Length
;	Comment line. Indicated by a semicolon (;) in column one. Semicolon must begin in column one for comment lines.	
; Finalist Definition File.	Comment. Enter a description of the file.	
; Created by:	Comment. Enter your name and date created.	
; Input file=Inputfile.def	Comment. Specifies the name of the file being defined.	
[INPUT]	Identifies the start of the field definition keywords. Recommended length is 7 characters exactly as shown.	
iAddress1=x,y	Identifies the position (x) and length (y) of the input street address.	70

Definition Keywords	Description	Recommended Length
iAddress2=x,y	Identifies the position (x) and length (y) of the second input address line.	70
iCity=x,y	Identifies the position (x) and length (y) of the input city name.	28
iCrrt=x,y	Identifies the position (x) and length (y) of the input carrier route.	4
iUserKey	Identifies the position [x] and length [y] of user key to identify this address in Finalist detail reports.	40
iFirm=x,y	Identifies the position (x) and length (y) of the input firm name.	40
iParsPMUnitDesignator=x,y	Identifies the position (x) and length (y) of the input parsed PMB/MSC unit designator.	3
iParsPMUnitNum=x,y	Identifies the position (x) and length (y) of the input parsed PMB/MSC unit number.	10
iParsPostDir=x,y	Identifies the position (x) and length (y) of the parsed postdirectional.	2
iParsPreDir=x,y	Identifies the position (x) and length (y) of the output parsed predirectional.	2
iParsRange=x,y	Identifies the position (x) and length (y) of the input parsed range.	10
iParsStreetName=x,y	Identifies the position (x) and length (y) of the input parsed street name.	28
iParsStreetSuffix=x,y	Identifies the position (x) and length (y) of the input parsed street suffix.	4
iParsUnit2Designator=x,y	Identifies the position (x) and length (y) of the input parsed unit 2 designator.	4
iParsUnit2Num=x,y	Identifies the position (x) and length (y) of the input parsed second unit number.	8

Definition Keywords	Description	Recommended Length
iParsUnitDesignator=x,y	Identifies the position (x) and length (y) of the input parsed unit designator.	4
iParsUnitNum=x,y	Identifies the position (x) and length (y) of the input parsed unit number.	8
iPUnit=x,y	Identifies the position (x) and length (y) of the input PMB or MSC unit.	14
iProcessDate=x,y	Identifies the position (x) and length (y) of the input process date.	4
iState=x,y	Identifies starting position (x) and length of the input state.	2
iUnit=x,y	Identifies the position (x) and length (y) of the input unit.	13
iUnit2=x,y	Identifies the position (x) and length (y) of the input unit 2.	13
iUrban=x,y	Identifies the position (x) and length (y) of the input urbanization name.	28
iZip=x,y	Identifies the position (x) and length (y) of the input ZIP Code.	5
iZip4=x,y	Identifies the position (x) and length (y) of the input ZIP + 4 code.	4

The following output (o) fields are valid for either input format specified above.

**Table 11: Definition File Output Field Component Keywords**

Definition Keywords	Description	Recommended Length
o5DigitBarCode=x,y	Identifies the position (x) and length (y) of the 5-digit barcode.	8
o5DigitScheme=x,y; a[,y]	Identifies the position (x) and length (y) of the 5-digit combined ZIP Code.	5

Definition Keywords	Description	Recommended Length
oAbbrCityName=x,y; a[,y]	Identifies the position (x) and length (y) of the output abbreviated city name.	13
oAddress1=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output street address.	70
oAddress2=x,y; a[,y]; o	Identifies the position (x) and length (y) of the second output address line.	70
oAdsInfo=x,y; a[,y]	Identifies the position (x) and length (y) of the output address information codes (PBFNAddressDataDef structure).  <b>Note:</b> The address information codes are binary data and extreme caution should be used when using CRLF or delimited files as the binary data may appear as the delimiter or line ending characters. FIXED records are recommended. COBOL cannot easily manipulate bit values. An interrogation routine like PBITBYTE is recommended for ease of use in a COBOL.	42
oAdvancedBarCode=x,y; a[,y]	Identifies the position (x) and length (y) of the output advanced barcode.	14
oAltStreetName=x,y; a[,y]	Identifies the position (x) and length (y) of the output alternate street name.	28
oAltStreetType=x,y; a[,y]	Identifies the position (x) and length (y) of the output alternate street type.	1
oAutoCR=x,y; a[,y]	Identifies the position (x) and length (y) of the output Auto CRRT indicator.	1
oCity=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output city name.	28
oCongressDist=x,y; a[,y]	Identifies the position (x) and length (y) of the output congressional district.	2
oCountyName=x,y; a[,y]	Identifies the position (x) and length (y) of the output county name.	25

Definition Keywords	Description	Recommended Length
oCrrt=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output carrier route.	4
oDefaultMatch=x,y; a[,y]	Identifies the position (x) and length (y) of the output default match flag.	1
oDelivpt=x,y; a[,y];	Identifies the position (x) and length (y) of the output delivery point code.	3
oDPBCCheckDigit=x,y; a[,y];	Identifies the position (x) and length (y) of the output Delivery Point Bar Code check digit	1
oDPVCMRA=x,y; a[,y]	Identifies the position (x) and length (y) of the output DPV CMRA indicator flag.	1
oDPVFalsePositive=x,y; a[,y];	Identifies the position (x) and length (y) of the output DPV False Positive flag.	1
oDpvFlags=x,y; a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) flags.	3
oDpvFootNote=x,y; a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) footnote codes.	12
oDpvNoStat=x,y; a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) No-Stat return code.	1
oDPVvacant=x,y,a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) vacant return code.	1
oDPVPBSA=x,y,a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) PBSA return code.	1
oDPVDNA=x,y,a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) Door Not Accessible (DNA) return code.	1
oDPVNSL=x,y,a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) No Secure Location (NSL) return code.	1
oDPVThrowback=x,y,a[,y]	Identifies the position (x) and length (y) of the output Delivery Point Validation (DPV) P.O. Box Throwback return code.	1

Definition Keywords	Description	Recommended Length
oError=x,y; a[,y]	Identifies the position (x) and length (y) of the output error code.	4
oEWSFailure=x,y; a[,y]	Identifies the position (x) and length (y) of the output EWS Failure flag (contains a 'Y' if address failed as a result of EWS processing).	1
oExtra=x,y; a[,y]	Identifies the position (x) and length (y) of the output Extra Data field.	70
oFips=x,y; a[,y]	Identifies the position (x) and length (y) of the output FIPS code.	5
oFirm=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output firm name.	40
oFullCityName=x,y; a[,y]	Identifies the position (x) and length (y) of the output full city name.	28
oLacs=x,y; a[,y];	Identifies the position (x) and length (y) of the output LACS <sup>Link</sup> indicator.	1
oLACSIndicator=x,y; a[,y];	Identifies the position (x) and length (y) of the output LACS <sup>Link</sup> processing indicator: <b>F</b> Seed violation occurred. <b>Y</b> Full LACS <sup>Link</sup> record match, <b>N</b> LACS <sup>Link</sup> attempt but no match condition (CASS Stage 2 LACS Indicator flag). <ul style="list-style-type: none"> <li>• S —Unit record match.</li> </ul>	1
oLLKS=x,y; a[,y]	Identifies the position (x) and length (y) of the output LACS <sup>Link</sup> return code.	2
oLLKSD=x,y; a[,y]	Identifies the position (x) and length (y) of the output LACS <sup>Link</sup> Seed detail.	70
oLot=x,y; a[,y];	Identifies the position (x) and length (y) of the output Line of Travel (LOT) code.	5



Definition Keywords	Description	Recommended Length
oLotAlpha=x,y; a[,y];	Identifies the position (x) and length (y) of the output alpha portion of the LOT code.	1
oLRtnAddrLine1=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of return for the first label line.	70
oLRtnAddrLine2=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of return for the second label line.	70
oLRtnAddrLine3=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of return for the third label line.	70
oLRtnAddrLine4=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of return for the fourth label line.	70
oLRtnAddrLine5=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of return for the fifth label line.	70
oNonMailingCityName=x,y; a[,y]	Identifies the position (x) and length (y) of the output nonmailing city name.	28
oMatchLevel=x,y; a[,y]	Identifies the position (x) and length (y) of the output level of match flag.	1
oOriCity=x,y; a[,y];	Identifies the position (x) and length (y) of the output original city name.	28
oOriCrrt=x,y; a[,y]	Identifies the position (x) and length (y) of the output carrier route code.	4
oOriState=x,y; a[,y]	Identifies the position (x) and length (y) of the output original state.	2
oOriZip=x,y; a[,y]	Identifies the position (x) and length (y) of the output original ZIP Code.	5
oOriZip4=x,y; a[,y]	Identifies the position (x) and length (y) of the output original ZIP + 4 code.	4
oParsAltRange=x,y; a[,y]	Identifies the position (x) and length (y) of the output parsed alternate range.	10

Definition Keywords	Description	Recommended Length
oParsAltPreDir=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed alternate predirectional.	2
oParsAltPostDir=x,y; a[,y]	Identifies the position (x) and length (y) of the output parsed alternate postdirectional.	2
oParsAltSuffix=x,y; a[,y]	Identifies the position (x) and length (y) of the output parsed alternate suffix.	4
oParsPMUnitDesignator=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed PMB/MSC unit designator.	3
oParsPMUnitNum=x,y; a[,y]	Identifies the position (x) and length (y) of the output parsed PMB/MSC unit number.	10
oParsPostDir=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed postdirectional.	2
oParsPreDir=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed predirectional.	2
oParsRange=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed range.	10
oParsStreetName=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed street name.	28
oParsStreetSuffix=x,y;a[,y]; o	Identifies the position (x) and length (y) of the output parsed street suffix.	4
oParsUnit2Designator=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed unit 2 designator.	4
oParsUnit2Num=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed second unit number.	8
oParsUnitDesignator=x,y,a[,y]; o	Identifies the position (x) and length (y) of the output parsed unit designator.	4
oParsUnitNum=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output parsed unit number.	8

Definition Keywords	Description	Recommended Length
oPreciselyID=x,y; a[,y]; o	Identifies the position(x) and length(y) of the output PreciselyID field.	12
oPreciselyIDFound=x,y; a[,y]; o	Identifies the position(x) and length(y) of the output PreciselyIDFound field.	1
oPMUnit=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output PMB or MSC unit.	14
oPOBoxZone=x,y; a[,y]	Identifies the position (x) and length (y) of the output P. O. Box Zone flag.	1
oPreferredCityName=x,y;a[,y]	<p>Identifies the position [x] and length [y] of the output preferred city name.</p> <p>For successfully-coded addresses, the oPreferredCityName and oPreferredState fields are always be populated. For non-coded addresses, the oPreferredCityName and oPreferredState fields are populated for the following scenarios:</p> <ul style="list-style-type: none"> <li>• ZIP Code only input (city not input, or not found)</li> <li>• Single ZIP Code city input (ZIP Code not input, or not found)</li> <li>• City/St/ZIP Code input and agree (ZIP Code is part of city)</li> </ul> <p>For all other non-coded scenarios, the preferred fields will be blank.</p>	28
oPreferredState=x,y;a[,y]	<p>Identifies the position [x] and length [y] of the output preferred state.</p> <p>For successfully coded addresses, the oPreferredCityName and oPreferredState fields are always populated.</p> <p>For non-coded addresses, the oPreferredCityName and oPreferredState fields are populated for the following scenarios:</p> <ul style="list-style-type: none"> <li>• ZIP Code only input (city not input, or not found)</li> <li>• Single ZIP Code city input (zip not input, or not found)</li> <li>• City/St/ZIP Code input and agree (zip is part of city)</li> </ul> <p>For all other non-coded scenarios, the preferred fields will be blank.</p>	2

Definition Keywords	Description	Recommended Length
oPreLACSAddress=x,y; a[,y]; o	Identifies the position (x) and length (y) of the original input address that was passed to LACS <sup>Link</sup> and LACS <sup>Link</sup> converted.	70
oProcessDate=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output process date.	4
oRdi=x,y; a[,y]	Identifies the position (x) and length (y) of the output Residential Delivery Indicator.	1
oSeasonalFlag=x,y; a[,y]	Identifies the position (x) and length (y) of the output seasonal flags.	12
oSikRtnCode=x,y; a[,y]	Identifies the position (x) and length (y) of the Suite <sup>Link</sup> return code.	2
oSikMatchCode=x,y; a[,y]	Identifies the position (x) and length (y) of the Suite <sup>Link</sup> match code.	1
oSikMatchFidelityCode=x,y; a[,y]	Identifies the position (x) and length (y) of the Suite <sup>Link</sup> fidelity code.	1
oState=x,y; a[,y]; o	Identifies starting position (x) and length of the output state.	2
oStreetType=x,y;a[,y]	Identifies the position [x] and length [y] of the output street type field.	1
oUnit=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output unit.	13
oUnit2=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output unit 2.	13
oUrban=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output urbanization name.	28
oXTern=x,y; a[,y]	Identifies the position (x) and length (y) of the output Precisely VeriMove move update product external CASS indicator.	250

Definition Keywords	Description	Recommended Length
oZip=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output ZIP Code.	5
oZip4=x,y; a[,y]; o	Identifies the position (x) and length (y) of the output ZIP + 4 code.	4
oZIPValid=x,y; a[,y]	Identifies the position (x) and length (y) of the output ZIPValid flag.	1

The attach/overlay options are described next.

**Table 12: Attach/Overlay Options**

Overlay/Attachment Option	Description
x,y	x=position, y=length This option allows you to specify the location for placement of an output field. For example, oFirm=1,28 indicates the output Firm field should be placed in position 1 of the output record for a length of 28.
a[,y]	a=attach, [y]=optional length This option allows you to specify the placement length for an output field that is to be attached to the end of the input record. For example, oFirm=a,28 indicates that the output Firm field will be attached to the end of the input record for a length of 28 characters.
a	a=attach This option provides you with the ability to attach individual fields to the end of your input record. This option attaches only the designated output field to the end of the input record. For example, oFirm=a indicates the output Firm field will be attached to the end of the input record.
o	o=overlay This option provides you with the ability to overlay individual fields over corresponding fields in your input record. This option overlays only the designated output field over the corresponding input field. This option is only available if a corresponding input field has been specified and you have specified jOverlay=NO. For example, oFirm=o indicates that the output Firm field will overlay the input Firm field (provided you have specified a corresponding iFirm field).

**Figure 10: Overlay Processing**

During overlay processing (jOverlay=YES), the keywords define the input and output locations. The driver overlays input information with corrected data on successful verification.

You can perform overlay processing at the field level when you have specified jOverlay=NO. The field level overlay option allows you to overlay a specified output field over the corresponding input

field. For example, oFirm=o indicates that the output Firm field will overlay the input Firm field (provided you have specified a corresponding iFirm field).

**Note:** Proceed with caution when selecting jOverlay=YES processing until fully tested to avoid losing your original address information.

### Figure 11: Attach Processing

During attach processing (jAttach=YES), the keywords are only used as reference for input to the driver program. The resulting information is attached to each input record and the new record is written to a separate output file as indicated in the Job File. Definition Files in this format can be constructed by the Finalist Workbench, which is distributed with the Windows product.

## Options for Returning Results From the Batch Driver

Depending on the level of detail you have within your address file, you can:

- Have the driver overlay the address data in the original input file
- Have the output directed to new output file(s)

It is strongly recommended that you fully understand use of the OVERLAY option and carefully consider which method best suits your processing needs. Special consideration should be paid to the level of detail available for specific address components. There are two methods available for returning results.

### Overlay Option

The driver overlays the original input record address fields with the results. Each specific address component identified as input in the Definition File is overlaid with the information returned from the Finalist API call. To use the Overlay Option, in the [OPTIONS] section of the Job File, specify jOverlay=YES.

Note that any address components that are not specified as input in the Definition File cannot be overlaid with return information and may result in incomplete and/or incompatible address information existing within your data file. For example, let's say your input ZIP Code exists as part of the input field defined as the city name (iCity in the Definition File). The driver program cannot return the ZIP Code because you did not define a location for it. This would result in you losing the original ZIP Code from your input record because the driver would overlay your original city, state and ZIP Code with just the city name.

You should only use the overlay option when you know the exact location of specific address components. Ideally, you can use this option to update your address file in a single pass without the need for a site-specific post processing application to analyze a separate output file containing the Finalist results.

You should use block-type address lines and the jOverlay=NO option if either of the following conditions are true:

- Your address data fields are not in fixed locations within the data record.
- Multiple address components float within a certain range in the record.

### Attach Option

The driver writes a copy of the input record and the returned information to separate output file(s). Specify this option in the Job File (jOverlay=NO). This option allows you to post-process the output file created by the driver. For information on the format of the output file(s) created by the driver, refer to **Contents of jAttach Record** on page 119.

This type of processing typically requires a custom post-processing application to analyze the results. The application would interrogate the output file created by the driver to inspect the input record, the error code field, and the addressing results. The application would determine the completeness of the original address, and may update your master file with the complete address information.

After processing your file with the driver for the first time, consider having a post-processing application reformat the original data records to include the returned address components. This would allow you to use the driver jOverlay=YES option for future executions. The jOverlay=YES option requires little or no post-processing of the results since the corrected information is written directly over the original data. Only records that could not be successfully coded would need to be processed, and you can easily identify these records using the fOutErrorsFile keyword in the Job File.

## Contents of jAttach Record

During driver processing, records written to any of the output files contain the following data.

- A copy of the original input record with any fields overlaid as requested.
- If you specify jAttach=YES, return information is appended to the end of the file. For more information, refer to the following table.
- If you requested an individual field to be attached, that field is appended following the optional jAttach=YES fields. Individual fields will be appended in the order specified in the .def file.

For each processing run, the corresponding .log file will contain the actual position for the location of each of these fields. Refer to the "Return Address Information Layout" section of the .log file. If your input is in a dBase format, a message displays indicating position and length do not apply.

**Table 13: Contents of Output Record**

Description	Length (in bytes)
Firm	40
Urbanization Code	28

Description	Length (in bytes)
Address	70
Address2	70
Unit <sup>1</sup>	13
Unit2 <sup>1</sup>	13
City	28
State	2
Carrier Route	4
Private Mail Box (PMB)/Mail Stop Code (MSC) <sup>1</sup>	14
FIPS Code	5
County Name	25
Left Framing Character (used for advanced barcode)	1
ZIP Code (used for advanced barcode)	5
ZIP + 4 Code (used for advanced barcode)	4
Delivery Point (includes check digit)	3
Right Framing Character (used for advanced barcode)	1
5 Digit Barcode	8
LOT	5
Full City Name	28
Abbreviated City Name	13
Nonmailing City Name	28



Description	Length (in bytes)
Match Level	1
Default Match	1
LACS Indicator	1
Auto Carrier Route	1
Residential Delivery Indicator (RDI)	1
5 Digit Scheme	5
Congressional District	2
DPV Flags (DPV, CMRA, False Positive)	3
Delivery Point Validation (DPV) Footnotes	12
Seasonal Indicators	12
Error Code	4
Process Date	4
Optional Information Codes (for use with the Precisely VeriMove product)	250
LACS <sup>Link</sup> Return Code	2
LACS <sup>Link</sup> Seed Detail	70
Suite <sup>Link</sup> Return Code	2
Suite <sup>Link</sup> Match Code	1
Suite <sup>Link</sup> Fidelity Code	1

<sup>1</sup> For compatibility reasons, these fields are not populated unless specifically requested as an additional field. In that case, these fields are populated.

# Building Batch Jobs with the Workbench

This section provides information on how you can use the Finalist Workbench to create the Job File (.job) and Definition File (.def) required to process with the batch driver. The Finalist Workbench runs in a Windows environment.

## Starting Finalist Workbench

The steps for accessing the Finalist Workbench are:

1. From the Start menu, select Programs.
2. Select Precisely.
3. From the PB menu, select FinalistNNN.
4. Select Workbench.

The Workbench window displays.

The bar along the top of the window displays the menus available from the Finalist Workbench window. A brief description of each menu follows.

### ***File Menu***

The File menu options are:

<b>New</b>	Select New from the File to create a new Definition File (.def) or a Job File (.job). The Definition File (.def) and Job File (.job) are needed to process with the driver.
<b>Open</b>	Select Open to open an existing file.
<b>Print Setup</b>	Select Print Setup to access the Print Setup window.
<b>Recent File</b>	This area of the File menu display recently-used files.
<b>Exit</b>	Select Exit to exit Finalist Workbench.

### ***View Menu***

The View menu options are:

<b>Toolbar</b>	Select Toolbar to indicate whether to display the toolbar on the Finalist Workbench windows.
<b>Status</b>	Select Status to indicate whether to display the status bar on the Finalist Workbench windows.

## Tools Menu

The Tools menu options are:

- Run Batch Job** Select Run Batch Job to start the driver program. For detailed information on the driver program, refer to [Running a Batch Job via the Windows Command Line](#) on page 137.
- Run Lookup** Select Run Lookup to run the Lookup Tool. You can use the Lookup Tool to search the contents of the postal data files for a specific entry. For detailed information on the Lookup Tool, refer to "Using the Windows Lookup Tool" .
- Run Distribution** Select Run Distribution to run the Distribution Tool. You can use the Distribution Tool to build a Finalist database that includes only selected states. For detailed information on the Distribution Tool, refer to "Using the Distribution Tool" in your Finalist Installation Guide.
- PBFN Setup** Select PBFN Setup to define the configuration settings. The configuration settings define various program specifications and options. You can also access PBFN Setup from the Edit menu in the Lookup Tool. For detailed information on configuration settings, refer to [Configuring Finalist](#) on page 13.

## Help Menu

The Help menu options are:

- About Workbench** Select About Workbench to access information on the current version of Finalist Workbench.
- Contents** Provides global online access to all of the Finalist documentation in help format.
- Installation Help** Provides online access to the Finalist Installation Guide
- User's Help** Provides online access to the Finalist User's Guide.
- Reference Help** Provides online access to the Finalist Reference Guide.

## Creating a Definition File (.def)

You can use Finalist Workbench to create the Definition File (.def) required for processing with the driver. The Definition File defines the layout of the input file to be processed by the driver program. The Definition File contains keywords that specify the location and length of the input fields that the driver program will pass to Finalist. For information on the Definition file keywords, refer to [Definition File](#) on page 103. To create a Definition File:

1. From the File menu, select New. The New Definition File dialog box displays.
2. Select def on the New Definition File dialog box. The Definition File Setup dialog box displays. Use this dialog box to define your input and output fields.

## Defining Your Input Fields

Use the top portion of the Definition File Setup dialog box to define the name, position, and length of your input fields. The Field Name box in the Input Defined section lists the keywords for the input fields you can add to your Definition File. The Input Defined list is blank when you first access this screen. You can use the buttons in the Input Field Actions section to build or change the Input Defined list.

1. Select the input field to add to your Definition file from the drop down Field Name list.
2. Select the delimited file type for your input file to identify the method used to separate components of the address data in your input file:
  - a) If your input file is in a fixed field format, click Fixed. Proceed to step 3.
  - b) If your input file is in a delimited format, click Delimited. Proceed to step 3.
  - c) If your input file is in a dBase format, select dBase field to define your dBase input file. Specify the appropriate dBase field names for the fields you want to add to your Definition file. Proceed to step 4.
3. Use the Position, and Length fields to define your input file by specifying the starting position and length of each field in the input file.
  - a) Use the Position/Field box to define the starting position in your input file for the field you selected in the Field Name box. Select a number in the range 1-1024 for the starting position of the field you selected.
  - b) Use the Length/Width box to define the length of the field you selected in the Field Name box. Select a number in the range 1-1024 for the length of the field you selected.
4. Click Insert Input Field.

### Input Field Actions

The Input Field Actions section lists the actions you can perform for the Input Definition File fields.

<b>Insert Input Field</b>	Adds the field you selected and defined to the Input Defined section and the Definition File.
<b>Up</b>	Moves a selected field up in the order specified in the Input Defined list.
<b>Down</b>	Moves a selected field down in the order specified in the Input Defined list.
<b>Delete Field</b>	Deletes the highlighted field from the Definition File.
<b>Update Field</b>	Updates a selected field in an existing Definition File. The steps to change a field in a Definition File are: <ol style="list-style-type: none"> <li>1. Open the existing Definition File.</li> <li>2. Select the field in the Input Defined list that you want to update.</li> <li>3. Make your changes to the Position/Field and/or Length/Width field(s) in the Delimited File Types section.</li> </ol>

	4. Click on the Update button to update the Definition File with your changes. Note that the changes you made are now reflected in the Input Defined list.
<b>Clear List</b>	Deletes all items in the Input Defined list.

## Defining Your Output Fields

Use the bottom portion of the Definition File Setup Screen to define the format for your output file. The Field Name box in the Output Defined section lists the keywords for the output fields you can add to your Definition File. This list is blank when you first access this screen. You can use the buttons in the Output Field Actions section to build or change the Output Defined list.

1. Select the output field to add to your Definition file from the drop down Field Name list. Before the field is added to the Output Defined list, you must define the location for the output field you selected. You can specify a location by:
  - Defining the starting position and length of the output field
  - Attaching the output field to the end of the input record
  - Overlaying the input field location with the output results
2. To identify the location for the output field by defining the actual position and length, click Field.
  - a) In the Position box, enter the starting position in your output file for the output field you selected.
  - b) In the Length box, enter the length in your output file for the selected output field.
  - c) Proceed to Step 5.

For example, if you specify a position of 1 and a length of 28 for oFirm, the output Firm field will be placed in position 1 of the output record for a length of 28.

3. To identify the location of an output field that is to be attached to the end of the input record, click Attach.
  - a) In the Length box, specify the length for the output field to attach to the end of the input record.
  - b) Proceed to Step 5.

For example, if you specify a length of 28 for oFirm, the output Firm field will be attached to the end of the input record for a length of 28 characters.

4. To overlay the selected output field over the corresponding input field, click Overlay. Proceed to Step 5.

**Note:** This option is only available if a corresponding input field has been specified.

For example, if you specify oFirm, the output Firm field (oFirm) will overlay the input Firm field (provided you have specified a corresponding iFirm field). In this example, if no iFirm has been defined, an initialization error will occur and be written in the output log.

5. Click Insert Output Field.

<b>Output Field Actions</b>	
The Output Field Actions section lists the actions you can perform for the Output Definition File fields.	
<b>Insert Output Field</b>	Adds the field you selected and defined to the Output Defined section and the Definition File.
<b>Up</b>	Moves a selected field up in the order specified in the Output Defined list.
<b>Down</b>	Moves a selected field down in the order specified in the Output Defined list.
<b>Delete Field</b>	Deletes the highlighted field from the Definition File.
<b>Update Field</b>	Updates a selected field in an existing Definition File. To change an output field in a Definition File, follow these steps. <ol style="list-style-type: none"> <li>1. Open the existing Definition File.</li> <li>2. Select the field in the Output Defined list that you want to update.</li> <li>3. Make your changes to the Field, Attach, or Overlay field in the Output Field Type section.</li> <li>4. Click on the Update button to update the Definition File with your changes. Note that the changes you made are now reflected in the Output Defined list.</li> </ol>
<b>Clear List</b>	Deletes all items in the Output Defined list.

## ***Saving Your Definition File***

Your changes do not become a permanent part of your Definition File until you save the changes. To save the Definition File you created or changed, click on the Save button or click on Save from the File menu.

To close the Definition File without saving any changes, click on the Close button in the top right corner of the screen or click Close from the File menu.

## **Creating the Job File (.job)**

You can use Finalist Workbench to create the Job File (.job) required for processing with the driver. The Job File defines your job setup including:

- Job processing options
- Input file format
- Processing mode (CASS vs. non-CASS processing)

- Job file locations
- Job reports
- Output files
- Output return options

For information on the Job file keywords, refer to **Job File** on page 90.

To create a Job File:

1. From the File menu, select New. The New Definition File dialog box displays.
2. In the New Definition File dialog box, select job. The Batch Job Set Up dialog box displays. To define the input and output options for your Job File (.job), select the Options tab.

## **Defining Your Job Options**

This section provides information for defining your job options.

### **Defining USPS 3553 (CASS Summary Report) Options**

This section provides information for defining your job options.

To process in a CASS-certified mode and generate the USPS Form 3553 (CASS Summary Report), click on CASS Mode.

**Note:** You must process in a CASS-certified mode to generate the USPS Form 3553 (CASS Summary Report). The CASS Flag in pbfm.cfg (or your configuration file) should also be set to ON in order to generate USPS Form 3553 (CASS Summary Report).

## **Defining Processing Options**

This section provides information for defining your job options.

1. In the Start after record number field, enter a number in the range 1-9999999 to bypass records before the driver begins processing.

For example, you want to begin processing with record number 25. You would enter 24 in the Start After field. The driver bypasses 24 records and begins processing with record number 25.

2. In the Stop after record number field, enter a number in the range 1-9999999 to define the number of records to process before the driver ends processing. The default value for this field is EOF indicating processing will end at the end of your input file. For example, to end processing after processing 50 records. You would enter 50 in the Stop after record number field. The driver processes 50 records and ends processing.

## **Defining AddrScan Processing Options**

This section provides information for defining your AddrScan processing options.

1. To perform AddrScan processing, click AddrScan. For more information on AddrScan processing, refer to "AddrScan Option" on page 159.

**Note:** AddrScan can only be used if your input is defined in label line format, using the iL1-iL6 input field names. If defining any address components, like iAddress1, or iCity, etc. will result in an initialization error being written out to the joblog.

2. To define the AddrScan addressing options, follow these steps:
  - a) To prevent AddrScan from concatenating lines beginning with # to existing address lines, enter # in the AddrScanAO field. This field is only available when AddrScan processing is selected (see previous step).
  - b) To prevent AddrScan from recognizing periods as valid characters and to specify that periods should be removed before scanning address lines, enter a P in the AddrScanAO field. This field is only available when AddrScan processing is selected (see previous step).
  - c) To prevent AddrScan from concatenating one-word lines to address lines, enter a D in the AddrScanAO field. This field is only available when AddrScan processing is selected (see previous step).

**Note:** The values #, P, and D can populate any of the first three positions of the AddrScanAO field.

3. To define the AddrScan return options and override the Returned Line Order, you can specify up to 9 codes in the AddrScanRO field. For more details, refer to Chapter 8 AddrScan Option. Valid Returned Line Order codes are:

**Table 14: Returned Line Order**

Code	Description
0	Firm line
1	PO Box address line
2	Address line with both a range and a suffix word
3	Address line with a range but no suffix, address line with a suffix but no range
4	Rural route address line
5	Personal name, firm name (without firm words), unidentified
6	Apartment type line
7	Possible city line
8	City line



Code	Description
9	Ignore this line
R	Rural route address line preceding a box line
B	Box address line following a rural route address line
M	Military address line

### Defining Barcode Framing Options

To define a barcode framing character for your job, specify one or two characters to be used for the beginning and/or ending characters for barcodes.

1. If you specify only one character in the Barcode Frame Character field, Finalist uses the specified character for both the beginning and ending frame character.
2. If you specify two characters, Finalist uses the first character as the beginning frame character and the second character as the ending frame character.
3. If the field is blank, Finalist uses the exclamation point (!) as the default value.

### Defining Your Input File

To define your input file options, follow these steps:

1. To define your input file type, select one of the following:
  - a) If your input file is in an ASCII fixed file format and each record in your input file occupies a fixed number of bytes (no CR/LF at the end of each record), click ASCII for Fixed Input. To define the record length of the records in your input file, enter a number in the range of 1-32,767 in the Record Size field.
  - b) If your input file is in a mainframe fixed file format and each record in your input file occupies a fixed number of bytes (no CR/LF at the end of each record), click MF for Fixed Input. To define the record length of the records in your input file, enter a number in the range of 1-32,767 in the Record Size field.
  - c) If your output file is in a mainframe file format or you want to define an output Record Size, enter a number in the range 1-32,767 in the Record Size Out field.
  - d) If each record in your input file is terminated with a carriage return/line feed (CR/LF) pair, click ASCII CR/LF Input.
2. If the records in your input file are delimited by a specific character called a delimiter (a character that marks the beginning or end of a unit of data), select one of the following:
  - a) If the records in your input file are delimited by a tab, click Tab.
  - b) If the records in your input file are delimited by a comma, click Comma.

- c) If the records in your input file are delimited by a specific character, click Character. Specify the delimiter character in the field to the right of Character.
  - d) If the records in your input file are delimited by a hexadecimal value, click Hex value. Specify the delimiter character in the field to the right of Hex value. For example, if you enter 09 in the delimiter field, 09 is the ASCII hexadecimal value for the tab character.
3. If your input file is a dBase file, click dBase Input File.

## Defining Your Job Files

To define the files for your batch job, follow these steps:

1. To identify the location of key files for your Job File (.job), from the Batch Job Set Up dialog box, select the File Locator tab.
2. In the Input File field, enter the name of the input file you want to process with the driver program. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
3. In the Definition File field, enter the name of the Definition File you want to process with the driver program. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.
4. In the PBFN.CFG File field, enter the name of your configuration file. This is an optional field. This field defaults to the pbfncfg located in the home directory. If you want to use an alternate pbfncfg, enter the path and file name of the alternate pbfncfg you want to use. Click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.

## Defining Your Report Files

To define the report information for your batch job, follow these steps:

1. To identify the filenames for the report files and the mailer's name and address information for Box D3 on the USPS Form 3553 (CASS Summary Report) for your Job File (.job), from the Batch Job Set Up dialog box, select the Report tab.
2. In the Batch Report field, enter the name of the file to contain the processing statistics. Click on the Browse button to open a dialog box to specify a path and file name for the Finalist Batch Report file. For information on the Finalist reports, refer to [Using the Finalist Reports](#).
3. In the Log File Name field, enter the name of the file to contain the Finalist engine log messages. Click on the Browse button to open a dialog box to specify a name for the Log File. For information on the Finalist reports, refer to [Using the Finalist Reports](#).
4. In the Detail Report field, enter the name of the file to contain the processing statistics. Click on the Browse button to open a dialog box to specify a name for the Address Detail Report File. For information on the Finalist reports, refer to [Using the Finalist Reports](#).
5. In the Mailer Name field, enter up to 30 alphanumeric characters for the mailer name to print in Box D3 on the USPS Form 3553 (CASS Summary Report).

**Note:** The mailer fields described here allow more than 30 characters; however, only 30 characters can be written to each line in the D3 box on the USPS Form 3553 (CASS Summary Report).

6. In the Mailer City Line field, enter up to 30 alphanumeric characters for the mailer city, state, and ZIP Code to print in Box D3 on the USPS Form 3553 (CASS Summary Report).
7. In the Mailer Address field, enter up to 30 alphanumeric characters for the mailer address to print in Box D3 on the USPS Form 3553 (CASS Summary Report).
8. In the List Processor field, enter the name of the list processor. This is usually your company's name.

## Defining Your Job Output

To define the output for your batch job, follow these steps:

1. To identify the returned data options and the path and filenames for the output files for your Job File (.job), from the Batch Job Set Up dialog box, select the Output tab.
2. In the Output File field (optional), enter the name of the Output File that will contain all records passed to the driver. Click on the Browse button (located to the right of the field) to open a dialog box to specify a path and file name for the Output File.
3. In the Valid File field (optional), enter the name of the file to contain all records passed to the driver that were successfully coded. Click on the Browse button (located to the right of the field) to open a dialog box to specify a path and file name for the Valid File.
4. In the Errors File (optional), enter the name of the file to contain all records passed to the driver that failed coding attempts. Click on the Browse button (located to the right of the field) to open a dialog box to specify a path and file name for the Errors File.
5. To overwrite your original input file with the processing results of the driver program, click Overlay Option. For information on overlay processing, refer to "Overlay Processing" in **Format Two: Address Component Keyword Definitions** on page 107.

**Note:** Proceed with caution when selecting overlay processing to avoid losing your original address information.

6. To have the driver program write the returned information and a copy of the original input record to a separate output file, click Attach Option. The driver writes the information in the following order:
  1. Original input record
  2. Processing results
  3. Error code

For detailed information on attach processing, refer to "Attach Processing" in **Format Two: Address Component Keyword Definitions** on page 107.
7. To return data in the Precisely VeriMove format, click FT CASS Option.

8. To prevent an output field from overlaying an input field unless the output and input field formats are identical, click OverlayCheck. If OverlayCheck is set to ON (default value), the Finalist driver ensures that output fields do not overlay input fields.

To create an output file with a format that is not identical to the format of the input file, turn off OverlayCheck. If OverlayCheck is set to OFF, the output file is initially blank and then only populated by output fields specified in the corresponding job definition (DEF) file.

For detailed information on OverlayCheck processing, refer to cOverlayCheck in the **{OPTIONS} Section** on page 93.

## ***Saving Your Job File***

Your changes do not become a permanent part of your Job file until you save the changes. To save the Job file you created or changed, click on the Save button or click on Save from the File menu.

To close the Job File without saving any changes, click on the Close button in the top right corner of the screen or click Close from the File menu.

## **Accessing the Finalist Help**

To access the entire set of Finalist documentation from a single access point in an online Help format, click Help > Contents.

You can also select Help then Installation Help, User's Help, or Reference Help to access each guide individually.

## **Accessing Version Information**

You can use the About Workbench Option to access current version information. From the Help menu, click About Workbench. The About Workbench dialog box displays the current version information for the Finalist Workbench including whether you are processing in a 32-bit or 64-bit mode and the expiration date for your software key. Click OK to return to the Workbench.

# 5 - Running a Job

## In this section

---

Running a Batch Job on z/OS.....	134
Running a Batch Job with the Workbench.....	137
Running a Batch Job via the Windows Command Line.....	137
Running a Batch Job via the Unix/Linux Command Line.....	139



## Running a Batch Job on z/OS

The following is sample JCL that executes Finalist with the Finalist driver program (FINALIST) on a z/OS platform.

**Note:** Finalist requires a region size of approximately 100M. Region size will vary depending on options selected.

```

1 //stepname EXEC PGM=FINALIST,REGION=0M
2 //STEPLIB DD DSN=user.loadlib,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
// DD DSN=CEE.SCEERUN2,DISP=SHR
3 //CBDATA DD DSN=finalist.datafile,DISP=SHR
4 //CBCTYST DD DSN=finalist.citystate,DISP=SHR
//* EWS FILE
//CBEWS DD DISP=SHR,DSN=finalist.EWSFILE
//* ELOT FILE
//LTDATA DD DISP=SHR,DSN=lot.DATAFILE
//* LACS FILES
//LLKDB DD DISP=SHR,DSN=hlq.FINALIST.LLKDB
//LLKSUD DD DISP=SHR,DSN=hlq.FINALIST.LLKSUD
//* DPV FILES
//DPVDB DD DISP=SHR,DSN=hlq.FINALIST.DPVDB
//DPVSDB DD DISP=SHR,DSN=hlq.FINALIST.DPVSDB
//DPVHDB DD DISP=SHR,DSN=hlq.FINALIST.DPVHDB
//DPVSUD DD DISP=SHR,DSN=hlq.FINALIST.DPVSUD
//* SUITELINK FILES
//SLKDB DD DISP=SHR,DSN=hlq.FINALIST.SLKDB
//RDIDB DD DISP=SHR,DSN=hlq.FINALIST.RDIDB
//PRCSLYDB DD DISP=SHR,DSN=hlq.FINALIST.PRCSLYDB
5 //PBFNJOB DD *
; These cards are sample JOB cards
[FILES]
fInput=INFILE
fDefinition=PBFNDEF
fOutErrorsFile=BADFILE
fOutValidFile=GOODFILE
fOutAllFile=OUTFILE
fConfigFile=PBFNCFG
[OPTIONS]
cInputFileType=MFFIXED
cRecSz=600
jAttach=no
6 //PBFNDEF DD *
; These cards are sample DEF cards
[INPUT]
iFirm=278,40
iUrban=318,28

```

```

iAddress1=346,64
iAddress2=410,64
iCity=474,42
7 //PBFNCFG DD *
--Sample config cards
FILE SECTION:
-----
CITY DIRECTORY FILENAME = CBCTYST
ZIP+4 DIRECTORY FILENAME 1 = CBDATA
CASS SECTION:
-----
BATCH REPORT FILENAME = SYSFINAL
3553 REPORT FILENAME = SYSPRT7
PRODUCT SECTION:
-----
SOFTWARE KEY = your software key
REPORT SECTION:
-----
REPORT FILENAME = SYSPRT3
REPORT TITLE = FINALIST REPORT FILE FROM PBFN.CFG
Address Detail Report PAGE LEN = 60
Address Detail Report MAX REC = 1000
Address Detail Report NTH REC = 1000
Address Detail Report TYPE = ON
Address Detail Report ISOL = ON
Address Detail Report SUGG = ON
Address Detail Report INFO = ON
LOG MSG SECTION:
-----
LOG FILENAME = SYSPRT1
LOG LEVEL = 3
8 //INFILE DD DSN=user.input.file,DISP=SHR
9 //OUTFILE DD DSN=user.output.file,DISP=SHR
10 //GOODFILE DD DSN=user.optional.file,DISP=SHR
//BADFILE DD DSN=user.optional.file,DISP=SHR
11 //SYSPRINT DD SYSOUT=*
12 //SYSFINAL DD SYSOUT=*
13 //SYSPRT1 DD SYSOUT=*
14 //SYSPRT3 DD SYSOUT=*
15 //SYSPRT7 DD SYSOUT=*
16 //EXCPIN DD *
17 L1CANTEN CANTON RP356
/*
//

```

Enter the values that appear in capital letters exactly as shown and in the same logical sequence. Enter the values that appear in lower-case letters with the appropriate substitutions to meet your installation requirements. A number appears to the left of each statement that contains a user-defined variable. These numbers correspond to the following descriptions.

Statement Number	Description
1	This statement executes FINALIST. Enter a statement name for "stepname," and any other values required for your site.
2	This statement identifies the load library containing the Finalist driver, programs, and subroutines. Replace "user.loadlib" with the name of the library containing the Finalist programs and subroutines. You may also need to specify the names of your LE runtime libraries.
3	This statement identifies your Finalist Data File on disk. Replace "finalist.datafile" with the name of your Data File.
4	This statement identifies your Finalist City/State File on disk. Replace "finalist.citystate" with the name of your City/State File.
5	This statement identifies the JOB cards necessary to run FINALIST.
6	This statement identifies the DEF cards necessary to run FINALIST. In this case the JOB card specified PBFNDEF for the DEF cards. It could have specified a different DD name.
7	This statement identifies the configuration cards necessary to run FINALIST. In this case the JOB card specified PBFNCNFG for the configuration cards. It could have specified a different DD name.
8	This statement defines your input name-and-address file as identified by the JOB card. It could have specified a different DD name.
9	This statement defines your output name-and-address file. It could have specified a different DD name.
10	This statement defines additional output file(s). It could have specified different DD names.
11	The SYSPRINT statement identifies the writer (print) class for the Finalist driver program (FINALIST). It contains a list of the parameters presented to the input and a list of options in effect for this run.
12	The SYSFINAL statement identifies the writer (print) class for Finalist. It contains the Finalist Batch Report.
13	This statement identifies where the Finalist system will write any error messages.
14	This statement identifies the location of optional reports.



Statement Number	Description
15	This statement identifies the writer (print) class for the USPS Form 3553 (CASS Summary Report).
16-17	Statement 16 activates the exceptions table processing logic when running the Finalist system Data File. Statement 17 is a sample entry to the exceptions table. You can also have exceptions table processing as a separate data set instead of in-stream data. If you use a separate data set, the record length must be 80. To use Finalist without the exceptions table, either omit these two statements or use the following single statement instead of statement 16 and 17:  <code>//EXCPIN DD DUMMY</code>

## Running a Batch Job with the Workbench

You can use the Workbench Batch Option to run a batch job on Windows:

1. From the Tools Menu, click Run Batch Job. The Open dialog box displays.
2. Use the Open dialog box to navigate to and select the batch job to run.
3. Click the Open button to start the driver program and the Finalist postal coding process.
4. A window displays processing statistics and error messages as processing continues along with the file names for all files created during processing.

## Running a Batch Job via the Windows Command Line

To execute Finalist via a Windows command line:

1. Navigate to the directory containing the Finalist executable (typically C:\Program Files\Precisely\FinalistXXX\bin)
2. Execute the Finalist batch driver by issuing the command:

```
.\Finalist.exe
```

3. The syntax for parameters passed into the batch driver is as follows:

```
Usage: finalist jobfile | -j jobfile [-g configfile][-l logfile][-c][-k][-v] [noconfirm][nokeypress][?][-h?]
```

Where:

Parameter	Description
jobfile	Job file name (argument).
-j jobfile	Job file name (option).
-g configfile	PBFN configuration file name.
-l logfile	Log file name.
-q	Do not print 'Processing record.
-c	Overlay will occur without confirmation.
-k	Processing will terminate without user intervention.
-v	Display PBFN version information.
noconfirm	(like -c) Overlay will occur without confirmation.
nokeypress	(like -k) Processing will end without user intervention.
?	Display PBFN version information.

- A job file name must be specified either as an argument to the command line or using the -j option.
- If the -g configfile option is not used, the default config file .\pbfncfg is used, unless a config file name is given in the job file.
- The default log file is the job file name with a .log extension.
- The default is to confirm overlay.
- The default is to prompt for a key stroke before terminating.
- Options and arguments can be in upper case, but file names are case sensitive.

Finalist.exe can also be safely executed via a batch script file (\*.bat). Finalist.exe requires access to pbfncfg.dll. This can be done by executing while in the ./bin folder described above or by placing the Finalist ./bin folder in the Windows "path".

If you are executing with the DPV or LACS<sup>Link</sup> options, the security files (xxX.su\$) must be in your current directory.

# Running a Batch Job via the Unix/Linux Command Line

To execute Finalist via a Unix/Linux command line:

1. Navigate to the directory containing the Finalist executable (typically /precisely/finalist/bin).
2. Execute the Finalist batch driver by issuing the command:

```
./finalist
```

3. The syntax for parameters passed into the batch driver is as follows:

```
Usage: finalist jobfile | -j jobfile [-g configfile][-l logfile][-c][-k][-v] [noconfirm][nokeypress][?][-h?]
```

Where:

Parameter	Description
jobfile	Job file name (argument).
-j jobfile	Job file name (option).
-g configfile	PBFN configuration file name.
-l logfile	Log file name.
-q	Do not print 'Processing record.
-c	Overlay will occur without confirmation.
-k	Processing will terminate without user intervention.
-v	Display PBFN version information.
noconfirm	(like -c) Overlay will occur without confirmation.
nokeypress	(like -k) Processing will end without user intervention.
?	Display PBFN version information.

- A job file name must be specified either as an argument to the command line or using the -j option.
- If the -c configfile option is not used, the default config file ./pbfm.cfg is used, unless a config file name is given in the job file.
- The default log file is the job file name with a .log extension.
- The default is to confirm overlay.
- The default is to prompt for a key stroke before terminating.
- Options and arguments can be in upper case, but file names are case sensitive.

If you are executing with the DPV or LACS<sup>Link</sup> options, the security files (xxX.su\$) must be in your current directory.

# 6 - Using the Finalist Reports

## In this section

---

Address Detail Report.....	142
Finalist Batch Report.....	152
USPS Form 3553 (CASS Summary Report).....	176



# Address Detail Report

The Address Detail Report consists of four sections providing processing information for each input address. You can set report options in the Configuration File (pbfncfg) or in the PBFNSetupDef structure.

**Note:** You can specify "ERR" for the "Address Detail Report Type" option in the pbfncfg configuration file or the PBFNSetupDef structure to include only records that did not match on the Address Detail Report.

For all fields on the Address Detail Report, if data is not found for a field, the field does not display on the report.

# Sample Address Detail Report - Page 1

```

Process Date: XX/XX/XX
Process Time: XX:XX:XX
X.XX.XX.X.XX
Finalist Rel: XXXX
Date: XX-XX-XXXX
Page: 1
DB Version:
DB Copyright

Finalist Report File From PBFN.CFG
Address Detail Report
RECORD NO. 7

##### ADDRESS INPUT/OUTPUT REPORT #####
Input Address:
Address:
ADDR1: 2358 S 9TH
* 2358 S 9TH
CSZ: MILW WI
MILW WI

4423
##### ADDRESS ISOLATION REPORT #####
RANGE: 2358
PRE-DIR: S
STREET: 9TH
NO.:
SUFFIX:
POST-DIR:
NO.:
PMB DES:
NUM:
##### ADDRESS SUGGESTION REPORT #####

2358 S 9TH PL, MILWAUKEE WI 53215-3212 C076
2358 S 9TH ST, MILWAUKEE WI 53215-3208 C027

```

## Sample Address Detail Report - Page 2

```

Process Date: XX/XX/XX
Process Time: XX:XX:XX
  X.XX.XX.X.XX
Finalist Rel: XXXX
Copyright Date: XX-XX-XXXX
                                     Finalist Report File From PBFN.CFG
                                     Address Detail Report
##### INFO CODES REPORT #####
Match:
  Return Level : Unknown
Locations:
  Address      : Line 1
Types:
  Address      : Street
  Lookup       : City Only
  City         : Single Zone, Primary
  ZIP          : Regular ZIP
Components:
  Exact Match  : Range, Pre Dir, Street, State
  No Input     : Suffix, PMB, PMB #, ZIP, Carr, LOT, LACS, URB
  Corrected   : City
  Invalid     : City
  Inv. Corre'd : City
  Multi Choice : Suffix, Carr
UserKey

```

## Address Input/Output Report Section

The Address Input/Output Report section displays your input address and the output address generated during Finalist processing. To print the Address Input/Output Report on the Address Detail Report, specify "ON" for "Address Detail Report Type" field in the Configuration File or "ON" for cAddrDtlRptType in the PBFNSetupDef structure.

**Table 15: Address Input/Output Report Section**

Field	Description
USERKEY	User key that identifies the address in the Address Detail Report.



Field	Description
FIRM	Firm name associated with the address.
URB	Urbanization name.
ADDR1	Information contained on the first address line. An asterisk (*) indicates the ADDR1 line was used as primary address line.
ADDR2	Information contained on the second address line. An asterisk (*) indicates the ADDR2 line was used as primary address line.
CSZ	City, state, and ZIP Code.
SS	ZIP + 4 code.
CR	Carrier route code.
COUNTY	County name.
DP	Delivery point.
ERROR	Any applicable error code.

## Address Isolation Report Section

The Address Isolation Report section provides details on how the Finalist engine isolated address elements from your input address before attempting to standardize and code the address according to USPS regulations. To print the Address Isolation Report on the Address Detail Report, specify "ON" for the "Address Detail Report Isol" field in the Configuration File or "ON" for cAddrDtlRptIsol in the PBFNSetupDef structure.

**Table 16: Address Isolation Report Section**

Field	Description
USERKEY	User key that identifies the address in the Address Detail Report.
RANGE	Primary range information.

Field	Description
PRE-DIR	Predirectional.
STREET	Street information.
SUFFIX	Suffix.
POST-DIR	Post-directional.
PMB DES	Private Mail Box (PMB) or Mail Stop Code (MSC) designator.
UNIT	First unit designator.
UNIT NO	Unit number.
UNIT2	Second unit designator.
UNIT2 NO	Second unit number.
PMB NUM	Private Mail Box (PMB) or Mail Stop Code (MSC) range.

## Address Suggestion Report Section

The Address Suggestion Report section shows complete address information for up to three suggestions for ambiguous address failures. For example, Finalist is unable to code your 47 Wilder Ave input address but generates a suggestion indicating an address 47 Wilder Rd exists for the same city, state, and ZIP Code. To print the Address Suggestion Report on the Address Detail Report, specify "ON" for the "Address Detail Report Sugg" field in the Configuration File or "ON" for cAddrDtlRptSugg in the PBFNSetupDef structure.

## Info Codes Report Section

The Info Codes Report section provides details on the processing performed on your input address. The PBFNAddressDataDef structure is the source for the information in this section. To print the Info Codes Report on the Address Detail Report, specify "ON" for the "Address Detail Report Info" field in the Configuration File or "ON" for cAddrDtlRptInfo in the PBFNSetupDef structure.

## Match

The following table describes the fields in the Match section. If data is not found for a field, the field does not display on the report.

**Table 17: Info Codes Report – Match Section**

Field	Description
USERKEY	User key that identifies the address in the Address Detail Report.
General	<p>Provides general information on the address assignment process.</p> <p><b>Address Unchanged</b> Indicates whether changes were made to the input address.</p> <p><b>Dual Address</b> Indicates if the street lines include a dual address. A dual address is an address that contains more than one mailable address (for example, an address that contains both a PO BOX and a street address).</p> <p><b>Non Deliverable</b> Indicates if a street line includes an address that is not deliverable by the USPS. This is an address that has successfully coded to a valid entry on the USPS database, but the USPS has marked the address as being non-deliverable.</p> <p><b>Zip Move</b> Indicates whether an address was coded using a ZIPMOVE record.</p> <p><b>Exceptions</b> Indicates whether an exceptions table entry was used to process the address.</p> <p><b>PO Box Only ZIP</b> Indicates that the input address ZIP Code is a P.O. Box only delivery zone.</p>

Field	Description	
Address Match Level	Level of matching achieved for the address.	
	<b>Street</b>	Street level default.
	<b>Secondary</b>	Matched at secondary range level.
	<b>Firm Primary</b>	Firm record match at primary range level.
	<b>Firm Secondary</b>	Firm record match at secondary range level.
	<b>High Rise Default</b>	High-rise default.
	<b>High Rise Sec Match</b>	High-rise secondary range match.
	<b>RR or HC Default</b>	RR or HC default.
	<b>RR or HC Sec Match</b>	RR or HC box range match.
	<b>General Delivery</b>	General delivery.
	<b>PO Box</b>	PO Box.
	<b>Coded Unique ZIP</b>	Coded to a unique ZIP Code.
	<b>Coded Unique ZIP Default</b>	Coded as a default unique ZIP Code.
<b>Military Default</b>	Coded as a default military.	
<b>Military Sec Match</b>	Coded as a military secondary match.	
Range Match	Level of matching achieved for the range. Possible values for this field are described below.	
	<b>Original Primary Ranked</b>	Original primary range ranked.
	<b>Original Secondary #1</b>	Original secondary #1 ranked.
	<b>Original Secondary #2</b>	Original secondary #2 ranked.
	<b>sec1 + sec2 combo</b>	Secondary 1 plus secondary 2 combined for matching.
	<b>Alpha Recombo</b>	Alpha component of range was moved to secondary for matching.
	<b>Dash-Recombo</b>	Range following the dash was moved to secondary for matching.
	<b>Address A@D Retry</b>	Primary range moved to secondary for alternate at delivery matching.
	<b>Unit Designator Secondary</b>	Unit designator used as the secondary range for matching.
Address Return Level	Level of postal coding information returned.	
	<b>Valid ZIP 4</b>	Returned valid ZIP + 4 code.
	<b>Valid ZIP and Carrt</b>	Returned valid ZIP Code and carrier route only.
	<b>Valid ZIP Code</b>	Returned valid ZIP Code only.
	<b>Unknown</b>	Returned valid carrier route only.

Field	Description
Chars30Flag	<p>Reason for the changed label line.</p> <ul style="list-style-type: none"> <li>Label street name is Abbreviated Alias. Preferred Alias and Base address are &gt; 30 chars. <ul style="list-style-type: none"> <li>The length of the Abbreviated Alias address is less than or equal to 30.</li> <li>The length of the Preferred Alias address is greater than 30.</li> <li>The length of the Base address is greater than 30.</li> </ul> </li> <li>Label street name is Base address. Preferred Alias is &gt; 30 chars. <ul style="list-style-type: none"> <li>Length of the Base address is less than or equal to 30.</li> <li>Length of the Preferred Alias address is greater than 30.</li> </ul> </li> <li>Label street name is Base address. Abbreviated Alias is &gt; 30 chars. <ul style="list-style-type: none"> <li>The length of the Base address is less than or equal to 30.</li> <li>The length of the Abbreviated Alias address is greater than 30.</li> </ul> </li> <li>Label street name is Base address. Other Alias is &gt; 30 chars. <ul style="list-style-type: none"> <li>The length of the Base address is less than or equal to 30.</li> <li>The length of the Other alias address is greater than 30.</li> </ul> </li> </ul>
Input Alias	<p><b>Input address was an alias street name</b> Alias address present on input. If an alias address is not present on input, the Input Alias field does not display on the report.</p>
DPV Results	<p>Indicates the DPV processing results.</p> <p><b>N</b> DPV failed</p> <p><b>Y</b> DPV confirmed</p> <p><b>S</b> Primary range confirmed, Secondary range failed</p> <p><b>D</b> Primary range confirmed, Secondary range missing</p> <p>If a match is found to a DPV table, the matched table name (CMRA, DNA, NoStat, NSL, PBSA, Throwback, Vacant) also displays on this line. If matching to more than one DPV table, more than one table name displays on this line. For example:</p> <p>DPV Results: DPV confirmed; CMRA; Vacant</p>

## Locations

**Table 18: Info Codes Report – Locations Section**

Field	Description
Address	Location where Finalist found the address information.
Unit	Location of the matched unit returned by Finalist.
Firm	Location of the matched firm returned by Finalist.
PMB	Location of the matched Private Mail Box (PMB) or Mail Stop Code (MSC) designator returned by Finalist.

## Types

**Table 19: Info Codes Report – Types Section**

Field	Description
Address	Type of address.
Street	Type of street matched to the address. <b>Preferred Alias</b> Preferred alias street name. <b>Nickname Alias</b> Non-preferred alias street name. <b>Alternate at Delivery</b> Alternate at delivery street name. <b>Abbreviated Alias</b> Abbreviated alias street name.
Lookup	Type of lookup PBFNProcess performed on the last supplied address.
Failure	Type of failure resulting from the PBFNProcess look up.

Field	Description
City	Type of city matched to the address.
	<b>Single ZIP</b> Single ZIP Code city.
	<b>Single Zone</b> Single finance city (city with one finance area).
	<b>Multi Zone</b> Multi finance city (city with more than one finance area).
	<b>Primary</b> Primary city name. Not a non-mailing city name.
	<b>Non Mailing</b> Non-mailing city name.
	<b>Abbr Name</b> Abbreviated city name.
	<b>Preferred Name</b> Preferred city name.
	<b>Output Preferred City Name Override</b> Preferred city override name.
ZIP	Type of ZIP Code matched to the address.
	<b>Regular ZIP</b> Standard ZIP Code.
	<b>Puerto Rico ZIP</b> Puerto Rico ZIP Code.
	<b>Military</b> ZIP Code for a military installation.
	<b>Small Town</b> Small town ZIP Code.
	<b>Unique</b> Unique ZIP Code.
	<b>+4</b> Address matched within the unique ZIP Code and input plus4 was found.
	<b>No +4</b> Address line was matched within the unique ZIP Code but input plus4 was not found.
	<b>No Match +4</b> Address line was not matched within the unique ZIP Code but input plus4 was matched.
	<b>No Match No +4</b> Address line and input plus4 was not matched within the unique ZIP Code. Defaults returned.

## Components

**Table 20: Info Codes Report – Components Section**

Field	Description
Exact Match	Address components resulted in an exact match.
No Input	Address components not found on input.

Field	Description
Corrected	Address components corrected.
Return Data	Address components not found on input but valid data returned.
Invalid	Address components found on input but invalid.
Invalid Corrected	Address components found on input but invalid. Corrections made.
Multi Choice	Address components for which input data resulted in an ambiguous match.
Alpha Mismatch	Alpha value of range was invalid.
Range Overlap	Range failure due to overlapping ranges found on the USPS database.
Cardinal Failure	Failed due to a directional change error. Directionals available for user outside USPS approved change rules (Cardinal Failure Rules).

## Finalist Batch Report

The Finalist Batch Report displays all configured options and statistics generated at the time of the processing run. To print the Finalist Batch Report, specify "ON" for the "Batch Report" field in your pbfncfg configuration file or for the cBatchRptOpt member in the PBFNSetupDef structure.

### Finalist Batch Report - Page 1

```

Finalist Batch Report © YYYY PITNEY BOWES INC.
Page 1
Software Version                LOT Database
Release Ver.   : X.XX.XX.X.XX   Product           : LOT OPTION

Build Date       : MMM DD YYYY HH:MM:SS   Version           : X.XX.XX
Expire Date     : 99-99-9999   Copyright Date    : 99-99-9999

ZIP+4 Database   LACS Database
Version         : X.XX.XX.X.XX   Product           :
LACSLinktest

```



```

Copyright Date : 99-99-9999
Expire Date    : XX-XX-XXXX

City Database
Version        : X.XX.XX.X.XX
Copyright Date : 99-99-9999
Expire Date    : XX-XX-XXXX

xxxxxxx,xxxxxxx
EWS Database
Copyright Date : 99-99-9999

RDI Database
Copyright Date : 99-99-9999

PreciselyID Database
Copyright Date : 99-99-9999

Version        : 9999
Release Date   : 99-99-9999

System Information
Processor      : 8664
OS            : WIN64
OS Version    : X.X.XXXX

System ID(s)   :

DPV Database
Copyright Date : 99-99-9999

SuiteLink Database
Product       : test
Release Number : 9999
Release Date  : 99-99-9999
Expire Date   : 99-99-9999

```

If you are processing in a non-CASS certified mode, the Finalist Batch Report (Page 1) Software Version Expire Date field displays “Non-expiring/Non-CASS”.

```

Software Version
Release Ver.    : X.XX.XX.X.XX
Build Date     : MMM DD YYYY HH:MM:SS
Expire Date    : Non-expiring/Non-CASS

```

## Finalist Batch Report - Page 2

```

Finalist Batch Report © YYYY PITNEY BOWES INC.
Page 2
Processing Parameters
File Section:
City Filename:
D:\PitneyBowes\CycleM\BASE\city.dir
ZIP+4 Filename 1:
D:\PitneyBowes\CycleM\BASE\zip4us.dir
Configuration Filename:          pbfncfg
Config LOAD/NOLOAD:             LOAD
CASS Section:
Batch Report Filename:          BatchRpt.txt
3553 Report Filename:          cass3553.txt
CASS Company Name:
CASS Product Name:
CASS Product Version:
Z4 Change Certified Company Name:
Z4 Change Certified Product Name:

```

```

Z4 Change Certified Product Version:
LOT Certified Company Name:
LOT Certified Product Name:
LOT Certified Product Version:
DPC Certified Company Name:
DPC Certified Product Name:
DPC Certified Product Version:
CASS Mailer Name:
CASS Mailer Address:
CASS Mailer City Line:
CASS Configuration: AAA
Batch Report: ON
3553 Report: ON
Process Section:
CASS Flag: ON CASS Standardize Case:
U Assign Carrier Routes: ON Return Input Firm:
ON Assign Unassign Records: OFF No. Exception Tbl Entries:
OFF Assign Abbreviated City: OFF Process Firms:
ON Cache Size: 99 Beginning Frame Char:
! End Frame Char: ! Return DPBC:
ON Dual Address Switch: P Return Alias Street Name:
XP Assign LOT: ON Padded String Data:
OFF Process LOT Only: OFF All Street Matching:
ON R777 Deliverable: ON Convert Secondary to PMB
OFF

```

## Finalist Batch Report - Page 3

```

Finalist Batch Report © YYYY PITNEY BOWES INC.
Page 3
Product Section:
Exception Table Filename:
EWS Filename:
D:\PitneyBowes\CycleM\BASE\ews.txt
LOT Filename:
D:\PitneyBowes\CycleM\eLOT\ELOT.dir
DPV Filepath: D:\PitneyBowes\CycleM\DPV\
RDI Filepath: D:\PitneyBowes\CycleM\RDI\
LACSLink Filepath: D:\PitneyBowes\CycleM\LACS

```

```

SuiteLink Filepath:
D:\PitneyBowes\CycleM\suitelink
  Early Warning System:                ON
  Delivery Point Validation:           FLT
  DPV Shutdown Indicator:              W
  Delivery Point Validation Tie Break: ON
  DPV Vacant Table:                    ON
  DPV No-Stat Table:                   ON
  DPV PBSA Table:                      OFF
  DPV DNA Table:                       ON
  DPV Throwback Table:                 ON
  DPV NSL Table:                       ON
  DPV Buffer Size:                     L
  LACSLink:                            ON
  LACSLink Memory:                     L
  Residential Delivery Indicator:      OFF
  PreciselyID processing:               OFF
  Commercial Mail Validation:          ON
  SuiteLink:                           ON
  SuiteLink Small Memory Flag:         L
  SuiteLink Shutdown Indicator:        S
  Return SLK Input Secondary:          B
  Report Section:
  Report Filename:                      report.txt
  Report Title:                         Finalist Report File From
PBFN.CFG
  Detail Report Page Length:           0
  Detail Report Max Records:           0
  Detail Report Nth Records:           0
  I/O Report Option:                   OFF
  Isol Report Option:                  OFF
  Sugg Report Option:                  OFF
  Info Report Option:                  OFF
  Log Section:
  Log Filename:                        log.txt
  Log Level:                            3

```

## Finalist Batch Report - Page 4

```

Finalist Batch Report
Page 4
  Input Filename:                       inputfile.txt
  Batch Start Time:                     MMM DD YYYY HH:MM:SS
  Batch End Time:                       MMM DD YYYY HH:MM:SS
  Total Process Time:                   HH:MM:SS
  Oldest Process Date:                  0
  Input Address Quality
  Address Assignment Summary

```

```

Count          %
  Total Assigned
    Assigned without Correction:
252739         52.60
    Assigned with Correction:
170144         35.41
    No match or Uncorrectable:
 57641         12.00

=====
  Total Records Processed:
480524        100.00
  Processing Summary
Count          %
  Total Zip Codes Assigned:
469782         97.76
  Total Carrier Routes Assigned:
422883         88.00
  Total ZIP+4 Codes Assigned:
397202         82.66
  Total records with LACS indicator:
 8214          1.71
  Total LOT Assigned:
397202         82.66
  Total RDI Identified:
   0           0.00
  Total Valid But Undeliverable Addresses on AMS (4600):
 446           0.09
  Total ZIP+4 Suppressed due to DPV Failure (4601):
25235          5.25
  Total ZIP+4 Suppressed due to R777 Failure (4602):
 125           0.08

```

## Finalist Batch Report - Page 5

```

Finalist Batch Report
Page 5
  Carrier Routes Assigned To:
Count          %
  Highway Contracts:
 2559          0.53
  Post Office Boxes:
48078          10.01
  Rural Routes:
67344          14.01
  All Others:
304902         63.45

```

```

=====
      Total
422883      88.00
      ZIP+4 Codes Assigned To:
Count      %
      Streets:
253099      52.67
      PO Boxes:
46629       9.70
      High Rises:
94424       19.65
      Rural Routes:
785         0.16
      Firms:
1629        0.34
      General Delivery:
261         0.05
      Military:
375         0.08

=====
      Total
397202      82.66

```

## Finalist Batch Report - Page 6

```

Finalist Batch Report
Page 6
  Nonassignment Codes
  Address Parsing
Count      %
  4301 - No Street or Primary Name in Address:
281        0.06
  4399 - Blank Address Record:
9          0.00
  Address Retrieval
Count      %
  4411 - No Primary Name Found:
19151      3.99
  4412 - No Primary Names Ranked with Certainty:
16910      3.52
  4421 - Invalid Range or House Number:
11314      2.35
  4422 - Incorrect or Missing Directional:
1161       0.24
  4423 - Incorrect or Missing Suffix:
170        0.04
  4425 - Incorrect or Missing Suffix and Directional:

```

```

      8      0.00
4450 - Missing Range:
3826      0.80
4451 - Multiple Choice or Ambiguous Component Errors:
101      0.02
4460 - EWS Failure:
24      0.00
4461 - LOT Errors:
0      0.00
4465 - Firm Only Match Failures:
176      0.04
4466 - Secondary Only Match Failures:
0      0.00
4467 - ZIP Move Failures:
0      0.00
4500 - Other Errors:
0      0.00
City, State, ZIP Code Parsing
Count      %
4101 - No City, State, and ZIP Code in Address:
1      0.00
4102 - No ZIP Code and No City Name in Address:
9      0.00
4103 - No ZIP Code and No State Name in Address:
99      0.02
City, State, ZIP Code Retrieval
Count      %
4211 - Invalid ZIP Code and no City Name in Address:
0      0.00
4212 - No ZIP Code and Invalid City Name in Address:
2861      0.60
4213 - Invalid ZIP Code and Invalid City Name in Address:
0      0.00
Total Nonassignment Codes
58087      12.09

```

## Finalist Batch Report - Page 7

```

Finalist Batch Report
Page 7
DPV Processing Statistics
Count      %
Total Records Through DPV Processing
422437      87.91
Total DPV Y Type
348267      72.48
Total DPV S Type
18808      3.91

```

```

Total DPV D Type
30127      6.27
Total DPV Not Validated
25235      5.25
Total DPV Vacant Found
14777      3.08
Total DPV No-Stat Found
9132       1.90
Total DPV PBSA Found
0          0.00
Total DPV DNA Found
0          0.00
Total CMRA Found
1235       0.26
Total DPV Throwback Found
0          0.00
Total DPV NSL Found
0          0.00
LACSLink Processing Statistics
Count      %
Total Records Through LACSLink Processing
75448      15.70
Total LACS Street not found ( )
0          0.00
Total LACS Successful Conversion (A )
7515       1.56
Total No LACS Match (00)
67851      14.12
Total LACS Match Found But No Conversion (14)
13         0.00
Total LACS Match Found With Unit (92)
69         0.01
Total LACS Match High Rise Default (09)
0          0.00
SuiteLink Processing Statistics
Count      %
Total Records Through SuiteLink Processing
4952       1.03
Total Successful SuiteLink Matches
411        0.09
Total Successful SuiteLink Matches with Corrected Suites
411        0.09
PreciselyID Processing Statistics
Count      %
Total Records Through PreciselyID Processing
0          0.00
Total Total PreciselyID 'Y' Matches
0          0.00
Total Total PreciselyID 'Y' Matches
0          0.00

```

The Structure Field below is the "C" field name. Refer to Chapter 2, Structures and Constants in your Finalist Reference Guide for the COBOL equivalent.

**Table 21: Finalist Batch Report Field Descriptions**

Report Field	Structure Field	Description
Finalist Batch Report ©YYYY Precisely Inc. Page N	N/A	Report title and page number that displays at the top of each page.  For the z/OS platform, the copyright symbol displays as EBCDIC characters as shown here:  Finalist Batch Report (c)YYYY PITNEY BOWES INC.
Finalist Batch Report - Page 1 - PBFNInfoDef Fields		
Software Version Release Ver.	cCassEngineVersion	Current CASS version number found on the USPS 3553 Form (CASS Summary Report). This number is returned in a N.NN.NN.D format where "N.NN.NN" is the release number and "D" is the alpha CASS cycle character required and defined by USPS. The release numbers are assigned according to USPS CASS rules.
Build Date	EngineBuild	Internal number for the engine build. This number is returned in a "FNxx.yy.zz \$Revision n \$" format where "xx.yy.zz" is the Finalist release number and "n" is the version number.
Expire Date	cEngineExpDate	This is the CASS expiration date for the Finalist engine. This date represents the end of a CASS cycle year as declared by the USPS. This date is returned in a mmddyyyy format (two-digit month, two-digit day, and four-digit year).
ZIP+4 Database Version	cZip4BaseVer	Three-digit version number for the ZIP + 4 database. This number returns without punctuation (for example, database X.XX will return as XXX).
Copyright Date	cZip4BaseDate	This is the maintenance date for the USPS data used for this engine build. This date is returned in a mmyyyy format (two-digit month and four-digit year).
Expire Date	cZip4BaseExpDate	This is the CASS expiration date for the ZIP + 4 File. This date is calculated according to current USPS rules for this engine build. This date is returned in a mmddyyyy format (two-digit month, two-digit day, and four-digit year).



Report Field	Structure Field	Description
City Database Version	cCityBaseVer	This is the three-digit version number for the City database. This number returns without punctuation (for example, database X.XX will return as XXX).
Copyright Date	cCityBaseDate	This is the maintenance date for the USPS data used for this engine build. This date returns in a mmyyyy format (two-digit month and four-digit year).
Expire Date	cCityBaseExpDate	This is the CASS expiration date for the City File. This date is calculated according to current USPS rules for this engine build. This date returns in a mmddyyyy format (two-digit month, two-digit day, and four-digit year).
EWS Database Copyright Date	cEWSBaseDate	This is the maintenance data for the USPS data used for this engine build. This date is returned in an mm-dd-yyyy format.
RDI Database Copyright Date	cRDIBaseDate	This is the maintenance data for the USPS data used for this engine build. This date is returned in an mm-dd-yyyy format. (RDI does not have an actual date as part of the file so the ZIP+4 date is provided.)
PreciselyID Copyright Date	cPreciselyIDDate	Date the PreciselyID database was built (mm-dd-yyyy format).
LOT Database Product	cLOTProdName	This is the name of the Line of Travel (LOT) database file being used in the run.
Version	cLOTBaseVer	This is the version number for the LOT database. This number is returned without punctuation (for example, 2.00 returns as 200).
Copyright Date	cLOTBaseDate	This is the maintenance date for the USPS data. This date is returned in an mmyyyy format (two-digit month and four-digit year).
LACS Database Product	cLACSProdName	Certified LACS <sup>Link</sup> product name.
Version	cLACSVersion	LACS <sup>Link</sup> product version number.
Release Date	cLACSBaseDate	LACS <sup>Link</sup> database date in format yyyyymmdd.

Report Field	Structure Field	Description
System Information Processor	cProcessorType	Specifies the type of processor in the system when available.
OS	cOSType (PBFN-BINF-OS-TYPE)	Operating system based on program value (MVS, VMS, WIN32) or query of the operating system information (for example, Unix - uname).
OS Version	cOSVersion	The version of the operating system.
System ID(s)	CPUID	Displays the System ID found on this system.
DPV Database Copyright Date	cDPVBaseDate	This is the maintenance data for the USPS data used for this engine build. This date is returned in an mm-dd-yyyy format.
Suite <sup>Link</sup> Database Product	cSuiteLinkProdName	Certified Suite <sup>Link</sup> product name.
Release Number	cSuiteLinkVersion	Suite <sup>Link</sup> product version number.
Release Date	cSuiteLinkBaseDate	Suite <sup>Link</sup> database date in format ccyyymmdd.
Expire Date	cSuiteLinkBaseExpDate	Suite <sup>Link</sup> database expiration date in format yyyyymmdd.
Finalist Batch Report - Page 2 - PBFNSetupDef Fields		
City Filename	cCityFileName	City Database file name.
ZIP+4 Filename 1	cZip4FileName	ZIP + 4 Database file name.
Configuration Filename	cConfigFileName	Path to and file name of the Configuration File.
Config LOAD/NOLOAD	cLoadSetup	Identifies whether or not to load the setup structure from the configuration file.
Batch Report Filename	cStatFileNameSetup	Output file name for the Finalist Batch Report and the USPS Form 3553 (CASS Summary Report). This field contains the file name found in the pbfm.cfg. If you do not enter the output file name in the pbfm.cfg File, the default file name batch.txt will be used.

Report Field	Structure Field	Description
3553 Report Filename	c3553FileNameSetup	Output file name for the USPS Form 3553 (CASS Summary Report). This field contains the file name found in the pbfm.cfg file. If you do not enter the output file name in the pbfm.cfg File, the default file name cass3553.txt will be used.
CASS Company Name	cCASSCompName	Name of the CASS-certified company. This information displays in section A box 1 on the USPS Form 3553 (CASS Summary Report).
CASS Product Name	cCASSProdName	Name of the CASS-certified product. This information displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
CASS Product Version	cCASSProdVer	Version number for the CASS-certified product. This information displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
Z4 Change Certified Company Name	cZ4ChngCertCompName	Z4Change-certified company name.
Z4 Change Certified Product Name	cZ4ChngProdName	Z4Change product name.
Z4 Change Certified Product Version	cZ4ChngProdVer	Z4Change product version number.
LOT Certified Company Name	cLOTCertCompName	Line of Travel (LOT)-certified company name. This information displays in section A box 7 on the USPS Form 3553 (CASS Summary Report).
LOT Certified Product Name	cLOTCertProdName	Line of Travel (LOT)-certified-product name. This information displays in section A box 8 on the USPS Form 3553 (CASS Summary Report).
LOT Certified Product Version	cLOTCertProdVer	Version number for the Line of Travel (LOT)-certified product. This information displays in section A box 8 on the USPS Form 3553 (CASS Summary Report).
DPC Certified Company Name	cDPCCertCompName	Delivery Point Code certified-company name. This information displays in section A box 10 on the USPS Form 3553 (CASS Summary Report).
DPC Certified Product Name	cDPCCertProdName	Delivery Point Code-certified product name. This information displays in section A box 11 on the USPS Form 3553 (CASS Summary Report).

Report Field	Structure Field	Description
DPC Certified Product Version	cDPCCertProdVer	Version number for the Delivery Point Code-certified product. This information displays in section A box 11 on the USPS Form 3553 (CASS Summary Report).
CASS Mailer Name	cMailerName	CASS-certified mailer name.
CASS Mailer Address	cMailerAddress	CASS-certified mailer address.
CASS Mailer City Line	cMailerCityLine	CASS-certified mailer city line.
CASS Configuration	cCASSConfiguration	CASS configuration value. This information displays in section A box 3 on the USPS Form 3553 (CASS Summary Report).
Batch Report	cBatchRptOpt	Indicates whether to print the Finalist Batch Report.
3553 Report	c3553RptOpt	Indicates whether to print the USPS Form 3553 (CASS Summary Report).
CASS Flag	cCASSFlag	<p>Indicates whether Finalist processes in non-CASS or CASS mode. For batch processing, cCASSFlag=ON ensures that CASS-required options are turned on including DPV, LACS<sup>Link</sup>, Suite<sup>Link</sup>, CASS configuration, Carrier Route (CR), and Delivery Point (DPBC). For DPV, the Flat file option is used when no other DPV option is specified.</p> <p><b>Note:</b> If CASS Flag = ON and a conflicting option is encountered (Configuration, Assign CR, Return DPBC, LACS<sup>Link</sup>=OFF, Suite<sup>Link</sup>=OFF, or DPV=OFF), a warning message is written to the log file indicating that CASS has been forced off and the reason for CASS being forced off. The message is similar to:</p> <p>Warning Message; CASS forced off: CASS Configuration, Return DPBC, Assign CR, Assign SuiteLink, Assign LACSLink, Assign DPV</p>
Assign Carrier Routes	cAssignCR	Indicates whether Finalist assigns carrier route codes to your file.

Report Field	Structure Field	Description
Assign Unassign Records	cProcessUnassigned	This field indicates how to code previously unassigned or expired addresses. Specifically, the member indicates whether only previously unassigned or expired addresses should be postal coded in batch mode or whether all supplied address records should be coded.
Assign Abbreviated City	cAssignAbbrevCity	Indicates whether Finalist returns abbreviated city names.
Cache Size	cCacheSize	Indicates whether Finalist should use internal caching.
End Frame Char	cEndFrame	Indicates the character to use for the end framing character for advanced barcodes. If you do not specify a value here, the BegFrame character is used. If you do not specify a character for BegFrame or EndFrame, Finalist uses the exclamation point (!).
Dual Address Switch	cDualAddrSwit	If the input file contains dual addresses (one a conventional address, a second containing a PO Box address), this field determines what order to use to process and match the addresses. If the selected address is valid, processing stops. If the selected address does not validate, Finalist will attempt to code the secondary address.
Assign LOT	cAssignLOT	Indicates whether Finalist assigns Line of Travel (LOT) codes.
Process LOT Only	cProcessLOTOnly	Determines whether Finalist performs only eLOT processing.
R777 Deliverable	cR777Deliverable	Determines whether Finalist performs Carrier Route code R777 processing.
CASS Standardize Case	cStandardCase	Indicates whether to return addresses in all upper, all lower, or mixed upper and lower case. This field affects the address coding output only. It does not affect the database APIs (APIs prefixed with PBCS). The database APIs always return upper case data.
Return Input Firm	cRetInputFirm	This field determines whether Finalist returns the input firm.

Report Field	Structure Field	Description
No. Exception Tbl Entries	cMaxExcpTblEntries	This field determines the maximum number of exception table entries.
Process Firms	cProcessFirms	Determines whether Finalist performs firm processing.
Beginning Frame Char	cBegFrame	Indicates the character to use for the front framing character for advanced barcodes. If you do not specify a character for BegFrame or EndFrame, Finalist uses the exclamation point (!).
Return DPBC	cRetDPBC	Indicates whether Finalist returns delivery point barcodes.
Return Alias Street Name	cRetAliStName	This field indicates whether Finalist returns alias street names in the label line. An alias street name is an alternate name for a street, maintained at the ranged Plus4 ZIP Code level.
Padded String Data	cPaddedStringData	If you are using a platform or coding language (for example, COBOL) that does not support null terminated strings, you will need to pad all fields with blanks. This field indicates whether you will be using input/output fields that are padded with blanks or null terminated.
All Street Matching	cAllStreetMatching	Determines whether Finalist performs All Street Matching (ASM) processing.
Convert Secondary to PMB	cConvertSecPMB	Determines whether Finalist converts input address secondary information to "PMB" under specific conditions.
Finalist Batch Report - Page 3 - PBFNSetupDef Fields		
Exception Table Filename	cExceptionTblFileName	Exceptions Table file name and path.
EWS Filename	cEWSFileName	Early Warning System (EWS) file name and path.
LOT Filename	cLOTFileName	Line of Travel (LOT) file name and path.
DPV Filepath	cDPVFilePath	Delivery Point Validation (DPV) File path.
RDI Filepath	cRDIFilePath	Residential Delivery Indicator (RDI) File path.

Report Field	Structure Field	Description
LACS <sup>Link</sup> Filepath	cLACSLinkFilePath	LACS <sup>Link</sup> File path.
Suite <sup>Link</sup> Filepath	cSuiteLinkFilePath	Suite <sup>Link</sup> File path.
Early Warning System	cAssignEWS	This field indicates whether Finalist will perform Early Warning System (EWS) processing.
Delivery Point Validation	cAssignDPV	Indicates whether Finalist performs Delivery Point Validation (DPV) processing.  <b>Note:</b> The USPS CASS regulations require Delivery Point Validation (DPV) processing to generate the USPS Form 3553 (USPS CASS Summary Report).
DPV Shutdown Indicator	cDPVShutdownIndicator	Action to take when encountering a DPV Seed during the processing run.
Delivery Point Validation Tie Break	cAssignDPVTie	Indicates whether to use DPV tie break processing.  <b>Note:</b> The USPS CASS regulations require DPV Tie Break processing to generate the USPS Form 3553 (USPS CASS Summary Report).
DPV Vacant Table	cAssignDPVVacant	Indicates whether Finalist uses the Vacant Table and returns the proper Vacant code. Also indicates whether Finalist loads the Vacant Table into memory or uses the Vacant Table outside of memory.
DPV No-Stat Table	cAssignDPVNoStat	Indicates whether Finalist uses the No-Stat Table and returns the proper No-Stat code. Also indicates whether Finalist loads the No-Stat Table into memory or uses the No-Stat Table outside of memory.
DPV PBSA Table	cAssignDPVPBSA	Indicates whether Finalist uses the PBSA Table and returns the proper PBSA code. Also indicates whether Finalist loads the PBSA Table into memory or uses the PBSA Table outside of memory.

Report Field	Structure Field	Description
DPV DNA Table	cAssignDPVDNA	Indicates whether Finalist uses the Door Not Accessible (DNA) Table and returns the proper DNA code. Also indicates whether Finalist loads the DNA Table into memory or uses the DNA Table outside of memory.
DPV Throwback Table	cAssignDPVTHRWBK	Indicates whether Finalist uses the DPV Throwback Table and returns the proper Throwback code. Also indicates whether Finalist loads the DPV Throwback Table into memory or uses the DPV Throwback Table outside of memory.
DPV NSL Table	cAssignDPVNSL	Indicates whether Finalist uses the DPV No Secure Location (NSL) Table and returns the proper NSL code. Also indicates whether Finalist loads the NSL Table into memory or uses the NSL Table outside of memory.
DPV Buffer Size	cDPVBufSize	Indicates the memory model for DPV processing.
LACS <sup>Link</sup>	cAssignLACSLink	This field determines whether Finalist performs LACS <sup>Link</sup> processing.  <b>Note:</b> The USPS CASS regulations require LACS <sup>Link</sup> processing. If you do not perform LACS <sup>Link</sup> processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report).
LACS <sup>Link</sup> Memory	cLACSLinkProcessing	Indicates the memory model for LACS <sup>Link</sup> processing:
Residential Delivery Indicator	cAssignRDI	This field indicates whether Finalist will perform Residential Delivery Indicator (RDI) processing.
PreciselyID processing	cAssignPreciselyID	This field indicates whether Finalist will perform PreciselyID processing.
Commercial Mail Validation	cAssignCMRA	This field indicates whether Finalist will perform CMRA processing. Also indicates whether Finalist loads the CMRA Table into memory or uses the CMRA Table outside of memory.
Suite <sup>Link</sup>	cAssignSuiteLink	This field indicates whether Finalist will perform Suite <sup>Link</sup> processing.



Report Field	Structure Field	Description
Suite <sup>Link</sup> Small Memory Flag	cSuiteLinkSmallMem	Indicates the memory model for Suite <sup>Link</sup> processing:
Suite <sup>Link</sup> Shutdown Indicator	cSuiteLinkShutdown	Action to take when encountering a Suite <sup>Link</sup> processing error during the processing run.
Return SLK Input Secondary	cRetSLKinputSecdry	Action to take when an input secondary is provided that does not match the Suite <sup>Link</sup> returned information.
Report Filename	cRptFileName	Address Detail Report output file name.
Report Title	cRptTitle	Report title for the Address Detail Report.
Detail Report Page Length	IAddrDtlRptIsolPageLen	Maximum page length for the Address Detail Isolation Report.
Detail Report Max Records	IAddrDtlRptIsolMaxRec	Maximum number of records to output to the Address Detail Isolation Report.
Detail Report Nth Records	IAddrDtlRptIsolNthRec	This field indicates the records to print on the Address Detail Isolation Report (for example., every 8th record).
I/O Report Option	cAddrDtlRptType	Indicates whether to print the Address Detail Report.
Isol Report Option	cAddrDtlRptIsol	Indicates whether to print the Address Detail Isolation Report.
Sugg Report Option	cAddrDtlRptSugg	Indicates whether to print the Address Detail Suggestion Report.
Info Report Option	cAddrDtlRptInfo	This field indicates whether to print the Address Detail Information Report.
Log Filename	cLogFileNames	Log File Name.
Log Level	cLogLevel	This field indicates the level of the message logged.
Finalist Batch Report - Page 4 - PBFNStatsDef Fields		

Report Field	Structure Field	Description
Input Filename	ListFileName	ListFileName field of the PBFNReportData structure.
Batch Start Time	IStartTime	The IStartTime is initialized when PBFNInit function is called and when PBFNStats is called with PBFNReset set to TRUE. Time equals the seconds elapsed since midnight (00:00:00), December 31, 1899, Universal Coordinated Time, according to the system clock.
Batch End Time	IStopTime	The value is set when the PBFNTerminate function is called. Time is the seconds elapsed since midnight (00:00:00), December 31, 1899, Universal Coordinated Time, according to the system clock.
Total Process Time	IElapsedTime	Seconds elapsed since first address assignment and the most recent address assignment.
Oldest Process Date	IOldProcDate	If the Assign Unassign Records value is set to a value other than OFF in either the PBFNSetupDef structure or the PBFN.CFG file, this value is the oldest process date (cProcessDate) encountered. Otherwise it contains a 0 (zero).
Assigned without Correction	ITotalAssigned minus ITotalCorrected	Total address records assigned without correction.
Assigned with Correction	ITotalCorrected	Total address records corrected.
No match or Uncorrectable	ITotalNotAssigned	Total number of records that did not code.
Total Records Processed	ITotalProcessed	Total address records processed or presented to PBFNProcess.
Total Zip Codes Assigned	ITotalZipsAssigned	Total ZIP Codes assigned.
Total Carrier Routes Assigned	ITotalCrtrt	Total number of carrier routes assigned.
Total ZIP+4 Codes Assigned	ITotalZ4Assigned	Total ZIP + 4 Codes assigned.
Total records with LACS indicator	ITotalLacs	Total number of records assigned a LACS indicator.

Report Field	Structure Field	Description
Total LOT Assigned	ITotalLOTAssigned	Total number of records assigned Line of Travel (LOT) codes.
Total RDI Identified	ITotalRDI	Total number of records assigned a Residential Delivery Indicator (RDI).
Total Valid But Undeliverable Addresses on AMS (4600)	ITotalNonDeliverable	Total number of undeliverable addresses.
Total ZIP+4 Suppressed due to DPV Failure (4601)	cNonDeliverableInd (PBFNAddressDataDef)	Total number of addresses that failed Delivery Point Validation (DPV) processing.
Total ZIP+4 Suppressed due to R777 Failure (4602)	IR7 (PBFNDPVStatsDef)	Total number of addresses where ZIP + 4 Code was not returned because address matched to an R777 record.
Finalist Batch Report - Page 5 - PBFNStatsDef Fields		
Without Verification	ICityStateZipMatches	Total number of records matching on city, state, and ZIP Code when the option Assign Unassign Records is requested.
From City and State	ICityStateMatches	Total number of records matching on city and ZIP Code.
Ignoring Bad City	IStateAndZipMatches + IStateOrZipMatches	Total number of records with no City but State and/or ZIP Code were presented.
Total	ILineItemTotal + ICityStateMatches + IStateOrZipMatches + IStateAndZipMatches	Category total.
Highway Contracts	IToCrrtHC	Total number of Highway Contract carrier routes.
Post Office Boxes	IToCrrtBox	Total number of Post Office Box carrier routes.
Rural Routes	IToCrrtRR	Total number of Rural Route carrier routes.
All Others	IToCrrtNorm	Total number of default carrier routes, including street, general delivery and postmaster.
Total	IToCrrtHC + IToCrrtBox + IToCrrtRR + IToCrrtNorm	Category total.

Report Field	Structure Field	Description
ZIP+4 Codes Assigned To Streets	ITotalStreetLevel	Total number of ZIP + 4 Codes assigned to streets.
PO Boxes	ITotalPOBox	Total number of ZIP + 4 Codes assigned to Post Office boxes.
High Rises	ITotalHR	Total number of high rises including defaults and secondary records.
Rural Routes	ITotalRRHC	Total number of ZIP + 4 Codes assigned to Rural Route and Highway Contract addresses.
Firms	ITotalFirm	Total number of ZIP + 4 Codes assigned to firms.
General Delivery	ITotalGenDelPostMaster	Total number of ZIP + 4 Codes assigned to general delivery.
Military	ITotalMilitary	Total number of ZIP + 4 Codes assigned to military ZIP Codes.
Finalist Batch Report - Page 6 - PBFNStatsDef Fields		
4301 - No Street or Primary Name in Address	INoStreetDataFound	Total number of address records without street data.
4399 - Blank Address Record	IBlankAddressLineData	Total number of address records with a blank address line.
4411 - No Primary Name Found	IStreetNameNotFound	Total number of address records where the street name was not found.
4412 - No Primary Names Ranked with Certainty	IStreetSuggOnly	Total records failed with suggested street records.
4421 - Invalid Range or House Number	IInvalidRangeForStreet	Total number of address records containing an invalid range for the street name.
4422 - Incorrect or Missing Directional	IDirectionalError	Total number of address records missing directionals or containing incorrect directionals.
4423 - Incorrect or Missing Suffix	ISuffixError	Total number of address records missing suffixes or containing incorrect suffixes.

Report Field	Structure Field	Description
4425 - Incorrect or Missing Suffix and Directional	ISuffixDirError	The total number of address records missing both the suffix and directional or containing incorrect suffix and directional.
4450 - Missing Range	INoRangeProvided	Total number of address records without a range.
4451 - Multiple Choice or Ambiguous Component Errors	IMultipleChoiceFailure	Total number of addresses that failed due to multiple choices for one or more address components. Each address component will be marked as multiple choice in the PBFNAddressDataDef structure (for example, duplicate street names, two or more directionals available with none on input, etc.).
4460 - EWS Failure	IEWSError	Total number of address records not matched because the records were found in the USPS Early Warning System (EWS) File.
4461 - LOT Errors	ILotFailure	Total number of address records where LOT assignment failed. Address coded successfully but LOT code was not assigned.
4465 - Firm Only Match Failures	IFirmFail	Total number of records failed where the address required a firm. No firm was provided or the firm failed to match.
4466 - Secondary Only Match Failures	ISecFail	Total number of records failed where the address required secondary addresses (there is no default street address). No secondary was provided or the secondary did not match.
4467 - ZIP Move Failures	IZipMove	Total number of records failed where the address coded but was flagged for ZIP Move processing. The address failed to meet the final ZIP Move criteria. ZIP Move processing requires an exact match of street, suffix, and directional (both pre and post).
4500 - Other Errors	IUnknownFailure	Total number of records failed without a specific reason.
4101 - No City, State, and ZIP Code in Address	IBlackCityLineData	Total number of records without a city, state, or ZIP Code.
4102 - No ZIP Code and No City Name in Address	INoCityZipData	Total number of records without city and ZIP Code.

Report Field	Structure Field	Description
4103 - No ZIP Code and No State Name in Address	INoStateZipData	Total number of records without state or ZIP Code.
4211 - Invalid ZIP Code and no City Name in Address	IBadZipNoCity	Total number of records with an invalid ZIP Code and no City provided.
4212 - No ZIP Code and Invalid City Name in Address	IBadCityNoZip	Total number of records with an invalid City and no ZIP Code provided.
4213 - Invalid ZIP Code and Invalid City Name in Address	IBadZipBadCity	Total number of records with an invalid ZIP Code and an invalid city.
Total Nonassignment Codes	Total of all of the 4xxx values.	Category total.
Finalist Batch Report - Page 7		
DPV Fields - PBFNDPVStatsDef Structure		
Total Records Through DPV Processing	IDPVInError + IDPVTypeYValid + IDPVTypeSValid + IDPVTypeDValid	Number of addresses that were processed through DPV.
Total DPV Y Type	IDPVTypeYValid	Total records with delivery point validated.
Total DPV S Type	IDPVTypeSValid	Total records with valid primary range but secondary range is not confirmed.
Total DPV D Type	IDPVTypeDValid	Total records with a valid primary range but secondary range is missing.
Total DPV Not Validated	IDPVInError	Total records not delivery point validated.
Total DPV Vacant Found	IDPVPvacantFound	Total records that matched to the Vacant table.
Total DPV No-Stat Found	IDPVNoStatFound	Total records that matched to the No-Stat table.
Total DPV PBSA Found	IPB	Total records that matched to the PBSA table.
Total DPV DNA Found	IDPVDNAFound	Total records that matched to the Door Not Accessible (DNA) table.
Total CMRA Found	ICMRAValid	Total records that matched to the CMRA table.

Report Field	Structure Field	Description
Total DPV Throwback Found	IDPVTHRWBKFound	Total records that matched to the P.O. Box Throwback table.
Total DPV NSL Found	IDPVNSLFound	Total records that matched to the No Secure Location (NSL) table.
LACSLink Fields - PBFNRtnLACSSStatsDef Structure		
Total Records Through LACSLink Processing	ITotalProcessed	Total number of records passed through LACSLink.
Total LACS Street not found ( )	ITotalProcessed - ITotalAMatches - ITotal00Matches - ITotal14Matches - ITotal92Matches - ITotal09Matches	Total number of records passed to LACSLink but did not receive an assignment.
Total LACS Successful Conversion (A )	ITotalAMatches	Total number of LACSLink matches with an assignment code of A.
Total No LACS Match (00)	ITotal00Matches	Total number of LACSLink matches with an assignment code of 00.
Total LACS Match Found But No Conversion (14)	ITotal14Matches	Total number of LACSLink matches with an assignment code of 14.
Total LACS Match Found With Unit (92)	ITotal92Matches	Total number of LACSLink matches with an assignment code of 92.
Total LACS Match High Rise Default (09)	ITotal09Matches	Total number of LACSLink matches with an assignment code of 09.
SuiteLink Fields - PBFNRtnSuiteLinkStatsDef Structure		
Total Records Through SuiteLink Processing	ITotalProcessed	Total number of records processed through SuiteLink.
Total Successful SuiteLink Matches	ITotalSuccessMatches	Total number of records successfully matched by SuiteLink.
Total Successful SuiteLink Matches with Corrected Suites	ITotalCorrectedSuite	Total number of records processed by SuiteLink with suites corrected.

Report Field	Structure Field	Description
Total Records Through PreciselyID Processing	IPreciselyIDAttempt	Total number of records processed through PreciselyID.
Total PreciselyID 'Y' Matches	IPreciselyIDY	Total number of PreciselyID matches where PreciselyIDFound=Y (full address match).
Total PreciselyID 'D' Matches	IPreciselyIDD	Total number of PreciselyID matches where PreciselyIDFound=D (secondary range information dropped to find PreciselyID).

## USPS Form 3553 (CASS Summary Report)

With each mailing submitted at an automation-based rate, a mailer must submit a completed USPS Form 3553 (CASS Summary Report). You can submit the original USPS Form or the facsimile generated by your program. Finalist produces a facsimile of USPS Form 3553 (CASS Summary Report) and partially completes it for you. For additional information on the USPS Form 3553 (CASS Summary Report), see the USPS Domestic Mail Manual (DMM).

```

      CASS SUMMARY REPORT
== A. SOFTWARE
=====
===== CASS - A1
=====
1. CASS CERTIFIED COMPANY NAME      | 2. CASS CERT. SOFTWARE NAME & VER
   | 3.CFG                            |
Precisely                          |Finalist                X.XX.XX.X | AAA
-----
4. Z4CHANGE CERTIFIED COMPANY NAME | 5. Z4CHANGE CERT.SOFTWARE NAME & VER
   | 6.CFG                            |
   |                                  |
-----
7. eLOT CERTIFIED COMPANY NAME      | 8. eLOT CERT. SOFTWARE NAME & VER
   | 9.CFG                            |
Precisely                          |Finalist                X.XX.XX.X | AAA
= B. LIST
=====
1. LIST PROCESSOR'S NAME            | 2. DATE LIST PROCESSED | 3. DATE OF DATABASE
   USED                              |
XX/XX/XXXX                          | MASTER FILE: XX/XX/XXXX | ZIP+4 FILE:
                                     | Z4CHANGE:                | Z4CHANGE:

```



```

                                |eLOT:          XX/XX/XXXX|eLOT:
XX/XX/XXXX
                                |CRIS:          |CRIS:
-----|-----
4. ADDRESS LIST NAME OR ID# |5. NUMBER OF LISTS |6. TOTAL RECORDS
SUBMITTED
INPUTFILE.TXT              |          1 |          1234567
== C. OUTPUT
=====
OUTPUT RATING              |VALIDATION PERIOD |OUTPUT RATING              |VALIDATION
PERIOD
TOTAL CODED|FROM      TO      |          TOTAL CODED|FROM
TO
-----|-----|-----|-----
a. ZIP+4/DPV CONFIRMED |XX/XX/XXXX          |c. 5-DIGIT CODED |XX/XX/XXXX
          1234567 |          XX/XX/XXXX|          1234567 |
XX/XX/XXXX
-----|-----|-----|-----
b. Z4CHANGE PROCESSED |////////////////////|d. CR RT CODED    |XX/XX/XXXX
          0 |////////////////////|          1234567 |
XX/XX/XXXX
-----|-----|-----|-----
////////////////////|////////////////////|e. eLOT ASSIGNED |XX/XX/XXXX
////////////////////|////////////////////|          1234567 |
XX/XX/XXXX
== D. MAILER
=====
I CERTIFY THAT THE MAILING |3. NAME & ADDRESS OF MAILER:
SUBMITTED WITH THIS FORM
HAS BEEN CODED (AS INDICATED ABOVE)
USING CASS-CERTIFIED SOFTWARE
MEETING ALL OF THE REQUIREMENTS
LISTED IN THE DMM SECTION 708.
=====
1. MAILER'S SIGNATURE |2. DATE
SIGNED
=====
== E. QUALITATIVE STATISTICAL SUMMARY (QSS)
=====
HR DEFAULT | HR EXACT | RR DFLT | RR EXACT | LACSLINK | EWS |
SUITELINK
        XXXXX|        XXXXX|        XX|        XXXX|        XXXX|        XXX|
        XXX
=====
PS Form 3553, September 2017

```

Complete the CASS Summary Report as follows:

1. Verify that the user of your application PRINTS in all the blanks indicated below, except in "SIGNATURE OF MAILER."

2. If this form represents a single list, check SINGLE LIST. If this form represents multiple lists, check MULTIPLE LISTS.
3. Verify that all fields in the A1, CASS section are completed.
4. If not already completed, print the company name in the B1, LIST PROCESSOR'S NAME section. This indicates that the company is the USPS customer using the Finalist program to process the list(s) for this mailing.
5. The Finalist program will complete the B2-B3, DATE LIST PROCESSED and DATE OF ZIP+4 DATA BASE report sections.
6. Complete the B4, ADDR LIST NAME OR ID# sections only if the report represents a single list. If the report represents a single list, print the address list's name or identification number. If you use an ID#, begin the number with "ID#".
7. If this report represents a single list, print the number "1" in the B5, NUMBER OF LISTS section. If this report represents multiple lists, print the number of lists the application processed for this mailing.
8. Your program completes sections B6, TOTAL RECORDS SUBMITTED.
9. Complete all of section C, OUTPUT.
10. The person at the company who is responsible for the mailing should sign his/her name in the D1, SIGNATURE section.
11. The person completing the report should enter the date he or she signed the form in the section D2 DATE.
12. If not supplied in the pbfm.cfg, the person completing the report should print his or her name and the company's address in the D3, NAME & ADDRESS OF MAILER section. The name printed here must match the name appearing in section D1.
13. Finalist completes the QUALITATIVE STATISTICAL SUMMARY section. Refer to Appendix 1 in the CASS Technical Guide for complete description of this section.

# 7 - Finalist Error Codes

## In this section

---

Error Codes (Returned Strings).....	180
Finalist Batch Driver Return Codes.....	182
DPV Error Messages.....	184
LACS <sup>Link</sup> Error Messages.....	185
Suite <sup>Link</sup> Error Messages.....	186
EWS Error Messages.....	186
eLOT Error Message.....	187



## Error Codes (Returned Strings)

This section describes error codes that may appear in the error field (cError) of the PBFNAddressDataDef structure. If an error occurs while processing, the function updates the PBFNAddressDataDef structure with detailed error information.

**Table 22: Error Codes (Returned Strings)**

Error Code	Description
4101	No city, state, and ZIP Code in address.
4102	No ZIP Code and no city name in address.
4103	No ZIP Code and no state name in address.
4104	Cannot match to Unique ZIP Code.
4211	Invalid ZIP Code and no city name in address.
4212	No ZIP Code and invalid city name in address.
4213	Invalid ZIP Code and invalid city name in address.
4301	No street name in input address.
4399	Blank address record.
4411	No primary street name found in Finalist database.
4412	No primary names ranked with certainty. Suggestions may be available.
4421	Invalid range or house number.
4422	Incorrect or missing directional.
4423	Incorrect or missing suffix.
4425	Incorrect or missing suffix and directional.

Error Code	Description
4450	No range in input address.
4451	Multiple component failure. An address component had multiple options causing the address to fail assignment. See field information in the PBFNAddressDataDef structure.
4460	EWS Failure. Address found on EWS table.
4461	LOT assignment has failed. Address coded successfully but LOT code not assigned.
4465	Address requires a firm. No firm was provided or the firm failed to match.
4466	Address requires secondary addresses (there is no default street address). No secondary was provided or the secondary did not match.
4467	Address coded but was flagged for ZIP Move processing. The address failed to meet the final ZIP Move criteria. ZIP Move processing requires an exact match of street, suffix, and directional (both pre and post).
4500	Unable to code. If reason for inability to code cannot be determined, this error is issued.
4600	Undeliverable address in Finalist database.  <b>Note:</b> This error code can still be valid with a return code of PBFN_SUCCESS. The combination of the two codes indicates that this is a valid address just not deliverable by USPS standards.
4601	Address failed Delivery Point Validation (DPV) processing. The PBFNAddressDataDef structure cNonDeliverableInd flag is set to "Y" (Address is a nondeliverable address).
4602	Address flagged as Carrier Route R777 is not eligible for street delivery. This error code is only generated if the option R777 Deliverable = OFF.
4801	Address is locked and will not be processed.
5101	Warning: Missing apt/suite number.
5102	Warning: Apt/suite was input but is not valid.
5103	Warning: Input firm name is missing or invalid.
5104	Warning: Multiple firms returned for address.

Error Code	Description
5105	Warning: Box number is invalid or unavailable.
5106	Warning: Apt/Suite was input but not allowed for this address.
5200	Information: Address bypassed counted correct in Process Unassign run.

## Finalist Batch Driver Return Codes

The following table describes return codes you may encounter while processing with the Finalist batch driver.

**Table 23: Finalist Batch Driver Return Codes**

Return Code	Message	Description
0	processComplete	Processing completed without errors.
1	logFileNameTooLong	Filename specified by the -L option on the command line was more than 256 characters long.
2	configFileNameTooLong	Filename specified by the -G option on the command line was more than 256 characters long.
3	jobFileNameTooLong	Filename specified by the -J option (or defaulted as the first command line parameter) was more than 256 characters long.
4	noJobFileDefined	No job file name was specified. Either the command line parameters were blank or no -J option was specified.
5	pbfnInitFailed	Finalist initialize process failed. Refer to the log.txt (SYSOUT DD on the mainframe) for details.
6	pbfnTermFailed	Finalist termination process failed. Refer to the log.txt (SYSOUT DD on the mainframe) for details.
7	pbfnInfoFailed	Finalist information call process failed. Refer to the log.txt (SYSOUT DD on the mainframe) for details.

Return Code	Message	Description
8	loadJobFailed	Finalist JOB file could not be processed. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
9	overlayNotOkay	The Finalist JOB requested OVERLAY processing but this was not confirmed during the run.
10	cantOpenLogFile	Finalist LOG file could not be opened. Refer to stderr (SYSPRINT DD on the mainframe) for details.
11	cantOpenJobFile	Finalist JOB file could not be opened. Refer to stderr (SYSPRINT DD on the mainframe) for details.
12	loadDbaseDefsFailed	The Finalist access of the Dbase file did not match the DEF file. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
13	loadDefsFailed	The Finalist DEF file could not be processed. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
14	delimitedFieldOverlap	The fields defined in the DEF files overlap each other. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
15	jobFileContentsBad	An error occurred processing the JOB options specified. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
16	addrLineInitFailed	An error occurred during initialization processing. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
17	processFailed	An error occurred during the Finalist PBFNProcess call. Refer to the Finalist run log file for details (Log Filename in the pbfncfg (PBFNCFG) file).
18	cantWriteLog	The Finalist batch driver log file could not be opened. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) or stderr for details.
19	pbfncfgResetStatsFailed	An error occurred during a call to PBFNStats. Refer to the Finalist log file (Log Filename in the pbfncfg (PBFNCFG) file).
20	initAddrFailed	An error occurred during initialization processing. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.

Return Code	Message	Description
21	invalidFileType	An invalid file type was specified in the JOB file. Refer to the Finalist batch log file (JOBLOG DD on the mainframe) for details.
22	cantOpenSeedLogFile	The Finalist engine could not open the log file to report a DPV or LACS <sup>Link</sup> seed violation. Refer to the Finalist run log file for details (Log Filename in the pbfncfg (PBFNCFG) file).

## DPV Error Messages

The following table describes the Delivery Point Validation (DPV) Option error messages.

**Table 24: Delivery Point Validation (DPV) Option Error Messages**

Error Message	Description
00005 - DPV Processing Error Problem opening XXXXXXXX file	Cannot open DPV file. Verify path or DDNAME.
20100 - DPV Processing Error Call Product tech support Case 0 DPV0	Cannot open the security file.
20101 - DPV Processing Error Call Product tech support Case 0 DPV1	Cannot read the security file.
20111 - DPV Processing Error Call Product tech support Terminate Case 1 DPV1	Cannot open the security file.
20112 - DPV Processing Error Call Product tech support Terminate Case 1 DPV2	Cannot read the security file.
20113 - DPV Processing Error Call Product tech support Terminate Case 1 DPV3	Cannot open the security file for write processing.
20120 - DPV Processing Error File is not same month (yymm) as ZIP+4 file (yymm)	The date of the Delivery Point Validation (DPV) Option data does not match the date of the Zip4us.dir File. Each month of DPV data corresponds to a specific month of ZIP + 4 data (for example, the September DPV data must be processed with the September ZIP + 4 data).



Error Message	Description
20121 - DPV Processing Error Call Product tech support Case 2 DPV1	Invalid security file - internal date.
20122 - DPV Processing Error Call Product tech support Case 2 DPV2	Invalid security file - internal check digit.
20123 - DPV Processing Error Call Product tech support Case 2 DPV3	Invalid security file - internal check digit.
20124 - DPV Processing Error Call Product tech support Case 2 DPV4	Seed violation occurred. Send data to the USPS and obtain a new key from the Precisely web site at <a href="http://www.g1.com/support">www.g1.com/support</a> .
20125 - DPV Processing Error Call Product tech support Case 2 DPV5	Cannot open the security file for write processing.
20126 - DPV Processing Error Call Product tech support Case 2 DPV6	Invalid security file - sequence number.
20127 - DPV Processing Error Call Product tech support Case 2 DPV7	Invalid security file - unknown error.

## LACS<sup>Link</sup> Error Messages

The LACS<sup>Link</sup> error messages are listed below. For more information, refer to the PBFNExtendedErrorDef structure.

**Table 25: LACS<sup>Link</sup> Option Error Messages**

Error Message	Description
PCLERR_LACSINITERROR (20150)	System errors at LACS <sup>Link</sup>
PCLERR_LACSINITFAIL (20151)	Occurred during LACS <sup>Link</sup> initialization
PCLERR_LACSVIOLATION (20152)	LACS <sup>Link</sup> seed violation

Error Message	Description
PCLERR_LACSDBEXPIRED (20153)	LACS <sup>Link</sup> database expired
PCLERR_LACSKEYERROR (20154)	Invalid LACS <sup>Link</sup> key
PCLERR_LACSTERMFAIL (20155)	LACS <sup>Link</sup> termination error

## Suite<sup>Link</sup> Error Messages

If Suite<sup>Link</sup> generates an error, Finalist outputs an error message to the Finalist log file.

## EWS Error Messages

The EWS error message is listed below. For more information, refer to the PBFNExtendedErrorDef structure.

**Table 26: EWS Option Error Messages**

Error Message	Description
4460: EWS Failure. Address found on EWS table.	The indicated address was found on the EWS table.

# eLOT Error Message

The eLOT error message is listed below. For more information, refer to the PBFNExtendedErrorDef structure.

**Table 27: eLOT Option Error Messages**

Error Message	Description
4461 - eLOT assignment has failed.	Address coded successfully but eLOT code was not assigned.

# 8 - AddrScan Option

## In this section

---

What is the AddrScan Option?.....	189
Why Should I Use the AddrScan Option?.....	190
How Do I Install AddrScan?.....	191
How Does AddrScan Processing Work?.....	191
Understanding AddrScan Output.....	196
Calling AddrScan.....	197
Setting Returned Line Options.....	198
Defining Processing Options.....	203
Activating AddrScan Advanced Options.....	203
Using the Lookup Tool to Define AddrScan Processing Options.....	205
CALLAREA Structure Definition.....	208
COBOL Copybook Version.....	212
Using ADDRTBLS for Exceptions Processing.....	214
Calling AddrScan With the Finalist Driver.....	215
Calling AddrScan With A User-Written Driver.....	215
AddrScan Examples.....	216
C Driver Example.....	218
COBOL Driver Example.....	219



# What is the AddrScan Option?

You can use the AddrScan function to:

- Identify individual components within a multiple-line address record
- Format and standardize the address information using USPS logic
- Return the processed address lines in an order you specify
- Identify firm, street, and city lines, so you can use the best combination of data from unstructured lists or multi-use lists to reformat and restructure your address files
- Preprocess addresses that contain multiple lines or "floating" lines

A floating line is an address line or city/state/ZIP Code line whose position appears in different parts of an address record. This line moves, or "floats", to any of the fixed number of fields allocated to a name/address record. The following example illustrates the concept of floating address lines.

## Example

In the first example address, the PO box is located on the first line. In the second example address, the PO box is located on the second line. The position of the PO box data moves or "floats" in the address record.

```
PO Box 99813
654 S Lawndale
Roseville, MI 48066
654 S Lawrence Ave
PO Box 9981B
Roseville, MI 48066
```

AddrScan lets you concatenate multiple address lines and return two address lines, a city/state line, a firm line, and a 10-byte ZIP + 4 field from the floating address lines. By returning fixed lines for processing, you eliminate the need to search for address, city/state, and ZIP Code components.

AddrScan attempts to preprocess addresses that contain multiple or floating lines. AddrScan cannot extract address information in the middle of other information on the same line. AddrScan may not be able to process address information spread across non-contiguous lines. The following is an example of address data that cannot be processed clearly.

## Example

```
2200 Western Ct
Ed's Place IS Apt 100
Account In Question 12345
Lisle IL 60532
3618
```

# Why Should I Use the AddrScan Option?

You can use AddrScan to:

- Consistently identify the appropriate address lines to pass to a postal coding API such as Finalist to ensure the best chance of coding
- Restructure your address lines
- Identify the ZIP Code and the ZIP + 4 code if present on input
- Identify and return Puerto Rico URB data
- Specify the format of the returned lines
- Return the original address lines in a standardized format and identify each line
- Accept up to six address lines as input and return four address lines and a 10-byte ZIP Code field
- Identify any of the six input address lines that were not used during processing
- Use your user-written driver to call AddrScan to identify and/or reorganize address lines before passing the data to a postal coding API such as Finalist
- Relate the returned line address to both your original input address and the standardized address
- Set up exceptions for AddrScan processing on the mainframe platform using the ADDRTBLS module (not available on other platforms)

The list below provides specific examples of how the AddrScan function benefits Finalist customers.

- Your input address record contains multiple or floating address lines. AddrScan accepts up to six address lines in an input record and returns four address lines and a 10-byte ZIP Code field.
- Your input address record has more than two address lines where the street address is found.
- Your input city/state line floats.
- Apartment or floor information appears on a separate line from the street information. This information often provides a more precise assignment of the sector segment number.
- You want to select the USPS-preferred address type from a choice of addresses.
- You need to specify the format of the returned line information for your calling program to receive.
- Your ZIP Code is not fix-fielded, but you want to perform a ZIP Code lookup using a postal coding API such as Finalist. A fix-fielded ZIP Code is a ZIP Code that always begins in the same position of the same field within an address record. The next example illustrates address records for which the ZIP Code is not fix-fielded.

## Example

The examples below do not contain fix-fielded ZIP Codes. The ZIP Code can appear in a different location within each address record.

```
123 Main St  
Roseville MI 48066  
451 S Washington Ave  
Apartment 1B
```

```
San Francisco, CA 94101
300 Birch Lane
Glen Ellyn IL
60137
```

## How Do I Install AddrScan?

AddrScan is installed as part of the standard Finalist installation.

## How Does AddrScan Processing Work?

AddrScan processing includes the following steps:

1. Identify the address lines. AddrScan identifies each input address line and assigns an input line type code to each line.

**Table 28: Input Line Type Codes**

Line Type Code	Description
0	Firm line
1	PO box address line
2	Address line with both a range and a suffix word
3	Address line with a range but no suffix Address line with a suffix but no range
4	Rural route address line
5	Personal name Firm name (without firm words) Unidentified
6	Apartment type line

Line Type Code	Description
7	Possible city line
8	City, state, ZIP Code (or ZIP + 4) line
9	Ignore this line
B	Box address line following a rural route address line
C	City line
M	Military address line
P	Urb (P on mainframe, p on Windows and *Nix)
R	Rural route address line preceding a box line
S	State line
W	PMB line
X	City and state line
Y	State and ZIP Code line
Z	ZIP Code line
Blank	No data present on input line

- Concatenate related data from the six input address lines. AddrScan uses the line type codes to determine which lines can be combined to form standardized addresses. The following concatenation rules determine the address line combination.

**Table 29: Concatenation Rules**

Rule Description
Line type code 6 lines append to line type code 2 lines.
Line type code 6 lines append to line type code 3 lines.



## Rule Description

---

Line type code 8 lines append to line type code 7 lines.

---

Line type code B lines append to line type code R lines.

---

**Note:** AddrScan combines lines only up to the point where the receiving line can hold all of the combined data. For example, assume there are three lines to combine, but the receiving line only has enough room for the first two lines. AddrScan will combine only the first two lines and leave the existing third line.

- a. Lines with a line type code 6, 7, or B append to the original street/city/rural line and form a new concatenated line.
- b. The new concatenated line overlays the original street/city/rural line.
- c. AddrScan "reassigns" the line type code 6, 7, and B lines a new line type code of "9".
- d. These lines are now part of the new concatenated line used to build the standardized address. In effect, the line type code "9" means "ignore this address line," and thereby prevents the lines from appearing twice in the standardized address. The following example demonstrates line type code assignment.

An example of an address passed for concatenation follows. Note the input line type codes assigned to each line during the identification process.

PO Box 123 \*\* 1 \*\*

123 Main Street \*\* 2 \*\*

Apt 12A \*\* 6 \*\*

Roseville MI 48066 \*\* 8 \*\*

AddrScan appends the third line (input line type code 6) to the second line (input line type code 2). After concatenation, the address lines are ready for standardization.

3. Standardize the address lines. Standardized lines are the original lines from your input file that are corrected and returned to the user input area (replacing your original input lines). The AddrScan standardization process steps are:
  - a) Return the standardized lines in any order. AddrScan places the standardized lines in the order identified by codes passed from the calling program. If you enter invalid codes, the system returns the lines in the order entered.
  - b) Return standardized line type codes to identify each of the standardized lines.

**Table 30: Standardized Line Type Codes**

Line Type Code	Description
0	Firm
1	P. O. box
2	Street (with suffix)
3	Street (without suffix)
4	Rural route
5	Personal name (probable)
8	City/state/ZIP Code
9	Miscellaneous (i.e., C/O)
M	Military
P	Puerto Rico URB Data <b>Note:</b> Displays "P" (upper case) for the z/OS platform. For all other platforms, displays "p" (lower case).

4. Return the processed lines.

- a) Build the standardized address in the order requested. If no order is requested, the lines are returned in the order entered.
- b) Identify the ZIP + 4 Code.
- c) Remove the ZIP + 4 Code from the returned city line and place the ZIP + 4 Code in a fixed field.

AddrScan only identifies the ZIP + 4 Code during this process. The Finalist API verifies the ZIP + 4 Code.

- d) Build two four-byte line indexes. The line indexes help you relate the returned address to your original input record and to the standardized address. Index1 contains the line numbers in your original input record that correspond to the returned lines. Index2 contains the line numbers in the standardized address that correspond to the returned lines.

**Table 31: Index Pointer Table**

Position	Description
1	Source of Returned Line 1
2	Source of Returned Line 2
3	Source of Puerto Rico urbanization data (Index1 only)
4	Source of Returned City Line

The ZIP Code is returned in a separate field and does not display in the returned address lines.

- a. Build the returned lines address by applying the format specified (Returned Line Options) to the standardized address to produce the returned lines address. AddrScan returns four 70-byte lines.

**Table 32: Returned Line Contents**

Line Number	Description
Line 1	Street address data
Line 2	Street address data
Line 3	City and state
Line 4	Firm name (if supplied in the input record)

- a. Pass the four-line address back to the calling program in the requested format. For additional examples, refer to [AddrScan Examples](#) on page 216.

Example

The Returned Line Options are set to "56N". The following table describes this setting.

**Table 33: Returned Line Options Example**

Parameter Position	Description
Return Line 1	5=Return first valid line above city line into Return Line 1

Parameter Position	Description
Return Line 2	6=Return second valid line above city line into Return Line 2
Return Line 3	N=No special options used
Return Line 4	Remains blank - no firm line present

5. Identify the input lines that were not used during processing.

#### Example

AddrScan used lines 1, 2, 3, and 5 to process the input address. Input lines 4 and 6 were not used. The unUsedLines (ADDRPASS-UNUSED) six-byte field displays:

```
46<blank><blank><blank><blank>
```

Blanks are used as placeholders to fill the six-byte field.

## Understanding AddrScan Output

The following list describes AddrScan processing output.

- The original input returns left-justified and with only one blank between words.
- Valid special characters are permitted: & - / ' (single quote character) . : \ , # , (comma)
  - Note:** The pound sign (#) is the only special character you can use to begin a word. All lower case alpha characters translate to upper case. All other special characters are blanked out.
- AddrScan allows words such as APT, Apartment, Suite, or Condo, to be the first words in a line when identifying an apartment type line. FLOOR, FL, or FLR can be the first or last word. Apartment type lines can appear anywhere and combine with the street address line. Up to two apartment type lines will be combined with the address line.
- AddrScan treats a one-word line as an apartment type line if the word is one of the following:
  - All numeric (12345)
  - A number that ends with a single alpha (5B)
  - A number that starts with a single alpha (C5)
  - A token that starts with an alpha, followed by a hyphen (-) followed by a number (D-5). By default AddrScan automatically concatenates these tokens to the address line.
- ZIP + 4 Codes are always returned, with a hyphen, in the ZIP Code field.

- If names, such as company and personal names, pass with the address line, AddrScan finds and returns the valid address and city/state lines.
- Even if the street address and city lines are not in the proper order, AddrScan returns the valid address and city/state lines. However, within the input lines, the ZIP Code must follow the city/state data with a space between state and ZIP Code.
- AddrScan includes tables that identify apartment, box, rural route, street, firms, and some Spanish words. These tables are used to identify the address lines.
- AddrScan output includes the unUsedLines field that identifies the input lines that were not used during processing.

## Calling AddrScan

You can call AddrScan from a user-written driver. Sample C and COBOL programs are provided. The following table describes the files provided to support AddrScan.

**Table 34: Finalist AddrScan Files**

File	Description
addr021d.cob	COBOL copybook for calling AddrScan
ADDRSCAN	z/OS load module
addrscan.h	Contains AddrScan definitions
addrscan.dll	Finalist AddrScan Windows .dll
addrscan.lib	Finalist AddrScan Windows library
addrscan.a	Finalist AddrScan UNIX library

# Setting Returned Line Options

To set the returned line order (ADDRPASS-OPTIONS), use the returned line options to specify the format of the lines returned for address line 1 and address line 2. You can also specify a special option, such as separating combined address lines.

## System Default

The system default for the returned line options is “95N”. To use the default value, pass “YES” or blanks for the Returned Line Options.

**Table 35: Default Returned Line Settings**

Returned Line	Value	Description
1	9	Best street or box line from top. Otherwise, use the first firm or rural route line from the top.
2	5	First valid line above city line.
Special Option	N	No special options.

## User-Specified Formats

To specify the format of your returned lines address, select three options from the following tables (one for each returned line and a special option).

**Note:** When you enter options, you must enter all three options or the results will be undesirable.

To specify a user-defined format for Returned Line 1 (ADDRPASS-OPTION1), select one of the options shown in the following table.

**Table 36: Returned Line 1 Table**

Code	Returned Line 1 Description
1	First valid line from top
3	First firm line from top
5	First valid line above city line
7	Street address line above city line
8	PO box or RR/HC line above city line
9	Best street or box line from top; if not found, the first firm or rural route line from top. When Option 9 is specified for returned line 1, AddrScan performs selection from the top down as shown below: <ol style="list-style-type: none"> <li>1. First box line or complete address Example: PO BOX 12 or 123 Main St.</li> <li>2. First street line with a range but no suffix Example: 123 Main 3. First street line with a suffix but no range Example: Main St.</li> <li>4. First rural route line Example: RR 1 Box 12</li> <li>5. First firm type Example: Precisely</li> </ol>

To specify a user-defined format for Returned Line 2 (ADDRPASS-OPTION2), select one of the options shown in the following table.

**Table 37: Returned Line 2 Table**

Code	Returned Line 2 Description
B	Blank line
1	First valid line from top
2	Second valid line from top
3	First firm line from top
4	Second half of combined address line if first half is returned in line 1
5	First valid line above city line

Code	Returned Line 2 Description
6	Second valid line above city line
7	Street address line above city line
8	PO box or RR/HC line above city line
9	First valid line above city line. If that criteria is not met resulting in a blank second line, AddrScan returns the topmost unused line in returned line 2.

**Note:** If you make the same selection for Returned Line 1 and Returned Line 2, the second returned line will be blank.

To specify a special option (ADDRPASS-OPTION3), select one option in the following table.

**Table 38: Special Options Table**

Code	Special Option Description
A	All special options
N	No special options
R	Add a range of "0" to street lines without a range Example: Main St becomes 0 Main St
S	Separate combined address lines

### Example

The first address below shows a combined address line. The second example shows the same address line using special option "S" (separate combined address lines).

```
123 MAIN ST RR 1 BOX 2
ROSEVILLE MI 48066
```

```
123 MAIN ST
RR 1 BOX 2
ROSEVILLE MI 48066
```



## Setting Returned Line Order

To define the order for returning the standardized lines (ADDRPASS-RTN-ORDER), specify a Returned Line Order code.

**Table 39: Return Line Order**

Code	Description
0	Firm line
1	PO Box address line
2	Address line with both a range and a suffix word
3	Address line with a range but no suffix, Address line with a suffix but no range
4	Rural route address line
5	Personal name, Firm name (w/o firm words), Unidentified
6	Apartment type line
7	Possible city line
8	City line
9	Ignore this line
B	Box address line following a rural route address line
M	Military address line
N	An N in the first return position indicates to return the default that is best address line 1, best address line 2, city state, firm. URB and ZIP Code are returned in separate fields.
R	Rural route address line preceding a box line

### Example

This example demonstrates how you can specify the returned line order for a multi-line address in order to affect what information is returned.

```

Input Address:      Mrs P. R. Farber
                   Office World Inc
                   123 Main
                   4064 Monroe Street
                   RR 1 Box 12345
                   Toledo OH 43060

Returned Line Options: 35N
Returned Line Order:  0543218
Standardized Address: OFFICE WORLD INC
                   MRS P. R. FARBER
                   RR 1 BOX 12345
                   123 MAIN
                   4064 MONROE STREET
                   TOLEDO OH 43060

Returned Address:   OFFICE WORLD INC
                   4064 MONROE STREET
                   TOLEDO OH

Returned ZIP Code:  43060
Returned Line Type Code: 054328

```

## Defining the Concatenation Line Maximum

Concatenation occurs when related data is located on multiple lines.

```

123 MAIN ST
BLDG 4
APT 12.

```

AddrScan concatenates the previous lines as:

```

123 MAIN ST BLDG 4 APT 12

```

Note that the concatenation line maximum (ADDRPASS-LNMX-NUM) is typically set to 70 characters. This may cause problems if you are going to use the standardized lines for reuse in your records and your original records are not large enough to hold the new line. In such cases, you can change the maximum standardized line length to prevent line overflow.

## Defining Processing Options

Because the Processing Options use a field that is also an output field (ADDRPASS-TYPES or pAddrTyps), the Processing Options must be reset before each call to AddrScan.

- To prevent AddrScan from concatenating lines beginning with a # to existing address lines, enter a # in one of the columns of ADDRPASS-TYPES or pAddrTyps.
- To prevent AddrScan from recognizing periods as a valid characters (to specify that a period should be removed before scanning address lines), enter a P in one of the columns of ADDRPASS-TYPES or pAddrTyps.
- To prevent AddrScan from concatenating one-word lines to address lines, enter a D in one of the columns of ADDRPASS-TYPES or pAddrTyps.
- To activate the AddrScan advanced options, place an A in one of the columns of ADDRPASS-TYPES or pAddrTyps.

## Activating AddrScan Advanced Options

For cases where the FIRM, URB, or city/state information is fixed and only the address lines are floating, AddrScan provides advanced options that allow you to select specific fields to exclude from AddrScan processing.

The AddrScan advanced options are, by default, turned off. To activate the AddrScan advanced options (cNoMergeSecPMB, cSkipFirm, cSkipUrb, cSkipCSZ), follow these steps:

1. Required. Set any position of ADDRPASS-TYPES or pAddrTyps to "A".
2. Required. Set the version field, cVersion, to the current version number. Please refer to Addrscan.h or the ADDR021D COBOL copybook for the current version of ADDRSCAN.DLL.
3. Optional. Set the desired option(s) to "S" (single character) to activate that option.

## Advanced Processing Options - Input Fields

Input Field	Description
char cVersion[4]	Version control information

Input Field	Description
char cNoMergeSecPMB	Do not merge separated 2nd unit and PMB information
char cSkipFirm	Skip firm process
char cSkipUrb	Skip URB process
char cSkipCSZ	Skip city, state and ZIP Code line process

## Advanced Processing Options - Output Fields

The six (6) values in this output field represent a description of each of the six (6) corresponding standardized address lines. For a list of the standardized line type codes, refer to [How Does AddrScan Processing Work?](#) on page 191. If a standardized address line is blank, a blank will appear in corresponding position in this output field.

Output Field	Description
char cCombineLines[6]	Combined line indexes
char pInputLineTypes[6]	Original input line type
char cFirmIdx	Firm index
char cZipIdx	ZIP Code index
char cUrbIdx	URB index

**Note:** This is duplicated information of the 3rd character of pAddrNdx1 variable.

# Using the Lookup Tool to Define AddrScan Processing Options

To use the Lookup Tool to define AddrScan processing options, follow these steps:

1. To open the Lookup Tool, click Start > Programs > Precisely > Finalist > Lookup.
2. Click Postal Code > AddrScan.
3. To set AddrScan advanced options, click the Adv button. Use the Process/Advanced Options dialog box to set advanced processing options.
4. Click the check box to activate the option displayed. When selected, the valid value (shown in parentheses) displays on the Adv button. For example, if you activate the first three options by clicking on the first three check boxes, the Adv button on the Code an Address dialog box displays as Adv(#PD).

Field	Description
Prevent AddrScan from concatenating lines beginning with # to existing lines (#)	To prevent AddrScan from concatenating lines beginning with a # to existing address lines, select this option.
Prevent AddrScan from recognizing periods as valid characters (P)	To prevent AddrScan from recognizing periods as a valid characters (to specify that a period should be removed before scanning address lines), select this option.
Prevent AddrScan from concatenating dangling lines (D)	To prevent AddrScan from concatenating one-word lines to address lines, select this option.
Skip Merging Secondary or PMB (S)	To prevent AddrScan from merging separated 2nd unit and PMB information, select this option.
Skip Firm (F)	To prevent AddrScan from performing firm processing, select this option.
Skip URB (U)	To prevent AddrScan from performing URB processing, select this option.
Skip City, State, or ZIP Code (Z)	To prevent AddrScan from performing City, State, or ZIP Code processing, select this option.

5. To set AddrScan returned line options, click the Rtn button. Use the Return Line Options dialog box to specify the format of your returned lines. Select three options (one for each returned line

and a special option). If you do not select a value for all three sections, processing results may be undesirable.

Click the check box to activate the option displayed. When selected, the valid value (shown in parentheses) displays on the Rtn button. For example, the Rtn button on the Code an Address dialog box displays as Rtn(12A) if you select the following options:

- Return Line 1 — First valid line from top (1)
- Return Line 2 — Second valid line from top (2)
- Special Option — All special options (A)

Field	Description																		
Return Line 1	<p>To specify the format for the first returned line, select one of the following options.</p> <table style="border: none;"> <tr> <td style="padding-right: 10px;"><b>1</b></td> <td>First valid line from top</td> </tr> <tr> <td><b>3</b></td> <td>First firm line from top</td> </tr> <tr> <td><b>5</b></td> <td>First valid line above city line</td> </tr> <tr> <td><b>7</b></td> <td>Street address line above city line</td> </tr> <tr> <td><b>8</b></td> <td>PO Box line above the city line</td> </tr> <tr> <td><b>9</b></td> <td>Best street or box line from the top; or else, first firm on rural line from top</td> </tr> </table>	<b>1</b>	First valid line from top	<b>3</b>	First firm line from top	<b>5</b>	First valid line above city line	<b>7</b>	Street address line above city line	<b>8</b>	PO Box line above the city line	<b>9</b>	Best street or box line from the top; or else, first firm on rural line from top						
<b>1</b>	First valid line from top																		
<b>3</b>	First firm line from top																		
<b>5</b>	First valid line above city line																		
<b>7</b>	Street address line above city line																		
<b>8</b>	PO Box line above the city line																		
<b>9</b>	Best street or box line from the top; or else, first firm on rural line from top																		
Return Line 2	<p>To specify the format for the second returned line, select one of the following options.</p> <table style="border: none;"> <tr> <td style="padding-right: 10px;"><b>B</b></td> <td>Blank line</td> </tr> <tr> <td><b>1</b></td> <td>First valid line from top</td> </tr> <tr> <td><b>2</b></td> <td>Second valid line from top</td> </tr> <tr> <td><b>3</b></td> <td>First line from top</td> </tr> <tr> <td><b>4</b></td> <td>Second half of combined address line, if the first half is returned in line 1</td> </tr> <tr> <td><b>5</b></td> <td>First valid line above city line</td> </tr> <tr> <td><b>6</b></td> <td>Second valid line above city line</td> </tr> <tr> <td><b>7</b></td> <td>Street address line above city line</td> </tr> <tr> <td><b>8</b></td> <td>PO Box or RR/HC line above city line</td> </tr> </table>	<b>B</b>	Blank line	<b>1</b>	First valid line from top	<b>2</b>	Second valid line from top	<b>3</b>	First line from top	<b>4</b>	Second half of combined address line, if the first half is returned in line 1	<b>5</b>	First valid line above city line	<b>6</b>	Second valid line above city line	<b>7</b>	Street address line above city line	<b>8</b>	PO Box or RR/HC line above city line
<b>B</b>	Blank line																		
<b>1</b>	First valid line from top																		
<b>2</b>	Second valid line from top																		
<b>3</b>	First line from top																		
<b>4</b>	Second half of combined address line, if the first half is returned in line 1																		
<b>5</b>	First valid line above city line																		
<b>6</b>	Second valid line above city line																		
<b>7</b>	Street address line above city line																		
<b>8</b>	PO Box or RR/HC line above city line																		

Field	Description
Special Option	To specify a special option for your returned lines, select one of the following options. <ul style="list-style-type: none"> <li><b>A</b> All special options</li> <li><b>N</b> No special options</li> <li><b>R</b> Add a range of "0" to street lines without a range (for example, Main St becomes 0 Main St)</li> <li><b>S</b> Separate combined address lines (S)</li> </ul>

6. To define the order for returning the standardized address lines, click the Ord button. Use the Return Line Order dialog box to set the return line order options.

Click the check box to activate the option displayed. When selected, the valid value (shown in parentheses) displays on the Ord button. For example, the Ord button on the Code an Address dialog box displays as Ord(0128) if you select the following line types:

- 0 — Firm line
- 1 —PO Box address line
- 2 —Address line with both a range and a suffix word
- 8 —Second valid line from top

Field	Description
Line Types	<p>Select the line types, in preferred order, for returning the standardized address lines. You may select up to 9 line types.</p> <p>To select a line type, click on the desired line. Then click on the arrow to move the selected line type value to the Chosen Opts box.</p> <p>Use the Clear button to delete the contents of the Chosen Opts box.</p>
	<p><b>0</b> Firm line</p> <p><b>1</b> PO Box address line</p> <p><b>2</b> Address line with both a range and a suffix word</p> <p><b>3</b> Address line (a range but no suffix) or (a suffix but no range)</p> <p><b>4</b> Rural route address line</p> <p><b>5</b> Personal name, firm name (without firm words), unidentified</p> <p><b>6</b> Apartment type line</p> <p><b>7</b> Possible city line</p> <p><b>8</b> City line</p> <p><b>9</b> Ignore this line</p> <p><b>B</b> Box address line following a rural route address line</p> <p><b>M</b> Military address line</p> <p><b>R</b> Rural route address line preceding a box line</p>

## CALLAREA Structure Definition

The definition of the CALLAREA structure that is used to communicate with the AddrScan function is contained in the header file addrscan.h as shown below.

```
#define LN 70
.
.
#define NUM_INLINE 6
#define NUM_OUTLINE 4
#define NUM_OPTS 3
#define INDEX_SIZE 4
.
.
typedef struct {
    char pInline[NUM_INLINE][LN];
```



```

    char  pOutline[NUM_OUTLINE][LN];
    char  urb[URB_SIZE];
    char  pAddrNdx1[INDEX_SIZE];
    char  pAddrNdx2[INDEX_SIZE];
    char  pAddrZip4[10];
    char  pAddrOpts[NUM_OPTS];
    char  pAddrTyps[6];          /* setting 'A' to active advanced
option.*/
    char  pAddrRord[9];
    short sAddrLnMx;
    char  University;
    char  Dormatory;
    char  pAddrAnch[8];
    /* Advanced options: */
    char  version[4];
    char  pCombineLines[6];
    char  pInputLineTypes[6];
    char  cNoMergeSecPMB;
    char  cSkipFirm;
    char  cSkipUrb;
    char  cSkipCSZ;
    char  cFirmIdx;
    char  cZipIdx;
    char  cUrbIdx;
    char  cFiller2[511];
} CALLAREA, *PCALLAREA;

```

**Table 40: CALLAREA Member Descriptions**

Member Name	Length	Description
char pInline[0]	70	Input address line 1.
char pInline[1]	70	Input address line 2.
char pInline[2]	70	Input address line 3.
char pInline[3]	70	Input address line 4.
char pInline[4]	70	Input address line 5.
char pInline[5]	70	Input address line 6.
char pOutline[0]	70	Returned address line 1.
char pOutline[1]	70	Returned address line 2.

Member Name	Length	Description
char pOutline[2]	70	Returned city/state line.
char pOutline[3]	70	Returned firm line (if a firm was provided on input).
char urb	30	Returned urbanization line.
char pAddrNdx1	4	Returned index containing the line numbers of your original input address which correspond to the returned address lines.
char pAddrNdx2	4	Returned index containing the line numbers of the standardized input address which correspond to the returned address lines.
char pAddrZip4	10	Returned ZIP + 4 Code. Consists of the 5-digit ZIP Code, a hyphen, and the sector/segment.
char pAddrOpts	3	<p>Returned Line Options (each 1 character long). Defines the option for selecting the line to be returned in output line 1 (pOutline[0]), the option for selecting the line to be returned in output line 2 (pOutline[1]), and the special option selector.</p> <p><b>Note:</b> If pAddrOpts contains HIGH-VALUES (0xFFFFFFFF), AddrScan terminates and frees all acquired storage.</p>
char pAddrTyps	6	<p>Returned line type codes that allow you to identify each of the standardized address lines.</p> <p>pAddrTyps can also be used as an input field to alter AddrScan processing.</p> <ul style="list-style-type: none"> <li><b>#</b> To prevent AddrScan from concatenating lines beginning with # to existing address lines, enter # in the pAddrTyps field.</li> <li><b>P</b> To prevent AddrScan from recognizing periods as valid characters and to specify that periods should be removed before scanning address lines, enter a P in the pAddrTyps field.</li> <li><b>D</b> To prevent AddrScan from concatenating a one-word line to an address line, enter a D in the pAddrTyps field. Note that, If an address line ends with a Unit Designator, a one-word line will always be concatenated.</li> <li><b>A</b> To activate the advanced options of AddrScan, place an A in the pAddrTyps field.</li> </ul> <p><b>Note:</b> The values #, P, D, and A can populate any of the positions of the pAddrTyps field.</p>

Member Name	Length	Description
char pAddrRord	9	Returned Line Order. Defines the order in which standardized lines are returned.
short sAddrLnMx	2	This field is not used at this time.
char University	1	Set to "U" if university word found.
char Dormatory	1	Set to "D" if dormitory word found.
char pAddrAnch	8	Set to HIGH-VALUES (0xFF) prior to the first call.
char version[4]	4	AddrScan version
char pCombineLines[6]	6	Combined line index when occurs
char pInputLineTypes[6]	6	Input line types.
char cNoMergeSecPMB	1	Leave secondary or PMB line as is. Do not merge with any address line.
char cSkipFirm	1	Skip firm line processing.
char cSkipUrb	1	Skip return URB line processing.
char cSkipCSZ	1	Skip city, state and ZIP Code line processing.
char cFirmIdx	1	Firm index was found on input line.
char cZipIdx	1	ZIP Code index was found on input line.
char cUrbIdx	1	URB index was found on input line.
char unUsedLines	6	Identify unused lines.
char cFiller2[511]	505	Reserved for future use.

# COBOL Copybook Version

```

*****
*           *           ADDRSCAN CALL-AREA           *           *
*           *****
* NOTE:    THE INITIAL VALUE OF ADDRPASS-ANCHOR WILL BE IGNORED *
*           BY THE COMPILER IF THIS IS IN THE FILE SECTION OR   *
*           LINKAGE SECTION. IN THAT CASE, MOVE SPACES TO      *
*           ADDRPASS-ANCHOR BEFORE THE FIRST BATCH ADDRSCAN   *
*           CALL BUT NOT ON SUBSEQUENT CALLS. IN CICS, THE VALUE *
*           DOES NOT NEED TO BE INITIALIZED AT ALL.           *
*****
01  ADDRPASS-CALL-AREA.
    05  ADDRPASS-PASSED-LINES.
        10  ADDRPASS-INLINE1    PIC X(70).
        10  ADDRPASS-INLINE2    PIC X(70).
        10  ADDRPASS-INLINE3    PIC X(70).
        10  ADDRPASS-INLINE4    PIC X(70).
        10  ADDRPASS-INLINE5    PIC X(70).
        10  ADDRPASS-INLINE6    PIC X(70).
    05  FILLER REDEFINES ADDRPASS-PASSED-LINES.
        10  ADDRPASS-INLINEX    OCCURS 6 TIMES PIC X(70).
    05  ADDRPASS-RETURNED-LINES.
        10  ADDRPASS-OUTLINE1   PIC X(70).
        10  ADDRPASS-OUTLINE2   PIC X(70).
        10  ADDRPASS-OUTLINE3   PIC X(70).
        10  ADDRPASS-OUTLINE4   PIC X(70).
    05  FILLER REDEFINES ADDRPASS-RETURNED-LINES.
        10  ADDRPASS-OUTLINEX   OCCURS 4 TIMES PIC X(70).
    05  ADDRPASS-OUTURB        PIC X(30).
    05  ADDRPASS-INDEXES.
        10  ADDRPASS-INDEX1     PIC X(04).
        10  ADDRPASS-INDEX2     PIC X(04).
    05  ADDRPASS-ZIP-PLUS-4.
        10  ADDRPASS-ZIP5       PIC X(05).
        10  ADDRPASS-HYPHEN     PIC X(01).
        10  ADDRPASS-SECSEG     PIC X(04).
    05  ADDRPASS-OPTIONS.
        10  ADDRPASS-OPTION1    PIC X(01) VALUE '9'.
        10  ADDRPASS-OPTION2    PIC X(01) VALUE '5'.
        10  ADDRPASS-OPTION3    PIC X(01) VALUE 'N'.
    05  ADDRPASS-TYPES.
        10  ADDRPASS-TYPE1      PIC X(01).
        10  ADDRPASS-TYPE2      PIC X(01).
        10  ADDRPASS-TYPE3      PIC X(01).
        10  ADDRPASS-TYPE4      PIC X(01).
        10  ADDRPASS-TYPE5      PIC X(01).
        10  ADDRPASS-TYPE6      PIC X(01).

```

```

05 FILLER REDEFINES ADDRPASS-TYPES.
   10 ADDRPASS-TYPEX OCCURS 6 TIMES PIC X(01).
05 ADDRPASS-RTN-ORDER.
   10 ADDRPASS-RTN-ORD1 PIC X(01) VALUE 'N'.
   10 ADDRPASS-RTN-ORD2 PIC X(01).
   10 ADDRPASS-RTN-ORD3 PIC X(01).
   10 ADDRPASS-RTN-ORD4 PIC X(01).
   10 ADDRPASS-RTN-ORD5 PIC X(01).
   10 ADDRPASS-RTN-ORD6 PIC X(01).
   10 ADDRPASS-RTN-ORD7 PIC X(01).
   10 ADDRPASS-RTN-ORD8 PIC X(01).
   10 ADDRPASS-RTN-ORD9 PIC X(01).
05 FILLER REDEFINES ADDRPASS-RTN-ORDER.
   10 ADDRPASS-RTN-ORDX OCCURS 9 TIMES PIC X(01).
05 ADDRPASS-LINE-MAX.
   10 ADDRPASS-LNMX-NUM PIC 9(04) BINARY VALUE 70.
   10 ADDRPASS-UNIV-WORD PIC X(01) VALUE SPACES.
   10 ADDRPASS-DORM-WORD PIC X(01) VALUE SPACES.
05 ADDRPASS-ANCHOR PIC X(08) VALUE SPACES.
05 ADDRPASS-VERSION PIC X(04) VALUE 'NNNN'.
05 ADDRPASS-COMB-LINES.
   10 ADDRPASS-CMB-LINE1 PIC X(01) VALUE SPACES.
   10 ADDRPASS-CMB-LINE2 PIC X(01) VALUE SPACES.
   10 ADDRPASS-CMB-LINE3 PIC X(01) VALUE SPACES.
   10 ADDRPASS-CMB-LINE4 PIC X(01) VALUE SPACES.
   10 ADDRPASS-CMB-LINE5 PIC X(01) VALUE SPACES.
   10 ADDRPASS-CMB-LINE6 PIC X(01) VALUE SPACES.
05 FILLER REDEFINES ADDRPASS-COMB-LINES.
   10 ADDRPASS-CMB-LINEX OCCURS 6 TIMES PIC X(01).
05 ADDRPASS-INPUT-LINE-TYPES.
   10 ADDRPASS-INPUT-LINE1 PIC X(01) VALUE SPACES.
   10 ADDRPASS-INPUT-LINE2 PIC X(01) VALUE SPACES.
   10 ADDRPASS-INPUT-LINE3 PIC X(01) VALUE SPACES.
   10 ADDRPASS-INPUT-LINE4 PIC X(01) VALUE SPACES.
   10 ADDRPASS-INPUT-LINE5 PIC X(01) VALUE SPACES.
   10 ADDRPASS-INPUT-LINE6 PIC X(01) VALUE SPACES.
05 FILLER REDEFINES ADDRPASS-INPUT-LINE-TYPES.
   10 ADDRPASS-INPUT-LINEX OCCURS 6 TIMES PIC X(01).
05 ADDRPASS-NO-MERGE-SEC-PMB PIC X(01) VALUE SPACES.
05 ADDRPASS-SKIP-FIRM PIC X(01) VALUE SPACES.
05 ADDRPASS-SKIP-URB PIC X(01) VALUE SPACES.
05 ADDRPASS-SKIP-CSZ PIC X(01) VALUE SPACES.
05 ADDRPASS-FIRM-IDX PIC X(01) VALUE SPACES.
05 ADDRPASS-ZIP-IDX PIC X(01) VALUE SPACES.
05 ADDRPASS-URB-IDX PIC X(01) VALUE SPACES.
05 ADDRPASS-UNUSED PIC X(06) VALUE SPACES.
05 FILLER PIC X(505).
***** END OF ADDRSCAN CALL-AREA *****

```

The following examples illustrate the values you can return within the ZIP Code field pAddrZip4.

### Example

AddrScan returns a ZIP Code field that consists of a five-digit ZIP Code, a hyphen, and a sector segment number. The hyphen separates the ZIP Code from the sector segment number as shown next.

```
60148-6494
```

### Example

If you do not have a sector segment number present in the record, the last five positions will be blank. If AddrScan is unable to locate a ZIP Code within your address data, the ZIP Code field returns as:

```
00000-0000
```

## Using ADDRTBLS for Exceptions Processing

The ADDRTBLS module provides a method for setting up AddrScan processing exceptions. For example, you might want to use the ADDRTBLS module to set up exceptions to process:

- P.O. SMITH as a name instead of as post office box
- VISTA SANTA ROSA as a city name instead of as an URB
- STAR RANCH as a city instead of an alias for STAR ROUTE

ADDRTBLS is provided as BAL source in the FNSOURCE library. Follow these steps to use ADDRTBLS.

1. Modify the ADDRTBLS source
2. Assemble ADDRTBLS and link into a load module. Link the load module using AMODE=31, RMODE=ANY.
3. Place the load module into the Finalist batch, CICS, and IMS load libraries.

**Note:** ADDRTBLS is only available on the z/OS platform. ADDRTBLS is not available on other platforms.

## Calling AddrScan With the Finalist Driver

You can call AddrScan using the Finalist driver by specifying the `cAddrScan=YES` option in your JOB file. Use the same `cAddrScan` parameter to specify AddrScan configuration options. For example, `cAddrScan=56N`.

## Calling AddrScan With A User-Written Driver

The steps to call AddrScan with a user-written driver are:

1. Define a local CALLAREA structure.
2. Initialize the entire CALLAREA structure to blanks except for `pAddrAnch` (ADDRPASS-ANCHOR) which should be HIGH-VALUES.
3. Set any members of the CALLAREA structure that you want to remain the same from call to call (for example, the maximum length of the standardized address lines).
4. Initialize the input address lines of the CALLAREA structure to blanks (also do this before each subsequent call to AddrScan).
5. Move the data from your input record to the appropriate members within the CALLAREA structure.
6. Call the AddrScan function, passing the address of the CALLAREA structure.
7. Perform any custom processing using the returned address lines, returned city/state line, and returned ZIP Code field within the CALLAREA structure.
8. After AddrScan processing, set `pAddrOpts` (ADDRPASS-OPTIONS) to HIGH-VALUES to terminate AddrScan and make one last call.

# AddrScan Examples

The following table provides you with samples of how AddrScan processes input records and builds the standardized lines, returned lines, and the line indexes. For these examples, option "54S" was specified for the Returned Line Options.

Input Address	Standardized Address	Returned Lines Address
PO Box 123 123 Main St Apt 12A Roseville Mi 48066	PO BOX 123 123 MAIN ST APT 12A ROSEVILLE MI 48066	123 MAIN ST APT 12A ROSEVILLE MI
Index = 2004	Index2 = 2003	ZIP Code = 48066

Input Address compared to Return Address Lines	Standardized Address compared to Return Address Lines
<ul style="list-style-type: none"> <li>• 2 - Return address line 1 was found on input line 2.</li> <li>• 0 - Return address line 2 was not found in input.</li> <li>• 0 - Return URB was not found on standardized address.</li> <li>• 4 - Return city/state line was found on input line 4.</li> </ul>	<ul style="list-style-type: none"> <li>• 2 - Return address line 1 was found on standardized line 2.</li> <li>• 0 - Return address line 2 was not found on standardized address.</li> <li>• 0 - Will always be 0.</li> <li>• 3 - Return city/state line was found on standardized line 3.</li> </ul>

Input Address	Standardized Address	Returned Lines Address
123 Main St Box 12A Roseville MI 48066	123 MAIN ST BOX 12A ROSEVILLE MI 48066	123 MAIN ST BOX 12A ROSEVILLE MI
Index = 1102	Index2 = 1203	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
123 Main St RR 1 Box 123 Roseville Mi 48066	123 MAIN ST RR 1 BOX 123 ROSEVILLE MI 48066	123 MAIN ST RR 1 BOX 123 ROSEVILLE MI



Input Address	Standardized Address	Returned Lines Address
Index = 1102	Index2 = 1203	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
123 Main St 2nd Fl Roseville Mi 48066	123 MAIN ST 2ND FL ROSEVILLE MI 48066	123 MAIN ST 2ND FL ROSEVILLE MI
Index = 1004	Index2 = 1002	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
C/O John Doe 123 Main St Roseville Mi 48066	C/O JOHN DOE 123 MAIN ST ROSEVILLE MI 48066	123 MAIN ST ROSEVILLE MI
Index = 2003	Index2 = 2003	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
RR1 Box 123 Roseville Mi 48066	RR1 BOX 123 ROSEVILLE MI 48066	RR1 BOX 123 ROSEVILLE MI
Index = 1003	Index2 = 1002	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
1 Elm St Apt 6 Bldg 1 Roseville Mi 48066	1 ELM ST APT 6 BLDG 1 ROSEVILLE MI 48066	1 ELM ST APT 6 BLDG 1 ROSEVILLE MI
Index = 1004	Index2 = 1002	ZIP Code = 48066

Input Address	Standardized Address	Returned Lines Address
25 Calle 3 Urb San Juan Bautista Barceloneta PR 00607	25 CALLE 3 URB SAN JUAN BAUTISTA BARCELONETA PR 00607	25 CALLE 3 BARCELONETA PR
Index = 1023	Index2 = 1003	ZIP Code = 00607

## C Driver Example

A C program listing is shown next.

```

/*****
/* AddrScan sample program          */
/*****
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "pbfndef.h"
#include "addrscan.h"
void main( int argc, char *argv[] )
{
    CALLAREA CallArea;
    short      i;
    char       pLine[ NUM_INLINE ][ LN+1 ];
    INTEGER    nDone;
    /* initialize CallArea with blank space
     * do NOT use '\0' char.
     */
    memset(&CallArea, ' ', sizeof (CALLAREA));
    memset(CallArea.pAddrAnch, 0xFF, sizeof(CallArea.pAddrAnch));
    /* Fill the input lines with data. */
    strcpy( pLine[0], "pbss corp." );
    strcpy( pLine[1], "123 MAIN street" );
    strcpy( pLine[2], "APT 12A" );
    strcpy( pLine[3], "Roseville Mi" );
    strcpy( pLine[4], "48066" );
    strcpy( pLine[5], "" );
    /* Set the maximum length of the standardized lines. */
    CallArea.sAddrLnMx = LL;
    /* Simulate a processing loop because a real application would */
    /* call addrscan for each of many records.                      */
    nDone = 0;
    while ( !nDone )
        {

```

```

/* Fill the input lines with blank characters. */
memset( CallArea.pInline[0], (char) ' ', LL * NUM_INLINE );
/* Copy the input lines into the CallArea structure. */
for ( i = 0 ; i < NUM_INLINE ; i++ )
    {
        memcpy( CallArea.pInline[i], pLine[i], strlen( pLine[i] ) );
    }
/* Set the format of the returned address lines to "54S". */
memcpy( CallArea.pAddrOpts, "54S", 3 );
/* Specify no special return order. */
memcpy( CallArea.pAddrRord, "N", 1 );
printf( "\nINPUT LINES:\n\n" );
for ( i = 0; i < NUM_INLINE ; i++ )
    {
        printf( "Line #%ld : [%40.40s]\n", i+1, CallArea.pInline[i] );
    }
addrscan( &CallArea );
printf( "\nSTANDARDIZED INPUT LINES:\n\n" );
for ( i = 0; i < NUM_INLINE ; i++ )
    {
        printf( "Line #%ld : [%40.40s]\n", i+1, CallArea.pInline[i] );
    }
printf( "\nRETURNED INFORMATION:\n\n" );
for ( i = 0 ; i < NUM_OUTLINE ; i++ )
    {
        printf( "Line #%ld : [%40.40s]\n", i+1, CallArea.pOutline[i]
);
    }
printf( "Index1  : [%4.4s]\n",    CallArea.pAddrNdx1 );
printf( "Index2  : [%4.4s]\n",    CallArea.pAddrNdx2 );
printf( "Zip4    : [%10.10s]\n",  CallArea.pAddrZip4 );
printf( "Types   : [%6.6s]\n",    CallArea.pAddrTyps );
/* Cause the loop to be exited. */
nDone = 1;
} /* while ( !nDone ) */
fflush( stdout );
} /* End of "main" */

```

## COBOL Driver Example

```

IDENTIFICATION DIVISION.
PROGRAM-ID. 'ADDRTEST'.
DATE-COMPILED.
** =====
*
```

```

*
*   AUTHOR.
*       PITNEY BOWES INC
*       ALL RIGHTS RESERVED.
*
*       PROGRAMS AND DESIGN (C) COPYRIGHT 2013
*
** =====
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-3270 WITH DEBUGGING MODE.
*OBJECT-COMPUTER.  IBM-3270.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
*+++++

*
* PROGRAM = ADDRTEST
*
* PROGRAM TYPE = BATCH DRIVER
*
* APPLICATION = FINALIST
*
* CALLED PROGRAMS:
*   ADDRSCAN - AN EXTERNAL ROUTINE FOR IDENTIFYING
*             ADDRESS LINES
*
* INPUTS:
*   NONE
*
* OUTPUTS:
*   NONE
*
* PROGRAM NARRATIVE:
*   ADDRTEST IS TEST DRIVER PROGRAM.
*   ITS PURPOSE IS TO DEMONSTRATE HOW TO CALL ADDRSCAN.
*
*+++++

*+++++

DATA DIVISION.
*+++++

*+++++

FILE SECTION.
*+++++

*+++++

WORKING-STORAGE SECTION.
*+++++

```

```

01 LOOP-COUNTER          PIC 99.
77 FILLER                PIC X(45)  VALUE
   '** ADDRTEST WORKING STORAGE STARTS HERE **'.
* ADDRPASS-CALL-AREA
  COPY ADDR021D.
77 FILLER                PIC X(43)  VALUE
   '** ADDRTEST WORKING STORAGE ENDS HERE **'.
*+++++*

LINKAGE SECTION.
*+++++*

*+++++*

PROCEDURE DIVISION.
*+++++*

0000-MAIN-LOGIC.
*
*
* ONE TIME INITIALIZATION
* - SET ANCHOR TO -1 (OR HIGH VALUES) FOR FIRST CALL
*
  MOVE HIGH-VALUES TO ADDRPASS-ANCHOR
*
* MAKE MULTIPLE CALLS TO ADDRSCAN
* (IN ACTUALITY, ADDRESSES WOULD BE FROM A DATABASE OR FILE)
*
  PERFORM VARYING LOOP-COUNTER FROM 2 BY -1
    UNTIL LOOP-COUNTER < 1
*
* SETTINGS FOR EACH CALL TO ADDRSCAN (INCLUDING FIRST CALL)
* - SET PROCESSING OPTIONS {ADDRPASS-OPTIONS}
* - SET RETURN LINE ORDER {ADDRPASS-RTN-ORDER}
* - SET (OR CLEAR) ADVANCED PROCESSING OPTIONS {ADDRPASS-TYPES}
* - SET LINE LENGTH {ADDRPASS-LNMX-NUM}
* - SET (OR CLEAR) ALL 6 INPUT LINES {ADDRPASS-PASSED-LINES}
*
  MOVE '9BN' TO ADDRPASS-OPTIONS
  MOVE 'N' TO ADDRPASS-RTN-ORD1
  MOVE SPACES TO ADDRPASS-TYPES
  MOVE 70 TO ADDRPASS-LNMX-NUM
  MOVE 'PITNEY BOWES'      TO ADDRPASS-INLINE1
  MOVE '2200 WESTERN CT'  TO ADDRPASS-INLINE2
  MOVE 'STE 100'          TO ADDRPASS-INLINE3
  MOVE 'LISLE'           TO ADDRPASS-INLINE4
  MOVE 'IL'              TO ADDRPASS-INLINE5
  MOVE '60532-3618'      TO ADDRPASS-INLINE6
*
* DISPLAY INPUT DATA
*
  DISPLAY 'ADDRSCAN INPUT VALUES'

```

```

        DISPLAY 'ADDROPTS:' ADDRPASS-OPTIONS
        DISPLAY 'ADDRRORD:' ADDRPASS-RTN-ORDER
        DISPLAY 'ADDRTYPES:' ADDRPASS-TYPES
        DISPLAY 'LNLENGTH:' ADDRPASS-LNMX-NUM
        DISPLAY 'INLINE1 :' ADDRPASS-INLINE1
        DISPLAY 'INLINE2 :' ADDRPASS-INLINE2
        DISPLAY 'INLINE3 :' ADDRPASS-INLINE3
        DISPLAY 'INLINE4 :' ADDRPASS-INLINE4
        DISPLAY 'INLINE5 :' ADDRPASS-INLINE5
        DISPLAY 'INLINE6 :' ADDRPASS-INLINE6
        DISPLAY ' '
*
* CALL ADDRSCAN
*
        CALL 'ADDRSCAN' USING BY REFERENCE ADDRPASS-CALL-AREA
*
* DISPLAY OUTPUT DATA
*
        DISPLAY 'ADDRSCAN OUTPUT VALUES'
        DISPLAY 'OUTLINE1      :' ADDRPASS-OUTLINE1
        DISPLAY 'OUTLINE2      :' ADDRPASS-OUTLINE2
        DISPLAY 'OUTLINE3(CS)   :' ADDRPASS-OUTLINE3
        DISPLAY 'OUTLINE4(FIRM) :' ADDRPASS-OUTLINE4
        DISPLAY 'URB           :' ADDRPASS-OUTURB
        DISPLAY 'ADDRNDX1      :' ADDRPASS-INDEX1
        DISPLAY 'ADDRNDX2      :' ADDRPASS-INDEX2
        DISPLAY 'ZIP-ZIP4      :' ADDRPASS-ZIP-PLUS-4
        DISPLAY 'LINE TYPES    :' ADDRPASS-TYPES
        DISPLAY 'UNIVERSITY    :' ADDRPASS-UNIV-WORD
        DISPLAY 'DORMITORY     :' ADDRPASS-DORM-WORD
        DISPLAY ' '
        END-PERFORM.
*
* TERMINATE ADDRSCAN
*
        MOVE HIGH-VALUES TO ADDRPASS-OPTIONS
        CALL 'ADDRSCAN' USING BY REFERENCE ADDRPASS-CALL-AREA
        GOBACK.
*
0000-EXIT.
        EXIT.
END PROGRAM 'ADDRTEST'.

```

# 9 - Exceptions Table Option

## In this section

---

What is the Exceptions Table Option?.....	224
Can I Use the Exceptions Table Option?.....	224
Does the Exceptions Table Option Affect CASS?.....	224
Why Should I Use the Exceptions Table Option?.....	224
How Do I Install the Exceptions Table Option?.....	225
How Do I Activate Exceptions Table Processing?.....	225
How Does Exceptions Table Processing Work?.....	230
Exceptions Table Input.....	231
Structures Containing Exceptions Table Information.....	241
Exceptions Table Output.....	241



## What is the Exceptions Table Option?

The exceptions table alters address input fields to enhance Finalist's coding ability. According to the rules you specify, the exceptions table modifies addresses that may be rejected so that the addresses match addresses found in the Finalist Data File. One potential use would be to look for commonly misspelled streets or city names on input and replace the misspelled name with the corrected name. These changes do not affect your input data. Be careful when making exceptions table entries. An exceptions table entry affects all records that meet the selection criteria. Again, the exceptions table does not change the input data, but rather the data that Finalist processes.

## Can I Use the Exceptions Table Option?

The Exceptions Table Option is available to all Finalist customers.

## Does the Exceptions Table Option Affect CASS?

The Exceptions Table processing does not affect CASS certification. Exceptions processing is performed prior to CASS-certified processing.

## Why Should I Use the Exceptions Table Option?

Some of the benefits available to Finalist customers through Exceptions Table processing are:

- Provides a method to change input address fields to increase coding
- Allows you to set rules for address modification using the exceptions table to ensure address matching
- You can use the exceptions table to:
  - Add missing suffixes to addresses
  - Combine two words
  - Correct misspelled street names



- Ignore extraneous address information
- Make address changes within the items listed next:
  - A specific ZIP Code
  - An SCF (first three digits of ZIP Code) area
  - A region (first digit of ZIP Code)
  - All input addresses

## How Do I Install the Exceptions Table Option?

The Exceptions Table Option is installed as part of your standard Finalist installation for all platforms.

## How Do I Activate Exceptions Table Processing?

To activate Exceptions Table processing, you will need to define the fields listed below.

<b>Exception Table Filename</b>	Define the Exceptions Table name.
<b>Number Exception Table Entries</b>	Specify the maximum number of exceptions table entries.

## Methods for Activating Exceptions Table Processing

To activate Exceptions Table processing, use one of the following methods.

<b>pbfn.cfg Configuration File</b>	If you prefer to configure your Finalist installation using global settings, you can define the Exceptions Table fields in your pbfn.cfg configuration file. For more information on the pbfn.cfg file, refer to <a href="#">Configuring Finalist</a> on page 13.
--	---

**Note:** Using pbfn.cfg is not valid for CICS and IMS online.

<b>PBFNSetupDef Structure</b>	If you prefer to configure your Finalist installation using the Finalist structures, you can define the Exceptions Table fields in the PBFNSetupDef structure. For more information on the PBFNSetupDef structure, refer to your Finalist Reference Guide.
-----------------------------------	--

**Workbench or Lookup Tool**

If you prefer to configure your Finalist installation using the Finalist Workbench or Lookup Tool GUI screens, you can define the Exceptions Table fields in the Product Tab on the PBFN Config Setting dialog box. For more information, refer to [Using the Workbench or Lookup Tool to Configure Finalist](#) on page 35.

**Compatibility Interface (CI)**

If you prefer to configure your Finalist installation using the CI, you can define the Exceptions Table fields in the Finalist call area. For more information on the CI, refer to Chapter 3, Using the Compatibility Interface in your Finalist Reference Guide.

## Using pbfncfg to Activate Exceptions Table Processing

To activate Exceptions Table processing using the pbfncfg file, complete the following fields in your pbfncfg file.

**Note:** Using pbfncfg is not valid for CICS and IMS online.

pbfncfg Field	Description
Exception Table Filename	Specify the Exceptions Table file name and path.
Number Exception Table Entries	Specify the maximum number of exceptions table entries. The default value is 1000.

## Using PBFNSetupDef to Activate Exceptions Table Processing

To activate Exceptions Table processing using the PBFNSetupDef structure, complete the following fields in the PBFNSetupDef structure.

PBFNSetupDef Field	Description
cExceptionTblFileName	Specify the Exceptions Table file name and path.  <b>Note:</b> In place of the filename, CICS and IMS online require the use of a load module. Specify the literal "PBFNEXTB", as the load module name, in the cExceptionTblFileName field. Instructions for building an exceptions table load module can be found in <a href="#">Building an Exceptions Table Load Module for CICS and IMS Online</a> on page 228.

PBFNSetupDef Field	Description
cMaxExcpTblEntries	Specify the maximum number of exceptions table entries. The default value is 1000.

## Using the Workbench to Activate Exceptions Table Processing

To activate Exceptions Table processing using the Finalist Workbench:

1. Launch the Finalist Workbench.
2. From the Tools menu, select PBFN Setup.
3. Select the Product tab on the PBFN Config Setting dialog box.
4. Complete the following field on the Product tab.

Field	Description
Exception Table File	Specify the Exceptions Table File file name and path or click on the Browse button (located to the right of the field) to open a dialog box and select the path and file name from a list of existing files.

5. Select the Process tab on the PBFN Config Setting dialog box.
6. Complete the following field on the Process tab.

Field	Description
Number of Exceptions Table Entries	Specify the maximum number of exceptions table entries. The default value is 1000.

7. Click OK to save your settings.

## Using the Compatibility Interface to Activate Exceptions Processing

To activate Exceptions Table processing using the CI, define the appropriate field for your platform in the Finalist call area.

COBOL Field Name/ C Field Name	Field Description
CAEXCPSW cExcpSw	Applies to CICS users only. CAEXCPSW indicates exceptions table processing is used. The field is populated only on the initialization call.

## Using an Exceptions Table in Windows, Unix, and Linux

To use an Exceptions Table in Windows, Unix, or Linux, create a plain text file that contains the data described in [Exceptions Table Input](#) on page 231. Point your pbfncfg or PBFNSetupDef to this file.

## Building an Exceptions Table Load Module for CICS and IMS Online

For information on Finalist exceptions processing, refer to [Exceptions Table Input](#) on page 231. CICS and IMS online Finalist exceptions processing is similar, but the Exceptions File is converted into load module format for performance and usability. The following section describes how to convert the Exceptions File statements as documented in [Exceptions Table Input](#) on page 231 into load module format for use in the CICS and IMS online environments.

To create a custom Exceptions Table load module for CICS and IMS online or to replace a current Exceptions Table with a new one, execute the JCL shown below. This JCL includes the job stream required to build the table and place the table in the CICS and/or IMS online load library.

### ***JCL to Build the Exceptions Table***

The figure below contains JCL to build the exceptions table.

```

1 //STEP1      EXEC PGM=PBFNEXCP
2 //STEPLIB   DD   DSN=hlq.BATCH.LOAD,DISP=SHR
3 //SYSPRINT  DD   SYSOUT=*
```

```

4 //SYSLOG DD SYSOUT=*
5 //SYSUDUMP DD SYSOUT=*
6 //EXCPIN DD *
  LIMANE MAIN RP60187 (Sample Control Card)
7 /*
8 //EXCPOUT DD DSN=&&TEMP,UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
  // DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(,CATLG)
  /*
9 //ASMCC EXEC PGM=ASMA90,
  // PARM=(ASA,DECK,NOBJECT,RENT,BATCH,NORLD)
10 //SYSIN DD DISP=SHR,DSN=*.STEP1.EXCPOUT
11 //SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
12 //SYSUT1 DD UNIT=(SYSDA,SEP=SYSLIB),SPACE=(CYL,(1,1))
13 //SYSPUNCH DD DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(1,1)),
  // DCB=(BLKSIZE=3200,LRECL=80,RECFM=FB)
14 //SYSPRINT DD SYSOUT=*
  /*
15 //LKED EXEC PGM=IEWL,AMODE=31,RMODE=ANY
  // PARM='LIST,MAP,RENT,XREF,AMODE=24,RMODE=24'
16 //SYSLIB DD DUMMY
17 //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
18 //SYSPRINT DD SYSOUT=*
19 //SYSLMOD DD DISP=SHR,DSN=hlq.xxxx.LOAD
20 //SYSLIN DD DSN=*.ASMCC.SYSPUNCH,DISP=SHR
  // DD *
  NAME PBFNEXTB(R)
  //

```

A number appears to the left of each statement that contains a user-defined variable. The numbers correspond to the following descriptions.

**Table 41: Exceptions Table JCL**

Statement Number	Description
1	This statement executes PBFNEXCP to process the Exceptions File created. This program processes the Exceptions Table control cards and creates an assembly listing that is passed to the following assemble and link edit steps. The second step performs the assembly of the program. The output from the assembly step is then passed to the third and final step. The linkage editor creates a load module and contains all statements contained in the original Exceptions File as input to the first step. This load module must be put in the CICS and/ or IMS online Finalist load library where the module can be accessed from CICS and/or IMS online transactions as well as any IMS BMP transactions calling FINALB modules. If this module is empty during processing, exceptions processing is ignored.
2	This statement identifies the load library containing your batch Finalist programs.
3	This statement identifies the print class for system messages or error messages generated during the execution program.

Statement Number	Description
4	Include this statement if you require system messages to be written to the system log.
5	This statement identifies the print class for a system dump in case of an ABEND.
6	This statement defines the input for the program. The input cards are the same as those used for Exceptions Table processing in the batch environment. For more information on Exceptions Table input, refer to <a href="#">Exceptions Table Input</a> on page 231.
7	This statement indicates the end of the input file.
8	This statement defines the output file created by the program. This file contains Assembler statements. These statements become the Exceptions Table in load module format for use in the CICS and/or IMS online environment with the Finalist CICS and/or IMS online system.
9-14	These statements execute the Assembler program using the output file from the first step as input. This step's output is the final step's input.
15-20	These statements execute the linkage editor using the output from the previous step as input. Output from this step is the Exceptions Table in load module format. It is important that the library referenced in statement 19 is the library in which you have loaded your Finalist IMS programs. The DSN parameter in statement 20 must appear as shown. If not, the Finalist CICS or IMS system cannot reference the Exceptions Table.

## How Does Exceptions Table Processing Work?

Finalist performs Exceptions Table as described below.

1. You create an exceptions table in a text format using any editor.
  - Note:** ALL exceptions should be in alphabetical order (based on the ARGU field and then the L# field).
2. During processing of your input file, Finalist uses your exceptions table file to apply changes to addresses in your input file that could be rejected to ensure a match to the Finalist Data File.
3. The exceptions table does not change the input data, but rather the data that Finalist processes.
4. The Address Detail Report indicates whether an exceptions table entry was used to process an address.

5. The number of entries in your exceptions table displays on the Finalist Batch Report. If exceptions table processing is not activated, the “No. Exception Table Entries” field displays OFF. The Exceptions Table filename also displays on the Finalist Batch Report.

## Exceptions Table Input

To use an exceptions table, you can use any text editor to create a file in a text format. Record lines must be no more than 80 characters in length. The following table describes the fields available for an exceptions table entry. All field positions need to be specified exactly as shown.

**Table 42: Exceptions Table Entry Fields and Positions**

Field Name	Position	Description
L#	1-2	This required field designates the address line that the entry affects. Possible values for L# are described next.  L1 = First address line L2 = Second address line L3 = City/state line
ARGU	3 - 17	This required field contains a one-word argument (keyword) or the first word of a two-word argument for which you want Finalist to search.
SUBST A	18 - 32	This required field contains the word that replaces, or is used with, ARGU.  For an IG-type entry, leave this field blank. Blanks are only valid with an IG-type entry. For a ++-type entry, this field becomes the second word of a two-word argument.
ATTRIBUTE	33 - 34	This required field contains the two-digit code to supply more information about the substitute word or the change being made on the argument word.
ZIP Region (first digit of ZIP Code) SCF (first 3 digits of ZIP Code) ZIP Code	35 35 - 37 35 - 39	This optional field specifies the region, SCF, or ZIP Code for which the exception logic works. If this field is blank, exception logic works for all input data.

Field Name	Position	Description
SUBST B	40 - 54	This optional field contains the replacement for ARGU and SUBST A in a ++-type exceptions table entry. This optional field is also used in the R+ operation, which concatenates SUBST B with SUBST A as a replacement for ARGU.

The following table provides a list of available attributes for each address line.

**Table 43: Attributes for Address Lines**

Address Line(s)	Attribute	Description
L1 and L2	DR	Replace argument word with substitute word. The substitute word is a directional.
	I2	Insert the substitute word as a unit designator before the target word.
	ID	Insert the substitute word as a directional before the argument word.
	IN	Insert the substitute word after the argument and make it part of the street name.
	IP	Insert the substitute word as a directional after the argument word.
	IS	Insert the substitute word as a suffix after the argument word.
	IU	Insert the substitute word as a unit designator after the target word.
	SF	Replace argument word with substitute word. The substitute word is a suffix.
	UD	Replace argument word with the substitute word. The substitute word is a unit designator.
	RR	Replace argument word with the substitute word. The substitute word is a Rural Route (RR) word.
L3	IC	Insert the substitute word as a city before the argument state abbreviation.
	ST	Insert the substitute word as the state abbreviation after the argument city word.



Address Line(s)	Attribute	Description
L1, L2, and L3	++	Combine the argument and substitute words to make one word.
	CT	Attach (concatenate) the substitute word to the following word in the input record.
	IG	Ignore the argument word.
	R+	Replace argument word with the concatenation of substitute word A and substitute word B.
	RP	Replace argument word with substitute word.

## Entry Guidelines

Follow these guidelines when making exceptions table entries.

- There is a default of a 1,000-entry maximum for the exceptions table. Sort the entries sequentially by argument, in ascending order. Alphabetic arguments should precede numeric arguments.

Example

```

1.....18.....33.....
1) L3ADDISON      IL          ST60101
2) L1AMERICAN    LEGION     IN02131
3) L1BEAVER      TER        IS01701
.
.
15) L1U          U          CT53913
16) L2ZEPHYR    N          ID79963
17) L135TH      W          IP99821
    
```

- If you use SCFs, regions, or ZIP Codes to limit alterations for an exceptions statement, you must pass a ZIP Code field as input to Finalist.
- For any function that performs a lookup on city/state, L3 alterations could be useful.

## Sample Entries

Sample applications using exceptions table logic are shown next.

**Table 44: Sample Applications of Exceptions Table Logic With ++ Attribute**

Attribute	Description
++	Combine the argument and substitute words to make one word.
Sample Application 1 Input Exception	<p>BAYVIEW exists in ZIP Code 60532. Your input file lists BAY VIEW. With the following exceptions statement, Finalist looks for records with the argument word BAY and the substitute word VIEW on the same line. The single word BAYVIEW (coded after the ZIP Code field) replaces both words. 13 BAY VIEW 60532 1BAY VIEW ++60532BAYVIEW</p> <p>This attribute combines words more specifically than the CT attribute. Instead of combining the argument word with whatever follows, ++ looks for specific word pairs.</p>
Sample Application 2 Input Exception	<p>STATE ROUTE 59 exists in ZIP Code 60185. It is commonly known as, and is listed in your input file as, ILLINOIS 59. With the following exceptions statement, the Finalist system looks for records with the argument word ILLINOIS and the second argument 59 on the same line. The single string STATEROUTE59 (coded after the ZIP Code field) replaces both words. 1N601 ILLINOIS 59 L1ILLINOIS 59 ++60185STATE ROUTE 59</p> <p>This example shows when you should use the ++ attribute instead of the RP (replace) attribute. Your purpose in using the ++ attribute is to replace the word ILLINOIS only when it is paired with the trailing word 59. Here, the ++ attribute replacement for ILLINOIS 59 would be STATE ROUTE 59.</p> <p>If you use the RP attribute to replace Illinois with State Route, Illinois will be changed to State Route whenever Illinois is encountered (for example, Illinois Ave. becomes State Route Ave).</p>

**Table 45: Sample Applications of Exceptions Table Logic With CT Attribute**

Attribute	Description
CT	Concatenate the substitute word (which is also the argument word in most cases) with the word that follows it on input.

Attribute	Description
Sample Application 1 Input Exception	<p>BROADVIEW exists as a street in ZIP Code 60148. Your file lists this street as BROAD VIEW. You could use the following exceptions statement to correct the problem: 10 BROAD VIEW 60148 L1BROAD BROAD CT60148</p> <p>The system can now locate the street on the Data File and code the record. Finalist doesn't alter the input record, but will return BROADVIEW in both the returned street name field and the label return area.</p> <p><b>Note:</b> All occurrences of BROAD will join with street name words that follow BROAD in the specified ZIP Code.</p>
Sample Application 2 Input Exception	<p>You can also use CT to make multiple changes. In Example 1 above, if your file spells BROADVIEW incorrectly, you can change the spelling and combine the words. 10 BOARD VIEW 60148 L1BOARD BROAD CT60148 The system replaces BOARD with BROAD and concatenates it to VIEW. Finalist returns BROADVIEW in the returned street name field and the label return area.</p>

**Table 46: Sample Application of Exceptions Table Logic With DR Attribute**

Attribute	Description
DR	Replace argument with substitute and identify the substitute word as a directional.
Sample Application Input Exception	<p>Some of the records in your file incorrectly use NROTH to represent NORTH. The following exceptions statement corrects the problem: 12 NROTH MAIN ST 60148 L1NROTH N DR The system replaced NROTH with N. Finalist will then be able to code to N MAIN ST. Finalist will then return N in the predirectional field of the Finalist return area and the label return area. If Finalist replaces a post-directional, it returns in the post-directional field and the label return area.</p>

**Table 47: Sample Application of Exceptions Table Logic With I2 Attribute**

Attribute	Description
I2	Insert the substitute word as a unit designator before the target word.
Sample Application Input Exception	<p>The address 730 GRAND CNCRS AB is missing APT before the unit number (AB). Create the following exceptions table entry to insert APT before AB in ZIP code 10451: 730 GRAND CNCRS AB 10451 L1AB APT I210451 The exception logic inserts APT to effectively process the address as 730 GRAND CNCRS APT AB.</p>

**Table 48: Sample Application of Exceptions Table Logic With ID Attribute**

Attribute	Description
ID	Insert a directional before street name (predirectional).
Sample Application Input Exception	The address 200 BERKSHIRE needs a directional to code. This exceptions statement adds the proper directional: 200 BERKSHIRE AVE 60148 L1BERKSHIRE E ID60148 The system can now code the record to E BERKSHIRE. The input address remains BERKSHIRE. Finalist returns E in the return area's predirectional field and the label return area.

**Table 49: Sample Application of Exceptions Table Logic With IG Attribute**

Attribute	Description
IG	Ignore the argument (here you use no substitute word).
Sample Application Input Exception	COUNTRY CORNERS RD exists in ZIP Code 01002 and the USPS lists it this way. However, your file lists this street as HWY COUNTRY CORNERS RD. The following exceptions statement makes the system recognize HWY COUNTRY CORNERS RD as COUNTRY CORNERS RD: 31 HWY COUNTRY CORNERS RD 01002 L1HWY IG01002 The system now ignores HWY during coding. Your input record remains intact. The street name that Finalist returns in the return area is COUNTRY CORNERS RD. Finalist also correctly standardizes the label return area.

**Table 50: Sample Application of Exceptions Table Logic With IN Attribute**

Attribute	Description
IN	Insert the substitute word after the argument and use it as part of the street name.
Sample Application Input Exception	AMERICAN LEGION HWY exists in ZIP Code 02131. Your input file has it listed as AMERICAN HWY. The following exceptions statement corrects the problem: 865 AMERICAN HWY 02131 L1AMERICAN LEGION IN02131 The record now codes as AMERICAN LEGION HWY and Finalist returns that name in the return area street field and the label return area.

**Table 51: Sample Application of Exceptions Table Logic With IP Attribute**

Attribute	Description
IP	Insert the substitute word as a directional after the street name (post-directional).
Sample Application Input Exception	The address 1 DAVIS RD needs a post-directional to code. Use the following exceptions statement to add the necessary post-directional: 1 DAVIS RD 06371 L1DAVIS E IP06371 The exception logic lets the record code to DAVIS RD E. The input remains unchanged. Finalist returns E in the post-directional field and in the return area label area.

**Table 52: Sample Application of Exceptions Table Logic With IS Attribute**

Attribute	Description
IS	Insert a suffix after the street name.
Sample Application Input Exception	BEAVER TER CIR exists in ZIP Code 01701. Your input record lists BEAVER CIR and will not code accurately. The following exceptions statement corrects the problem: 100 BEAVER CIR 01701 L1BEAVER TER IS01701 The record now codes correctly without changing the input record. If you are using parsed fields, this may be a concern. However, the suffix1 field in the Finalist return area contains TER and the suffix2 field contains CIR. The label return area contains 100 BEAVER TERRACE CIR.

**Table 53: Sample Application of Exceptions Table Logic With IU Attribute**

Attribute	Description
IU	Insert the substitute word as a unit designator after the target word.
Sample Application Input Exception	The address 730 GRAND CNCRS AB is missing APT after the street word (CNCRS) and before the unit (AB). Create the following exceptions table entry to insert APT following CNCRS in ZIP code 10451: 730 GRAND CNCRS AB 10451 L1CNCRS APT IU10451 The exception logic inserts APT to effectively process the address as 730 GRAND CNCRS APT AB.

**Table 54: Sample Applications of Exceptions Table Logic With RP Attribute**

Attribute	Description
RP	Replace the argument word with a substitute.

Attribute	Description
Sample Application 1 Input Exception	LEXINGTON AVE exists in New York City but is commonly known as LEX AVE. Your file contains this reference, but is not a sufficient match for LEXINGTON. To code this record, use the following exception statement: 142 LEX AVE 10016 L1LEX LEXINGTON RP10016 Finalist will leave LEX in your input record. It will return LEXINGTON in the return area street field and the label area. Reason code 5 will have a value of 2.
Sample Application 2 Input Exception	If your file misspells a state abbreviation (for example, II for IL), you can use the following exceptions statement to correct the problem: 500 WEST RD LOMBARD II L3II IL RP Without altering the input data, the exception logic uses IL instead of II. The FINAL subroutine codes the record. Since II is not a valid city or state, no ZIP Code, SCF, or region is necessary.
Sample Application 3 Input Exception	If your file contains a non-standard city abbreviation (for example, CHG for CHICAGO), the following exceptions statement corrects the problem: 600 W MADISON ST CHG IL 60602 L3CHG CHICAGO RP606 The city of Chicago uses the SCF 606 exclusively. If you do not pass a ZIP Code to Finalist, you do not need an SCF in the exceptions statement.
Sample Application 4 Input Exception	If a directional and a street name are combined (for example, NORTHVIEW), you need two exception entries to correct the problem. 3109 NORTHVIEW ROCKFORD IL 61107 L1NORTHVIEW VIEW RP61107 L1NORTHVIEW N ID61107 The first entry replaces the double word with the correct primary street name. The second entry inserts the correct directional.

**Table 55: Sample Application of Exceptions Table Logic With R+ Attribute**

Attribute	Description
R+	Replace the argument word with a substitute.
Sample Application Input Exception	ADAM CLAYTON POWELL JR BLVD exists in New York City but does not fit into the input address file space allowed for the street name. Create the following exceptions table entry to expand ACP to ADAM CLAYTON POWEL JR in ZIP code 10026: 1800 ACP BLVD 10026 L1ACP ADAM CLAYTON POR+10026WELL JR Finalist leaves ACP in your input record and returns ADAM CLAYTON POWELL JR in the return area street field and the label area.

**Table 56: Sample Application of Exceptions Table Logic With RR Attribute**

Attribute	Description
RR	Replace the argument word with a substitute. The substitute word is a Rural Route.

Attribute	Description
Sample Application Input Exception	RURAL ROUTE is misspelled as RVRALRTE in ZIP 77622. Create the following exceptions table entry to convert RVRALRTE to RR in ZIP code 77622: RVRALRTE 2 BOX 703 77622 L1RVRALRTE RR RR77622 Finalist leaves RVRALRTE in your input record and returns RR in the return area street field and the label area. In this sample, RR 2 BOX 703 will LACS <sup>Link</sup> convert to 12396 E HAMPSHIRE RD.

**Table 57: Sample Application of Exceptions Table Logic With SF Attribute**

Attribute	Description
SF	Replace the argument word with the substitute words which the system will identify as a suffix.
Sample Application Input Exception	DRIVE is a street suffix and DR is its recognized abbreviation. DV, a non-recognized abbreviation, appears in your file in both address lines. The following exceptions statements help Finalist recognize DV as a suffix when it appears in either address line. This exceptions statement is especially important when the suffix is crucial to coding. 2100 BELLEAU WOODS DV 60187 L1DV DR SF L2DV DR SF The system will now recognize DV as the suffix DR and Finalist can code these records. Please note that because both DR and CT are valid for the above range, the exceptions logic could affect coding if you altered the improper suffix. Using the above exceptions logic, Finalist will leave DV in your input record, and return DR in the Finalist return area suffix field. The label area will also contain this correctly standardized suffix.

## City/State Only Lookups

The following two exceptions table attributes are useful for city/state only lookup. Determining whether to use region, SCF, or ZIP Code limitations in the exceptions statement depends on whether you'll pass the input ZIP Code field with the city/state field. If you are not including a ZIP Code in the input, do not use a ZIP Code in the exceptions statement.

**Note:** In these examples, Finalist alters any word(s) located in the specified ZIP Code that contains the argument word.

**Table 58: Sample Application of Exceptions Table Logic With IC Attribute**

Attribute	Description
IC	Insert the substitute word as the city before the argument state word.

Attribute	Description
Sample Application Exception	Use this type of alteration only if you're passing ZIP Codes in the ZIP Code field and passing the city/state field (since you only want to insert a city for certain ZIP Codes within the state). For example, a few records at ZIP Code 60104 have only IL in the city/state field. If you only use a city/state lookup for your file, this record would fail because the city is missing. Use the following exception statement to insert a city in your record: L3IL BELLWOOD IC60104 This statement inserts the city BELLWOOD in all your IL records for the ZIP Code 60104. Function 7 is now able to code the address. Functions 4, 5, and 6 do not need a city name to code.

**Table 59: Sample Application of Exceptions Table Logic With ST Attribute**

Attribute	Description
ST	Insert this as the state.
Sample Application Input Exception	If you perform a city/state-only lookup, some records might fail because the state name is missing. Use an exceptions table entry to insert the state for these records. Since identical city names exist in different states, it is safest to restrict this alteration to a specific SCF or ZIP Code. 22 E LAKE ST ADDISON 60101 L3ADDISON IL ST60101 The system uses IL for the state and your input city of ADDISON. The record codes.

**Table 60: Sample Application of Exceptions Table Logic With UD Attribute**

Attribute	Description
UD	Replace the argument word with a substitute and flag as a unit designator.
Sample Application Exception	You use the word LT as an abbreviation for the unit designator LOT. Finalist will leave the word spelled as LT but flag the word as a unit designator. L1LT LT UD



# Structures Containing Exceptions Table Information

The structures listed below include data to facilitate Exceptions Table processing. For detailed information on these structures, refer to your Finalist Reference Guide.

- PBFNAddressDataDef
- PBFNIMSSetupDef
- PBFNSetupDef

## Exceptions Table Output

The following sections describe the output generated during Exceptions Table processing.

### Exceptions Table Information on Finalist Batch Report

The Finalist Batch Report displays the Exceptions Table settings from the pbfncfg configuration file or the PBFNSetupDef structure and the number of entries in your Exceptions Table. If Exceptions Table processing is not activated, the “No. Exceptions Table Entries” field displays OFF. For detailed information on the Finalist Batch Report, refer to [Using the Finalist Reports](#).

### Exceptions Table Information on the Address Detail Report

The Address Detail Report indicates whether an Exceptions Table entry was used to process an address. For detailed information on the Address Detail Report, refer to [Using the Finalist Reports](#).

# 10 - CASS Certification

## In this section

---

What is CASS Certification?.....	243
Why is CASS Certification Important for Finalist?.....	244
Types of Certification.....	244
What is the Address Management System II (AMS II)?.....	245
National Customer Support Center (NCSC).....	245



# What is CASS Certification?

The USPS designed the CASS certification process to test postal coding software and to influence companies to produce high-quality, addressed mail. High-quality addressed mail is mail that contains all appropriate address elements standardized in a manner regulated by the USPS. The CASS certification process tests the accuracy of the five-digit ZIP Codes, ZIP + 4 codes, and Delivery Point Barcodes that appear on mail pieces. The USPS offers postal discounts to those mailers complying with USPS addressing regulations. To qualify for these discounts, a mailer must use a CASS-certified postal coding software package to assign ZIP Codes, ZIP + 4 codes, and Delivery Point Barcodes to mailings. Software vendors supplying postal coding software must pass a test designed by the USPS in order to have their software designated as CASS-certified.

The CASS certification test is divided into two stages. The Stage 1 File is a test with answers. This stage allows processing results to be compared with those provided by the USPS and to resolve any problems before proceeding to Stage 2 testing.

Stage 2 is the actual certification process (graded by the National Customer Support Center). The Stage 2 File is a test without answers. Vendors must process the USPS-supplied file containing 150,000 address records through their software and return the file to the USPS for grading. Vendors must achieve a USPS-specified level of accuracy to be CASS-certified. The table below details the levels of CASS certification and the accuracy requirement for each level.

**Table 61: USPS CASS Certification Levels and Percentages**

Certification Level	Required Accuracy Level
ZIP	98.5%
Carrier Route	98.5%
ZIP + 4	98.5%
Delivery Point Barcode	100%
eLOT	100%
Perfect Addresses	100%

CASS certification is valid through the end of the current CASS cycle. Vendors must apply and achieve a specified level of accuracy for each CASS cycle. In addition to the above overall percentages, address matching software must attain a 98.5% rating in all required categories.

# Why is CASS Certification Important for Finalist?

The Finalist software is CASS-certified. This feature assures Finalist customers that their mailings meet USPS regulations and qualify for postal discounts achieved through high-quality addressed mail.

## Types of Certification

The USPS performs the following types of mailing certifications:

**Table 62: USPS Certification Types**

Certification Type	Description
CASS Certification	The USPS uses CASS certification to certify postal coding software packages. This type of certification applies to the Finalist software.
Vendor Certification	The software vendor certifies their postal coding software using the Stage 1 and Stage 2 tests. Precisely CASS-certifies every release of Finalist before any release of the software to customers.
Client Certification	A customer may certify their Finalist integration if they wish. If you decide to certify your Finalist integration, Precisely is not responsible for support of your USPS certification process and audits.  <b>Note:</b> Stage test certifications require the use of special static databases. Please contact Technical Support to obtain these databases if you are trying to certify using Finalist.
Mailing List Certification	A mailer can certify a mailing list by processing address files through CASS-certified software (such as Finalist) every six months. This procedure is required for all mailings receiving postal discounts. The USPS requires the USPS Form 3553 (CASS Summary Report) as proof that a mailing is CASS-certified. For more information on the USPS Form 3553 (CASS Summary Report), refer to <a href="#">USPS Form 3553 (CASS Summary Report)</a> on page 176.
PAVE Certification	The USPS uses PAVE certification to certify presort software packages.

Certification Type	Description
Z4Change Certification	The USPS uses Z4Change certification to certify ZIP + 4 change software. This type of certification applies to the Precisely Z4Select software.

## What is the Address Management System II (AMS II)?

AMS II is the USPS system that produces the City/State File and ZIP + 4 File. Precisely uses these files to create the Finalist data files.

## National Customer Support Center (NCSC)

The NCSC is the USPS customer support center. This support center is located in Memphis, Tennessee. The NCSC includes the four departments described next.

### General Customer Service

The General Customer Service department is responsible for:

- Basic Address Information Systems product information
- AIS product and postal guide orders
- Reporting problems in AMS II files

### CASS Certification

The CASS Certification Department is responsible for:

- CASS certification
- Z4Change certification
- Ordering Stage 1 and Stage 2 files
- Any questions regarding certification

## AIS File Quality

The AIS File Quality Department is responsible for quality measures for the following files:

- AMS II City/State
- AMS II ZIP + 4
- Z4Change
- ZIPMOVE
- Dropship
- Enhanced Line of Travel (eLOT)

## PAVE Certification

The PAVE Certification Department is responsible for PAVE certification.



2 Blue Hill Plaza, #1563  
Pearl River, NY 10965  
USA

[www.precisely.com](http://www.precisely.com)

© 1984, 2020 Precisely. All rights reserved.