



Finalist[®]

Compatibility Interface (CI) Migration

This document contains information to help you migrate from the Finalist[®] Compatibility Interface (CI) to the Native Finalist[®] API.

Contents:

Purpose	2
Finalist [®] History	2
Finalist [®] Provided Drivers and Other Programs	3
Migrating from CI to Native	4
Initialize - 0 Call	5
Process - 4, 5, 6, and 7 Calls	14
Terminate - 9 Call	40
Recent Finalist Changes	46
Planning for the Future with Finalist	47
Technical Support	47

UNITED STATES

<http://www.pitneybowes.com/us>

Technical Support

<https://www.pitneybowes.com/us/support.html>

©2019 Pitney Bowes Inc.



Purpose

This document describes how to migrate from the Finalist[®] Compatibility Interface (CI) to the Finalist[®] native interface. The Finalist[®] CI is available for the z/OS platform only. The Finalist[®] native interfaces are shared across all platforms. Programs (applications) written to use the native interface easily migrate from one platform to another.

This document focuses on COBOL customers who represent the majority of the Finalist[®] customer application driver base on z/OS. For Assembler (BAL) customers, the same concepts apply but Assembler field names must be used instead of the COBOL equivalent.

This document is designed to serve all Finalist[®] customers and about 95% of their applications. The remaining 5% includes CICS or IMS specific code and/or LPAM (database access routines). This document does not address the latter group.

The information in this document is current as of publication date. Changes may be made after publication and this document may not address all issues applicable to your specific site. Finalist[®] Technical Support and/or Development are available to answer your questions. However, if the task is complex, you can engage Pitney Bowes Professional Services for assistance.

Finalist[®] History

Finalist[®] was originally developed as a mainframe (Assembler - BAL) product designed and built to perform USPS[®] CASS[™] processing on customer's addresses as the first step in the process to obtain USPS[®] postal discounts using PAVE[™]. Additional Finalist[®] products were developed for the Windows and UNIX platforms. Finalist[®] was also adapted for the CICS and IMS mainframe environments.

To optimize performance and take advantage of technical advancements, the Finalist source code base was unified on all platforms and Finalist[®] was re-written into C. The new Finalist[®] interface merged the Assembler code that called programs (APIs) like FINAL, and the Windows and UNIX applications that called programs (APIs) like ProcessAddress.

To minimize required changes for mainframe customers, the Assembler interface was kept through the Compatibility Interface (CI). Support for the CI was originally planned to last two years, to provide time for the mainframe platform to rewrite to the new (native) interface with routines like PBFNInit, PBFNProcess, and PBFNTerminate. Windows and UNIX users were required to re-write their code immediately.

Finalist[®] Provided Drivers and Other Programs

This section provides support information for Finalist drivers and programs that existed before the Finalist native interface was created.

LPFNCC

The Assembler version of Finalist included the LPFNCC driver. LPFNCC was a control card based driver that would read an input file, process that file through the Finalist engine, and write the results to an output file. When Finalist was re-written, a new driver was created called FINALIST. The FINALIST driver performs the same functions as the LPFNCC driver but internally uses the native Finalist interface. LPFNCC is no longer supported.

LPFNIORP and LPFNSMPL

The CI includes additional interface programs like LPFNIORP and LPFNSMPL that generate specialized reports based on the CI. These programs provide extract reports based on sample records in the input file. The Finalist native interface does not support the LPFNIORP and LPFNSMPL programs.

Z4SELECT™

The USPS provided a Z4CHANGE file that identified areas where address changes occurred. You could use Z4SELECT to pre-process your file. If Z4SELECT processing determined that no address changes had occurred in the ZIP Code where an address had previously been coded, CASS processing was not required. The Finalist native interface does not support the Z4SELECT interface.

ADDRSCAN

AddrScan runs separately from Finalist and continues to be fully supported. Originally, AddrScan processing defaulted to returning a single address line. Modern AddrScan processing returns two address lines. Finalist selects the better of the two lines. Also, AddrScan now includes enhanced processing to skip the Firm, Urbanization (URB), and City/State/ZIP (CSZ) lines if these are in a known, fixed location. Pitney Bowes continues to fully support AddrScan.

FNCIRCAC

FNCIRCAC is a native interface only program that converts the Finalist native Address Information values into the CI style Return Code, Reason Codes, and Address Information Codes. The Finalist native values are "bit flags". COBOL cannot easily handle the bit flags so an additional helper routine is also provided called PBITBYTE. PBITBYTE converts 8 bits of a byte to 8 bytes that can be easily interrogated by a COBOL application. Later sections of this document provide more information on FNCIRCAC.

Migrating from CI to Native


The original intent was for the CI to be a short lived, temporary resource that would provide Finalist mainframe customers with time to migrate, test, and become familiar with the new Finalist native interface. This section describes the easiest path to migrate from the CI to the native interface. The Finalist CI has a variety of names that are essentially aliases of each other. These names include FINAL, FINALOL, FINALI, LPFN000, and others. The CI should not be confused with the Finalist LPAM database access routines. LPAM is a separate callable interface (API) that is not addressed in this document.

The Finalist CI passes one parameter to the interface. The callarea parameter is defined by the COBOL copybook LPFNCL01 (BAL uses the copybook LPFNCL04, C uses LPFNCL0C). The callarea is 10,000 bytes long. This parameter serves multiple definitions depending on what function is being performed. By contrast, the Finalist native interface uses dedicated parameters for each function call (API).

The CI consists of three calls.

CI Call	Description
Initialize (0)	Initializes the Finalist environment based on setup parameters.
Process (4, 5, 6, or 7)	Cleanses an individual address and returns information about the address. In early Finalist versions, the 4, 5, 6, and 7 calls were different. These calls are now identical.
Terminate (9)	Terminates the Finalist environment, frees storage, and produces reports.

Each CI call is described in detail in the following sections.

 Prior to the 8.3 release of Finalist, COBOL applications on the mainframe were required to compile with special options to be able to call the Finalist native interface. Starting with the 8.3 release, that limitation was eliminated.

Initialize - 0 Call

The CI Initialize (0) call is called PBFNInit (PBFNINIT) in the native interface. The parameter passed in to PBFNInit is called SetupDef (PBFNGCFG copybook). All CI initialization options are covered in the SetupDef structure along with additional options that are not available in the CI. For more information on how to migrate from the Finalist callarea to the SetupDef structure, please refer to "[Initialize - 0 Call](#)" on [page 5](#).

Process - 4, 5, 6, and 7 Calls

The CI Process (4, 5, 6, or 7) call is called PBFNProcess (PBFNPROC) in the native interface. The parameter passed into PBFNProcess is called AddressData (PBFNADRS copybook). PBFNProcess can optionally accept two essentially identical parameters. If one AddressData parameter is passed in, AddressData contains the input address when PBFNProcess is called. The same fields are updated in place with the returned address information when PBFNProcess has ended. Optionally, you can pass a second AddressData structure to PBFNProcess. In this case, the first AddressData contains your unchanged input address. The second AddressData (PBFNRRTN copybook) is updated with the returned address information. For more information, please refer to "[Process - 4, 5, 6, and 7 Calls](#)" on [page 14](#).

Terminate - 9 Call

The CI Terminate (9) call is called PBFNTerminate (PBFNTERM) in the native interface. From zero (0) to ten (10) parameters can be passed to the PBFNTerminate call depending on the information you want returned.

Initialize - 0 Call

The CI Initialize (0) call is called PBFNInit (PBFNINIT) in the native interface. The parameter passed in to PBFNInit is called SetupDef (PBFNGCFG copybook). Details of how to migrate from the Finalist callarea to the SetupDef structure are covered below. All CI initialization options are covered in the SetupDef structure along with additional options that are not available in the CI.

The PBFNInit API initializes the Finalist environment and generates a return code to indicate success or failure. A successful call generates a return code of one (1). Other positive numbers indicate success but include a warning that something may not be right. For example, the Finalist database may be about to expire. A zero (0) or negative value (-1) indicates a failure of the initialization. In all cases, a PBFNTerminate call should ultimately be done to ensure all storage allocated by Finalist has been released. A call to PBFNProcess should not be done if PBFNInit generates a zero (0) or negative (-1) return code.

There are two methods available for using PBFNInit to initialize the Finalist environment.

1. You can pass in (or allow to default) all parameters in the SetupDef structure, or you can tell Finalist to read in a configuration file (typically DD:PBFNCFG) or a combination of both. This is controlled via the PBFN-GCFG-LOADSETUP field. Any field passed in via the SetupDef structure will override the matching option in DD:PBFNGCFG. You can use this to force required options (for example, CASS parameters) while allowing others (for example, report DD names) to be modified dynamically.
2. In the second option, applications initialize the CI using the SetupDef structure. This is a hybrid solution and common where applications are already using options that do not exist in the CI. It is mentioned here in case you encounter this situation in your code. For that interface, COBOL applications would specify FINAL-USE-SETUP-Y or FINAL-USE-SETUPDEF = Y. The CI then uses the pre-populated PBFN-GCFG-SETUP as part of the CI parameter file instead of the CI fields when internally calling PBFNInit. In that case, the application logic is already using PBFN-GCFG-SETUP and therefore no additional cross-mapping needs to be done to directly call PBFNInit.

The next section covers how to convert many, but not all, fields from the "0" callarea to the SetupDef structure. First, you must initialize the SetupDef structure. In COBOL, use this statement:

```
INITIALIZE PBFN-GCFG-SETUP.
```

Many options in LPFNCL01 (CALLAREA) are specified as Y|N. In SetupDef (PBFNGCFG), the same options are ON|OFF.

Migrating the Initialize (0) Call

This migration table maps the CI Initialize callarea to the Finalist native APIs and structures. Most of the CI information in this table maps to the Finalist native PBFNInit call. Section B information on the USPS Form 3553 (CASS Summary Report) is passed in the PBFNReportData (PBFNRDAT) structure of the PBFNTerminate call as noted below. The native structures, fields, and any special notes are identified in the third column.

Finalist CI Initialize (0) Callarea Fields (Part 1 of 7)

COBOL (LPFNCL01) C (LPFNCL0C) Assembler (LPFNCL04)	Position	CI Structure/Field/Notes
FINAL_FILLER LAnchorPointers \$REREGS	1-12	
FINAL-FUNCTION-CODE cFunctionCode[0] FUNCODE	13-13	

Finalist CI Initialize (0) Callarea Fields (Part 2 of 7)

COBOL (LPFNCL01)		
C (LPFNCL0C)		
Assembler (LPFNCL04)	Position	CI Structure/Field/Notes
FINAL-FUNCTION-OPTION n/a CATAILOR	14-16	
CALL-INIT CallInit CALLINIT	17-28	
INITMOD CalnitMod INITMOD	17-24	
INITBUFF LInitBuff INITBUFF	25-28	PBFN-GCFG-CACHE-SIZE PBFNSetupDef.cCacheSize
CALL-CICS CallCics CALLCICS	17-28	
CAMODNAM CaModName CAMODNAM	17-24	
CAEXCPSW CExcpSw CAEXCPSW	25	
CALL-IMS CallImS CALL-IMS	17-28	
IMS-INITMOD IMS-INITMOD	17-24	
IMS-INITDAT pPSBData @PBSDATA	17-20	PBFNSetupDef.pPSBData PBFN-GCFG-PPSBDATA
IMS-INITCITY pPSBCity @PBSCITY	21-24	PBFNSetupDef.pPSBCity PBFN-GCFG-PPSBCITY
IMS-ALT-BLKSIZE LAItBlkSz ALTBKLSZ	25-28	

Finalist CI Initialize (0) Callarea Fields (Part 3 of 7)

COBOL (LPFNCL01)

C (LPFNCL0C)

Assembler (LPFNCL04)

Position	CI Structure/Field/Notes
FINAL-CNFIG-ID caConfigID CNFIGID	29-36 PBFN-GCFG-CASS-CONFIG-ID PBFNSetupDef.cCASSConfiguration If your configuration setting is AAR, cAssignAbbrevCity=ON. If your configuration setting is AAJ, cAssignAbbrevCity=OFF.
INIT-CITY CInitCity INITCITY	29
FINAL-UNIQUE-OPT cTUniqueZip TUNIQUE	29
FINAL-STRTHPON-OPT cTStreetPhonetics TSTRPHON	30
FINAL-FIRMCORR-OPT cTFirmCorrection TFRMCORR	31
FINAL-CITYPHONE-OPT cTCityPhonetics TCTYPHON	32
FINAL-WEIGHT-OPT cTWeighting TWEIGHT	33
FINAL-ZIPCORR-OPT cTZipCorrection TZIPCORR	34 PBFN-GCFG-ASSIGN-ZIP-OPT PBFNSetupDef.cAssignZipCodes
FINAL-CITYCORR-OPT cTCityCorrection TCTYCORR	35
FINAL-STRCOSM-OPT cTStreetCosmetics TSTRCOSM	36
FINAL-FRMPRS-OPT cTFirmParse TFRMPRS	37 Firm return line logic is automatic. See the PBFNLabelDef structure for more information.
FINAL-UNITDES-OPT cTUnitDesignator TUNITDES	38 Unit designators are corrected and returned to the calling program in return unit fields. You can decide to use input or return field.

Finalist CI Initialize (0) Callarea Fields (Part 4 of 7)

COBOL (LPFNCL01)

C (LPFNCL0C)

Assembler (LPFNCL04)

Position	CI Structure/Field/Notes
39 FINAL-CTYLONG-OPT cTLongCityName TCTYLONG	PBFN-GCFG-ABBREV-CITY-OPT PBFNSetupDef.cAssignAbbrevCity=N (Return long names returned in label line.) Both long and short names returned on a process call in separate fields.
40 FINAL-ALSLBL-OPT cTAliasLabelLine TALSLBL	PBFN-GCFG-ALIAS-STRT-OPT PBFNSetupDef.cRetAliStName=Y (Alias street names are returned in label line.) Both base and alias street names are returned on a process call in separate fields.
41-43 FINAL-VSE1-PRT caVSESysNo[0] VSYNO1	
44-46 FINAL-VSE2-PRT caVSESysNo[1] VSYNO2	
47-49 FINAL-VSE3-PRT caVSESysNo[2] VSYNO3	
50-52 FINAL-VSE4-PRT caVSESysNo[3] VSYNO4	
53-55 FINAL-VSE5-PRT caVSESysNo[4] VSYNO5	
56-58 FINAL-VSE6-PRT caVSESysNo[5] VSYNO6	
59-61 FINAL-VSE7-PRT caVSESysNo[6] VSYNO7	
62-69 FINAL-STAT-REPORT-OPTIONS caOptions CAFNSTAT	
70-75 FINAL-REPORT-SELECT1-6 caSelect[0]-[5] CARPT1-6	Address Detail Report turned on with I/O, Isol, and Info Reports. If any report option is set to "Y", you must set cAddrDtlRptInfo, cAddrDtlRptIsol, and cAddrDtlRptInfo to "ON". If FINAL-REPORTSELECT4 is set to "Y", you must set cAddrDtlRptSugg to "ON".

Finalist CI Initialize (0) Callarea Fields (Part 5 of 7)

COBOL (LPFNCL01)		
C (LPFNCL0C)		
Assembler (LPFNCL04)	Position	CI Structure/Field/Notes
FINAL-FIRMLBL-OPT caFirmLabel TFRMLBL	76	PBFN-GCFG-RETINPUTFIRM PBFNSetupDef.cRetInputFirm
FINAL-REPORT-KEY caKey CAKEY	77-116	
FINAL-LOT-OPT caLOT CALOT	77	PBFN-GCFG-ASSIGNLOT PBFNSetupDef.cAssignLOT
FINAL-REPORT-NTH caNth CARPTNTH	78-81	PBFN-GCFG-ADDRDTLRPTISOLNTHREC PBFNSetupDef.lAddrDtlRptIsolNthRec
FINAL-REPORT-TITLE caTitle CARPTITL	82-116	PBFN-GCFG-RPTTITLE PBFNSetupDef.cRptTitle
FINAL-REPORT1-MAX IMaxReport[0] CARP1MAX	117-120	PBFN-GCFG-ADDRDTLRPTISOLMAXREC PBFNSetupDef.lAddrDtlRptIsolMaxRec
FINAL-REPORT2-MAX IMaxReport[1] CARP2MAX	121-124	The setting used by the first max field is the setting used for all max fields.
FINAL-REPORT3-MAX IMaxReport[2] CARP3MAX	125-128	The setting used by the first max field is the setting used for all max fields.
FINAL-REPORT4-MAX IMaxReport[3] CARP4MAX	129-132	The setting used by the first max field is the setting used for all max fields.
FINAL-REPORT5-MAX IMaxReport[4] CARP5MAX	133-136	The setting used by the first max field is the setting used for all max fields.
FINAL-REPORT6-MAX IMaxReport[5] CARP6MAX	137-140	The setting used by the first max field is the setting used for all max fields.
FINAL-NUMBER-OF-LINES SNumLines CALINES	141-142	PBFN-GCFG-ADDRDTLRPTISOLPG-LEN PBFNSetupDef.lAddrDtlRptIsolPageLen

Finalist CI Initialize (0) Callarea Fields (Part 6 of 7)

COBOL (LPFNCL01)

C (LPFNCL0C)

Assembler (LPFNCL04)

Position	CI Structure/Field/Notes
FINAL-3553-SECA1A caA1A SECA1A	143-174 PBFN-GCFG-CASSCOMP-NAME PBFNSetupDef.cCASSCompName
FINAL-3553-SECA1B caA1B SECA1B	175-206 PBFN-GCFG-CASSPROD-NAME PBFNSetupDef.cCASSProdName
FINAL-3553-SECA1C caA1C SECA1C	207-238 PBFN-GCFG-LOTCERTCOMP-NAME PBFNSetupDef.cLOTCertCompName
FINAL-3553-SECA1D caA1D SECA1D	239-270 PBFN-GCFG-DPCCERTPROD-NAME PBFNSetupDef.cDPCCertProdName
FINAL-3553-SECB1 caB1 SECB1	303-328 PBFN-RDAT-LISTPROCESSOR-NAME PBFNReportData.ListProcessorName Passed on the PBFNTerminate call.
FINAL-3553-SECB4 caB4 SECB4	329-354 PBFN-RDAT-LISTFILE-NAME PBFNReportData.ListFileName Passed on the PBFNTerminate call.
FINAL-3553-SECB5 caB5 SECB5	355-377 PBFN-RDAT-LISTNUMBER PBFNReportData.ListNumber Passed on the PBFNTerminate call.
FINAL-3553-SECD3A caD3A SECD3A	378-415 PBFN-GCFG-MAILER-NAME PBFNSetupDataDef.cMailerName
FINAL-3553-SECD3B caD3B SECD3B	416-453 PBFN-GCFG-MAILERADDRESS PBFNSetupDataDef.cMailerAddress
FINAL-3553-SECD3C caD3C SECD3C	454-491 PBFN-GCFG-MAILERCITYLINE PBFNSetupDataDef.cMailerCityLine
FINAL-3553-SECD3D caD3D SECD3D	492-529 PBFN-GCFG-MAILERADDRESS2 PBFNSetupDataSet.cMailerAddress2
FINAL-3553-SECD3E caD3E SECD3E	530-567 PBFN-GCFG-MAILERADDRESS3 PBFNSetupDataSet.cMailerAddress3

Finalist CI Initialize (0) Callarea Fields (Part 7 of 7)

COBOL (LPFNCL01)		
C (LPFNCL0C)		
Assembler (LPFNCL04)	Position	CI Structure/Field/Notes
FINAL-3553-SECD3F caD3F SECD3F	568-605	PBFN-GCFG-MAILERADDRESS4 PBFNSetupDataSet.cMailerAddress4
FINAL-DPV-OPT caDPV CADPV	726	PBFN-GCFG-ASSIGNDPV PBFNSetupDataDef.cAssignDPV
FINAL-EWS-OPT caEWS CAEWS	727	PBFN-GCFG-ASSIGNEWS PBFNSetupDataDef.cAssignEWS
FINAL-LLK-OPT caLLK CALLK	728	PBFN-GCFG-ASSIGNLACSLINK PBFNSetupDataDef.cAssignLACSLink
FINAL-DUAL-ADDR cDualAddrSwt CADUALAD	729	PBFN-GCFG-DUALADDRSWT PBFNSetupDataDef.cDualAddrSwt
FINAL-DPV-BUF-SIZE cDPVBufSize CADPVBUF	733-736	PBFN-GCFG-DPVBUFSIZE PBFNSetupDataDef.cDPVBufSize
FINAL-SLK-OPT caSLK CASLK	737	PBFN-GCFG-ASSIGNSUITELINK PBFNSetupDataDef.cAssignSuiteLink
FINAL-ASM-OPT caASM CAASM	738	PBFN-GCFG-ALLSTREETMATCHING PBFNSetupDataDef.cAllStreetMatching

Migrating the Initialize (0) Return Callarea

This migration table maps the CI Return callarea for the Initialize call to the Finalist native APIs and structures. The Finalist native PBFNInfo call now returns the Finalist CI Init call information.

Finalist CI Initialize (0) Return Callarea Fields (Part 1 of 2)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-ONLINE-RETURN-CODE-0 cOnlineRC OLRETCD	1	If the PBFNProcess call returns PBFN_SUCCESS, then checkPBFNAddressDataDef.cReturnLevel. <ul style="list-style-type: none"> A cReturnLevel of PBFN_RTNZIP4 indicates a 0. A cReturnLevel of PBFN_RTNZIPCRRT indicates a 1. A cReturnLevel value of PBFN_RTNZIP indicates a 2. Any other value indicates a value of 9. If the PBFNProcess call returns any other value, FINALONLINE-RETURN-CODE is 9.
FINAL-OLD-RETURN-CODE2-0 cOldRC2 OLRETCD2	2	
FINAL-OLD-RETURN-CODE3-0 cOldRC3 OLRETCD3	3	
CAERRMOD-0 caErrorModule CAERRMOD	4-11	
CAERRSRC-0 caResource CAERRSRC	12-19	
CAERRDSC-0 caDescription CAERRDSC	20-28	
FINAL-TIME-STAMP caTime CATSTAMP	4-11	
FINAL-DATE-STAMP caDate CADSTAMP	12-21	
FINAL-DATA-FILE-VER caDataFileVer CABASERV	22-25	PBFN-GCFG-ZIP4BASEVER PBFNSetupDef.cZip4BaseVer

Finalist CI Initialize (0) Return Callarea Fields (Part 2 of 2)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-DATA-FILE-UPDATE caDataFileUpdate CABASEDT	26-35	PBFN-GCFG-ZIP4BASEDATE PBFNSetupDef.cZip4BaseDate The Finalist native interface does not return DPV date information. The Finalist native interface verifies that DPV dates match the ZIP+4 (CBDATA) dates. You may substitute ZIP+4 dates for DPV dates.
FINAL-CITY-FILE-VER caCityFileVer CACITYVR	36-39	PBFN-GCFG-CITYBASEVER PBFNSetupDef.cCityBaseVer
FINAL-CITY-FILE-UPDATE caCityFileUpdate CACITYDT	40-49	PBFN-GCFG-CITYBASEDATE PBFNSetupDef.cCityBaseDate
FINAL-PRODUCT-VERSION caProductVersion CAFNVVER	50-52	PBFN-GCFG-CASSENGINEVERSION PBFNSetupDef.cCassEngineVersion
FINAL-DPV-DATA-FILE-DATE caDPVDataFileDate CAHSADTE	53-62	PBFN-GCFG-DPVBASEDATE The Finalist native interface does not return DPV date information. The Finalist native interface verifies that DPV dates match the ZIP+4 (CBDATA) dates. You may substitute ZIP+4 dates for DPV dates.
FINAL-CMRA-FILE-DATE caCMRA-FileDate CAHSCDTE	63-72	The Finalist native interface does not return DPV date information. The Finalist native interface verifies that DPV dates match the ZIP+4 (CBDATA) dates. You may substitute ZIP+4 dates for DPV dates.
FINAL-SEED-FILE-DATE caSeedFileDate CAHSFDTE	73-82	The Finalist native interface does not return DPV date information. The Finalist native interface verifies that DPV dates match the ZIP+4 (CBDATA) dates. You may substitute ZIP+4 dates for DPV dates.
FINAL-LCD-FILE-DATE caLCDFileDate CALCDDTE	83-92	The Finalist native interface does not return DPV date information. The Finalist native interface verifies that DPV dates match the ZIP+4 (CBDATA) dates. You may substitute ZIP+4 dates for DPV dates.
FINAL-SLK-RELEASE-DATE caSLKReleaseDate CSSLKRLD	93-102	The Finalist native interface returns the Suite ^{Link} database release date info.

Process - 4, 5, 6, and 7 Calls

The CI Process (4, 5, 6, or 7) call is called PBFNProcess (PBFNPROC) in the native interface. The parameter passed into PBFNProcess is called AddressData (PBFNADRS copybook). PBFNProcess can optionally accept two essentially identical parameters. If one AddressData parameter is passed in, AddressData contains the input address when PBFNProcess is called. The same fields are updated in place with the returned address information when PBFNProcess has ended. Optionally, you can pass a second AddressData structure to PBFNProcess. In this case, the first AddressData contains your unchanged input address. The second AddressData (PBFNRRTN copybook) is updated with the returned address information.

Two different address types can be passed into Finalist. A complete address (123 E MAIN ST) is the default. Optionally a parsed address (Primary range=123, pre-directional=E, street name=MAIN, etc.) can be passed in. Finalist processes a pre-parsed address faster because Finalist does not need to decipher the variations (isolations in Finalist terminology) of a complete address. Please note that the parsed address components must be accurate and in the format that Finalist originally returned. Failure to follow this procedure may result in the parsed address failing to code.

The Finalist CI does not support parsed input data. To specify parsed input for Finalist AddressData, use this statement:

```
SET PBFN-ADRS-PARSED TO TRUE.
```

To initialize the AddressData structure in COBOL:

- For an Input or an Input/Output structure, use this statement:

```
INITIALIZE PBFN-ADRS-ADDRESSDATA
```

- For an Output structure, use this statement:

```
INITIALIZE PBFN-RRTN-RRTN
```

Reason Codes

The most significant difference between the Finalist Compatibility Interface (CI) and the Finalist native interface is the return, reason, and address-information codes. The CI version has a limited set of return information but significant customer business logic was written that depends on those values. The direct AddressInfo information returned from the native interface does not always directly map to the old return, reason, and address-information codes. However, Finalist includes the FNCIRCAC conversion routine to map the AddressInfo fields to return, reason, and address-information fields.

Using the FNCIRCAC Conversion Routine

You can use the FNCIRCAC conversion routine to map the AddressInfo fields to return, reason, and address-information fields. To use FNCIRCAC, you must pass in seven parameters. The first three parameters are the input parameters to FNCIRCAC. The last four parameters are output parameters.

1. The first parameter is a pointer to a one byte Y or N indication of your firm processing ([FINAL-FIRMLBL-OPT](#) or derived from [PBFN-GCFG-RETINPUTFIRM](#)).
2. The second parameter is a pointer to a fullword (4 byte integer), that contains the return code from the PBFNProcess call (see ["Initialize - 0 Call" on page 5](#)).
3. The third parameter is a pointer to the output AddressData structure after the call to PBFNProcess.
4. The fourth parameter is a pointer to a three-byte field formerly known as FINAL-OUTPUT-SELECT.
5. The fifth parameter is a pointer to a one-byte area formerly known as FINAL-RETURN-CODES.
6. The sixth parameter is a pointer to a 12-byte area formerly known as FINAL-REASON-CODES.
7. The seventh and final parameter is a pointer to the 10-byte area formerly known as FINAL-ADDRESS-INFO-CODES.

To use FNCIRCAC, you need to:

1. Define the local fields.

2. Successfully call PBFNProcess (does not generate return code -1).
3. Call FNCIRCAC passing the parameters as described immediately above.

After the call to FNCIRCAC, you can continue with your business logic using the return, reason, and address information codes that FNCIRCAC returned.

i FNCIRCAC returns the codes as defined when the CI was first developed. If you are not satisfied with the values returned from FNCIRCAC, Finalist includes the C source for FNCIRCAC. You are free to modify the source and reuse it for your own purposes. Pitney Bowes does not provide support for the FNCIRCAC source.

Migrating the Process (4, 5, 6 and 7) Call

This migration table maps the CI Process callarea to the Finalist native APIs and structures. The Finalist CI process (4, 5, 6, 7) call is replaced with the Finalist native PBFNProcess call. Because the CI did not accept real parsed data input, all address information would normally be passed using the PBFNAddressData structure. If your address data is in parsed component format (for example, range, pre-directional, street name, suffix), you can use the PBFN-ADRS-PARSED option (described above) to identify parsed data as input to the Finalist native PBFNProcess call.

Finalist CI Process (4, 5, 6 and 7) Callarea Fields (Part 1 of 2)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-FILLER lAnchorPointers \$REREGRS	1-12	
FINAL-FUNCTION-CODE cFunctionCode[0] FUNCODE	13	
FINAL-REPORT-KEY cakey CAKEY	77-116	PBFN-ADRS-USERKEY PBFNAddressDataDef.cUserKey
USER-INPUT-FIRM-LINE caInputFirm FIRMLN	143-212	PBFN-ADRS-FIRM PBFNAddressDataDef.cFirm
USER-INPUT-URB-LINE caInputUrb URBIN	213-242	PBFN-ADRS-URB PBFNAddressDataDef.cUrb
USER-INPUT-ADDRESS1 caInputAddr1 ADDRLN1	243-312	PBFN-ADRS-ADDRESS1 PBFNAddressDataDef.cAddress1 Address line, unit, and PMB information can be entered into the address lines.
USER-INPUT-ADDRESS2 caInputAddr2 ADDRLN2	313-382	PBFN-ADRS-ADDRESS2 PBFNAddressDataDef.cAddress2

Finalist CI Process (4, 5, 6 and 7) Callarea Fields (Part 2 of 2)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
USER-INPUT-CITY-STATE caInputCitySt CTYST	383-452	PBFN-ADRS-CITY PBFNAddressDataDef.cCity Input city, state, and ZIP Code can all be entered into the cCity field.
USER-INPUT-ZIP caInputZip ZIPIN	453-457	PBFN-ADRS-ZIP PBFNAddressDataDef.cZip
USER-INPUT-SEC-SEG caInputSecSeg SECSGIN	458-461	PBFN-ADRS-ZIP4 PBFNAddressDataDef.cZip4
USER-INPUT-CR caInputCrRte CRRTEIN	462-466	PBFN-ADRS-CRRTE PBFNAddressDataDef.cCrRte
Z4-RESERVED caZ4Reserved Z4RESVD	606-725	No mapping.
FINAL-DPV-OPT caDPV CADPV	726	No mapping in native during Process call.

Migrating the Process (4, 5, 6, and 7) Return Callarea

This migration table maps the CI Return callarea for the Process call to the Finalist native APIs and structures. The return structures passed to the Finalist native PBFNProcess call now return the information previously returned from the Finalist CI Process call. Finalist CI returned a combination of address line and parsed data components.

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 1 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-ONLINE-RETURN-CODE cOnlineRC OLRETCD	1	If the PBFNProcess call returns PBFN_SUCCESS, then check PBFNAddressDataDef.cReturnLevel.: <ul style="list-style-type: none"> • A cReturnLevel of PBFN_RTNZIP4 indicates a 0. • A cReturnLevel of PBFN_RTNZIPCRRT indicates a 1. • A cReturnLevel value of PBFN_RTNZIP indicates a 2. • Any other value indicates a value of 9. If PBFNProcess returns any other value, FINAL-ONLINE-RETURNCODE is 9.
FINAL-OLD-RETURN-CODE2 cOldRC2 OLRETCD2	2	

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 2 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-OLD-RETURN-CODE3 cOldRC3 OLRETC3	3	
CAERRMOD union4567.OnlineError.caErrorModule ACERRMOD	4-11	
CAERRSRC union4567.OnlineError.caResource CAERRSRC	12-19	
CAERRDSC union4567.OnlineError.caResource CAERRDSC	20-28	
FINAL-ISOL union4567.FinalArea4567.cIsol ISOLATOR	4	No longer returned in the Finalist native interface. Finalist native internal logic has multiple isolation paths and logic that make this an obsolete piece of information.
FINAL-RECORD-NUMBER union4567.FinalArea4567.caRecNumber RECNUMBR	5-9	
FINAL-ZIP union4567.FinalArea4567.Zip RTNZIP	10-14	PBFN-RRTN-ZIP PBFNAddressDataDef.cZip
FINAL-SCF union4567.FinalArea4567.Zip.caScf RTNSCF	10-12	Not returned in a parsed format in the Finalist native interface. Use the first three positions of the following fields: PBFN-RRTN-ZIP field PBFNAddressDataDef.cZip
FINAL-ZONE union4567.FinalArea4567.Zip.caZone RTNZONE	13-14	Not returned in a parsed format in the Finalist native interface. Use the last two positions of the following fields: PBFN-RRTN-ZIP field PBFNAddressDataDef.cZip
FINAL-SEC-SEG caSecSeg RTNSS#	15-18	PBFN-RRTN-ZIP4 PBFNAddressDataDef.cZip4
FINAL-CR-RTE caCrRte RTNCR	19-23	PBFN-RRTN-CRRTE PBFNAddressDataDef.cCrRte
FINAL-STATE caState RTNSTATE	24-25	PBFN-RRTN-STATE PBFNAddressDataDef.cState

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 3 of 13)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-CITY caCity RTNCITY	26-38	PBFN-RRTN-CITY PBFNAddressDataDef.cCity
FINAL-COUNTY caCounty RTNCNTY	39-43	PBFN-RRTN-COUNTY-NAME PBFNAddressDataDef.cCountyName
FINAL-DIR11 Direction[0].caDir1 DIRCTN11	44-45	PBFN-RRTN-PREDIRECTIONAL PBFNParsedAdrDefinition.cPreDirectional The Finalist native interface does not support returning information on failed isolations.
FINAL-SFX11 Direction[0].caSfx1 SUFFIX11	46-49	PBFN-RRTN-STREETSUFFIX PBFNParsedAdrDefinition.cStreetSuffix The Finalist native interface does not support returning information on failed isolations.
FINAL-SFX12 Direction[0].caSfx2 SUFFIX12	50-53	The Finalist native interface does not support returning information on failed isolations.
FINAL-PDIR1 Direction[0].caPDir DIRCTN12	54-55	The Finalist native interface does not support returning information on failed isolations.
FINAL-DIR21 Direction[1].caDir1 DIRCTN21	56-57	The Finalist native interface does not support returning information on failed isolations.
FINAL-SFX21 Direction[1].caSfx1 SUFFIX21	58-61	The Finalist native interface does not support returning information on failed isolations.
FINAL-SFX22 Direction[1].caSfx2 SUFFIX22	62-65	The Finalist native interface does not support returning information on failed isolations.
FINAL-PDIR2 Direction[1].caPDir DIRCTN22	66-67	The Finalist native interface does not support returning information on failed isolations.
FINAL-DIR31 Direction[2].caDir1 DIRCTN31	68-69	The Finalist native interface does not support returning information on failed isolations.
FINAL-SFX31 Direction[2].caSfx1 SUFFIX31	70-73	The Finalist native interface does not support returning information on failed isolations.

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 4 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-SFX32 Direction[2].caSfx2 SUFF32	74-77	The Finalist native interface does not support returning information on failed isolations.
FINAL-PDIR3 Direction[2].caPDir DIRCTN32	78-79	The Finalist native interface does not support returning information on failed isolations.
FINAL-VALIDCICS-OPTION caValidCICSOption CAFNOPT	80	
FINAL-HOUSE-NUM caHouseNum HOUSNMBR	84-93	PBFN-RRTN-RANGE PBFNAddressDataDef.cRange
FINAL-PRE-DIR caPreDir PREDRCTN	94-95	PBFN-RRTN-PREDIRECTIONAL PBFNAddressDataDef.cPreDirectional
FINAL-STREET-NAME caStreetName STNAME	96-123	PBFN-RRTN-STREET-NAME PBFNAddressDataDef.cStreetName
FINAL-POST-DIR caPostDir PSTDRCTN	124-125	PBFN-RRTN-POSTDIRECTIONAL PBFNAddressDataDef.cPostDirectional
FINAL-SFX1 caSfx1 STSFX1	126-129	PBFN-RRTN-STREETSUFFIX PBFNAddressDataDef.cStreetSuffix
FINAL-SFX2 caSfx2 STSFX2	130-133	The Finalist native interface and the USPS data no longer support the concept of two suffixes.
FINAL-APT-NUM caAptNum APTNMBR	134-143	PBFN-RRTN-UNIT1 or PBFN-RRTN-UNIT2 PBFNAddressDataDef.cUnit1 PBFNAddressDataDef.cUnit2
FINAL-EXTRA caExtra EXTRA	144-152	PBFN-RRTN-EXTRA PBFNAddressDataDef.cExtra
ORIG-ZIP OrigZip ORIGZIP	153-157	PBFN-RRTN-ORIG-ZIP PBFNOrigDataDef.cOrigZip

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 5 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
ORIG-SCF caScf ORIGSCF	153-155	Not returned in a parsed format in native. Use the first 3 positions of: PBFN-RRTN-ORIG-ZIP PBFNOrigDataDef.cOrigZip
ORIG-ZONE caZone ORIGZONE	156-157	Not returned in a parsed format in native. Use the last 2 positions of the following fields: PBFN-RRTN-ORIGZIP PBFNOrigDataDef.cOrigZip
ORIG-SEC-SEG caOrigSecSeg ORIGSS#	158-161	PBFN-RRTN-ORIGZIP4 PBFNOrigDataDef.cZip4
ORIG-CR-RTE caOrigCrRte ORIGCRRT	162-166	PBFN-RRTN-ORIGCRRTE PBFNOrigDataDef.cOrigCrRte
ORIG-STATE caOrigState ORIGSTE	167-168	PBFN-RRTN-ORIGSTATE PBFNOrigDataDef.cOrigState
ORIG-CITY caOrigCity ORIGCITY	169-196	PBFN-RRTN-ORIGCITY PBFNOrigDataDef.cOrigCity
ORIG-ZIPP caOrigZipP ORIGZIPP	197-199	The Finalist native interface does not return packed data.
ORIG-SECSEGP caSecSegP ORIGSS#P	200-202	The Finalist native interface does not return packed data.
FINAL-STATE-SCF-VER caStateScfVer SSVERIND	203	PBFN-RRTN-ADSCITY PBFN-RRTN-ADSSTATE PBFNAddressDataDef.cAdsCity PBFNAddressDataDef.cAdsState settings
FINAL-RETSCF caRetScf RETSCF	204-206	Not returned in a parsed format in the Finalist native interface. Use the first three positions of the following fields: PBFN-RRTN-ZIP PBFNAddressDataDef.cZip
FINAL-PMUNIT caPMUnit PMUNIT	207-210	PBFN-RRTN-ADSPMBDESIGNATOR PBFNAddressDataDef.cPMUnitDesignator
FINAL-PMNUMBER caPMNumber PMNUMBER	211-220	PBFN-RRTN-PMUNITNUMBER PBFNAddressDataDef.cPMUnitNumber

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 6 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-DPV-FOOTNOTES caDPVFootnotes DPVFTNOE	221-230	PBFN-RRTN-DPVFOOTNOTE PBFNAddressDataDef.cDPVFootnote
FINAL-UNITDES caUnitDes UNITDES	231-234	PBFN-RRTN-ADSUNIT1DESIGNATOR PBFNAddressDataDef.cUnit1Designator
FINAL-CITY-IND caCityInd CITYIND	237	PBFN-RRTN-CITY-TYPE PBFNAddressDataDef.cCityType
FINAL-OUTSEL-GOOD cGood CAOUTSLG	238	PBFNProcess return code equal to PBFN_SUCCESS. cReturnLevel in PBFNAddressDataDef set to PBFN_RTNZIP4. cCity and cZIP in PBFNAddressDataDef are not set to PBFN_CORRECTED.
FINAL-OUTSEL-BAD cBad CAOUTSLB	239	PBFNProcess return code not equal to PBFN_SUCCESS.
FINAL-OUTSEL-CHANGE cChange CAOUTSLC	240	PBFNProcess return code equal to PBFN_SUCCESS. cReturnLevel in PBFNAddressDataDef set to PBFN_RTNZIP4 cCity or cZIP in PBFNAddressDataDef are set to PBFN_CORRECTED.
FINAL-RETURN-CODE-1 cRc1 CARETCD1	241	The PBFNProcess return code is set to: <ul style="list-style-type: none"> • 0 = PBFNProcess return code equal to PBFN_SUCCESS and cReturnLevel in PBFNAddressDataDef set to PBFN_RTNZIP4. • 1 = PBFNProcess return code not equal to PBFN_SUCCESS and cReturnLevel in PBFNAddressDataDef set to PBFN_RTNZIPCRRT. • 2 = PBFNProcess return code not equal to PBFN_SUCCESS and cReturnLevel in PBFNAddressDataDef set to PBFN_RTNZIP. • 9 = All other cases.
FINAL-REASON-CODE1 cReasonCode[0] CARESN1	243	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE2 cReasonCode[1] CARESN2	244	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE3 caReasonCode[2] CARESN3	245	Refer to the section " Migrating the CI Reason Codes ".

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 7 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-REASON-CODE4 caReasonCode[3] CARESN4	246	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE5 caReasonCode[4] CARESN5	247	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE6 caReasonCode[5] CARESN6	248	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE7 caReasonCode[6] CARESN7	249	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE8 caReasonCode[7] CARESN8	250	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE9 caReasonCode[8] CARESN9	251	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE10 caReasonCode[9] CARESN10	252	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE11 caReasonCode[10] CARESN11	253	Refer to the section " Migrating the CI Reason Codes ".
FINAL-REASON-CODE12 caReasonCode[11] CARESN12	254	Refer to the section " Migrating the CI Reason Codes ".
FINAL-INFO-CODE1 cInformationCode[0] CAINFO1	255	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE2 cInformationCode[1] CAINFO2	256	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE3 cInformationCode[2] CAINFO3	257	Refer to the section " Migrating the CI Address Information Codes ".

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 8 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-INFO-CODE4 cInformationCode[3] CAINFO4	258	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE5 cInformationCode[4] CAINFO5	259	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE678 cCode678 CAINF678	260-262	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE9 cInformationCode[8] CAINFO9	263	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-INFO-CODE10 cInformationCode[9] CAINFO10	264	Refer to the section " Migrating the CI Address Information Codes ".
FINAL-5D-BARCODE (See next four fields.) CA5BAR	265-272	PBFN-RRTN-FIVEDIGITBARCODE PBFNAddressDataDef.cFiveDigitBarcode
FINAL-5D-BEG cBeg CA5BARB	265	PBFN-RRTN-FIVEDIGITBARCODE Not separately defined in the Finalist native interface. Use position 1 of PBFNAddressDataDef.cFiveDigitBarcode.
FINAL-5D-ZIP caZip CA5RZIP	266-270	PBFN-RRTN-FIVEDIGITBARCODE Not separately defined in the Finalist native interface. Use positions 2–6 of PBFNAddressDataDef.cFiveDigitBarcode.
FINAL-5D-CKDIGIT caChkdigit CA5CHKD	271	PBFN-RRTN-FIVEDIGITBARCODE Not separately defined in native. Use position 7 of PBFNAddressDataDef.cFiveDigitBarcode.
FINAL-5D-END cEnd CA5BARE	272	PBFN-RRTN-FIVEDIGITBARCODE Not separately defined in native. Use position 8 of PBFNAddressDataDef.cFiveDigitBarcode.
FINAL-LINE-OF-TRAVEL CALOTCDS	273-277	PBFN-RRTN-LOTCODE PBFNAddressDataDef.cLOTCode
FINAL-LOT-CODE caLotCode CALOTOD	273-276	Pos 1–4 of: PBFN-RRTN-LOTCODE PBFNAddressDataDef.cLotCode
FINAL-LOT-ASCDS caLotAD CALOTAD	277	Position 5 of: PBFN-RRTN-LOTCODE PBFNProcessDataDef.cLotCode

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 9 of 13)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-LACS-FLAG cLACS CALACS	278	PBFN-RRTN-LACS PBFNAddressDataDef.cLACS
FINAL-ADVANCED-BARCODE AdvBarCode CAABARCD	285-298	PBFN-RRTN-ADVANCEBARCODE PBNFProcessDataDef.cAdvanceBarcode
FINAL-ABC-BEG cBeg CAABARB	285	Position 1 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ABC-ZIP caZip CAARZIP	286-290	Position 2-6 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ABC-ZIP4 caZip4 CARZIP4	291-294	Position 7-10 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ABC-WSEQ caWseq CARWALK	295-296	Position 11-12 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ABC-CKDIGIT caChkdigit CAARCHKD	297	Position 13 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ABC-END cEnd CAABARE	298	Position 14 of: PBFN-RRTN-ADVANCEBARCODE PBFNAddressDataDef.cAdvanceBarcode
FINAL-ALT-ISOLATION AltIsol CAARISOL	299-357	
FINAL-ALT-RANGE caRange CAARANGE	299-308	
FINAL-ALT-PRE-DIR caPreDir CAARDIR	309-310	
FINAL-ALT-STREET caStreet CAASTRT	311-335	

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 10 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-ALT-SUFFIX caSuffix CAASFX	336-339	
FINAL-ALT-POST-DIR caPostDir CAARPDIR	340-341	
FINAL-ALT-UNIT caUnit CAARUNIT	342-347	
FINAL-ALT-RANGE2 caRange2 CAARNGE2	348-357	
FINAL-FULL-CITY-NAME caFullCityName CARCITY	358-382	PBFN-RRTN-FULLCITY-NAME PBFNAddressDataDef.cFullCityName
FINAL-LABEL-RC cRc1 CALRTCDE	383	Label lines are always built from the Finalist area, except for the Firm name. This is a selection option in the setup data.
Unused Unused CAFLLGTH	384	
FINAL-FIRM-LENGTH1 caLen[0] CAFLCLGH	385-386	PBFN-RRTN-FIRM-LEN
FINAL-FIRM-LINE caLine[0] CAFLADDR	387-456	Label lines built and returned from the bottom up. On successful addresses: <ul style="list-style-type: none"> • Line 5=CSZ • Line 4=Address • Line 3=URB if input, else Firm if input • Line 2=Firm if input and not on line 3 On failed address: <ul style="list-style-type: none"> • Line 5=CSZ • Line 4=address line 2 if entered, else address line 1 • Line 3=address line 1 if not on Line 4 Lines based on data received.
FINAL-LABEL-LENGTH1 caLen[1] CAL1CLGH	458-459	PBFN-RRTN-ADDRESSLINE1-LEN PBFNAddressDataDef.sAddressLine1Len

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 11 of 13)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-LABEL-LINE1 caLine[1] CAL1ADDR	460-529	PBFN-RRTN-ADDRESSLINE1 PBFNLabelLineDef.cAddressLine1
FINAL-LABEL-LENGTH2 caLen[2] CAL2CLGH	531-532	PBFN-RRTN-ADDRESSLINE2-LEN PBFNLabelLineDef.sAddressLine2Len
FINAL-LABEL-LINE2 caLine[2] CAL2ADDR	533-602	PBFN-RRTN-ADDRESSLINE2 PBFNLabelLineDef.cAddressLine2
FINAL-LABEL-LENGTH3 caLen[3] CAL3CLGH	604-605	PBFN-RRTN-ADDRESSLINE3-LEN PBFNLabelLineDef.sAddressLine3Len
FINAL-LABEL-LINE3 caLine[3] CALN3CTY	606-675	PBFN-RRTN-ADDRESSLINE3 PBFNLabelLineDef.cAddressLine3
FINAL-ALT-LABEL-LENGTH2 caLen[4] CAALCLGH	677-678	See FINAL-LABEL_RETURN_AREA for full explanation.
FINAL-ALT-LABEL-LINE2 caLine[4] CAALADDR	679-748	If PBFN-ADRS-LACSSEEDHIT contains a blank, FINAL-ALT-LABELLINE2 contains the alternate label line. If PBFN-ADRS-LACSSEEDHIT contains any other value, FINAL-ALT-LABEL-LINE2 contains the LACS ^{Link} seed detail record.
FINAL-MAIL-FIRM-NAME caMailFirmName CACMFIRM	749-788	These fields are not directly supported. Consider using the pRtnAddressData.cFirm field (truncated if necessary).
FINAL-CITY-VANITY-NAME caCityVanityName CACTYVAN	789-801	Vanity names no longer used. Long, abbreviated, and non-mailing city names are now returned in individual fields. Consider using the pRtnAddressData.cNonMailingCityName field.
FINAL-URB-RETURN caURBReturn CAURBRTN	802-831	PBFN-RRTN-URB PBFNAddressDataDef.cUrb
FINAL-DEFAULT-IND cDefault CADFLT	832	PBFN-RRTN-DEFAULTMATCH PBFNAddressDataDef.cMatch
FINAL-POUND-FIELD1 ca#Field1 CA#FLD1	833-842	All secondary info is now returned in the following fields. PBFN-RRTN-UNIT-1 or PBFN-RRTN-UNIT-2 PBFNAddressDataDef.cUnit1 and PBFNAddressDataDef.cUnit2

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 12 of 13)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-POUND-FIELD2 ca#Field2 CA#FLD2	843-852	All secondary info is now returned in the fields below. PBFN-RRTN-UNIT1 or PBFN-RRTN-UNIT2 PBFNAddressDataDef.cUnit1 and PBFNAddressDataDef.cUnit2
FINAL-DPV-IND caDPVIND CADPVIND	853	PBFN-RRTN-DPVFLAGS PBFNAddressDataDef.cDPVFlags[0]
FINAL-CMRA-IND caDPVCMRA CADPVCMR	854	PBFN-RRTN-DPVFLAGS PBFNAddressDataDef.cDPVFlags[1]
FINAL-DPV-FALSE-POSITIVE caDPVFalsePositive CADPVFLS	855	
FINAL-MATCH-LEVEL-IND caMatchLevel CAMATLVL	856	PBFN-RRTN-MATCHLEVEL PBFNAddressDataDef.cMatchLevel
FINAL-LACS-SEED-HIT cLacsSeedHit CALACSSH	858	PBFN-ADRS- LACSSSEEDHITPBFNAddressDataDef.LACSSeedHit.
FINAL-IDMSFLDS caIDMSMessage IDMSFLDS	859-892	The Finalist native interface no longer supports the IDMS operating system/platform.
FINAL-ISMSERMD caIDMSModuleName IDMSERMD	893-900	The Finalist native interface no longer supports the IDMS operating system/platform.
FINAL-SEASONAL-DATA caSeasonInd CASEASNL	901-912	PBFN-RRTN-SEASONALFLAGS PBFNAddressDataDef.cSeasonalFlags
FINAL-SLK-RETURN-CODE caSLKReturnCode CASLKRC	913-915	PBFN-RRTN-STELNKRTNCODE PBFNAddressDataDef.cStelNkRtnCode
FINAL-SLK-MATCH-CODE caSLKMatchCode CASLKMC	916	PBFN-RRTN-STELNKMATCHCODE PBFNAddressDataDef.cStelNkMatchCode
FINAL-SLK-FIDELITY-CODE caSLKFidelityCode CASLKFC	917	PBFN-RRTN-STELNKFIDELITYCODE PBFNAddressDataDef.cStelNkFidelityCode

Finalist CI Process (4, 5, 6 and 7) Return Callarea Fields (Part 13 of 13)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-DPV-SEED-HIT caSeedViolationFlag CASEEDFG	918	PBFN-RRTN-DPVSEEDHIT sSeedViolationEncountered
FINAL-DPV-NCOAKEY-FLAG caDPVKeyNCOAFlag CADPNSTP	919	PBFN-RRTN-SEEDVIOLATIONENC sDPVKeyNCOA

Migrating the CI Reason Codes

The migration table below maps the CI reason codes to the Finalist native interface reason codes. Also, refer to the program notes on "[Using the FNCIRCAC Conversion Routine](#)" on page 14. Programs can also call FNCIRCAC to convert the Finalist native address information codes to the CI format.



C field names are provided in the following table since COBOL cannot easily process "bit flags". For more information, refer to the description of PBITBYTE in "[FNCIRCAC](#)" on page 3.

Finalist CI Reason Codes (Part 1 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
General Return Code		
0 ZIP Code, ZIP+4, and carrier route assigned	PBFN_SUCCESS	Return code from PBFNProcess
1 ZIP Code (5-digit) and carrier route assigned	PBFN_RTNZIPCRRT	PBFNAddressDataDef.cReturnLevel
2 ZIP Code only assigned	PBFN_RTNZIP	PBFNAddressDataDef.cReturnLevel
9 No codes assigned	PBFN_FAIL	Return code from PBFNProcess call
Reason Code 1	Return ZIP Code explanation	
0 ZIP Code verified	PBFN_EXACT	PBFNAddressDataDef.cZip
1 ZIP Code returned	PBFN_RTN_DATA	PBFNAddressDataDef.cZip

Finalist CI Reason Codes (Part 2 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
2 ZIP Code unchanged due to unique ZIP Code.	N/A	The Finalist native interface always performs unique ZIP Code processing. This value is returned if the engine processing generated error code 4104 (error for not coding to unique ZIP Code).
3 ZIP Code unchanged due to ZIP Code correction	N/A	The Finalist native interface always performs ZIP Code processing. This is no longer a valid return code and is not supported.
4 ZIP Code determined valid in multiple choice situations	See notes	PBFNAddressDataDef.cZip has PBFN_EXACT set. PBFNAddressDataDef.cFailureType has PBFN_COMPONENT_MULTI_CHOICE set.
5 ZIP Code determined unique-changed	PBFN_CORRECTED	PBFNAddressDataDef.cZip and PBFNAddressDataDef.cZipType have PBFN_UNIQUE set.
6 Last 2 digits of 5-digit ZIP Code changed	N/A	Not supported in the Finalist native interface. The PBFNAddressDataDef.cZip indicates changes have been made to the ZIP Code. To continue to use this information, a user driver can compare ZIP Codes.
7 First 3 digits of 5 digit ZIP Code changed	N/A	Not supported in the Finalist native interface. The PBFNAddressDataDef.cZip indicates changes have been made to the ZIP Code. To continue to use this information, a user driver can compare ZIP Codes.
9 ZIP Code not determined	PBFN_INVALID	PBFNAddressDataDef.cZip
Reason Code 2	Return city explanation	
0 City verified as input	PBFN_EXACT	PBFNAddressDataDef.cCity
1 City returned-none input	PBFN_RTN_DATA	PBFNAddressDataDef.cCity
2 City standardized on input	PBFN_CORRECTED	PBFNAddressDataDef.cCity
3 USPS preferred name returned	N/A	Not supported in the Finalist native interface. See PBFNAddressDataDef.cCityType for available information.
4 City unchanged due to CTYCORR=NO	N/A	The Finalist native interface always performs city change processing. This is no longer a valid return code and not supported.

Finalist CI Reason Codes (Part 3 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
5 City name changed-corrected	PBFN_CORRINPUT	PBFNAddressDataDef.cCity
6 Duplicate city name within state	N/A	Not supported in the Finalist native interface. This situation is very rare.
7 Non-mailing name	PBFN_NONMAILING_NAME	PBFNAddressDataDef.cCityType
9 City not determined	PBFN_INVALID	PBFNAddressDataDef.cCity
Reason Code 3	Return carrier route explanation	
0 Carrier route verified	PBFN_EXACT	PBFNAddressDataDef.cCarrier
1 Carrier route returned	PBFN_RTN_DATA	PBFNAddressDataDef.cCarrier
2 Carrier route determined valid in multiple choice situation	PBFN_EXACT	PBFNAddressDataDef.cCarrier has PBFN_EXACT set. PBFNAddressDataDef.cFailureType has PBFN_COMPONENT_MULTI_CHOICE set.
3 Carrier route changed	PBFN_CORRINPUT	PBFNAddressDataDef.cCarrier
5 Weighting used to determine carrier route	N/A	Not supported in the Finalist native interface.
6 Default carrier route returned	See notes	PBFNAddressDataDef.cDefaultMatch=Y. Set if carrier route, ZIP+4, or DPBC is set to a default value.
7 Non-deliverable address	See notes	PBFNAddressDataDef.cNonDeliverableInd=Y
9 Carrier route not determined	PBFN_INVALID	PBFNAddressDataDef.cCarrier
Reason Code 4	Return ZIP+4 explanation	
0 ZIP+4 verified	PBFN_EXACT	PBFNAddressDataDef.cZip4

Finalist CI Reason Codes (Part 4 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
1 ZIP+4 returned	PBFN_RTN_DATA	PBFNAddressDataDef.cZip4
2 ZIP+4 determined valid in multiple choice situation	PBFN_EXACT	PBFNAddressDataDef.cZip4 has PBFN_EXACT set. PBFNAddressDataDef.cFailureType has PBFN_COMPONENT_MULTI_CHOICE set.
3 ZIP+4 changed	PBFN_CORRINPUT	PBFNAddressDataDef.cZip4
5 Weighting used to determine ZIP+4	N/A	Not supported in the Finalist native interface.
6 Default ZIP+4 returned	See notes	PBFNAddressDataDef.cDefaultMatch=Y. Set if carrier route, ZIP+4, or DPBC is set to a default value.
7 Non-deliverable address	PBFN_NONMAILING_NAME	PBFNAddressDataDef.cCityType
8 AMS-II has duplicate or overlapping range	PBFN_RANGE_OVERLAP	PBFNAddressDataDef.cRange
9 ZIP+4 not determined.	PBFN_INVALID	PBFNAddressDataDef.cZip4
Reason Code 5	Return street explanation	
0 Street verified as input	PBFN_EXACT	PBFNAddressDataDef.cStreetName
1 Input street standardized	PBFN_CORRECTED	PBFNAddressDataDef.cStreetName
2 Street found using exceptions table	See notes	PBFNAddressDataDef.cExceptionsInd=Y. Exceptions table used during processing.
3 Street found using phonetics match	PBFN_CORRINPUT	PBFNAddressDataDef.cStreetName
4 Street matched using up to three character variation	See notes	Internal logic has changed. This is no longer a valid reason code in the Finalist native interface. Refer to PBFNAddressDataDef.cStreet for information on street name changes.

Finalist CI Reason Codes (Part 5 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
5 Street failed because of EWS match	4460	PBFNAddressDataDef.cError
6 Street matched using secondary isolation	N/A	Internal logic has changed. This is no longer a valid reason code in the Finalist native interface.
7 Street found using dual address rules	See notes	PBFNAddressDataDef.cDualAddressInd=Y
9 Street not matched	PBFN_INVALID	PBFNAddressDataDef.cStreetName
Reason Code 6	Return street range explanation	
0 Street range correct	PBFN_EXACT	PBFNAddressDataDef.cRange
1 No match on alpha portion of range	PBFN_RANGE_ALPHA_MISMATCH	PBFNAddressDataDef.cRange
2 No input range	PBFN_NODATA	PBFNAddressDataDef.cRange
3 Out of range	PBFN_INVALID	PBFNAddressDataDef.cRange
4 Valid ranges and street names from multiple ZIP Codes	See notes	Check for ambiguous flags in field settings.
9 Range not determined	PBFN_INVALID	PBFNAddressDataDef.cRange
Reason code 7	Return suffix/directional explanation	
0 Suffix and directional correct	PBFN_EXACT	PBFNAddressDataDef.cSuffix, cPreDir, and cPostDir fields all set to value.
1 Suffix missing or incorrect	PBFN_NODATA or PBFN_INVALID	PBFNAddressDataDef.cSuffix
2 Directional missing or incorrect	PBFN_NODATA or PBFN_INVALID	PBFNAddressDataDef.cPreDir or cPostDir fields set to value.
3 Directional and suffix incorrect	PBFN_NODATA or PBFN_INVALID	PBFNAddressDataDef.cSuffix, cPreDir, and cPostDir fields all set to value.

Finalist CI Reason Codes (Part 6 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
4 Suffix multiple choice	PBFN_COMPONENT_MULTI_CHOICE	PBFNAddressDataDef.cSuffix
5 Directional multiple choice	PBFN_COMPONENT_MULTI_CHOICE	PBFNAddressDataDef.cPreDir or cPostDir fields set to value.
6 Directional outside cardinal point	PBFN_CARDINAL_FAILURE	PBFNAddressDataDef.cPreDir or cPostDir fields set to value.
9 Suffix/directional could not be determined	PBFN_INVALID	PBFNAddressDataDef.cSuffix or cPreDir or cPostDir fields set to value.
Reason code 7	Alias street explanation	
0 Input street is not an alias	0	PBFNAddressDataDef.cStreetType
1 Input street is a preferred alias	PBFN_PREF_ALIAS	PBFNAddressDataDef.cStreetType
2 Input street is a nickname alias	PBFN_OTHER_ALIAS	PBFNAddressDataDef.cStreetType
4 Input street is an alias multiple choice	PBFN_COMPONENT_MULTI_CHOICE	PBFNAddressDataDef.cStreetName
5 Input street is an alternate at delivery	PBFN_ALT_AT_DEL	PBFNAddressDataDef.cStreetType
9 Input street could not be matched	PBFN_INVALID	PBFNAddressDataDef.cStreetName
Reason Code 9	Firm name description	
0 System returned the input firm name (if any) to the label area (or there was no firm on input).	PBFN_NODATA or PBFN_EXACT	PBFNAddressDataDef.cFirm

Finalist CI Reason Codes (Part 7 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
1 System changed and returned the firm name per your specification of FIRM CORR= YES and FIMRLBL= DB	N/A	Not supported in the Finalist native interface. Finalist allows bypass firm processing; however, this is a non-CASS feature. Finalist will not load that portion of the database if specified not to process firms. Set this option in the configuration file using "Process Firms".
2 System changed the firm name for matching purposes but was not returned due to FIRM CORR= YES and FIRMLBL= INP (system returns input firm name)	N/A	Not supported in the Finalist native interface. The corrected firm name is always returned in return structure firm field. You decide which address to use based on the PBFNAddressDataDef.cFirm field and the original input address.
3 Firm record matched but the firm name was missing	5103	PBFNAddressDataDef.cError
9 Firm processing was not successful. There was information on the firm line but it could not be matched to the address	5104	PBFNAddressDataDef.cError
Reason Code 10	Unit Descriptor	Description
0 Unit designator (if any) is correct	PBFN_EXACT or PBFN_NODATA	PBFNAddressDataDef.cUnit1Designator or cUnit2Designator
1 Unit designator changed or added	PBFN_RTN_DATA or PBFN_CORRINPUT	PBFNAddressDataDef.cUnit1Designator or cUnit2Designator
2 Unit designator abbreviated	PBFN_CORRECTED	PBFNAddressDataDef.cUnit1Designator or cUnit2Designator
3 Unit designator missing or no unit number	5101	PBFNAddressDataDef.cError
4 Unit designator invalid	PBFN_INVALID	PBFNAddressDataDef.cUnit1Designator or cUnit2Designator

Finalist CI Reason Codes (Part 8 of 9)


Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
9 Secondary address error	5102	PBFNAddressDataDef.cError
Reason Code 11	Unit number description	
0 Unit number (if any) is correct	PBFN_EXACT, PBFN_NODATA, or PBFN_CORRECTED	PBFNAddressDataDef.cUnit1Range or cUnit2Range
1 Unit number alpha character transposed	PBFN_RANGE_ALPHA_MISMATCH	PBFNAddressDataDef.cRange or cUnit1Range or cUnit2Range
2 Unit number valid in multiple choice	PBFN_COMPONENT_MULTI_CHOICE	PBFNAddressDataDef - any address component may receive this disposition if the effect of the address coding was ambiguous.
3 Unit number missing	5101	PBFNAddressDataDef.cError
4 Unit number invalid	PBFN_INVALID	PBFNAddressDataDef, any address component may receive this disposition if the matched address indicates the input address was incorrect.
7 RR/HC box number alpha character transposed	PBFN_RANGE_ALPHA_MISMATCH	PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_RR or PBFN_ADSTYPE_HC
8 PO box number alpha character transposed	PBFN_RANGE_ALPHA_MISMATCH	PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_PO_BOX
9 Unit number not determined	PBFN_INVALID	
Reason Code 12	Non-conventional description	
0 Non-conventional address correct (if present)	Y	PBFNAddressDataDef.cAddressUnchanged=Y. Check PBFNAddressDataDef.cAddressType.
1 RR/HC corrected	N	PBFNAddressDataDef.cAddressUnchanged=N. PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_RR or PBFN_ADSTYPE_HC.
2 PO Box corrected	N	PBFNAddressDataDef.cAddressUnchanged=N. PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_PO_BOX.

Finalist CI Reason Codes (Part 9 of 9)

Finalist CI Reason Code Description	Finalist Native Field Setting	Field/Notes
3 General delivery corrected	N	PBFNAddressDataDef.cAddressUnchanged=N. PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_GD.
9 Non-conventional address not determined	N	PBFNAddressDataDef.cAddressUnchanged=N. PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_UNKNOWN.

Migrating the CI Address Information Codes

This migration table maps the CI address information codes to the Finalist native address information codes. Also, refer to ["Using the FNCIRCAC Conversion Routine" on page 14](#). Programs can also call FNCIRCAC to convert the Finalist native address information codes to the CI format.

 C field names are provided in the following table since COBOL cannot easily process "bit flags". For more information, refer to the description of PBITBYTE in ["FNCIRCAC" on page 3](#).

Finalist CI Address Information Codes (Part 1 of 4)

Finalist CI Address Info Code Description	Finalist Native Field Setting	Field/Notes
Address Info Code 1	City type information	
0 Single ZIP code city	PBFN_SINGLEZONE	PBFNAddressDataDef.cCityType
1 Multi-ZIP code city	PBFN_MULTIZONE	PBFNAddressDataDef.cCityType
9 City type not determined	0	PBFNAddressDataDef.cCityType
Address Info Code 2	Address type information	
0 Conventional address	PBFN_ADSTYPE_STREET or PBFN_ADSTYPE_HIGHRISE	PBFNAddressDataDef.cAddressType
1 PO Box	PBFN_ADSTYPE_PO_BOX	PBFNAddressDataDef.cAddressType
2 Rural route	PBFN_ADSTYPE_RR	PBFNAddressDataDef.cAddressType
3 Highway contract route	PBFN_ADSTYPE_HC	PBFNAddressDataDef.cAddressType
4 General delivery	PBFN_ADSTYPE_GD	PBFNAddressDataDef.cAddressType
5 Firm address	PBFN_ADSTYPE_FIRM	PBFNAddressDataDef.cAddressType

Finalist CI Address Information Codes (Part 2 of 4)

Finalist CI Address Info Code Description	Finalist Native Field Setting	Field/Notes
6 Pre-directional address	N/A	Not supported in the Finalist native interface. Use the PBFNAddressDataDef.cPreDir flag to determine predirectional attributes.
7 Post-directional address	N/A	Not supported in the Finalist native interface. Use the PBFNAddressDataDef.cPostDir flag to determine postdirectional attributes.
8 Postmaster	N/A	Not supported in the Finalist native interface. Postmasters are not currently flagged in the engine.
9 Address type not determined	PBFN_ADSTYPE_UNKNOWN	PBFNAddressDataDef.cAddressType
Address Info Code 3	Delivery type information	
0 City delivery	PBFN_ADSTYPE_STREET	PBFNAddressDataDef.sAddrMatchLevel
1 City delivery default	None	Not supported in the Finalist native interface.
2 PO Box	PBFN_ADSSLVL_PO	PBFNAddressDataDef.sAddrMatchLevel
3 Rural route/highway contract route	PBFN_ADSSLVL_RRHC_DEFAULT or PBFN_ADSSLVL_RRHC_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel
4 General Delivery	PBFN_ADSSLVL_GENDEL	PBFNAddressDataDef.sAddrMatchLevel
9 Delivery type not determined	0	PBFNAddressDataDef.sAddrMatchLevel
Address Info Code 4	Input address information	
0 Firm found on firm line. Delivery found on line 1 and line 2.	See notes	PBFNAddressDataDef.cFirmLoc=PBFN_FIRMLOC_FIRM and PBFNAddressDataDef.cAddressLoc=PBFN_ADSLOC_L1_L2
1 Address found on address line 1	PBFN_ADSLOC_L1	PBFNAddressDataDef.cAddressLoc
2 Address found on address line 2	PBFN_ADSLOC_L2	PBFNAddressDataDef.cAddressLoc
3 Address found on both address line 1 and line 2	PBFN_ADSLOC_L1_L2	PBFNAddressDataDef.cAddressLoc
4 Firm found on firm line, no delivery address found	See notes	PBFNAddressDataDef.cFirmLoc=PBFN_FIRMLOC_FIRM and PBFNAddressDataDef.cAddressLoc=PBFN_ADSLOC_UNKNOWN

Finalist CI Address Information Codes (Part 3 of 4)

Finalist CI Address Info Code Description	Finalist Native Field Setting	Field/Notes
5 Firm found on address line 1, no delivery address found	See notes	PBFNAddressDataDef.cFirmLoc=PBFN_ FIRMLOC_L1 and PBFNAddressDataDef.cAddressLoc=PBFN_ ADSLOC_UNKNOWN
6 Address found on firm line	See notes	PBFNAddressDataDef.cDualAddressInd indicates the address was found on the firm line. Firm was found on the address line.
7 Firm found on firm line-delivery address found on address line 1	PBFN_ADSLOC_FIRM_L1	PBFNAddressDataDef.cAddressLoc
8 Firm found on firm line-delivery address found on address line 2	PBFN_ADSLOC_FIRM_L2	PBFNAddressDataDef.cAddressLoc
9 All address lines used-address failed	PBFN_FAILED	Return code from PBFNProcess.
Address Info Code 5	Address lookup information	
0 Address found on primary lookup	N/A	Internal logic has changed. This is no longer a valid reason code in the Finalist native interface.
1 Address found on backtrack lookup	N/A	Internal logic has changed. This is no longer a valid reason code in the Finalist native interface.
Address Info Code 6-8	Unit information	
051 through 058 Unit number missing for MFDU with over 59 units	N/A	Data is no longer available. Not supported in the Finalist native interface.
060 Sector/segment is not deliverable	Y	PBFNAddressDataDef.cNonDeliverableInd
101 through 300 Unit number missing for MFDU with 04-59 units	N/A	Data is no longer available. Not supported in the Finalist native interface.
400 and 000 Unit number entered is unavailable or not determined	N/A	Data is no longer available. Not supported in the Finalist native interface.
Address Info Code 8	Output address type	
0 Conventional address coded	PBFN_ADSSLVL_STREET	PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.
1 PO Box address coded	PBFN_ADSSLVL_PO	PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.

Finalist CI Address Information Codes (Part 4 of 4)

Finalist CI Address Info Code Description	Finalist Native Field Setting	Field/Notes
2 Rural route address coded	PBFN_ADSSLVL_RRHC_DEFAULT or PBFN_ADSSLVL_RRHC_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel PBFNAddressDataDef.cAddressType=PBFN_ ADSTYPE_RR Return code from PBFNProcess is PBFN_ SUCCESS.
3 Highway contract address coded	PBFN_ADSSLVL_RRHC_DEFAULT or PBFN_ADSSLVL_RRHC_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel PBFNAddressDataDef.cAddressType =PBFN_ADSTYPE_RR Return code from PBFNProcess is PBFN_ SUCCESS.
4 General deliver address coded	PBFN_ADSSLVL_GENDEL	PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.
5 Firm address coded	PBFN_ADSSLVL_FIRM_PRIMARY or PBFN_ADSSLVL_FIRM_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.
6 Highrise address coded	PBFN_ADSSLVL_HIGHRISE_DEFAULT or PBFN_ADSSLVL_HIGHRISE_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.
7 Firm at highrise address coded	PBFN_ADSSLVL_FIRM_PRIMARY or PBFN_ADSSLVL_FIRM_SECONDARY and PBFN_ADSSLVL_HIGHRISE_DEFAULT or PBFN_ADSSLVL_HIGHRISE_SECONDARY	Combination of bit setting from these bits. PBFNAddressDataDef.sAddrMatchLevel return code from PBFNProcess is PBFN_ SUCCESS.
8 Military address coded	PBFN_ADSSLVL_MIL_DEFAULT or PBFN_ADSSLVL_MIL_SECONDARY	PBFNAddressDataDef.sAddrMatchLevel or PBFNAddressDataDef.cZipType = PBFN_ MILITARY
9 Cannot code output address	PBFN_FAILED	Return code from PBFNProcess
Address Info Code 10	ZIPMove type	
0 Match determined using ZIPMove record	Y	PBFNAddressDataDef.cZipMove
9 ZIPMove forwarding did not occur	N	PBFNAddressDataDef.cZipMove

Terminate - 9 Call

The CI Terminate (9) call is called PBFNTerminate (PBFNTERM) in the native interface. From zero (0) to ten (10) parameters can be passed to the PBFNTerminate call depending on the information you want returned. Full information for calling PBFNTerminate can be found in your *Finalist Reference Guide* in Chapter 1, Using the Finalist APIs.

If your application performs a PBFNInit call, even if the call fails (return code was -1), you should do a PBFNTerminate call. PBFNTerminate frees all storage that Finalist allocated. Failure to do this may result in a system ABEND due to un-freed storage.

The PBFNTerminate call can be a simple or complex API. If you wish to simply terminate Finalist and produce the reports requested during initialization, then call PBFNTerminate only NULL parameters.

If you wish additional information about the run, you can request summary information from the PBFNTerminate call. If you want details about all of the records that Finalist processed in this run, pass in the StatsDef structure (PBFNGSTS).

FINAL-RECORD-COUNT	PBFN-GSTS-TOTALPROCESSED
FINAL-ZIP-COUNT	PBFN-GSTS-TOTALZIPSASSIGNED
FINAL-LACS-COUNT	PBFN-GSTS-TOTALLACS

Another optional parameter is the 3553Def structure (PBFNNCAS). This structure returns the counts and information that allow you to create your own USPS Form 3553 (CASS Summary Report).

The other parameters allow you to receive *Link statistics.

PBFNTerminate only generates a 0 return code.

Migrating the Terminate (9) Call

This migration table maps the CI Terminate callarea to the Finalist native APIs and structures. The Finalist CI Terminate call is replaced with the Finalist native PBFNTerminate call. Although no parameters were required for the Finalist CI terminate call, the Finalist CI initialization call passed in the Section B information for the USPS Form 3553 (CASS Summary Report). That information is now passed using the PBFNReportData (PBFNRDAT) structure in the Finalist native PBFNTerminate call.

Finalist CI Terminate (9) Call Fields

COBOL C Assembler	Position	Finalist native Structure/Field/ Notes
FINAL-FILLER LanchorPointers \$RREGS	1-12	
FINAL-FUNCTION-CODE cFunctionCode[0] FUNCODE	13	

Migrating the Terminate (9) Return Callarea

This migration table maps the CI Return callarea for the Term call to the Finalist native APIs and structures. The Finalist native PBFNStats or PBFNTerminate call now returns the Finalist CI Terminate call information.

Finalist CI Terminate (9) Return Callarea Fields (Part 1 of 5)

COBOL		
C		
Assembler	Position	CI Structure/Field/Notes
FINAL-ONLINE-RETURN-CODE-9 cOnlineRC OLRETCD	1	If the PBFNProcess call returns PBFN_SUCCESS, then check PBFN-RRTN-RETURNLEVEL or PBFNAddressDataDef.cReturnLevel. <ul style="list-style-type: none"> A value of PBFN_RTNZIP4 indicates a 0. A value of PBFN_RTNZIPCRRT indicates a 1. A value of PBFN_RTNZIP indicates a 2. Any other value indicates a value of 9. If PBFNProcess returns any other value, FINAL-ONLINE-RETURN-CODE is 9.
FINAL-OLD-RETURN-CODE2-9 cOldRC2 OLRETCD2	2	
FINAL-OLD-RETURN-CODE3-9 cOldRC3 OLRETCD3	3	
CAERRMOD-9 caErrorModule CAERRMOD	4-11	
CAERRSRC-9 caResource CAERRSRC	12-19	
CAERRDSC-9 caDescription CAERRDSC	20-28	
FINAL-CERT-NAME caCASSCertName CACERNAME	4-18	PBFN-NCAS-3553 PBFN-NCAS-CASS-SOFTWARE-NAME PBFN3553Def.CASS_Software_Name
FINAL-CERT-VERSION caCASSCertVersion CACERVER	19-28	PBFN-NCAS-3553 PBFN-NCAS-CASS-SOFTWARE-VER PBFN3553Def.CASS_Software_Ver
FINAL-CERT-CONFIG caCASSCertConfig CACERCFG	29-31	PBFN-NCAS-3553 PBFN-NCAS-CASS-CFG PBFN3553Def.CASS_Cfg

Finalist CI Terminate (9) Return Callarea Fields (Part 2 of 5)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-Z4-NAME caZ4ChangeCertName CAZ4NAM	32-46	PBFN-NCAS-3553 PBFN-NCAS-Z4CHG-SOFTWARE-NAM PBFN3553Def.Z4Change_Software_Name
FINAL-Z4-VERSION caZ4ChangeCertVersion CAZ4VER	47-56	PBFN-NCAS-3553 PBFN-NCAS-Z4CHG-SOFTWARE-VER PBFN3553Def.Z4Change_Software_Ver
FINAL-Z4-CONFIG caZ4ChangeCertConfig CAZ4CFG	57-59	PBFN-NCAS-3553 PBFN-NCAS-Z4CHANGE-CFG PBFN3553Def.Z4Change_cfg
FINAL-LOT-NAME caLOTCertName CALOTNAME	60-74	PBFN-NCAS-3553 PBFN-NCAS-LOT-SOFTWARE-NAME PBFN3553Def.LOT_Software_Name
FINAL-LOT-VERSION caLOTCertVersion CALOTVER	75-84	PBFN-NCAS-3553 PBFN-NCAS-LOT-SOFTWARE-VER PBFN3553Def.LOT_Software_Ver
FINAL-LOT-CONFIG caLOTCertConfig CALOTCFG	85-87	PBFN-NCAS-3553 PBFN-NCAS-LOT-UTILITY-CFG PBFN3553Def.LOT_Utility_Cfg
FINAL-DPC-NAME caDPCCertname CADPCNAM	88-102	PBFN-NCAS-3553 PBFN-NCAS-DPBC-SOFTWARE-NAME PBFN3553Def.DPC_Software_Name
FINAL-DPC-VERSION caDPCCertVersion CADPCVER	103-112	PBFN-NCAS-3553 PBFN-NCAS-DPC-SOFTWARE-VER PBFN3553Def.DPC_Software_Ver
FINAL-DPC-CONFIG caDPCCertConfig CADPCCFG	113-115	PBFN-NCAS-3553 PBFN-NCAS-DPC-CFG PBFN3553Def.DPC_Utility_Cfg
FINAL-VALID-FROM caFromValidDate CAVFROM	141-150	PBFN-NCAS-3553 PBFN-NCAS-MF-PROCESS-DATE PBFN3553Def.MasterFile_Process_Date
FINAL-ZIP4-DATE caZIP4Date CAZPDATE	151-160	PBFN-NCAS-3553 PBFN-NCAS-ZIP4-DATABASE-DATE PBFN3553Def.ZIP4_Database_Date
FINAL-Z4-DATE caZ4ChangeDate CAZ4DATE	161-170	PBFN-NCAS-3553 PBFN-NCAS-Z4CHG-DATABASE-DATE PBFN3553Def.Z4Change_Database_Date

Finalist CI Terminate (9) Return Callarea Fields (Part 3 of 5)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-LOT-DATE caLOTDate CALOTDTE	171-180	PBFN-NCAS-3553 PBFN-NCAS-LOT-DATABASE-DATE PBFN3553Def.LOT_Database_Date
FINAL-CRIS-DATE caCRISDate CACRISDT	181-190	PBFN-NCAS-3553 PBFN-NCAS-CRIS-DATABASE-DATE PBFN3553Def.CRIS_Database_Date
FINAL-RECORD-COUNT IRecordCount CARECCNT	193-196	PBFN-NCAS-3553 PBFN-NCAS-TOTAL-RECORDS PBFN3553Def.Total_records
FINAL-ZIP-COUNT IZIPCodedCount CAZIPCNT	197-200	PBFN-NCAS-3553 PBFN-NCAS-ZIP5-TOTAL-CODED PBFN3553Def.Zip5Digit_Total_coded
FINAL-CARRIER-COUNT ICRTCodedCount CACRTCNT	201-204	PBFN-NCAS-3553 PBFN-NCAS-CRIS-TOTAL-CODED PBFN3553Def.CRIS_Total_coded
FINAL-ZIP4-COUNT IZIP4CodedCount CAP4CNT	205-208	PBFN-NCAS-3553 PBFN-NCAS-ZIP4-TOTAL-CODED PBFN3553Def.Zip4_Total_coded
FINAL-DPC-COUNT IDPCCodedCount CADPCCNT	209-212	PBFN-NCAS-3553 PBFN-NCAS-DIRECTDPV-ASSIGNED PBFN3553Def.DPBC_assigned
FINAL-Z4-COUNT IZ4ChangeCodedCount CAZ4CNT	213-216	PBFN-NCAS-3553 PBFN-NCAS-Z4CHANGE-PROCESSED PBFN3553Def.Z4Change_processed
FINAL-LOT-COUNT ILOTCodedCount CALOTCNT	217-220	PBFN-NCAS-3553 PBFN-NCAS-LOT-TOTAL-CODED PBFN3553Def.LOT_Total_coded
FINAL-ZIP-TO-DATE caZIPto CAZIPTO	221-230	PBFN-NCAS-3553 PBFN-NCAS-ZIP5-VALID-TO-DATE PBFN3553Def.Zip5Digit_Valid_To_Date
FINAL-CARRIER-TO-DATE caCRTto CACRTTO	231-240	PBFN-NCAS-3553 PBFN-NCAS-CRIS-VALID-TO-DATE PBFN3553Def.CRIS_Valid_To_Date
FINAL-ZIP4-TO-DATE caZIP4to CAP4TO	241-250	PBFN-NCAS-3553 PBFN-NCAS-ZIP4-VALID-TO-DATE PBFN3553Def.Zip4_Valid_To_Date

Finalist CI Terminate (9) Return Callarea Fields (Part 4 of 5)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-DPC-TO-DATE caDPCto CADPCTO	251-260	PBFN-NCAS-3553 PBFN-NCAS-DIRECTDPV-TO-DATE PBFN3553Def.DPBC_Valid_To_Date
FINAL-LOT-TO-DATE caLOTto CALOTTO	271-280	PBFN-NCAS-3553 PBFN-NCAS-LOT-VALID-TO-DATE PBFN3553Def.LOT_Valid_To_Date
FINAL-HREXACT-COUNT IHRExactCount CAHRECT	281-284	PBFN-NCAS-3553 PBFN-NCAS-HIGHRISEEXACT PBFN3553Def.HighRiseExact
FINAL-HRDFLT-COUNT IHRDefltCount CAHRDCT	285-288	PBFN-NCAS-3553 PBFN-NCAS-HIGHRISEDEFAULT PBFN3553Def.HighRiseDefault
FINAL-RREXCT-COUNT IRRExactCount CARRECT	289-292	PBFN-NCAS-3553 PBFN-NCAS-RURALEXACT PBFN3553Def.RuralExact
FINAL-RRDFLT-COUNT IRRDefltCount CARRDCT	293-296	PBFN-NCAS-3553 PBFN-NCAS-RURALDEFAULT PBFN3553Def.RuralDefault
FINAL-LACS-COUNT ILACSCount CALACCT	297-300	PBFN-NCAS-3553 PBFN-NCAS-LACS PBFN3553Def.LACS
FINAL-DPV-PROCESS-COUNT IDPVCount CADPVCT	301-304	
FINAL-DPV-CONFIRM-COUNT IDPVConfirmCount CADPVYCT	305-308	PBFN-NCAS-3553 PBFN-NCAS-DIRECTDPV-ASSIGNED PBFN3553Def.TotalDPV
FINAL-DPV-CMRA-COUNT IDPVCMRACount CADPVCCT	309-312	
FINAL-EWS-MATCHED-COUNT IEWSMatchCount CAEWSCT	317-320	PBFN-NCAS-3553 PBFN-NCAS-TOTALEWS PBFN3553Def.TotalEWS
FINAL-DPV-DATE caDPVDate CADPVDTE	321-330	PBFN-NCAS-3553 PBFN-NCAS-DPVDATE PBFN3553Def.DPVDate

Finalist CI Terminate (9) Return Callarea Fields (Part 5 of 5)

COBOL

C

Assembler	Position	CI Structure/Field/Notes
FINAL-LACS-SEED-HEADER caLACSSeedHeader CALLKSHD	331-485	PBFN-PSDH-USPS USPSPBDLACSHdrDefinition structure
FINAL-SLK-PROCESS-COUNT ISLKCount SLKICNT	486-489	PBFN-SSLK-RTNSTELNKSTATS PBFN-SSLK-TOTALPROCESSED PBFNRtnSuiteLinkStatsDef.ITotalProcessed
FINAL-SLK-CORRECT-SUITES ISLKCorrectedSuites SLKCCNT	490-493	PBFN-SSLK-RTNSTELNKSTATS PBFN-SSLK-TOTALCORRECTEDSUITES PBFNRtnSuiteLinkStatsDef.ITotalCorrectedSuites
FINAL-SLK-MATCHED-COUNT ISLKSuccessfulMatches SLKSMCNT	494-497	PBFN-SSLK-RTNSTELNKSTATS PBFN-SSLK-TOTALSUCCESSMATCHES PBFNRtnSuiteLinkStatsDef.ITotalSuccessfulMatches

Recent Finalist Changes

This section describes some of the enhancements included in recent Finalist releases.

Finalist 8.3

A new ExtraData (EXTRA-DATA) field was added to the Finalist 8.3 address structures. If the address codes, this field contains all words that Finalist did not use to process the coded address. If an input address does not code, this field is not populated. Finalist accepts two address lines. Finalist may use one or the other or both of these address lines to successfully code an address. If Finalist only uses input line 1, any data in line 2 is not returned in ExtraData. The address lines Finalist used are indicated in the information portion of the returned AddressData structure (PBFN-ADRS-ADDRESSLOC or PBFN-RRTN-ADDRESSLOC).

Finalist includes a simplified interface for using COBOL to call Finalist. Previously, COBOL programs had to be compiled with DLL and PGMNAME(LONG, MIXED). These restrictions have been removed for the main Finalist APIs (the database API interfaces, PBCSxxxx, still require the compile options). Finalist includes eight-character, uppercase COBOL interface names, such as PBFNINIT and PBFNPROC, to simplify calling the Finalist native interface for COBOL applications.

Finalist 9.0

The Finalist 9.0 release introduced a single, simplified address structure called AddressData to replace the multiple variations of ProcessData and ParsedAdr. A ProcessData structure would hold simplified address information (for example, 123 E MAIN ST). A ParsedAdr structure would hold individual address components (Primary range=123, pre-directional=E, Street name = MAIN, suffix = ST). Some information such as city names and county names were contained in both structures. Starting with the Finalist 9.0 release, both structures were combined into a new structure called AddressData (PBFNADRS copybook). In addition, AddressData contains other previously optional parameters from the PBFNProcess call. With the AddressData structure, only a single parameter is passed into PBFNProcess.

In addition to the AddressData structure, the SetupDef structure was updated for the Finalist 9.0 release. Instead of just passing the configuration parameters, the AddressData structure now also includes return information including, but not limited to, version and date of the databases, version and date of the product, and CASS information. This eliminates the need for the old PBFNInfo (PBFNINFO) API.

Finalist 9.1

The Finalist 9.1 release introduced a new API called PBFNTransact (PBFNTRAN). Previously, to process a single address, an application would have to make three calls to Finalist - PBFNInit, PBFNProcess, and PBFNTerminate. PBFNTransact combines all three calls into a single call. The PBFNTransact API accepts two or three parameters - the SetupDef, and AddressData or SetupDef, and input AddressData, and output AddressData. The API is designed for quick processing of single addresses. PBFNTransact does not generate reports, uses minimal memory (storage), and is ideal for online types of applications.

Planning for the Future with Finalist

Although the Finalist CI was designed for a two-year life cycle, the CI is still fully supported. When 64-bit support was recently added in the Finalist 8.3.0 release, the CI was not brought forward due to combined requirements of the 64-bit vs. the CI. At some future date, support for the CI may be eliminated. We encourage Finalist customers to migrate to the Finalist native interface so as not to be impacted if/when support for the CI is discontinued.

The Finalist native interface offers many advantages not included with the Finalist CI. For example, the Finalist native interface allows your applications to run on any platform. This is an important consideration as your company moves forward and possibly away from z/OS.

The ways Finalist customers use clean customer addresses may change or expand, but the need for clean customer addresses will never go away. For example, on one end of the spectrum, the addresses may simply record and ensure customers remain unique and properly identified. A more expanded use may include that addresses are no longer used primarily for USPS mailings but now used for shipping (aka Amazon™, or eBay™). Finalist intends to remain a leader in the address verification market and to continue to be responsive to the new and changing ways our customers maintain and use their customer addresses.

Enhancements and other product suggestions are always welcome.

Technical Support

Questions? You can contact us at <https://www.pitneybowes.com/us/support.html> or by telephone at 1-800-367-6950 to speak with a Finalist Technical Support Representative.

