precisely

Finalist

Finalist Reference Guide

Version 9.22.0

Table of Contents

1 - Using the Finalist APIs

In this section

Using the Database APIs

You can use the Finalist database APIs to retrieve information directly from the Finalist database. The Finalist engine must be initialized prior to using the database APIs. Refer to PBFNInit for more information. The database APIs are listed alphabetically and described in detail.

PBCSCreateSuggestionList

The PBCSCreateSuggestionList function builds a list of suggestions or valid addresses from the database based on a parsed address.

> **Note:** Calling the PBFNProcess function is a prerequisite for the PBCSCreateSuggestionList function. Use the PBFNProcess request return PBFNAddressDataDef structure to populate the DbxGetSuggestionInDef structure.

Syntax

```
int PBFN_API PBCSCreateSuggestionList(pDbxGetSuggestionInDef  In,
                                      pDbxGetSuggestionOutDef * Out,
                                      pDbxStateDef  State);
```

Parameters

Table 1: PBCSCreateSuggestionList API Parameters

Parameter	Description
In	In is a pointer to a data structure of type DbxGetSuggestionInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetSuggestionOutDef. This structure defines the output fields.
State	State is a pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 2: PBCSCreateSuggestionList Return Values

Return Value	Description	Value
n	Number of suggestions returned.	Value in the range of 0 up to listSize.

Related APIs

PBFNProcess

Related Structures

PBFNAddressDataDef

DbxGetSuggestionInDef

DbxGetSuggestionOutDef

DbxStateDef

PBCSCreateSuggestionListFlat

The PBCSCreateSuggestionListFlat function builds a list of suggestions or valid addresses from the database based on a parsed address.

> **Note:** Calling the PBFNProcess function is a prerequisite for the PBCSCreateSuggestionListFlat function. Use the PBFNProcess request return PBFNAddressDataDef structure to populate the DbxGetSuggestionInDef structure.

Syntax

```
int PBCSCreateSuggestionListFlat(pDbxGetSuggestionInDef  In,
                                 pDbxGetSuggestionOutDef Out1,
                                 pDbxGetSuggestionOutDef Out2,
                                 pDbxGetSuggestionOutDef Out3,
                                 pDbxGetSuggestionOutDef Out4,
                                 pDbxGetSuggestionOutDef Out5,
                                 pDbxGetSuggestionOutDef Out6,
                                 pDbxGetSuggestionOutDef Out7,
                                 pDbxGetSuggestionOutDef Out8,
```

```
                                    pDbxGetSuggestionOutDef Out9,
                                    pDbxGetSuggestionOutDef Out10,
                                    pDbxStateDef  State);
```

Parameters

Table 3: PBCSCreateSuggestionListFlat API Parameters

Parameter	Description
In	In is a pointer to a data structure of type DbxGetSuggestionInDef. This structure defines the required input fields.
Out1 through Out10	Out1 through Out10 are pointers to a data structure of type DbxGetSuggestionOutDef. This structure defines the output fields.
State	State is a pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 4: PBCSCreateSuggestionListFlat Return Values

Return Value	Description	Value
n	Number of suggestions returned.	A value in the range of 0 up to listSize.

Related APIs

PBFNProcess

Related Structures

PBFNAddressDataDef

DbxGetSuggestionInDef

DbxGetSuggestionOutDef

DbxStateDef

PBCSGetCityList

The PBCSGetCityList function builds a list of cities based on the first characters of a city name.

Syntax

```
int PBCSGetCityList(pDbxGetZipCityInDef  In,
                    pDbxGetZipOutDef * Out,
                    pDbxStateDef  State);
```

Parameters

Table 5: PBCSGetCityList API Parameters

Parameter	Description
In	In is a pointer to a data structure of type DbxGetZipCityInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetZipOutDef. This structure defines the output fields.
State	State is a pointer to a data structure of type DbxStateDef. Use the state pointer to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 6: PBCSGetCityList Return Values

Return Value	Description	Value
n	Number of cities that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipCityInDef

DbxGetZipOutDef

DbxStateDef

PBCSGetCityListFlat

The PBCSGetCityListFlat function builds a list of cities based on the first characters of a city name.

Syntax

```
int PBCSGetCityListFlat(pDbxGetZipCityInDef  In,
                        pDbxGetZipOutDef  Out1,
                        pDbxGetZipOutDef  Out2,
                        pDbxGetZipOutDef  Out3,
                        pDbxGetZipOutDef  Out4,
                        pDbxGetZipOutDef  Out5,
                        pDbxGetZipOutDef  Out6,
                        pDbxGetZipOutDef  Out7,
                        pDbxGetZipOutDef  Out8,
                        pDbxGetZipOutDef  Out9,
                        pDbxGetZipOutDef  Out10,
                        pDbxStateDef  State);
```

Parameters

Table 7: PBCSGetCityListFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetZipCityInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetZipOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 8: PBCSGetCityListFlat Return Values

Return Value	Description	Value
n	Number of cities that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipCityInDef

DbxGetZipOutDef

DbxStateDef

PBCSGetFirmListCity

The PBCSGetFirmListCity function builds a list of firms based on a provided city, state, street, and either the primary range or secondary range.

> **Note:** The PBCSGetStreetListCity function is a prerequisite for the PBCSGetFirmListCity function. Use the street list returned from PBCSGetStreetListCity to pass a street index to PBCSGetPrimaryListCity to return the primary range values. The PBCSGetPrimaryListCity function is a prerequisite for the PBCSGetFirmListCity function. Use the primary range list returned from PBCSGetPrimaryListCity to pass a primary range index to PBCSGetSecondaryListCity to return the secondary range values. The PBCSGetSecondaryListZip function is a prerequisite for the PBCSGetFirmListCity function. Use the secondary range list returned from PBCSGetSecondaryListZip to pass a secondary range index to PBCSGetFirmListCity to return the firm list.

Syntax

```
int PBCSGetCityList(pDbxGetZipCityInDef  In,
                    pDbxGetZipOutDef * Out,
                    pDbxStateDef  State);
```

Parameters

Table 9: PBCSGetFirmListCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetFirmInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetFirmOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 10: PBCSGetFirmListCity Return Values

Return Value	Description	Value
n	Number of firms that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListCity

PBCSGetPrimaryListCity

PBCSGetSecondaryListCity

Related Structures

DbxGetFirmInDef

DbxGetFirmOutDef

DbxStateDef

PBCSGetFirmListCityFlat

The PBCSGetFirmListCityFlat function builds a list of firms based on a provided city, state, street, and either the primary range or secondary range.

> **Note:** The PBCSGetStreetListCityFlat function is a prerequisite for the PBCSGetFirmListCityFlat function. Use the street list returned from PBCSGetStreetListCityFlat to pass a street index to PBCSGetPrimaryListCityFlat to return the primary range values. The PBCSGetPrimaryListCityFlat function is a prerequisite for the PBCSGetFirmListCityFlat function. Use the primary range list returned from PBCSGetPrimaryListCityFlat to pass a primary range index to PBCSGetSecondaryListCityFlat to return the secondary range values. The PBCSGetSecondaryListZipFlat function is a prerequisite for the PBCSGetFirmListCityFlat function. Use the secondary range list returned from PBCSGetSecondaryListZipFlat to pass a secondary range index to PBCSGetFirmListCityFlat to return the firm list.

Syntax

```
int PBCSGetFirmListCityFlat(pDbxGetFirmInDef  In,
                            pDbxGetFirmOutDef Out1,
                            pDbxGetFirmOutDef Out2,
                            pDbxGetFirmOutDef Out3,
                            pDbxGetFirmOutDef Out4,
                            pDbxGetFirmOutDef Out5,
                            pDbxGetFirmOutDef Out6,
                            pDbxGetFirmOutDef Out7,
                            pDbxGetFirmOutDef Out8,
                            pDbxGetFirmOutDef Out9,
                            pDbxGetFirmOutDef Out10,
                            pDbxStateDef  State);
```

Parameters

Table 11: PBCSGetFirmListCityFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetFirmInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetFirmOutDef. This structure defines the output fields.

Parameter	Description
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 12: PBCSGetFirmListCityFlat Return Values

Return Value	Description	Value
n	Number of firms that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListCityFlat

PBCSGetPrimaryListCityFlat

PBCSGetSecondaryListZipFlat

Related Structures

DbxGetFirmInDef

DbxGetFirmOutDef

DbxStateDef

PBCSGetFirmListZip

The PBCSGetFirmListZip function builds a list of firms based on a provided ZIP Code, street, and either the primary or secondary range.

> **Note:** The PBCSGetStreetListZip function is a prerequisite for the PBCSGetFirmListZip function. Use the street list returned from PBCSGetStreetListZip to pass a street index to PBCSGetPrimaryListZip to return the primary range values. The PBCSGetPrimaryListZip function is a prerequisite for the PBCSGetFirmListZip function. Use the primary range list returned from PBCSGetPrimaryListZip to pass a primary range index to PBCSGetSecondaryListZip to return the secondary range values. The PBCSGetSecondaryListZip function is a prerequisite for the PBCSGetFirmListZip function.

Use the secondary range list returned from PBCSGetSecondaryListZip to pass a secondary range index to PBCSGetFirmListZip to return the firm list.

Syntax

```
int PBCSGetFirmListZip(pDbxGetFirmInDef  In,
                       pDbxGetFirmOutDef * Out,
                       pDbxStateDef  State);
```

Parameters

Table 13: PBCSGetFirmListZip API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetFirmInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetFirmOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 14: PBCSGetFirmListZip Return Values

Return Value	Description	Value
n	Number of firms that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListZip

PBCSGetPrimaryListZip

PBCSGetSecondaryListZip

Related Structures

DbxGetFirmInDef

DbxGetFirmOutDef

DbxStateDef

PBCSGetFirmListZipFlat

The PBCSGetFirmListZipFlat function builds a list of firms based on a provided ZIP Code, street, and either the primary or secondary range.

> **Note:** The PBCSGetStreetListZipFlat function is a prerequisite for the PBCSGetFirmListZipFlat function. Use the street list returned from PBCSGetStreetListZipFlat to pass a street index to PBCSGetPrimaryListZipFlat to return the primary range values. The PBCSGetPrimaryListZipFlat function is a prerequisite for the PBCSGetFirmListZipFlat function. Use the primary range list returned from PBCSGetPrimaryListZipFlat to pass a primary range index to PBCSGetSecondaryListZipFlat to return the secondary range values. The PBCSGetSecondaryListZipFlat function is a prerequisite for the PBCSGetFirmListZipFlat function. Use the secondary range list returned from PBCSGetSecondaryListZipFlat to pass a secondary range index to PBCSGetFirmListZipFlat to return the firm list.

Syntax

```
int PBFN_API PBCSGetFirmListZipFlat(pDbxGetFirmInDef  In,
                                    pDbxGetFirmOutDef Out1,
                                    pDbxGetFirmOutDef Out2,
                                    pDbxGetFirmOutDef Out3,
                                    pDbxGetFirmOutDef Out4,
                                    pDbxGetFirmOutDef Out5,
                                    pDbxGetFirmOutDef Out6,
                                    pDbxGetFirmOutDef Out7,
                                    pDbxGetFirmOutDef Out8,
                                    pDbxGetFirmOutDef Out9,
                                    pDbxGetFirmOutDef Out10,
                                    pDbxStateDef  State);
```

Parameters

Table 15: PBCSGetFirmListZipFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetFirmInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetFirmOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 16: PBCSGetFirmListZipFlat Return Values

Return Value	Description	Value
n	Number of firms that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListZipFlat

PBCSGetPrimaryListZipFlat

PBCSGetSecondaryListZipFlat

Related Structures

DbxGetFirmInDef

DbxGetFirmOutDef

DbxStateDef

PBCSGetPrimaryListCity

The PBCSGetPrimaryListCity function builds a list of primary ranges based on a provided city, state, and street.

> **Note:** The PBCSGetStreetListCity function is a prerequisite for the PBCSGetPrimaryListCity function. Use the street list returned from PBCSGetStreetListCity to pass a street index to PBCSGetPrimaryListCity to return the primary range values.

Syntax

```
int PBCSGetPrimaryListCity(pDbxGetRangeInDef  In,
                           pDbxGetRangeOutDef * Out,
                           pDbxStateDef  State);
```

Parameters

Table 17: PBCSGetPrimaryListCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 18: PBCSGetPrimaryListCity Return Values

Return Value	Description	Value
n	Number of primary ranges that match the search criteria.	A value in the range of 0 up to count.

Related API

PBCSGetStreetListCity

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetPrimaryListCityFlat

The PBCSGetPrimaryListCityFlat function builds a list of primary ranges based on a provided city, state, and street.

> **Note:** The PBCSGetStreetListCityFlat function is a prerequisite for the PBCSGetPrimaryListCityFlat function. Use the street list returned from PBCSGetStreetListCityFlat to pass a street index to PBCSGetPrimaryListCityFlat to return the primary range values.

Syntax

```
int PBCSGetPrimaryListCityFlat(pDbxGetRangeInDef  In,
                               pDbxGetRangeOutDef Out1,
                               pDbxGetRangeOutDef Out2,
                               pDbxGetRangeOutDef Out3,
                               pDbxGetRangeOutDef Out4,
                               pDbxGetRangeOutDef Out5,
                               pDbxGetRangeOutDef Out6,
                               pDbxGetRangeOutDef Out7,
                               pDbxGetRangeOutDef Out8,
                               pDbxGetRangeOutDef Out9,
                               pDbxGetRangeOutDef Out10,
                               pDbxStateDef  State);
```

Parameters

Table 19: PBCSGetPrimaryListCityFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 20: PBCSGetPrimaryListCityFlat Return Values

Return Value	Description	Value
n	Number of primary ranges that match the search criteria.	A value in the range of 0 up to count.

Related API

PBCSGetStreetListCityFlat

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetPrimaryListZip

The PBCSGetPrimaryListZip function builds a list of primary ranges based on a provided ZIP Code and street.

> **Note:** The PBCSGetStreetListZip function is a prerequisite for the PBCSGetPrimaryListZip function. Use the street list returned for PBCSGetStreetListZip to pass a street index to PBCSGetPrimaryListZip to return the primary range values.

Syntax

```
int PBCSGetPrimaryListZip(pDbxGetRangeInDef  In,
                          pDbxGetRangeOutDef * Out,
                          pDbxStateDef  State);
```

Parameters

Table 21: PBCSGetPrimaryListZip API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 22: PBCSGetPrimaryListZip Return Values

Return Value	Description	Value
n	Number of primary ranges that match the search criteria.	A value in the range of 0 up to count.

Related API

PBCSGetStreetListZip

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetPrimaryListZipFlat

The PBCSGetPrimaryListZipFlat function builds a list of primary ranges based on a provided ZIP Code and street.

> **Note:** The PBCSGetStreetListZipFlat function is a prerequisite for the PBCSGetPrimaryListZipFlat function. Use the street list returned for PBCSGetStreetListZipFlat to pass a street index to PBCSGetPrimaryListZipFlat to return the primary range values.

Syntax

```
int PBCSGetPrimaryListZipFlat(pDbxGetRangeInDef  In,
                              pDbxGetRangeOutDef Out1,
                              pDbxGetRangeOutDef Out2,
                              pDbxGetRangeOutDef Out3,
                              pDbxGetRangeOutDef Out4,
                              pDbxGetRangeOutDef Out5,
                              pDbxGetRangeOutDef Out6,
                              pDbxGetRangeOutDef Out7,
                              pDbxGetRangeOutDef Out8,
                              pDbxGetRangeOutDef Out9,
                              pDbxGetRangeOutDef Out10,
                              pDbxStateDef  State);
```

Parameters

Table 23: PBCSGetPrimaryListZipFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 24: PBCSGetPrimaryListZipFlat Return Values

Return Value	Description	Value
n	Number of primary ranges that match the search criteria.	A value in the range of 0 up to count.

Related API

PBCSGetStreetListZipFlat

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetSecondaryListCity

The PBCSGetSecondaryListCity function builds a list of secondary ranges based on a provided city, state, street, and primary range.

> **Note:** The PBCSGetStreetListCity function is a prerequisite for the PBCSGetSecondaryListCity function. Use the street list returned from PBCSGetStreetListCity to pass a street index to PBCSGetPrimaryListCity to return the primary range values. The PBCSGetPrimaryListCity function is a prerequisite for the PBCSGetSecondaryListCity function. Use the primary range list returned from PBCSGetPrimaryListCity to pass a primary range index to PBCSGetSecondaryListCity to return the secondary range values.

Syntax

```
int PBCSGetSecondaryListCity(pDbxGetRangeInDef  In,
                             pDbxGetSecRangeOutDef * Out,
                             pDbxStateDef  State);
```

Parameters

Table 25: PBCSGetSecondaryListCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetSecRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 26: PBCSGetSecondaryListCity Return Values

Return Value	Description	Value
n	Number of secondary ranges that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListCity

PBCSGetPrimaryListCity

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetSecondaryListCityFlat

The PBCSGetSecondaryListCityFlat function builds a list of secondary ranges based on a provided city, state, street, and primary range.

> **Note:** The PBCSGetStreetListCityFlat function is a prerequisite for the PBCSGetSecondaryListCityFlat function. Use the street list returned from PBCSGetStreetListCityFlat to pass a street index to PBCSGetPrimaryListCityFlat to return the primary range values. The PBCSGetPrimaryListCityFlat function is a prerequisite for the PBCSGetSecondaryListCityFlat function. Use the primary range list returned from PBCSGetPrimaryListCityFlat to pass a primary range index to PBCSGetSecondaryListCityFlat to return the secondary range values.

Syntax

```
int PBCSGetSecondaryListCityFlat(pDbxGetRangeInDef  In,
                                 pDbxGetSecRangeOutDef Out1,
                                 pDbxGetSecRangeOutDef Out2,
                                 pDbxGetSecRangeOutDef Out3,
```

```
                                pDbxGetSecRangeOutDef Out4,
                                pDbxGetSecRangeOutDef Out5,
                                pDbxGetSecRangeOutDef Out6,
                                pDbxGetSecRangeOutDef Out7,
                                pDbxGetSecRangeOutDef Out8,
                                pDbxGetSecRangeOutDef Out9,
                                pDbxGetSecRangeOutDef Out10,
                                pDbxStateDef  State);
```

Parameters

Table 27: PBCSGetSecondaryListCityFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetSecRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 28: PBCSGetSecondaryListCityFlat Return Values

Return Value	Description	Value
n	Number of secondary ranges that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListCityFlat

PBCSGetPrimaryListCityFlat

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef.dita

PBCSGetSecondaryListZip

The PBCSGetSecondaryListZip function builds a list of secondary ranges based on a provided ZIP Code.

> **Note:** The PBCSGetStreetListZip function is a prerequisite for the PBCSGetSecondaryListZip function. Use the street list returned from PBCSGetStreetListZip to pass a street index to PBCSGetPrimaryListZip to return the primary range values. Use the primary range list returned from PBCSGetPrimaryListZip to pass a primary range index to PBCSGetSecondaryListZip to return the secondary range values.

Syntax

```
int PBCSGetSecondaryListZip(pDbxGetRangeInDef  In,
                            pDbxGetSecRangeOutDef * Out,
                            pDbxStateDef  State);
```

Parameters

Table 29: PBCSGetSecondaryListZip API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetSecRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 30: PBCSGetSecondaryListZip Return Values

Return Value	Description	Value
n	Number of secondary ranges that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListZip

PBCSGetPrimaryListZip

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetSecondaryListZipFlat

The PBCSGetSecondaryListZipFlat function builds a list of secondary ranges based on a provided ZIP Code.

> **Note:** The PBCSGetStreetListZipFlat function is a prerequisite for the PBCSGetSecondaryListZipFlat function. Use the street list returned from PBCSGetStreetListZipFlat to pass a street index to PBCSGetPrimaryListZipFlat to return the primary range values. Use the primary range list returned from PBCSGetPrimaryListZipFlat to pass a primary range index to PBCSGetSecondaryListZipFlat to return the secondary range values.

Syntax

```
int PBCSGetSecondaryListZipFlat(pDbxGetRangeInDef  In,
                                pDbxGetSecRangeOutDef Out1,
                                pDbxGetSecRangeOutDef Out2,
                                pDbxGetSecRangeOutDef Out3,
                                pDbxGetSecRangeOutDef Out4,
```

```
                                pDbxGetSecRangeOutDef Out5,
                                pDbxGetSecRangeOutDef Out6,
                                pDbxGetSecRangeOutDef Out7,
                                pDbxGetSecRangeOutDef Out8,
                                pDbxGetSecRangeOutDef Out9,
                                pDbxGetSecRangeOutDef Out10,
                                pDbxStateDef  State);
```

Parameters

Table 31: PBCSGetSecondaryListZipFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetRangeInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetSecRangeOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 32: PBCSGetSecondaryListZipFlat Return Values

Return Value	Description	Value
n	Number of secondary ranges that match the search criteria.	A value in the range of 0 up to count.

Related APIs

PBCSGetStreetListZipFlat

PBCSGetPrimaryListZipFlat

Related Structures

DbxGetRangeInDef

DbxGetRangeOutDef

DbxStateDef

PBCSGetStreetAliasCity

The PBCSGetStreetAliasCity function builds a list of alias streets or rural routes based on a provided city.

> **Note:** The PBCSGetPrimaryListCity function is a prerequisite for the PBCSGetStreetAliasCity function. Some cities may be aliases as indicated by the alias field in pDbxGetRangeOutDef. An application such as a custom Lookup Tool would call PBCSGetStreetAliasCity to get information about that real city. Before calling PBCSGetPrimaryListCity, you would need to call PBCSGetStreetListCity and PBCSGetCityList.

Syntax

```
int PBCSGetStreetAliasCity(pDbxGetAliasInDef  In,
                           pDbxGetAliasOutDef Out,
                           pDbxStateDef  State);
```

Parameters

Table 33: PBCSGetStreetAliasCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetAliasInDef. This structure defines the required input fields.
Out	Pointer to a data structure of type DbxGetAliasOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 34: PBCSGetStreetAliasCity Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully returned alias information.	1
PBFN_FAIL	Unsuccessful return of alias information. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related APIs

PBCSGetStreetListCity

PBCSGetStreetListCity

Related Structures

DbxGetAliasInDef

DbxGetAliasOutDef

DbxStateDef

PBFNExtendedErrorDef

PBCSGetStreetAliasCityFlat

The PBCSGetStreetAliasCityFlat function builds a list of alias streets or rural routes based on a provided city.

> **Note:** The PBCSGetPrimaryListCityFlat function is a prerequisite for the PBCSGetStreetAliasCityFlat function. Some cities may be aliases as indicated by the alias field in pDbxGetRangeOutDef. An application such as a custom Lookup Tool would call PBCSGetStreetAliasCityFlat to get information about that real city. Before calling PBCSGetPrimaryListCityFlat, you would need to call PBCSGetStreetListCityFlat and PBCSGetCityListFlat.

Syntax

```
int PBCSGetStreetAliasCityFlat(pDbxGetAliasInDef  In,
                               pDbxGetAliasOutDef Out1,
                               pDbxStateDef  State);
```

Parameters

Table 35: PBCSGetStreetAliasCityFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetAliasInDef. This structure defines the required input fields.
Out1	Pointer to a data structure of type DbxGetAliasOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 36: PBCSGetStreetAliasCityFlat Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully returned alias information.	1
PBFN_FAIL	Unsuccessful return of alias information. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related APIs

PBCSGetCityListFlat

PBCSGetStreetListCityFlat

PBCSGetPrimaryListCity

Related Structures

DbxGetAliasInDef

DbxGetAliasOutDef

DbxGetRangeOutDef

DbxStateDef

PBFNExtendedErrorDef

PBCSGetStreetAliasZip

The PBCSGetStreetAliasZip function builds a list of alias streets or rural routes based on a provided ZIP Code.

> **Note:** NOTE: The PBCSGetPrimaryListZip function is a prerequisite for the PBCSGetStreetAliasZip function. Some primaries may be aliases as indicated by the alias field in pDbxGetRangeOutDef. An application such as a custom Lookup Tool would call PBCSGetStreetAliasZip to get information about that real primary. Before calling PBCSGetPrimaryListZip, you would need to call PBCSGetStreetListZip to get a list of primaries and PBCSGetZipList to get a list of ZIP Codes.

Syntax

```
int PBCSGetStreetAliasZip(pDbxGetAliasInDef  In,
                          pDbxGetAliasOutDef Out,
                          pDbxStateDef  State);
```

Parameters

Table 37: PBCSGetStreetAliasZip API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetAliasInDef. This structure defines the required input fields.
Out	Pointer to a data structure of type DbxGetAliasOutDef. This structure defines the output fields.

Parameter	Description
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 38: PBCSGetStreetAliasZip Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully returned alias information.	1
PBFN_FAIL	Unsuccessful return of alias information. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related APIs

PBCSGetPrimaryListZip

PBCSGetStreetListZip

PBCSGetZipList

Related Structures

DbxGetAliasInDef

DbxGetAliasOutDef

DbxGetRangeOutDef

DbxStateDef

PBFNExtendedErrorDef

PBCSGetStreetAliasZipFlat

The PBCSGetStreetAliasZipFlat function builds a list of alias streets or rural routes based on a provided ZIP Code.

> **Note:** The PBCSGetPrimaryListZipFlat function is a prerequisite for the PBCSGetStreetAliasZipFlat function. Some primaries may be aliases as indicated by the alias field in pDbxGetRangeOutDef. An application such as a custom Lookup Tool would call PBCSGetStreetAliasZipFlat to get information about that real primary. Before calling PBCSGetPrimaryListZipFlat, you would need to call PBCSGetStreetListZipFlat to get a list of primaries and PBCSGetZipListFlat to get a list of ZIP Codes.

Syntax

```
int PBCSGetStreetAliasZipFlat(pDbxGetAliasInDef  In,
                              pDbxGetAliasOutDef Out1,
                              pDbxStateDef  State);
```

Parameters

Table 39: PBCSGetStreetAliasZipFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetAliasInDef. This structure defines the required input fields.
Out1	Pointer to a data structure of type DbxGetAliasOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 40: PBCSGetStreetAliasZipFlat Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully returned alias information.	1
PBFN_FAIL	Unsuccessful return of alias information. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related APIs

PBCSGetPrimaryListZip

PBCSGetStreetListZip

PBCSGetZipList

Related Structures

DbxGetAliasInDef

DbxGetAliasOutDef

DbxGetRangeOutDef

DbxStateDef

PBFNExtendedErrorDef

PBCSGetStreetListCity

The PBCSGetStreetListCity function builds a list of streets or rural routes based on a provided city and state.

Syntax

```
int PBCSGetStreetListCity(pDbxGetStreetInDef  In,
                          pDbxGetStreetOutDef * Out,
                          pDbxStateDef  State);
```

Parameters

Table 41: PBCSGetStreetListCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetStreetOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 42: PBCSGetStreetListCity Return Values

Return Value	Description	Value
n	Number of streets or rural routes that match the search criteria or zero if no streets match the search criteria or the city is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetInDef

DbxGetStreetOutDef

DbxStateDef

PBCSGetStreetListCityFlat

The PBCSGetStreetListCityFlat function builds a list of streets or rural routes based on a provided city and state.

Syntax

```
int PBCSGetStreetListCityFlat(pDbxGetStreetInDef  In,
                              pDbxGetStreetOutDef  Out1,
                              pDbxGetStreetOutDef  Out2,
                              pDbxGetStreetOutDef  Out3,
                              pDbxGetStreetOutDef  Out4,
                              pDbxGetStreetOutDef  Out5,
                              pDbxGetStreetOutDef  Out6,
                              pDbxGetStreetOutDef  Out7,
                              pDbxGetStreetOutDef  Out8,
                              pDbxGetStreetOutDef  Out9,
                              pDbxGetStreetOutDef  Out10,
                              pDbxStateDef State);
```

Parameters

Table 43: PBCSGetStreetListCityFlat API Parameters

Parameter	Description
In	In is a pointer to a data structure of type DbxGetStreetInDef. This structure defines the required input fields.
Out1 through Out10	Out1 through Out10 are pointers to a data structure of type DbxGetStreetOutDef. This structure defines the output fields.
State	State is a pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 44: PBCSGetStreetListCityFlat Return Values

Return Value	Description	Value
n	Number of streets or rural routes that match the search criteria or zero if no streets match the search criteria or the city is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetInDef

DbxGetStreetOutDef

DbxStateDef

PBCSGetStreetListCityQ

The PBCSGetStreetListCityQ function builds a list of streets based on a provided city, state, and the beginning letters in the street name.

Syntax

```
int PBCSGetStreetListCityQ(pDbxGetStreetQInDef In,
                           pDbxGetStreetQOutDef * Out,
                           pDbxStateDef  State);
```

Parameters

Table 45: PBCSGetStreetListCityQ API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetQInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetStreetQOutDef. This structure defines the output fields.

Parameter	Description
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 46: PBCSGetStreetListCityQ Return Values

Return Value	Description	Value
n	Number of streets that match the search criteria or zero if no streets match the search criteria or the city is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetQInDef

DbxGetStreetQOutDef

DbxStateDef

PBCSGetStreetListCityQFlat

The PBCSGetStreetListCityQFlat function builds a list of streets based on a provided city, state, and the beginning letters in the street name.

Syntax

```
int PBCSGetStreetListCityQFlat(pDbxGetStreetQInDef In,
                               pDbxGetStreetQOutDef Out1,
                               pDbxGetStreetQOutDef Out2,
                               pDbxGetStreetQOutDef Out3,
                               pDbxGetStreetQOutDef Out4,
                               pDbxGetStreetQOutDef Out5,
                               pDbxGetStreetQOutDef Out6,
                               pDbxGetStreetQOutDef Out7,
                               pDbxGetStreetQOutDef Out8,
                               pDbxGetStreetQOutDef Out9,
```

```
                                pDbxGetStreetQOutDef Out10,
                                pDbxStateDef  State);
```

Parameters

Table 47: PBCSGetStreetListCityQFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetQInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetStreetQOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values.

Table 48: PBCSGetStreetListCityQFlat Return Values

Return Value	Description	Value
n	Number of streets that match the search criteria or zero if no streets match the search criteria or the city is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetQInDef

DbxGetStreetQOutDef

DbxStateDef

PBCSGetStreetListZip

The PBCSGetStreetListZip function builds a list of streets or rural routes based on a provided ZIP Code.

Syntax

```
int PBCSGetStreetListZip(pDbxGetStreetInDef  In,
                         pDbxGetStreetOutDef * Out,
                         pDbxStateDef  State);
```

Parameters

Table 49: PBCSGetStreetListZip API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetStreetOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 50: PBCSGetStreetListZip Return Values

Return Value	Description	Value
n	Number of streets or rural routes that match the search criteria or zero if no streets match the search criteria or if the ZIP Code is invalid.	A value in the range of 0 up to count.

Example

This example retrieves groups of 30 streets starting at the 40th street. ZIP Code 60148 has 267 streets. This example will loop through returning 30 streets until reaching 267.

```
PBCSState.lCount = 30;
PBCSState.lStartElem = 40;
do
{
    count = PBCSGetStreetListZip( PBCSstreetIn,
                                  PBCSstreetList, &PBCSState);
    PBCSState.lStartElem =  PBCSState.lStartElem +
         PBCSState.lCount;
} while ( PBCSState.cMore == 'Y'  );
```

Related Structures

DbxGetStreetInDef

DbxGetStreetOutDef

DbxStateDef

PBCSGetStreetListZipFlat

The PBCSGetStreetListZipFlat function builds a list of streets or rural routes based on a provided ZIP Code.

Syntax

```
int PBCSGetStreetListZipFlat(pDbxGetStreetInDef  In,
                             pDbxGetStreetOutDef  Out1,
                             pDbxGetStreetOutDef  Out2,
                             pDbxGetStreetOutDef  Out3,
                             pDbxGetStreetOutDef  Out4,
                             pDbxGetStreetOutDef  Out5,
                             pDbxGetStreetOutDef  Out6,
                             pDbxGetStreetOutDef  Out7,
                             pDbxGetStreetOutDef  Out8,
                             pDbxGetStreetOutDef  Out9,
                             pDbxGetStreetOutDef  Out10,
                             pDbxStateDef State);
```

Parameters

Table 51: PBCSGetStreetListZipFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetStreetOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 52: PBCSGetStreetListZipFlat Return Values

Return Value	Description	Value
n	Number of streets or rural routes that match the search criteria or zero if no streets match the search criteria or if the ZIP Code is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetInDef

DbxGetStreetOutDef

DbxStateDef

PBCSGetStreetListZipQ

The PBCSGetStreetListZipQ function builds a list of streets based on a provided ZIP Code and the beginning letters in the street name.

Syntax

```
int PBCSGetStreetListZipQ(pDbxGetStreetQInDef In,
                          pDbxGetStreetQOutDef * Out,
                          pDbxStateDef  State);
```

Parameters

Table 53: PBCSGetStreetListZipQ API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetQInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetStreetQOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 54: PBCSGetStreetListZipQ Return Values

Return Value	Description	Value
n	Number of streets that match the search criteria or zero if no streets match the search criteria or if the ZIP Code is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetQInDef

DbxGetStreetQOutDef

DbxStateDef

PBCSGetStreetListZipQFlat

The PBCSGetStreetListZipQFlat function builds a list of streets based on a provided ZIP Code and the beginning letters in the street name.

Syntax

```
int PBCSGetStreetListZipQFlat(pDbxGetStreetQInDef In,
                              pDbxGetStreetQOutDef Out1,
                              pDbxGetStreetQOutDef Out2,
                              pDbxGetStreetQOutDef Out3,
                              pDbxGetStreetQOutDef Out4,
                              pDbxGetStreetQOutDef Out5,
                              pDbxGetStreetQOutDef Out6,
                              pDbxGetStreetQOutDef Out7,
                              pDbxGetStreetQOutDef Out8,
                              pDbxGetStreetQOutDef Out9,
                              pDbxGetStreetQOutDef Out10,
                              pDbxStateDef  State);
```

Parameters

Table 55: PBCSGetStreetListZipQFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetStreetQInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetStreetQOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 56: PBCSGetStreetListZipQFlat Return Values

Return Value	Description	Value
n	Number of streets that match the search criteria or zero if no streets match the search criteria or if the ZIP Code is invalid.	A value in the range of 0 up to count.

Related Structures

DbxGetStreetQInDef

DbxGetStreetQOutDef

DbxStateDef

PBCSGetZipList

The PBCSGetZipList function builds a list of ZIP Codes based on the search criteria.

Syntax

```
int PBCSGetZipList(pDbxGetZipInDef  In,
                   pDbxGetZipOutDef * Out,
                   pDbxStateDef State);
```

Parameters

Table 57: PBCSGetZipList API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetZipInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetZipOutDef. This structure defines the output fields.

Parameter	Description
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 58: PBCSGetZipList Return Values

Return Value	Description	Value
n	Number of ZIP Codes that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipInDef

DbxGetZipOutDef

DbxStateDef

PBCSGetZipListFlat

The PBCSGetZipListFlat function builds a list of ZIP Codes based on the search criteria.

Syntax

```
int PBCSGetZipListFlat(pDbxGetZipInDef In,
                       pDbxGetZipOutDef  Out1,
                       pDbxGetZipOutDef  Out2,
                       pDbxGetZipOutDef  Out3,
                       pDbxGetZipOutDef  Out4,
                       pDbxGetZipOutDef  Out5,
                       pDbxGetZipOutDef  Out6,
                       pDbxGetZipOutDef  Out7,
                       pDbxGetZipOutDef  Out8,
                       pDbxGetZipOutDef  Out9,
                       pDbxGetZipOutDef  Out10,
                       pDbxStateDef  State);
```

Parameters

Table 59: PBCSGetZipListFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetZipInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetZipOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 60: PBCSGetZipListFlat Return Values

Return Value	Description	Value
n	Number of ZIP Codes that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipInDef

DbxGetZipOutDef

DbxStateDef

PBCSGetZipListCity

The PBCSGetZipListCity function builds a list of ZIP Codes for a given city and state.

Syntax

```
int PBCSGetZipListCity(pDbxGetZipCityInDef  In,
                       pDbxGetZipOutDef * Out,
                       pDbxStateDef State);
```

Parameters

Table 61: PBCSGetZipListCity API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetZipCityInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxGetZipOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 62: PBCSGetZipListCity Return Values

Return Value	Description	Value
n	Number of ZIP Codes that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipCityInDef

DbxGetZipOutDef

DbxStateDef

PBCSGetZipListCityFlat

The PBCSGetZipListCityFlat function builds a list of ZIP Codes for a given city and state.

Syntax

```
int PBCSGetZipListCityFlat(pDbxGetZipCityInDef In,
                           pDbxGetZipOutDef Out1,
                           pDbxGetZipOutDef Out2,
                           pDbxGetZipOutDef Out3,
                           pDbxGetZipOutDef Out4,
                           pDbxGetZipOutDef Out5,
                           pDbxGetZipOutDef Out6,
                           pDbxGetZipOutDef Out7,
                           pDbxGetZipOutDef Out8,
                           pDbxGetZipOutDef Out9,
                           pDbxGetZipOutDef Out10,
                           pDbxStateDef State);
```

Parameters

Table 63: PBCSGetZipListCityFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxGetZipCityInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxGetZipOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 64: PBCSGetZipListCityFlat Return Values

Return Value	Description	Value
n	Number of ZIP Codes that match the search criteria.	A value in the range of 0 up to count.

Related Structures

DbxGetZipCityInDef

DbxGetZipOutDef

DbxStateDef

PBCSLastLine

The PBCSLastLine function builds a list of matching cities, states, and ZIP Codes based on the last line of an address.

Syntax

```
int PBCSLastLine (pDbxLastLineInDef  In,
                  pDbxLastLineOutDef * Out,
                  pDbxStateDef State);
```

Parameters

Table 65: PBCSLastLine API Parameters

Parameters	Description
In	Pointer to a data structure of type DbxLastLineInDef. This structure defines the required input fields.
Out	Pointer to a list of pointers to data structures of type DbxLastLineOutDef. This structure defines the output fields.

Parameters	Description
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 66: PBCSLastLine Return Values

Return Value	Description	Value
n	Number of cities, states, and ZIP Codes that match the search criteria.	A value in the range of 0 up to count. -1 indicates too many matches.

Related Structures

DbxLastLineInDef

DbxLastLineOutDef

DbxStateDef

PBCSLastLineFlat

The PBCSLastLineFlat function builds a list of matching cities, states, and ZIP Codes based on the last line of an address.

Syntax

```
int PBCSLastLineFlat (pDbxLastLineInDef   In,
                      pDbxLastLineOutDef  Out1,
                      pDbxLastLineOutDef  Out2,
                      pDbxLastLineOutDef  Out3,
                      pDbxLastLineOutDef  Out4,
                      pDbxLastLineOutDef  Out5,
                      pDbxLastLineOutDef  Out6,
                      pDbxLastLineOutDef  Out7,
                      pDbxLastLineOutDef  Out8,
                      pDbxLastLineOutDef  Out9,
```

```
                        pDbxLastLineOutDef  Out10,
                        pDbxStateDef State);
```

Parameters

Table 67: PBCSLastLineFlat API Parameters

Parameter	Description
In	Pointer to a data structure of type DbxLastLineInDef. This structure defines the required input fields.
Out1 through Out10	Pointers to a data structure of type DbxLastLineOutDef. This structure defines the output fields.
State	Pointer to a data structure of type DbxStateDef. The state pointer allows you to call each function with different attributes to return a specific size and quantity in the returned list.

Return Values

Table 68: PBCSLastLineFlat Return Values

Return Value	Description	Value
n	Number of cities, states, and ZIP Codes that match the search criteria.	A value in the range of 0 up to count. -1 indicates too many matches.

Related Structures

DbxLastLineInDef

DbxLastLineOutDef

DbxStateDef

Using the Product APIs

You can use the Finalist product APIs to set up and process with Finalist. The product APIs are listed alphabetically and described in detail.

Note: When you pass a structure to any of the Finalist APIs, the structure must contain a valid identifier (ApiId) and a current version.

z/OS users may need to use the IMPORT library that was created as part of the Finalist product installation (see your Finalist Installation Guide). The IMPORT library is your "definition side-deck" (DD SYSDEFSD) that is part of your pre-link process.

PBFNCase

The PBFNCase function allows you to change the case of a single instance of a character string. The database APIs return data in an all upper case format. When used with the database APIs, PBFNCase should process each segment of return data.

C Syntax

```
int   PBFNCase(char* pCaseOption, char * string, int  * pStringLen);
```

COBOL Syntax

```
CALL PBFNCASE USING
                        BY REFERENCE WS-CASE-OPTION,
                        BY REFERENCE WS-STRING-FIELD,
                        BY REFERENCE WS-STRING-LENGTH,
                        RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 69: PBFNCase API Parameters

Parameter	Description
pCaseOption WS-CASE-OPTION	One of the characters described below: **U** Convert returned data to an all upper case format (for example, DATA). **L** Convert returned data to an all lower case format (for example, data). **M** Convert returned data a mixed upper and lower case format (for example, Data). **Other** If a value other than U, L, or M is specified, the default is to convert returned data based on the value specified by cStandardCase in the PBFNSetupDef structure or the value specified for "CASS Standardize Case = " in the pbfn.cfg configuration file.
string WS-STRING-FIELD	Defines the data string to convert.
pStringLen WS-STRING-LENGTH	Length of the data string to convert. COBOL: (PIC S9(08) BINARY

Return Values

Table 70: PBFNCase Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully converted returned data.	1
PBFN_FAIL	Unsuccessfully converted returned data.	0

PBFNDecryptProcDate

The PBFNDecryptProcDate function allows you to pass in the encrypted date assigned to a record during a Process Unassign run and return the full formatted decrypted date. Use this function to retrieve the date and pass the date on to other programs that need dates in a decrypted format.

C Syntax

```
int  PBFNDecryptProcDate(char encryptedDate[4],
                         char decryptedDate[11]);
```

COBOL Syntax

```
CALL PBFNDECR USING
                      BY VALUE PBFN-ADRS-PROCESSDATE,
                      BY VALUE WS-FORMATTED-DATE,
                      RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 71: PBFNDecryptProcDate API Parameters

Parameter	Description
encryptedDate PBFN-ADRS-PROCESSDATE	Four-character cProcessDate field returned in any one of the return address data structures.
decryptedDate WS-FORMATTED-DATE	11-character field that returns the date in an MM/DD/YYYY format.

Return Values

Table 72: PBFNDecryptProcDate Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully processed encrypted dates.	1

Return Value	Description	Value
PBFN_FAIL	Unsuccessfully processed encrypted dates.	0

Related APIs

PBFNProcess

Related Structures

PBFNAddressDataDef

PBFNInfo

The PBFNInfo function returns database, engine, and system information (for example, available memory, operating system version, etc.) for the user's platform if that information is available. Use this call to retrieve information without going through initialization processing.

> **Note:** The PBFNInit API now returns the PBFNInfoDef structure information as part of the SetupDataDef structure. This makes the PBFNInfo API obsolete if you are already performing a PBFNInit call.

C Syntax

```
int PBFNInfo(PBFNInfoDef *PBFNInfo);
```

COBOL Syntax

```
CALL PBFNINFO USING
                       BY REFERENCE PBFN-BINF-INFO,
                       RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 73: PBFNInfo API Parameters

Parameter	Description
PBFNInfoDef PBFN-BINF-INFO	Pointer to a data structure of type PBFNInfoDef. The PBFNInfoDef structure contains database, engine, and system information provided or retrieved by the Finalist engine and the PBFNInit API.

Return Values

Table 74: PBFNInfo Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully returned information.	1
PBFN_FAIL	Unsuccessful return of information due to a system failure. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related APIs

PBFNInit

Related Structures

PBFNExtendedErrorDef

PBFNInfoDef

PBFNSetupDef

PBFNInit

The PBFNInit function initializes Finalist processing. Parameters are used to pass file names, file handling options, and processing options to the Finalist processing engine. The Finalist engine

verifies the parameters and initializes file processing, statistical data, and USPS Form 3553 (CASS Summary Report) reporting.

C Syntax

```
int  PBFNInit(char *initType,
              PBFNSetupDef *pSetup,
              void *Reserved1,
              void *Reserved2,
              void *Reserved3,
              void *Reserved4,
              void *Reserved5,
              void *Reserved6,
              void *Reserved7)
```

COBOL Syntax

```
CALL PBFNINIT USING
                     BY VALUE NULL-POINTER,
                     BY REFERENCE PBFN-GCFG-SETUP,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
                     BY VALUE NULL-POINTER,
            RETURNING FINALIST-RETURN-INT.
```

Note: The PBFNSetupDef structure now contains return information from the PBFNInfoDef structure. This makes the PBFNInfo API obsolete. The PBFNSetupDef structure now also contains information from the PBFNExtendedErrorDef structure. This makes passing in a separate PBFNExtendedErrorDef structure redundant.

Parameters

Table 75: PBFNInit API Parameters

Parameter	Description
initType	Set a null pointer (for normal processing) or use the following value: **PBFN_INIT_DEF** Perform normal processing (DEF). `COBOL: PIC X(003) VALUE 'DEF'`

Parameter	Description
PBFNSetupDef PBFN-GCFG-SETUP	Optional argument. Pointer to a data structure of type PBFNSetupDef. The PBFNSetupDef structure contains file names, paths, etc. for the configuration and database files. If you do not need the information in this structure, pass a pointer set to NULL. Values specified in the setup structure will override the default (and configured) values for the engine. The Finalist engine will attempt to load a configuration file by default. The values in the configuration file will also override the default values for the engine unless the value is already specified in the setup structure. The setup structure contains the following sections. **Files** — Defines the location of the database files. **Process** — Defines how Finalist processes addresses (for example, standardizes case, assigns 13-character USPS abbreviation for the city name). **CASS** — Describes CASS mailing information, Z4Change information, and CASS reporting output file names. **Report** — Contains information for report generation. The calling program will set flags in this structure to indicate the reports to generate and any special processing to be performed. **Logging** — Sets up the message logging process. You decide the level of logging to be performed. **Product** — Defines the optional processing to be performed (for example, Line of Travel (eLOT) Option, Delivery Point Validation (DPV) Option). The product section also defines the software key and the operating system. **Return Error** — Provides return information on any errors you may encounter during initialization. **Return Info** — Provides return information on the environment, postal coding engine, ZIP + 4 database, and City database.
Reserved1 through Reserved6 **PBFNIMSSetupDef** PBFN-ICFG-IMS	Pointers to pass optional structures in addition to the PBFNSetupDef structure. If pointers are not used, pass void pointers set to NULL. The available options are: **IMS Setup Structure** — Pointer to a structure of type PBFNIMSSetupDef. The PBFNIMSSetupDef structure contains pointers to the IMS PCBs. This structure is required if you are running in an IMS environment. This structure should not be passed in any other environment.

Return Values

Table 76: PBFNInit Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully initialized the postal coding engine.	1
PBFN_FAIL	Unsuccessful initialization of the postal coding engine.	0
PBFN_HAVE_WARNING	Databases will expire at the end of the current month. This message will begin to appear 10 days before the expiration date.	4
PBFN_DB_EXPIRED	Database has expired. CASS processing turned off.	5
PBFN_ENGINE_WARNING	Product will expire at end of the current month. The product will not be able to function in USPS CASS-certified mode.	7
PBFN_NOLOT	The enhanced LOT database was unavailable and LOT codes will not be assigned in this run. Finalist issues this message if the LOT File name was invalid and/or Finalist was unable to open the LOT File.	8
PBFN_ENGINE_EXPIRED	CASS engine has expired. CASS processing turned off.	9
PBFN_ERROR	An error occurred during initialization of the postal coding engine. For detailed error information, use the PBFNExtendedErrorDef structure.	-1

Related Structures

PBFNIMSSetupDef

PBFNSetupDef

PBFNProcess

The PBFNProcess function performs postal coding processing on input addresses.

C Syntax

```
int  PBFNProcess(void *Reserved1,
                 void *Reserved2,
                 void *Reserved3,
                 void *Reserved4,
                 void *Reserved5,
                 void *Reserved6,
                 void *Reserved7,
                 void *Reserved8,
                 void *Reserved9)
                 void *Reserved10);
```

COBOL Syntax

```
CALL PBFNPROC USING
                      BY REFERENCE PBFN-ADRS-ADDRESSDATA,
                      BY REFERENCE PBFN-RRTN-RRTN,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      BY VALUE 0,
                      RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 77: PBFNProcess API Parameters

Parameter	Description
Reserved1 through Reserved10	Pointers to pass optional structures. If pointers are not used, pass void pointers set to NULL. The available options are:

Parameter		Description
PBFNAddressDataDef PBFN-ADRS-ADDRESSDATA	**Address Data**	Pointer to a data structure of address data type PBFNAddressDataDef with the ID set to "ADRS". This is used to send address data into the engine for coding. If no return structure is provided, the engine fills this structure with the coded address information.
PBFNAddressDataDef PBFN-RRTN-RRTN	**Return Address Data**	Pointer to a data structure of address data type PBFNAddressDataDef with ID set to "RRTN". The engine fills this structure with the coded address information. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNExtendedErrorDef PBFN-IERR-EXTERROR	**Error Information**	Pointer to a data structure of type PBFNExtendedErrorDef with the ID set to "IERR". If an error occurs while processing the function, the PBFNExtendedErrorDef structure is updated with detailed error information. This is an optional argument. NOTE: The Return Address Data structure includes the Error Information structure information. If you use the Return Address Data structure, you do not need to pass in an Error Information structure.

Return Values

Table 78: PBFNProcess Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully processed address. All address information standardized and assigned.	1
PBFN_FAIL	Unsuccessfully processed address. Errors encountered during processing. For detailed error information, use the PBFNExtendedErrorDef structure.	0

Related API

PBCSCreateSuggestionList

Related Structures

PBFNAddressDataDef

PBFNExtendedErrorDef

PBFNStats

The PBFNStats function collects postal coding processing statistics for each record during batch job processing. Examples of the type of statistics collected include the number of records processed, number of records found and not found, city/state information, address type information, etc. PBFNStats also collects statistics for the USPS Form 3553 (CASS Summary Report), the Finalist Batch Report, and the Address Detail Report. This function can be called twice in a batch environment. The first call (PBFNReset set to TRUE) zeros out statistics at the beginning of a processing run. The second call (PBFNReset set to FALSE) returns the statistics at the end of the processing run. When PBFNReset is set to FALSE and statistics are collected, the USPS Form 3553 (CASS Summary Report) and the Batch Report are generated based on the configuration options.

C Syntax

```
int  PBFN_API PBFNStats(int PBFNReset,
                        void *Reserved1,
                        void *Reserved2,
                        void *Reserved3,
                        void *Reserved4,
                        void *Reserved5,
                        void *Reserved6,
                        void *Reserved7,
                        void *Reserved8,
                        void *Reserved9,
                        void *Reserved10);
```

COBOL Syntax

```
CALL PBFNSTAT USING
                       BY VALUE WS-RESET-FLAG,
                       BY REFERENCE PBFN-GSTS-STATS,
                       BY REFERENCE PBFN-SDPV-DPVSTATS,
                       BY REFERENCE PBFN-TDPV-DPVHDR,
                       BY REFERENCE PBFN-RDAT-RDATA,
                       BY REFERENCE PBFN-NCAS-3553,
                       BY REFERENCE PBFN-BINF-INFO,
                       BY REFERENCE PBFN-PRST-RTNLACSSTATS,
                       BY REFERENCE PBFN-SSLK-RTNSTELNKSTATS,
                       BY REFERENCE PBFN-YDRS-LACS,
```

```
                    BY VALUE NULL-POINTER,
                    RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 79: PBFNStats API Parameters

Parameter	Description

PBFNReset WS-RESET-FLAG	The reset flag is an integer that indicates whether statistics should be accumulated, returned, or reset. The available options are:
	PBFNReset Resets the statistics collecting process.
	`COBOL: PIC S9(08) BINARY VALUE(1)`
	PBFNGetStats Returns processing statistics without resetting the statistics collecting process.
	`COBOL: PIC S9(08) BINARY VALUE(0)`

Reserved1 through Reserved10	Use these pointers to pass optional structures. If pointers are not used, pass void pointers set to NULL. There are currently eight possible values with two more reserved for future use. The available options are:
PBFNStatsDef PBFN-GSTS-STATS	**Return Statistics** Pointer to a data structure of address type PBFNStatsDef. The PBFNStatsDef structure returns the statistical data collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNDPVStatsDef PBFN-SDPV-DPVSTATS	**Return DPV Statistics** Pointer to a data structure of address type PBFNDPVStatsDef. The PBFNDPVStatsDef structure returns the DPV statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNDPVHdrDef PBFN-TDPV-DPVHDR	**Return DPV Header Statistics for DPV False Positive (Seed) Table Processing** Pointer to a data structure of address type PBFNDPVHdrDef. The PBFNDPVHdrDef structure returns the DPV processing statistics required for DPV False Positive (Seed) Table error requirements. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.

Parameter		Description
PBFNRDataDef on page 198 PBFN-RDAT-RDATA	**Return Report Data**	Pointer to a data structure of address type PBFNReportData. The PBFNReportData structure provides information for generating reports. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFN3553Def PBFN-NCAS-3553	**Return 3553 Stats**	Pointer to a data structure of address type PBFN3553Def. The PBFN3553Def structure returns the USPS Form 3553 (CASS Summary Report) statistics in the USPS electronic format. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNExtendedErrorDef PBFN-IERR-EXTERROR	**Error Information**	Pointer to a data structure of type PBFNExtendedErrorDef. If an error occurs while processing the function, the PBFNExtendedErrorDef structure is updated with detailed error information. This is an optional argument.
PBFNRtnLACSStatsDef PBFN-PRST-RTNLACSSTATS	**Return LACS^{Link} Statistics**	Pointer to a data structure of address type LACSStatsDef. The LACSStatsDef structure returns the LACS^{Link} statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNRtnSuiteLinkStatsDef PBFN-SSLK-RTNSTELNKSTATS	**Return Suite^{Link} Statistics**	Pointer to a data structure of address type SuiteLinkStatsDef. The SuiteLinkStatsDef structure returns the Suite^{Link} statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNLACSSeedHdrDef PBFN-YDRS-LACS	**Return LACS^{Link} Header Statistics for LACS^{Link} False Positive (Seed) Table Processing**	Pointer to a data structure of address type PBFNLACSSeedHdrDef. The PBFNLACSSeedHdrDef structure returns the LACS^{Link} processing statistics required for False Positive (Seed) Table error requirements. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.

Return Values

Table 80: PBFNStats Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully retrieved statistics.	1
PBFN_FAIL	Unsuccessful retrieval of statistics.	0

Related Structures

PBFN3553Def

PBFNDPVHdrDef

PBFNDPVStatsDef

PBFNExtendedErrorDef

PBFNLACSSeedHdrDef

PBFNRDataDef

PBFNRtnLACSStatsDef

PBFNRtnSuiteLinkStatsDef

PBFNStatsDef

PBFNTerminate

The PBFNTerminate function performs any necessary housekeeping at the end of a process and returns statistics. PBFNTerminate also produces the processing reports based on the configuration options. The PBFNTerminate function and the PBFNStats function use the same process to collect statistics.

C Syntax

```
int  PBFN_API PBFNTerminate(void *Reserved1,
                            void *Reserved2,
                            void *Reserved3,
                            void *Reserved4,
                            void *Reserved5,
                            void *Reserved6,
```

```
                              void *Reserved7,
                              void *Reserved8,
                              void *Reserved9,
                              void *Reserved10);
```

COBOL Syntax

```
CALL PBFNTERM USING
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY VALUE NULL-POINTER,
                        BY REFERENCE PBFN-IERR-EXTERROR,
                        RETURNING FINALIST-RETURN-INT.
```

Parameters

Table 81: PBFNTerminate API Parameters

Parameter	Description	
Reserved1 through Reserved10	Pointers to pass optional structures. If pointers are not used, pass void pointers set to NULL. There are currently eight possible values with two more reserved for future use. The available options are described next.	
PBFNStatsDef PBFN-GSTS-STATS	**Return Statistics**	Pointer to a data structure of address type PBFNStatsDef. The PBFNStatsDef structure returns the statistical data collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNDPVStatsDef PBFN-SDPV-DPVSTATS	**Return DPV Statistics**	Pointer to a data structure of address type PBFNDPVStatsDef. The PBFNDPVStatsDef structure returns the DPV statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.

Parameter		Description
PBFNDPVHdrDef PBFN-TDPV-DPVHDR	**Return DPV Header Statistics for DPV False Positive (Seed) Table Processing**	Pointer to a data structure of address type PBFNDPVHdrDef. The PBFNDPVHdrDef structure returns the DPV processing statistics required for DPV False Positive (Seed) Table error requirements. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNRDataDef PBFN-RDAT-RDATA	**Return Report Data**	Pointer to a data structure of address type PBFNReportData. The PBFNReportData structure provides information for generating reports. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFN3553Def PBFN-NCAS-3553	**Return 3553 Stats**	Pointer to a data structure of address type PBFN3553Def. The PBFN3553Def structure returns the USPS Form 3553 (CASS Summary Report) statistics in the USPS electronic format. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNExtendedErrorDef PBFN-IERR-EXTERROR	**Error Information**	Pointer to a data structure of type PBFNExtendedErrorDef. If an error occurs while processing the function, the PBFNExtendedErrorDef structure is updated with detailed error information. This is an optional argument.
PBFNRtnLACSStatsDef PBFN-PRST-RTNLACSSTATS	**Return LACS^{Link} Statistics**	Pointer to a data structure of address type LACSStatsDef. The LACSStatsDef structure returns the LACS^{Link} statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.
PBFNRtnSuiteLinkStatsDef PBFN-SSLK-RTNSTELNKSTATS	**Return Suite^{Link} Statistics**	Pointer to a data structure of address type SuiteLinkStatsDef. The SuiteLinkStatsDef structure returns the Suite^{Link} statistics collected during the postal coding process. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.

Parameter	Description	
PBFNLACSSeedHdrDef PBFN-YDRS-LACS	**Return LACS^{Link} Header Statistics for LACS^{Link} False Positive (Seed) Table Processing**	Pointer to a data structure of address type PBFNLACSSeedHdrDef. The PBFNLACSSeedHdrDef structure returns the LACS^{Link} processing statistics required for False Positive (Seed) Table error requirements. If you do not need the information in this structure, pass a pointer set to NULL. This is an optional argument.

Return Values

Table 82: PBFNTerminate Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully ended processing.	0

Related APIs

PBFNInit

PBFNStats

Related Structures

PBFN3553Def

PBFNDPVStatsDef

PBFNDPVHdrDef

PBFNExtendedErrorDef

PBFNLACSSeedHdrDef

PBFNRDataDef

PBFNRtnLACSStatsDef

PBFNRtnSuiteLinkStatsDef

PBFNStatsDef

PBFNTransact

The PBFNTransact function is used for on-line processing. Instead of requiring a PBFNInit, PBFNProcess, and PBFNTerminate call for each address processed in an on-line environment, PBFNTransact combines all three calls in a single API.

The PBFNTransact API was designed for single address coding. PBFNTransact does not include the USPS Form 3553 (CASS Summary Report) or other reporting options. If specified, Finalist ignores the CASS and reporting options.

C Syntax

```
int  PBFN_API PBFNTransact(PBFNSetupDef *pSetup,
                           PBFNAddressDataDef *pAddressData,
                           PBFNAddressDataDef *pRtnAddressData);
```

Cobol Syntax

```
CALL PBFNTRAN USING
                          BY REFERENCE PBFN-GCFG-SETUP,
                          BY REFERENCE PBFN-ADRS-ADDRESSDATA,
                          BY REFERENCE PBFN-RRTN-RRTN,
                          RETURNING WS-PBFN-RETURN-CODE.
```

Parameters

Table 83: PBFNTransact API Parameters

Parameter	Description
pSetup PBFN-GCFG-SETUP	Pointer to a data structure of type PBFNSetupDef. The PBFNSetupDef structure contains file names, paths, etc. for the configuration and database files. If you do not need the information in this structure, pass a pointer set to NULL. Values specified in the setup structure override the default (and configured) values for the engine. The Finalist engine will attempt to load a configuration file by default. The values in the configuration file also override the default values for the engine unless the value is already specified in the setup structure.
PBFNAddressDataDef PBFN-ADRS-ADDRESSDATA	Pointer to a data structure of address data type PBFNAddressDataDef with the ID set to "ADRS". This is used to send address data to the engine for coding. If no return structure is provided, the engine fills this structure with the coded address information.

Parameter	Description
PBFNAddressDataDef PBFN-RRTN-RRTN	Optional argument. Pointer to a data structure of address data type PBFNAddressDataDef with ID set to "RRTN". The engine fills this structure with the coded address information. If you do not need the information in this structure, pass a pointer set to NULL. If the return PBFNAddressData parameter is not passed in, Finalist updates the input PBFNAddressData structure (as is typical with PBFNProcess).

Return Values

Table 84: PBFNTransact Return Values

Return Value	Description	Value
PBFN_SUCCESS	Successfully processed single address.	1
PBFN_FAIL	Unsuccessful process of single address.	0

If PBFNTransact cannot be initialized, the return code is based on the PBFNInit API. Otherwise, the PBFNTransact return code is based on the PBFNProcess API. In this second case, the PBFNInit return code is still available in a new field in SetupDef called iInitRtnCode (C) or PBFN-GCFG-IINITRTNCODE (COBOL). These fields would typically include warning messages that the database is about to expire.

For a normal PBFNInit API call, the iInitRtnCode (PBFN-GCFG-IINITRTNCODE) field can be ignored as the field simply represents the return code for that call.

The ExtendedError portions of both the SetupDef and AddressDataDef structures are populated if an error occurs. Either structure can be used to report any kind of failure on the API call.

Related APIs

PBFNInit

PBFNProcess

PBFNTerminate

Related Structures

PBFNAddressDataDef

PBFNSetupDef

2 - Using the Structures and Constants

In this section

Using the Finalist Structures

This section describes the Finalist structures and constants. The structure descriptions include sample COBOL copybooks. C headers and COBOL copybooks are provided with all distributions. BAL copybooks are only provided with mainframe distributions.

> **Note:** For COBOL, we recommend you place each Finalist COBOL copybook in its own 01 structure. If this is not possible, each Finalist COBOL copybook should be aligned on a fullword boundary by placing the line below before each COPY statement. 05 FILLER PIC S9(08) BINARY.
>
> Finalist should be called using the current PBFN_CURRENT_VERSION for all structures. If your application calls Finalist using the previous version of PBFN_CURRENT_VERSION, Finalist generates a warning message indicating this version is deprecated: PBFN - Version is deprecated {NNNNxxxx } Finalist only flags the first structure encountered with this message.
>
> For field lengths in the structures, the length represents the amount of data to be returned. For those using null terminated strings, the length MUST include the null terminating byte.

Table 85: Finalist Structures

Use this C/COBOL Structure	To Perform this Task
DbxGetAliasInDef PBCS-CGAI-GETALIASIN	Define input fields for suggestion processing based on alias information.
DbxGetAliasOutDef PBCS-CGAO-GETALIASOUT	Define output fields for suggestion processing based on alias information.
DbxGetFirmInDef PBCS-CSFI-GETFIRMIN	Define input fields for suggestion processing based on firm information.
DbxGetFirmOutDef PBCS-CSFO-GETFIRMOUT	Define output fields for suggestion processing based on firm information.
DbxGetRangeInDef PBCS-CGRI-GETRANGEIN	Define input fields for suggestion processing based on primary range information.

Use this C/COBOL Structure	To Perform this Task
DbxGetRangeOutDef PBCS-CGRO-GETRANGEOUT	Define output fields for suggestion processing based on primary range information.
DbxGetSecRangeOutDef PBCS-CSRO-GETSECRANGEOUT	Define output fields for suggestion processing based on secondary range information.
DbxGetStreetInDef PBCS-CGSI-GETSTREETIN	Define input fields for suggestion processing based on street information.
DbxGetStreetOutDef PBCS-CGSO-GETSTREETOUT	Define output fields for suggestion processing based on street information.
DbxGetStreetQInDef PBCS-CSQI-STREETQ	Define input fields for suggestion processing based on street information.
DbxGetStreetQOutDef PBCS-CSQO-STREETQ	Define output fields for suggestion processing based on street information.
DbxGetSuggestionInDef PBCS-CSGI-SUGGESTION	Define input fields for suggestion processing based on street information.
DbxGetSuggestionOutDef PBCS-CSGO-SUGGESTION	Define output fields for suggestion processing based on street information.
DbxGetZipCityInDef PBCS-CGCI-GETZIPCITYIN	Define input fields for suggestion processing based on city and state information.
DbxGetZipInDef PBCS-CGZI-GETZIPIN	Define input fields for suggestion processing based on ZIP Code information.

Use this C/COBOL Structure	To Perform this Task
DbxGetZipOutDef PBCS-CGZO-GETZIPOUT	Define output fields for suggestion processing based on ZIP Code information.
DbxLastLineInDef PBCS-CLLI-GETLASTLINE	Define input fields for suggestion processing based on last line information.
DbxLastLineOutDef PBCS-CLLO-GETLASTLINE	Define output fields for suggestion processing based on last line information.
DbxStateDef PBCS-CSSI-STATE	Call each function with different attributes to return a specific size and quantity for the returned list.
PBFN3553Def PBFN-NCAS-3553	Find CASS Stage 2 header information.
PBFNAddressDataDef PBFN-ADRS-ADDRESSDATA	Pass parsed and unparsed address information to and from the postal coding functions. Retrieve postal coding address information. Return DPV False Positive (Seed) Table detail record. Create label lines. Return LACS^{Link} False Positive violation detail information for USPS. Process with parsed address elements. Process with parsed address elements using string pointers. Pass address information using character arrays. Pass address information using string pointers. Return suggested firm names. Return original data.
PBFNDPVHdrDef PBFN-TDPV-DPVHDR	Return DPV header record.

Use this C/COBOL Structure	To Perform this Task
PBFNDPVStatsDef PBFN-SDPV-DPVSTATS	Find DPV processing statistics.
PBFNExtendedErrorDef PBFN-IERR-EXTERROR	Find Information on Errors.
PBFNIMSSetupDef PBFN-ICFG-IMS	Provides a place to pass in IMS PCB information.
PBFNInfoDef PBFN-BINF-INFO	Pass environment, engine and database Information.
PBFNLACSSeedHdrDef PBFN-YDRS-LACS	Return LACS^{Link} False Positive violation header information.
PBFNRDataDef PBFN-RDAT-RDATA	Set up the Finalist Batch Report.
PBFNRtnLACSStatsDef PBFN-PRST-RTNLACSSTATS	Return LACS^{Link} processing statistics.
PBFNRtnSuiteLinkStatsDef PBFN-SSLK-RTNSTELNKSTATS	Returns Suite^{Link} processing statistics.
PBFNSetupDef PBFN-GCFG-SETUP	Define your system configuration.
PBFNStatsDef PBFN-GSTS-STATS	Retrieve processing statistics.

Use this C/COBOL Structure	To Perform this Task
PBFNSuggestionDef PBFN-DSUG-SUGGESTION	Return address suggestions.
USPSDetailDef PBFN-PSDH-USPS	Return USPS-required DPV False Positive (Seed) Table violation detail.
USPSDPVHdrDef PBFN-PSDD-USPS	Return DPV False Positive (Seed) Table violation header data.
USPSPBLACSDetDef PBFN-PSLD-USPS	Return LACS^{Link} False Positive violation detail data required by the USPS.
USPSPBLACSHdrDef PBFN-PSLH-USPS	Return LACS^{Link} False Positive violation header data required by USPS for LACS^{Link} processing in USPS-required format.

Using the Database Structures

DbxGetAliasInDef

DbxGetAliasInDef defines the required input fields for suggestion processing based on alias information. The following Data Access APIs use DbxGetAliasInDef.

- PBCSGetStreetAliasCity (PBCSGetStreetAliasCityFlat)
- PBCSGetStreetAliasZip (PBCSGetStreetAliasZipFlat)

Syntax

```
typedef struct DbxGetAliasInDefinition
{
    char    cVersion[VERSION_LEN];
```

```
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lZip;
    long    lStreetPrime;
    long    lStreetIndex;
    char    cState[10];
    char    cFiller2[2];
    long    lStateLen;
    char    cCity[50];
    char    cFiller3[2];
    long    lCityLen;
} DbxGetAliasInDef, * pDbxGetAliasInDef;
```

Field Descriptions

Table 86: DbxGetAliasInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lZip	ZIP Code.
lStreetPrime	Primary street.
lStreetIndex	Index entry in the list of streets for the ZIP Code used as the search criteria.
cState	State.
lStateLen	State length.
cCity	City.
lCityLen	City length.

COBOL Copybook

```
       ****************************************************************

       **                                                            **
```

```
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                          **
      ** Name: PBCSCGAI.CPY                                      **
      **                                                          **
      ** Use: COBOL Copybook for DbxGetAliasInDefinition         **
      **                                                          **
      ** Distributed Source Member for the Finalist product from  **
      ** Precisely                                               **
      **                                                          **
      ************************************************************

       05  PBCS-CGAI-GETALIASIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGAI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CGAI-ZIP                   PIC S9(08) BINARY.
         10  PBCS-CGAI-STREETPRIME           PIC S9(08) BINARY.
         10  PBCS-CGAI-STREETINDEX           PIC S9(08) BINARY.
         10  PBCS-CGAI-STATE                 PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CGAI-STATE-LEN             PIC S9(08) BINARY.
         10  PBCS-CGAI-CITY                  PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CGAI-CITY-LEN              PIC S9(08) BINARY.
```

Related APIs

PBCSGetStreetAliasCity

PBCSGetStreetAliasCityFlat

PBCSGetStreetAliasZip

PBCSGetStreetAliasZipFlat

DbxGetAliasOutDef

DbxGetAliasOutDef defines the output fields for suggestion processing based on alias information. The following Data Access APIs use DbxGetAliasOutDef.

- PBCSGetStreetAliasCity (PBCSGetStreetAliasCityFlat)

- PBCSGetStreetAliasZip (PBCSGetStreetAliasZipFlat)

Syntax

```
typedef struct DbxGetAliasOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    rangeLow[11];
    long    rangeLowLen;
    char    rangeHi[11];
    char    cFiller1;
    long    rangeHiLen;
    char    preDir[4];
    long    preDirLen;
    char    posDir[4];
    long    posDirLen;
    char    suffix[5];
    char    cFiller2[3];
    long    suffixLen;
    char    carrRte[5];
    char    cFiller3[3];
    long    carrRteLen;
    char    zip4Low[5];
    char    cFiller4[3];
    long    zip4LowLen;
    char    zip4Hi[5];
    char    cFiller5[3];
    long    zip4HiLen;
    char    alternate[50];
    char    cFiller6[2];
    long    alternateLen;
    char    urbname[29];
    char    cFiller7;
    long    urbnameLen;
    char    eob;
    char    zipcode[6];
    char    cFiller8;
    long    zipLen;
    int     lSecRangeCnt;
    int     lFirmCnt;
    char    alias;
    char    aliasNameOut[50];
    char    cFiller9;
    long    AliasLen;
} DbxGetAliasOutDef, * pDbxGetAliasOutDef;
```

Field Descriptions

Table 87: DbxGetAliasOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
rangeLow	Low range.
rangeLowLen	Low range length.
rangeHi	High range.
rangeHiLen	High range length.
preDir	Predirectional.
preDirLen	Predirectional length.
posDir	Postdirectional.
posDirLen	Postdirectional length.
suffix	Suffix.
suffixLen	Suffix length.
carrRte	Carrier route code.
carrRteLen	Carrier route code length.
zip4Low	Primary range low ZIP + 4 Code.
zip4LowLen	Primary range low ZIP + 4 Code length.
zip4Hi	Primary range high ZIP + 4 Code.
zip4HiLen	Primary range high ZIP + 4 Code length.

Field	Description
alternate	Alternate name.
alternateLen	Alternate name length.
urbname	Street urbanization name.
urbnameLen	Street urbanization name length.
eob	Type of range. **E** Even only range **O** Odd only range **B** Both even and odd ranges **C** Consolidated. Even and odd ranges consolidated as one using the lowest ZIP Code
zipcode	ZIP Code.
zipLen	ZIP Code length.
lSecRangeCnt	Secondary range count.
lFirmCnt	Number of firms associated with the primary range.
alias	"Y" indicates an alias.
aliasNameOut	Alias street name.
AliasLen	Alias street name length.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBCSCGAO.CPY                                         **
```

```
      **                                                              **
      ** Use: COBOL Copybook for DbxGetAliasOutDefinition             **
      **                                                              **
      ** Distributed Source Member for the Finalist product from      **
      ** Precisely                                                    **
      **                                                              **
      ******************************************************************

       05  PBCS-CGAO-GETALIASOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGAO'.
         10  PBCS-CGAO-RANGELOW              PIC X(011).
         10  PBCS-CGAO-RANGELOW-LEN          PIC S9(08) BINARY.
         10  PBCS-CGAO-RANGEHI               PIC X(011).
         10  FILLER                          PIC X(001).
         10  PBCS-CGAO-RANGEHI-LEN           PIC S9(08) BINARY.
         10  PBCS-CGAO-PREDIR                PIC X(004).
         10  PBCS-CGAO-PREDIR-LEN            PIC S9(08) BINARY.
         10  PBCS-CGAO-POSDIR                PIC X(004).
         10  PBCS-CGAO-POSDIR-LEN            PIC S9(08) BINARY.
         10  PBCS-CGAO-SUFFIX                PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CGAO-SUFFIX-LEN            PIC S9(08) BINARY.
         10  PBCS-CGAO-CARRRTE               PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CGAO-CARRRTE-LEN           PIC S9(08) BINARY.
         10  PBCS-CGAO-ZIP4LOW               PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CGAO-ZIP4LOW-LEN           PIC S9(08) BINARY.
         10  PBCS-CGAO-ZIP4HI                PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CGAO-ZIP4HI-LEN            PIC S9(08) BINARY.
         10  PBCS-CGAO-ALTERNATE             PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CGAO-ALTERNATE-LEN         PIC S9(08) BINARY.
         10  PBCS-CGAO-URBNAME               PIC X(029).
         10  FILLER                          PIC X(003).
         10  PBCS-CGAO-URBNAME-LEN           PIC S9(08) BINARY.
         10  PBCS-CGAO-EOB                   PIC X(001).
           88  PBCS-CGAO-EVEN                           VALUE 'E'.
           88  PBCS-CGAO-ODD                            VALUE 'O'.
           88  PBCS-CGAO-BOTH                           VALUE 'B'.
           88  PBCS-CGAO-CONSOLIDATED                   VALUE 'C'.
         10  PBCS-CGAO-ZIPCODE               PIC X(006).
         10  FILLER                          PIC X(001).
         10  PBCS-CGAO-ZIP-LEN               PIC S9(08) BINARY.
         10  PBCS-CGAO-SECRANGECNT           PIC S9(08) BINARY.
         10  PBCS-CGAO-FIRMCNT               PIC S9(08) BINARY.
```

```
          10  PBCS-CGAO-ALIAS                  PIC X(001).
          10  PBCS-CGAO-ALIASNAME-OUT          PIC X(050).
          10  FILLER                           PIC X(001).
          10  PBCS-CGAO-ALIAS-LEN              PIC S9(08) BINARY.
```

Related APIs

PBCSGetStreetAliasCity

PBCSGetStreetAliasCityFlat

PBCSGetStreetAliasZip

PBCSGetStreetAliasZipFlat

DbxGetFirmInDef

DbxGetFirmInDef defines the required input fields for suggestion processing based on firm information. The following Data Access APIs use DbxGetFirmInDef.

- PBCSGetFirmListCity (PBCSGetFirmListCityFlat)
- PBCSGetFirmListZip (PBCSGetFirmListZipFlat)

Syntax

```
typedef struct DbxGetFirmInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lZip;
    long    lStreetmode;
    long    lStreetIndex;
    long    lCount;
    long    lNtPrime;
    long    lNtSec;
    char    cState[10];
    char    cFiller2[2];
    long    lStateLen;
    char    cCity[50];
    char    cFiller3[2];
    long    lCityLen;
} DbxGetFirmInDef, * pDbxGetFirmInDef;
```

Field Descriptions

Table 88: DbxGetFirmInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lZip	ZIP Code.
lStreetmode	Builds a list of streets.
lStreetIndex	Index entry in the list of streets for the ZIP Code used as the search criteria.
lCount	Maximum number of results to return.
lNtPrime	Index entry in the list of primary ranges for the street.
lNtSec	Index entry in the list of secondary ranges for a primary range.
cState	State.
lStateLen	State length.
cCity	City.
lCityLen	City length.

COBOL Copybook

```
      ******************************************************************
      **                                                              **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                              **
      ** Name: PBCSCSFI.CPY                                           **
      **                                                              **
```

```
      ** Use: COBOL Copybook for DbxGetFirmInDefinition              **
      **                                                             **
      ** Distributed Source Member for the Finalist product from     **
      ** Precisely                                                    **
      **                                                             **
      *****************************************************************

       05  PBCS-CSFI-GETFIRMIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSFI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSFI-ZIP                   PIC S9(08) BINARY.
         10  PBCS-CSFI-STREETMODE            PIC S9(08) BINARY.
         10  PBCS-CSFI-STREETINDEX           PIC S9(08) BINARY.
         10  PBCS-CSFI-COUNT                 PIC S9(08) BINARY.
         10  PBCS-CSFI-NTPRIME               PIC S9(08) BINARY.
         10  PBCS-CSFI-NTSEC                 PIC S9(08) BINARY.
         10  PBCS-CSFI-STATE                 PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSFI-STATE-LEN             PIC S9(08) BINARY.
         10  PBCS-CSFI-CITY                  PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CSFI-CITY-LEN              PIC S9(08) BINARY.
         10  FILLER                          PIC X(004).
```

Related APIs

PBCSGetFirmListCity

PBCSGetFirmListCityFlat

PBCSGetFirmListZip

PBCSGetFirmListZipFlat

DbxGetFirmOutDef

DbxGetFirmOutDef defines the output fields for suggestion processing based on firm information. The following Data Access APIs use DbxGetFirmOutDef.

- PBCSGetFirmListCity (PBCSGetFirmListCityFlat)
- PBCSGetFirmListZip (PBCSGetFirmListZipFlat)

Syntax

```
typedef struct pDbxGetFirmOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    firmName[41];
    char    cFiller1[2];
    long    firmLen;
    char    zip4Low[5];
    char    cFiller2[3];
    long    zip4LowLen;
    char    zip4Hi[5];
    char    cFiller3[3];
    long    zip4HiLen;
} DbxGetFirmOutDef, * pDbxGetFirmOutDef;
```

Field Descriptions

Table 89: DbxGetFirmOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
firmName[41]	Firm name.
firmLen	Firm name length.
zip4Low	Primary range low ZIP + 4 Code.
zip4LowLen	Primary range low ZIP + 4 Code length.
zip4Hi	Primary range high ZIP + 4 Code.
zip4HiLen	Primary range high ZIP + 4 Code length.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **

      ** Name: PBCSCSFO.CPY                                         **

      **                                                            **

      ** Use: COBOL Copybook for pDbxGetFirmOutDefinition           **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBCS-CSFO-GETFIRMOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSFO'.
         10  PBCS-CSFO-FIRM-NAME             PIC X(041).
         10  FILLER                          PIC X(002).
         10  PBCS-CSFO-FIRM-LEN              PIC S9(08) BINARY.
         10  PBCS-CSFO-ZIP4LOW               PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSFO-ZIP4LOW-LEN           PIC S9(08) BINARY.
         10  PBCS-CSFO-ZIP4HI                PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSFO-ZIP4HI-LEN            PIC S9(08) BINARY.
```

Related APIs

- **PBCSGetFirmListCity**
- **PBCSGetFirmListCityFlat**
- **PBCSGetFirmListZip**
- **PBCSGetFirmListZipFlat**

DbxGetRangeInDef

DbxGetRangeInDef defines the required input fields for suggestion processing based on primary range information. The following Data Access APIs use DbxGetRangeInDef.

- PBCSGetPrimaryListCity (PBCSGetPrimaryListCityFlat)
- PBCSGetPrimaryListZip (PBCSGetPrimaryListZipFlat)
- PBCSGetSecondaryListCity (PBCSGetSecondaryListCityFlat)
- PBCSGetSecondaryListZip (PBCSGetSecondaryListZipFlat)

Syntax

```
typedef struct DbxGetRangeInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lZip;
    long    lStreetmode;
    long    lStreetIndex;
    long    lCount;
    long    lNtPrime;
    char    cState[10];
    char    cFiller2[2];
    long    lStateLen;
    char    cCity[50];
    char    cFiller3[2];
    long    lCityLen;
} DbxGetRangeInDef, * pDbxGetRangeInDef;
```

Field Descriptions

Table 90: DbxGetRangeInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lZip	ZIP Code.
lStreetmode	Builds list of streets.

Field	Description
lStreetIndex	Index entry in the list of streets for the ZIP Code used as the search criteria.
lCount	Maximum number of results to be returned.
lNtPrime	Index entry in the list of primary ranges for the street.
cState	State.
lStateLen	Length of state.
cCity	City.
lCityLen	Length of city.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBCSCGRI.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxGetRangeInDefinition            **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************

       05  PBCS-CGRI-GETRANGEIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGRI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CGRI-ZIP                   PIC S9(08) BINARY.
         10  PBCS-CGRI-STREETMODE            PIC S9(08) BINARY.
```

```
          10  PBCS-CGRI-STREETINDEX           PIC S9(08) BINARY.
          10  PBCS-CGRI-COUNT                 PIC S9(08) BINARY.
          10  PBCS-CGRI-NTPRIME               PIC S9(08) BINARY.
          10  PBCS-CGRI-STATE                 PIC X(010).
          10  FILLER                          PIC X(002).
          10  PBCS-CGRI-STATE-LEN             PIC S9(08) BINARY.
          10  PBCS-CGRI-CITY                  PIC X(050).
          10  FILLER                          PIC X(002).
          10  PBCS-CGRI-CITY-LEN              PIC S9(08) BINARY.
```

Related APIs

PBCSGetPrimaryListCity

PBCSGetPrimaryListCityFlat

PBCSGetFirmListZip

PBCSGetPrimaryListZipFlat

PBCSGetSecondaryListCity

PBCSGetSecondaryListCityFlat

PBCSGetSecondaryListZip

PBCSGetSecondaryListZipFlat

DbxGetRangeOutDef

DbxGetRangeOutDef defines the output fields for suggestion processing based on primary range information. The following Data Access APIs use DbxGetRangeOutDef.

- PBCSGetPrimaryListCity (PBCSGetPrimaryListCityFlat)
- PBCSGetPrimaryListZip (PBCSGetPrimaryListZipFlat)

Syntax

```
typedef struct DbxGetRangeOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    rangeLow[11];
    long    rangeLowLen;
    char    rangeHi[11];
    char    cFiller1;
    long    rangeHiLen;
    char    preDir[4];
    long    preDirLen;
```

```
    char    posDir[4];
    long    posDirLen;
    char    suffix[5];
    char    cFiller2[3];
    long    suffixLen;
    char    carrRte[5];
    char    cFiller3[3];
    long    carrRteLen;
    char    zip4Low[5];
    char    cFiller4[3];
    long    zip4LowLen;
    char    zip4Hi[5];
    char    cFiller5[3];
    long    zip4HiLen;
    char    alternate[50];
    char    cFiller6[2];
    long    alternateLen;
    char    urbname[29];
    char    cFiller7[3];
    long    urbnameLen;
    char    eob;
    char    zipcode[6];
    char    cFiller8;
    long    zipLen;
    int     lSecRangeCnt;
    int     lFirmCnt;
    char    alias;
} DbxGetRangeOutDef, * pDbxGetRangeOutDef;
```

Field Descriptions

Table 91: DbxGetRangeOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
rangeLow	Low range.
rangeLowLen	Low range length.
rangeHi	High range.
rangeHiLen	High range length.

Field	Description
preDir	Pre-directional.
preDirLen	Pre-directional length.
posDir	Post-directional.
posDirLen	Post-directional length.
suffix	Suffix.
suffixLen	Suffix length.
carrRte[5]	Carrier route code.
carrRteLen	Carrier route code length.
zip4Low	Primary range low ZIP + 4 Code.
zip4LowLen	Primary range low ZIP + 4 Code.
zip4Hi	Primary range high ZIP + 4 Code.
zip4HiLen	Primary range high ZIP + 4 Code length.
alternate	Street alternate name.
alternateLen	Street alternate name length.
urbname	Street urbanization name.
urbnameLen	Street urbanization name length.
eob	Type of range. **E** Even only range **O** Odd only range **B** Both even and odd ranges **C** Consolidated. Even and odd ranges consolidated as one using the lowest ZIP Code

Field	Description
zipcode	ZIP Code.
zipLen	ZIP Code length.
lSecRangeCnt	Secondary range count.
lFirmCnt	Number of firms associated with the primary range.
alias	Alias name.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **
      ** Name: PBCSCGRO.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxGetRangeOutDefinition           **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************

       05  PBCS-CGRO-GETRANGEOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGRO'.
         10  PBCS-CGRO-RANGELOW              PIC X(011).
         10  PBCS-CGRO-RANGELOW-LEN          PIC S9(08) BINARY.
         10  PBCS-CGRO-RANGEHI               PIC X(011).
         10  FILLER                          PIC X(001).
         10  PBCS-CGRO-RANGEHI-LEN           PIC S9(08) BINARY.
         10  PBCS-CGRO-PREDIR                PIC X(004).
         10  PBCS-CGRO-PREDIR-LEN            PIC S9(08) BINARY.
         10  PBCS-CGRO-POSDIR                PIC X(004).
```

```
          10  PBCS-CGRO-POSDIR-LEN              PIC S9(08) BINARY.
          10  PBCS-CGRO-SUFFIX                  PIC X(005).
          10  FILLER                            PIC X(003).
          10  PBCS-CGRO-SUFFIX-LEN              PIC S9(08) BINARY.
          10  PBCS-CGRO-CARRRTE                 PIC X(005).
          10  FILLER                            PIC X(003).
          10  PBCS-CGRO-CARRRTE-LEN             PIC S9(08) BINARY.
          10  PBCS-CGRO-ZIP4LOW                 PIC X(005).
          10  FILLER                            PIC X(003).
          10  PBCS-CGRO-ZIP4LOW-LEN             PIC S9(08) BINARY.
          10  PBCS-CGRO-ZIP4HI                  PIC X(005).
          10  FILLER                            PIC X(003).
          10  PBCS-CGRO-ZIP4HI-LEN              PIC S9(08) BINARY.
          10  PBCS-CGRO-ALTERNATE               PIC X(050).
          10  FILLER                            PIC X(002).
          10  PBCS-CGRO-ALTERNATE-LEN           PIC S9(08) BINARY.
          10  PBCS-CGRO-URBNAME                 PIC X(029).
          10  FILLER                            PIC X(003).
          10  PBCS-CGRO-URBNAME-LEN             PIC S9(08) BINARY.
          10  PBCS-CGRO-EOB                     PIC X(001).
            88  PBCS-CGRO-EVEN                             VALUE 'E'.
            88  PBCS-CGRO-ODD                              VALUE 'O'.
            88  PBCS-CGRO-BOTH                             VALUE 'B'.
            88  PBCS-CGRO-CONSOLIDATED                     VALUE 'C'.
          10  PBCS-CGRO-ZIPCODE                 PIC X(006).
          10  FILLER                            PIC X(001).
          10  PBCS-CGRO-ZIP-LEN                 PIC S9(08) BINARY.
          10  PBCS-CGRO-SECRANGECNT             PIC S9(08) BINARY.
          10  PBCS-CGRO-FIRMCNT                 PIC S9(08) BINARY.
          10  PBCS-CGRO-ALIAS                   PIC X(001).
          10  FILLER                            PIC X(007).
```

Related APIs

PBCSGetPrimaryListCity

PBCSGetSecondaryListCityFlat

PBCSGetPrimaryListZip

PBCSGetPrimaryListZipFlat

DbxGetSecRangeOutDef

DbxGetSecRangeOutDef defines the output fields for suggestion processing based on secondary range information. The following Data Access APIs use DbxGetSecRangeOutDef.

- PBCSGetSecondaryListCity (PBCSGetSecondaryListCityFlat)
- PBCSGetSecondaryListZip (PBCSGetSecondaryListZipFlat)

Syntax

```
typedef struct DbxGetSecRangeOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    rangeLow[11];
    long    rangeLowLen;
    char    rangeHi[11];
    char    cFiller1;
    long    rangeHiLen;
    char    unit[5];
    char    cFiller2[3];
    long    unitLen;
    char    carrRte[5];
    char    cFiller3[3];
    long    carrRteLen;
    char    zip4Low[5];
    char    cFiller4[3];
    long    zip4LowLen;
    char    zip4Hi[5];
    char    cFiller5[3];
    long    zip4HiLen;
    char    eob;
    char    cFiller6[3];
    int     lFirmCnt;
} DbxGetSecRangeOutDef, * pDbxGetSecRangeOutDef;
```

Field Descriptions

Table 92: DbxGetSecRangeOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
rangeLow	Low range.
rangeLowLen	Low range length.
rangeHi	High range.
rangeHiLen	High range length.

Field	Description
unit	Secondary range unit designator.
unitLen	Secondary range unit designator length.
carrRte	Carrier route.
carrRteLen	Carrier route length.
zip4Low	Low ZIP + 4 Code.
zip4LowLen	Low ZIP + 4 Code length.
zip4Hi	High ZIP + 4 Code.
zip4HiLen	High ZIP + 4 Code length.
eob	Type of range. **E** Even only range **O** Odd only range **B** Both even and odd ranges **C** Consolidated. Even and odd ranges consolidated as one using lowest ZIP Code
lFirmCnt	Number of firms associated with the secondary range.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBCSCSRO.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxGetSecRangeOutDefinition        **
      **                                                            **
```

```
      ** Distributed Source Member for the Finalist product from     **

      ** Precisely                                                    **
      **                                                              **

      ******************************************************************

       05  PBCS-CSRO-GETSECRANGEOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSRO'.
         10  PBCS-CSRO-RANGELOW              PIC X(011).
         10  PBCS-CSRO-RANGELOW-LEN          PIC S9(08) BINARY.
         10  PBCS-CSRO-RANGEHI               PIC X(011).
         10  FILLER                          PIC X(001).
         10  PBCS-CSRO-RANGEHI-LEN           PIC S9(08) BINARY.
         10  PBCS-CSRO-UNIT                  PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSRO-UNIT-LEN              PIC S9(08) BINARY.
         10  PBCS-CSRO-CARRRTE               PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSRO-CARRRTE-LEN           PIC S9(08) BINARY.
         10  PBCS-CSRO-ZIP4LOW               PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSRO-ZIP4LOW-LEN           PIC S9(08) BINARY.
         10  PBCS-CSRO-ZIP4HI                PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSRO-ZIP4HI-LEN            PIC S9(08) BINARY.
         10  PBCS-CSRO-EOB                   PIC X(001).
           88  PBCS-CSRO-EVEN                           VALUE 'E'.
           88  PBCS-CSRO-ODD                            VALUE 'O'.
           88  PBCS-CSRO-BOTH                           VALUE 'B'.
           88  PBCS-CSRO-CONSOLIDATED                   VALUE 'C'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSRO-FIRMCNT               PIC S9(08) BINARY.
```

Related APIs

PBCSGetSecondaryListCity

PBCSGetSecondaryListCityFlat

PBCSGetSecondaryListZip

PBCSGetSecondaryListZipFlat

DbxGetStreetInDef

DbxGetStreetInDef defines the required input fields for suggestion processing based on street information. The following Data Access APIs use DbxGetStreetInDef.

- PBCSGetStreetListCity (PBCSGetStreetListCityFlat)
- PBCSGetStreetListZip (PBCSGetStreetListZipFlat)

Syntax

```
typedef struct DbxGetStreetInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lZip;
    long    lStreetmode;
    long    lCount;
    char    cCity[50];
    char    cFiller2[2];
    long    lCityLen;
    char    cState[10];
    char    cFiller3[2];
    long    lStateLen;
} DbxGetStreetInDef, * pDbxGetStreetInDef;
```

Field Descriptions

Table 93: DbxGetStreetInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lZip	ZIP Code.
lStreetmode	Builds list of streets.
lCount	The maximum number of results to be returned.
cCity	City name.

Field	Description
lCityLen	City name length.
cState	State.
lStateLen	State length.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBCSCGSI.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for DbxGetStreetInDefinition          **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBCS-CGSI-GETSTREETIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGSI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CGSI-ZIP                   PIC S9(08) BINARY.
         10  PBCS-CGSI-STREETMODE            PIC S9(08) BINARY.
         10  PBCS-CGSI-COUNT                 PIC S9(08) BINARY.
         10  PBCS-CGSI-CITY                  PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CGSI-CITY-LEN              PIC S9(08) BINARY.
         10  PBCS-CGSI-STATE                 PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CGSI-STATE-LEN             PIC S9(08) BINARY.
```

Related APIs

PBCSGetStreetListCity

PBCSGetStreetListCityFlat

PBCSGetStreetListZip

PBCSGetStreetListZipFlat

DbxGetStreetOutDef

DbxGetStreetOutDef defines the output fields for suggestion processing based on street information. The following Data Access APIs use DbxGetStreetOutDef.

- PBCSGetStreetListCity (PBCSGetStreetListCityFlat)
- PBCSGetStreetListZip (PBCSGetStreetListZipFlat)

Syntax

```
typedef struct DbxGetStreetOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    street[30];
    char    cFiller1;
    long    streetLen;
} DbxGetStreetOutDef, * pDbxGetStreetOutDef;
```

Field Descriptions

Table 94: DbxGetStreetOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
street	Street Name.
streetLen	Street name length.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **

      ** Name: PBCSCGSO.CPY                                         **

      **                                                            **

      ** Use: COBOL Copybook for DbxGetStreetOutDefinition          **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBCS-CGSO-GETSTREETOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGSO'.
         10  PBCS-CGSO-STREET                PIC X(030).
         10  FILLER                          PIC X(001).
         10  PBCS-CGSO-STREET-LEN            PIC S9(08) BINARY.
         10  FILLER                          PIC X(004).
```

Related APIs

PBCSGetStreetListCity

PBCSGetStreetListZip

DbxGetStreetQInDef

DbxGetStreetQInDef structure defines the required input fields for suggestion processing based on street information. The following Data Access APIs use DbxGetStreetQInDef structure.

- PBCSGetStreetListCityQ (PBCSGetStreetListCityQFlat)

Syntax

```
typedef struct DbxGetStreetQInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lZip;
    long    lCount;
    char    cStartLetters[50];
    char    cFiller2[2];
    long    lStartLettersLen;
    char    cState[10];
    char    cFiller3[2];
    long    lStateLen;
    char    cCity[50];
    char    cFiller4[2];
    long    lCityLen;
} DbxGetStreetQInDef, * pDbxGetStreetQInDef;
```

Field Descriptions

Table 95: DbxGetStreetQInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lZip	ZIP Code.
lCount	Maximum number of results to be returned.
cStartLetters	Beginning letters in the street name.
lStartLettersLen	Length of beginning letters in street name.
cState	State.
lStateLen	State length.
cCity	City name.

Field	Description
lCityLen	City name length.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBCSCSQI.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for DbxGetStreetQInDefinition         **

      **                                                            **

      ** Distributed Source Member for the Finalist product from   **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBCS-CSQI-STREETQ.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSQI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSQI-ZIP                   PIC S9(08) BINARY.
         10  PBCS-CSQI-COUNT                 PIC S9(08) BINARY.
         10  PBCS-CSQI-STARTLETTERS          PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CSQI-STARTLETTERS-LEN      PIC S9(08) BINARY.
         10  PBCS-CSQI-STATE                 PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSQI-STATE-LEN             PIC S9(08) BINARY.
         10  PBCS-CSQI-CITY                  PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CSQI-CITY-LEN              PIC S9(08) BINARY.
         10  FILLER                          PIC X(004).
```

Related APIs

PBCSGetStreetListCityQ

PBCSGetStreetListCityQFlat

DbxGetStreetQOutDef

DbxGetStreetQOutDef defines the output fields for suggestion processing based on street information. The following Data Access APIs use DbxGetStreetQOutDef.

- PBCSGetStreetListCityQ (PBCSGetStreetListCityQFlat)

Syntax

```
typedef struct DbxGetStreetQOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    int     streetIndex;
    char    streetName[50];
    char    cFiller2[2];
    long    streetLen;
} DbxGetStreetQOutDef, * pDbxGetStreetQOutDef;
```

Field Descriptions

Table 96: DbxGetStreetQOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
streetIndex	Index entry in the list of streets for the ZIP Code used as the search criteria.
streetName	Street name.
streetLen	Street name length.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **

      ** Name: PBCSCSQO.CPY                                         **

      **                                                            **

      ** Use: COBOL Copybook for DbxGetStreetQOutDefinition         **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBCS-CSQO-STREETQ.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSQO'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSQO-STREETINDEX           PIC S9(08) BINARY.
         10  PBCS-CSQO-STREET-NAME           PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CSQO-STREET-LEN            PIC S9(08) BINARY.
```

Related APIs

PBCSGetStreetListCityQ

PBCSGetStreetListCityQFlat

DbxGetSuggestionInDef

The DbxGetSuggestionInDef structure defines the required input fields for suggestion processing based on street information. The following is a list of the Data Access APIs that use the DbxGetSuggestionInDef structure.

- PBCSCreateSuggestionList (PBCSCreateSuggestionListFlat)

Syntax

```
typedef struct DbxGetSuggestionInDefinition
{
    char    cVersion[VERSION_LEN];      /* Version of structure   */
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lCount;
    long    lMode;
    char    firm[60];
    char    cFiller2[2];
    long    lFirmLen;
    char    urb[29];
    char    cFiller3[3];
    long    lUrbLen;
    char    range[11];
    char    cFiller4;
    long    lRangeLen;
    char    preDir[4];
    long    lPreDirLen;
    char    streetName[30];
    char    cFiller5[2];
    long    lStreetNameLen;
    char    postDir[4];
    long    lPostDirLen;
    char    suffix[5];
    char    cFiller6[3];
    long    lSuffixLen;
    char    unit[20];
    long    lUnitLen;
    char    unitNum[20];
    long    lUnitNumLen;
    char    unit2[20];
    long    lUnit2Len;
    char    unitNum2[20];
    long    lUnitNum2Len;
    char    pmb[20];
    long    lPmbLen;
    char    pmbNum[20];
    long    lPmbNumLen;
    char    city[30];
    char    cFiller7[2];
    long    lCityLen;
    char    state[20];
    long    lStateLen;
    char    zip[6];
    char    cFiller8[2];
    long    lZipLen;
    char    zip4[5];
    char    cFiller9[3];
    long    lZip4Len;
    char    carr[5];
```

```
    char    cFiller10[3];
    long    lCarrLen;
}DbxGetSuggestionInDef, *pDbxGetSuggestionInDef;
```

Field Descriptions

Table 97: DbxGetSuggestionInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lCount	Maximum number of results to be returned.
lMode	The type of list to build. **STREET_MODE** Build a list of streets. Value is 0x01. **RR_MODE** Build a list of rural routes. Value is 0x02. **POB_MODE** Build a list of post office boxes. Value is 0x00.
firm	Firm name.
lFirmLen	Length of firm name.
urb	Street urbanization name.
lUrbLen	Length of street urbanization name.
range	Primary range.
lRangeLen	Length of primary range.
preDir	Pre-directional.
lPreDirLen	Length of pre-directional.
streetName	Street name.
lStreetNameLen	Length of street name.

Field	Description
postDir	Post-directional.
lPostDirLen	Length of post-directional.
suffix	Suffix.
lSuffixLen	Length of suffix.
unit	Unit designator 1 type.
lUnitLen	Length of unit designator 1 type.
unitNum	Secondary range.
lUnitNumLen	Length of secondary range.
unit2	Unit designator 2 type.
lUnit2Len	Length of unit designator 2 type.
unitNum2	Unit 2 secondary range.
lUnitNum2Len	Length of unit 2 secondary range.
pmb	PMB or MSC designator.
lPmbLen	Length of PMB or MSC designator.
pmbNum	PMB or MSC number.
lPmbNumLen	Length of PMB or MSC number.
city	City name.
lCityLen	Length of city name.
state	State.
lStateLen	Length of state.

Field	Description
zip	ZIP Code.
lZipLen	Length of ZIP Code.
zip4	ZIP + 4 Code.
lZip4Len	Length of ZIP + 4 Code.
carr	Carrier route.
lCarrLen	Length of carrier route.

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBCSCSGI.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for DbxGetSuggestionInDefinition      **

      **                                                            **

      ** Distributed Source Member for the Finalist product from   **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBCS-CSGI-SUGGESTION.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSGI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSGI-COUNT                 PIC S9(08) BINARY.
         10  PBCS-CSGI-MODE                  PIC S9(08) BINARY.
         10  PBCS-CSGI-FIRM                  PIC X(060).
         10  PBCS-CSGI-FIRM-LEN              PIC S9(08) BINARY.
         10  PBCS-CSGI-URB                   PIC X(029).
```

```
          10  FILLER                          PIC X(003).
          10  PBCS-CSGI-URB-LEN               PIC S9(08) BINARY.
          10  PBCS-CSGI-RANGE                 PIC X(011).
          10  FILLER                          PIC X(001).
          10  PBCS-CSGI-RANGE-LEN             PIC S9(08) BINARY.
          10  PBCS-CSGI-PREDIR                PIC X(004).
          10  PBCS-CSGI-PREDIR-LEN            PIC S9(08) BINARY.
          10  PBCS-CSGI-STREET-NAME           PIC X(030).
          10  FILLER                          PIC X(002).
          10  PBCS-CSGI-STREETNAME-LEN        PIC S9(08) BINARY.
          10  PBCS-CSGI-POSTDIR               PIC X(004).
          10  PBCS-CSGI-POSTDIR-LEN           PIC S9(08) BINARY.
          10  PBCS-CSGI-SUFFIX                PIC X(005).
          10  FILLER                          PIC X(003).
          10  PBCS-CSGI-SUFFIX-LEN            PIC S9(08) BINARY.
          10  PBCS-CSGI-UNIT                  PIC X(020).
          10  PBCS-CSGI-UNIT-LEN              PIC S9(08) BINARY.
          10  PBCS-CSGI-UNITNUM               PIC X(020).
          10  PBCS-CSGI-UNITNUM-LEN           PIC S9(08) BINARY.
          10  PBCS-CSGI-UNIT2                 PIC X(020).
          10  PBCS-CSGI-UNIT2-LEN             PIC S9(08) BINARY.
          10  PBCS-CSGI-UNITNUM2              PIC X(020).
          10  PBCS-CSGI-UNITNUM2-LEN          PIC S9(08) BINARY.
          10  PBCS-CSGI-PMB                   PIC X(020).
          10  PBCS-CSGI-PMB-LEN               PIC S9(08) BINARY.
          10  PBCS-CSGI-PMBNUM                PIC X(020).
          10  PBCS-CSGI-PMBNUM-LEN            PIC S9(08) BINARY.
          10  PBCS-CSGI-CITY                  PIC X(030).
          10  FILLER                          PIC X(002).
          10  PBCS-CSGI-CITY-LEN              PIC S9(08) BINARY.
          10  PBCS-CSGI-STATE                 PIC X(020).
          10  PBCS-CSGI-STATE-LEN             PIC S9(08) BINARY.
          10  PBCS-CSGI-ZIP                   PIC X(006).
          10  FILLER                          PIC X(002).
          10  PBCS-CSGI-ZIP-LEN               PIC S9(08) BINARY.
          10  PBCS-CSGI-ZIP4                  PIC X(005).
          10  FILLER                          PIC X(003).
          10  PBCS-CSGI-ZIP4-LEN              PIC S9(08) BINARY.
          10  PBCS-CSGI-CARR                  PIC X(005).
          10  FILLER                          PIC X(003).
          10  PBCS-CSGI-CARR-LEN              PIC S9(08) BINARY.
```

Related APIs

PBCSCreateSuggestionList

PBCSCreateSuggestionListFlat

DbxGetSuggestionOutDef

The DbxGetSuggestionOutDef structure defines the output fields for suggestion processing based on street information. The following is a list of the Data Access APIs that use the DbxGetSuggestionOutDef structure.

- PBCSCreateSuggestionList (PBCSCreateSuggestionListFlat)

Syntax

```
typedef struct DbxGetSuggestionOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFirm[60];
    char    cFiller1[3];
    long    lFirmLen;
    char    cUrb[29];
    char    cFiller2[3];
    long    lUrbLen;
    char    cRange[10];
    char    cFiller3[2];
    long    lRangeLen;
    char    cPreDirectional[3];
    char    cFiller4;
    long    lPreDirectionalLen;
    char    cStreetName[30];
    char    cFiller4a[2];
    long    lStreetNameLen;
    char    cPostDirectional[3];
    char    cFiller5;
    long    lPostDirectionalLen;
    char    cStreetSuffix[5];
    char    cFiller6[3];
    long    lStreetSuffixLen;
    char    cUnitDesignator[5];
    char    cFiller7[3];
    long    lUnitDesignatorLen;
    char    cUnitNumber[10];
    char    cFiller8[2];
    long    lUnitNumberLen;
    char    cRangeLow[10];
    char    cFiller9[2];
    long    lRangeLowLen;
    char    cRangeHi[10];
    char    cFiller10[2];
    long    lRangeHiLen;
    char    eob;
    char    cUnit2Designator[5];
```

```
    char    cFiller11[2];
    long    lUnit2DesignatorLen;
    char    cUnit2Number[10];
    char    cFiller12[2];
    long    lUnit2NumberLen;
    char    cPMUnitDesignator[4];
    long    lPMUnitDesignatorLen;
    char    cPMUnitNumber[10];
    char    cFiller13[2];
    long    lPMUnitNumberLen;
    char    cCity[30];
    char    cFiller13a[2];
    long    lCityLen;
    char    cState[3];
    char    cFiller14;
    long    lStateLen;
    char    cZip[6];
    char    cFiller15[2];
    long    lZipLen;
}DbxGetSuggestionOutDef, *pDbxGetSuggestionOutDef;
```

Field Descriptions

Table 98: DbxGetSuggestionOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
firm	Firm name.
lFirmLen	Length of firm name.
cUrb	Street urbanization name.
lUrbLen	Length of street urbanization name.
cRange	Primary range.
lRangeLen	Length of primary range.
cPreDirectional	Pre-directional.

Field	Description
lPreDirectionalLen	Length of pre-directional.
cStreetName	Street name.
lStreetNameLen	Length of street name.
cPostDirectional	Post-directional.
lPostDirectionalLen	Length of post-directional.
cStreetSuffix	Street suffix.
lStreetSuffixLen	Length of street suffix.
cUnitDesignator	Unit designator 1 type.
lUnitDesignatorLen	Length of unit designator 1 type.
cUnitNumber[10]	Secondary range.
lUnitNumberLen	Length of secondary range.
cRangeLow	Low range for ranged suggestions.
lRangeLowLen	Length of low range for ranged suggestions
cRangeHi	High range for ranged suggestions.
lRangeHiLen	Length of high range for ranged suggestions.
eob	Type of range. **E** Even only range **O** Odd only range **B** Both even and odd ranges **C** Consolidated. Even and odd ranges consolidated as one using the lowest ZIP Code.
cUnit2Designator	Unit designator 2 type.

Field	Description
lUnit2DesignatorLen	Length of unit designator 2 type.
cUnit2Number	Unit 2 secondary range.
lUnit2NumberLen	Length of unit 2 secondary range.
cPMUnitDesignator	PMB or MSC designator.
lPMUnitDesignatorLen	Length of PMB or MSC designator.
cPMUnitNumber	PMB or MSC number.
lPMUnitNumberLen	Length of PMB or MSC number.
cCity	City name.
lCityLen	Length of city name.
cState	State.
lStateLen	Length of state.
cZip	ZIP Code.
lZipLen	Length of ZIP Code.

COBOL Copybook

```
      ******************************************************************
      **                                                              **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                              **
      ** Name: PBCSCSGO.CPY                                           **
      **                                                              **
      ** Use: COBOL Copybook for DbxGetSuggestionOutDefinition        **
```

```
      **                                                              **

      ** Distributed Source Member for the Finalist product from     **

      ** Precisely                                                    **
      **                                                              **

      ******************************************************************

       05  PBCS-CSGO-SUGGESTION.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSGO'.
         10  PBCS-CSGO-FIRM                  PIC X(060).
         10  FILLER                          PIC X(003).
         10  PBCS-CSGO-FIRM-LEN              PIC S9(08) BINARY.
         10  PBCS-CSGO-URB                   PIC X(029).
         10  FILLER                          PIC X(003).
         10  PBCS-CSGO-URB-LEN               PIC S9(08) BINARY.
         10  PBCS-CSGO-RANGE                 PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-RANGE-LEN             PIC S9(08) BINARY.
         10  PBCS-CSGO-PREDIRECTIONAL        PIC X(003).
         10  FILLER                          PIC X(001).
         10  PBCS-CSGO-PREDIRECTIONAL-LEN    PIC S9(08) BINARY.
         10  PBCS-CSGO-STREET-NAME           PIC X(030).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-STREETNAME-LEN        PIC S9(08) BINARY.
         10  PBCS-CSGO-POSTDIRECTIONAL       PIC X(003).
         10  FILLER                          PIC X(001).
         10  PBCS-CSGO-POSTDIRECTIONAL-LEN   PIC S9(08) BINARY.
         10  PBCS-CSGO-STREETSUFFIX          PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSGO-STREETSUFFIX-LEN      PIC S9(08) BINARY.
         10  PBCS-CSGO-UNITDESIGNATOR        PIC X(005).
         10  FILLER                          PIC X(003).
         10  PBCS-CSGO-UNITDESIGNATOR-LEN    PIC S9(08) BINARY.
         10  PBCS-CSGO-UNITNUMBER            PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-UNITNUMBER-LEN        PIC S9(08) BINARY.
         10  PBCS-CSGO-RANGELOW              PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-RANGELOW-LEN          PIC S9(08) BINARY.
         10  PBCS-CSGO-RANGEHI               PIC X(010).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-RANGEHI-LEN           PIC S9(08) BINARY.
         10  PBCS-CSGO-EOB                   PIC X(001).
           88  PBCS-CSGO-EVEN                           VALUE 'E'.
           88  PBCS-CSGO-ODD                            VALUE 'O'.
           88  PBCS-CSGO-BOTH                           VALUE 'B'.
           88  PBCS-CSGO-CONSOLIDATED                   VALUE 'C'.
         10  PBCS-CSGO-UNIT2DESIGNATOR       PIC X(005).
         10  FILLER                          PIC X(002).
         10  PBCS-CSGO-UNIT2DESIGNATOR-LEN   PIC S9(08) BINARY.
```

```
          10  PBCS-CSGO-UNIT2NUMBER            PIC X(010).
          10  FILLER                           PIC X(002).
          10  PBCS-CSGO-UNIT2NUMBER-LEN        PIC S9(08) BINARY.
          10  PBCS-CSGO-PMUNITDESIGNATOR       PIC X(004).
          10  PBCS-CSGO-PMUNITDESIGNATOR-LEN   PIC S9(08) BINARY.
          10  PBCS-CSGO-PMUNITNUMBER           PIC X(010).
          10  FILLER                           PIC X(002).
          10  PBCS-CSGO-PMUNITNUMBER-LEN       PIC S9(08) BINARY.
          10  PBCS-CSGO-CITY                   PIC X(030).
          10  FILLER                           PIC X(002).
          10  PBCS-CSGO-CITY-LEN               PIC S9(08) BINARY.
          10  PBCS-CSGO-STATE                  PIC X(003).
          10  FILLER                           PIC X(001).
          10  PBCS-CSGO-STATE-LEN              PIC S9(08) BINARY.
          10  PBCS-CSGO-ZIP                    PIC X(006).
          10  FILLER                           PIC X(002).
          10  PBCS-CSGO-ZIP-LEN                PIC S9(08) BINARY.
```

Related APIs

PBCSCreateSuggestionList

PBCSCreateSuggestionListFlat

DbxGetZipCityInDef

The DbxGetZipCityInDef structure defines the required input fields for suggestion processing based on city and state information. The following is a list of the Data Access APIs that use the DbxGetZipCityInDef structure.

- PBCSGetCityList (PBCSGetCityListFlat)
- PBCSGetZipListCity (PBCSGetZipListCityFlat)

Syntax

```
typedef struct DbxGetZipCityInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    state[10];
    char    cFiller1;
    long    stateLen;
    char    city[50];
    char    cFiller2[2];
    long    cityLen;
    long    count;
} DbxGetZipCityInDef, * pDbxGetZipCityInDef;
```

Field Descriptions

Table 99: DbxGetZipCityInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
state	State.
stateLen	Length of state.
city	City name.
cityLen	Length of city name.
count	The maximum number of results to be returned.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBCSCGCI.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxGetZipCityInDefinition          **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************
```

```
       05  PBCS-CGCI-GETZIPCITYIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGCI'.
         10  PBCS-CGCI-STATE                 PIC X(010).
         10  FILLER                          PIC X(001).
         10  PBCS-CGCI-STATE-LEN             PIC S9(08) BINARY.
         10  PBCS-CGCI-CITY                  PIC X(050).
         10  FILLER                          PIC X(002).
         10  PBCS-CGCI-CITY-LEN              PIC S9(08) BINARY.
         10  PBCS-CGCI-COUNT                 PIC S9(08) BINARY.
         10  FILLER                          PIC X(004).
```

Related APIs

PBCSGetCityList

PBCSGetCityListFlat

PBCSGetZipListCity

PBCSGetZipListCity

DbxGetZipInDef

The DbxGetZipInDef structure defines the required input fields for suggestion processing based on ZIP Code information. The following Data Access APIs use the DbxGetZipInDef structure.

- PBCSGetZipList (PBCSGetZipListFlat)

Syntax

```
typedef struct DbxGetZipInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    zipExpr[10];
    char    cFiller1;
    long    zipLen;
    long    count;
} DbxGetZipInDef, * pDbxGetZipInDef;
```

Field Descriptions

Table 100: DbxGetZipInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
zipExpr	ZIP Code.
zipLen	Length of ZIP Code.
count	Maximum number of results to be returned.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **
      ** Name: PBCSCGZI.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxGetZipInDefinition              **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************

       05  PBCS-CGZI-GETZIPIN.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGZI'.
         10  PBCS-CGZI-ZIPEXPR               PIC X(010).
         10  FILLER                          PIC X(001).
```

```
          10  PBCS-CGZI-ZIP-LEN                PIC S9(08) BINARY.
          10  PBCS-CGZI-COUNT                 PIC S9(08) BINARY.
          10  FILLER                          PIC X(004).
```

Related APIs

PBCSGetZipList

PBCSGetZipListFlat

DbxGetZipOutDef

The DbxGetZipOutDef structure defines the required input fields for suggestion processing based on ZIP Code information. The following Data Access APIs use the DbxGetZipOutDef structure.

- PBCSGetCityList (PBCSGetCityListFlat)
- PBCSGetZipList (PBCSGetZipListFlat)
- PBCSGetZipListCity (PBCSGetZipListCityFlat)

Syntax

```
typedef struct DbxGetZipOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    zipcode[6];
    char    cFiller1;
    long    zipcodelen;
    char    cityName[29];
    char    cFiller2[2];
    char    cCityInd;
    long    cityNamelen;
    char    countyName[26];
    char    cFiller3[2];
    long    countyNamelen;
    char    stateAbbreviation[3];
    char    cFiller4;
    long    stateAbbreviationlen;
} DbxGetZipOutDef, * pDbxGetZipOutDef;
```

Field Descriptions

Table 101: DbxGetZipOutDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
zipcode	ZIP Code.
zipcodelen	Length of ZIP Code.
cityName	City name.
cCityInd	City relationship to ZIP Code: • P — City name is the preferred city for the ZIP Code. • A — City name is an acceptable city for the ZIP Code. • N — City name is a non-mailing city for the ZIP Code.
cityNamelen	Length of city name.
countyName	County name.
countyNamelen	Length of county name.
stateAbbreviation	Two-character state abbreviation.
stateAbbreviationlen	Length of state abbreviation.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) 2012 Precisely           All Rights Reserved.   **
      **                                                            **
      ** Name: PBCSCGZO.CPY                                         **
```

```
      **                                                               **
      ** Use: COBOL Copybook for DbxGetZipOutDefinition                **
      **                                                               **
      ** Distributed Source Member for the Finalist product from       **
      ** Precisely                                                     **
      **                                                               **
      *******************************************************************

       05  PBCS-CGZO-GETZIPOUT.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CGZO'.
         10  PBCS-CGZO-ZIPCODE               PIC X(006).
         10  FILLER                          PIC X(001).
         10  PBCS-CGZO-ZIPCODELEN            PIC S9(08) BINARY.
         10  PBCS-CGZO-CITY-NAME             PIC X(029).
         10  FILLER                          PIC X(002).
         10  PBCS-CGZO-CITY-IND              PIC X(001).
           88  PBCS-CGZO-PREFERRED                      VALUE 'P'.
           88  PBCS-CGZO-ACCEPTABLE                     VALUE 'A'.
           88  PBCS-CGZO-NON-MAILING                    VALUE 'N'.
         10  PBCS-CGZO-CITYNAMELEN           PIC S9(08) BINARY.
         10  PBCS-CGZO-COUNTY-NAME           PIC X(026).
         10  FILLER                          PIC X(002).
         10  PBCS-CGZO-COUNTYNAMELEN         PIC S9(08) BINARY.
         10  PBCS-CGZO-STATEABBREVIATION     PIC X(003).
         10  FILLER                          PIC X(001).
         10  PBCS-CGZO-STATEABBREVIATIONLEN  PIC S9(08) BINARY.
```

Related APIs

PBCSGetCityList

PBCSGetCityListFlat

PBCSGetZipList

PBCSGetZipListFlat

PBCSGetZipListCity

PBCSGetZipListCityFlat

DbxLastLineInDef

The DbxLastLineInDef structure defines the required input fields for suggestion processing based on last line information. The following is a list of the Data Access APIs that use the DbxLastLineInDef structure.

- PBCSLastLine (PBCSLastLineFlat)

Syntax

```
typedef struct DbxLastLineInDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    char    cLastLine[100];
    long    lLastLineLen;
    long    lCount;
} DbxLastLineInDef, * pDbxLastLineInDef;
```

Field Descriptions

Table 102: DbxLastLineInDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
cLastLine	Last line information includes city, state, and ZIP Code.
lLastLineLen	Length of the last line field.
lCount	The maximum number of results to be returned.

COBOL Copybook

```
      ************************************************************************

      **                                                                    **
```

```
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                          **
      ** Name: PBCSCLLI.CPY                                      **
      **                                                          **
      ** Use: COBOL Copybook for DbxLastLineInDefinition         **
      **                                                          **
      ** Distributed Source Member for the Finalist product from  **
      ** Precisely                                               **
      **                                                          **
      **************************************************************

       05  PBCS-CLLI-GETLASTLINE.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CLLI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CLLI-LASTLINE              PIC X(100).
         10  PBCS-CLLI-LASTLINE-LEN          PIC S9(08) BINARY.
         10  PBCS-CLLI-COUNT                 PIC S9(08) BINARY.
```

Related APIs

PBCSLastLine

PBCSLastLineFlat

DbxLastLineOutDef

The DbxLastLineOutDef structure defines the output fields for suggestion processing based on last line information. The following is a list of the Data Access APIs that use the DbxLastLineOutDef structure.

- PBCSLastLine (PBCSLastLineFlat)

Syntax

```
typedef struct DbxLastLineOutDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
```

```
    char    cCity[29];
    char    cFiller1[2];
    long    lCityLen;
    char    cState[3];
    char    cFiller2;
    long    lStateLen;
    char    cZip[6];
    char    cFiller3[2];
    long    lZipLen;
    unsigned char mailingInd;
    char    cPreferredCity[29];
    char    cFiller4[2];
    long    lPreferredCityLen;
    int     zipType;
} DbxLastLineOutDef, *pDbxLastLineOutDef;
```

Field Descriptions

Table 103: DbxLastLineOutDef Field Descriptions

Field	Descriptions
cVersion	Structure version number.
cApiId	Structure identifier.
cCity	City name.
lCityLen	Length of city name.
cState	State.
lStateLen	Length of state.
cZip	ZIP Code.
lZipLen	Length of ZIP Code.
mailingInd	Mailing indicator. **1** Mailing address **2** Nonmailing address
cPreferredCity	Preferred city name.

Field	Descriptions
lPreferredCityLen	Length of preferred city name.
zipType	ZIP Code type. **0** Unique ZIP Code **1** P. O. Box only **2** Military installation **3** Standard ZIP Code

COBOL Copybook

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBCSCLLO.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for DbxLastLineOutDefinition          **

      **                                                            **

      ** Distributed Source Member for the Finalist product from   **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBCS-CLLO-GETLASTLINE.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CLLO'.
         10  PBCS-CLLO-CITY                  PIC X(029).
         10  FILLER                          PIC X(002).
         10  PBCS-CLLO-CITY-LEN              PIC S9(08) BINARY.
         10  PBCS-CLLO-STATE                 PIC X(003).
         10  FILLER                          PIC X(001).
         10  PBCS-CLLO-STATE-LEN             PIC S9(08) BINARY.
         10  PBCS-CLLO-ZIP                   PIC X(006).
         10  FILLER                          PIC X(002).
         10  PBCS-CLLO-ZIP-LEN               PIC S9(08) BINARY.
```

```
          10  PBCS-CLLO-MAILING-IND            PIC X(001).
            88  PBCS-CLLO-MAILING                       VALUE '1'.
            88  PBCS-CLLO-NONMAILING                    VALUE '2'.
          10  PBCS-CLLO-PREFERREDCITY         PIC X(029).
          10  FILLER                          PIC X(002).
          10  PBCS-CLLO-PREFERREDCITY-LEN     PIC S9(08) BINARY.
          10  PBCS-CLLO-ZIP-TYPE              PIC S9(08) BINARY.
            88  PBCS-CLLO-UNIQUE                        VALUE +0.
            88  PBCS-CLLO-POBOX-ONLY                    VALUE +1.
            88  PBCS-CLLO-MILITARY                      VALUE +2.
            88  PBCS-CLLO-STANDARD                      VALUE +3.
```

Related APIs

PBCSLastLine

PBCSLastLineFlat

DbxStateDef

The DbxStateDef structure allows you to call each function with different attributes to return a specific size and quantity for the returned list.

Syntax

```
typedef struct DbxStateDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    long    lStartElem;
    long    lCount;
    char    cMore;
    char    cFiller2[3];
} DbxStateDef, * pDbxStateDef;
```

Example

This example retrieves groups of 30 streets starting at the 40th street. ZIP Code 60148 has 267 streets. This example will loop through returning 30 streets until it reaches 267.

```
     PBCSState.lCount = 30;
     PBCSState.lStartElem = 40;
     do
     {
         count = PBCSGetStreetListZip( PBCSstreetIn,
```

```
                                  PBCSstreetList, &PBCSState);
        PBCSState.lStartElem =  PBCSState.lStartElem +
             PBCSState.lCount;
    } while ( PBCSState.cMore == 'Y'  );
```

Field Descriptions

Table 104: DbxStateDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier.
lStartElem	Number to start at for a count of lCount.
lCount	Maximum number of results to be returned.
cMore	cMore returns from the function call signaling whether there are more elements available. **Y** Yes. Additional elements available. **N** No additional elements available.

COBOL Copybook

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                            **
      ** Name: PBCSCSSI.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for DbxStateDefinition                 **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
```

```
      ** Precisely                                                  **
      **                                                                **

      ********************************************************************

       05  PBCS-CSSI-STATE.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'CSSI'.
         10  FILLER                          PIC X(003).
         10  PBCS-CSSI-STARTELEM             PIC S9(08) BINARY.
         10  PBCS-CSSI-COUNT                 PIC S9(08) BINARY.
         10  PBCS-CSSI-MORE                  PIC X(001).
         10  FILLER                          PIC X(003).
```

Related APIs

The following Data Access APIs use the DbxStateDef structure:

PBCSCreateSuggestionList

PBCSCreateSuggestionListFlat

PBCSGetCityList

PBCSGetCityListFlat

PBCSGetFirmListCity

PBCSGetFirmListCityFlat

PBCSGetFirmListZip

PBCSGetFirmListZipFlat

PBCSGetPrimaryListCity

PBCSGetPrimaryListCityFlat

PBCSGetPrimaryListZip

PBCSGetPrimaryListZipFlat

PBCSGetSecondaryListCity

PBCSGetSecondaryListCityFlat

PBCSGetSecondaryListZip

PBCSGetSecondaryListZipFlat

PBCSGetStreetAliasCity

PBCSGetStreetAliasCityFlat

PBCSGetStreetAliasZip

PBCSGetStreetAliasZipFlat

Using the Product Structures

PBFN3553Def

The PBFN3553Def C structure contains the CASS Stage 2 header information.

Syntax

```
typedef struct PBFN3553Definition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    /* Information to build the 3553 stage2 header record */
    char    filler1[45];
    char    Z4Change_Company_Name[40];
    char    LOT_Utility_Company_Name[40];
    char    Z4Change_Cfg[3];
    char    LOT_Utility_Cfg[3];
    char    Z4Change_Software_Name[30];
    char    Z4Change_Software_Ver[16];
    char    LOT_Software_Name[30];
    char    LOT_Software_Ver[16];
```

```
    char    List_Processors_name[25];
    char    MasterFile_Process_Date[8];
    char    Z4Change_Process_Date[8];
    char    LOT_Process_Date[8];
    char    CRIS_Process_Date[8];
    char    ZIP4_Database_Date[8];
    char    filler2[8];
    char    LOT_Database_Date[8];
    char    CRIS_Database_Date[8];
    char    Adress_list_name[25];
    char    Number_of_lists[3];
    char    Total_records[6];
    char    Zip4_Total_coded[6];
    char    Zip4_Valid_From_Date[8];
    char    Zip4_Valid_To_Date[8];
    char    Z4Change_processed[6];
    char    filler3[38];
    char    Zip5Digit_Total_coded[6];
    char    Zip5Digit_Valid_From_Date[8];
    char    Zip5Digit_Valid_To_Date[8];
    char    CRIS_Total_coded[6];
    char    CRIS_Valid_From_Date[8];
    char    CRIS_Valid_To_Date[8];
    char    LOT_Total_coded[6];
    char    LOT_Valid_From_Date[8];
    char    LOT_Valid_To_Date[8];
    char    filler4[8];
    char    Z4Change_Database_Date[8];
    char    HighRiseExact[6];
    char    HighRiseDefault[6];
    char    RuralExact[6];
    char    RuralDefault[6];
    char    LACS[6];
    char    TotalEWS[6];
    char    TotalSuiteLink[6];
    char    filler5[40];
    char    DPVDataUsed;
    char    DPVDate[8];
    char    TestPlatform[12];
    char    TestConfig[3];
#define CASS_STATE2_HEADER_RECORD_LENGTH 600
    /* Additional information to build the 3553 print report */
    char    CASS_Company_Name[40];
    char    CASS_Software_Name[30];
    char    CASS_Software_Ver[16];
    char    CASS_Cfg[3];
    char    MailerName[30];
    char    MailerAddr1[30];
    char    MailerAddr2[30];
    char    MailerAddr3[30];
    char    MailerAddr4[30];
    char    MailerAddr5[30];
```

```
    char    cFiller;
} PBFN3553Def, *pPBFN3553Def;
```

Field Descriptions

Table 105: PBFN3553Def Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to NCAS.
Z4Change_Company_Name	Z4CHANGE/DirectDPV-certified company name field.
LOT_Utility_Company_Name	eLOT-certified company name field.
Z4Change_Cfg	Z4Change/DirectDPV configuration setting.
LOT_Utility_Cfg	eLOT configuration setting.
Z4Change_Software_Name	Z4Change/DirectDPV-certified software name.
Z4Change_Software_Ver	Z4Change/DirectDPV-certified software version number.
LOT_Software_Name	eLOT-certified software name.
LOT_Software_Ver	eLOT-certified software version number.
List_Processors_name	Name of company or individual who processed the address list.
MasterFile_Process_Date	Master file process date.
Z4Change_Process_Date	Z4Change/DirectDPV process date.
LOT_Process_Date	eLOT process date.
CRIS_Process_Date	CRIS process date.
ZIP4_Database_Date	ZIP + 4/DPV database date.

Field	Description
LOT_Database_Date	eLOT database date.
CRIS_Database_Date	CRIS database date.
Adress_list_name	Address list name.
Number_of_lists	Number of lists processed.
Total_records	Total records submitted.
Zip4_Total_coded	Total records ZIP + 4 Coded.
Zip4_Valid_From_Date	ZIP + 4 valid from date.
Zip4_Valid_To_Date	ZIP + 4 valid to date.
Z4Change_processed	Total records processed through Z4Change.
Zip5Digit_Total_coded	Total number of records 5-digit coded.
Zip5Digit_Valid_From_Date	5-digit valid from date.
Zip5Digit_Valid_To_Date	5-digit valid to date.
CRIS_Total_coded	Total number of records assigned a carrier route.
CRIS_Valid_From_Date	Carrier route valid from date.
CRIS_Valid_To_Date	Carrier route valid to date.
LOT_Total_coded	Total number of records assigned a LOT code.
LOT_Valid_From_Date	eLOT valid from date.
LOT_Valid_To_Date	eLOT valid to date.
Z4Change_Database_Date	Z4Change/DirectDPV database date.
HighRiseExact	Total number of exact matches to highrise records coded to in this run.

Field	Description
HighRiseDefault	Total number of default matches to highrise records coded to in this run.
RuralExact	Total number of exact matches to rural route/highway contract records coded to in this run.
RuralDefault	Total number of default matches to rural route/highway contract records coded to in this run.
LACS	Total number of LACS processed records.
TotalEWS	Total number of EWS processed records.
TotalSuiteLink	Total number of SuiteLink processed records.
DPVDataUsed	DPV file type used: **F** DPV Full File **S** DPV Split File **L** DPV Flat File
DPVDate	Date of the Delivery Point Validation (DPV) File used for processing.
TestPlatform	Platform on which the Stage II test was run (for example, Windows).
TestConfig	System configuration used for the Stage II test.
CASS_Company_Name	CASS-certified company name.
CASS_Software_Name	CASS-certified software name.
CASS_Software_Ver	CASS-certified software version number.
CASS_Cfg	CASS-certified configuration value used to process list.
MailerName	Mailer name to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.
MailerAddr1	Mailer address line 1 to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.

Field	Description
MailerAddr2	Mailer address line 2 to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.
MailerAddr3	Mailer address line 3 to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.
MailerAddr4	Mailer address line 4 to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.
MailerAddr5	Mailer address line 5 to display in the D3 box on USPS Form 3553 (CASS Summary Report). Maximum length allowed is 30 characters.

COBOL Copybook - PBFNNCAS

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNNCAS.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for PBFN3553Definition                 **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************

       05  PBFN-NCAS-3553.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'NCAS'.
         10  FILLER                          PIC X(045).
         10  PBFN-NCAS-Z4CHG-COMPANY-NAME    PIC X(040).
         10  PBFN-NCAS-LOT-UTIL-COMP-NAME    PIC X(040).
         10  PBFN-NCAS-Z4CHANGE-CFG          PIC X(003).
         10  PBFN-NCAS-LOT-UTILITY-CFG       PIC X(003).
         10  PBFN-NCAS-Z4CHG-SOFTWARE-NAME   PIC X(030).
         10  PBFN-NCAS-Z4CHG-SOFTWARE-VER    PIC X(016).
```

```
          10  PBFN-NCAS-LOT-SOFTWARE-NAME      PIC X(030).
          10  PBFN-NCAS-LOT-SOFTWARE-VER       PIC X(016).
          10  PBFN-NCAS-LIST-PROCESSORS-NAME   PIC X(025).
          10  PBFN-NCAS-MF-PROCESS-DATE        PIC X(008).
          10  PBFN-NCAS-Z4CHG-PROCESS-DATE     PIC X(008).
          10  PBFN-NCAS-LOT-PROCESS-DATE       PIC X(008).
          10  PBFN-NCAS-CRIS-PROCESS-DATE      PIC X(008).
          10  PBFN-NCAS-ZIP4-DATABASE-DATE     PIC X(008).
          10  FILLER                           PIC X(008).
          10  PBFN-NCAS-LOT-DATABASE-DATE      PIC X(008).
          10  PBFN-NCAS-CRIS-DATABASE-DATE     PIC X(008).
          10  PBFN-NCAS-ADRESS-LIST-NAME       PIC X(025).
          10  PBFN-NCAS-NUMBER-OF-LISTS        PIC X(003).
          10  PBFN-NCAS-TOTAL-RECORDS          PIC X(006).
          10  PBFN-NCAS-ZIP4-TOTAL-CODED       PIC X(006).
          10  PBFN-NCAS-ZIP4-VALID-FROM-DATE   PIC X(008).
          10  PBFN-NCAS-ZIP4-VALID-TO-DATE     PIC X(008).
          10  PBFN-NCAS-Z4CHANGE-PROCESSED     PIC X(006).
          10  FILLER                           PIC X(038).
          10  PBFN-NCAS-ZIP5-TOTAL-CODED       PIC X(006).
          10  PBFN-NCAS-ZIP5-VALID-FROM-DATE   PIC X(008).
          10  PBFN-NCAS-ZIP5-VALID-TO-DATE     PIC X(008).
          10  PBFN-NCAS-CRIS-TOTAL-CODED       PIC X(006).
          10  PBFN-NCAS-CRIS-VALID-FROM-DATE   PIC X(008).
          10  PBFN-NCAS-CRIS-VALID-TO-DATE     PIC X(008).
          10  PBFN-NCAS-LOT-TOTAL-CODED        PIC X(006).
          10  PBFN-NCAS-LOT-VALID-FROM-DATE    PIC X(008).
          10  PBFN-NCAS-LOT-VALID-TO-DATE      PIC X(008).
          10  FILLER                           PIC X(008).
          10  PBFN-NCAS-Z4CHG-DATABASE-DATE    PIC X(008).
          10  PBFN-NCAS-HIGHRISEEXACT          PIC X(006).
          10  PBFN-NCAS-HIGHRISEDEFAULT        PIC X(006).
          10  PBFN-NCAS-RURALEXACT             PIC X(006).
          10  PBFN-NCAS-RURALDEFAULT           PIC X(006).
          10  PBFN-NCAS-LACS                   PIC X(006).
          10  PBFN-NCAS-TOTALEWS               PIC X(006).
          10  PBFN-NCAS-TOTALSUITELINK         PIC X(006).
          10  FILLER                           PIC X(040).
          10  PBFN-NCAS-DPVDATAUSED            PIC X(001).
          10  PBFN-NCAS-DPVDATE                PIC X(008).
          10  PBFN-NCAS-TESTPLATFORM           PIC X(012).
          10  PBFN-NCAS-TESTCONFIG             PIC X(003).
          10  PBFN-NCAS-CASS-COMPANY-NAME      PIC X(040).
          10  PBFN-NCAS-CASS-SOFTWARE-NAME     PIC X(030).
          10  PBFN-NCAS-CASS-SOFTWARE-VER      PIC X(016).
          10  PBFN-NCAS-CASS-CFG               PIC X(003).
          10  PBFN-NCAS-MAILER-NAME            PIC X(030).
          10  PBFN-NCAS-MAILERADDR1            PIC X(030).
          10  PBFN-NCAS-MAILERADDR2            PIC X(030).
          10  PBFN-NCAS-MAILERADDR3            PIC X(030).
          10  PBFN-NCAS-MAILERADDR4            PIC X(030).
          10  PBFN-NCAS-MAILERADDR5            PIC X(030).
          10  FILLER                           PIC X(001).
```

PBFNAddressDataDef

The PBFNAddressDataDef C structure passes parsed and unparsed address information to and from the postal coding functions using character arrays.

Using the PBFNAddressDataDef structure simplifies the PBFNProcess API by replacing the referenced structures.

NOTE: The PBFNAddressDataDef structure has replaced the following structures:

- PBFNAddressInfoDef
- PBFNDPVDetailDef
- PBFNLabelLineDef
- PBFNLACSSeedDetDef
- PBFNParsedAdrAltDef
- PBFNParsedAdrDef
- PBFNProcessDataAltDef
- PBFNProcessDataDef
- PBFNRtnFirmDef
- PBFNRtnOrigDataDef

The PBFNAddressDataDef structure also replaces the following structure that is not being deprecated:

- **PBFNExtendedErrorDef**

Syntax

```
typedef struct PBFNAddressDataDefinition
{
    char        cVersion[VERSION_LEN];
    char        cApiId[APIID_LEN];
    char        cFiller0;
    char        cFirm[128];
    short       sFirmLen;
    char        cUrb[128];
    short       sUrbLen;
    char        cAddress1[128];
    short       sAddress1Len;
    char        cAddress2[128];
    short       sAddress2Len;
    char        cUnit1[20];
    short       sUnit1Len;
    char        cUnit2[20];
    short       sUnit2Len;
    char        cCity[128];
    short       sCityLen;
    char        cState[20];
```

```
    short       sStateLen;
    char        cZip[ZIP_LEN];
    char        cZip4[ZIP4_LEN];
    char        cCrRte[CRRT_LEN];
    char        cPMBUnit[16];
    short       sPMBUnitLen;
    char        cInputAddressType;
    char        cRange[11];
    short       sRangeLen;
    char        cFiller2;
    char        cPreDirectional[3];
    short       sPreDirectionalLen;
    char        cFiller3;
    char        cStreetName[29];
    short       sStreetNameLen;
    char        cFiller4;
    char        cPostDirectional[3];
    short       sPostDirectionalLen;
    char        cFiller5;
    char        cStreetSuffix[5];
    short       sStreetSuffixLen;
    char        cFiller6;
    char        cUnitDesignator[11];
    short       sUnitDesignatorLen;
    char        cFiller7;
    char        cUnitNumber[11];
    short       sUnitNumberLen;
    char        cFiller8;
    char        cUnit2Designator[11];
    short       sUnit2DesignatorLen;
    char        cFiller9;
    char        cUnit2Number[11];
    short       sUnit2NumberLen;
    char        cPMUnitDesignator[4];
    short       sPMUnitDesignatorLen;
    char        cFiller10;
    char        cPMUnitNumber[11];
    short       sPMUnitNumberLen;
    char        cAlternateStreet[80];
    short       sAlternateStreetLen;
    char        cAlternateStreetType;
    char        cAltRange[11];
    short       sAltRangeLen;
    char        cFiller11;
    char        cAltPreDir[3];
    short       sAltPreDirLen;
    char        cFiller12;
    char        cAltStreetName[29];
    short       sAltStreetNameLen;
    char        cFiller13;
    char        cAltPostDir[3];
    short       sAltPostDirLen;
    char        cFiller14;
```

```
    char        cAltSuffix[5];
    short       sAltSuffixLen;
    char        cFullCityName[30];
    short       sFullCityNameLen;
    char        cAbbrCityName[14];
    short       sAbbrCityNameLen;
    char        cNonMailingCityName[30];
    short       sNonMailingCityNameLen;
    char        cFiller15;
    char        cPreferredCityName[29];
    short       sPreferredCityNameLen;
    char        cPreferredState[3];
    char        cStreetType;
    char        cDelPoint[4];
    char        cFIPSCode[FIPS_LEN];
    char        cCountyName[30];
    short       sCountyNameLen;
    char        cAdvanceBarcode[ADVBC_LEN];
    char        cFiveDigitBarcode[FIVEBC_LEN];
    char        cLOTCode[LOT_LEN];
    char        cMatchLevel;
    char        cDefaultMatch;
    char        cLACS;
    char        cLACSRtnCode[LACS_RTN_CODE];
    char        cAutoCR;
    char        cRDI;
    char        cFiveDigitScheme[FIVE_DGT_SCH_LEN];
    char        cCongressionalDist[CONG_DIST_LEN];
    char        cDPVFlags[DPV_FLAGS_LEN];
    char        cDPVFootnote[20];
    char        cFiller16;
    short       sDPVFootnoteLen;
    char        cSeasonalFlags[SEASONAL_FLGS_LEN];
    char        cError[ERROR_LEN];
    char        cProcessDate[PROC_DATE_LEN];
    char        cDPVNoStatFound;
    char        cUserKey[41];
    char        cSteLnkMatchCode;
    char        cSteLnkFidelityCode;
    char        cSteLnkRtnCode[SLK_RTN_CODE];
    char        cDPVDNAFound;
    char        cExtra[256];
    short       sExtraLen;
    short       sSeedViolationEnc;
    short       sDPVKeyNCOA;
    char        cDPVVacantFound;
    char        cDPVZip4[5];
    char        cDPVPBSAFound;
    char        cPOBoxZone;
    char        cZipValid;
    /* AdsInfo information */
    char            cAddressUnchanged;
    char            cDualAddressInd;
```

```
    char            cNonDeliverableInd;
    char            cZipMove;
    char            cExceptionsInd;
    unsigned short  sAddrMatchLevel;
    /************** sAddrMatchLevel settings
*****************************/
    #define PBFN_ADSLVL_STREET              0x0001
    #define PBFN_ADSLVL_SECONDARY           0x0002
    #define PBFN_ADSLVL_FIRM_PRIMARY        0x0004
    #define PBFN_ADSLVL_FIRM_SECONDARY      0x0006
    #define PBFN_ADSLVL_HIGHRISE_DEFAULT    0x0008
    #define PBFN_ADSLVL_HIGHRISE_SECONDARY  0x0010
    #define PBFN_ADSLVL_RRHC_DEFAULT        0x0020
    #define PBFN_ADSLVL_RRHC_SECONDARY      0x0022
    #define PBFN_ADSLVL_GENDEL              0x0040
    #define PBFN_ADSLVL_PO                  0x0080
    #define PBFN_ADSLVL_UNIQUE_FOUND        0x0100
    #define PBFN_ADSLVL_UNIQUE_DEFAULT      0x0200
    #define PBFN_ADSLVL_MIL_DEFAULT         0x0400
    #define PBFN_ADSLVL_MIL_SECONDARY       0x0402

/*****************************************************************************/

    unsigned char   cRangeMatchInd;
    /************** cRangeMatchInd settings
*****************************/
    #define PBFN_RNKRNGPRIME                0x00
    #define PBFN_RNKRNGSEC1                 0x01
    #define PBFN_RNKRNGSEC2                 0x02
    #define PBFN_RNKRNGSEC12COMBO           0x03
    #define PBFN_RNKRNGALPHARECOMBO         0x04
    #define PBFN_RNKRNGDASHRECOMBO          0x05
    #define PBFN_RNKRNGALTERNATE            0x06
    #define PBFN_RNKRNGSECUD                0x07

/*****************************************************************************/

    unsigned char   cReturnLevel;
    /************** cReturnLevel settings
*******************************/
    #define PBFN_RTNZIP4                    0x00
    #define PBFN_RTNZIPCRRT                 0x01
    #define PBFN_RTNZIP                     0x02
    #define PBFN_RTNUNKNOWN                 0x80

/*****************************************************************************/

    /* Location settings */
    unsigned char   cAddressLoc;
    /************** cAddressLoc settings
*******************************/
    #define PBFN_ADSLOC_L1                  0x01
    #define PBFN_ADSLOC_L2                  0x02
```

```
    #define PBFN_ADSLOC_L1_L2               0x03
    #define PBFN_ADSLOC_FIRM_L1             0x04
    #define PBFN_ADSLOC_FIRM_L2             0x08
    #define PBFN_ADSLOC_UNKNOWN             0x80

/*****************************************************************************/

    unsigned char   cUnitLoc;
    /************** cUnitLoc settings
**********************************/
    #define PBFN_UNITLOC_NONE               0x00
    #define PBFN_UNITLOC_L1                 0x01
    #define PBFN_UNITLOC_L2                 0x02
    #define PBFN_UNITLOC_L1_L2              0x03
    #define PBFN_UNITLOC_U1                 0x04
    #define PBFN_UNITLOC_U2                 0x08
    #define PBFN_UNITLOC_U1_U2              0x0b
    #define PBFN_UNITLOC_L1_U1              0x05
    #define PBFN_UNITLOC_L1_U2              0x09
    #define PBFN_UNITLOC_L2_U1              0x06
    #define PBFN_UNITLOC_L2_U2              0x0a
    #define PBFN_UNITLOC_UNKNOWN            0x80

/*****************************************************************************/

    unsigned char   cFirmLoc;
   /************** cFirmLoc settings **********************************/

    #define PBFN_FIRMLOC_FIRM               0x01
    #define PBFN_FIRMLOC_L1                 0x02
    #define PBFN_FIRMLOC_L2                 0x04
    #define PBFN_FIRMLOC_UNKNOWN            0x80

/*****************************************************************************/

    unsigned char   cPMLoc;
    /************** cPMLoc settings
**********************************/
    #define PBFN_PMLOC_NONE                 0x00
    #define PBFN_PMLOC_L1                   0x01
    #define PBFN_PMLOC_L2                   0x02
    #define PBFN_PMLOC_L1_L2                0x03
    #define PBFN_PMLOC_PM                   0x04
    #define PBFN_PMLOC_U1                   0x08
    #define PBFN_PMLOC_U2                   0x10
    #define PBFN_PMLOC_UNKNOWN              0x80

/*****************************************************************************/

    /* Address type information */
    unsigned char   cAddressType;
   /************** cAddressType settings ******************************/
```

```
    #define PBFN_ADSTYPE_STREET             0x00
    #define PBFN_ADSTYPE_PO_BOX             0x01
    #define PBFN_ADSTYPE_RR                 0x02
    #define PBFN_ADSTYPE_HC                 0x04
    #define PBFN_ADSTYPE_GD                 0x08
    #define PBFN_ADSTYPE_FIRM               0x10
    #define PBFN_ADSTYPE_HIGHRISE           0x20
    #define PBFN_ADSTYPE_UNKNOWN            0x80

/*****************************************************************************/

    unsigned char   cAdsStreetType;
    /************** cStreetType
*************************************/
    #define PBFN_BASE_STREET                0x00
    #define PBFN_PREF_ALIAS                 0x01
    #define PBFN_OTHER_ALIAS                0x02
    #define PBFN_ABBR_ALIAS                 0x04
    #define PBFN_ALT_AT_DEL                 0x10

/*****************************************************************************/

    unsigned char   cLookupType;
    /************** cLookupType settings
*********************************/
    #define PBFN_LOOKUPTYPE_ZIPONLY         0x01
    #define PBFN_LOOKUPTYPE_ZIPOK_CTYOK     0x02
    #define PBFN_LOOKUPTYPE_ZIPOK_CTYERR    0x04
    #define PBFN_LOOKUPTYPE_CTYONLY         0x08
    #define PBFN_LOOKUPTYPE_FROM_CTYSUGG    0x10
    #define PBFN_LOOKUPTYPE_UNKNOWN         0x80

/*****************************************************************************/

    unsigned char   cFailureType;
    /************** cFailureType settings
*********************************/
    #define PBFN_FAILTYPE_ADS_ELEMENT       0x01
    #define PBFN_FAILTYPE_ZIPMOVE           0x02
    #define PBFN_FAILTYPE_AMBIGUOUS_DATA    0x04
    #define PBFN_FAILTYPE_EWS               0x08
    #define PBFN_FAILTYPE_SYSTEM_ERROR      0x10
    #define PBFN_FAILTYPE_UNKNOWN           0x80

/*****************************************************************************/

    unsigned char   cCityType;
    /************** cCityType settings
*********************************/
    #define PBFN_SINGLEZIP                  0x10
    #define PBFN_SINGLEZONE                 0x20
    #define PBFN_MULTIZONE                  0x40
    #define PBFN_PRIMARY                    0x80
```

```
    #define PBFN_NONMAILING_NAME            0x01
    #define PBFN_ABBREV_NAME                0x02
    #define PBFN_PREFERRED_NAME             0x04
    #define PBFN_PREFERRED_OVERRIDE         0x08

/*************************************************************************/

    unsigned char   cZipType;
    /************** cZipType settings
***********************************/
    #define PBFN_REGULARZIP                 0x00
    #define PBFN_PRZIP                      0x10
    #define PBFN_MILITARY                   0x20
    #define PBFN_UNIQUE                     0x40
    #define PBFN_UNIQUE_ADDRPLUS4           0x41
    #define PBFN_UNIQUE_ADDRNOPLUS4         0x42
    #define PBFN_UNIQUE_NOADDRPLUS4         0x44
    #define PBFN_UNIQUE_NOADDRNOPLUS4       0x48
    #define PBFN_DEFAULT                    0x80

/*************************************************************************/

    /***************** Components and their settings
*********************/
    #define PBFN_EXACT                      0x80
    #define PBFN_NODATA                     0x01
    #define PBFN_CORRECTED                  0x02
    #define PBFN_RTN_DATA                   0x03
    #define PBFN_INVALID                    0x04
    #define PBFN_CORRINPUT                  0x06
    #define PBFN_COMPONENT_MULTI_CHOICE     0x08
   /* additional component settings for cRange, cUnit1Range, cUnit2Range
 */
    #define PBFN_RANGE_ALPHA_MISMATCH       0x10
    #define PBFN_RANGE_OVERLAP              0x20
    /* additional component settings for cPreDir and cPostDir */
    #define PBFN_CARDINAL_FAILURE           0x10

/*************************************************************************/

    unsigned char   cAdsFirm;
    unsigned char   cAdsRange;
    unsigned char   cAdsPreDir;
    unsigned char   cAdsStreetName;
    unsigned char   cAdsSuffix;
    unsigned char   cAdsPostDir;
    unsigned char   cAdsUnit1Designator;
    unsigned char   cAdsUnit1Range;
    unsigned char   cAdsUnit2Designator;
    unsigned char   cAdsUnit2Range;
    unsigned char   cAdsPMBDesignator;
    unsigned char   cAdsPMBRange;
    unsigned char   cAdsCity;
```

```
    unsigned char   cAdsState;
    unsigned char   cAdsZip;
    unsigned char   cAdsZip4;
    unsigned char   cAdsDPBC;
    unsigned char   cAdsCarrier;
    unsigned char   cAdsLOT;
    unsigned char   cAdsLACS;
    unsigned char   cAdsUrb;
    unsigned char   cAdsChars30Flag;
    char            cAdsAliasInputInd;
 /* LabelLineDef information */
    char            cAddressLine1[255];
    short           sAddressLine1Len;
    char            cFiller18;
    char            cAddressLine2[255];
    short           sAddressLine2Len;
    char            cFiller19;
    char            cAddressLine3[255];
    short           sAddressLine3Len;
    char            cFiller20;
    char            cAddressLine4[255];
    short           sAddressLine4Len;
    char            cFiller21;
    char            cAddressLine5[255];
    short           sAddressLine5Len;
    /* ErrorDef information */
    short           sErrorNum;
    int             lPCL74Error;
    char            cMessageLevel[4];
    char            cMessageCode[128];
    char            pFileName[PBFN_FILENAME_SIZE];
    /* RtnOrigDataDef information */
    char            cOrigZip[6];
    char            cOrigZip4[5];
    char            cOrigCrRte[6];
    char            cOrigState[41];
    char            cOrigCity[255];
    char            cFiller22;
    short           sOrigCityLen;
    /* RtnFirmDef information */
    short           sRtnFirmlen;
    char            cRtnFirm[MAX_FIELD];
    /* LACSSeedDetDef information */
    USPSPBLACSDetDef USPSLACSDetail;
    char            LACSSeedHit;
    /* DPVDetailDef information */
    USPSDetailDef   USPSDPVDetail;
    char            DPVSeedHit;
    char            cPreLACSAddress[71];
    char            cDPVNSLFound;
    char            cDPVTHRWBKFound;
    char            cPreciselyIDFound;
    char            cPreciselyID[13];
```

```
    char            cFiller99[178];
    char            cFillerK[4];
} PBFNAddressDataDef, *pPBFNAddressDataDef;
```

Field Descriptions

Table 106: PBFNAddressDataDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier. **ADRS** This is an input data structure. If no return structure indicated, data will be r (overwritten) in this structure. **RRTN** This is a return data structure (output only).
cFirm	Data field containing the firm name.
sFirmLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUrb	Data field containing the urbanization name.
sUrbLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAddress1	Data field containing the first address line. If the unit and PMB fields are not requested as depen fields, those fields are concatenated to this field.
sAddress1Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur. **Note:** Placing a value of -1 in this field will "lock" the record. No processing will be perfo and the record will be returned as entered. The record will be flagged as failed and coun failed. The error code for this is 4801.
cAddress2	Data field containing the second address line. If the unit and PMB fields are not requested as de fields, those fields are concatenated to this field.
sAddress2Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.

Field	Description
cUnit1	Data field containing the first unit field in the address. This field is used for input and output.
sUnit1Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUnit2	Data field containing the second unit field in the address. This field is used for input only.
sUnit2Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cCity	Data field containing the city name. The city name returned here is the city name mandated by U regulations. Variations of the city name (i.e., full, abbreviated, non-mailing, etc.) are returned in a fields.
sCityLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cState	Data field containing the returned state field in the address data.
sStateLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cZip	Character array containing the returned ZIP Code field in the address data.
cZip4	Character array containing the returned ZIP + 4 field in the address data.
cCrRte	Character array containing the returned carrier route field in the address data.
cPMBUunit	Data field containing returned PMB/MSC field. Used for output information only.
sPMBUnitLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cInputAddressType	Data field containing the input address type for the address being passed in: **B** Blank or low-values. Defaults to standard address data. **P** Parsed address data.
cRange	Data field containing the range.
sRangeLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.

Field	Description
cPreDirectional	Data field containing the pre-directional.
sPreDirectionalLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cStreetName	Data field containing street name.
sStreetNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur. **Note:** Placing a value of -1 in this field will "lock" the record. No processing will be perfo and the record will be returned as entered. The record will be flagged as failed and coun failed. The error code for this is 4801.
cPostDirectional	Data field containing post-directional.
sPostDirectionalLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cStreetSuffix	Data field containing the street suffix.
sStreetSuffixLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUnitDesignator	Data field containing unit designator.
sUnitDesignatorLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUnitNumber	Data field containing unit number.
sUnitNumberLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUnit2Designator	Data field containing the second unit designator field.
sUnit2DesignatorLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cUnit2Number	Data field containing the second unit number.
sUnit2NumberLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.

Field	Description
cPMUnitDesignator	Data field containing PMB/MSC unit designator.
sPMUnitDesignatorLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cPMUnitNumber	Data field containing PMB/MSC unit number.
sPMUnitNumberLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur..
cAlternateStreet	Data field containing the returned alternate street name field in the address data.
sAlternateStreetLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAlternateStreetType	The field containing the returned alternate street name type field in the address data. **B** Base street. **A** Alias (other). **D** Alternate street. **P** Preferred alias. **X** Abbreviated alias.
cAltRange	Data field containing the alternate range.
sAltRangeLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAltPreDir	Data field containing the alternate pre-directional.
sAltPreDirLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAltStreetName	Data field containing the alternate street name.
sAltStreetNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAltPostDir	Data field containing the alternate post directional.
sAltPostDirLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.

Field	Description
cAltSuffix	Data field containing the alternate suffix.
sAltSuffixLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cFullCityName	Data field containing the full city name.
sFullCityNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAbbrCityName	Data field containing the abbreviated city name.
sAbbrCityNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cNonMailingCityName	Data field containing the non-mailing city name field.
sNonMailingCityNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cPreferredCityName	Preferred city name for ZIP Code. **Note:** For successfully-coded addresses, the cPreferredCityName and cPreferredState are always populated. For non-coded addresses, the cPreferredCityName and cPreferredState fields are popul the following scenarios: • ZIP Code only input (city not input, or not found) • Single ZIP Code city input (ZIP Code not input, or not found) • City/St/ZIP Code input and agree (ZIP Code is part of city) For all other non-coded scenarios, the preferred fields are blank.
sPreferredCityNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur. **Note:** The length controls the length of the cPreferredCityName. However, if the value is (0), both cPreferredCityName and cPreferredState are blank.

Field	Description
cPreferredState	Preferred state abbreviation for preferred city name. **Note:** For successfully-coded addresses, the cPreferredCityName and cPreferredState are always populated. For non-coded addresses, the cPreferredCityName and cPreferredState fields are popul the following scenarios: • ZIP Code only input (city not input, or not found) • Single ZIP Code city input (ZIP Code not input, or not found) • City/St/ZIP Code input and agree (ZIP Code is part of city) For all other non-coded scenarios, the preferred fields are blank.
cStreetType	Returned street name type field in the address data: **B** Base **X** Abbreviated **P** Preferred **A** Other alias **D** Alternate at Delivery
cDelPoint	Character array containing the returned delivery point barcode field in the address data. Formerl as the DPBC field.
cFIPSCode	Character array containing the returned five-digit FIPS code. Positions 1 and 2 contain the state Positions 3 through 5 contain the county code. Used for output information only. Formerly found i RTNINFO structure.
cCountyName	Data field containing the returned county name field in the address data.
sCountyNameLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAdvanceBarcode	Character array containing the returned 14-digit barcode consisting of the beginning frame charac Code, ZIP + 4, delivery point, check digit, and end framing character (as specified in the PBFNS structure).
cFiveDigitBarcode	Character array containing the returned five-digit barcode.
cLOTCode	Character array containing returned Line of Travel (LOT) code field. Used for output information LOT unavailable, the default value is 0000D.

Field	Description
cMatchLevel	Character defining the returned level of match flag. **F** Firm record match. **G** General delivery match. **H** Highrise match. **P** PO box match. **R** Rural route/highway contract match. **S** Street level match.
cDefaultMatch	Character defining the returned default match flag field. **Y** Carrier route, or ZIP + 4, or DPBC default values returned. **Blank** No default values returned.
cLACS	Character defining the returned LACS indicator. **L** Eligible for LACS processing. **Blank** No LACS.
cLACSRtnCode	LACS^{Link} return code. **A** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. **00** LACS^{Link} processing failed. No matching record found during LACS^{Link} processing. **09** LACS^{Link} processing matched the input address to an older highrise default addres address has been converted. However, rather than provide an imprecise address, L processing does not provide a new address. **14** LACS^{Link} processing failed. Match found during LACS^{Link} processing but conversion occur due to other USPS regulations. **92** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. Unit dropped on input.
cAutoCR	Character defining the returned Auto CRRT indicator.
cRDI	Character defining the returned Residential Delivery Indicator. **Y** Address is a residential delivery. **N** Address is a business delivery. **Blank** Failed address lookup (did not return a +4), or RDI was not active.
cFiveDigitScheme	Character array containing the returned 5-digit combined ZIP Code.

Field	Description
cCongressionalDist	Character defining the returned congressional district.
cDPVFlags	Character array containing the returned DPV indicators. • Byte 1 (DPV) N — Not a valid delivery point. The USPS cannot deliver mail to this address. Y — Delivery point validated. Primary range and secondary range (when present) are valid. Th can deliver mail to this address. S — Valid primary range. Secondary range is present but is not confirmed. The USPS can deli to this address. D — Valid primary range. Secondary range is missing. The USPS can deliver mail to this addr • Byte 2 (CMRA) Y — The address is a valid Commercial Mail Receiving Agent (CMRA). N — The address is a confirmed delivery point but is not a valid CMRA. Blank — This field is blank if the address is not a confirmed delivery point. • Byte 3 (False Positive Flag) Y — The address is not a confirmed delivery point and a positive response is received from th Positive File. N — The address is not a confirmed delivery point and a negative response is received from t Positive File. This field is blank if the address is a confirmed delivery point. Blank — The False/Positive Table was not queried.

Field	Description
cDPVFootnote	Field containing the allocated field containing the returned Delivery Point Validation (DPV) Option field. This field defines the standard footnote codes returned during Delivery Point Validation (DPV) processing. Finalist returns up to six two-byte footnote codes for each address. **A1** Input address did not match to the ZIP + 4 File. **AA** Input address matched to the ZIP + 4 File. **BB** Input address matched to DPV (all components). **CC** Input address primary number matched to DPV but secondary number did no (present but invalid). **F1** Input address matched to a military ZIP Code. **G1** Input address matched to a General Delivery address. **M1** Input address primary number missing. **M3** Input address primary number is invalid. **N1** Input address primary number matched to DPV but address is missing seco number. **P1** Input address missing PO, RR, or HC Box number. **P3** Input address PO, RR, or HC box number invalid. **PB** Input address is a P. O. Box Street Address (PBSA). **R1** Input address matched to CMRA but secondary number is not present. **R7** Input address is a Carrier Route R77x. Footnote code R7 is only assigned to that are DPV confirmed. **RR** Input address matched to CMRA. **U1** Input address matched to a unique ZIP Code.
sDPVFootnoteLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cSeasonalFlags	Character array containing the seasonal delivery indicator beginning with January. **Y** Deliver mail in this month. **N** Do not deliver mail in this month.
cError	Character array containing the returned error code field.
cProcessDate	Character array containing an encrypted process date.
cDPVNoStatFound	DPV No-Stat Table status. **Y** Found in the DPV No-Stat Table. **N** Not found in the DPV No-Stat Table. **Blank** The DPV No-Stat Table was not queried.

Field	Description
cUserKey	User key for display.
cSteLnkMatchCode	The Suite^{Link} match code. **A** Match found. **B** No match found. **C** Business name normalized to a blank value. **D** ZIP + 4 Code not recognized as a high-rise default. **E** Suite^{Link} database expired.
cSteLnkFidelityCode	The Suite^{Link} fidelity code. **1** All words in the business name match. **2** Acceptable match. One or more words in the business name were not matched, but acce criteria was still met. **3** Unacceptable match. One or more words in the business name were not matched. Acce criteria was not met.
cSteLnkRtnCode	The Suite^{Link} return code. **A** Successful Suite^{Link} match. **00** Failed Suite^{Link} match.
cDPVDNAFound	DPV Door Not Accessible (DNA) Table status indicator. **Y** Found in the DPV DNA Table. **N** Not found in the DPV DNA Table. **Blank** The DPV DNA Table was not queried.
cExtra	Data that was not used by the Finalist address parser when processing the address line(s).
sExtraLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
sSeedViolationEnc	The False Positive (Seed) Violation indicator. **1** Finalist has encountered a False Positive (Seed) violation.
sDPVKeyNCOA	The NCOA DPV Key indicator. **1** Finalist has encountered an NCOA DPV key.

Field	Description
cDPVVacantFound	DPV Vacant Table status indicator. **Y** Found in the DPV Vacant Table. **N** Not found in the DPV Vacant Table. **Blank** The DPV Vacant Table was not queried.
cDPVZip4	ZIP + 4 used by DPV.
cDPVPBSAFound	DPV PBSA Table status indicator. **Y** Found in the DPV PBSA Table. **N** Not found in the DPV PBSA Table. **Blank** The DPV PBSA Table was not queried.
cPOBoxZone	P. O. Box only delivery zone status indicator: **Y** Address ZIP Code is a P.O. Box only delivery zone. **N** Address ZIP Code is not a P.O. Box only delivery zone. **Blank** Invalid input ZIP Code. P.O. Box only delivery zone cannot be determined.
cZipValid	Indicates if the address ZIP Code is valid. **Y** Address ZIP Code is a valid USPS ZIP Code. **N** Address ZIP Code is not a valid USPS ZIP Code. **Blank** Unable to determine USPS ZIP Code for input address.
cAddressUnchanged	Indicates whether changes were made to the address: **Y** No changes, including standardization, were made to this address **N** Changes have been made to this address
DualAddressInd	Indicates if the street lines include a dual address. A dual address is an address that contains m one mailable address (for example, an address that contains both a PO BOX and a street addre **Y** Address is a dual address **N** Address is not a dual address

Field	Description
cNonDeliverableInd	Indicates if a street line includes an address that is not deliverable by the USPS. This is an addre has successfully coded to a valid entry on the USPS database, but the USPS has marked the ad as being nondeliverable: **Y** Address is a nondeliverable address **N** Address is not a nondeliverable address
cZipMove	Indicates whether an address was coded using a ZIPMOVE record: **Y** A ZIPMOVE record was used to code the address **N** A ZIPMOVE record was not used to code the address
cExceptionsInd	Indicates whether an exceptions table entry was used to process the address: **Y** A component of the address was changed due to an exceptions table entry **N** If exceptions table processing is enabled, no change was made. If exceptions table proc is not enabled, this value will always be N
sAddrMatchLevel	Level of matching achieved for the address. A combination of levels can be obtained. For examp unique ZIP Code was coded to a street level, the value returned here would have both PBFN_ADSLVL_STREET and PBFN_ADSLVL_UNIQUE_FOUND turned on. **PBFN_ADSLVL_STREET** Street level default. **PBFN_ADSLVL_SECONDARY** Secondary level match obtained. **PBFN_ADSLVL_FIRM_PRIMARY** Firm record match at primary level. **PBFN_ADSLVL_FIRM_SECONDARY** Firm record match at secondary level. **PBFN_ADSLVL_HIGHRISE_DEFAULT** High-rise default. **PBFN_ADSLVL_HIGHRISE_SECONDARY** High-rise secondary unit match. **PBFN_ADSLVL_RRHC_DEFAULT** RR or HC default. **PBFN_ADSLVL_RRHC_SECONDARY** RR or HC secondary default. **PBFN_ADSLVL_GENDEL** General delivery. **PBFN_ADSLVL_PO** PO Box. **PBFN_ADSLVL_UNIQUE_FOUND** Coded to a unique ZIP Code. **PBFN_ADSLVL_UNIQUE_DEFAULT** Coded as a default unique ZIP Code. **PBFN_ADSLVL_MIL_DEFAULT** Coded as a default military. **PBFN_ADSLVL_MIL_SECONDARY** Coded as a military secondary match.

Field	Description	
cRangeMatchInd	Level of matching achieved for the range. A combination of levels can be obtained.	
	PBFN_RNKRNGPRIME	Original primary range ranked.
	PBFN_RNKRNGSEC1	Original secondary #1 ranked.
	PBFN_RNKRNGSEC2	Original secondary #2 ranked.
	PBFN_RNKRNGSEC12COMBO	Secondary1 plus secondary2 combined for match
	PBFN_RNKRNGALPHARECOMBO	Alpha component of range was moved to seconda matching.
	PBFN_RNKRNGDASHRECOMBO	Range following the dash was moved to secondar matching.
	PBFN_RNKRNGALTERNATE	Primary range moved to secondary for alternate at matching.
	PBFN_RNKRNGSECUD	Unit Designator used for secondary rank.
cReturnLevel	Level of postal coding information returned.	
	PBFN_RTNZIP4	Returned valid ZIP + 4 Code.
	PBFN_RTNZIPCRRT	Returned valid ZIP Code and carrier route only.
	PBFN_RTNZIP	Returned valid ZIP Code only.
	PBFN_RTNUNKNOWN	Returned valid carrier route only.
cAddressLoc	Location where Finalist found the address information.	
	PBFN_ADSLOC_L1	Address found on address line 1.
	PBFN_ADSLOC_L2	Address found on address line 2.
	PBFN_ADSLOC_L1_L2	Address found on both address line 1 and line
	PBFN_ADSLOC_FIRM_L1	Unused at this time.
	PBFN_ADSLOC_FIRM_L2	Unused at this time.
	PBFN_ADSLOC_UNKNOWN	Address location unknown.

Field	Description	
cUnitLoc	Location of the matched unit Finalist returned.	
	PBFN_UNITLOC_NONE	Unit not found on input.
	PBFN_UNITLOC_L1	Unit found on line 1.
	PBFN_UNITLOC_L2	Unit found on line 2.
	PBFN_UNITLOC_L1_L2	Unit found on line 1 and line 2.
	PBFN_UNITLOC_U1	Unit found in Unit1 field.
	PBFN_UNITLOC_U2	Unit found in Unit2 field.
	PBFN_UNITLOC_U1_U2	Unit found in Unit1 and Unit2 fields.
	PBFN_UNITLOC_L1_U1	Unit found on line 1 and in Unit1 field.
	PBFN_UNITLOC_L1_U2	Unit found on line 1 and in Unit2 field.
	PBFN_UNITLOC_L2_U1	Unit found on line 2 and in Unit1 field.
	PBFN_UNITLOC_L2_U2	Unit found on line 2 and in Unit2 field.
	PBFN_UNITLOC_UNKNOWN	Unit unknown.
cFirmLoc	Location of the matched firm Finalist returned.	
	PBFN_FIRMLOC_FIRM	Firm found on firm line.
	PBFN_FIRMLOC_L1	Firm found on address line 1.
	PBFN_FIRMLOC_L2	Firm found on address line 2.
	PBFN_FIRMLOC_UNKNOWN	Firm unknown.
cPMLoc	Location of the matched Private Mail Box (PMB) or Mail Stop Code (MSC) designator Finalist re	
	PBFN_PMLOC_NONE	PMB/MSC not found on input.
	PBFN_PMLOC_L1	PMB/MSC found on address line 1.
	PBFN_PMLOC_L2	PMB/MSC found on address line 2.
	PBFN_PMLOC_L1_L2	PMB/MSC found on both address lines 1 and 2.
	PBFN_PMLOC_PM	PMB/MSC found in PMUnit field.
	PBFN_PMLOC_U1	PMB/MSC found in Unit1 field.
	PBFN_PMLOC_U2	PMB/MSC found in Unit2 field.
	PBFN_PMLOC_UNKNOWN	PMB/MSC unknown.

Field	Description
cAddressType	Type of address. **PBFN_ADSTYPE_STREET** Street address. **PBFN_ADSTYPE_PO_BOX** Post office box. **PBFN_ADSTYPE_RR** Rural route. **PBFN_ADSTYPE_HC** Highway contract. **PBFN_ADSTYPE_GD** General delivery. **PBFN_ADSTYPE_FIRM** Firm address. **PBFN_ADSTYPE_HIGHRISE** Address is a multi-dwelling building. **PBFN_ADSTYPE_UNKNOWN** Address type unknown.
cAdsStreetType	Type of street matched to the address. • PBFN_BASE_STREET — The street is a base street name. **PBFN_PREF_ALIAS** The street is a preferred alias street name. **PBFN_OTHER_ALIAS** The street is a non-preferred alias street name. **PBFN_ABBR_ALIAS** The street is an abbreviated street name. **PBFN_ALT_AT_DEL** The street is an alternate at delivery street name.
cLookupType	Type of lookup PBFNProcess performed on the last supplied address. **PBFN_LOOKUPTYPE_ZIPONLY** PBFNProcess used ZIP Code only to look up the ad because the input city name was missing. **PBFN_LOOKUPTYPE_ZIPOK_CTYOK** PBFNProcess used the ZIP Code verified against th city to look up the supplied address. All information m PBFNProcess also compared the input city name ag the city name associated with the input ZIP Code. **PBFN_LOOKUPTYPE_ZIPOK_CTYERR** PBFNProcess used the ZIP Code only to look up the s address because the city name was invalid and simi names could not be found to generate suggestions. interactive mode only. **PBFN_LOOKUPTYPE_CTYONLY** PBFNProcess used city and state information only t up supplied address (ZIP Code was not used). **PBFN_LOOKUPTYPE_FROM_CTYSUGG** PBFNProcess used the city/state/ZIP Code informat was selected from the suggestion list to look up the s address. For interactive mode only. **PBFN_LOOKUPTYPE_UNKNOWN** The lookup type cannot be defined. The PBFNProce function may not have successfully looked up the su address.

Field	Description	
cFailureType	Type of failure resulting from the PBFNProcess look up.	
	PBFN_FAILTYPE_ADS_ELEMENT	Address assignment failed due to an invalid address e
	PBFN_FAILTYPE_ZIPMOVE	Address failed due to a ZIPMOVE rule violation. The a has been moved to another city or finance area and th address could not be matched exactly in the new area
	PBFN_FAILTYPE_AMBIGUOUS_DATA	Address assignment failed due to ambiguous data in a
	PBFN_FAILTYPE_EWS	Address assignment failed due to address being found USPS Early Warning System (EWS).
	PBFN_FAILTYPE_SYSTEM_ERROR	Address assignment failed due to system error.
	PBFN_FAILTYPE_UNKNOWN	Reason for address assignment failure unknown or und
cCityType	Type of city matched to the address.	
	PBFN_SINGLEZIP	The city is a single ZIP Code city.
	PBFN_SINGLEZONE	The city is a single finance city (a city with one finance
	PBFN_MULTIZONE	The city is a multi finance city (a city with more than o finance area).
	PBFN_PRIMARY	The input city name represents a primary city name an a non-mailing city name.
	PBFN_NONMAILING_NAME	The input city name represents a non-mailing city nam
	PBFN_ABBREV_NAME	The input city is an abbreviated city name.
	PBFN_PREFERRED_NAME	The input city is a preferred city name.
	PBFN_PREFERRED_OVERRIDE	The returned city is the preferred city override name.
cZipType	Type of ZIP Code matched to the address.	
	PBFN_REGULARZIP	The ZIP Code is a standard ZIP Code.
	PBFN_PRZIP	The ZIP Code is a Puerto Rico ZIP Code.
	PBFN_MILITARY	The ZIP Code represents a military installation.
	PBFN_UNIQUE	The ZIP Code is a unique ZIP Code.
	PBFN_UNIQUE_ADDRPLUS4	Address line was matched within the unique ZIP Code ar plus4 was found.
	PBFN_UNIQUE_ADDRNOPLUS4	Address line was matched within the unique ZIP Code b plus4 was not found.
	PBFN_UNIQUE_NOADDRPLUS4	Address line was not matched within the unique ZIP Co the input plus4 was matched.
	PBFN_UNIQUE_NOADDRNOPLUS4	Address line and input plus4 was not matched within the ZIP Code. Defaults returned.
	PBFN_DEFAULT	The ZIP Code is a small town ZIP Code.

Field	Description
Component Settings	The values below are valid for the members in the Components Settings section of the PBFNAddressDataDef structure. Finalist sets each character field with one or more of these hex to indicate field status. **PBFN_EXACT** Input field was an exact match (0x80). **PBFN_NODATA** Input field was empty (0x01). **PBFN_CORRECTED** Data corrected from original. If this is the only flag settin address data just standardized (0x02). **PBFN_RTN_DATA** No input data given, but valid data returned (combinatio PBFN_NODATA and PBFN_CORRECTED) (0x03). **PBFN_INVALID** Input data was received and invalid (0x04). **PBFN_CORRINPUT** Input data invalid and corrected (combination of PBFN_CORRECTED and PBFN_INVALID) (0x06). **PBFN_COMPONENT_MULT_CHOICE** Input data resulted in ambiguous match (0x08).
cAdsFirm	Information for the firm.
cAdsRange	Information for the range. **PBFN_RANGE_ALPHA_MISMATCH** Alpha value of range was invalid. **PBFN_RANGE_OVERLAP** Range failure due to overlapping ranges found or USPS database.
cAdsPreDir	Information for the pre-directional. **PBFN_CARDINAL_FAILURE** Address failed due to a directional change error. Directionals ava for use were outside the USPS approved change rules (Cardinal Rules). For example, North can only be changed to NE, NW, or d However, a match to any directional is allowed when only one of matches found is delivery point validated.
cAdsStreetName	Information for the street name.
cAdsSuffix	Information for the suffix.
cAdsPostDir	Information for the post-directional. **PBFN_CARDINAL_FAILURE** Address failed due to a directional change error. Directionals ava for use were outside the USPS approved change rules (Cardinal Rules). For example, North can only be changed to NE, NW, or d However, a match to any directional is allowed when only one of matches found is delivery point validated.
cAdsUnit1Designator	Information for the unit 1 designator.

Field	Description	
cAdsUnit1Range	Information for the unit 1 range.	
	PBFN_RANGE_ALPHA_MISMATCH	Alpha value of range was invalid.
	PBFN_RANGE_OVERLAP	Range failure due to overlapping ranges found or USPS database.
cAdsUnit2Designator	Information for the unit 2 designator.	
cAdsUnit2Range	Information for the unit 2 range.	
	PBFN_RANGE_ALPHA_MISMATCH	Alpha value of range was invalid.
	PBFN_RANGE_OVERLAP	Range failure due to overlapping ranges found or USPS database.
cAdsPMBDesignator	Information for the Private Mail Box (PMB) or Mail Stop Code (MSC) designator.	
cAdsPMBRange	Information for the PMB or MSC range.	
cAdsCity	Information for the city. This is the city name mandated by USPS regulations. Variations of the cit (for example, full, abbreviated, non-mailing) are returned in alternate fields.	
cAdsState	Information for the state.	
cAdsZip	Information for the ZIP Code.	
cAdsZip4	Information for the ZIP + 4 Code.	
cAdsDPBC	Information for the DPBC.	
cAdsCarrier	Information for the carrier route.	
cAdsLOT	Information for the Line of Travel (eLOT) code.	
cAdsLACS	Information for the LACS.	
cAdsUrb	Information for the Urbanization field.	

Field	Description
cAdsChars30Flag	Information on the reason for the changed label line. Per USPS regulations, when a preferred Ali presented as input, CASS-certified software must return the Preferred Alias address and follow th 30-character abbreviation rule. **1** Return the Abbreviated Alias address when the following conditions exist: The length of the Preferred Alias address is greater than 30 The length of the Base address is greater than 30 The length of the Abbreviated Alias address is less than or equal to 30 **2** Return the Base address when the following conditions exist: The length of the Preferred Alias address is greater than 30 The length of the Base address is less than or equal to 30 **3** Return the Base address when the following conditions exist: The length of the Abbreviated Alias address is greater than 30 The length of the Base address is less than or equal to 30 **4** Return the Base address when the following conditions exist: The length of the Other alias address is greater than 30 The length of the Base address is less than or equal to 30
cAdsAliasInputInd	Indicates whether an alias was provided on input: **Y** Alias address present on input. **N** Alias address not present on input.
cAddressLine1	For successfully-coded addresses, output label line cAddressLine1 will be empty. For failed addr output label line cAddressLine1 contains any input firm information received.
sAddressLine1Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAddressLine2	For successfully-coded addresses, output label line cAddressLine2 contains firm information if fi information is present and not returned in cAddressLine3. For failed addresses, output label line cAddressLine2 contains any input URB information received.
sAddressLine2Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAddressLine3	For successfully-coded addresses, output label line cAddressLine3 contains the URB informatio is present. If URB is not present, output label line cAddressLine3 contains the firm information if information is present. For failed addresses, cAddressLine3 contains the information received in line 1.

Field	Description
sAddressLine3Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAddressLine4	For successfully-coded addresses, output label line cAddressLine4 contains the coded address information. For failed addresses, output label line cAddressLine 4 contains the information received address line 2.
sAddressLine4Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cAddressLine5	For successfully-coded addresses, output label line cAddressLine5 contains the city/state/ZIP Code information. For failed addresses, output label line cAddressLine5 contains the information received city/state/ZIP Code.
sAddressLine5Len	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
sErrorNum	Index into error message array.
1PCL74Error	Finalist 7.4 error codes.
cMessageLevel	Error message severity level. This is the message identifier. Values and error message type are: **0** No logging. **1** Critical messages only. **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.
cMessageCode	This is the text message for the cMessageLevel identifier.
pFileName	Source file location of the error.
cOrigZip	Character array containing the input ZIP Code.
cOrigZip4	Character array containing the input sector segment.
cOrigCrRte	Character array containing the input carrier route.
cOrigState	Character array containing the input state.

Field	Description
cOrigCity	Character array containing the input city.
sOrigCityLen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
sRtnFirmlen	If the length is 0, the data field is not populated. If the length is non-0, the field is populated up to length. If the specified length is not sufficient, truncation may occur.
cRtnfirm	Returned firm name.
USPSLACSDetail	Data structure of address type USPSPBLACSDetDef.
LACSSeedHit	Indicates if record was found in the LACS^{Link} False Positive (Seed) Table. **Y** Record found in the LACS^{Link} False Positive (Seed) Table. **N** Record not found in the LACS^{Link} False Positive (Seed) Table.
USPSDPVDetail	Data structure of address type USPSDetailDef.
DPVSeedHit	Indicates if record was found in the DPV False Positive (Seed) Table. **Y** Record found in the DPV False Positive (Seed) Table. **N** Record not found in the DPV False Positive (Seed) Table.
cPreLACSAddress	For LACS^{Link} converted addresses, this field contains the input address that was passed to LAC
cDPVNSLFound	DPV No Secure Location (NSL) Table status. **Y** Found in the DPV NSL Table. **N** Not found in the DPV NSL Table. **Blank** The DPV NSL Table was not queried.
cDPVTHRWBKFound	DPV P.O. Box Throwback Table status. **Y** Found in the DPV Throwback Table. **N** Not found in the DPV Throwback Table. **Blank** The DPV Throwback Table was not queried.

Field	Description	
cPreciselyIDFound	Identifies a PreciselyID was found for the address.	
	Y	PreciselyID was found for the full address.
	D	PreciselyID was found for the primary address (secondary information was d to find a match).
	N	PreciselyID was not found.
	Blank	PreciselyID database was not queried.
cPreciselyID	PreciselyID for the address.	

COBOL Copybook - PBFNADRS

This is an input data structure. If no return structure is indicated, the data is returned (overwritten) in this structure.

```
      ****************************************************************

         **
**
         ** (C) YYYY Precisely All Rights Reserved.              **
         **
**
         ** Name: PBFNADRS.cpy
**
         **
**
         ** Use: COBOL Copybook for PBFNAddressDataDefinition
**
         **
**
         ** Distributed Source Member for the Finalist product from
**
         ** Precisely                                             **
         **
**
        ****************************************************************

         05  PBFN-ADRS-ADDRESSDATA.
         10  FILLER                          PIC X(004) VALUE '0900'.
         10  FILLER                          PIC X(005) VALUE 'ADRS'.
         10  FILLER                          PIC X(001).
         10  PBFN-ADRS-FIRM                  PIC X(128).
         10  PBFN-ADRS-FIRM-LEN              PIC S9(04) BINARY.
         10  PBFN-ADRS-URB                   PIC X(128).
         10  PBFN-ADRS-URB-LEN               PIC S9(04) BINARY.
         10  PBFN-ADRS-ADDRESS1              PIC X(128).
```

```
           10  PBFN-ADRS-ADDRESS1-LEN           PIC S9(04) BINARY.
           10  PBFN-ADRS-ADDRESS2               PIC X(128).
           10  PBFN-ADRS-ADDRESS2-LEN           PIC S9(04) BINARY.
           10  PBFN-ADRS-UNIT1                  PIC X(020).
           10  PBFN-ADRS-UNIT1-LEN              PIC S9(04) BINARY.
           10  PBFN-ADRS-UNIT2                  PIC X(020).
           10  PBFN-ADRS-UNIT2-LEN              PIC S9(04) BINARY.
           10  PBFN-ADRS-CITY                   PIC X(128).
           10  PBFN-ADRS-CITY-LEN               PIC S9(04) BINARY.
           10  PBFN-ADRS-STATE                  PIC X(020).
           10  PBFN-ADRS-STATE-LEN              PIC S9(04) BINARY.
           10  PBFN-ADRS-ZIP                    PIC X(006).
           10  PBFN-ADRS-ZIP4                   PIC X(005).
           10  PBFN-ADRS-CRRTE                  PIC X(005).
           10  PBFN-ADRS-PMBUNIT                PIC X(016).
           10  PBFN-ADRS-PMBUNIT-LEN            PIC S9(04) BINARY.
           10  PBFN-ADRS-INPUTADDRESS-TYPE      PIC X(001).
           88  PBFN-ADRS-STANDARD                       VALUE ' '.
           88  PBFN-ADRS-PARSED                         VALUE 'P'.
           10  PBFN-ADRS-RANGE                  PIC X(011).
           10  PBFN-ADRS-RANGE-LEN              PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-PREDIRECTIONAL         PIC X(003).
           10  PBFN-ADRS-PREDIRECTIONAL-LEN     PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-STREET-NAME            PIC X(029).
           10  PBFN-ADRS-STREETNAME-LEN         PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-POSTDIRECTIONAL        PIC X(003).
           10  PBFN-ADRS-POSTDIRECTIONAL-LEN    PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-STREETSUFFIX           PIC X(005).
           10  PBFN-ADRS-STREETSUFFIX-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-UNITDESIGNATOR         PIC X(011).
           10  PBFN-ADRS-UNITDESIGNATOR-LEN     PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-UNITNUMBER             PIC X(011).
           10  PBFN-ADRS-UNITNUMBER-LEN         PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-UNIT2DESIGNATOR        PIC X(011).
           10  PBFN-ADRS-UNIT2DESIGNATOR-LEN    PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-UNIT2NUMBER            PIC X(011).
           10  PBFN-ADRS-UNIT2NUMBER-LEN        PIC S9(04) BINARY.
           10  PBFN-ADRS-PMUNITDESIGNATOR       PIC X(004).
           10  PBFN-ADRS-PMUNITDESIGNATOR-LEN   PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-PMUNITNUMBER           PIC X(011).
           10  PBFN-ADRS-PMUNITNUMBER-LEN       PIC S9(04) BINARY.
           10  PBFN-ADRS-ALTERNATESTREET        PIC X(080).
           10  PBFN-ADRS-ALTERNATESTREET-LEN    PIC S9(04) BINARY.
           10  PBFN-ADRS-ALTERNATESTREET-TYPE   PIC X(001).
```

```
          88  PBFN-ADRS-ALT-BASE                       VALUE 'B'.
          88  PBFN-ADRS-ALT-ALIAS                      VALUE 'A'.
          88  PBFN-ADRS-ALT-ALTATDEL                   VALUE 'D'.
          88  PBFN-ADRS-ALT-PREFERRED                  VALUE 'P'.
          88  PBFN-ADRS-ALT-ABBREVIATED                VALUE 'X'.
          10  PBFN-ADRS-ALTRANGE                 PIC X(011).
          10  PBFN-ADRS-ALTRANGE-LEN             PIC S9(04) BINARY.
          10  FILLER                             PIC X(001).
          10  PBFN-ADRS-ALTPREDIR                PIC X(003).
          10  PBFN-ADRS-ALTPREDIR-LEN            PIC S9(04) BINARY.
          10  FILLER                             PIC X(001).
          10  PBFN-ADRS-ALTSTREET-NAME           PIC X(029).
          10  PBFN-ADRS-ALTSTREETNAME-LEN        PIC S9(04) BINARY.
          10  FILLER                             PIC X(001).
          10  PBFN-ADRS-ALTPOSTDIR               PIC X(003).
          10  PBFN-ADRS-ALTPOSTDIR-LEN           PIC S9(04) BINARY.
          10  FILLER                             PIC X(001).
          10  PBFN-ADRS-ALTSUFFIX                PIC X(005).
          10  PBFN-ADRS-ALTSUFFIX-LEN            PIC S9(04) BINARY.
          10  PBFN-ADRS-FULLCITY-NAME            PIC X(030).
          10  PBFN-ADRS-FULLCITYNAME-LEN         PIC S9(04) BINARY.
          10  PBFN-ADRS-ABBRCITY-NAME            PIC X(014).
          10  PBFN-ADRS-ABBRCITYNAME-LEN         PIC S9(04) BINARY.
          10  PBFN-ADRS-NONMAILINGCITY-NAME      PIC X(030).
          10  PBFN-ADRS-NONMAIL-CITYNAME-LEN     PIC S9(04) BINARY.
          10  FILLER                             PIC X(001).
          10  PBFN-ADRS-PREFERREDCITY-NAME       PIC X(029).
          10  PBFN-ADRS-PREF-CITYNAME-LEN        PIC S9(04) BINARY.
          10  PBFN-ADRS-PREFERREDSTATE           PIC X(003).
          10  PBFN-ADRS-STREET-TYPE              PIC X(001).
          88  PBFN-ADRS-BASE                           VALUE 'B'.
          88  PBFN-ADRS-ALIAS                          VALUE 'A'.
          88  PBFN-ADRS-ALTATDEL                       VALUE 'D'.
          88  PBFN-ADRS-PREFERRED                      VALUE 'P'.
          88  PBFN-ADRS-ABBREVIATED                    VALUE 'X'.
          10  PBFN-ADRS-DELPOINT                 PIC X(004).
          10  PBFN-ADRS-FIPSCODE                 PIC X(006).
          10  PBFN-ADRS-COUNTY-NAME              PIC X(030).
          10  PBFN-ADRS-COUNTYNAME-LEN           PIC S9(04) BINARY.
          10  PBFN-ADRS-ADVANCEBARCODE           PIC X(015).
          10  PBFN-ADRS-FIVEDIGITBARCODE         PIC X(009).
          10  PBFN-ADRS-LOTCODE                  PIC X(006).
          10  PBFN-ADRS-MATCHLEVEL               PIC X(001).
          88  PBFN-ADRS-FIRM-MATCH                     VALUE 'F'.
          88  PBFN-ADRS-GEN-DEL-MATCH                  VALUE 'G'.
          88  PBFN-ADRS-HIGHRISE-MATCH                 VALUE 'H'.
          88  PBFN-ADRS-POBOX-MATCH                    VALUE 'P'.
          88  PBFN-ADRS-RR-HC-MATCH                    VALUE 'R'.
          88  PBFN-ADRS-STREET-MATCH                   VALUE 'S'.
          10  PBFN-ADRS-DEFAULTMATCH             PIC X(001).
          10  PBFN-ADRS-LACS                     PIC X(001).
          10  PBFN-ADRS-LACSRTNCODE              PIC X(003).
          88  PBFN-ADRS-LACS-MATCH                     VALUE 'A '.
```

```
           88  PBFN-ADRS-NO-MATCH                      VALUE '00'.
           88  PBFN-ADRS-MATCH-NO-CONV                 VALUE '14'.
           88  PBFN-ADRS-MATCH-UNIT-DROP               VALUE '92'.
           88  PBFN-ADRS-MATCH-HRD                     VALUE '09'.
           10  PBFN-ADRS-AUTOCR                  PIC X(001).
           10  PBFN-ADRS-RDI                     PIC X(001).
           10  PBFN-ADRS-FIVEDIGITSCHEME         PIC X(006).
           10  PBFN-ADRS-CONGRESSIONALDIST       PIC X(003).
           10  PBFN-ADRS-DPVFLAGS                PIC X(004).
           10  PBFN-ADRS-DPVFOOTNOTE             PIC X(020).
           10  FILLER                            PIC X(001).
           10  PBFN-ADRS-DPVFOOTNOTE-LEN         PIC S9(04) BINARY.
           10  PBFN-ADRS-SEASONALFLAGS           PIC X(013).
           10  PBFN-ADRS-ERROR                   PIC X(005).
           10  PBFN-ADRS-PROCESSDATE             PIC X(004).
           10  PBFN-ADRS-DPVNOSTATFOUND          PIC X(001).
           88  PBFN-ADRS-STAT-FOUND                    VALUE 'Y'.
           88  PBFN-ADRS-NO-STAT-FOUND                 VALUE 'N'.
           10  PBFN-ADRS-USERKEY                 PIC X(041).
           10  PBFN-ADRS-STELNKMATCHCODE         PIC X(001).
           88  PBFN-ADRS-SLK-MATCH-FOUND               VALUE 'A'.
           88  PBFN-ADRS-SLK-NO-MATCH-FOUND            VALUE 'B'.
           88  PBFN-ADRS-SLK-FIRM-BLANKED              VALUE 'C'.
           88  PBFN-ADRS-SLK-NOT-HIRISE                VALUE 'D'.
           88  PBFN-ADRS-SLK-DB-EXPIRED                VALUE 'E'.
           10  PBFN-ADRS-STELNKFIDELITYCODE      PIC X(001).
           88  PBFN-ADRS-SLK-EXACT-MATCH               VALUE '1'.
           88  PBFN-ADRS-SLK-ACCEPTABLE-MATCH          VALUE '2'.
           88  PBFN-ADRS-SLK-UNACCEPTBL-MATCH          VALUE '3'.
           10  PBFN-ADRS-STELNKRTNCODE           PIC X(003).
           88  PBFN-ADRS-SUCCESS                       VALUE 'A '.
           88  PBFN-ADRS-FAIL                          VALUE '00'.
           10  PBFN-ADRS-DPVDNAFOUND             PIC X(001).
           88  PBFN-ADRS-DNA                           VALUE 'Y'.
           88  PBFN-ADRS-NOT-DNA                       VALUE 'N'.
           10  PBFN-ADRS-EXTRA                   PIC X(256).
           10  PBFN-ADRS-EXTRA-LEN               PIC S9(04) BINARY.
           10  PBFN-ADRS-SEEDVIOLATIONENC        PIC S9(04) BINARY.
           10  PBFN-ADRS-DPVKEYNCOA              PIC S9(04) BINARY.
           10  PBFN-ADRS-DPVVACANTFOUND          PIC X(001).
           88  PBFN-ADRS-VACANT                        VALUE 'Y'.
           88  PBFN-ADRS-NOT-VACANT                    VALUE 'N'.
           10  PBFN-ADRS-DPVZIP4                 PIC X(005).
           10  PBFN-ADRS-DPVPBSAFOUND            PIC X(001).
           88  PBFN-ADRS-PBSA                          VALUE 'Y'.
           88  PBFN-ADRS-NOT-PBSA                      VALUE 'N'.
           10  PBFN-ADRS-POBOXZONE               PIC X(001).
           88  PBFN-ADRS-PO-BOX-INVALID-ZIP            VALUE ' '.
           88  PBFN-ADRS-PO-BOX-ZONE                   VALUE 'Y'.
           88  PBFN-ADRS-NOT-PO-BOX-ZONE               VALUE 'N'.
           10  PBFN-ADRS-ZIPVALID                PIC X(001).
           88  PBFN-ADRS-VALID-ZIP                     VALUE 'Y'.
           88  PBFN-ADRS-INVALID-ZIP                   VALUE 'N'.
```

```
           88  PBFN-ADRS-UNDETERMINED                  VALUE ' '.
           10  PBFN-ADRS-ADDRESSUNCHANGED        PIC X(001).
           88  PBFN-ADRS-ADDR-UNCHANGED                VALUE 'Y'.
           88  PBFN-ADRS-ADDR-CHANGED                  VALUE 'N'.
           10  PBFN-ADRS-DUALADDRESS-IND         PIC X(001).
           88  PBFN-ADRS-DUAL-ADDR                     VALUE 'Y'.
           88  PBFN-ADRS-NO-DUAL-ADDR                  VALUE 'N'.
           10  PBFN-ADRS-NONDELIVERABLE-IND      PIC X(001).
           88  PBFN-ADRS-NON-DEL                       VALUE 'Y'.
           88  PBFN-ADRS-NOT-NON-DEL                   VALUE 'N'.
           10  PBFN-ADRS-ZIPMOVE                 PIC X(001).
           88  PBFN-ADRS-ZIPMOVED                      VALUE 'Y'.
           88  PBFN-ADRS-NO-ZIPMOVED                   VALUE 'N'.
           10  PBFN-ADRS-EXCEPTIONS-IND          PIC X(001).
           88  PBFN-ADRS-MATCH-EXCP                    VALUE 'Y'.
           88  PBFN-ADRS-NO-MATCH-EXCP                 VALUE 'N'.
           10  PBFN-ADRS-ADDRMATCHLEVEL          PIC S9(04) BINARY.
           88  PBFN-ADRS-STREET                        VALUE +1.
           88  PBFN-ADRS-SECONDARY                     VALUE +2.
           88  PBFN-ADRS-FIRM-PRIM                     VALUE +4.
           88  PBFN-ADRS-FIRM-SEC                      VALUE +6.
           88  PBFN-ADRS-HR-DEFAULT                    VALUE +8.
           88  PBFN-ADRS-HR-SEC                        VALUE +16.
           88  PBFN-ADRS-RRHC-DEFAULT                  VALUE +32.
           88  PBFN-ADRS-RRHC-SEC                      VALUE +34.
           88  PBFN-ADRS-GEN-DEL                       VALUE +64.
           88  PBFN-ADRS-POB                           VALUE +128.
           88  PBFN-ADRS-UNIQ-FOUND                    VALUE +256.
           88  PBFN-ADRS-UNIQ-DEFAULT                  VALUE +512.
           88  PBFN-ADRS-MIL-DEFAULT                   VALUE +1024.
           88  PBFN-ADRS-MIL-SEC                       VALUE +1026.
           10  PBFN-ADRS-RANGEMATCH-IND          PIC X(001).
           10  PBFN-ADRS-RETURNLEVEL             PIC X(001).
           10  PBFN-ADRS-ADDRESSLOC              PIC X(001).
           10  PBFN-ADRS-UNITLOC                 PIC X(001).
           10  PBFN-ADRS-FIRMLOC                 PIC X(001).
           10  PBFN-ADRS-PMLOC                   PIC X(001).
           10  PBFN-ADRS-ADDRESS-TYPE            PIC X(001).
           10  PBFN-ADRS-ADSSTREET-TYPE          PIC X(001).
           10  PBFN-ADRS-LOOKUP-TYPE             PIC X(001).
           10  PBFN-ADRS-FAILURE-TYPE            PIC X(001).
           10  PBFN-ADRS-CITY-TYPE               PIC X(001).
           10  PBFN-ADRS-ZIP-TYPE                PIC X(001).
           10  PBFN-ADRS-ADSFIRM                 PIC X(001).
           10  PBFN-ADRS-ADSRANGE                PIC X(001).
           10  PBFN-ADRS-ADSPREDIR               PIC X(001).
           10  PBFN-ADRS-ADSSTREET-NAME          PIC X(001).
           10  PBFN-ADRS-ADSSUFFIX               PIC X(001).
           10  PBFN-ADRS-ADSPOSTDIR              PIC X(001).
           10  PBFN-ADRS-ADSUNIT1DESIGNATOR      PIC X(001).
           10  PBFN-ADRS-ADSUNIT1RANGE           PIC X(001).
           10  PBFN-ADRS-ADSUNIT2DESIGNATOR      PIC X(001).
           10  PBFN-ADRS-ADSUNIT2RANGE           PIC X(001).
```

```
           10  PBFN-ADRS-ADSPMBDESIGNATOR       PIC X(001).
           10  PBFN-ADRS-ADSPMBRANGE            PIC X(001).
           10  PBFN-ADRS-ADSCITY                PIC X(001).
           10  PBFN-ADRS-ADSSTATE               PIC X(001).
           10  PBFN-ADRS-ADSZIP                 PIC X(001).
           10  PBFN-ADRS-ADSZIP4                PIC X(001).
           10  PBFN-ADRS-ADSDPBC                PIC X(001).
           10  PBFN-ADRS-ADSCARRIER             PIC X(001).
           10  PBFN-ADRS-ADSLOT                 PIC X(001).
           10  PBFN-ADRS-ADSLACS                PIC X(001).
           10  PBFN-ADRS-ADSURB                 PIC X(001).
           10  PBFN-ADRS-ADSCHARS30-FLAG        PIC X(001).
           10  PBFN-ADRS-ADSALIASINPUT-IND      PIC X(001).
           88  PBFN-ADRS-ALIAS-INPUT                   VALUE 'Y'.
           88  PBFN-ADRS-NO-ALIAS-INPUT                VALUE 'N'.
           10  PBFN-ADRS-ADDRESSLINE1           PIC X(255).
           10  PBFN-ADRS-ADDRESSLINE1-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-ADDRESSLINE2           PIC X(255).
           10  PBFN-ADRS-ADDRESSLINE2-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-ADDRESSLINE3           PIC X(255).
           10  PBFN-ADRS-ADDRESSLINE3-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-ADDRESSLINE4           PIC X(255).
           10  PBFN-ADRS-ADDRESSLINE4-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-ADDRESSLINE5           PIC X(255).
           10  PBFN-ADRS-ADDRESSLINE5-LEN       PIC S9(04) BINARY.
           10  PBFN-ADRS-ERRORNUM               PIC S9(04) BINARY.
           10  PBFN-ADRS-PCL74ERROR             PIC S9(08) BINARY.
           10  PBFN-ADRS-MESSAGELEVEL           PIC X(004).
           10  PBFN-ADRS-MESSAGECODE            PIC X(128).
           10  PBFN-ADRS-PFILE-NAME             PIC X(256).
           10  PBFN-ADRS-ORIGZIP                PIC X(006).
           10  PBFN-ADRS-ORIGZIP4               PIC X(005).
           10  PBFN-ADRS-ORIGCRRTE              PIC X(006).
           10  PBFN-ADRS-ORIGSTATE              PIC X(041).
           10  PBFN-ADRS-ORIGCITY               PIC X(255).
           10  FILLER                           PIC X(001).
           10  PBFN-ADRS-ORIGCITY-LEN           PIC S9(04) BINARY.
           10  PBFN-ADRS-RTNFIRMLEN             PIC S9(04) BINARY.
           10  PBFN-ADRS-RTNFIRM                PIC X(255).
           COPY PBFNPSLD.
           10  PBFN-ADRS-LACSSEEDHIT            PIC X(001).
           88  PBFN-ADRS-LACS-SEED-HIT                 VALUE 'Y'.
           88  PBFN-ADRS-NO-LACS-SEED-HIT              VALUE 'N'.
           COPY PBFNPSDD.
           10  PBFN-ADRS-DPVSEEDHIT             PIC X(001).
           88  PBFN-ADRS-DPV-SEED-HIT                  VALUE 'Y'.
           88  PBFN-ADRS-NO-DPV-SEED-HIT               VALUE 'N'.
           10  PBFN-ADRS-PRELACSADDRESS         PIC X(071).
           10  PBFN-ADRS-DPVNSLFOUND            PIC X(001).
```

```
   88  PBFN-ADRS-NSL                          VALUE 'Y'.
   88  PBFN-ADRS-NOT-NSL                     VALUE 'N'.
   10  PBFN-ADRS-DPVTHRWBKFOUND         PIC X(001).
   88  PBFN-ADRS-THROWBACK                   VALUE 'Y'.
   88  PBFN-ADRS-NOT-THROWBACK               VALUE 'N'.
   10  PBFN-ADRS-PRECISELYIDFOUND       PIC X(001).
   88  PBFN-ADRS-PRCSLY-FOUND                 VALUE 'Y'.
   88  PBFN-ADRS-PRCSLY-NOT-FOUND             VALUE 'N'.
   88  PBFN-ADRS-PRCSLY-FOUND-DROPSEC         VALUE 'D'.
   88  PBFN-ADRS-PRCSLY-NO-LOOKUP             VALUE ' '.
   10  PBFN-ADRS-PRECISELYID            PIC X(013).
   10  FILLER                           PIC X(178).
   10  FILLER                           PIC X(004).
```

COBOL Copybook - PBFNRRTN

This is a return data structure (output only).

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBFNRRTN.cpy                                        **

      **                                                            **

      ** Use: COBOL Copybook for PBFNAddressDataDefinition         **

      **                                                            **

      ** Distributed Source Member for the Finalist product from   **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBFN-RRTN-ADDRESSDATA.
         10  FILLER                         PIC X(004) VALUE '0900'.
         10  FILLER                         PIC X(005) VALUE 'ADRS'.
         10  FILLER                         PIC X(001).
         10  PBFN-RRTN-FIRM                 PIC X(128).
         10  PBFN-RRTN-FIRM-LEN             PIC S9(04) BINARY.
         10  PBFN-RRTN-URB                  PIC X(128).
         10  PBFN-RRTN-URB-LEN              PIC S9(04) BINARY.
         10  PBFN-RRTN-ADDRESS1             PIC X(128).
         10  PBFN-RRTN-ADDRESS1-LEN         PIC S9(04) BINARY.
         10  PBFN-RRTN-ADDRESS2             PIC X(128).
         10  PBFN-RRTN-ADDRESS2-LEN         PIC S9(04) BINARY.
```

```
           10  PBFN-RRTN-UNIT1                  PIC X(020).
           10  PBFN-RRTN-UNIT1-LEN              PIC S9(04) BINARY.
           10  PBFN-RRTN-UNIT2                  PIC X(020).
           10  PBFN-RRTN-UNIT2-LEN              PIC S9(04) BINARY.
           10  PBFN-RRTN-CITY                   PIC X(128).
           10  PBFN-RRTN-CITY-LEN               PIC S9(04) BINARY.
           10  PBFN-RRTN-STATE                  PIC X(020).
           10  PBFN-RRTN-STATE-LEN              PIC S9(04) BINARY.
           10  PBFN-RRTN-ZIP                    PIC X(006).
           10  PBFN-RRTN-ZIP4                   PIC X(005).
           10  PBFN-RRTN-CRRTE                  PIC X(005).
           10  PBFN-RRTN-PMBUNIT                PIC X(016).
           10  PBFN-RRTN-PMBUNIT-LEN            PIC S9(04) BINARY.
           10  PBFN-RRTN-INPUTADDRESS-TYPE      PIC X(001).
             88  PBFN-RRTN-STANDARD                      VALUE ' '.
             88  PBFN-RRTN-PARSED                        VALUE 'P'.
           10  PBFN-RRTN-RANGE                  PIC X(011).
           10  PBFN-RRTN-RANGE-LEN              PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-PREDIRECTIONAL         PIC X(003).
           10  PBFN-RRTN-PREDIRECTIONAL-LEN     PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-STREET-NAME            PIC X(029).
           10  PBFN-RRTN-STREETNAME-LEN         PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-POSTDIRECTIONAL        PIC X(003).
           10  PBFN-RRTN-POSTDIRECTIONAL-LEN    PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-STREETSUFFIX           PIC X(005).
           10  PBFN-RRTN-STREETSUFFIX-LEN       PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-UNITDESIGNATOR         PIC X(011).
           10  PBFN-RRTN-UNITDESIGNATOR-LEN     PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-UNITNUMBER             PIC X(011).
           10  PBFN-RRTN-UNITNUMBER-LEN         PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-UNIT2DESIGNATOR        PIC X(011).
           10  PBFN-RRTN-UNIT2DESIGNATOR-LEN    PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-UNIT2NUMBER            PIC X(011).
           10  PBFN-RRTN-UNIT2NUMBER-LEN        PIC S9(04) BINARY.
           10  PBFN-RRTN-PMUNITDESIGNATOR       PIC X(004).
           10  PBFN-RRTN-PMUNITDESIGNATOR-LEN   PIC S9(04) BINARY.
           10  FILLER                           PIC X(001).
           10  PBFN-RRTN-PMUNITNUMBER           PIC X(011).
           10  PBFN-RRTN-PMUNITNUMBER-LEN       PIC S9(04) BINARY.
           10  PBFN-RRTN-ALTERNATESTREET        PIC X(080).
           10  PBFN-RRTN-ALTERNATESTREET-LEN    PIC S9(04) BINARY.
           10  PBFN-RRTN-ALTERNATESTREET-TYPE   PIC X(001).
             88  PBFN-RRTN-ALT-BASE                      VALUE 'B'.
             88  PBFN-RRTN-ALT-ALIAS                     VALUE 'A'.
             88  PBFN-RRTN-ALT-ALTATDEL                  VALUE 'D'.
```

```
           88  PBFN-RRTN-ALT-PREFERRED                  VALUE 'P'.
           88  PBFN-RRTN-ALT-ABBREVIATED                VALUE 'X'.
         10  PBFN-RRTN-ALTRANGE                 PIC X(011).
         10  PBFN-RRTN-ALTRANGE-LEN             PIC S9(04) BINARY.
         10  FILLER                             PIC X(001).
         10  PBFN-RRTN-ALTPREDIR                PIC X(003).
         10  PBFN-RRTN-ALTPREDIR-LEN            PIC S9(04) BINARY.
         10  FILLER                             PIC X(001).
         10  PBFN-RRTN-ALTSTREET-NAME           PIC X(029).
         10  PBFN-RRTN-ALTSTREETNAME-LEN        PIC S9(04) BINARY.
         10  FILLER                             PIC X(001).
         10  PBFN-RRTN-ALTPOSTDIR               PIC X(003).
         10  PBFN-RRTN-ALTPOSTDIR-LEN           PIC S9(04) BINARY.
         10  FILLER                             PIC X(001).
         10  PBFN-RRTN-ALTSUFFIX                PIC X(005).
         10  PBFN-RRTN-ALTSUFFIX-LEN            PIC S9(04) BINARY.
         10  PBFN-RRTN-FULLCITY-NAME            PIC X(030).
         10  PBFN-RRTN-FULLCITYNAME-LEN         PIC S9(04) BINARY.
         10  PBFN-RRTN-ABBRCITY-NAME            PIC X(014).
         10  PBFN-RRTN-ABBRCITYNAME-LEN         PIC S9(04) BINARY.
         10  PBFN-RRTN-NONMAILINGCITY-NAME      PIC X(030).
         10  PBFN-RRTN-NONMAIL-CITYNAME-LEN     PIC S9(04) BINARY.
         10  FILLER                             PIC X(001).
         10  PBFN-RRTN-PREFERREDCITY-NAME       PIC X(029).
         10  PBFN-RRTN-PREF-CITYNAME-LEN        PIC S9(04) BINARY.
         10  PBFN-RRTN-PREFERREDSTATE           PIC X(003).
         10  PBFN-RRTN-STREET-TYPE              PIC X(001).
           88  PBFN-RRTN-BASE                           VALUE 'B'.
           88  PBFN-RRTN-ALIAS                          VALUE 'A'.
           88  PBFN-RRTN-ALTATDEL                       VALUE 'D'.
           88  PBFN-RRTN-PREFERRED                      VALUE 'P'.
           88  PBFN-RRTN-ABBREVIATED                    VALUE 'X'.
         10  PBFN-RRTN-DELPOINT                 PIC X(004).
         10  PBFN-RRTN-FIPSCODE                 PIC X(006).
         10  PBFN-RRTN-COUNTY-NAME              PIC X(030).
         10  PBFN-RRTN-COUNTYNAME-LEN           PIC S9(04) BINARY.
         10  PBFN-RRTN-ADVANCEBARCODE           PIC X(015).
         10  PBFN-RRTN-FIVEDIGITBARCODE         PIC X(009).
         10  PBFN-RRTN-LOTCODE                  PIC X(006).
         10  PBFN-RRTN-MATCHLEVEL               PIC X(001).
           88  PBFN-RRTN-FIRM-MATCH                     VALUE 'F'.
           88  PBFN-RRTN-GEN-DEL-MATCH                  VALUE 'G'.
           88  PBFN-RRTN-HIGHRISE-MATCH                 VALUE 'H'.
           88  PBFN-RRTN-POBOX-MATCH                    VALUE 'P'.
           88  PBFN-RRTN-RR-HC-MATCH                    VALUE 'R'.
           88  PBFN-RRTN-STREET-MATCH                   VALUE 'S'.
         10  PBFN-RRTN-DEFAULTMATCH             PIC X(001).
         10  PBFN-RRTN-LACS                     PIC X(001).
         10  PBFN-RRTN-LACSRTNCODE              PIC X(003).
           88  PBFN-RRTN-LACS-MATCH                     VALUE 'A '.
           88  PBFN-RRTN-NO-MATCH                       VALUE '00'.
           88  PBFN-RRTN-MATCH-NO-CONV                  VALUE '14'.
           88  PBFN-RRTN-MATCH-UNIT-DROP                VALUE '92'.
```

```
           88  PBFN-RRTN-MATCH-HRD                        VALUE '09'.
         10  PBFN-RRTN-AUTOCR                  PIC X(001).
         10  PBFN-RRTN-RDI                     PIC X(001).
         10  PBFN-RRTN-FIVEDIGITSCHEME         PIC X(006).
         10  PBFN-RRTN-CONGRESSIONALDIST       PIC X(003).
         10  PBFN-RRTN-DPVFLAGS                PIC X(004).
         10  PBFN-RRTN-DPVFOOTNOTE             PIC X(020).
         10  FILLER                            PIC X(001).
         10  PBFN-RRTN-DPVFOOTNOTE-LEN         PIC S9(04) BINARY.
         10  PBFN-RRTN-SEASONALFLAGS           PIC X(013).
         10  PBFN-RRTN-ERROR                   PIC X(005).
         10  PBFN-RRTN-PROCESSDATE             PIC X(004).
         10  PBFN-RRTN-DPVNOSTATFOUND          PIC X(001).
           88  PBFN-RRTN-STAT-FOUND                      VALUE 'Y'.
           88  PBFN-RRTN-NO-STAT-FOUND                   VALUE 'N'.
         10  PBFN-RRTN-USERKEY                 PIC X(041).
         10  PBFN-RRTN-STELNKMATCHCODE         PIC X(001).
           88  PBFN-RRTN-SLK-MATCH-FOUND                 VALUE 'A'.
           88  PBFN-RRTN-SLK-NO-MATCH-FOUND              VALUE 'B'.
           88  PBFN-RRTN-SLK-FIRM-BLANKED                VALUE 'C'.
           88  PBFN-RRTN-SLK-NOT-HIRISE                  VALUE 'D'.
           88  PBFN-RRTN-SLK-DB-EXPIRED                  VALUE 'E'.
         10  PBFN-RRTN-STELNKFIDELITYCODE      PIC X(001).
           88  PBFN-RRTN-SLK-EXACT-MATCH                 VALUE '1'.
           88  PBFN-RRTN-SLK-ACCEPTABLE-MATCH            VALUE '2'.
           88  PBFN-RRTN-SLK-UNACCEPTBL-MATCH            VALUE '3'.
         10  PBFN-RRTN-STELNKRTNCODE           PIC X(003).
           88  PBFN-RRTN-SUCCESS                         VALUE 'A '.
           88  PBFN-RRTN-FAIL                            VALUE '00'.
         10  PBFN-RRTN-DPVDNAFOUND             PIC X(001).
           88  PBFN-RRTN-DNA                             VALUE 'Y'.
           88  PBFN-RRTN-NOT-DNA                         VALUE 'N'.
         10  PBFN-RRTN-EXTRA                   PIC X(256).
         10  PBFN-RRTN-EXTRA-LEN               PIC S9(04) BINARY.
         10  PBFN-RRTN-SEEDVIOLATIONENC        PIC S9(04) BINARY.
         10  PBFN-RRTN-DPVKEYNCOA              PIC S9(04) BINARY.
         10  PBFN-RRTN-DPVVACANTFOUND          PIC X(001).
           88  PBFN-RRTN-VACANT                          VALUE 'Y'.
           88  PBFN-RRTN-NOT-VACANT                      VALUE 'N'.
         10  PBFN-RRTN-DPVZIP4                 PIC X(005).
         10  PBFN-RRTN-DPVPBSAFOUND            PIC X(001).
           88  PBFN-RRTN-PBSA                            VALUE 'Y'.
           88  PBFN-RRTN-NOT-PBSA                        VALUE 'N'.
         10  PBFN-RRTN-POBOXZONE               PIC X(001).
           88  PBFN-RRTN-PO-BOX-INVALID-ZIP              VALUE ' '.
           88  PBFN-RRTN-PO-BOX-ZONE                     VALUE 'Y'.
           88  PBFN-RRTN-NOT-PO-BOX-ZONE                 VALUE 'N'.
         10  PBFN-RRTN-ZIPVALID                PIC X(001).
           88  PBFN-RRTN-VALID-ZIP                       VALUE 'Y'.
           88  PBFN-RRTN-INVALID-ZIP                     VALUE 'N'.
           88  PBFN-RRTN-UNDETERMINED                    VALUE ' '.
         10  PBFN-RRTN-ADDRESSUNCHANGED        PIC X(001).
           88  PBFN-RRTN-ADDR-UNCHANGED                  VALUE 'Y'.
```

```
           88  PBFN-RRTN-ADDR-CHANGED                    VALUE 'N'.
         10  PBFN-RRTN-DUALADDRESS-IND          PIC X(001).
           88  PBFN-RRTN-DUAL-ADDR                      VALUE 'Y'.
           88  PBFN-RRTN-NO-DUAL-ADDR                   VALUE 'N'.
         10  PBFN-RRTN-NONDELIVERABLE-IND       PIC X(001).
           88  PBFN-RRTN-NON-DEL                        VALUE 'Y'.
           88  PBFN-RRTN-NOT-NON-DEL                    VALUE 'N'.
         10  PBFN-RRTN-ZIPMOVE                  PIC X(001).
           88  PBFN-RRTN-ZIPMOVED                       VALUE 'Y'.
           88  PBFN-RRTN-NO-ZIPMOVED                    VALUE 'N'.
         10  PBFN-RRTN-EXCEPTIONS-IND           PIC X(001).
           88  PBFN-RRTN-MATCH-EXCP                     VALUE 'Y'.
           88  PBFN-RRTN-NO-MATCH-EXCP                  VALUE 'N'.
         10  PBFN-RRTN-ADDRMATCHLEVEL           PIC S9(04) BINARY.
           88  PBFN-RRTN-STREET                         VALUE +1.
           88  PBFN-RRTN-SECONDARY                      VALUE +2.
           88  PBFN-RRTN-FIRM-PRIM                      VALUE +4.
           88  PBFN-RRTN-FIRM-SEC                       VALUE +6.
           88  PBFN-RRTN-HR-DEFAULT                     VALUE +8.
           88  PBFN-RRTN-HR-SEC                         VALUE +16.
           88  PBFN-RRTN-RRHC-DEFAULT                   VALUE +32.
           88  PBFN-RRTN-RRHC-SEC                       VALUE +34.
           88  PBFN-RRTN-GEN-DEL                        VALUE +64.
           88  PBFN-RRTN-POB                            VALUE +128.
           88  PBFN-RRTN-UNIQ-FOUND                     VALUE +256.
           88  PBFN-RRTN-UNIQ-DEFAULT                   VALUE +512.
           88  PBFN-RRTN-MIL-DEFAULT                    VALUE +1024.
           88  PBFN-RRTN-MIL-SEC                        VALUE +1026.
         10  PBFN-RRTN-RANGEMATCH-IND           PIC X(001).
         10  PBFN-RRTN-RETURNLEVEL              PIC X(001).
         10  PBFN-RRTN-ADDRESSLOC               PIC X(001).
         10  PBFN-RRTN-UNITLOC                  PIC X(001).
         10  PBFN-RRTN-FIRMLOC                  PIC X(001).
         10  PBFN-RRTN-PMLOC                    PIC X(001).
         10  PBFN-RRTN-ADDRESS-TYPE             PIC X(001).
         10  PBFN-RRTN-ADSSTREET-TYPE           PIC X(001).
         10  PBFN-RRTN-LOOKUP-TYPE              PIC X(001).
         10  PBFN-RRTN-FAILURE-TYPE             PIC X(001).
         10  PBFN-RRTN-CITY-TYPE                PIC X(001).
         10  PBFN-RRTN-ZIP-TYPE                 PIC X(001).
         10  PBFN-RRTN-ADSFIRM                  PIC X(001).
         10  PBFN-RRTN-ADSRANGE                 PIC X(001).
         10  PBFN-RRTN-ADSPREDIR                PIC X(001).
         10  PBFN-RRTN-ADSSTREET-NAME           PIC X(001).
         10  PBFN-RRTN-ADSSUFFIX                PIC X(001).
         10  PBFN-RRTN-ADSPOSTDIR               PIC X(001).
         10  PBFN-RRTN-ADSUNIT1DESIGNATOR       PIC X(001).
         10  PBFN-RRTN-ADSUNIT1RANGE            PIC X(001).
         10  PBFN-RRTN-ADSUNIT2DESIGNATOR       PIC X(001).
         10  PBFN-RRTN-ADSUNIT2RANGE            PIC X(001).
         10  PBFN-RRTN-ADSPMBDESIGNATOR         PIC X(001).
         10  PBFN-RRTN-ADSPMBRANGE              PIC X(001).
         10  PBFN-RRTN-ADSCITY                  PIC X(001).
```

```
          10  PBFN-RRTN-ADSSTATE                PIC X(001).
          10  PBFN-RRTN-ADSZIP                 PIC X(001).
          10  PBFN-RRTN-ADSZIP4                PIC X(001).
          10  PBFN-RRTN-ADSDPBC                PIC X(001).
          10  PBFN-RRTN-ADSCARRIER             PIC X(001).
          10  PBFN-RRTN-ADSLOT                 PIC X(001).
          10  PBFN-RRTN-ADSLACS                PIC X(001).
          10  PBFN-RRTN-ADSURB                 PIC X(001).
          10  PBFN-RRTN-ADSCHARS30-FLAG        PIC X(001).
          10  PBFN-RRTN-ADSALIASINPUT-IND      PIC X(001).
            88  PBFN-RRTN-ALIAS-INPUT                    VALUE 'Y'.
            88  PBFN-RRTN-NO-ALIAS-INPUT                 VALUE 'N'.
          10  PBFN-RRTN-ADDRESSLINE1           PIC X(255).
          10  PBFN-RRTN-ADDRESSLINE1-LEN       PIC S9(04) BINARY.
          10  FILLER                           PIC X(001).
          10  PBFN-RRTN-ADDRESSLINE2           PIC X(255).
          10  PBFN-RRTN-ADDRESSLINE2-LEN       PIC S9(04) BINARY.
          10  FILLER                           PIC X(001).
          10  PBFN-RRTN-ADDRESSLINE3           PIC X(255).
          10  PBFN-RRTN-ADDRESSLINE3-LEN       PIC S9(04) BINARY.
          10  FILLER                           PIC X(001).
          10  PBFN-RRTN-ADDRESSLINE4           PIC X(255).
          10  PBFN-RRTN-ADDRESSLINE4-LEN       PIC S9(04) BINARY.
          10  FILLER                           PIC X(001).
          10  PBFN-RRTN-ADDRESSLINE5           PIC X(255).
          10  PBFN-RRTN-ADDRESSLINE5-LEN       PIC S9(04) BINARY.
          10  PBFN-RRTN-ERRORNUM               PIC S9(04) BINARY.
          10  PBFN-RRTN-PCL74ERROR             PIC S9(08) BINARY.
          10  PBFN-RRTN-MESSAGELEVEL           PIC X(004).
          10  PBFN-RRTN-MESSAGECODE            PIC X(128).
          10  PBFN-RRTN-PFILE-NAME             PIC X(256).
          10  PBFN-RRTN-ORIGZIP                PIC X(006).
          10  PBFN-RRTN-ORIGZIP4               PIC X(005).
          10  PBFN-RRTN-ORIGCRRTE              PIC X(006).
          10  PBFN-RRTN-ORIGSTATE              PIC X(041).
          10  PBFN-RRTN-ORIGCITY               PIC X(255).
          10  FILLER                           PIC X(001).
          10  PBFN-RRTN-ORIGCITY-LEN           PIC S9(04) BINARY.
          10  PBFN-RRTN-RTNFIRMLEN             PIC S9(04) BINARY.
          10  PBFN-RRTN-RTNFIRM                PIC X(255).
          COPY PBFNPSLD.
          10  PBFN-RRTN-LACSSEEDHIT            PIC X(001).
            88  PBFN-RRTN-LACS-SEED-HIT                  VALUE 'Y'.
            88  PBFN-RRTN-NO-LACS-SEED-HIT               VALUE 'N'.
          COPY PBFNPSDD.
          10  PBFN-RRTN-DPVSEEDHIT             PIC X(001).
            88  PBFN-RRTN-DPV-SEED-HIT                   VALUE 'Y'.
            88  PBFN-RRTN-NO-DPV-SEED-HIT                VALUE 'N'.
          10  PBFN-RRTN-PRELACSADDRESS         PIC X(071).
          10  PBFN-RRTN-DPVNSLFOUND            PIC X(001).
            88  PBFN-RRTN-NSL                            VALUE 'Y'.
            88  PBFN-RRTN-NOT-NSL                        VALUE 'N'.
          10  PBFN-RRTN-DPVTHRWBKFOUND         PIC X(001).
```

```
           88  PBFN-RRTN-THROWBACK                        VALUE 'Y'.
           88  PBFN-RRTN-NOT-THROWBACK                    VALUE 'N'.
         10  PBFN-RRTN-PRECISELYIDFOUND       PIC X(001).
           88  PBFN-RRTN-PRCSLY-FOUND                     VALUE 'Y'.
           88  PBFN-RRTN-PRCSLY-NOT-FOUND                 VALUE 'N'.
           88  PBFN-RRTN-PRCSLY-FOUND-DROPSEC             VALUE 'D'.
           88  PBFN-RRTN-PRCSLY-NO-LOOKUP                 VALUE ' '.
         10  PBFN-RRTN-PRECISELYID            PIC X(013).
         10  FILLER                           PIC X(178).
         10  FILLER                           PIC X(004).
```

PBFNDPVHdrDef

The PBFNDPVHdrDef C structure returns the Delivery Point Validation (DPV) Header record.

Syntax

```
typedef struct PBFNDPVHdrDefinition{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    USPSDPVHdrDef USPSHdr;
    char    cFiller[79];
} PBFNDPVHdrDef, *pPBFNDPVHdrDef;
```

Field Descriptions

Table 107: PBFNDPVHdrDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to TDPV.
USPSDPVHdrDef	Pointer to a data structure of address type USPSDPVHdrDef.

COBOL Copybook - PBFNTDPV

```
      ************************************************************************

      **                                                                    **
```

```
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                          **
      ** Name: PBFNTDPV.CPY                                      **
      **                                                          **
      ** Use: COBOL Copybook for PBFNDPVHdrDefinition            **
      **                                                          **
      ** Distributed Source Member for the Finalist product from  **
      ** Precisely                                               **
      **                                                          **
      ************************************************************

       05  PBFN-TDPV-DPVHDR.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'TDPV'.
         COPY PBFNPSDH.
         10  FILLER                          PIC X(079).
```

PBFNDPVStatsDef

The PBFNDPVStatsDef C structure provides processing statistics for Delivery Point Validation (DPV) processing.

Syntax

```
typedef struct PBFNDPVStatsDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    unsigned int     lDPVInError;
    unsigned int     lDPVTypeYValid;
    unsigned int     lDPVTypeSValid;
    unsigned int     lDPVTypeDValid;
    unsigned int     lCMRAValid;
    unsigned int     lCMRAInError;
    unsigned int     lAA;
    unsigned int     lA1;
    unsigned int     lM1;
    unsigned int     lM3;
    unsigned int     lP1;
```

```
    unsigned int     lP3;
    unsigned int     lBB;
    unsigned int     lRR;
    unsigned int     lCC;
    unsigned int     lN1;
    unsigned int     lR1;
    unsigned int     lStreetValid;
    unsigned int     lStCMRAPresented;
    unsigned int     lStCMRAValid;
    unsigned int     lHRValid;
    unsigned int     lHRCMRAPresented;
    unsigned int     lHRCMRAValid;
    unsigned int     lPOBoxValid;
    unsigned int     lRRHCValid;
    unsigned int     lRRHCCMRAPresented;
    unsigned int     lRRHCCMRAValid;
    unsigned int     lFirmValid;
    unsigned int     lFirmCMRAPresented;
    unsigned int     lFirmCMRAValid;
    unsigned int     lGenDelValid;
    unsigned int     lStreetPrimeFail;
    unsigned int     lHRPrimeFail;
    unsigned int     lPOBoxFail;
    unsigned int     lRRHCPrimeFail;
    unsigned int     lFirmPrimeFail;
    unsigned int     lStreetSecondaryFail;
    unsigned int     lHRSecondaryFail;
    unsigned int     lFirmSecondaryFail;
    unsigned int     lDPVNoStatFound;
    unsigned int     lU1;
    unsigned int     lF1;
    unsigned int     lG1;
    unsigned int     lDpvSeedHits;
    unsigned int     lDPVZipsOnFile;
    unsigned int     lTotalDPVProcessed;
    unsigned int     lTotalDPVwithZip4;
    unsigned int     lTotalZip4Suppressed;
    unsigned int     lDPVVacantFound;
    unsigned int     lPB;
    unsigned int     lDPVDNAFound;
    unsigned int     lR7;
    unsigned int     lDPVNSLFound;
    unsigned int     lDPVTHRWBKFound;
    char             cFiller2[66];
    char             cFiller3[2];
} PBFNDPVStatsDef, *pPBFNDPVStatsDef;
```

Field Descriptions

Table 108: PBFNDPVStatsDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to SDPV.
lDPVInError	Total number of records not delivery point validated.
lDPVTypeYValid	Total number of records delivery point validated.
lDPVTypeSValid	Total number of records containing a valid primary range but the secondary range is not confirmed.
lDPVTypeDValid	Total number of records containing a valid primary range but the secondary range is missing.
lCMRAValid	Total number of records that are confirmed Commercial Mail Receiving Agents (CMRA).
lCMRAInError	Total number of records that are not confirmed Commercial Mail Receiving Agents (CMRA).
1AA	Total number of input addresses that matched to the ZIP + 4 File (includes nondeliverable matches).
1A1	Total number of records where the input address did not match to the ZIP + 4 File.
1M1	Total number of records with the input address primary number missing.
1M3	Total number of records with the input address primary number invalid.
1P1	Total number of records with the input address missing PO, RR, or HC box number.
1P3	Total number of records in which the input address PO box, rural route, or highway contract box number is invalid.
1BB	Total number of records in which the input address matched to DPV (all components).

Field	Description
1RR	Total number of records in which the input address matched to CMRA and the PMB designator is present.
1CC	Total number of records in which the input address primary number matched to DPV but the secondary number did not match (present but invalid).
1N1	Total number of records in which the input address primary number matched to DPV but the address is missing the secondary number.
lF1	Total number of records that matched to a military ZIP Code.
lG1	Total number of records that matched to General Delivery.
1R1	Total number of records in which the input address matched to the CMRA but the PMB designator is not present.
lStreetValid	Total number of street records DPV validated.
lStCMRAPresented	Total number of street records confirmed as Commercial Mail Receiving Agents (CMRA) with a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lStCMRAValid	Total number of street records confirmed as Commercial Mail Receiving Agents (CMRA) with or without a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lHRValid	Total number of highrise records DPV validated.
lHRCMRAPresented	Total number of highrise records confirmed as Commercial Mail Receiving Agents (CMRA) with a Private Mail Box (PMB) or Mail Stop Code (MSC) designator in the input address.
lHRCMRAValid	Total number of highrise records confirmed as Commercial Mail Receiving Agents (CMRA) with or without a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lPOBoxValid	Total number of post office box records DPV validated.
lRRHCValid	Total number of rural route/highway contract records DPV validated.
lRRHCCMRAPresented	Total number of rural route/highway contract records confirmed as Commercial Mail Receiving Agents (CMRA) with a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.

Field	Description
lRRHCCMRAValid	Total number of rural route/highway contract records confirmed as Commercial Mail Receiving Agents (CMRA) with or without a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lFirmValid	Total number of firm records DPV validated.
lFirmCMRAPresented	Total number of firm records confirmed as Commercial Mail Receiving Agents (CMRA) with a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lFirmCMRAValid	Total number of firm records confirmed as Commercial Mail Receiving Agents (CMRA) with or without a Private Mail Box (PMB) or Mail Stop Code (MSC) designator present in the input address.
lGenDelValid	Total number of general delivery records DPV validated.
lStreetPrimeFail	Total number of street records DPV failed at the primary range level.
lHRPrimeFail	Total number of highrise records DPV failed at the primary range level.
lPOBoxFail	Total number of post office box records DPV failed at the primary range level.
lRRHCPrimeFail	Total number of rural route/highway contract records DPV failed at the primary range level.
lFirmPrimeFail	Total number of firm records DPV failed at the primary range level.
lStreetSecondaryFail	Total number of street records DPV failed at the secondary range level.
lHRSecondaryFail	Total number of highrise records DPV failed at the secondary range level.
lFirmSecondaryFail	Total number of firm records DPV failed at the primary range level.
lDPVNoStatFound	Total number of records found in the No-Stat Table.
lU1	Total number of records that matched to a unique ZIP Code.
lF1	Total number of records that matched to a military ZIP Code.
lG1	Total number of records that matched to a General Delivery.
lDpvSeedHits	Total number of Seed Table violations.

Field	Description
lDPVZipsOnFile	Total number of DPV ZIP Codes in file.
lTotalDPVProcessed	Total number of records processed through DPV processing.
lTotalDPVwithZip4	Total number of records with ZIP + 4 found through DPV processing.
lDPVVacantFound	Total number of records found in the Vacant Table.
lPB	Total number of records that matched to a P. O. Box Street Address (PBSA).
lDPVDNAFound	Total number of records found in the DPV Door Not Accessible (DNA) Table.
lR7	Total number of records that matched to a Carrier Route R777.
lDPVNSLFound	Total number of records found in the DPV No Secure Location (NSL) Table.
lDPVTHRWBKFound	Total number of records found in the DPV P.O. Box Throwback Table.

COBOL Copybook - PBFNSDPV

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNSDPV.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for PBFNDPVStatsDefinition             **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************
       05  PBFN-SDPV-DPVSTATS.
```

```
           10  FILLER                          PIC X(004) VALUE 'NNNN'.
           10  FILLER                          PIC X(005) VALUE 'SDPV'.
           10  FILLER                          PIC X(003).
           10  PBFN-SDPV-DPVINERROR            PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVTYPEYVALID         PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVTYPESVALID         PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVTYPEDVALID         PIC S9(08) BINARY.
           10  PBFN-SDPV-CMRAVALID             PIC S9(08) BINARY.
           10  PBFN-SDPV-CMRAINERROR           PIC S9(08) BINARY.
           10  PBFN-SDPV-AA                    PIC S9(08) BINARY.
           10  PBFN-SDPV-A1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-M1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-M3                    PIC S9(08) BINARY.
           10  PBFN-SDPV-P1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-P3                    PIC S9(08) BINARY.
           10  PBFN-SDPV-BB                    PIC S9(08) BINARY.
           10  PBFN-SDPV-RR                    PIC S9(08) BINARY.
           10  PBFN-SDPV-CC                    PIC S9(08) BINARY.
           10  PBFN-SDPV-N1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-R1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-STREETVALID           PIC S9(08) BINARY.
           10  PBFN-SDPV-STCMRAPRESENTED       PIC S9(08) BINARY.
           10  PBFN-SDPV-STCMRAVALID           PIC S9(08) BINARY.
           10  PBFN-SDPV-HRVALID               PIC S9(08) BINARY.
           10  PBFN-SDPV-HRCMRAPRESENTED       PIC S9(08) BINARY.
           10  PBFN-SDPV-HRCMRAVALID           PIC S9(08) BINARY.
           10  PBFN-SDPV-POBOXVALID            PIC S9(08) BINARY.
           10  PBFN-SDPV-RRHCVALID             PIC S9(08) BINARY.
           10  PBFN-SDPV-RRHCCMRAPRESENTED     PIC S9(08) BINARY.
           10  PBFN-SDPV-RRHCCMRAVALID         PIC S9(08) BINARY.
           10  PBFN-SDPV-FIRMVALID             PIC S9(08) BINARY.
           10  PBFN-SDPV-FIRMCMRAPRESENTED     PIC S9(08) BINARY.
           10  PBFN-SDPV-FIRMCMRAVALID         PIC S9(08) BINARY.
           10  PBFN-SDPV-GENDELVALID           PIC S9(08) BINARY.
           10  PBFN-SDPV-STREETPRIMEFAIL       PIC S9(08) BINARY.
           10  PBFN-SDPV-HRPRIMEFAIL           PIC S9(08) BINARY.
           10  PBFN-SDPV-POBOXFAIL             PIC S9(08) BINARY.
           10  PBFN-SDPV-RRHCPRIMEFAIL         PIC S9(08) BINARY.
           10  PBFN-SDPV-FIRMPRIMEFAIL         PIC S9(08) BINARY.
           10  PBFN-SDPV-STREETSECONDARYFAIL   PIC S9(08) BINARY.
           10  PBFN-SDPV-HRSECONDARYFAIL       PIC S9(08) BINARY.
           10  PBFN-SDPV-FIRMSECONDARYFAIL     PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVNOSTATFOUND        PIC S9(08) BINARY.
           10  PBFN-SDPV-U1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-F1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-G1                    PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVSEEDHITS           PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVZIPSONFILE         PIC S9(08) BINARY.
           10  PBFN-SDPV-TOTALDPVPROCESSED     PIC S9(08) BINARY.
           10  PBFN-SDPV-TOTALDPVWITHZIP4      PIC S9(08) BINARY.
           10  PBFN-SDPV-TOTALZIP4SUPPRESSED   PIC S9(08) BINARY.
           10  PBFN-SDPV-DPVVACANTFOUND        PIC S9(08) BINARY.
           10  PBFN-SDPV-PB                    PIC S9(08) BINARY.
```

```
          10  PBFN-SDPV-DPVDNAFOUND            PIC S9(08) BINARY.
          10  PBFN-SDPV-R7                     PIC S9(08) BINARY.
          10  PBFN-SDPV-DPVNSLFOUND            PIC S9(08) BINARY.
          10  PBFN-SDPV-DPVTHRWBKFOUND         PIC S9(08) BINARY.
          10  FILLER                           PIC X(066).
          10  FILLER                           PIC X(002).
```

PBFNExtendedErrorDef

The PBFNExtendedErrorDef C structure provides information on any errors you may encounter when processing with the Finalist software.

> **Note:** The PBFNAddressDataDef structure includes the PBFNExtendedErrorDef structure information. If you use the PBFNAddressDataDef structure, you do not need to pass in a PBFNExtendedErrorDef structure.

Syntax

```
typedef struct PBFNErrorHandDef
{
    char     cVersion[VERSION_LEN];
    char     cApiId[APIID_LEN];
    char     cFiller;
    short    sError;
    int      lPCL74Error;
    char     cMessageLevel[4];
    char     cMessageCode[128];
    char     pFileName[PBFN_FILENAME_SIZE];
    char     cFiller1[16];
    char     cFiller2[4];
} PBFNExtendedErrorDef, *pPBFNExtendedErrorDef;
```

Field Descriptions

Table 109: PBFNExtendedErrorDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to IERR.
sError	Index into error message array.

Field	Description
1PCL74Error	Finalist 7.4 error codes.
cMessageLevel	Error message severity level. This is the message identifier. Values and error message type are: **0** No logging. **1** Critical messages only. **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.
cMessageCode	This is the text message for the cMessageLevel identifier.
pFileName	Pointer to the source file location of the error.

COBOL Copybook - PBFNIERR

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBFNIERR.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for PBFNExtendedErrorDefinition       **

      **                                                            **

      ** Distributed Source Member for the Finalist product from   **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

       05  PBFN-IERR-EXTERROR.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'IERR'.
         10  FILLER                          PIC X(001).
```

```
          10  PBFN-IERR-ERROR                  PIC S9(04) BINARY.
          10  PBFN-IERR-PCL74ERROR             PIC S9(08) BINARY.
          10  PBFN-IERR-MESSAGELEVEL           PIC X(004).
          10  PBFN-IERR-MESSAGECODE            PIC X(128).
          10  PBFN-IERR-PFILE-NAME             PIC X(256).
          10  FILLER                           PIC X(016).
          10  FILLER                           PIC X(004).
```

PBFNIMSSetupDef

The PBFNIMSSetupDef C structure provides a place to pass in IMS PCB information. Pass this structure in the PBFNInit call.

> **Note:** While the PCB fields in the structure are char[4] fields, these fields should actually contain the addresses of the respective PCBs. These fields should be considered as a void * field. If you are not using a particular Finalist ancillary database, the field should contain NULL or 4 0x00's.

Syntax

```
typedef struct PBFNIMSSetupDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFiller1[3];
    char    cCITYFILE[4];
    char    cDATAFILE[4];
    char    cEWS[4];
    char    cELOT[4];
    char    cDPV[4];
    char    cDPVSUD[4];
    char    cLLK[4];
    char    cLLKSUD[4];
    char    cSLK[4];
    char    cRDI[4];
    char    cPBK[4];
    char    cFiller8[84];
    char    cFiller9[4];
} PBFNIMSSetupDef, *pPBFNIMSSetupDef;
```

Field Descriptions

Table 110: PBFNIMSSetupDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiID	Structure identifier to ICFG.
cCITYFILE	Address of the PCB for CBCTYST. **Note:** This field is not currently used. The CITYFILE PCB still needs to be entered in the cPSBCity field of the PBFNSetupDef structure.
cDATAFILE	Address of the PCB for CBDATA. **Note:** This field is not currently used. The DATAFILE PCB still needs to be entered in the cPSBData field of the PBFNSetupDef structure.
cEWS	Address of the PCB for CBEWS (optional).
cELOT	Address of the PCB for LTDATA (optional).
cDPV	Address of the PCB for FDPVDB, FDPVSDB, or FDPVHDB depending on which version of DPV is used (optional). **Note:** FDPVDB is for DPV Flat, FDPVSDB is for DPV Split, and FDPVHDB is for DPV Full.
cDPVSUD	Address of the PCB for FDPVSUD (optional).
cLLK	Address of the PCB for FLLKDB (optional).
cLLKSUD	Address of the PCB for FLLKSUD (optional).
cSLK	Address of the PCB for FSLKDB (optional).
cRDI	Address of the PCB for FRDIDB (optional).
cPBK	Address of the PCB for FPBKDB (optional).

COBOL Copybook - PBFNICFG

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **

      ** Name: PBFNICFG.CPY                                         **

      **                                                            **

      ** Use: COBOL Copybook for PBFNIMSSetupDefinition             **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBFN-ICFG-IMS.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'ICFG'.
         10  FILLER                          PIC X(003).
         10  PBFN-ICFG-CITYFILE              PIC X(004).
         10  PBFN-ICFG-DATAFILE              PIC X(004).
         10  PBFN-ICFG-EWS                   PIC X(004).
         10  PBFN-ICFG-ELOT                  PIC X(004).
         10  PBFN-ICFG-DPV                   PIC X(004).
         10  PBFN-ICFG-DPVSUD                PIC X(004).
         10  PBFN-ICFG-LLK                   PIC X(004).
         10  PBFN-ICFG-LLKSUD                PIC X(004).
         10  PBFN-ICFG-SLK                   PIC X(004).
         10  PBFN-ICFG-RDI                   PIC X(004).
         10  PBFN-ICFG-PBK                   PIC X(004).
         10  FILLER                          PIC X(084).
         10  FILLER                          PIC X(004).
```

PBFNInfoDef

The PBFNInfoDef C structure returns information on the environment, postal coding engine, ZIP + 4 database, and City database from the postal coding functions.

> **Note:** The PBFNInfoDef structure is deprecated from the PBFNInit call; however, the PBFNInfoDef structure is still valid for the PBFNInfo call.

Syntax

```
typedef struct PBFNInfoDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    /* engine info data members */
    char    cCassEngineVersion[FMT_VERSION_LEN];
    char    cEngineBuild[FMT_TIME_LEN];
    char    cEngineExpDate[FMT_DATE_LEN];
    /* base info data members */
    char    cZip4BaseVer[FMT_VERSION_LEN];
    char    cZip4BaseDate[FMT_DATE_LEN];
    char    cZip4BaseExpDate[FMT_DATE_LEN];
    char    cCityBaseVer[FMT_VERSION_LEN];
    char    cCityBaseDate[FMT_DATE_LEN];
    char    cCityBaseExpDate[FMT_DATE_LEN];
    char    cCZdBTimeStamp[FMT_TIME_LEN];
    char    cCZdBPlatform[INFO_LEN];
    char    cLOTBaseVer[FMT_VERSION_LEN];
    char    cLOTBaseDate[FMT_DATE_LEN];
    char    cLOTProdName[16];
    char    cLOTCfg[4];
    /* system info data members */
    char    cOSVersion[INFO_LEN];
    char    cTotalMemory[INFO_LEN];
    char    cAvailMemory[INFO_LEN];
    char    cProcessorType[INFO_LEN];
    char    cCSDVersion[INFO_LEN];
    char    cOSType[PLATFORM_NAME];
    char    cLACSProdName[31];
    char    cLACSVersion[13];
    char    cLACSBaseDate[9];
    char    cSuiteLinkProdName[31];
    char    cSuiteLinkVersion[13];
    char    cSuiteLinkBaseDate[11];
    char    cSuiteLinkBaseExpDate[11];
    char    CPUID[7];
    char    CPUID2[7];
    char    CPUID3[7];
    char    CPUID4[7];
    char    CPUID5[7];
    char    CPUID6[7];
    char    CPUID7[7];
    char    CPUID8[7];
    char    CPUID9[7];
    char    CPUID10[7];
    char    cDPVBaseDate[FMT_DATE_LEN];
    char    cEWSBaseDate[FMT_DATE_LEN];
    char    cRDIBaseDate[FMT_DATE_LEN];
    char    cDBBuildDate[FMT_DATE_LEN];
    char    cSCBuildDate[FMT_DATE_LEN];
```

```
    char    cPreciselyIDDate[FMT_DATE_LEN];
    char    cFiller1[106];
    char    cFiller2[6];
}PBFNInfoDef, *pPBFNInfoDef;
```

Field Descriptions

Table 111: PBFNInfoDef Field Descriptions

Field	Description
Version	Structure version number.
cApiId	Structure identifier to BINF.
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.
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.
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).
cZip4BaseVer	Three-digit version number for the ZIP + 4 database. This number returns without punctuation (for example, database X.XX will return as XXX).
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).
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).
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).
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).

Field	Description
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).
cCZdBTimeStamp	The date and time the city base was built. In case multiple files were created in a month, this will help you identify which file is being used by build date and time. The date and time display in a MM DD YYYY and HH:MM:SS format.
cCZdBPlatform	Describes the platform for which the city database was built. **PC** Windows platform. **UNIX** Unix platform. **MF** Mainframe platform.
cLOTBaseVer	This is the version number for the LOT database. This number is returned without punctuation (for example, 2.00 returns as 200).
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).
cLOTProdName	This is the name of the Line of Travel (LOT) database file being used in the run.
cLOTCfg	The LOT configuration in use.
cOSVersion	The version of the operating system.
cTotalMemory	Total amount of system memory when available.
cAvailMemory	Amount of available system memory when available.
cProcessorType	Specifies the type of processor in the system when available.
cCSDVersion	Operating system service pack level when available.
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).
cLACSProdName	Certified LACS^{Link} product name.
cLACSVersion	LACS^{Link} product version number.
cLACSBaseDate	LACS^{Link} database date in format yyyymmdd.

Field	Description
cSuiteLinkProdName	Certified Suite^{Link} product name.
cSuiteLinkVersion	Suite^{Link} product version number.
cSuiteLinkBaseDate	Suite^{Link} database date in format yyyymmdd.
cSuiteLinkBaseExpDate	Suite^{Link} database expiration date in format yyyymmdd.
char CPUID char CPUID6 char CPUID2 char CPUID7 char CPUID3 char CPUID8 char CPUID4 char CPUID9 char CPUID5 char CPUID10	Up to 10 CPU IDs/MAC Addresses.
cDPVBaseDate	DPV database expiration date in format mm-dd-yyyy.
cEWSBaseDate	EWS database expiration date in format mm-dd-yyyy.
cRDIBaseDate	RDI database expiration date in format mm-dd-yyyy.
cDBBuildDate	Date the Finalist database was built in format mm-dd-yyyy.
cSCBuildDate	Date the StateCut database was built in format mm-dd-yy (blank if StateCut database was not built).
cPreciselyIDDate	Date the PreciselyID database was built in format mm-dd-yyyy.

COBOL Copybook - PBFNBINF

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNBINF.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for PBFNInfoDefinition                 **
      **                                                            **
```

```
      ** Distributed Source Member for the Finalist product from     **

      ** Precisely                                                    **
      **                                                              **

      ******************************************************************

       05  PBFN-BINF-INFO.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'BINF'.
         10  PBFN-BINF-CASSENGINEVERSION     PIC X(013).
         10  PBFN-BINF-ENGINEBUILD           PIC X(025).
         10  PBFN-BINF-ENGINEEXPDATE         PIC X(011).
         10  PBFN-BINF-ZIP4BASEVER           PIC X(013).
         10  PBFN-BINF-ZIP4BASEDATE          PIC X(011).
         10  PBFN-BINF-ZIP4BASEEXPDATE       PIC X(011).
         10  PBFN-BINF-CITYBASEVER           PIC X(013).
         10  PBFN-BINF-CITYBASEDATE          PIC X(011).
         10  PBFN-BINF-CITYBASEEXPDATE       PIC X(011).
         10  PBFN-BINF-CZDBTIMESTAMP         PIC X(025).
         10  PBFN-BINF-CZDBPLATFORM          PIC X(020).
         10  PBFN-BINF-LOTBASEVER            PIC X(013).
         10  PBFN-BINF-LOTBASEDATE           PIC X(011).
         10  PBFN-BINF-LOTPROD-NAME          PIC X(016).
         10  PBFN-BINF-LOTCFG                PIC X(004).
         10  PBFN-BINF-OSVERSION             PIC X(020).
         10  PBFN-BINF-TOTALMEMORY           PIC X(020).
         10  PBFN-BINF-AVAILMEMORY           PIC X(020).
         10  PBFN-BINF-PROCESSOR-TYPE        PIC X(020).
         10  PBFN-BINF-CSDVERSION            PIC X(020).
         10  PBFN-BINF-OS-TYPE               PIC X(020).
         10  PBFN-BINF-LACSPROD-NAME         PIC X(031).
         10  PBFN-BINF-LACSVERSION           PIC X(013).
         10  PBFN-BINF-LACSBASEDATE          PIC X(009).
         10  PBFN-BINF-SUITELINKPROD-NAME    PIC X(031).
         10  PBFN-BINF-SUITELINKVERSION      PIC X(013).
         10  PBFN-BINF-SUITELINKBASEDATE     PIC X(011).
         10  PBFN-BINF-SUITELINKBASEEXPDATE  PIC X(011).
         10  PBFN-BINF-CPUID                 PIC X(007).
         10  PBFN-BINF-CPUID2                PIC X(007).
         10  PBFN-BINF-CPUID3                PIC X(007).
         10  PBFN-BINF-CPUID4                PIC X(007).
         10  PBFN-BINF-CPUID5                PIC X(007).
         10  PBFN-BINF-CPUID6                PIC X(007).
         10  PBFN-BINF-CPUID7                PIC X(007).
         10  PBFN-BINF-CPUID8                PIC X(007).
         10  PBFN-BINF-CPUID9                PIC X(007).
         10  PBFN-BINF-CPUID10               PIC X(007).
         10  PBFN-BINF-DPVBASEDATE           PIC X(011).
         10  PBFN-BINF-EWSBASEDATE           PIC X(011).
         10  PBFN-BINF-RDIBASEDATE           PIC X(011).
         10  PBFN-BINF-DBBUILDDATE           PIC X(011).
```

```
          10  PBFN-BINF-SCBUILDDATE            PIC X(011).
          10  PBFN-BINF-PRECISELYIDDATE        PIC X(011).
          10  FILLER                           PIC X(106).
          10  FILLER                           PIC X(006).
```

PBFNLACSSeedHdrDef

Use the PBFNLACSSeedHdrDef structure on a PBFNStats call to return LACS^{Link} False Positive violation header information.

Syntax

```
typedef struct PBFNLACSSeedHdrDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    unsigned int  lTotalSeedHits;
    USPSPBLACSHdrDef USPSHdr;
} PBFNLACSSeedHdrDef, *pPBFNLACSSeedHdrDef;
```

Field Descriptions

Table 112: PBFNLACSSeedHdrDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier is YDRS.
cTotalSeedHits	Total number of records that hit seed records. For normal processing, this will be 1. NCOA clients may see a count greater than 1.
USPSPBLACSHdrDef	Pointer to a data structure of address type USPSPBLACSHdrDef.

COBOL Copybook - PBFNYDRS

```
      ****************************************************************

      **                                                            **
```

```
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                          **
      ** Name: PBFNYDRS.CPY                                      **
      **                                                          **
      ** Use: COBOL Copybook for PBFNLACSSeedHdrDefinition       **
      **                                                          **
      ** Distributed Source Member for the Finalist product from  **
      ** Precisely                                               **
      **                                                          **
      ************************************************************

       05  PBFN-YDRS-LACS.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'YDRS'.
         10  FILLER                          PIC X(003).
         10  PBFN-YDRS-TOTALSEEDHITS         PIC S9(08) BINARY.
         COPY PBFNPSLH.
```

Related APIs

PBFNStats

PBFNRDataDef

The PBFNRDataDef C structure defines the available options for the Finalist Batch Report.

Syntax

```
typedef struct PBFNRDataDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    OverWrite[ON_OFF_FLAG];
    /* File Name of client input file containing address data */
    char    ListFileName[PBFN_FILENAME_SIZE];
    char    ListProcessorName[PBFN_FILENAME_SIZE];
    char    cFiller;
    short   ListNumber;
    char    Title[REPORT_TITLE_NAME];
```

```
    char    cFiller2[4];
} PBFNReportData, *pPBFNRDataDef;
```

Field Descriptions

Table 113: PBFNRDataDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to RDAT.
OverWrite	This field determines whether Finalist overwrites the current Finalist Batch Report file. **OFF** Append to the current Finalist Batch Report file. Do not overwrite the current report. **ON** Overwrite the current Finalist Batch Report file. **Blank** Defaults to OFF.
ListFileName	This field contains your input file name.
ListProcessorName	This field contains the name of the list processor. This is usually your company's name.
ListNumber	This field contains the list number.
Title	This field contains your specified report title.

COBOL Copybook - PBFNRDAT

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNRDAT.CPY                                         **
      **                                                            **
```

```
      ** Use: COBOL Copybook for PBFNRDataDefinition                  **
      **                                                              **
      ** Distributed Source Member for the Finalist product from     **
      ** Precisely                                                     **
      **                                                              **
      ******************************************************************

       05  PBFN-RDAT-RDATA.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'RDAT'.
         10  PBFN-RDAT-OVERWRITE             PIC X(004).
           88  PBFN-RDAT-OVERWRITE-RPT                 VALUE 'ON'.
           88  PBFN-RDAT-NO-OVERWRITE-RPT              VALUE 'OFF'.
         10  PBFN-RDAT-LISTFILE-NAME         PIC X(256).
         10  PBFN-RDAT-LISTPROCESSOR-NAME    PIC X(256).
         10  FILLER                          PIC X(001).
         10  PBFN-RDAT-LISTNUMBER            PIC S9(04) BINARY.
         10  PBFN-RDAT-TITLE                 PIC X(060).
         10  FILLER                          PIC X(004).
```

PBFNRtnLACSStatsDef

You can pass the PBFNRtnLACSStatsDef structure on the PBFNStats or PBFNTerminate call to return LACS^{Link} processing statistics.

Syntax

```
typedef struct PBFNRtnLACSStatsDefinition
{
    char             cVersion[VERSION_LEN];
    char             cApiId[APIID_LEN];
    char             cFiller[3];
    unsigned  int    lTotalProcessed;
    unsigned  int    lTotalAMatches;
    unsigned  int    lTotal00Matches;
    unsigned  int    lTotal14Matches;
    unsigned  int    lTotal92Matches;
    unsigned  int    lTotal09Matches;
    char             cLACSProdName[LACS_PROD_NAME];
    char             cLACSVersion[LACS_VER_NAME];
    char             cLACSBaseDate[LACS_BASE_DATE];
    char             cFiller1[7];
} PBFNRtnLACSStatsDef, *pPBFNRtnLACSStatsDef;
```

Field Descriptions

Table 114: PBFNRtnLACSStatsDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier is PRST.
lTotalProcessed	Count of the number of records submitted for LACS^{Link} processing.
lTotalAMatches	Total number of records for LACS^{Link} return code A indicating LACS^{Link} record match.
lTotal00Matches	Total number of records for LACS^{Link} return code 00 indicating no LACS^{Link} record match.
lTotal14Matches	Total number of records for LACS^{Link} return code 14 indicating match found on LACS^{Link}. Conversion did not occur due to USPS regulations.
lTotal92Matches	Total number of records for LACS^{Link} return code 92 indicating match with unit number dropped on input.
lTotal09Matches	Total number of records for LACS^{Link} return code 09 indicating LACS^{Link} processing matched the input address to an older highrise default address. The address has been converted. However, rather than provide an imprecise address, LACS^{Link} processing does not provide a new address.
cLACSProdName	LACS^{Link} product name.
cLACSVersion	LACS^{Link} product version number.
cLACSBaseDate	LACS^{Link} database date in the format yyyymmdd.

COBOL Copybook - PBFNPRST

```
        **********************************************************************
        **                                                                  **
```

```
      ** (C) YYYY Precisely All Rights Reserved.              **
      **                                                        **
      ** Name: PBFNPRST.CPY                                    **
      **                                                        **
      ** Use: COBOL Copybook for PBFNRtnLACSStatsDefinition    **
      **                                                        **
      ** Distributed Source Member for the Finalist product from **
      ** Precisely                                             **
      **                                                        **
      ************************************************************

       05  PBFN-PRST-RTNLACSSTATS.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'PRST'.
         10  FILLER                          PIC X(003).
         10  PBFN-PRST-TOTALPROCESSED        PIC S9(08) BINARY.
         10  PBFN-PRST-TOTALAMATCHES         PIC S9(08) BINARY.
         10  PBFN-PRST-TOTAL00MATCHES        PIC S9(08) BINARY.
         10  PBFN-PRST-TOTAL14MATCHES        PIC S9(08) BINARY.
         10  PBFN-PRST-TOTAL92MATCHES        PIC S9(08) BINARY.
         10  PBFN-PRST-TOTAL09MATCHES        PIC S9(08) BINARY.
         10  PBFN-PRST-LACSPROD-NAME         PIC X(031).
         10  PBFN-PRST-LACSVERSION           PIC X(013).
         10  PBFN-PRST-LACSBASEDATE          PIC X(009).
         10  FILLER                          PIC X(007).
```

Related APIs

PBFNStats

PBFNTerminate

PBFNRtnSuiteLinkStatsDef

The PBFNRtnSuiteLinkStatsDef C structure returns Suite^{Link} processing statistics.

Syntax

```
typedef struct PBFNRtnSuiteLinkStatsDefinition
{
    char             cVersion[VERSION_LEN];
    char             cApiId[APIID_LEN];
    char             cFiller[3];
    unsigned  int    lTotalProcessed;
    unsigned  int    lTotalCorrectedSuites;
    unsigned  int    lTotalSuccessfulMatches;
    unsigned  int    lTotalA1Matches;
    unsigned  int    lTotalA2Matches;
    unsigned  int    lTotalA3Matches;
    unsigned  int    lTotalBMatches;
    unsigned  int    lTotalCMatches;
    unsigned  int    lTotalDMatches;
    unsigned  int    lTotalEMatches;
    char             cFiller2[4];
} PBFNRtnSuiteLinkStatsDef, *pPBFNRtnSuiteLinkStatsDef;
```

Field Descriptions

Table 115: PBFNRtnSuiteLinkStatsDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to SSLK.
lTotalProcessed	Count of the number of records submitted for Suite^{Link} processing.
lTotalCorrectedSuites	Total number of records successfully matched with suite corrections during Suite^{Link} processing.
lTotalSuccessfulMatches	Total number of records successfully matched during Suite^{Link} processing.

Field	Description
lTotalA1Matches	Total number of records successfully matched with all words in the business name matched during Suite[Link] processing.
lTotalA2Matches	Total number of records successfully matched with one word in the business name not matched during Suite[Link] processing.
lTotalA3Matches	Total number of records with an unacceptable match because multiple words in the business name did not match during Suite[Link] processing.
lTotalBMatches	Total number of records with no match found during Suite[Link] processing.
lTotalCMatches	Total number of records with the business name normalized to a blank value during Suite[Link] processing.
lTotalDMatches	Total number of records with a ZIP + 4 Code not recognized as a high-rise default during Suite[Link] processing.
lTotalEMatches	Total number of records failed because the Suite[Link] database has expired.

COBOL Copybook - PBFNSSLK

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNSSLK.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for PBFNRtnSuiteLinkStatsDefinition    **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************
```

```
       05  PBFN-SSLK-RTNSTELNKSTATS.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'SSLK'.
         10  FILLER                          PIC X(003).
         10  PBFN-SSLK-TOTALPROCESSED        PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALCORRECTEDSUITES  PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALSUCCESSMATCHES   PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALA1MATCHES        PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALA2MATCHES        PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALA3MATCHES        PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALBMATCHES         PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALCMATCHES         PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALDMATCHES         PIC S9(08) BINARY.
         10  PBFN-SSLK-TOTALEMATCHES         PIC S9(08) BINARY.
         10  FILLER                          PIC X(004).
```

Related APIs

PBFNStats

PBFNTerminate

PBFNSetupDef

The PBFNSetupDef C structure defines the system configuration.

Syntax

```
typedef struct PBFNSetupDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cCityFileName[PBFN_FILENAME_SIZE];
    char    cZip4FileName[PBFN_FILENAME_SIZE];
    char    cConfigFileName[PBFN_FILENAME_SIZE];
    char    cLoadSetup[LOAD_NOLOAD_FLAG];
    char    cCASSFlag[ON_OFF_FLAG];
    char    cProcessUnassigned[ON_OFF_FLAG];
    char    cAssignCR[ON_OFF_FLAG];
    char    cAssignAbbrevCity[ON_OFF_FLAG];
    char    cAssignLOT[ON_OFF_FLAG];
    char    cRetDPBC[ON_OFF_FLAG];
    char    cCacheSize[20];
    char    cBegFrame[2];
    char    cEndFrame[2];
    char    cStandardCase[2];
    char    cDualAddrSwt[ON_OFF_FLAG];
```

```
    char    cRetAliStName[ON_OFF_FLAG];
    char    cPaddedStringData[ON_OFF_FLAG];
    char    cUppercaseIn[ON_OFF_FLAG];
    char    cRemoveNoiseChar[ON_OFF_FLAG];
    char    cProcessFirms[ON_OFF_FLAG];
    char    cRetInputFirm[ON_OFF_FLAG];
    char    cAllStreetMatching[ON_OFF_FLAG];
    char    cSuggestionCount[4];
    char    cMaxExcpTblEntries[5];
    char    cReserved1[60];
    char    cStatFileNameSetup[PBFN_FILENAME_SIZE];
    char    c3553FileNameSetup[PBFN_FILENAME_SIZE];
    char    cCASSCompName[COMPANY_NAME];
    char    cCASSProdName[COMPANY_NAME];
    char    cCASSProdVer[VERSION_NAME];
    char    cZ4ChngCertCompName[COMPANY_NAME];
    char    cZ4ChngProdName[COMPANY_NAME];
    char    cZ4ChngProdVer[VERSION_NAME];
    char    cLOTCertCompName[COMPANY_NAME];
    char    cLOTCertProdName[COMPANY_NAME];
    char    cLOTCertProdVer[VERSION_NAME];
    char    cMailerName[30];
    char    cMailerAddress[30];
    char    cMailerCityLine[30];
    char    cCASSConfiguration[4];
    char    cBatchRptOpt[ON_OFF_FLAG];
    char    c3553RptOpt[ON_OFF_FLAG];
    char    cReserved3[30];
    char    cRptFileName[PBFN_FILENAME_SIZE];
    char    cRptTitle[REPORT_TITLE_NAME];
    char    cAddrDtlRptType[ON_OFF_FLAG];
    char    cAddrDtlRptIsol[ON_OFF_FLAG];
    char    cReservedRS;
    int     lAddrDtlRptIsolPageLen;
    int     lAddrDtlRptIsolMaxRec;
    int     lAddrDtlRptIsolNthRec;
    char    cAddrDtlRptSugg[ON_OFF_FLAG];
    char    cAddrDtlRptInfo[ON_OFF_FLAG];
    char    cReserved4[30];
    char    cLogFileName[PBFN_FILENAME_SIZE];
    char    cLogLevel[20];
    char    cSoftwareKey[PBFN_SOFTWAREKEY_SIZE];
    char    cExceptionTblFileName[PBFN_FILENAME_SIZE];
    char    cEWSFileName[PBFN_FILENAME_SIZE];
    char    cLOTFileName[PBFN_FILENAME_SIZE];
    char    cDPVFilePath[PBFN_FILENAME_SIZE];
    char    cRDIFilePath[PBFN_FILENAME_SIZE];
    char    cLACSLinkFilePath[PBFN_FILENAME_SIZE];
    char    cSuiteLinkFilePath[PBFN_FILENAME_SIZE];
    char    cDPVKeyName[PBFN_FILENAME_SIZE];
    char    cLACSLinkKey[PBFN_FILENAME_SIZE];
    char    cOSSelected[PLATFORM_NAME];
    char    cAssignDPV[ON_OFF_FLAG];
```

```
    char    cAssignDPVTie[ON_OFF_FLAG];
    char    cAssignDPVNoStat[ON_OFF_FLAG];
    char    cAssignRDI[ON_OFF_FLAG];
    char    cAssignEWS[ON_OFF_FLAG];
    char    cAssignCMRA[ON_OFF_FLAG];
    char    cAssignLACSLink[ON_OFF_FLAG];
    char    cAssignSuiteLink[ON_OFF_FLAG];
    char    cDPVBufSize[4];
    char    cMailerAddress2[30];
    char    cMailerAddress3[30];
    char    cMailerAddress4[30];
    char    cFiller2[33];
    char    cPSBData[4];
    char    cPSBCity[4];
    char    cSLKOverride[4];
    char    cSuiteLinkSmallMem[ON_OFF_FLAG];
    char    cSuiteLinkShutdown[2];
    char    cFiller3[24];
    char    cLACSLinkProcessing[2];
    char    cDPVShutdownIndicator[2];
    char    cAssignDPVVacant[ON_OFF_FLAG];
    char    cRetSLKinputSecdry[ON_OFF_FLAG];
    char    cAssignDPVPBSA[ON_OFF_FLAG];
    char    cR777Deliverable[ON_OFF_FLAG];
    char    cConvertSecPMB[ON_OFF_FLAG];
    /* ErrorDef information */
    char    cFiller4;
    short   sErrorNum;
    char    cFiller5[2];
    int     lPCL74Error;
    char    cMessageLevel[4];
    char    cMessageCode[128];
    char    pFileName[PBFN_FILENAME_SIZE];
    /* InfoDef information */
    /* engine info data members */
    char    cCassEngineVersion[FMT_VERSION_LEN];
    char    cEngineBuild[FMT_TIME_LEN];
    char    cEngineExpDate[FMT_DATE_LEN];
    /* base info data members */
    char    cZip4BaseVer[FMT_VERSION_LEN];
    char    cZip4BaseDate[FMT_DATE_LEN];
    char    cZip4BaseExpDate[FMT_DATE_LEN];
    char    cCityBaseVer[FMT_VERSION_LEN];
    char    cCityBaseDate[FMT_DATE_LEN];
    char    cCityBaseExpDate[FMT_DATE_LEN];
    char    cCZdBTimeStamp[FMT_TIME_LEN];
    char    cCZdBPlatform[INFO_LEN];
    char    cLOTBaseVer[FMT_VERSION_LEN];
    char    cLOTBaseDate[FMT_DATE_LEN];
    char    cLOTProdName[16];
    char    cLOTCfg[4];
    /* system info data members */
    char    cOSVersion[INFO_LEN];
```

```
    char    cTotalMemory[INFO_LEN];
    char    cAvailMemory[INFO_LEN];
    char    cProcessorType[INFO_LEN];
    char    cCSDVersion[INFO_LEN];
    char    cOSType[PLATFORM_NAME];
    /* LACSLink Stuff */
    char    cLACSProdName[31];
    char    cLACSVersion[13];
    char    cLACSBaseDate[9];
    /* SuiteLink Stuff */
    char    cSuiteLinkProdName[31];
    char    cSuiteLinkVersion[13];
    char    cSuiteLinkBaseDate[11];
    char    cSuiteLinkBaseExpDate[11];
    char    CPUID[7];
    char    CPUID2[7];
    char    CPUID3[7];
    char    CPUID4[7];
    char    CPUID5[7];
    char    CPUID6[7];
    char    CPUID7[7];
    char    CPUID8[7];
    char    CPUID9[7];
    char    CPUID10[7];
    char    cDPVBaseDate[FMT_DATE_LEN];
    char    cEWSBaseDate[FMT_DATE_LEN];
    char    cRDIBaseDate[FMT_DATE_LEN];
    char    cDBBuildDate[FMT_DATE_LEN];
    char    cSCBuildDate[FMT_DATE_LEN];
    int     iInitRtnCode;
    char    cAssignDPVDNA[ON_OFF_FLAG];
    char    cAssignDPVNSL[ON_OFF_FLAG];
    char    cAssignDPVTHRWBK[ON_OFF_FLAG];
    char    cAssignPreciselyID[ON_OFF_FLAG];
    char    cPreciselyIDDate[FMT_DATE_LEN];
    char    cFiller7[136];
    char    cFiller8[5];
} PBFNSetupDef, *pPBFNSetupDef;
```

Field Descriptions

Table 116: PBFNSetupDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to GCFG.

Field	Description
cCityFileName	City Database file name.
cZip4FileName	ZIP + 4 Database file name.
cConfigFileName	Path to and file name of the Configuration File.
cLoadSetup	Identifies whether or not to load the setup structure from the configuration file. **LOAD** Load the setup structure from the configuration file specified by cConfigFileName. **NOLOAD** Do not load the setup structure from the configuration file. **Blank** Defaults to LOAD.
cCASSFlag	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^{Link}, Suite^{Link}, CASS configuration, Carrier Route (CR), and Delivery Point (DPBC). For DPV, the Flat file option is used when no other DPV option is specified. **OFF** Process all addresses in non-CASS mode. **ON** Process all address in CASS mode. **Blank** Defaults to ON. **Note:** If CASS Flag = ON and a conflicting option is encountered (Configuration, Assign CR, Return DPBC, LACS^{Link}=OFF, Suite^{Link}=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 will be similar to: Warning Message; CASS forced off: CASS Configuration, Return DPBC, Assign CR, Assign SuiteLink, Assign LACSLink, Assign DPV

Field	Description
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. **OFF** Do not process only previously unassigned addresses. Process all supplied addresses. **ON** Process only previously unassigned or expired addresses. If an encrypted address assignment date in the PBFNAddressDataDef structure is passed to Finalist and that assignment date meets USPS regulations for valid address records, Finalist will not process that record but will include 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. **XXX** 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. **Blank** Defaults to OFF.
cAssignCR	Indicates whether Finalist assigns carrier route codes to your file. **OFF** Do not assign carrier route codes. **ON** Assign carrier route codes. **Blank** Defaults to ON.
cAssignAbbrevCity	Indicates whether Finalist returns abbreviated city names. **OFF** Do not return abbreviated city names. **ON** Return abbreviated city names. **Blank** Defaults to OFF.
cAssignLOT	Indicates whether Finalist assigns Line of Travel (LOT) codes. **OFF** Do not assign Line of Travel (LOT) codes. **ON** Assign Line of Travel (LOT) codes. **Blank** Defaults to OFF.
cRetDPBC	Indicates whether Finalist returns delivery point barcodes. **OFF** Do not return delivery point barcodes. **ON** Return delivery point barcodes. **Blank** Defaults to ON.

Field	Description
cCacheSize	Indicates whether Finalist should use internal caching. **OFF** Turn off internal caching for Finalist. **XX** Replace "XX" with 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.
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 (!).
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 (!).
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. **U** Return addresses in all upper case. **L** Return addresses in all lower case. **M** Return addresses in upper and lower case. **Blank** Defaults to U.

Field	Description
cDualAddrSwt	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. **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. **Blank** Defaults to ACZ. • 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 1. PO Box 2. Firm 3. Highrise 4. Street 5. Rural Route 6. General Delivery
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. • 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.

Field	Description
	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. **Note:** Other alias street names are returned only if specified on input.
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. **OFF** Null terminated strings used. **ON** Blank filled fields used. **Blank** Defaults to OFF.
cUppercaseIn	Indicates whether your input data is in an all uppercase format. **OFF** Input data may not be in an all uppercase format. Finalist will convert your input data to an all uppercase format. **ON** All input data is in an all uppercase format. To improve performance, Finalist will not perform the all uppercase format conversion. **Blank** Defaults to OFF. **Note:** Mainframe data tends to be all uppercase.

Field	Description
cRemoveNoiseChar	Indicates whether Finalist needs to perform the routine that removes noise characters (unnecessary punctuation and blanks). **OFF** All unnecessary punctuation and blanks have been removed from input data. Bypass the Finalist routine that removes noise characters. Finalist will bypass the routine that checks each line for "noise characters" to improve performance. **ON** Input data may contain unnecessary punctuation and blanks. Finalist will perform the routine that removes noise characters. **Blank** Defaults to ON.
cProcessFirms	Determines whether Finalist performs firm processing. **OFF** Do not processes 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. **ON** Finalist performs normal CASS processing for firms. **Blank** Defaults to ON.
cRetInputFirm	This field determines whether Finalist returns the input firm. **OFF** Do not return the input firm. **ON** Return the input firm. **Blank** Defaults to OFF.
cAllStreetMatching	Determines whether Finalist performs All Street Matching (ASM) processing. All Street Matching (ASM) applies 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. **OFF** Do not perform ASM processing. **ON** Perform ASM processing. **Blank** Defaults to OFF.
cSuggestionCount	This field contains the maximum number of returned suggestions per address based on the specified suggestion criteria.
cMaxExcpTblEntries	This field determines the maximum number of exception table entries. The default value is 1000.

Field	Description
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 pbfn.cfg. If you do not enter the output file name in the pbfn.cfg File, the default file name batch.txt will be used.
c3553FileNameSetup	Output file name for the USPS Form 3553 (CASS Summary Report). This field contains the file name found in the pbfn.cfg file. If you do not enter the output file name in the pbfn.cfg File, the default file name cass3553.txt will be used.
cCASSCompName	Name of the CASS-certified company. This information displays in section A box 1 on the USPS Form 3553 (CASS Summary Report).
cCASSProdName	Name of the CASS-certified product. This information displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
cCASSProdVer	Version number for the CASS-certified product. This information displays in section A box 2 on the USPS Form 3553 (CASS Summary Report).
cZ4ChngCertCompName	Z4Change-certified company name.
cZ4ChngProdName	Z4Change product name.
cZ4ChngProdVer	Z4Change product version number.
cLOTCertCompName	Line of Travel (LOT)-certified company name. This information displays in section A box 7 on the USPS Form 3553 (CASS Summary Report).
cLOTCertProdName	Line of Travel (LOT)-certified-product name. This information displays in section A box 8 on the USPS Form 3553 (CASS Summary Report).
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).
cMailerName	CASS-certified mailer name.
cMailerAddress	CASS-certified mailer address.
cMailerCityLine	CASS-certified mailer city line.

Field	Description
cCASSConfiguration	CASS configuration value. This information displays in section A box 3 on the USPS Form 3553 (CASS Summary Report). Finalist certifies under the value AAA. Any other value represents a self-certification of Finalist that is specific to a user environment. This value is used with cCASSFlag. **AAA** Default if cCASSFlag is set to ON. **Blank** Default if cCASSFlag is set to OFF. Any other three-character value represents a user-certification for CASS processing. **Note:** This value does not affect CASS certification, only what is presented on the USPS Form 3553 (CASS Summary Report).
cBatchRptOpt	Indicates whether to print the Finalist Batch Report. **OFF** Do not print the Finalist Batch Report. **ON** Print the Finalist Batch Report. **Blank** Defaults to OFF.
c3553RptOpt	Indicates whether to print the USPS Form 3553 (CASS Summary Report). **OFF** Do not print the USPS Form 3553 (CASS Summary Report). **ON** Print the USPS Form 3553 (CASS Summary Report). **Blank** Defaults to OFF.
cRptFileName	Address Detail Report output file name. The default file name is rpt.txt.
cRptTitle	Report title for the Address Detail Report.
cAddrDtlRptType	Indicates whether to print 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.
cAddrDtlRptIsol	Indicates whether to print the Address Detail Isolation Report. **OFF** Do not print the Address Detail Isolation Report. **ON** Print the Address Detail Isolation Report. **Blank** Defaults to OFF.
cReservedRS	Reserved.

Field	Description
lAddrDtlRptIsolPageLen	Maximum page length for the Address Detail Isolation Report.
lAddrDtlRptIsolMaxRec	Maximum number of records to output to the Address Detail Isolation Report.
lAddrDtlRptIsolNthRec	This field indicates the records to print on the Address Detail Isolation Report (for example., every 8th record).
cAddrDtlRptSugg	Indicates whether to print the Address Detail Suggestion Report. **OFF** Do not print the Address Detail Suggestion Report. **ON** Print the Address Detail Suggestion Report. **Blank** Defaults to OFF.
cAddrDtlRptInfo	This field indicates whether to print the Address Detail Information Report. **OFF** Do not print the Address Detail Information Report. **ON** Print the Address Detail Information Report. **Blank** Defaults to OFF.
cLogFileName	Log File Name.
cLogLevel	This field indicates the level of the message logged. **0** No logging. **1** Critical messages only. **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.
cSoftwareKey	Software key.
cExceptionTblFileName	Exceptions Table file name and path. For the CICS and IMS platforms, specify the literal "PBFNEXTB" in the cExceptionTblFileName field.
cEWSFileName	Early Warning System (EWS) file name and path.
cLOTFileName	Line of Travel (LOT) file name and path.
cDPVFilePath	Delivery Point Validation (DPV) File path.

Field	Description
cRDIFilePath	Residential Delivery Indicator (RDI) File path.
cLACSLinkFilePath	LACS^{Link} File path.
cSuiteLinkFilePath	Suite^{Link} File path.
cDPVKeyName	Delivery Point Validation (DPV) software key.
cLACSLinkKey	LACS^{Link} software key.
cOSSelected	Operating system name.
cAssignDPV	Indicates whether Finalist performs Delivery Point Validation (DPV) processing. **Note:** The USPS CASS regulations require Delivery Point Validation (DPV) processing to generate the USPS Form 3553 (USPS CASS Summary Report). **OFF** Do not perform Delivery Point Validation (DPV) processing. **ON** Perform Delivery Point Validation (DPV) processing using 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. NOTE: 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. 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.

Field	Description
cAssignDPVTie	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. **Note:** The USPS CASS regulations require DPV Tie Break processing to generate the USPS Form 3553 (USPS CASS Summary Report). EXAMPLE Input address 123 E Main St ZIP + 4 File 100 - 200 N Main St (Delivery Point Validated=Yes) 100 - 200 S Main St (Delivery Point Validated=No) Match Allowed 123 N Main St **OFF** Do not perform DPV Tie Breaker processing. **ON** Perform DPV Tie Breaker processing. The DPV Tie Breaker Option can increase your matching percentages but can also negatively impact performance. **Blank** Defaults to ON. **Note:** The default value is "ON"; however, the default value is only applied when cAssignDPV is defined as ON, MEM, SPL, or FLT.
cAssignDPVNoStat	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 "Delivery Point Validation (DPV) Option" 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.

Field	Description
cAssignRDI	This field indicates whether Finalist will perform Residential Delivery Indicator (RDI) processing. **OFF** Do not perform Residential Delivery Indicator (RDI) processing. **ON** Perform Residential Delivery Indicator (RDI) processing. **Blank** Defaults to OFF.
cAssignEWS	This field indicates whether Finalist will perform Early Warning System (EWS) processing. **OFF** Do not perform Early Warning System (EWS) processing. **ON** Perform Early Warning System (EWS) processing. **Blank** Defaults to OFF.
cAssignCMRA	This field indicates whether Finalist will perform CMRA processing. **OFF** Do not perform CMRA processing. **ON** Perform CMRA processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the CMRA Table is used in memory or outside of memory. **IN** Perform CMRA Table processing. Load the CMRA Table into memory. Disregard the DPV Buffer Size setting (cDPVBufSize). **OUT** Perform CMRA Table processing. Use the CMRA Table outside of memory. Disregard the DPV Buffer Size setting (cDPVBufSize). **Blank** Defaults to OFF.
cAssignLACSLink	This field determines whether Finalist performs LACS^{Link} processing. The USPS CASS regulations require LACS^{Link} processing. If you do not perform LACS^{Link} processing, Finalist does not generate a USPS Form 3553 (CASS Summary Report). **OFF** Do not perform LACS^{Link} processing. Finalist does not generate the USPS Form 3553 (CASS Summary Report). **ON** Perform LACS^{Link} processing. Finalist generates the USPS Form 3553 (CASS Summary Report). **Blank** Defaults to OFF.
cAssignSuiteLink	This field indicates whether Finalist will perform Suite^{Link} processing. **OFF** Do not perform Suite^{Link} processing. **ON** Perform Suite^{Link} processing. **Blank** Defaults to OFF.

Field	Description
cDPVBufSize	Indicates the memory model for DPV processing: **P** Pico. Stores no data in memory. No tables or indexes are loaded. **U** Ultra-small. Stores no data in memory. Partial indexes are loaded. **S** Small **M** Medium **L** Large **H** Huge. Stores all data in memory. **Blank** Defaults to M.
cMailerAddress2	CASS Mailer address 2 line.
cMailerAddress3	CASS Mailer address 3 line.
cMailerAddress4	CASS Mailer address 4 line.
pPSBData	Address of the PSB to use in an IMS program for access HDAM database version of the Finalist Data and City files.
pPSBCity	Address of the PSB to use in an IMS program for access HDAM database version of the Finalist Data and City files.
cSLKOverride[4]	Reserved for future use.
cSuiteLinkSmallMem	Indicates the memory model for Suite^{Link} processing: **P** Pico. Stores no data in memory. No tables or indexes are loaded. **U** Ultra-small. Stores no data in memory. Partial indexes are loaded. **S** Small **M** Medium **L** Large **H** Huge. Stores all data in memory. **Blank** Defaults to L.

Field	Description
cSuiteLinkShutdown	Action to take when encountering a Suite^{Link} processing error during the processing run. **W** Issue warning and turn off Suite^{Link} processing. **S** Stop Finalist processing when encountering a Suite^{Link} error. **I** Ignore errors and continue Suite^{Link} processing. **Blank** Defaults to W. **Note:** USPS regulations require Suite^{Link} 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^{Link} 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^{Link} Shutdown Indicator to “S” in order to stop processing if the Suite^{Link} initialization fails and to prevent generation of an invalid USPS Form 3553 (CASS Summary Report).
cLACSLinkProcessing	Indicates the memory model for LACS^{Link} processing: **P** Pico. Stores no data in memory. No tables or indexes are loaded. **U** Ultra-small. Stores no data in memory. Partial indexes are loaded. **S** Small **M** Medium **L** Large **H** Huge. Stores all data in memory. **Blank** Defaults to S.
cDPVShutdownIndicator	Action to take when encountering a DPV Seed during the processing run. **W** Issue warning message when a DPV Seed has been encountered. Processing continues but DPV processing is disabled. **S** Stop processing when encountering a DPV Seed. Finalist issues an error message, stops processing succeeding addresses, and exits the program. **Blank** Defaults to W. **Note:** If a DPV seed is encountered and W is set, CASS processing is turned off.

Field	Description
cAssignDPVVacant	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. This field indicates whether Finalist uses the Vacant Table and returns the result. **OFF** Do not perform Vacant Table processing. **ON** Perform Vacant Table processing. The DPV Buffer Size setting (cDPVBufSize) determines whether the Vacant 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 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.
cRetSLKinputSecdry	Indicate how to return secondary information when SuiteLink 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. # Return both the SuiteLink information and the unmatched secondary as "#". **Blank** Defaults to B. **NOTE:** If SuiteLink 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 SuiteLink secondary. Input secondary information includes the “#” designator for purposes of this option.

Field	Description
cAssignDPVPBSA	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.
cR777Deliverable	Addresses with Carrier Route code R77x are phantom routes and 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 code R77x marked as deliverable, set this option to OFF and the following actions are performed for the address: • No ZIP + 4 code is assigned. • Address is not counted on the USPS Form 3553 (CASS Summary Report). • DPV Footnote of R7 is returned if the record is DPV confirmed. Valid values are: **OFF** R77x addresses are not deliverable. **ON** R77x addresses are deliverable. **Blank** Defaults to ON.
cConvertSecPMB	Indicate whether to convert secondary information to “PMB” under the following conditions: • A secondary number is present in the returned ZIP + 4 address • The secondary number does not DPV confirm • The primary number (and/or other secondary number) confirms as a CMRA • The unconfirmed unit designator is not a pound sign (#) This processing only applies if the primary address codes to a CMRA. Valid values are: **OFF** Do not perform Secondary to PMB conversion processing. **ON** Perform Secondary to PMB conversion processing. **Blank** Defaults to OFF.

Field	Description
sErrorNum	Index into error message array.
1PCL74Error	Finalist 7.4 error codes.
cMessageLevel	Error message severity level. This is the message identifier. Values and error message type are: **0** No logging. **1** Critical messages only. **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.
cMessageCode	This is the text message for the cMessageLevel identifier.
pFileName	Pointer to the source file location of the error.
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.
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.
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).
cZip4BaseVer	Three-digit version number for the ZIP + 4 database. This number returns without punctuation (i.e., database X.XX will return as XXX).
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).
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).
cCityBaseVer	This is the three-digit version number for the City database. This number returns without punctuation (i.e., database X.XX will return as XXX).

Field	Description
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).
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).
cCZdBTimeStamp	The date and time the city base was built. In case multiple files were created in a month, this will help you identify which file is being used by build date and time. The date and time display in a MM DD YYYY and HH:MM:SS format.
cCZdBPlatform	Describes the platform for which the city database was built. **PC** Windows platform. **UNIX** Unix platform. **MF** Mainframe platform.
cLOTBaseVer	This is the version number for the LOT database. This number is returned without punctuation (i.e., 2.00 returns as 200).
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).
cLOTProdName	This is the name of the Line of Travel (LOT) database file being used in the run.
cLOTCfg	The LOT configuration in use.
cOSVersion	The version of the operating system.
cTotalMemory	Total amount of system memory when available.
cAvailMemory	Amount of available system memory when available.
cProcessorType	Specifies the type of processor in the system when available.
cCSDVersion	Operating system service pack level when available.
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).
cLACSProdName	Certified LACS^{Link} product name.

Field	Description
cLACSVersion	LACS[Link] product version number.
cLACSBaseDate	LACS[Link] database date in format yyyymmdd.
cSuiteLinkProdName	Certified Suite[Link] product name.
cSuiteLinkVersion	Suite[Link] product version number.
cSuiteLinkBaseDate	Suite[Link] database date in format yyyymmdd.
cSuiteLinkBaseExpDate	Suite[Link] database expiration date in format yyyymmdd.
char CPUID char CPUID2 char CPUID3 char CPUID4 char CPUID5 char CPUID6 char CPUID7 char CPUID8 char CPUID9 char CPUID10	Up to 10 CPU IDs/MAC Addresses.
cDPVBaseDate	DPV database expiration date in format mm-dd-yyyy.
cEWSBaseDate	EWS database expiration date in format mm-dd-yyyy.
cRDIBaseDate	RDI database expiration date in format mm-dd-yyyy.
cDBBuildDate	Date the Finalist database was built in format mm-dd-yyyy.
cSCBuildDate	Date the StateCut database was built in format mm-dd-yy (blank if StateCut database was not built).
iInitRtnCode	PBFNInit call return code.

Field	Description
cAssignDPVDNA	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 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.
cAssignDPVNSL	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.

Field	Description
cAssignDPVTHRWBK	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.
cAssignPreciselyID	This field indicates whether Finalist will perform PreciselyID processing. **OFF** Do not perform PreciselyID processing. **ON** Perform PreciselyID processing. **Blank** Defaults to OFF.
cPreciselyIDDate	Date the PreciselyID database was built (format mm-dd-yyyy).

COBOL Copybook - PBFNGCFG

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNGCFG.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for PBFNSetupDefinition                **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
```

```
** Precisely                                                    **
**                                                                 **

*********************************************************************

 05  PBFN-GCFG-SETUP.
   10  FILLER                          PIC X(004) VALUE '0900'.
   10  FILLER                          PIC X(005) VALUE 'GCFG'.
   10  PBFN-GCFG-CITYFILE-NAME         PIC X(256).
   10  PBFN-GCFG-ZIP4FILE-NAME         PIC X(256).
   10  PBFN-GCFG-CONFIGFILE-NAME       PIC X(256).
   10  PBFN-GCFG-LOADSETUP             PIC X(007).
     88  PBFN-GCFG-SETUPLOAD                    VALUE 'LOAD'.
     88  PBFN-GCFG-SETUPNOLOAD                VALUE 'NOLOAD'.

   10  PBFN-GCFG-CASS-FLAG             PIC X(004).
     88  PBFN-GCFG-CASS-ON                      VALUE 'ON'.
     88  PBFN-GCFG-CASS-OFF                     VALUE 'OFF'.
   10  PBFN-GCFG-PROCESSUNASSIGNED     PIC X(004).
     88  PBFN-GCFG-PROC-ASSIGN                  VALUE 'ON'.
     88  PBFN-GCFG-NO-PROC-ASSIGN               VALUE 'OFF'.
   10  PBFN-GCFG-ASSIGNCR              PIC X(004).
     88  PBFN-GCFG-ASSIGN-CR                    VALUE 'ON'.
     88  PBFN-GCFG-NO-ASSIGN-CR                 VALUE 'OFF'.
   10  PBFN-GCFG-ASSIGNABBREVCITY      PIC X(004).
     88  PBFN-GCFG-ASSIGN-CITY                  VALUE 'ON'.
     88  PBFN-GCFG-NO-ASSIGN-CITY               VALUE 'OFF'.
   10  PBFN-GCFG-ASSIGNLOT             PIC X(004).
     88  PBFN-GCFG-ASSIGN-LOT                   VALUE 'ON'.
     88  PBFN-GCFG-NO-ASSIGN-LOT                VALUE 'OFF'.
   10  PBFN-GCFG-RETDPBC               PIC X(004).
     88  PBFN-GCFG-ASSIGN-DPBC                  VALUE 'ON'.
     88  PBFN-GCFG-NO-ASSIGN-DPBC               VALUE 'OFF'.
   10  PBFN-GCFG-CACHESIZE             PIC X(020).
     88  PBFN-GCFG-CACHE-MIN                    VALUE '000'.
     88  PBFN-GCFG-CACHE-MAX                    VALUE '999'.
     88  PBFN-GCFG-CACHE-OFF                    VALUE 'OFF'.
   10  PBFN-GCFG-BEGFRAME              PIC X(002).
   10  PBFN-GCFG-ENDFRAME              PIC X(002).
   10  PBFN-GCFG-STANDARDCASE          PIC X(002).
     88  PBFN-GCFG-LOWER                        VALUE 'L '.
     88  PBFN-GCFG-MIXED                        VALUE 'M '.
     88  PBFN-GCFG-UPPER                        VALUE 'U '.
   10  PBFN-GCFG-DUALADDRSWT           PIC X(004).
     88  PBFN-GCFG-LINE-2-ONLY                  VALUE 'ACZ'.
     88  PBFN-GCFG-LINE-1-PRIORITY              VALUE 'L12'.
     88  PBFN-GCFG-LINE-2-PRIORITY              VALUE 'L21'.
     88  PBFN-GCFG-FIRST-CORRECT                VALUE '1ST'.
     88  PBFN-GCFG-POBOX                        VALUE 'P'.
     88  PBFN-GCFG-ADDR                         VALUE 'A'.
   10  PBFN-GCFG-RETALIST-NAME         PIC X(004).
     88  PBFN-GCFG-RET-ALIASSTR                 VALUE 'ON'.
```

```
           88  PBFN-GCFG-NO-RET-ALIASSTR                VALUE 'OFF'.
           88  PBFN-GCFG-PREFERRED                      VALUE 'P'.
           88  PBFN-GCFG-PREF-OR-ABBR                   VALUE 'PX'.
           88  PBFN-GCFG-PREF-ABBR-OTHER                VALUE 'PXO'.
           88  PBFN-GCFG-ABBREVIATED                    VALUE 'X'.
           88  PBFN-GCFG-ABBR-OR-PREF                   VALUE 'XP'.
           88  PBFN-GCFG-ABBR-PREF-OTHER                VALUE 'XPO'.
         10  PBFN-GCFG-PADDEDSTRINGDATA       PIC X(004).
           88  PBFN-GCFG-PADDED                         VALUE 'ON'.
           88  PBFN-GCFG-NOT-PADDED                     VALUE 'OFF'.
         10  PBFN-GCFG-UPPERCASE-IN           PIC X(004).
           88  PBFN-GCFG-UPPERCASEIN                    VALUE 'ON'.
           88  PBFN-GCFG-NO-UPPERCASEIN                 VALUE 'OFF'.
         10  PBFN-GCFG-REMOVENOISECHAR        PIC X(004).
           88  PBFN-GCFG-REMOVENOISE                    VALUE 'ON'.
           88  PBFN-GCFG-NO-REMOVENOISE                 VALUE 'OFF'.
         10  PBFN-GCFG-PROCESSFIRMS           PIC X(004).
           88  PBFN-GCFG-PROCFIRMS                      VALUE 'ON'.
           88  PBFN-GCFG-NO-PROCFIRMS                   VALUE 'OFF'.
         10  PBFN-GCFG-RETINPUTFIRM           PIC X(004).
           88  PBFN-GCFG-RETINPUTFIRMS                  VALUE 'ON'.
           88  PBFN-GCFG-NO-RETINPUTFIRM                VALUE 'OFF'.
         10  PBFN-GCFG-ALLSTREETMATCHING      PIC X(004).
           88  PBFN-GCFG-ALLSTREETMATCH                 VALUE 'ON'.
           88  PBFN-GCFG-NO-ALLSTREETMATCH              VALUE 'OFF'.
         10  PBFN-GCFG-SUGGESTIONCOUNT        PIC X(004).
         10  PBFN-GCFG-MAXEXCPTBLENTRIES      PIC X(005).
         10  PBFN-GCFG-RESERVED1              PIC X(060).
         10  PBFN-GCFG-STATFILENAMESETUP      PIC X(256).
         10  PBFN-GCFG-C3553FILENAMESETUP     PIC X(256).
         10  PBFN-GCFG-CASSCOMP-NAME          PIC X(060).
         10  PBFN-GCFG-CASSPROD-NAME          PIC X(060).
         10  PBFN-GCFG-CASSPRODVER            PIC X(060).
         10  PBFN-GCFG-Z4CHNGCERTCOMP-NAME    PIC X(060).
         10  PBFN-GCFG-Z4CHNGPROD-NAME        PIC X(060).
         10  PBFN-GCFG-Z4CHNGPRODVER          PIC X(060).
         10  PBFN-GCFG-LOTCERTCOMP-NAME       PIC X(060).
         10  PBFN-GCFG-LOTCERTPROD-NAME       PIC X(060).
         10  PBFN-GCFG-LOTCERTPRODVER         PIC X(060).
         10  PBFN-GCFG-MAILER-NAME            PIC X(030).
         10  PBFN-GCFG-MAILERADDRESS          PIC X(030).
         10  PBFN-GCFG-MAILERCITYLINE         PIC X(030).
         10  PBFN-GCFG-CASSCONFIGURATION      PIC X(004).
         10  PBFN-GCFG-BATCHRPTOPT            PIC X(004).
           88  PBFN-GCFG-BATCH-REPORT                   VALUE 'ON'.
           88  PBFN-GCFG-NO-BATCH-REPORT                VALUE 'OFF'.
         10  PBFN-GCFG-C3553RPTOPT            PIC X(004).
           88  PBFN-GCFG-CASS-REPORT                    VALUE 'ON'.
           88  PBFN-GCFG-NO-CASS-REPORT                 VALUE 'OFF'.
         10  PBFN-GCFG-RESERVED3              PIC X(030).
         10  PBFN-GCFG-RPTFILE-NAME           PIC X(256).
         10  PBFN-GCFG-RPTTITLE               PIC X(060).
         10  PBFN-GCFG-ADDRDTLRPT-TYPE        PIC X(004).
```

```
           88  PBFN-GCFG-DETAIL-RPT                    VALUE 'ON'.
           88  PBFN-GCFG-NO-DETAIL-RPT                 VALUE 'OFF'.
           88  PBFN-GCFG-ERROR                         VALUE 'ERR'.
         10  PBFN-GCFG-ADDRDTLRPTISOL          PIC X(004).
           88  PBFN-GCFG-ISOL-RPT                      VALUE 'ON'.
           88  PBFN-GCFG-NO-ISOL-RPT                   VALUE 'OFF'.
         10  PBFN-GCFG-RESERVEDRS              PIC X(001).
         10  PBFN-GCFG-ADDRDTLRPTISOLPG-LEN    PIC S9(08) BINARY.
         10  PBFN-GCFG-ADDRDTLRPTISOLMAXREC    PIC S9(08) BINARY.
         10  PBFN-GCFG-ADDRDTLRPTISOLNTHREC    PIC S9(08) BINARY.
         10  PBFN-GCFG-ADDRDTLRPTSUGG          PIC X(004).
           88  PBFN-GCFG-SUGG-RPT                      VALUE 'ON'.
           88  PBFN-GCFG-NO-SUGG-RPT                   VALUE 'OFF'.
         10  PBFN-GCFG-ADDRDTLRPTINFO          PIC X(004).
           88  PBFN-GCFG-INFO-RPT                      VALUE 'ON'.
           88  PBFN-GCFG-NO-INFO-RPT                   VALUE 'OFF'.
         10  PBFN-GCFG-RESERVED4               PIC X(030).
         10  PBFN-GCFG-LOGFILE-NAME            PIC X(256).
         10  PBFN-GCFG-LOGLEVEL                PIC X(020).
           88  PBFN-GCFG-LOG-NONE                      VALUE '0'.
           88  PBFN-GCFG-LOG-CRITICAL                  VALUE '1'.
           88  PBFN-GCFG-LOG-ERROR                     VALUE '2'.
           88  PBFN-GCFG-LOG-WARNING                   VALUE '3'.
           88  PBFN-GCFG-LOG-INFO                      VALUE '4'.
           88  PBFN-GCFG-LOG-DEBUG                     VALUE '5'.
         10  PBFN-GCFG-SOFTWAREKEY             PIC X(060).
         10  PBFN-GCFG-EXCP-TBLFILE-NAME       PIC X(256).
         10  PBFN-GCFG-EWSFILE-NAME            PIC X(256).
         10  PBFN-GCFG-LOTFILE-NAME            PIC X(256).
         10  PBFN-GCFG-DPVFILEPATH             PIC X(256).
         10  PBFN-GCFG-RDIFILEPATH             PIC X(256).
         10  PBFN-GCFG-LACSLINKFILEPATH        PIC X(256).
         10  PBFN-GCFG-SUITELINKFILEPATH       PIC X(256).
         10  PBFN-GCFG-DPVKEY-NAME             PIC X(256).
         10  PBFN-GCFG-LACSLINKKEY             PIC X(256).
         10  PBFN-GCFG-OSSELECTED              PIC X(020).
         10  PBFN-GCFG-ASSIGNDPV               PIC X(004).
           88  PBFN-GCFG-ASSIGN-DPV                    VALUE 'ON'.
           88  PBFN-GCFG-NO-ASSIGN-DPV                 VALUE 'OFF'.
           88  PBFN-GCFG-IN-MEMORY                     VALUE 'MEM'.
           88  PBFN-GCFG-SPLIT                         VALUE 'SPL'.
           88  PBFN-GCFG-FLAT                          VALUE 'FLT'.
         10  PBFN-GCFG-ASSIGNDPVTIE            PIC X(004).
           88  PBFN-GCFG-DPV-TIE                       VALUE 'ON'.
           88  PBFN-GCFG-NO-DPV-TIE                    VALUE 'OFF'.
         10  PBFN-GCFG-ASSIGNDPVNOSTAT         PIC X(004).
           88  PBFN-GCFG-DPV-NOSTAT                    VALUE 'ON'.
           88  PBFN-GCFG-NO-DPV-NOSTAT                 VALUE 'OFF'.
         10  PBFN-GCFG-ASSIGNRDI               PIC X(004).
           88  PBFN-GCFG-ASSIGN-RDI                    VALUE 'ON'.
           88  PBFN-GCFG-NO-ASSIGN-RDI                 VALUE 'OFF'.
           88  PBFN-GCFG-ASSIGN-NOSTAT-IN              VALUE 'IN'.
           88  PBFN-GCFG-ASSIGN-NOSTAT-OUT             VALUE 'OUT'.
```

```
          10  PBFN-GCFG-ASSIGNEWS              PIC X(004).
            88  PBFN-GCFG-ASSIGN-EWS                      VALUE 'ON'.
            88  PBFN-GCFG-NO-ASSIGN-EWS                   VALUE 'OFF'.
          10  PBFN-GCFG-ASSIGNCMRA            PIC X(004).
            88  PBFN-GCFG-ASSIGN-CMRA                     VALUE 'ON'.
            88  PBFN-GCFG-NO-ASSIGN-CMRA                  VALUE 'OFF'.
            88  PBFN-GCFG-ASSIGN-CMRA-IN                  VALUE 'IN'.
            88  PBFN-GCFG-ASSIGN-CMRA-OUT                 VALUE 'OUT'.
          10  PBFN-GCFG-ASSIGNLACSLINK        PIC X(004).
            88  PBFN-GCFG-ASSIGN-LACS                     VALUE 'ON'.
            88  PBFN-GCFG-NO-ASSIGN-LACS                  VALUE 'OFF'.
          10  PBFN-GCFG-ASSIGNSUITELINK       PIC X(004).
            88  PBFN-GCFG-ASSIGN-SLK                      VALUE 'ON'.
            88  PBFN-GCFG-NO-ASSIGN-SLK                   VALUE 'OFF'.
          10  PBFN-GCFG-DPVBUFSIZE            PIC X(004).
            88  PBFN-GCFG-DPV-PICO                        VALUE 'P'.
            88  PBFN-GCFG-DPV-ULTRA-SMALL                 VALUE 'U'.
            88  PBFN-GCFG-DPV-SMALL                       VALUE 'S'.
            88  PBFN-GCFG-DPV-MEDIUM                      VALUE 'M'.
            88  PBFN-GCFG-DPV-LARGE                       VALUE 'L'.
            88  PBFN-GCFG-DPV-HUGE                        VALUE 'H'.
          10  PBFN-GCFG-MAILERADDRESS2        PIC X(030).
          10  PBFN-GCFG-MAILERADDRESS3        PIC X(030).
          10  PBFN-GCFG-MAILERADDRESS4        PIC X(030).
          10  FILLER                          PIC X(033).
          10  PBFN-GCFG-PSBDATA               PIC X(004).
          10  PBFN-GCFG-PSBCITY               PIC X(004).
          10  PBFN-GCFG-SLKOVERRIDE           PIC X(004).
          10  PBFN-GCFG-SUITELINKSMALLMEM     PIC X(004).
            88  PBFN-GCFG-SLK-PICO                        VALUE 'P'.
            88  PBFN-GCFG-SLK-ULTRA-SMALL                 VALUE 'U'.
            88  PBFN-GCFG-SLK-SMALL                       VALUE 'S'.
            88  PBFN-GCFG-SLK-MEDIUM                      VALUE 'M'.
            88  PBFN-GCFG-SLK-LARGE                       VALUE 'L'.
            88  PBFN-GCFG-SLK-HUGE                        VALUE 'H'.
          10  PBFN-GCFG-SUITELINKSHUTDOWN     PIC X(002).
            88  PBFN-GCFG-SLK-WARNING                     VALUE 'W'.
            88  PBFN-GCFG-SLK-SHUTDOWN                    VALUE 'S'.
            88  PBFN-GCFG-SLK-IGNORE                      VALUE 'I'.
          10  FILLER                          PIC X(024).
          10  PBFN-GCFG-LACSLINKPROCESSING    PIC X(002).
            88  PBFN-GCFG-LLK-PICO                        VALUE 'P'.
            88  PBFN-GCFG-LLK-ULTRA-SMALL                 VALUE 'U'.
            88  PBFN-GCFG-LLK-SMALL                       VALUE 'S'.
            88  PBFN-GCFG-LLK-MEDIUM                      VALUE 'M'.
            88  PBFN-GCFG-LLK-LARGE                       VALUE 'L'.
            88  PBFN-GCFG-LLK-HUGE                        VALUE 'H'.
          10  PBFN-GCFG-DPVSHUTDOWNINDICATOR  PIC X(002).
            88  PBFN-GCFG-DPV-WARNING                     VALUE 'W'.
            88  PBFN-GCFG-DPV-SHUTDOWN                    VALUE 'S'.
            88  PBFN-GCFG-DPV-IGNORE                      VALUE 'I'.
          10  PBFN-GCFG-ASSIGNDPVVACANT       PIC X(004).
            88  PBFN-GCFG-DPV-VACANT                      VALUE 'ON'.
```

```
           88  PBFN-GCFG-DPV-NO-VACANT                  VALUE 'OFF'.
           88  PBFN-GCFG-ASSIGN-VACANT-IN               VALUE 'IN'.
           88  PBFN-GCFG-ASSIGN-VACANT-OUT              VALUE 'OUT'.
         10  PBFN-GCFG-RETSLKINPUTSECDRY      PIC X(004).
           88  PBFN-GCFG-RETSLKINPUTSECB                VALUE 'B'.
           88  PBFN-GCFG-RETSLKINPUTSECS                VALUE 'S'.
           88  PBFN-GCFG-RETSLKINPUTSECI                VALUE 'I'.
           88  PBFN-GCFG-RETSLKINPUTSECN                VALUE 'N'.
         10  PBFN-GCFG-ASSIGNDPVPBSA          PIC X(004).
           88  PBFN-GCFG-DPV-PBSA                       VALUE 'ON'.
           88  PBFN-GCFG-NO-DPV-PBSA                    VALUE 'OFF'.
           88  PBFN-GCFG-ASSIGN-PBSA-IN                 VALUE 'IN'.
           88  PBFN-GCFG-ASSIGN-PBSA-OUT                VALUE 'OUT'.
         10  PBFN-GCFG-R777DELIVERABLE        PIC X(004).
           88  PBFN-GCFG-R777-DELIVERABLE               VALUE 'ON'.
           88  PBFN-GCFG-R777-NOT-DELIVERABLE           VALUE 'OFF'.
         10  PBFN-GCFG-CONVERTSECPMB          PIC X(004).
           88  PBFN-GCFG-CONVERTSECPMB-ON               VALUE 'ON'.
           88  PBFN-GCFG-CONVERTSECPMB-OFF              VALUE 'OFF'.
         10  FILLER                           PIC X(001).
         10  PBFN-GCFG-ERRORNUM               PIC S9(04) BINARY.
         10  FILLER                           PIC X(002).
         10  PBFN-GCFG-PCL74ERROR             PIC S9(08) BINARY.
         10  PBFN-GCFG-MESSAGELEVEL           PIC X(004).
         10  PBFN-GCFG-MESSAGECODE            PIC X(128).
         10  PBFN-GCFG-PFILE-NAME             PIC X(256).
         10  PBFN-GCFG-CASSENGINEVERSION      PIC X(013).
         10  PBFN-GCFG-ENGINEBUILD            PIC X(025).
         10  PBFN-GCFG-ENGINEEXPDATE          PIC X(011).
         10  PBFN-GCFG-ZIP4BASEVER            PIC X(013).
         10  PBFN-GCFG-ZIP4BASEDATE           PIC X(011).
         10  PBFN-GCFG-ZIP4BASEEXPDATE        PIC X(011).
         10  PBFN-GCFG-CITYBASEVER            PIC X(013).
         10  PBFN-GCFG-CITYBASEDATE           PIC X(011).
         10  PBFN-GCFG-CITYBASEEXPDATE        PIC X(011).
         10  PBFN-GCFG-CZDBTIMESTAMP          PIC X(025).
         10  PBFN-GCFG-CZDBPLATFORM           PIC X(020).
         10  PBFN-GCFG-LOTBASEVER             PIC X(013).
         10  PBFN-GCFG-LOTBASEDATE            PIC X(011).
         10  PBFN-GCFG-LOTPROD-NAME           PIC X(016).
         10  PBFN-GCFG-LOTCFG                 PIC X(004).
         10  PBFN-GCFG-OSVERSION              PIC X(020).
         10  PBFN-GCFG-TOTALMEMORY            PIC X(020).
         10  PBFN-GCFG-AVAILMEMORY            PIC X(020).
         10  PBFN-GCFG-PROCESSOR-TYPE         PIC X(020).
         10  PBFN-GCFG-CSDVERSION             PIC X(020).
         10  PBFN-GCFG-OS-TYPE                PIC X(020).
         10  PBFN-GCFG-LACSPROD-NAME          PIC X(031).
         10  PBFN-GCFG-LACSVERSION            PIC X(013).
         10  PBFN-GCFG-LACSBASEDATE           PIC X(009).
         10  PBFN-GCFG-SUITELINKPROD-NAME     PIC X(031).
         10  PBFN-GCFG-SUITELINKVERSION       PIC X(013).
         10  PBFN-GCFG-SUITELINKBASEDATE      PIC X(011).
```

```
          10  PBFN-GCFG-SUITELINKBASEEXPDATE   PIC X(011).
          10  PBFN-GCFG-CPUID                 PIC X(007).
          10  PBFN-GCFG-CPUID2                PIC X(007).
          10  PBFN-GCFG-CPUID3                PIC X(007).
          10  PBFN-GCFG-CPUID4                PIC X(007).
          10  PBFN-GCFG-CPUID5                PIC X(007).
          10  PBFN-GCFG-CPUID6                PIC X(007).
          10  PBFN-GCFG-CPUID7                PIC X(007).
          10  PBFN-GCFG-CPUID8                PIC X(007).
          10  PBFN-GCFG-CPUID9                PIC X(007).
          10  PBFN-GCFG-CPUID10               PIC X(007).
          10  PBFN-GCFG-DPVBASEDATE           PIC X(011).
          10  PBFN-GCFG-EWSBASEDATE           PIC X(011).
          10  PBFN-GCFG-RDIBASEDATE           PIC X(011).
          10  PBFN-GCFG-DBBUILDDATE           PIC X(011).
          10  PBFN-GCFG-SCBUILDDATE           PIC X(011).
          10  PBFN-GCFG-IINITRTNCODE          PIC S9(08) BINARY.
          10  PBFN-GCFG-ASSIGNDPVDNA          PIC X(004).
            88  PBFN-GCFG-DPV-DNA                        VALUE 'ON'.
            88  PBFN-GCFG-NO-DPV-DNA                     VALUE 'OFF'.
            88  PBFN-GCFG-ASSIGN-DNA-IN                  VALUE 'IN'.
            88  PBFN-GCFG-ASSIGN-DNA-OUT                 VALUE 'OUT'.
          10  PBFN-GCFG-ASSIGNDPVNSL          PIC X(004).
            88  PBFN-GCFG-DPV-NSL                        VALUE 'ON'.
            88  PBFN-GCFG-NO-DPV-NSL                     VALUE 'OFF'.
            88  PBFN-GCFG-ASSIGN-NSL-IN                  VALUE 'IN'.
            88  PBFN-GCFG-ASSIGN-NSL-OUT                 VALUE 'OUT'.
          10  PBFN-GCFG-ASSIGNDPVTHRWBK       PIC X(004).
            88  PBFN-GCFG-DPV-THROWBACK                  VALUE 'ON'.
            88  PBFN-GCFG-NO-DPV-THROWBACK               VALUE 'OFF'.
            88  PBFN-GCFG-ASSIGN-THROWBACK-IN            VALUE 'IN'.
            88  PBFN-GCFG-ASSIGN-THROWBACK-OUT           VALUE 'OUT'.
          10  PBFN-GCFG-ASSIGNPRECISELYID     PIC X(004).
            88  PBFN-GCFG-ASSIGN-PRCSLYID                VALUE 'ON'.
            88  PBFN-GCFG-NO-ASSIGN-PRCSLYID             VALUE 'OFF'.
          10  PBFN-GCFG-PRECISELYIDDATE       PIC X(011).
          10  FILLER                          PIC X(136).
          10  FILLER                          PIC X(005).
```

PBFNStatsDef

The PBFNStatsDef C structure provides information on processing statistics.

Syntax

```
typedef struct PBFNStatsDefinition
{
    char            cVersion[VERSION_LEN];
```

```
    char             cApiId[APIID_LEN];
    char             cFiller[3];
    /* batch run times */
    unsigned int     lElapsedTime;
    unsigned int     lStartTime;
    unsigned int     lStopTime;
    /* totals  counts */
    unsigned int     lTotalProcessed;
    unsigned int     lTotalNotAssigned;
    unsigned int     lTotalAssigned;
    unsigned int     lTotalZipsAssigned;
    unsigned int     lTotalZ4Assigned;
    unsigned int     lTotalCorrected;
    unsigned int     lTotalDPBCAssigned;
    unsigned int     lTotalLOTAssigned;
    unsigned int     lTotalZ4ChangeProcessed;
    unsigned int     lTotalLACS;
    unsigned int     lTotalRDI;
    /* City assignment counts */
    unsigned int     lCityStateZipMatches;
    unsigned int     lStateOrZipMatches;
    unsigned int     lCityStateMatches;
    unsigned int     lStateAndZipMatches;
    /* City Line Data errors */
    unsigned int     lBlankCityLineData;
    unsigned int     lNoCityZipData;
    unsigned int     lNoStateZipData;
    /* City assignment error counts */
    unsigned int     lBadZipNoCity;
    unsigned int     lBadCityNoZip;
    unsigned int     lBadZipBadCity;
    unsigned int     lZipNotInDataBase;
    unsigned int     lTotalCityErrors;
    unsigned int     lTotalStateErrors;
    unsigned int     lTotalZipErrors;
    /* Address assignment counts */
    unsigned int     lTotalStreetLevel;
    unsigned int     lTotalFirm;
    unsigned int     lTotalPOBox;
    unsigned int     lTotalNonDeliverable;
    unsigned int     lTotalGenDelPostMaster;
    unsigned int     lTotalMilitary;
    /* Carrier Route breakdown counts */
    unsigned int     lTotalCrrt;
    unsigned int     lToCrrtNorm;
    unsigned int     lToCrrtBox;
    unsigned int     lToCrrtRR;
    unsigned int     lToCrrtHC;
    /* HighRise counts */
    unsigned int     lTotalHR;
    unsigned int     lToHRDefault;
    unsigned int     lToHRSecondary;
    /* RR HC Counts */
```

```
    unsigned int     lTotalRRHC;
    unsigned int     lToRRHCDefault;
    unsigned int     lToRRHCSecondary;
    unsigned int     lToRRBox;
    unsigned int     lToRRDefault;
    unsigned int     lToHCBox;
    unsigned int     lToHCDefault;
    /* Address Line Data errors */
    unsigned int     lNoStreetDataFound;
    unsigned int     lBlankAddressLineData;
    unsigned int     lNoRangeProvided;
    /* Address assignment error counts */
    unsigned int     lStreetNameNotFound;
    unsigned int     lStreetSuggOnly;
    unsigned int     lInvalidRangeForStreet;
    unsigned int     lSuffixError;
    unsigned int     lDirectionalError;
    unsigned int     lSufxDirError;
    unsigned int     lEWSFailure;
    unsigned int     lLOTFailure;
    unsigned int     lMultipleChoiceFailure;
    unsigned int     lUnknownFailure;
    unsigned int     lInvalidBoxNum;
    unsigned int     lMissingBoxNum;
    unsigned int     lOldProcDate;
    unsigned int     lR777Fail;
    unsigned int     lFirmFail;
    unsigned int     lSecFail;
    unsigned int     lZipMove;
    unsigned int     lUniqueZip;
    unsigned int     lPreciselyIDAttempt;
    unsigned int     lPreciselyIDY;
    unsigned int     lPreciselyIDD;
    char             cFiller1[32];
} PBFNStatsDef, *pPBFNStatsDef;
```

Field Descriptions

Table 117: PBFNStatsDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to GSTS.
lElapsedTime	Seconds elapsed since first address assignment and the most recent address assignment.

Field	Description
lStartTime	The lStartTime 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.
lStopTime	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.
lTotalProcessed	Total address records processed or presented to PBFNProcess.
lTotalNotAssigned	Total address records not assigned.
lTotalAssigned	Total address records assigned.
lTotalZipsAssigned	Total ZIP Codes assigned.
lTotalZ4Assigned	Total ZIP + 4 Codes assigned.
lTotalCorrected	Total address records corrected.
lTotalDPBCAssigned	Total number of address records assigned delivery point barcodes.
lTotalLOTAssigned	Total number of records assigned Line of Travel (LOT) codes.
lTotalZ4ChangeProcessed	Reserved for future use.
lTotalLacs	Total number of records assigned a LACS indicator.
lTotalRDI	Total number of records assigned a Residential Delivery Indicator (RDI).
lCityStateZipMatches	Total number of records where the input city, state, and ZIP Code exist.
lStateOrZipMatches	Total number of records where the input state or ZIP Code exist but city does not exist.
lCityStateMatches	Total number of records where the input city and state exist but ZIP Code does not exist.
lStateAndZipMatches	Total number of records where input state and ZIP Code exist but city does not exist.

Field	Description
lBlankCityLineData	Total number of records with a blank city line.
lNoCityZipData	Total number of records without city and ZIP Code.
lNoStateZipData	Total number of records without state or ZIP Code.
lBadZipNoCity	Total number of records with an invalid ZIP Code and no City provided.
lBadCityNoZip	Total number of records with an invalid City and no ZIP Code provided.
lBadZipBadCity	Total number of records with an invalid ZIP Code and an invalid city.
lZipNotInDataBase	Total number of records containing a ZIP Code not listed in the Finalist database.
lTotalCityErrors	Total number of records containing city errors.
lTotalStateErrors	Total number of records containing state errors.
lTotalZipErrors	Total number of records containing ZIP Code errors.
lTotalStreetLevel	Total number of ZIP + 4 Codes assigned to streets.
lTotalFirm	Total number of ZIP + 4 Codes assigned to firms.
lTotalPOBox	Total number of ZIP + 4 Codes assigned to Post Office boxes.
lTotalNonDeliverable	Total number of undeliverable addresses.
lTotalGenDelPostMaster	Total number of ZIP + 4 Codes assigned to general delivery.
lTotalMilitary	Total number of ZIP + 4 Codes assigned to military ZIP Codes.
lTotalCrrt	Total number of carrier routes assigned.
lToCrrtNorm	Total number of default carrier routes, including street, general delivery and postmaster.
lToCrrtBox	Total number of Post Office Box carrier routes.
lToCrrtRR	Total number of Rural Route carrier routes.

Field	Description
lToCrrtHC	Total number of Highway Contract carrier routes.
lTotalHR	Total number of high rises including defaults and secondary records.
lToHRDefault	Total number of high rise default records.
lToHRSecondary	Total number of high rise records with secondary information.
lTotalRRHC	Total number of rural route and highway contract records.
lToRRHCDefault	Total number of rural route and highway contract default records.
lToRRHCSecondary	Total number of rural routes and highway contract records with secondary box information.
lToRRBox	Total number of rural route records with secondary box.
lToRRDefault	Total number of rural route records without a secondary box.
lToHCBox	Total highway contract records with secondary box.
lToHCDefault	Total highway contract records without a secondary box.
lNoStreetDataFound	Total number of address records without street data.
lBlankAddressLineData	Total number of address records with a blank address line.
lNoRangeProvided	Total number of address records without a range.
lStreetNameNotFound	Total number of address records with a street name not found on the Finalist database.
lStreetSuggOnly	Total number of records that failed with suggested street records.
lInvalidRangeForStreet	Total number of address records containing an invalid range for the street name.
lSuffixError	Total number of address records missing suffixes or containing incorrect suffixes.
lDirectionalError	Total number of address records missing directionals or containing incorrect directionals.

Field	Description
lSufxDirError	The total number of address records missing both the suffix and directional or containing incorrect suffix and directional.
lEWSFailure	Total number of address records not matched because the records were found in the USPS Early Warning System (EWS) File.
lLOTFailure	Total number of address records where LOT assignment failed. Address coded successfully but LOT code was not assigned.
lMultipleChoiceFailure	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 (i.e., duplicate street names, two or more directionals available with none on input, etc.).
lUnknownFailure	Total number of address records with other errors that did not fall into one of the categories previously described.
lInvalidBoxNum	Total number of highway contract, rural route, and P. O. Box records with invalid box numbers.
lMissingBoxNum	Total number of highway contract, rural route, and P. O. Box records with missing box numbers.
lOldProcDate	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).
lR777Fail	Total number of R777 failures (E4602).
lFirmFail	Firm exists but not specified.
lSecFail	Secondary exists but not specified.
lZipMove	Failed due to ZIPMove processing.
lUniqueZip	Cannot match into Unique ZIP Code.
lPreciselyIDAttempt	Number of PreciselyID lookup attempts.
lPreciselyIDY	Number of PreciselyID found when PreciselyIDFound is Y.
lPreciselyIDD	Number of PreciselyID found when PreciselyIDFound is D.

COBOL Copybook - PBFNGSTS

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **

      ** Name: PBFNGSTS.CPY                                         **

      **                                                            **

      ** Use: COBOL Copybook for PBFNStatsDefinition                **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBFN-GSTS-STATS.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'GSTS'.
         10  FILLER                          PIC X(003).
         10  PBFN-GSTS-ELAPSEDTIME           PIC S9(08) BINARY.
         10  PBFN-GSTS-STARTTIME             PIC S9(08) BINARY.
         10  PBFN-GSTS-STOPTIME              PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALPROCESSED        PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALNOTASSIGNED      PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALASSIGNED         PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALZIPSASSIGNED     PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALZ4ASSIGNED       PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALCORRECTED        PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALDPBCASSIGNED     PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALLOTASSIGNED      PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALZ4CHGPROCESSED   PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALLACS             PIC S9(08) BINARY.
         10  PBFN-GSTS-TOTALRDI              PIC S9(08) BINARY.
         10  PBFN-GSTS-CITYSTATEZIPMATCHES   PIC S9(08) BINARY.
         10  PBFN-GSTS-STATEORZIPMATCHES     PIC S9(08) BINARY.
         10  PBFN-GSTS-CITYSTATEMATCHES      PIC S9(08) BINARY.
         10  PBFN-GSTS-STATEANDZIPMATCHES    PIC S9(08) BINARY.
         10  PBFN-GSTS-BLANKCITYLINEDATA     PIC S9(08) BINARY.
         10  PBFN-GSTS-NOCITYZIPDATA         PIC S9(08) BINARY.
         10  PBFN-GSTS-NOSTATEZIPDATA        PIC S9(08) BINARY.
         10  PBFN-GSTS-BADZIPNOCITY          PIC S9(08) BINARY.
         10  PBFN-GSTS-BADCITYNOZIP          PIC S9(08) BINARY.
         10  PBFN-GSTS-BADZIPBADCITY         PIC S9(08) BINARY.
```

```
           10  PBFN-GSTS-ZIPNOTINDATABASE       PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALCITYERRORS        PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALSTATEERRORS       PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALZIPERRORS         PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALSTREETLEVEL       PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALFIRM              PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALPOBOX             PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALNONDELIVERABLE    PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALGENDELPOSTMSTR    PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALMILITARY          PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALCRRT              PIC S9(08) BINARY.
           10  PBFN-GSTS-TOCRRTNORM             PIC S9(08) BINARY.
           10  PBFN-GSTS-TOCRRTBOX              PIC S9(08) BINARY.
           10  PBFN-GSTS-TOCRRTRR               PIC S9(08) BINARY.
           10  PBFN-GSTS-TOCRRTHC               PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALHR                PIC S9(08) BINARY.
           10  PBFN-GSTS-TOHRDEFAULT            PIC S9(08) BINARY.
           10  PBFN-GSTS-TOHRSECONDARY          PIC S9(08) BINARY.
           10  PBFN-GSTS-TOTALRRHC              PIC S9(08) BINARY.
           10  PBFN-GSTS-TORRHCDEFAULT          PIC S9(08) BINARY.
           10  PBFN-GSTS-TORRHCSECONDARY        PIC S9(08) BINARY.
           10  PBFN-GSTS-TORRBOX                PIC S9(08) BINARY.
           10  PBFN-GSTS-TORRDEFAULT            PIC S9(08) BINARY.
           10  PBFN-GSTS-TOHCBOX                PIC S9(08) BINARY.
           10  PBFN-GSTS-TOHCDEFAULT            PIC S9(08) BINARY.
           10  PBFN-GSTS-NOSTREETDATAFOUND      PIC S9(08) BINARY.
           10  PBFN-GSTS-BLANKADDRESSLINEDATA   PIC S9(08) BINARY.
           10  PBFN-GSTS-NORANGEPROVIDED        PIC S9(08) BINARY.
           10  PBFN-GSTS-STREETNAMENOTFOUND     PIC S9(08) BINARY.
           10  PBFN-GSTS-STREETSUGGONLY         PIC S9(08) BINARY.
           10  PBFN-GSTS-INV-RANGEFORSTREET     PIC S9(08) BINARY.
           10  PBFN-GSTS-SUFFIXERROR            PIC S9(08) BINARY.
           10  PBFN-GSTS-DIRECTIONALERROR       PIC S9(08) BINARY.
           10  PBFN-GSTS-SUFXDIRERROR           PIC S9(08) BINARY.
           10  PBFN-GSTS-EWSFAILURE             PIC S9(08) BINARY.
           10  PBFN-GSTS-LOTFAILURE             PIC S9(08) BINARY.
           10  PBFN-GSTS-MULTI-CHOICEFAILURE    PIC S9(08) BINARY.
           10  PBFN-GSTS-UNKNOWNFAILURE         PIC S9(08) BINARY.
           10  PBFN-GSTS-INVALIDBOXNUM          PIC S9(08) BINARY.
           10  PBFN-GSTS-MISSINGBOXNUM          PIC S9(08) BINARY.
           10  PBFN-GSTS-OLDPROCDATE            PIC S9(08) BINARY.
           10  PBFN-GSTS-R777FAIL               PIC S9(08) BINARY.
           10  PBFN-GSTS-FIRMFAIL               PIC S9(08) BINARY.
           10  PBFN-GSTS-SECFAIL                PIC S9(08) BINARY.
           10  PBFN-GSTS-ZIPMOVE                PIC S9(08) BINARY.
           10  PBFN-GSTS-UNIQUEZIP              PIC S9(08) BINARY.
           10  PBFN-GSTS-PRECISELYIDATTEMPT     PIC S9(08) BINARY.
           10  PBFN-GSTS-PRECISELYIDY           PIC S9(08) BINARY.
           10  PBFN-GSTS-PRECISELYIDD           PIC S9(08) BINARY.
           10  FILLER                           PIC X(032)
```

PBFNSuggestionDef

The PBFNSuggestionDef C structure contains an address suggestion generated by PBFNCreateSuggestionList().

Syntax

```
typedef struct PBFNSuggestionDefinition
{
    char    cVersion[VERSION_LEN];
    char    cApiId[APIID_LEN];
    char    cFirm[COMPANY_NAME];
    char    cUrb[29];
    char    cRange[10];
    char    cPreDirectional[3];
    char    cStreetName[28];
    char    cPostDirectional[3];
    char    cStreetSuffix[5];
    char    cUnitDesignator[5];
    char    cUnitNumber[10];
    char    cRangeLow[10];
    char    cRangeHi[10];
    unsigned char eob;
    char    cUnit2Designator[5];
    char    cUnit2Number[10];
    char    cPMUnitDesignator[4];
    char    cPMUnitNumber[10];
    char    cCity[28];
    char    cState[3];
    char    cZip[ZIP_LEN];
    char    cFiller1a[24];
    char    cFiller1[4];
}PBFNSuggestionDef, *pPBFNSuggestionDef;
```

Field Descriptions

Table 118: PBFNSuggestionDef Field Descriptions

Field	Description
cVersion	Structure version number.
cApiId	Structure identifier to DSUG.

Field	Description
cFirm	Firm name.
cUrb	Urbanization name.
cRange	Perfect match range.
cPreDirectional	Pre-directional.
cStreetName	Street name.
cPostDirectional	Post-directional.
cStreetSuffix	Suffix.
cUnitDesignator	Unit designator 1 type.
cUnitNumber[10]	Secondary range.
cRangeLow	Low range for ranged suggestions.
cRangeHi	High range for ranged suggestions.
unsigned char eob	Type of range. **E** Even only range. **O** Odd only range. **B** Both even and odd ranges. **C** Consolidated. Even and odd ranges consolidated as one using the lowest ZIP Code.
cUnit2Designator	Unit designator 2 type.
cUnit2Number	Unit 2 secondary range.
cPMUnitDesignator	PMB or MSC designator.
cPMUnitNumber	PMB or MSC number.
cCity	City name.

Field	Description
cState	State.
cZip	ZIP Code.

COBOL Copybook - PBFNDSUG

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                   **
      **                                                            **

      ** Name: PBFNDSUG.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for PBFNSuggestionDefinition           **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                  **
      **                                                            **

      ****************************************************************

       05  PBFN-DSUG-SUGGESTION.
         10  FILLER                          PIC X(004) VALUE 'NNNN'.
         10  FILLER                          PIC X(005) VALUE 'DSUG'.
         10  PBFN-DSUG-FIRM                  PIC X(060).
         10  PBFN-DSUG-URB                   PIC X(029).
         10  PBFN-DSUG-RANGE                 PIC X(010).
         10  PBFN-DSUG-PREDIRECTIONAL        PIC X(003).
         10  PBFN-DSUG-STREET-NAME           PIC X(028).
         10  PBFN-DSUG-POSTDIRECTIONAL       PIC X(003).
         10  PBFN-DSUG-STREETSUFFIX          PIC X(005).
         10  PBFN-DSUG-UNITDESIGNATOR        PIC X(005).
         10  PBFN-DSUG-UNITNUMBER            PIC X(010).
         10  PBFN-DSUG-RANGELOW              PIC X(010).
         10  PBFN-DSUG-RANGEHI               PIC X(010).
         10  PBFN-DSUG-EOB                   PIC X(001).
           88  PBFN-DSUG-EVEN                           VALUE 'E'.
           88  PBFN-DSUG-ODD                            VALUE 'O'.
           88  PBFN-DSUG-BOTH                           VALUE 'B'.
           88  PBFN-DSUG-CONSOLIDATED                   VALUE 'C'.
```

```
          10  PBFN-DSUG-UNIT2DESIGNATOR        PIC X(005).
          10  PBFN-DSUG-UNIT2NUMBER            PIC X(010).
          10  PBFN-DSUG-PMUNITDESIGNATOR       PIC X(004).
          10  PBFN-DSUG-PMUNITNUMBER           PIC X(010).
          10  PBFN-DSUG-CITY                   PIC X(028).
          10  PBFN-DSUG-STATE                  PIC X(003).
          10  PBFN-DSUG-ZIP                    PIC X(006).
          10  FILLER                           PIC X(024).
          10  FILLER                           PIC X(004).
```

Related APIs

PBCSCreateSuggestionList

USPSDetailDef

The USPSDetailDef C structure returns the detail data required by the USPS for each record creating a Delivery Point Validation (DPV) False Positive (Seed) Table violation.

Syntax

```
typedef struct USPSDetailDefinition{
    char    cStreetPreDir[2];
    char    cStreetName[28];
    char    cStreetSfx[4];
    char    cStreetPostDir[2];
    char    cPrimeRange[10];
    char    cUnitDesignator[4];
    char    cUnitNumber[8];
    char    cMatchedZipCode[5];
    char    cMatchedPlus4[4];
    char    filler[117];
} USPSDetailDef, *pUSPSDetailDef;
```

Field Descriptions

Table 119: USPSDetailDef Field Descriptions

Field	Description
cStreetPreDir	Parsed street predirectional for the record creating the False Positive (Seed) Table violation.

Field	Description
cStreetName	Parsed street name for the record creating the False Positive (Seed) Table violation.
cStreetSfx	Parsed street suffix for the record creating the False Positive (Seed) Table violation.
cStreetPostDir	Parsed street postdirectional for the record creating the False Positive (Seed) Table violation.
cPrimeRange	Parsed street primary range for the record creating the False Positive (Seed) Table violation.
cUnitDesignator	Parsed unit designator for the record creating the False Positive (Seed) Table violation.
cUnitNumber	Parsed unit number for the record creating the False Positive (Seed) Table violation.
cMatchedZipCode	Parsed ZIP Code for the record creating the False Positive (Seed) Table violation.
cMatchedPlus4	Parsed Plus4 ZIP Code for the record creating the False Positive (Seed) Table violation.

COBOL Copybook - PBFNPSDD

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNPSDD.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for USPSDetailDefinition               **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
```

```
      ************************************************************

         10  PBFN-PSDD-USPS.
           15  PBFN-PSDD-STREETPREDIR         PIC X(002).
           15  PBFN-PSDD-STREET-NAME          PIC X(028).
           15  PBFN-PSDD-STREETSFX            PIC X(004).
           15  PBFN-PSDD-STREETPOSTDIR        PIC X(002).
           15  PBFN-PSDD-PRIMERANGE           PIC X(010).
           15  PBFN-PSDD-UNITDESIGNATOR       PIC X(004).
           15  PBFN-PSDD-UNITNUMBER           PIC X(008).
           15  PBFN-PSDD-MATCHEDZIPCODE       PIC X(005).
           15  PBFN-PSDD-MATCHEDPLUS4         PIC X(004).
           15  FILLER                         PIC X(117).
```

USPSDPVHdrDef

The USPSDPVHdrDef C structure returns the header data required by the USPS for Delivery Point Validation (DPV) False Positive (Seed) Table violations.

Syntax

```
typedef struct USPDDPVHdrDefinition{
    char    cMailerCompany[40];
    char    cMailerAddress[58];
    char    cMailerCity[28];
    char    cMailerState[2];
    char    cMailerZip[9];
    char    cTotalRecProcessed[9];
    char    cTotalRecDPVMatched[9];
    char    cPerMatchtoDPV[9];
    char    cPerMatchtoZip4[9];
    char    cNumberZipsOnFile[5];
    char    cNumberofFalsePos[2];
    char    cFiller1[4];
} USPSDPVHdrDef, *pUSPSDPVHdrDef;
```

Field Descriptions

Table 120: USPSDPVHdrDef Field Descriptions

Field	Description
cMailerCompany	Mailer's name.

Field	Description
cMailerAddress	Mailer's address line.
cMailerCity	Mailer's city information.
cMailerState	Mailer's state information.
cMailerZip	Mailer's ZIP Code information.
cTotalRecProcessed	The total number of records processed.
cTotalRecDPVMatched	The total number of records matched to the DPV File.
cPerMatchtoDPV	The percent of the records processed that matched to the DPV File.
cPerMatchtoZip4	The percent of the records processed that matched to the ZIP + 4 File.
cNumberZipsOnFile	The number of different ZIP Codes contained in the input file.
cNumberofFalsePos	The number of addresses resulting in a hit against the False Positive (Seed) Table.

COBOL Copybook - PBFNPSDH

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                    **
      **                                                            **
      ** Name: PBFNPSDH.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for USPDDPVHdrDefinition               **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
```

```
      ****************************************************************

         10  PBFN-PSDH-USPS.
           15  PBFN-PSDH-MAILERCOMPANY        PIC X(040).
           15  PBFN-PSDH-MAILERADDRESS        PIC X(058).
           15  PBFN-PSDH-MAILERCITY           PIC X(028).
           15  PBFN-PSDH-MAILERSTATE          PIC X(002).
           15  PBFN-PSDH-MAILERZIP            PIC X(009).
           15  PBFN-PSDH-TOTALRECPROCESSED    PIC X(009).
           15  PBFN-PSDH-TOTALRECDPVMATCHED   PIC X(009).
           15  PBFN-PSDH-PERMATCHTODPV        PIC X(009).
           15  PBFN-PSDH-PERMATCHTOZIP4       PIC X(009).
           15  PBFN-PSDH-NUMBERZIPSONFILE     PIC X(005).
           15  PBFN-PSDH-NUMBEROFFALSEPOS     PIC X(002).
           15  FILLER                         PIC X(004).
```

USPSPBLACSDetDef

The USPSPBLACSDetDef structure returns the LACS^{Link} False Positive violation detail data required by the USPS. For each False Positive violation, a record is created in the USPS-required format.

Syntax

```
typedef struct USPSPBLACSDetDefinition
{
    char    cStreetPreDir[2];
    char    cStreetName[28];
    char    cStreetSfx[4];
    char    cStreetPostDir[2];
    char    cPrimeRange[10];
    char    cUnitDesignator[4];
    char    cUnitNumber[8];
    char    cMatchedZipCode[5];
    char    cMatchedPlus4[4];
    char    filler[117];
} USPSPBLACSDetDef, *pUSPSPBLACSDetDef;
```

Field Descriptions

Table 121: USPSPBLACSDetDef Field Descriptions

Field	Description
cStreetPreDir	Parsed street predirectional.

Field	Description
cStreetName	Parsed street name.
cStreetSfx	Parsed street suffix.
cStreetPostDir	Parsed street postdirectional.
cPrimeRange	Parsed street primary range.
cUnitDesignator	Parsed unit designator.
cUnitNumber	Parsed unit number.
cMatchedZipCode	Parsed ZIP Code.
cMatchedPlus4	Parsed Plus4 ZIP Code.

COBOL Copybook - PBFNPSLD

```
      ****************************************************************
      **                                                            **
      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **
      ** Name: PBFNPSLD.CPY                                         **
      **                                                            **
      ** Use: COBOL Copybook for USPSPBLACSDetDefinition            **
      **                                                            **
      ** Distributed Source Member for the Finalist product from    **
      ** Precisely                                                  **
      **                                                            **
      ****************************************************************

         10  PBFN-PSLD-USPS.
           15  PBFN-PSLD-STREETPREDIR         PIC X(002).
           15  PBFN-PSLD-STREET-NAME          PIC X(028).
           15  PBFN-PSLD-STREETSFX            PIC X(004).
```

```
           15  PBFN-PSLD-STREETPOSTDIR        PIC X(002).
           15  PBFN-PSLD-PRIMERANGE           PIC X(010).
           15  PBFN-PSLD-UNITDESIGNATOR       PIC X(004).
           15  PBFN-PSLD-UNITNUMBER           PIC X(008).
           15  PBFN-PSLD-MATCHEDZIPCODE       PIC X(005).
           15  PBFN-PSLD-MATCHEDPLUS4         PIC X(004).
           15  FILLER                         PIC X(117).
```

USPSPBLACSHdrDef

The USPSPBLACSHdrDef structure returns the LACS^{Link} False Positive violation header data required by the USPS for LACS^{Link} processing in the USPS-required format.

Syntax

```
typedef struct USPSPBDLACSHdrDefinition
{
    char    cMailerCompany[40];
    char    cMailerAddress[58];
    char    cMailerCity[28];
    char    cMailerState[2];
    char    cMailerZip[9];
    char    cTotalRecProcessed[9];
    char    cTotalRecLACSMatched[9];
    char    cFiller[29];
} USPSPBLACSHdrDef, *pUSPSPBLACSHdrDef;
```

Field Descriptions

Table 122: USPSPBLACSHdrDef Field Descriptions

Field	Description
cMailerCompany	Mailer company name.
cMailerAddress	Mailer address.
cMailerCity	Mailer city name.
cMailerState	Mailer state.
cMailerZip	Mailer ZIP Code.

Field	Description
cTotalRecProcessed	Total number of records sent to LACS[Link] for processing.
cTotalRecLACSMatched	Total number of records successfully matched and returned with LACS[Link] return codes A and 92.

COBOL Copybook - PBFNPSLH

```
      ****************************************************************

      **                                                            **

      ** (C) YYYY Precisely All Rights Reserved.                  **
      **                                                            **

      ** Name: PBFNPSLH.CPY                                        **

      **                                                            **

      ** Use: COBOL Copybook for USPSPBDLACSHdrDefinition           **

      **                                                            **

      ** Distributed Source Member for the Finalist product from    **

      ** Precisely                                                 **
      **                                                            **

      ****************************************************************

         10  PBFN-PSLH-USPS.
           15  PBFN-PSLH-MAILERCOMPANY        PIC X(040).
           15  PBFN-PSLH-MAILERADDRESS        PIC X(058).
           15  PBFN-PSLH-MAILERCITY           PIC X(028).
           15  PBFN-PSLH-MAILERSTATE          PIC X(002).
           15  PBFN-PSLH-MAILERZIP            PIC X(009).
           15  PBFN-PSLH-TOTALRECPROCESSED    PIC X(009).
           15  PBFN-PSLH-TOTALRECLACSMATCHED  PIC X(009).
           15  FILLER                         PIC X(029).
```

3 - Using the Compatibility Interface

In this section

Finalist Keys

Finalist Compatibility Interface (CI) users are required to store the Finalist software key using PGM=KEYSTORE. CICS and IMS transactions use the Compatibility Interface internally and therefore require the Finalist software key to be stored.

> **Note:** This process applies to the Finalist software key only. This process does not apply for the LACS^{Link} or DPV keys.

Mainframe (Batch and CICS)

Mainframe users should reference the KEYSTORE member in the FNSOURCE library for sample JCL to run the KEYSTORE program.

Using the Street Listing Program (LPFNLIST)

This section explains how to use LPFNLIST, the Finalist Street Listing Program, to generate the Street Directory Listing. The Street Directory Listing indicates sector segment numbers and carrier route codes associated with specific street ranges or firms. You can use this information to research addresses that Finalist was unable to code.

Control Cards

Control cards help you use the Street Listing Program. Note the following as you specify which portion of the Data File you want to print:

- You must enter at least one control card for each processing run, but there is no limit to the total number of cards you can enter.
- Begin entering data in column one.
- Finalist permits only one ZIP Code per card.
- The control card fields and their entry positions are as described next.

> **Note:** You cannot use positions 8-9 or 70-80 for entries to the Street Listing Program.

Table 123: LPFNLIST Control Cards

Field	Positions	Description
ZIP (Required)	1-5	ZIP stands for your target ZIP Code. The Street Listing Program returns data for the ZIP Code you entered in this position. This is a required field.
VAR	6	To print street name variations on the listing, enter Y. If this field is blank or contains any letter other than Y, no street variations will print. Street name variations are the spelling and phonetic variations Finalist uses to identify a street name if an input record includes a misspelling.
COLS	7	To print this report in a one-column format, enter "1". To print this report in a two-column format, enter "2". The default value is "2".
BEGIN ST	10-39	To print the entire ZIP listing, enter an asterisk. To limit the amount of information returned for the target ZIP Code, enter a beginning street or firm in this position. The Street Listing Program will print all entries beginning with the street or firm entered. You can also use a partial street name. For example, CA* would retrieve all streets beginning with CA.
END ST	40-69	The END ST position provides a stopping point for the Street Directory Listing. The Street Listing Program will print all streets and firms through the one listed. If you don't make an entry, the Street Directory Listing will contain all entries through the ZIP Code's end.

Examples

- To print all data for ZIP Code 60126:

```
COLS  1...5...10.........................................70
      60126   *
```

- To print all data for ZIP Code 60148 beginning with Maple Street:

```
COLS  1...5...10.....................40..................70
      60148    MAPLE
```

The resulting Street Directory Listing would include the data for Maple Street and all streets and firms following Maple Street in alphanumeric sequence.

- To print all streets and firms in ZIP Code 60148 on Maple Street only:

```
COLS  1...5...10.....................40..................70
      60148   MAPLE          MAPLE
```

- To print all data in ZIP Code 60148 beginning with Lexington Street through Maple Street in a one-column format:

```
COLS  1...5...10.....................40.................70
      60148 1  LEXINGTON               MAPLE
```

The data for Lexington through Maple would appear on the Street Directory Listing.

Street Directory Listing

The Street Listing Program prints the your specified information on the Street Directory Listing. This report lists portions of your Data File by ZIP Code, by streets and firms within a ZIP Code, or both. Also included on the report are sector segment numbers and carrier route codes associated with specific street ranges or firms. Use this information to modify addresses that Finalist was unable to code.

The Street Directory Listing prints in a two-up format -- meaning that data for two addresses or firms is given side-by-side. Left-side column headings are identical to column headings on the right side of the report. Here, the left side of the report is illustrated. The following is a sample Street Directory Listing Report.

Sample Street Directory Listing

```
INPUT ZIP 60148   MOTHER ZIP 60148    FINALIST VSAM STREET DIRECTORY
LISTING   DATE:  PAGE 1 [1]
      [2]                          [3]
 STREET NAME                      TYPE       SEC/SEG/CR SEC/SEG/CR
 SFX  ALS  DIRS ....RNG.LO ....RNG.HI ALPHS ...EVEN... ....ODD...
 ---- ---- ---- ---------- ---------- -- -- ---------- ----------
 [4]  [5]  [6]       [7]        [8]    [9]           [10]
 CHARING CROSS                 PRIME
 RD                  100        199        3919 C026  3918 C026
 RD                 1000       1099        3920 C026  3943 C026
 CHARINGCROSS  (a)             VAR OF CHARING CROSS
 CHARLE                        VAR OF CHARLES
 CHARLES                       PRIME
 LN                  500        599        2475 C013  2442 C013
 LN            (c)  20W501     20W599                  2442 C013
 CHARLOTTE                     PRIME
 ST        N            1         99        2312 C023  2311 C023
 ST        N   (b)      17 HIRS                         2319 C023
 ST        APT          1A         1F        2337 C023  2337 C023
 ST        APT          2G         2L        2362 C023  2362 C023
 ST        N            29 HIRS                         2366 C023
 ST        APT           A          C        2366 C023  2366 C023
 ST        S           100        138        2608 C003  2607 C003
 ST        N           100        136        2314 C023  2313 C023
```

```
ST          N             137          149          2314 C023   2364 C023
ST          S             139          199          2608 C003   2659 C003
ST          N             150          199          2367 C023   2364 C023
ST          N             200          299          2036 C023   2035 C023
ST          N             300          399          2038 C023   2037 C023
ST          N             500          599          1755 C023   1754 C023
ST          S             500          599          2741 C038   2740 C038
ST          S             600          699          3402 C032   3401 C032
ST          S             700          799          3404 C032   3403 C032
ST          S             800          899          3406 C032   3405 C032
ST          S             900          999          3408 C032   3407 C032
ST          S            1000         1008          3922 C026   3921 C026
ST          S            1009         1099          3922 C026   3944 C026
CHASE                                 PRIME
AVE         N               1          100          2051 C041   2050 C041
CT                          1           99          3625 C028   3625 C028
AVE         S               1           99          2902 C005   2901 C005
AVE         S             100          199          2904 C038   2903 C038
AVE         N             101          199          2051 C041   2082 C041
AVE         N             200          299          NDEL C041   2084 C041
AVE         S             200          299          2906 C038   2905 C038
AVE         S             239 HIRS                                2995 C038
AVE         APT             A            C          2995 C038   2995 C038
AVE         S             242 HIRS                   2963 C038
AVE         APT             A            F          2963 C038   2963 C038
AVE         S             300          305          2998 C038   NDEL C016
AVE         S             304 HIRS                   2999 C038
AVE         APT             A            D          2999 C038   2999 C038
AVE         S             306          399          2927 C016   2989 C016
AVE         S             400          499          2929 C016   2928 C016
AVE         S             500          599          2931 C016   2930 C016
AVE         S             600          699          NDEL C016   NDEL C016
LN                        700          714          3627 C028   3626 C028
LN                        715          799          3627 C028   3654 C028
LN                        800          899          3629 C028   3628 C028
AVE                     1S001        1S009          4713 C007   4713 C007
```

Numbers appear inside brackets "[ ]" on the sample report. These numbers refer to the following descriptions.

1. HEADING LINE — Heading line. This line contains:
   - INPUT ZIP AND MOTHER ZIP — The input ZIP Code is the ZIP Code that you requested information about. The mother ZIP Code is the Post Office designated center for the input ZIP Code.
   - REPORT TITLE — The report title appears on the heading line. Here, the title is: Finalist VSAM STREET DIRECTORY LISTING
   - PAGE NUMBER — The page numbers of the Street Directory Listing are given in the heading line.
2. STREET NAME — Name of the street or firm listed on the report.

3. TYPE

   - FIRM — The listing prints FIRM to the right of each firm name, and prints the firm sector segment number on the line below.
   - HIRS — The listing prints HIRS to the right of each high-rise name, and prints the high-rise sector segment number on the line below.
   - PRIME — This is the street name's primary spelling.
   - VAR — Indicates a variation used by Finalist to correct an input address. The street name's full spelling follows VAR. These variations only appear if you've specified "Y" on the control card column 6.

     Example

     In the listing, the line marked with (a) shows entry CHARINGCROSS is a variation of the street name CHARING CROSS.

4. SFX — This column shows the street's suffix (ST, AVE, or RD, etc.).
5. ALS — This column shows the alias indicator for the street name.

   - BLANK — Not an alias
   - A — Alias on secondary descriptor (apt., suite, etc.)
   - K — Nickname alias
   - P — Preferred alias
   - T — Standardized preferred alias (USPS-supplied preferred alias which matches the base street name as standardized by Finalist)

6. DIRS — This column shows the street directional, for example N, S, E, W. Values in positions one and two of this field refer to a pre-directional. Values in positions three and four refer to a post-directional.

   Example

   Pre-Directional — 18 W GUNTHER PKWY

   Post-Directional — 25 RIVERSIDE DR E

7. RNG.LO — This column contains the lowest address number for that particular street name, directional, suffix and sector segment.
8. RNG.HI — This column contains the highest address number for that particular street name, directional, suffix, and sector segment. Ranges beginning with a hyphen such as "-S200" indicate there is a leading zero in that address. When the word UNIT appears in the range category, the numbers that follow are apartment numbers.

   Example

   The line marked as (b) on the sample report shows that 17 N CHARLOTTE ST is a high rise. The line marked as (c) on the sample report shows a low range of 20W501 and a high range of 20W599. This indicates that this report line covers all odd-numbered addresses on CHARLES LN from 20W501 CHARLES through 20W599 CHARLES.

9. ALPHS — Finalist lists alphabetical characters in this field when the characters in the range do not follow a sequence that Finalist recognizes.
10. SEC/SEG/CR — The final two columns represent the sector segment and carrier routes defined for the even and odd street addresses.
    - NDEL indicates that a non-deliverable address, such as a vacant lot or cemetery.

MVS Execution JCL

The following is sample MVS JCL to execute the Street Listing Program. Enter the values that appear here in capital letters exactly as shown. Enter the values that appear here in lower-case letters with the appropriate substitutions to meet your installation requirements.

```
1   //stepname   EXEC  PGM=LPFNLIST
2   //STEPLIB    DD    DSN=finalist.loadlib,DISP=SHR
3   //SYSPRINT   DD    SYSOUT=*
4   //SYSUDUMP   DD    SYSOUT=*
5   //SYSLOG     DD    SYSOUT=*
6   //CBDATA     DD    DSN=finalist.datafile,DISP=SHR
7   //CBCTYST    DD    DSN=finalist.cityfile,DISP=SHR
8   //SYSIN      DD    *
9   60148Y      CH
10  /*
```

The numbers to the left of the JCL statements correspond to the following descriptions.

Statement Number	Description
1	EXEC statement identifies and invokes the Street Listing Program. Enter a statement name for "stepname" and any other required parameters.
2	STEPLIB DD statement identifies the program library containing the Street Listing Program. Replace "finalist.loadlib" with your library name.
3	SYSPRINT DD statement identifies the writer (print) class for the Street Directory Listing.
4	SYSUDUMP DD statement identifies the writer class for a dump of the Street Listing Program in the event of an ABEND.
5	SYSLOG DD statement identifies the writer class for VSAM I/O errors and the job's start/stop time.
6	CBDATA DD statement identifies the Finalist Data File. Replace "finalist.datafile" with your Data File name.

Statement Number	Description
7	CBCTYST DD statement identifies the Finalist City/St file. Replace "finalist.cityfile" with your City/St File name.
8	SYSIN DD statement indicates that control card input follows.
9	This statement is a control card for this processing run. This card indicates which street information the Street Listing Program prints.
10	This statement identifies the end of the instream control card file.

Using the Database APIs for Batch and CICS

You can use the Finalist Compatibility Interface (CI) database APIs to retrieve information directly from the Finalist database. This access method consists of a set of routines callable through the LPAM or LPAMC interface subroutine. With these routines, you can retrieve information from the City/State Files or Data File without knowledge of internal file structures. The Compatibility Interface database APIs are listed alphabetically and described in detail.

Parameter Summary

This table describes the Compatibility Interface (CI) database API parameters. Use this information in conjunction with the sample Assembler and COBOL calling sequences supplied for each of the Access Method functions.

Table 124: CI Database API Parameter Summary

Parameter	Description
anchor@	The INIT function fills in this fullword and other functions use it in subsequent calls. You must not modify this field after the INIT call.
callinfo	This is the call area for each specific function. Finalist supplies the copybook for each function. The Access Method never modifies this area.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>errinfo</td><td>The Access Method subroutine returns diagnostic information in the event of a non-zero return code to this area.

Note: This field is not automatically initialized. Check the retcode field prior to the errinfo field. (Refer to retcode later in this section.)

The Errinfo layout is:

<table>
<tr><th></th><th>Field</th><th>Length</th><th>Position</th></tr>
<tr><td></td><td>Component ID</td><td>8</td><td>1-8</td></tr>
<tr><td></td><td>Blank (filler)</td><td>1</td><td>9-9</td></tr>
<tr><td></td><td>Service/macro in error</td><td>8</td><td>10-17</td></tr>
<tr><td></td><td>Blank (filler)</td><td>1</td><td>18-18</td></tr>
<tr><td></td><td>Return code(s)</td><td>16</td><td>19-34</td></tr>
<tr><td></td><td>Blank (filler)</td><td>1</td><td>35-35</td></tr>
<tr><td></td><td>Message text</td><td>45</td><td>36-80</td></tr>
</table>
</td></tr>
<tr><td>LPAM</td><td>The Access Method subroutine for batch processing</td></tr>
<tr><td>LPAM Options</td><td>The LPAM options are:

• CLOSE
• GETCITY
• GETCNTL
• GETRNGE
• GETSTRT
• GETZIP
• INIT
• OPEN
• TERM</td></tr>
<tr><td>product</td><td>The value in this field is ignored.</td></tr>
<tr><td>profile</td><td>The value in this field is ignored.</td></tr>
</table>

Parameter	Description
rcb@	Doubleword filled in by the OPEN routine that acts as a resource control block for an opened file. You must not modify this field after the OPEN call. You can issue multiple opens against the same file. Finalist generates a new resource control block (rcb) after each request.
resource	The value in this field is ignored.
retcode	Two-character field for the value that the Access Method call returns. Check this field after each call. **00** Successful **04** EOF or non-fatal error occurred **08 or 12** Fatal error has occurred **Note:** If retcode is non-zero, errinfo contains specific diagnostic information.
retinfo	Return area for each specific function as defined in the Finalist copybook.

Copybooks

You can make Access Method call routines from a COBOL or Assembler program. This guide provides sample COBOL and CICS calling sequences. The MVS (batch) and CICS (MVS) environments

support the Access Method. The Finalist source library contains copybook members that you can use to create a driver program.

General Structure

To initiate a function, your driver program calls the LPAM (batch) or LPAMC (CICS) Access Method subroutine. This subroutine loads and calls the appropriate Access Method programs to complete the request.

Profile Tables

Specific profile tables are no longer supported.

CLOSE Function

The CLOSE function frees the RCB.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING CLOSE,anchor@,retcode,errinfo,rcb@.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING
DFHEIBLK,DFHCOMMAREA,CLOSE,anchor@,retcode,errinfo,rcb@.
```

Return Codes

Table 125: CLOSE Function Return Codes

Value	Description
4	Not used
8	Invalid parameter list Resource manager error

GETCNTL Function

The GETCNTL function retrieves relevant control information from a Finalist file and returns it to the driver program. The returned information includes file type, file version, and maintenance date.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING GETCNTL,anchor@,retcode,errinfo,rcb@,retinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,GETCNTL,anchor@,retcode,errinfo,
rcb@,retinfo.
```

Retinfo Layout

Table 126: Retinfo Layout

Physical Length	From/To Position	Field Description
8	1-8	File type: 'ZIPBASE '(Finalist Data File) 'CITYBASE' (Finalist City/State File)
1	9-9	File contents: **'F'** Full - ZIP + 4 and carrier route **' '** Blank if CITYBASE
1	10-10	File coverage: **'F'** Full country is in base **'R'** Regional base covers selected areas
4	11-14	File version (xxxx)
10	15-24	File maintenance date (mm/dd/yyyy)
10	25-34	Reserved for future expansion

Copybooks

Table 127: GETCNTL Copybooks

Type	COBOL Copybook	Assembler Copybook
Retinfo layout	LPAM006C	LPAM006M

Return Codes

Table 128: GETCNTL Return Codes

Value	Description
4	Not used
8	Invalid parameter list

GETCITY Function

The GETCITY function retrieves information about the requested city from the Finalist City/State File and returns it to the caller.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING GETCITY,anchor@,retcode,errinfo,rcb@,callinfo,retinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,GETCITY,anchor@,retcode,errinfo,
rcb@,callinfo,retinfo.
```

Callinfo Layout

Table 129: Callinfo Layout

Physical Length	From/To Position	Field Description
2	1-2	Target state. This field must contain the standard two-character state abbreviation code.
15	3-17	Target city name. The name can be generic from left to right or fully qualified.
5	18-22	Search control indicator for city name. Valid values are 'FIRST' and 'NEXT.'
1	23-23	Consider city name spelling variations in search. If you set this field to "Y", the program considers city name variations when searching the City/State File. To skip city name variations, set this field to "N".
3	24-26	Replication factor. Indicates how many city records it should retrieve (starting with the requested one). Note that additional contiguous retinfo areas must exist for each replication.
30	27-56	Reserved for future expansion.

Retinfo Layout

Table 130: Retinfo Layout

Physical Length	From/To Position	Field Description
3	1-3	Number of entries returned.
15	4-18	City name as it exists on the City/State File.
1	19-19	Returned city name. **P** Returned city name is a prime entry **V** Returned city name is a spelling variation
15	20-34	Prime city name.
5	35-39	ZIP Code of prime city name.

Physical Length	From/To Position	Field Description
5	40-44	County code of prime city name.
5	45-49	Reserved for future expansion.

Copybooks

Table 131: GETCITY Copybooks

Type	COBOL Copybook Member	Assembler Copybook Member
Callinfo layout	LPAM013C	LPAM013M
Retinfo layout	LPAM014C	LPAM014M

Return Codes

Table 132: GETCITY Return Codes

Value	Description
4	Replication incomplete
8	City not found Invalid parameter list
12	Pointing to wrong file (path)

Note: The GETCITY function sets the return code to "8" when it does not find the city. All other functions set the return code to "4".

GETRNGE Function

The GETRNGE function retrieves requested street range information from the Finalist Data File and returns it to the caller. Data returned includes street name, city, state, ZIP Code, low range, high range, pre- and post-directional, etc.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING GETRNGE,anchor@,retcode,errinfo,rcb@,callinfo,retinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,GETRNGE,anchor@,retcode,errinfo,
rcb@,callinfo,retinfo.
```

Callinfo Layout

Table 133: Callinfo Layout

Physical Length	From/To Position	Field Description
5	1-5	Target ZIP Code. Must be five digits.
25	6-30	Target street name. Must be fully qualified.
5	31-35	Search control indicator for street name. Valid values are 'FIRST' and 'NEXT'.
1	36-36	Consider street name spelling variations in search? If you set this field to "Y", the program considers street name variations when searching the Data File for a match against the input street name. If you set this field to "N", Finalist ignores street name variations on the Data File.
3	37-39	Replication factor. Indicates how many range records it should retrieve (starting with the requested one). Note that additional contiguous retinfo areas must exist for each replication.
30	40-69	Reserved for future expansion.

Retinfo Layout

Table 134: Retinfo Layout

Physical Length	From/To Position	Field Description
3	1-3	Number of entries returned.
5	4-8	ZIP Code.
1	9-9	**M** Returned ZIP Code is the main ZIP Code **N** Any other ZIP Code
5	10-14	ZIP Code (lowest ZIP Code in a USPS finance unit).
1	15-15	Multi-ZIP Code city? (Y or N).
1	16-16	Multi-city ZIP Code? (Y or N).
5	17-21	FIPS county code (from Data File).
15	22-36	City name USPS-preferred last line name (from Data File).
2	37-38	State name (from Data File).
8	39-46	Reserved for future expansion.
25	47-71	Returned street name.
1	72-72	Returned street name. **P** Prime entry **V** Spelling variation **F** Firm **A** Ambiguous (invalid) entry
25	73-97	Prime street name.
10	98-107	Reserved for future expansion.

Physical Length	From/To Position	Field Description
10	108-117	Primary low range.
10	118-127	Primary high range.
10	128-137	Secondary low range (unit numbers such as apartment or suite).
10	138-147	Secondary high range (unit numbers such as apartment or suite).
2	148-149	Pre-directional.
2	150-151	Post-directional.
4	152-155	Suffix #1.
4	156-159	Suffix #2.
4	160-163	Odd range sector segment.
4	164-167	Even range sector segment.
4	168-171	Odd range carrier route.
4	172-175	Even range carrier route.
2	176-177	Primary low-range alpha characters.
2	178-179	Primary high-range alpha characters.
4	180-183	Unit Designator (i.e., APT).
1	184-184	Alias street name flag (blank if not an alias): **A** Alias on secondary description (for example, Apt., Suite, etc.) **K** Nickname alias **P** Preferred alias
10	185-194	Reserved for future expansion.

Copybooks

Table 135: GETRNGE Copybooks

Type	COBOL Copybook Member	Assembler Copybook Member
Callinfo layout	LPAM011C	LPAM011M
Retinfo layout	LPAM012C	LPAM012M

Return Codes

Table 136: GETRNGE Return Codes

Value	Description
4	Range not found End of ranges for search Replication incomplete
8	Invalid parameter list

GETSTRT Function

The GETSTRT function retrieves the requested street names from the Finalist Data File and returns them to the caller. Data returned includes street, ZIP Code, city, state, county code, etc.

Note: Firm and high rise names are included in sequence with street names.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING GETSTRT,anchor@,retcode,errinfo,rcb@,callinfo,retinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,GETSTRT,anchor@,retcode,errinfo,
rcb@,callinfo,retinfo.
```

Callinfo Layout

Table 137: Callinfo Layout

Physical Length	From/To Position	Field Description
5	1-5	Target ZIP Code. Must be five-digit.
25	6-30	Target street name. Can be generic from left to right or fully qualified.
5	31-35	Search control indicator for street name. Valid values are 'FIRST' and 'NEXT'.
1	36-36	Consider street name spelling variations in search? If you set this field to "Y", the program considers street name variations when searching the Data File for a match against the input street name. If you set this field to "N", it ignores street name variations on the Data File.
3	37-39	Replication factor. Indicates how many street records it should retrieve (starting with the requested one). Note that additional contiguous retinfo areas must exist for each replication.
30	40-69	Reserved for future expansion

Retinfo Layout

Table 138: Retinfo Layout

Physical Length	From/To Position	Field Description
3	1-3	Number of entries returned
5	4-8	ZIP Code

Physical Length	From/To Position	Field Description
1	9-9	Returned ZIP Code is: **M** Main ZIP Code **N** Any other ZIP Code
5	10-14	ZIP Code (lowest ZIP Code in a USPS finance unit)
1	15-15	Multi-ZIP Code city? (Y or N)
1	16-16	Multi-city ZIP Code? (Y or N)
5	17-21	FIPS county code (from Data File)
15	22-36	City name USPS-preferred last line name (from Data File)
2	37-38	State name (from Data File)
8	39-46	Reserved for future expansion
25	47-71	Returned street name
1	72-72	Returned street name is: **P** Prime entry **V** Spelling variation **F** Firm **A** Ambiguous (invalid) entry **D** Alternate at delivery **H** High rise
25	73-97	Prime street name
10	98-107	Reserved for future expansion

Copybooks

Table 139: GETSTRT Copybooks

Type	COBOL Copybook Member	Assembler Copybook Member
Callinfo layout	LPAM009C	LPAM009M
Retinfo layout	LPAM010C	LPAM010M

Return Codes

Table 140: GETSTRT Return Codes

Value	Description
4	Street not found End of generic search Replication incomplete
8	Invalid parameter list

GETZIP Function

The GETZIP function retrieves ZIP Code information from the Finalist Data File and returns it to the driver program. The data returned includes ZIP Code, county code, city, state, flag to indicate multi-ZIP Code city, etc.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING GETZIP,anchor@,retcode,errinfo,rcb@,callinfo,retinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,GETZIP,anchor@,retcode,errinfo,
rcb@,callinfo,retinfo.
```

Callinfo Layout

Table 141: GETZIP Callinfo Layout

Physical Length	From/To Position	Field Description
5	1-5	Target ZIP Code. Can be generic from left to right (605*) or fully qualified.
5	6-10	Search control indicator for ZIP Code. Valid values are 'FIRST' and 'NEXT'.
3	11-13	Replication factor. Indicates how many ZIP Code records it should retrieve (starting with the requested one). Note that additional contiguous retinfo areas must exist for each replication.
30	14-43	Reserved for future expansion.

Retinfo Layout

Table 142: GETZIP Retinfo Layout

Physical Length	From/To Position	Field Description
3	1-3	Number of entries returned
5	4-8	ZIP Code
1	9-9	The returned ZIP Code is: **M** Main ZIP Code **N** Any other ZIP Code
5	10-14	ZIP Code (lowest ZIP Code in a USPS finance unit)
1	15-15	Multi-ZIP Code city? (Y or N)

Physical Length	From/To Position	Field Description
1	16-16	Multi-city ZIP Code? (Y or N)
5	17-21	FIPS county code (from Data File)
15	22-36	City name (USPS-preferred last-line name from Data File)
2	37-38	State name (from Data File)
8	39-46	Reserved for future expansion

Copybooks

Table 143: GETZIP Copybooks

Type	COBOL Copybook	Assembler Copybook
Callinfo layout	LPAM007C	LPAM007M
Retinfo layout	LPAM008C	LPAM008M

Return Codes

Table 144: GETZIP Return Codes

Value	Description
4	End of generic search request Replication incomplete ZIP Code not found
8	Invalid parameter list

INIT Function

When you use the database access method, the first call required is the INIT function. The database access method performs all one-time processing, such as acquiring memory for internal work areas. After processing, the physical address of these areas are saved in a fullword known as an anchor pointer. The driver program provides the anchor pointer when it initiates the INIT function and passes this address on all subsequent database access method calls. After the INIT call, you must not modify the contents of this address.

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING INIT,anchor@,retcode,errinfo,profile.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING
DFHEIBLK,DFHCOMMAREA,INIT,anchor@,retcode,errinfo,profile.
```

Note: You must not modify the anchor@ field. The database access method uses the anchor@ field on all subsequent calls. The profile field value is no longer supported and will be ignored.

Return Codes

Table 145: INIT Function Return Codes

Value	Description
8	Invalid parameter list Not enough main storage available Access method already initialized

OPEN Function

The OPEN function establishes a path to a resource (in this case, a file). The database access method places the address of the internal control block that manages activity against the resource in the RCB@ field. The caller must not modify the RCB@ field after the OPEN call. The database

access method uses the RCB@ field for all subsequent database access method calls against the resource (i.e., GETCNTL, GETZIP, etc.).

COBOL Batch Calling Sequence

```
CALL 'LPAM' USING OPEN,anchor@,retcode,errinfo,rcb@,product,resource.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,OPEN,anchor@,retcode,errinfo,rcb@,
product,resource.
```

Note: The product and resource field values are no longer supported and will be ignored.

Return Codes

Table 146: OPEN Function Return Codes

Value	Description
4	Not used
8	Invalid parameter list Resource manager error

TERM Function

This function terminates Access Method processing. Calling the TERM function performs an orderly shutdown of all processing. The program closes any opened files and releases all reserved storage.

COBOL Batch Calling Sequences

```
CALL 'LPAM' USING TERM,anchor@,retcode,errinfo.
```

COBOL CICS Calling Sequence

```
CALL 'LPAMC' USING DFHEIBLK,DFHCOMMAREA,TERM,anchor@,retcode,errinfo.
```

Sample Program

The source file on the mainframe distribution includes a sample COBOL program. The sample program uses the 'OPEN', 'CLOSE', and 'GETRNGE' functions. The program name is LPAMCTST. The mapset name is COBMAP.

Using the Database APIs for IMS

LPAMI, the access method interface module, allows you to access Finalist files such as the City/State or Data Files in an IMS on-line environment.

> **Note:** Throughout this chapter, LPAMI and LPAM are used interchangeably. However, please note that while the batch interface to LPAM and the IMS interface to LPAMI are similar, the parameter lists are not the same. Verify you are using the correct parameter list when calling IMS LPAMI and batch LPAM.
>
> An INIT and a TERM call must be done. Failure to do so could result in a wide variety of abends.

You can use the access method in an IMS or batch environment to obtain data from either of the Finalist Data Files.

Summary of Parameters

Use the following parameters to call the access method:

- INPUT — Address of the LPAM input call area
- OUTPUT — Address of the LPAM output area

There is one input call area for all LPAM functions. The data that passes to LPAM varies depending on the function you want to invoke. A unique output area for each of the various LPAM functions

accommodates the different types of data returned. The first 54 bytes of each of the return areas are identical. This area has the following layout.

```
LPAM-RETURN-CODE   PIC X(02).
LPAM-ERRMOD        PIC X(08).
LPAM-ERRMSG        PIC X(44).
```

The LPAM-RETURN-CODE field contains the return code from the invoked function. Check this field after each call. The following table displays the possible values for the LPAM-RETURN-CODE field.

Table 147: Possible Values for Function Return Code

Value	Description
Value	Description
00	Successful
04	EOF or non-fatal error occurred
08 or 12	Invalid parameter list City not found

Note: If retcode is non-zero, errinfo contains specific diagnostic information. In the event of an error, the LPAM-ERRMOD field contains the name of the module that detected the error. The LPAM-ERRMSG field contains a message describing the error.

Access Method Calls

You can make access method calls from any language that uses standard IBM linkage conventions. Sample Assembler and COBOL calling sequences are provided.

General Structure

To initiate a function, your driver program calls the Access Method subroutine, LPAMI. This subroutine then loads and calls the appropriate Access Method programs to complete the request.

GETCNTL Function

The GETCNTL function retrieves relevant control information from a Finalist file and returns it to the driver program. Return information includes file type, file version, and maintenance date.

IMS Calling Sequence

CALL 'LPAMI' USING LPAMINC, LPAMCCTL.

Input Layout

```
01  LPAM-INPUT.
    05  LPAM-INPUT-FUNCTION          PIC X(08).
    05  LPAM-INPUT-TYPE              PIC X(05).
      88 FIRST-REQUEST         VALUE 'FIRST'.
      88 NEXT-REQUEST          VALUE 'NEXT '.
    05  LPAM-INPUT-VARIATIONS        PIC X(01).
    05  LPAM-INPUT-REPZIP.
        10  LPAM-INPUT-REPLICATION   PIC X(03).
        10  LPAM-INPUT-TARGET-ZIP    PIC X(05).
    05  LPAM-INPUT-PCBS REDEFINES LPAM-INPUT-REPZIP.
        10  LPAM-INPUT-DATPCB        PIC S9(5) COMP.
        10  LPAM-INPUT-CTYPCB        PIC S9(5) COMP.
    05  LPAM-INPUT-TARGET-STREET     PIC X(25).
    05  LPAM-INPUT-TARGET-STATE      PIC X(02).
    05  LPAM-INPUT-TARGET-CITY       PIC X(15).
    05  LPAM-INPUT-RESERVED          PIC X(30).
    05  LPAM-INPUT-RESERVED-2        PIC X(32).
```

Set the following field before you issue a call to the GETCNTL function.

LPAM-INPUT-FUNCTION	Must contain "GETCNTL ". The LPAM GETCNTL output follows.

Output

```
01  LPAM-GETCNTL-OUTPUT.
    05  LPAM-GETCNTL-ERROR-INFO.
      10 LPAM-GETCNTL-RC                  PIC X(02).
      10 LPAM-GETCNTL-ERRMOD              PIC X(08).
      10 LPAM-GETCNTL-ERRMSG              PIC X(44).
    05  LPAM-GETCNTLD-TYPE                PIC X(08).
    05  LPAM-GETCNTLD-CONTENTS            PIC X(01).
    05  LPAM-GETCNTLD-COVERAGE            PIC X(01).
    05  LPAM-GETCNTLD-VERSION             PIC X(04).
    05  LPAM-GETCNTLD-MAINT-DATE          PIC X(08).
    05  LPAM-GETCNTLD-RESERVED            PIC X(12).
    05  LPAM-GETCNTLC-TYPE                PIC X(08).
    05  LPAM-GETCNTLC-CONTENTS            PIC X(01).
    05  LPAM-GETCNTLC-COVERAGE            PIC X(01).
    05  LPAM-GETCNTLC-VERSION             PIC X(04).
    05  LPAM-GETCNTLC-MAINT-DATE          PIC X(08).
    05  LPAM-GETCNTLC-RESERVED            PIC X(12).
```

Copybooks

Table 148: GETCNTL Copybooks

Type	COBOL Copybook	Assembler Copybook
Input	LPAMINC	LPAMIN
Output	LPAMCCTL	LPAMACTL

Return Codes

Table 149: GETCNTL Return Codes

Value	Description
4	Not used
8	Invalid parameter list

Generic Search Criteria

The next four sections explain the GETCITY, GETRNGE, GETSTRT, and GETZIP functions. When you use these functions, you must supply search criteria, such as a target ZIP Code or target city name. You can use fully qualified search criteria (for example, a five-digit ZIP Code) or a generic qualifier. You can specify generic qualifiers from left to right only. For example, 605* is a valid generic ZIP Code, but *0137 is not a valid generic ZIP Code.

After the criteria fields (for example, ZIP Code, city name), there is a five-character field for you to specify how to use the supplied search criteria. For example, if you specify 'FIRST' in this field, Finalist will return the first item on the file that matches the supplied criteria. If you specify 'NEXT ' in this field and a previous call was made, Finalist returns the next sequential item.

Example

The following examples use generic qualifiers.

Requested ZIP	Indicator	Action
60137	FIRST	Finalist returns ZIP Code 60137.
60137	NEXT	Finalist returns that ZIP. If you made a previous request for 60137, Finalist returns 60138. If you make another call, Finalist returns 60139.
605*	FIRST	Finalist returns the first valid ZIP Code for SCF 605.
605*	NEXT	If this is the first request with 605* specified as the target ZIP Code, the program returns the first valid ZIP Code for SCF 605. If you made a previous request with 605* specified as the target ZIP Code, this request will return the next valid ZIP Code for SCF 605.

Note: Unless field LPAM-INPUT-VARIATIONS is set to "Y", entries in callinfo or retinfo refer to the primary spelling of a city or street name rather than a variation.

GETCITY Function

The GETCITY function retrieves information about the requested city from the Finalist City/State File and returns the information to the caller.

IMS Calling Sequences

```
CALL 'LPAMI' USING LPAMINC, LPAMCCTY.
```

Input Layout

```
  01  LPAM-INPUT.
      05  LPAM-INPUT-FUNCTION          PIC X(08).
      05  LPAM-INPUT-TYPE              PIC X(05).
        88 FIRST-REQUEST         VALUE 'FIRST'.
        88 NEXT-REQUEST          VALUE 'NEXT '.
      05  LPAM-INPUT-VARIATIONS        PIC X(01).
      05  LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-REPLICATION   PIC X(03).
          10  LPAM-INPUT-TARGET-ZIP    PIC X(05).
      05  LPAM-INPUT-PCBS REDEFINES LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-DATPCB        PIC S9(5) COMP.
          10  LPAM-INPUT-CTYPCB        PIC S9(5) COMP.
      05  LPAM-INPUT-TARGET-STREET     PIC X(25).
      05  LPAM-INPUT-TARGET-STATE      PIC X(02).
      05  LPAM-INPUT-TARGET-CITY       PIC X(15).
      05  LPAM-INPUT-RESERVED          PIC X(30).
      05  LPAM-INPUT-RESERVED-2        PIC X(32).
```

Set the following fields before issuing a call to the GETCITY function.

Field	Description
LPAM-INPUT-FUNCTION	Must be set to "GETCITY ".
LPAM-INPUT-TARGET-STATE	This field must contain the standard two-character state abbreviation.
LPAM-INPUT-TARGET-CITY	The name can be generic from left to right or fully qualified.
LPAM-INPUT-TYPE	Valid values are "FIRST" and "NEXT ".

Field	Description
LPAM-INPUT-VARIATIONS	Consider city name spelling variations in search. If you set this field "Y", the program considers city name variations when searching the City/State File. To skip city name variations, set this field to "N".
LPAM-INPUT-REPLICATION	Indicates how many city records it should retrieve (starting with the requested one). Note that additional contiguous retinfo areas must exist for each replication.
LPAM-INPUT-RESERVED	The access method uses this area to save data and pointers between calls when "NEXT" processing is being used. If you do not maintain this area, "NEXT" processing will not function properly.
LPAM-INPUT-RESERVED-2	The access method uses this area to maintain pointers to loaded modules and getmain areas. If this area is not maintained between calls, the access method will perform additional getmains and program loads. Previous getmain areas and programs remain loaded possibly resulting in storage shortage problems over time. Results of "NEXT" processing may be unpredictable.

Sample Call

```
MOVE SPACES                TO LPAM-INPUT.
MOVE 'GETCITY '            TO LPAM-INPUT-FUNCTION.
MOVE 'FIRST'               TO LPAM-INPUT-TYPE.
MOVE '010'                 TO LPAM-INPUT-REPLICATION.
MOVE 'IL'                  TO LPAM-INPUT-TARGET-STATE.
MOVE 'CHICAGO'             TO LPAM-INPUT-TARGET-CITY.
MOVE WS-SAVE-LPAM-RES      TO LPAM-INPUT-RESERVED.
MOVE WS-SAVE-LPAM-RES-2    TO LPAM-INPUT-RESERVED-2.
CALL 'LPAMI'               USING LPAMINC,LPAMCCTY.
```

Output

```
  01  LPAM-GETCITY-OUTPUT.
      05  LPAM-ERROR-INFO.
        10 LPAM-GETCITY-RC                  PIC X(02).
        10 LPAM-GETCITY-ERRMOD              PIC X(08).
        10 LPAM-GETCITY-ERRMSG              PIC X(44).
      05  LPAM-GETCITY-RETURNED             PIC X(03).
      05  LPAM-GETCITY-DATA.
          10  LPAM-GETCITY-RCITY            PIC X(15).
          10  LPAM-GETCITY-CIND             PIC X(01).
          10  LPAM-GETCITY-PCITY            PIC X(15).
          10  LPAM-GETCITY-MZIP             PIC X(05).
```

```
          10  LPAM-GETCITY-CNTY            PIC X(05).
          10  FILLER                       PIC X(05).
```

Copybooks

Table 150: GETCITY Copybooks

Type	COBOL Copybook	Assembler Copybook
Input	LPAMINC	LPAMIN
Output	LPAMCCTY	LPAMACTY

Return Codes

Table 151: GETCITY Return Codes

Value	Description
4	Replication incomplete (i.e., if you requested 10 cities starting with "Z" and found only eight, you would receive a return code of 4)
8	City not found Invalid parameter list

GETRNGE Function

The GETRNGE function retrieves requested street range information from the Finalist Data File and returns it to the caller.

COBOL IMS Calling Sequence

```
CALL 'LPAMI' USING LPAMINC, LPAMCRNG.
```

Input Layout

```
01  LPAM-INPUT.
    05  LPAM-INPUT-FUNCTION          PIC X(08).
    05  LPAM-INPUT-TYPE              PIC X(05).
      88 FIRST-REQUEST         VALUE 'FIRST'.
      88 NEXT-REQUEST          VALUE 'NEXT '.
    05  LPAM-INPUT-VARIATIONS        PIC X(01).
    05  LPAM-INPUT-REPZIP.
        10  LPAM-INPUT-REPLICATION   PIC X(03).
        10  LPAM-INPUT-TARGET-ZIP    PIC X(05).
    05  LPAM-INPUT-PCBS REDEFINES LPAM-INPUT-REPZIP.
        10  LPAM-INPUT-DATPCB        PIC S9(5) COMP.
        10  LPAM-INPUT-CTYPCB        PIC S9(5) COMP.
    05  LPAM-INPUT-TARGET-STREET     PIC X(25).
    05  LPAM-INPUT-TARGET-STATE      PIC X(02).
    05  LPAM-INPUT-TARGET-CITY       PIC X(15).
    05  LPAM-INPUT-RESERVED          PIC X(30).
    05  LPAM-INPUT-RESERVED-2        PIC X(32).
```

Set the following fields before you issue a call to the GETRNGE function.

Field	Description
LPAM-INPUT-FUNCTION	Must be set to "GETRNGE ".
LPAM-INPUT-TARGET-ZIP	Must be fully qualified.
LPAM-INPUT-TARGET-STREET	Must be fully qualified.
LPAM-INPUT-TYPE	Valid values are "FIRST" and "NEXT ".
LPAM-INPUT-VARIATIONS	Consider street name spelling variations in search. If you set this field to "Y", the program considers street name variations when searching the Data File for a match against the input street name. If you set this field to "N", Finalist ignores street name variations on the Data File.
LPAM-INPUT-REPLICATION	Indicates how many range records it should retrieve (starting with the requested one). Note that additional contiguous return areas must exist for each replication.
LPAM-INPUT-RESERVED	The access method uses this area to save data and pointers between calls when "NEXT" processing is being used. If you do not maintain this area, "NEXT" processing will not function properly.

Field	Description
LPAM-INPUT-RESERVED-2	The access method uses this area to maintain pointers to loaded modules and getmain areas. If this area is not maintained between calls, the access method will perform additional getmains and program loads. Previous getmain areas and programs remain loaded possibly resulting in storage shortage problems over time. Results of "NEXT" processing may be unpredictable.

Sample Call

```
MOVE SPACES                TO LPAM-INPUT.
MOVE 'GETRNGE '            TO LPAM-INPUT-FUNCTION.
MOVE 'FIRST'               TO LPAM-INPUT-TYPE.
MOVE '010'                 TO LPAM-INPUT-REPLICATION.
MOVE '60187'               TO LPAM-INPUT-TARGET-ZIP.
MOVE 'MAIN'                TO LPAM-INPUT-TARGET-STREET.
MOVE WS-SAVE-LPAM-RES      TO LPAM-INPUT-RESERVED.
MOVE WS-SAVE-LPAM-RES-2    TO LPAM-INPUT-RESERVED-2.
CALL 'LPAMI'               USING LPAMINC, LPAMCRNG.
```

Output

```
  01  LPAM-GETRNGE-OUTPUT.
      05  LPAM-GETRNGE-ERROR-INFO.
        10 LPAM-GETRNGE-RC            PIC X(02).
        10 LPAM-GETRNGE-ERRMOD        PIC X(08).
        10 LPAM-GETRNGE-ERRMSG        PIC X(44).
      05  LPAM-GETRNGE-RETURNED       PIC X(03).
    03  LPAM-GETRNGE-DATA.
      05  LPAM-GETRNGE-ZDATA.
          10  LPAM-GETRNGE-RZIP       PIC X(05).
          10  LPAM-GETRNGE-ZIPTYP     PIC X(01).
          10  LPAM-GETRNGE-MZIP       PIC X(05).
          10  LPAM-GETRNGE-MZCIND     PIC X(01).
          10  LPAM-GETRNGE-MCZIND     PIC X(01).
          10  LPAM-GETRNGE-RCOUNTY    PIC X(05).
          10  LPAM-GETRNGE-RCITY      PIC X(15).
          10  LPAM-GETRNGE-RSTATE     PIC X(02).
          10  FILLER                  PIC X(08).
      05  LPAM-GETRNGE-SDATA.
          10  LPAM-GETRNGE-RSTREET    PIC X(25).
          10  LPAM-GETRNGE-PVIND      PIC X(01).
          10  LPAM-GETRNGE-PSTREET    PIC X(25).
          10  FILLER                  PIC X(10).
      05  LPAM-GETRNGE-RDATA.
          10  LPAM-GETRNGE-PLOW       PIC X(10).
          10  LPAM-GETRNGE-PHI        PIC X(10).
```

```
           10  LPAM-GETRNGE-SLOW       PIC X(10).
           10  LPAM-GETRNGE-SHI        PIC X(10).
           10  LPAM-GETRNGE-DIR        PIC X(04).
           10  LPAM-GETRNGE-SFX1       PIC X(04).
           10  LPAM-GETRNGE-SFX2       PIC X(04).
           10  LPAM-GETRNGE-OSS        PIC X(04).
           10  LPAM-GETRNGE-ESS        PIC X(04).
           10  LPAM-GETRNGE-OCAR       PIC X(04).
           10  LPAM-GETRNGE-ECAR       PIC X(04).
           10  LPAM-GETRNGE-PLALPH     PIC X(02).
           10  LPAM-GETRNGE-PHALPH     PIC X(02).
           10  FILLER                  PIC X(15).
```

Copybooks

Table 152: GETRNGE Copybooks

Type	COBOL Copybook	Assembler Copybook
Input	LPAMINC	LPAMIN
Output	LPAMCRNG	LPAMARNG

Return Codes

Table 153: GETRNGE Return Codes

Value	Description
4	Range not found End of ranges for search Replication incomplete. (i.e., 12 ranges requested - only 10 available)
8	Invalid parameter list

GETSTRT Function

The GETSTRT function retrieves the requested street names from the Finalist Data File and returns the street names to the caller.

Note: Firm and high rise names are included sequentially with street names.

COBOL IMS Calling Sequences

```
CALL 'LPAMI' USING LPAMINC, LPAMCSTR.
```

Input Layout

```
  01  LPAM-INPUT.
      05  LPAM-INPUT-FUNCTION          PIC X(08).
      05  LPAM-INPUT-TYPE              PIC X(05).
        88 FIRST-REQUEST         VALUE 'FIRST'.
        88 NEXT-REQUEST          VALUE 'NEXT '.
      05  LPAM-INPUT-VARIATIONS        PIC X(01).
      05  LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-REPLICATION   PIC X(03).
          10  LPAM-INPUT-TARGET-ZIP    PIC X(05).
      05  LPAM-INPUT-PCBS REDEFINES LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-DATPCB        PIC S9(5) COMP.
          10  LPAM-INPUT-CTYPCB        PIC S9(5) COMP.
      05  LPAM-INPUT-TARGET-STREET     PIC X(25).
      05  LPAM-INPUT-TARGET-STATE      PIC X(02).
      05  LPAM-INPUT-TARGET-CITY       PIC X(15).
      05  LPAM-INPUT-RESERVED          PIC X(30).
      05  LPAM-INPUT-RESERVED-2        PIC X(32).
```

Set the following fields before you issue a call to the GETSTRT function.

Field	Description
LPAM-INPUT-FUNCTION	Must be set to "GETSTRT ".
LPAM-INPUT-TARGET-ZIP	Must be fully qualified.
LPAM-INPUT-TARGET-STREET	Can be generic from left to right or fully qualified.
LPAM-INPUT-TYPE	Valid values are "FIRST" and "NEXT".

Field	Description
LPAM-INPUT-VARIATIONS	Consider street name spelling variations in search? If you set this field to "Y", the program considers street name variations when searching the Data File for a match against the input street name. If you set this field to "N", it ignores street name variations on the Data File.
LPAM-INPUT-REPLICATION	Indicates how many street records it should retrieve (starting with the requested one). Note that additional contiguous output areas must exist for each replication.
LPAM-INPUT-RESERVED	The access method uses this area to save data and pointers between calls when you use "NEXT" processing. If you do not maintain this area, "NEXT" processing will not function properly.
LPAM-INPUT-RESERVED-2	The access method uses this area to maintain pointers to loaded modules and getmain areas. If this area is not maintained between calls, the access method will perform additional getmains and program loads. Previous getmain areas and programs remain loaded possibly resulting in storage shortage problems over time. Results of "NEXT" processing may be unpredictable.

Note: Finalist returns all streets belonging to the finance unit of the target ZIP Code.

Output Layout

```
01  LPAM-GETSTRT-OUTPUT.
    05  LPAM-GETSTRT-ERROR-INFO.
        10 LPAM-GETSTRT-RC            PIC X(02).
        10 LPAM-GETSTRT-ERRMOD        PIC X(08).
        10 LPAM-GETSTRT-ERRMSG        PIC X(44).
    05  LPAM-GETSTRT-RETURNED         PIC X(03).
  03  LPAM-GETSTRT-DATA.
    05  LPAM-GETSTRT-ZDATA.
        10 LPAM-GETSTRT-RZIP          PIC X(05).
        10 LPAM-GETSTRT-ZIPTYP        PIC X(01).
        10 LPAM-GETSTRT-MZIP          PIC X(05).
        10 LPAM-GETSTRT-MZCIND        PIC X(01).
        10 LPAM-GETSTRT-MCZIND        PIC X(01).
        10 LPAM-GETSTRT-RCOUNTY       PIC X(05).
        10 LPAM-GETSTRT-RCITY         PIC X(15).
        10 LPAM-GETSTRT-RSTATE        PIC X(02).
        10 FILLER                     PIC X(08).
    05  LPAM-GETSTRT-SDATA.
        10  LPAM-GETSTRT-RSTREET      PIC X(25).
        10  LPAM-GETSTRT-PVIND        PIC X(01).
        10  LPAM-GETSTRT-PSTREET      PIC X(25).
        10  FILLER                    PIC X(10).
```

Copybooks

Table 154: GETSTRT Copybooks

Type	COBOL Copybook Member	Assembler Copybook Member
Input	LPAMINC	LPAMIN
Output	LPAMCSTR	LPAMASTR

Return Codes

Table 155: GETSTRT Return Codes

Value	Description
4	Street not found End of generic search Replication incomplete (i.e., 10 streets requested - only eight available)
8	Invalid parameter list

GETZIP Function

The GETZIP function retrieves ZIP Code information from the Finalist Data File and returns it to the driver program.

COBOL IMS Calling Sequences

```
CALL 'LPAMI' USING LPAMINC, LPAMCZIP.
```

Input Layout

```
  01  LPAM-INPUT.
      05  LPAM-INPUT-FUNCTION          PIC X(08).
      05  LPAM-INPUT-TYPE              PIC X(05).
        88 FIRST-REQUEST          VALUE 'FIRST'.
        88 NEXT-REQUEST           VALUE 'NEXT '.
```

```
      05  LPAM-INPUT-VARIATIONS        PIC X(01).
      05  LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-REPLICATION   PIC X(03).
          10  LPAM-INPUT-TARGET-ZIP    PIC X(05).
      05  LPAM-INPUT-PCBS REDEFINES LPAM-INPUT-REPZIP.
          10  LPAM-INPUT-DATPCB        PIC S9(5) COMP.
          10  LPAM-INPUT-CTYPCB        PIC S9(5) COMP.
      05  LPAM-INPUT-TARGET-STREET     PIC X(25).
      05  LPAM-INPUT-TARGET-STATE      PIC X(02).
      05  LPAM-INPUT-TARGET-CITY       PIC X(15).
      05  LPAM-INPUT-RESERVED          PIC X(30).
      05  LPAM-INPUT-RESERVED-2        PIC X(32).
```

You must set the following fields before you issue a call to the GETZIP function.

Field	Description
LPAM-INPUT-FUNCTION	Must be set to "GETZIP ".
LPAM-INPUT-TYPE	Must be "FIRST" or "NEXT".
LPAM-INPUT-TARGET-ZIP	Can be generic from left to right (605*) or fully qualified.
LPAM-INPUT-REPLICATION	Indicates how many ZIP Code records it should retrieve (starting with the requested one). Note that additional contiguous input areas must exist for each replication.
LPAM-INPUT-RESERVED	The access method uses this area to save data and pointers between calls when "NEXT" processing is being used. If you do not maintain this area, "NEXT" processing will not function properly.
LPAM-INPUT-RESERVED-2	The access method uses this area to maintain pointers to loaded modules and getmain areas. If this area is not maintained between calls, the access method will perform additional getmains and program loads. Previous getmain areas and programs remain loaded possibly resulting in storage shortage problems over time. Results of "NEXT" processing may be unpredictable.

Output Layout

```
  01  LPAM-GETZIP-OUTPUT.
      05  LPAM-GETZIP-ERROR-INFO.
        10 LPAM-GETZIP-RC           PIC X(02).
        10 LPAM-GETZIP-ERRMOD       PIC X(08).
        10 LPAM-GETZIP-ERRMSG       PIC X(44).
      05  LPAM-GETZIP-RETURNED      PIC X(03).
```

```
      05  LPAM-GETZIP-DATA.
        10 LPAM-GETZIP-RZIP          PIC X(05).
        10 LPAM-GETZIP-MZIPI         PIC X(01).
        10 LPAM-GETZIP-MZIP          PIC X(05).
        10 LPAM-GETZIP-MZCIND        PIC X(01).
        10 LPAM-GETZIP-MCZIND        PIC X(01).
        10 LPAM-GETZIP-RCNTY         PIC X(05).
        10 LPAM-GETZIP-RCITY         PIC X(15).
        10 LPAM-GETZIP-RSTATE        PIC X(02).
        10 FILLER                    PIC X(08).
```

Copybooks

Table 156: GETZIP Copybooks

Type	COBOL Copybook Member	Assembler Copybook Member
Input	LPAMINC	LPAMIN
Output	LPAMCZIP	LPAMAZIP

Sample Call

```
MOVE SPACES                 TO LPAM-INPUT.
MOVE 'GETZIP  '             TO LPAM-INPUT-FUNCTION.
MOVE 'FIRST'                TO LPAM-INPUT-TYPE.
MOVE '010'                  TO LPAM-INPUT-REPLICATION.
MOVE '60187'                TO LPAM-INPUT-TARGET-ZIP.
MOVE WS-SAVE-LPAM-RES       TO LPAM-INPUT-RESERVED.
MOVE WS-SAVE-LPAM-RES-2     TO LPAM-INPUT-RESERVED-2.
CALL 'LPAMI'                USING LPAMINC, LPAMCZIP.
```

Return Codes

Table 157: GETZIP Return Codes

Value	Description
4	End of generic search request Replication incomplete (i.e., 10 ZIP Codes requested-only eight available) ZIP not found
8	Invalid parameter list

INIT Function

The INIT function establishes a working environment for access method processing.

COBOL IMS Calling Sequence

```
MOVE 'INIT    '                 TO LPAM-INPUT-FUNCTION.
MOVE 'N'                        TO LPAM-INPUT-VARIATIONS.
MOVE IMS-INITDAT                TO LPAM-INPUT-DATPCB.
MOVE IMS-INITCITY               TO LPAM-INPUT-CTYPCB.
CALL 'LPAMI' USING LPAM-INPUT, LPAM-OUTPUT.
```

TERM Function

The TERM function will require a working environment for access method processing.

COBOL IMS Calling Sequence

```
MOVE 'TERM    '                 TO LPAM-INPUT-FUNCTION.
MOVE 'N'                        TO LPAM-INPUT-VARIATIONS.
MOVE IMS-INITDAT                TO LPAM-INPUT-DATPCB.
MOVE IMS-INITCITY               TO LPAM-INPUT-CTYPCB.
CALL 'LPAMI' USING LPAM-INPUT, LPAM-OUTPUT.
```

Using the Compatibility Interface Product APIs

The Finalist distribution contains the following copybooks for the Finalist call area:

- **LPFNCL01** - COBOL copybook
- **LPFNCL0C** - C copybook
- **LPFNCL04** - Assembler copybook

> **Note:** The LPFNCL01 COBOL copybook references both PBFN and PBNF structures. This is not an error. The PBNF structures do not include fixed version and identifier information as COBOL does not support those items as part of a redefined structure. The lack of version and identifier information in the copybook does not impact the copybook usage as the Compatibility Interface places the respective information into the PBNF structures for you.

Finalist Call Area - Function 0 Init Call

Table 158: Finalist Call Area - Function 0 Init Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-RETURN-CODE	Defines a general return code that describes the success or failure of the Initialization call. A return code of less than FINALIST-RC-PBFN-SUCCESS indicates that Finalist failed to initialize. Finalist should be terminated and the log files should be queried for complete details. A return code greater than FINALIST-RC-PBFN-SUCCESS indicates a warning condition that may require your immediate attention. Finalist log files should be queried to determine if you consider this a successful initialization. A return code equal to FINALIST-RC-PBFN-SUCCESS indicates Finalist was able to successfully initialize.
FINAL-CALL-AREA FinalistCallArea	Defines the entire call and return areas necessary for processing Finalist.
FINAL-FILLER LAnchorPointers	Contains the fullwords used to carry the address of dynamically acquired storage from one Finalist call to the next. You must initialize the fullwords to HIGH-VALUES (X'FF' or Binary-1) on the initialization call to Finalist. This prompts Finalist to acquire storage and address it with the anchors. You must not alter the anchor area after the initialization call.
FINAL-FUNCTION-AREA	The first byte of this field indicates which function to perform. The three remaining bytes are reserved for Precisely use only. The first digit is required.
FINAL-FUNCTION-CODE cFunctionCode[0]	Determines the function to be performed when Finalist is called.
FINAL-FUNCTION-OPTION	This field is reserved for future use.
CALL-INIT CallInit	A group definition consisting of fields INITMOD and INITBUFF (caInitMod and lInitBuff).

COBOL Field Name/ C Field Name	Field Description
INITMOD caInitMod	Applies to batch processing only. INITMOD specifies the name of the Finalist processing module. The default value is LPFNMODS. Finalist uses this field only on the initialization call.
INITBUFF LInitBuff	Applies to batch processing only. INITBUFF specifies the number of I/O buffers to use during Finalist processing. The default is 65. The minimum is 6. The field is populated only on the initialization call.
CALL-CICS CallCics	Can contain optional override parameters for on-line (CICS) Finalist execution. This field is a redefinition of field INITMOD and populated only on the initialization call.
CAMODNAM caModName	Applies to CICS users only. CAMODNAM specifies the name of the on-line Finalist processing module to use. The default value is LPFNMODC. The field is populated only on the initialization call.
CAEXCPSW cExcpSw	Applies to CICS users only. CAEXCPSW indicates exceptions table processing is used. The field is populated only on the initialization call.
CALL-IMS CallIms	This group definition applies to users of Finalist/IMS only. This field is populated only on the initialization call.
IMS-INITMOD	Reserved for future use.
IMS-INITDAT pPSBData	Applies only to Finalist/IMS users. You must supply the PCB address in this field when calling the Finalist interface.
IMS-INITCITY pPSBCity	Applies only to Finalist/IMS users. You must supply the PCB address in this field when calling the Finalist interface.
IMS-ALT-BLKSZ LAltBlkSz	Not currently used.
FINAL-CNFIG-ID caConfigID	Contains the configuration value for processing. The format is "CNFIGxxx" replacing "xxx" with a valid CASS-certified configuration setting. This field is populated only on the initialization call. Valid values are: **CNFIGAAJ** Sets FINAL-CTYLONG-OPT to “N”. **CNFIGAAR** Sets FINAL-CTYLONG-OPT to “Y”.
INITCITY cInitCity	Not currently used.

COBOL Field Name/ C Field Name	Field Description
FINAL-TAILOR-OPTS TailoringOptions	Upon the initial call to Finalist, use this field to set the first eight tailoring options for processing. **Note:** This field occupies the same storage as FINAL-CNFIG-ID. Do not use this field if you have defined FINAL-CNFIG-ID.
FINAL-UNIQUE-OPT cTUniqueZip	No longer used.
FINAL-STRTPHON-OPT cTStreetPhonetics	No longer used.
FINAL-FIRMCORR-OPT cTFirmCorrection	No longer used.
FINAL-CITYPHON-OPT cTCityPhonetics	No longer used.
FINAL-WEIGHT-OPT cTWeighting	No longer used.
FINAL-ZIPCORR-OPT cTZipCorrection	No longer used.
FINAL-CITYCORR-OPT cTCityCorrection	No longer used.
FINAL-STRCOSM-OPT cTStreetCosmetics	No longer used.
FINAL-FRMPRS-OPT cTFirmParse	No longer used.
FINAL-UNITDES-OPT cTUnitDesignator	No longer used.

COBOL Field Name/ C Field Name	Field Description
FINAL-CTYLONG-OPT cTLongCityName	Indicates whether Finalist should return a long version (25 bytes) of the input city name. The field is populated only on the initialization call. Valid values are "Y" or "N".
FINAL-ALSLBL-OPT cTAliasLabelLine	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. • **1 (same as the Native Interface value 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. • **2 (same as the Native Interface value 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. • **4 (same as the Native Interface value 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. • **5 (same as the Native Interface value XP)** - 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. • **6 (same as the Native Interface value 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. • **N | 3 (same as the Native Interface value P)** - If a Preferred alias exists, return the Preferred alias. Otherwise, return the base street. • **Y (same as the Native Interface value 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. • **Blank** - Defaults to N.
FINAL-EXPANDED-INFO ExpandedInfo	Defines the expanded call area in Finalist.
FINAL-STAT-REPORT-OPTIONS caOptions	Defines the statistical report generation for Finalist. The default value Y prints all sections of the Processing Statistics Report. To suppress printing specific sections, use a value of 'N(+ the section number(s) to suppress)'. To suppress printing all sections of this report, use the value 'N123456'. Set this value on the initialization call to Finalist.

COBOL Field Name/ C Field Name	Field Description
FINAL-REPORT-SELECT1 caSelect[0]	Turns on the Address Detail Report. Set this value on the initialization call to Finalist.
FINAL-REPORT-SELECT2 caSelect[1]	Turns on the Isolation Report if the Address Detail Report is requested. Set this value on the initialization call to Finalist.
FINAL-REPORT-SELECT3 caSelect[2]	Turns on the Suggestion Report if the Address Detail Report is requested. Set this value on the initialization call to Finalist.
FINAL-REPORT-SELECT4 caSelect[3]	Turns on the AdsInfo Report if the Address Detail Report is requested. Set this value on the initialization call to Finalist.
FINAL-REPORT-SELECT5 caSelect[4]	No longer used.
FINAL-REPORT-SELECT6 caSelect[5]	No longer used.
FINAL-FIRMLBL-OPT caFirmLabel	Determines which firm name Finalist returns to the label line. If this field contains the value "I", the system returns the input firm name. If this field contains the value "D", the system returns the firm name as matched to the ZIP + 4 data base. The default value is D.
FINAL-LOT-OPT caLOT	Determines whether Finalist performs LOT processing. If this field contains a "Y", the system performs LOT processing.
FINAL-REPORT-NTH caNth	Use this field to print every nth record on the Input/Output Reports. Set this value on the initialization call to Finalist. Valid values are 1 through 9999.
FINAL-REPORT-TITLE caTitle	Contains the title specified for printing on the Input/Output Reports. Set this value on the initialization call to Finalist.
FINAL-REPORT-KEY caKey	Specify a user-defined key so each record printed on the Input/Output Reports also prints the key.

COBOL Field Name/ C Field Name	Field Description
FINAL-REPORT1-MAX lMaxReport[0]	Contains the number of records you want to print on Input/Output Report One. The default value is 1000.
FINAL-REPORT2-MAX lMaxReport[1]	Contains the number of records you want to print on Input/Output Report Two. The default value is 1000.
FINAL-REPORT3-MAX lMaxReport[2]	Contains the number of records you want to print on Input/Output Report Three. The default value is 1000.
FINAL-REPORT4-MAX lMaxReport[3]	Contains the number of records you want to print on Input/Output Report Four. The default value is 1000.
FINAL-REPORT5-MAX lMaxReport[4]	Contains the number of records you want to print on Input/Output Report Five. The default value is 1000.
FINAL-REPORT6-MAX lMaxReport[5]	Contains the number of records you want to print on Input/Output Report Six. The default value is 1000.
FINAL-NUMBER-OF-LINES SNumLines	Indicates the number of lines to print on Finalist reports before a page break is forced. The default value is 57. Valid values range from 25 to 999.
FINAL-3553-SECTIONS Section3553	Upon the initial call to Finalist, use this field to set the values to appear in sections A1, B1, B4, B5, and D3 of the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECA1A caA1A	Defines the value to supply to line 1 of section A1 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECA1B caA1B	Defines the value to supply to line 2 of section A1 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECA1C caA1C	Defines the value to supply to line 3 of section A1 on the USPS Form 3553 (CASS Summary Report).

COBOL Field Name/ C Field Name	Field Description
FINAL-3553-SECA1D caA1D	Defines the value to supply to line 4 of section A1 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECB1 caB1	Defines the value to display in section B1 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECB4 caB4	Defines the value to display in section B4 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECB5 caB5	Defines the value to display in section B5 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3A caD3A	Defines the value to display in line 1 of section D3 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3B caD3B	Defines the value to display in line 2 of section D3 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3C caD3C	Defines the value to display in line 3 of section D3 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3D caD3D	Defines the value to display in line 4 of section D3 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3E caD3E	Defines the value to display in line 5 of section D3 on the USPS Form 3553 (CASS Summary Report).
FINAL-3553-SECD3F caD3F	Defines the value to display in line 6 of section D3 on the USPS Form 3553 (CASS Summary Report).

COBOL Field Name/ C Field Name	Field Description
FINAL-DPV-OPT caDPV	Determines whether Finalist performs Delivery Point Validation (DPV) Option processing: **N** No DPV processing. **Y** DPV processing using DPV Full File. **S** DPV processing using DPV Split File. **F** DPV processing using DPV Flat File. **Blank** Defaults to N.
FINAL-EWS-OPT caEWS	Determines whether Finalist performs Early Warning System (EWS) processing: **N** No EWS processing. **Y** Perform EWS processing. **Blank** Defaults to N.
FINAL-LLK-OPT caLLK	Determines whether Finalist performs LACS^{Link} Option processing: **N** No LACS^{Link} processing. **Y** LACS^{Link} processing with Large memory model. **U** LACS^{Link} processing with Ultra-small memory model. **S** LACS^{Link} processing with Small memory model. **M** LACS^{Link} processing with Medium memory model. **L** LACS^{Link} processing with Large memory model. **H** LACS^{Link} processing with Huge memory model. **Blank** Defaults to S.

COBOL Field Name/ C Field Name	Field Description
FINAL-DUAL-ADDR cDualAddrSwt	If the input file contains dual addresses (one a conventional address, a second containing a PO Box address), this field determines the 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 attempts to code the secondary address. Valid values are: **F** The first valid address (in the order Address line 1 and then Address line 2) is given the highest priority in the match process. **A** 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. **2** Finalist gives Line 2 in the dual address the highest priority in the match process. **1** Finalist gives Line 1 in the dual address the highest priority in the match process. **P** Finalist gives the PO Box address the highest priority in the match process. **C** Finalist gives the conventional address the highest priority in the match process. **Blank** Defaults to A.
FINAL-DPV-BUFFER-SIZE caDPVbuf	Indicates the memory model for DPV processing: **P** Pico. Stores no data in memory. No tables or indexes are loaded. **U** Ultra-small. Stores no data in memory. Partial indexes are loaded. **S** Small **M** Medium **L** Large **H** Huge. Stores all data in memory. **Blank** Defaults to M. Values from releases prior to the Finalist 8.0.0 release are accepted as: • If you specified 0, M is substituted. • If you specified 1, U is substituted • If you specified a value greater than 1 but less than or equal to 30, S is substituted. • If you specified a value greater than 30 but less than or equal to 500, M is substituted. • If you specified a value greater than 500, but less than or equal to 900, L is substituted. • If you specified a value greater than 900, H is substituted.

COBOL Field Name/ C Field Name	Field Description
FINAL-SLK-OPT caSLK	Determines whether Finalist performs Suite^{Link} processing. If this field contains a "Y", Finalist performs Suite^{Link} processing.
FINAL-ASM-OPT caASM	Determines whether Finalist performs All Street Matching (ASM) processing. If this field contains a "Y", Finalist performs ASM processing.
FINAL-DPV-SI caDPVSDI	Determines the action to take when encountering a DPV Seed during the processing run: **W** Issue warning message when a DPV Seed has been encountered. Processing continues but DPV processing is disabled. **S** Stop processing when encountering a DPV Seed. Finalist issues an error message, stops processing succeeding addresses, and exits the program. **Blank** Defaults to W.
FINAL-USE-SETUPDEF caUseSetupDef	Use SetupDef instead of the normal Init fields. If this field contains a "Y", Finalist uses SetupDef instead of the normal Init fields. See the COPY PBFNGCFG/PBFNSetupDef information in the Finalist Return Area - Function 0 Init Call. **Note:** If you are using this option, you must MOVE LOW-VALUES to PBFN-GCFG-SETUP with initialization.
FINAL-RDI-OPT caRDI	Determines whether Finalist performs RDI processing. If this field contains a "Y", Finalist performs RDI processing.

COBOL Field Name/ C Field Name	Field Description
FINAL-RET-SLK-INSEC caRetSLKInsec	Indicates how to return secondary information when Suite^{Link} secondary information is available. **B** Both. Return both Suite^{Link} and input secondary information. (This is the Default). **S** Suite^{Link}. Return Suite^{Link} secondary only. Do not return input secondary. **I** Input. Return input secondary only. Do not return Suite^{Link} secondary. **N** None. Do not return Suite^{Link} secondary or input secondary. **#** Return both the Suite^{Link} information and the unmatched secondary as "#". **Blank** Defaults to B. **Note:** If Suite^{Link} 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^{Link} secondary. Input secondary information includes the “#” designator for purposes of this option.

Finalist Call Area - Function 9 Term Call

Table 159: Finalist Call Area - Function 9 Term Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-RETURN-CODE	Defines a general return code that describes the success or failure of the Termination call. A return code of less than FINALIST-RC-PBFN-SUCCESS indicates that Finalist failed to terminate. Finalist should be terminated and the log files should be queried for complete details. A return code equal to FINALIST-RC-PBFN-SUCCESS means Finalist was able to successfully terminate.

COBOL Field Name/ C Field Name	Field Description
FINAL-CALL-AREA FinalistCallArea	Defines the entire call and return areas necessary for processing Finalist.
FINAL-FILLER lAnchorPointers	Do not alter the anchor area after the initialization call.
FINAL-FUNCTION-CODE cFunctionCode[0]	Determines the function to be performed when calling Finalist.

Finalist Call Area - Function 4, 5, 6, or 7 Process Call

Table 160: Finalist Call Area - Function 4, 5, 6, or 7 Process Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-RETURN-CODE	Defines a general return code that describes the success or failure of the Process call. A return code of FINALIST-RC-PBFN-ERROR indicates that Finalist encountered a severe error during processing. Finalist should be terminated and the log files should be queried. A return code equal to FINALIST-RC-PBFN-FAIL means Finalist was not able to code this address. This does not indicate a failure in Finalist. It indicates a failure with the input address. A return code equal to FINALIST-RC-PBFN-SUCCESS means Finalist was able to successfully process the address.
FINAL-CALL-AREA FinalistCallArea	Defines the entire call and return areas necessary for processing Finalist.
FINAL-FILLER lAnchorPointers	Do not alter the anchor area after the initialization call.

COBOL Field Name/ C Field Name	Field Description
FINAL-FUNCTION-CODE cFunctionCode[0]	Determines the function to be performed when Finalist is called.
FINAL-REPORT-KEY caKey	Defines a user-defined key so each record printed on the Input/Output Reports also prints the key.
FINAL-INPUT-ADDR-AREA InputAddress	This field is a group definition for the input address lines, city/state, ZIP Code, ZIP + 4 Code, and carrier route code.
USER-INPUT-FIRM-LINE caInputFirm	Defines the user input firm address line.
USER-INPUT-URB-LINE caInputURB	Defines the urbanization name when attempting to code a Puerto Rican address.
USER-INPUT-ADDRESS-1 caInputAddr1	Defines input address line one when attempting to code an address.
USER-INPUT-ADDRESS-2 caInputAddr2	Defines input address line two when attempting to code an address.
USER-INPUT-CSZ-AREA	This field is a group definition for the input city/state, ZIP Code, ZIP + 4 Code, and carrier route code.
USER-INPUT-CITY-STATE caInputCitySt	Defines the city/state from the input file.
USER-INPUT-ZIP caInputZip	Defines input ZIP Code when attempting to code an address.
USER-INPUT-SEC-SEG caInputSecSeg	Defines the input sector segment (ZIP + 4 Code).

COBOL Field Name/ C Field Name	Field Description
USER-INPUT-CR caInputCrRte	Defines the input carrier route here.
Z4-RESERVED caZ4Reserved	Reserved for the Z4Select product.
pPCBFNEWS PCBFNEWS	Applies only to Finalist/IMS users. You must supply the PCB address of the EWS file in this field when calling the Finalist interface.
pPCBFNLOT PCBFNLOT	Applies only to Finalist/IMS users. You must supply the PCB address of the eLOT file in this field when calling the Finalist interface.
pPCBFNDPV PCBFNDPV	Applies only to Finalist/IMS users. You must supply the PCB address of the DPV file (DPVDB, DPVHDB or DPVSDB) in this field when calling the Finalist interface.
pPCBFNDPVSUD PCBFNDPVSUD	Applies only to Finalist/IMS users. This field should remain 0 (LOW-VALUES) for now.
pPCBFNLLK PCBFNLLK	Applies only to Finalist/IMS users. You must supply the PCB address of the LACS^{Link} (LLKDB) file in this field when calling the Finalist interface.
pPCBFNLLKSUD PCBFNLLKSUD	Applies only to Finalist/IMS users. This field should remain 0 (LOW-VALUES) for now.
pPCBFNSLK PCBFNSLK	Applies only to Finalist/IMS users. You must supply the PCB address of the Suite^{Link} (SLKDB) file in this field when calling the Finalist interface.
pPCBFNRDI PCBFNRDI	Applies only to Finalist/IMS users. You must supply the PCB address of the RDIDB file in this field when calling the Finalist interface.

Finalist Return Area - Function 0 Init Call

Table 161: Finalist Return Area - Function 0 Init Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-INIT-RETURN InitReturn	Contains the information returned following an "Init" call. Move spaces to this field before calling Finalist.
FINAL-ONLINE-RETURN-CODE-0 cOnlineRC	If a fatal error within Finalist On-Line occurs, this area contains an E. This field may also contain values documented in the section **"Using the FINALOL Subroutine" on page 285**.
FINAL-OLD-RETURN-CODE2-0 cOldRC2	In a Windows environment, this field indicates an address pop-up window will display.
FINAL-OLD-RETURN-CODE3-0 cOldRC3	In a Windows environment, this field indicates a city/state/ZIP Code pop-up window will display.
CAERRMOD-0 caErrorModule	Applies to CICS users only. If the General Return Code is "E" (fatal error), this field contains the specific module where the error occurred.
CAERRSRC-0 caResource	Applies to CICS users only. CAERRSRC contains the sub-module name where the error occurred.
CAERRDSC-0 caDescription	Applies to CICS users only. CAERRDSC contains the CICS error message generated by a HANDLE condition.
FINAL-TIME-STAMP caTime	Defines the time stamp (HH:MM:SS) for Finalist processing.
FINAL-DATE-STAMP caDate	Defines the date stamp (MM/DD/YYYY) for Finalist processing.

COBOL Field Name/ C Field Name	Field Description
FINAL-DATA-FILE-VER caDataFileVer	Defines the version number (X.XX) of the Data (ZIP + 4) File.
FINAL-DATA-FILE-UPDATE caDataFileUpdate	Defines the creation date (MM/DD/YYYY) of the Data (ZIP + 4) File.
FINAL-CITY-FILE-VER caCityFileVer	Defines the version number (X.XX) of the City File.
FINAL-CITY-FILE-UPDATE caCityFileUpdate	Defines the creation date (MM/DD/YYYY) of the City File.
FINAL-PRODUCT-VERSION caProductVersion	Defines the version number (XXX) of the Finalist product.
FINAL-DPV-DATA-FILE-DATE caDPVDataFileDate	Contains the copyright date (MM/DD/YYYY) of the Delivery Point Validation (DPV) Data File.
FINAL-CMRA-FILE-DATE caCMRAFileDate	Contains the copyright date (MM/DD/YYYY) of the Delivery Point Validation (DPV) Commercial Mail Receiving Agent (CMRA) File.
FINAL-SEED-FILE-DATE caSeedFileDate	Contains the copyright date (MM/DD/YYYY) of the Delivery Point Validation (DPV) Seed (False Positive) File.
FINAL-LCD-FILE-DATE caLCDFileDate	Contains the copyright date (MM/DD/YYYY) of the Delivery Point Validation (DPV) Lowest Common Denominator (LCD) File.
FINAL-SLK-RELEASE-DATE caSLKReleaseDate	Contains the copyright date (MM/DD/YYYY) of the Suite^{Link} File.
COPY PBFNGCFG PBFNSetupDef	For information on the PBFNSetupDef C structure that defines the system configuration, refer to **PBFNSetupDef**.

Finalist Return Area - Function 9 Term Call

Table 162: Finalist Return Area - Function 9 Term Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-TERM-RETURN TermReturn	Contains the information returned following an “Term” call. Move spaces to this field before calling Finalist.
FINAL-ONLINE-RETURN-CODE-9 cOnlineRC	If a fatal error within Finalist On-Line occurs, this area contains an E. This field may also contain values documented in the section **Using the FINALOL Subroutine** on page 359.
FINAL-OLD-RETURN-CODE2-9 cOldRC2	In a Windows environment, this field indicates an address pop-up window will display.
FINAL-OLD-RETURN-CODE3-9 cOldRC3	In a Windows environment, this field indicates a city/state/ZIP Code pop-up window will display.
CAERRMOD-9 caErrorModule	Applies to CICS users only. If the General Return Code is "E" (fatal error), this field defines the specific module where the error occurred.
CAERRSRC-9 caResource	Applies to CICS users only. CAERRSRC defines the sub-module name where the error occurred.
CAERRDSC-9 caDescription	Applies to CICS users only. CAERRDSC defines the CICS error message generated by a HANDLE condition.
FINAL-CERT-NAME caCASSCertName	Defines the certified product name.
FINAL-CERT-VERSION caCASSCertVersion	Defines the certified product version.

COBOL Field Name/ C Field Name	Field Description
FINAL-CERT-CONFIG caCASSCertConfig	Defines the certified product configuration.
FINAL-Z4-NAME caZ4ChangeCertName	Defines the certified Z4Change product name.
FINAL-Z4-VERSION caZ4ChangeCertVersion	Defines the certified Z4Change product version.
FINAL-Z4-CONFIG caZ4ChangeCertConfig	Defines the certified Z4Change product configuration.
FINAL-LOT-NAME caLOTCertName	Defines the certified Line of Travel (LOT) product name.
FINAL-LOT-VERSION caLOTCertVersion	Defines the certified Line of Travel (LOT) product version.
FINAL-LOT-CONFIG caLOTCertConfig	Defines the certified Line of Travel (LOT) product configuration.
FINAL-VALID-FROM caFromValidDate	Defines the date (MM/DD/YYYY) on which the processing occurred.
FINAL-ZIP4-DATE caZIP4Date	Defines the date (MM/DD/YYYY) of the ZIP + 4 File.
FINAL-Z4-DATE caZ4ChangeDate	Defines the date (MM/DD/YYYY) of the Z4Change File.
FINAL-LOT-DATE caLOTDate	Defines the date (MM/DD/YYYY) of the Line of Travel (LOT) File.

COBOL Field Name/ C Field Name	Field Description
FINAL-CRIS-DATE caCRISDate	Defines the date (MM/DD/YYYY) of the CRIS File.
FINAL-RECORD-COUNT lRecordCount	Defines the count of records processed in this run.
FINAL-ZIP-COUNT lZIPCodedCount	Defines the count of ZIP Coded records coded in this run.
FINAL-CARRIER-COUNT lCRTCodedCount	Defines the count of carrier route records coded in this run.
FINAL-ZIP4-COUNT lZIP4CodedCount	Defines the count of ZIP + 4 records coded in this run.
FINAL-Z4-COUNT lZ4ChangeCodedCount	Defines the count of Z4Change records coded in this run.
FINAL-LOT-COUNT lLOTCodedCount	Defines the count of Line of Travel (LOT) records coded in this run.
FINAL-ZIP-TO-DATE caZIPto	Defines the last date (MM/DD/YYYY) that 5-digit ZIP coding is valid.
FINAL-CARRIER-TO-DATE caCRTto	Defines the last date (MM/DD/YYYY) that carrier route coding is valid.
FINAL-ZIP4-TO-DATE caZIP4to	Defines the last date (MM/DD/YYYY) that ZIP + 4 coding is valid.
FINAL-LOT-TO-DATE caLOTto	Defines the last date (MM/DD/YYYY) that LOT coding is valid.

COBOL Field Name/ C Field Name	Field Description
FINAL-HREXCT-COUNT lHRExactCount	Defines the count of exact matches to high-rise records coded to in this run. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-HRDFLT-COUNT lHRDefltCount	Defines the count of default matches to high-rise records coded to in this run. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-RREXCT-COUNT lRRExactCount	Defines the count of exact matches to rural route/highway contract records coded to in this run. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-RRDFLT-COUNT lRRDefltCount	Defines the count of default matches to rural route/highway contract records coded to in this run. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-LACS-COUNT lLACSCount	Defines the count of records that have been through LACS processing. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-DPV-PROCESS-COUNT 1DPVCount	Defines the count of records that have been processed through the Delivery Point Validation (DPV) Option.
FINAL-DPV-CONFIRM-COUNT lDPVConfirmCount	Defines the count of records delivery point validated in this processing run. This count is used for the QSS section of the USPS Form 3553 (CASS Summary Report).
FINAL-DPV-CMRA-COUNT lDPVCMRACount	Defines the count of records that are confirmed Commercial Mail Receiving Agents (CMRA).
FINAL-EWS-MATCHED-COUNT lEWSMatchCount	Defines the count of records that are failed because the records matched the Early Warning System (EWS) File.
FINAL-DPV-DATE caDPVDate	Defines the date (MM/DD/YYY) of the Delivery Point Validation (DPV) Option files.
FINAL-LACS-SEED-HEADER caLACSSeedHeader	Defines the LACS^{Link} Seed Header.

COBOL Field Name/ C Field Name	Field Description
FINAL-SLK-PROCESS-COUNT lSLKCount	Count of records passed to SuiteLink.
FINAL-SLK-CORRECT-SUITES lSLKCorrectedSuites	Number of corrected suites.
FINAL-SLK-MATCHED-COUNT lSLKSuccessfulMatches	Count of successful SuiteLink matches.
COPY PBNFGSTS PBFNStatsDef StatsDef	Refer to **PBFNStatsDef** on page 235 for information on the PBFNStatsDef C structure that provides information on processing statistics.
COPY PBNFSDPV PBFNDPVStatsDef DPVStatsDef	Refer to**PBFNDPVStatsDef** on page 180 for information on the PBFNDPVStatsDef C structure that provides Delivery Point Validation (DPV) processing statistics.
COPY PBNFPRST PBFNRtnLACSStatsDef LACSStatsDef	Refer to **PBFNRtnLACSStatsDef** on page 200 for information on the PBFNRtnLACSStatsDef structure that you can pass on the PBFNStats or PBFNTerminate call to return LACSLink processing statistics.
COPY PBNFTDPV PBFNDPVHdrDef DPVSeedHeader	Refer to **PBFNDPVHdrDef** on page 179 for information on the PBFNDPVHdrDef C structure that returns the Delivery Point Validation (DPV) header record.
COPY PBNFYDRS PBFNLACSSeedHdrDef LACSSeedHeader	Refer to **PBFNLACSSeedHdrDef** on page 197 for information on how to use the PBFNLACSSeedHdrDef structure in a PBFNStats call to return LACSLink False Positive violation header information.
COPY PBNFNCAS PBFN3553Def R3553Data	Refer to**PBFN3553Def** on page 131 for information on the PBFN3553Def C structure that contains the CASS Stage 2 header information.

COBOL Field Name/ C Field Name	Field Description
COPY PBNFRDAT PBFNReportData ReportData	Refer to **PBFNRDataDef** on page 198 for information on the PBFNRDataDef C structure that defines the available options for the Finalist Batch Report.
COPY PBNFIERR PBFNExtendedErrorDef ExtError	Refer to **PBFNExtendedErrorDef** on page 187 for information on the PBFNExtendedErrorDef C structure that provides error information.
COPY PBNFSSLK SuiteLinkStatsDef SLKStats	Refer to**PBFNRtnSuiteLinkStatsDef** on page 203 for information on SuiteLink processing statistics.

Finalist Return Area - Function 4, 5, 6, or 7 Process Call

Table 163: Finalist Return Area - Function 4, 5, 6, or 7 Process Call - Field Descriptions

COBOL Field Name/ C Field Name	Field Description
FINAL-PROCESS-RETURN ProcessReturn	Defines the information returned following a Process call. Move spaces to this field before calling Finalist.
FINAL-ORIGINAL-RETURN-AREA ReturnArea	Defines the start of the first of two return areas in Finalist. Move spaces to this field before each call to Finalist.
FINAL-ONLINE-RETURN-CODE cOnlineRC	If a fatal error within Finalist On-Line occurs, this area contains an E. This field may also contain values documented in the section **Using the FINALOL Subroutine** on page 359.
FINAL-OLD-RETURN-CODE2 cOldRC2	In a windows environment, this field indicates an address pop-up window will display.

COBOL Field Name/ C Field Name	Field Description
FINAL-OLD-RETURN-CODE3 cOldRC3	In a windows environment, this field indicates a city/state/ZIP Code pop-up window will display.
CAERRMOD caErrorModule	Applies to CICS users only. If the General Return Code is "E" (fatal error), this field defines the specific module where the error occurred.
CAERRSRC caResource	Applies to CICS users only. CAERRSRC defines the sub-module name where the error occurred.
CAERRDSC caDescription	Applies to CICS users only. CAERRDSC defines the CICS error message generated by a HANDLE condition.
FINAL-ISOL cIsol	Defines the number of isolation attempts (the number of times the subroutine isolated your address before Finalist could process your address). Valid values are in the range of 1-3.
FINAL-RECORD-NUMBER caRecNumber	Defines the record number assigned by Finalist during processing.
FINAL-ZIP Zip	Defines the output ZIP Code identified during Finalist processing.
FINAL-SCF caScf	Defines the first three bytes (SCF) of the output ZIP Code assigned by Finalist.
FINAL-ZONE caZone	Defines the last two positions of the output ZIP Code assigned by Finalist.
FINAL-SEC-SEG caSecSeg	Defines the sector segment (ZIP + 4 Code) returned after Finalist processing.
FINAL-CR-RTE caCrRte	Defines the carrier route returned after Finalist processing.

COBOL Field Name/ C Field Name	Field Description
FINAL-STATE caState	Defines the state abbreviation returned after Finalist processing.
FINAL-CITY caCity	Defines the USPS-standardized city name returned after Finalist processing.
FINAL-COUNTY caCounty	Defines the Federal Information Processing (FIPS) state/county code returned after Finalist processing.
FINAL-DIRECTION-1 Direction[0]	Defines the pre-directional, suffixes, and post-directional used for a coded address. For an unsuccessful address, this field defines the first set of pre-directional, suffixes, and post-directional combination on the Finalist Data File.
FINAL-DIR11 caDir1[0]	Defines the pre-directional for the first combination Finalist returned.
FINAL-SFX11 caSfx1[0]	Defines suffix-1 for the first combination Finalist returned.
FINAL-SFX12 caSfx2[0]	Defines suffix-2 for the first combination Finalist returned.
FINAL-PDIR1 caDir2[0]	Defines the post-directional for the first combination Finalist returned.
FINAL-DIRECTION-2 Direction[1]	Defines an additional combination of valid pre-directional, suffixes, and post-directional not used for a coded address. For an unsuccessful address, this defines the second set of pre-directional, suffixes, and post-directional combination on the Finalist Data File.
FINAL-DIR21 caDir1[1]	Defines the pre-directional for the second combination Finalist returned.
FINAL-SFX21 caSfx1[1]	Defines suffix-1 for the second combination Finalist returned.

COBOL Field Name/ C Field Name	Field Description
FINAL-SFX22 caSfx2[1]	Defines suffix-2 for the second combination Finalist returned.
FINAL-PDIR2 caDir2[1]	Defines the post-directional for the second combination Finalist returned.
FINAL-DIRECTION-3 Direction[2]	Defines an additional valid combination of pre-directional, suffixes, and post-directional not used for a coded address. For an unsuccessful address, this contains the third set of pre-directional, suffixes, and post-directional combination on the Finalist Data File.
FINAL-DIR31 caDir1[2]	Defines the pre-directional for the third combination Finalist returned.
FINAL-SFX31 caSfx1[2]	Defines suffix-1 for the third combination Finalist returned.
FINAL-SFX32 caSfx2[2]	Defines suffix-2 for the third combination Finalist returned.
FINAL-PDIR3 caDir2[2]	Defines the post-directional for the third combination Finalist returned.
FINAL-VALIDCICS-OPTION caValidCICSOption	This field describes the valid Finalist CICS or IMS option that is available for use. Returned values are "O" for On-Line, "W" for Windows, or "B" for batch only (no On-Line or Windows). Not returned in a batch environment.
FINAL-HOUSE-NUM caHouseNum	This field contains the house number returned by Finalist. This field also contains the PO box number, rural route number, etc. On a successfully coded record, this is the house number used to assign the postal codes, but on an unsuccessful record, this field defines the first isolation value. This field is right-justified.
FINAL-PRE-DIR caPreDir	Defines the pre-directional returned by Finalist after processing. On a successfully coded record, this is the pre-directional used to assign the postal codes. On an unsuccessful record, this field defines the first isolation value.

COBOL Field Name/ C Field Name	Field Description
FINAL-STREET-NAME caStreetName	Defines the street name or firm name returned by Finalist after processing. On a record successfully coded to a street, this is the street name used to assign the postal codes. On a record successfully coded to a firm (FINAL-REASON-CODE9 is set to the value "0"), this is the matched firm name. On an unsuccessful record, this field defines the first isolation value.
FINAL-POST-DIR caPostDir	Defines the post-directional returned by Finalist after processing. On a successfully coded record, Finalist uses this post-directional to assign the postal codes. On an unsuccessful record, this field defines the first isolation value.
FINAL-SFX1 caSfx1	Defines the first suffix returned by Finalist after processing. On a successfully coded record, this is the first suffix used to assign the postal codes. On an unsuccessful record, this field defines the first isolation value.
FINAL-SFX2 caSfx2	Defines the second suffix returned by Finalist after processing. On a successfully coded record, this is the second suffix used to assign the postal codes. On an unsuccessful record, this field defines the first isolation value.
FINAL-APT-NUM caAptNum	Defines the apartment number returned by Finalist after processing. On a successfully coded record, this is the apartment number used to assign the postal codes. On an unsuccessful record, this field defines the first isolation value. This field is right-justified.
FINAL-EXTRA caExtra	Defines the extraneous information that Finalist found during processing. The information in this field does not impact the coding of addresses.
ORIG-ZIP OrigZip	Defines the original ZIP Code as defined to Finalist.
ORIG-SCF caScf	Defines the first three digits (SCF) of your original input ZIP Code.
ORIG-ZONE caZone	Defines the remaining two digits of your original input ZIP Code.
ORIG-SEC-SEG caOrigSecSeg	Defines the original sector segment (ZIP + 4 Code) from the input file as defined to Finalist.
ORIG-CR-RTE caOrigCrRte	Defines the original carrier route from the input file as defined to Finalist.

COBOL Field Name/ C Field Name	Field Description
ORIG-STATE caOrigState	Defines the original state abbreviation when it is present on the input file.
ORIG-CITY caOrigCity	Defines the original city when it is present on the input file.
ORIG-ZIPP caOrigZipP	Defines a packed decimal translation of the original ZIP Code from the input file.
ORIG-SECSEGP caSecSegP	Defines a packed decimal translation of the original sector segment (ZIP + 4 Code) from the input file.
FINAL-STATE-SCF-VER cStateScfVer	No longer supported. Always returns x'03' (Input state and ZIP Code agree).
FINAL-RETSCF caRetScf	Defines the SCF (first three characters of the ZIP Code) returned by Finalist.
FINAL-PMUNIT caPMUnit	Private Mail Box (PMB) or Mail Stop Code (MSC) designator.
FINAL-PMNUMBER caPMNumber	Private Mail Box (PMB) or Mail Stop Code (MSC) unit number. The unit number will always be left-justified.

COBOL Field Name/ C Field Name	Field Description
FINAL-DPV-FOOTNOTES caDPVFootnotes	Defines the standard footnote codes returned during Delivery Point Validation (DPV) Option processing. **A1** Input address did not match to the ZIP + 4 File. **AA** Input address matched to the ZIP + 4 File. **BB** Input address matched to DPV (all components). **CC** Input address primary number matched to DPV but secondary number did not match (present but invalid). **F1** Input address matched to a military ZIP Code. **G1** Input address matched to a General Delivery address. **M1** Input address primary number missing. **M3** Input address primary number is invalid. **N1** Input address primary number matched to DPV but address is missing secondary number. **P1** Input address missing PO, RR, or HC Box number. **P3** Input address PO, RR, or HC box number invalid. **PB** Input address is a P. O. Box Street Address (PBSA). **R1** Input address matched to CMRA but secondary number is not present. **R7** Input address is a Carrier Route R77x. Footnote code R7 is only assigned to records that are DPV confirmed. **RR** Input address matched to CMRA. **U1** Input address matched to a unique ZIP Code.
FINAL-UNITDES caUnitDes	Defines the USPS Publication 28 unit designator if requested.
FINAL-CITY-IND caCityInd	Contains "S" if Finalist matched the input city to the USPS-standardized city name. An "L" indicates Finalist matched the input city to the full city name. This field is blank if Finalist could not match the input city.
FINAL-EXPANDED-RETURN-AREA ExpReturnArea	This field is the second of two return areas in Finalist. Move spaces to this field before each call to Finalist.
FINAL-OUTPUT-SELECT uSelect	Defines the high-level description for the three flags you can set for each address record to be written to one of the three output files.

COBOL Field Name/ C Field Name	Field Description
FINAL-OUTSEL-GOOD cGood	Defines a “Y” when the address successfully codes without reservation (good). The field is blank if the address either does not code completely (bad) or if Finalist changes the city name and/or ZIP Code (change). Finalist sets this value for each address processed.
FINAL-OUTSEL-BAD cBad	Defines a “Y” when the address does not code completely (bad). The ZIP + 4 Code was not assigned to the input address. The field is blank if the address either coded without reservation (good) or if Finalist changed the city name and/or ZIP Code (change). Finalist sets this value for each address processed.
FINAL-OUTSEL-CHANGE cChange	Defines a “Y” when the address codes but the city name and/or the ZIP Code changed (change). The field is blank if the address either coded without reservation (good) or did not code completely (bad). Finalist sets this value for each address processed.
FINAL-RETURN-CODE1 cRc1	Defines the General Return Code, returned after each address call to Finalist: **0** ZIP Code, ZIP + 4, and carrier route assigned. Same as PBFNProcess return code = PBFN_SUCCESS. **1** ZIP Code (five digit only) and carrier route assigned. Same as PBFNAddressDataDef.cReturnLevel = PBFN_RTNZIPCRRT. **2** ZIP Code (five digit only) assigned. Same as PBFNAddressDataDef.cReturnLevel = PBFN_RTNZIP. **9** No codes assigned. Same as PBFNProcess return code = PBFN_FAIL.
FINAL-REASON-CODES cReasonCode	Defines the 12 Finalist reason codes returned after each address call from Finalist.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE1 cReasonCode[0]	Reason Code 1 - Return ZIP Code Explanation: **0** ZIP Code verified. Same as PBFNAddressDataDef.cAdsZip = PBFN_EXACT. **1** ZIP Code returned. Same as PBFNAddressDataDef.cAdsZip = PBFN_RTN_DATA. **2** ZIP Code undetermined. Could not code into a Unique ZIP code. Same as PBFNAddressDataDef.cError = 4104. **4** ZIP Code determined valid in multiple choice situations. Same as PBFNAddressDataDef.cAdsZip = PBFN_EXACT if PBFNAddressDataDef.cFailureType = PBFN_COMPONENT_MULTI_CHOICE. **5** ZIP Code determined unique-changed. Same as PBFNAddressDataDef.cAdsZip = PBFN_CORRECTED if PBFNAddressDataDef.cZipType = PBFN_UNIQUE. **6** Last two digits of ZIP code changed. Same as comparing the original ZIP code with the returned ZIP code. **7** First three digits of ZIP code changed. Same as comparing the original ZIP code with the returned ZIP code. **9** ZIP Code not determined. Same as PBFNAddressDataDef.cAdsZip = PBFN_INVALID.
FINAL-REASON-CODE2 cReasonCode[1]	Reason Code 2 - Return City/State Explanation: **0** City verified as input. Same as PBFNAddressDataDef.cAdsCity = PBFN_EXACT. **1** City returned-none input. Same as PBFNAddressDataDef.cAdsCity = PBFN_RTN_DATA. **2** City standardized on input. Same as PBFNAddressDataDef.cAdsCity = PBFN_CORRECTED. **5** City/State name changed/corrected. Same as PBFNAddressDataDef.cAdsCity = PBFN_ CORRINPUT or PBFNAddressDataDef.cAdsState = PBFN_ CORRINPUT. **7** Nonmailing name. Same as PBFNAddressDataDef.cCityType = PBFN_NONMAILING_NAME. **9** City not determined. Same as PBFNAddressDataDef.cAdsCity = PBFN_INVALID.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE3 cReasonCode[2]	Reason Code 3 - Return Carrier Route Explanation: **0** Carrier Route verified. Same as PBFNAddressDataDef.cAdsCarrier = PBFN_EXACT. **1** Carrier Route returned. Same as PBFNAddressDataDef.cAdsCarrier = PBFN_RTN_DATA. **2** Carrier Route determined valid in multiple choice situation. Same as PBFNAddressDataDef.cAdsCarrier = PBFN_EXACT if PBFNAddressDataDef.cFailureType = PBFN_COMPONENT_MULTI_CHOICE. **3** Carrier Route changed. Same as PBFNAddressDataDef.cAdsCarrier = PBFN_CORRINPUT. **6** Default Carrier Route returned. Same as pAdsInfo->cAdsCarrier = PBFN_RTN_DATA AND (pAdsInfo->sAddrMatchLevel = PBFN_ADSLVL_RRHC_DEFAULT or PBFN_ADSLVL_UNIQUE_DEFAULT or PBFN_ADSLVL_GENDEL). **7** Non-deliverable address. Same as PBFNAddressDataDef.cNonDeliverableInd=Y. **9** Carrier Route not determined. Same as PBFNAddressIDataDef.cAdsCarrier = PBFN_INVALID.
FINAL-REASON-CODE4 cReasonCode[3]	Reason Code 4 - Return ZIP + 4 Explanation: **0** ZIP + 4 verified. Same as PBFNAddressDataDef.cAdsZip4 = PBFN_EXACT. **1** ZIP + 4 returned. Same as PBFNAddressDataDef.cAdsZip4 = PBFN_RTN_DATA. **2** ZIP + 4 determined valid in multiple choice situation. Same as PBFNAddressDataDef.cAdsZip4 = PBFN_EXACT if PBFNAddressDataDef.cFailureType = PBFN_COMPONENT_MULTI_CHOICE. **3** ZIP + 4 changed. Same as PBFNAddressDataDef.cAdsZip4 = PBFN_CORRINPUT. **6** Default ZIP + 4 returned. Same as cDefaultMatch=Y if Carrier Route, ZIP + 4, or DPBC is set to a default value. cDefaultMatch is found in the PBFNAddressDataDef structure. **7** Non-deliverable address. Same as PBFNAddressDataDef.cCityType = PBFN_NONMAILING_NAME. **8** AMSII has duplicate or overlapping range. Same as PBFNAddressDataDef.cAdsRange = PBFN_RANGE_OVERLAP. **9** ZIP + 4 not determined. Same as PBFNAddressDataDef.cAdsZip4 = PBFN_INVALID.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE5 cReasonCode[4]	Reason Code 5 - Street Explanation: **0** Street verified as input. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_EXACT. **1** Input street standardized. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_CORRECTED. **2** Street found using exceptions table. Same as PBFNAddressDataDef.cExceptionsInd=Y. **3** Street found using phonetics match. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_CORRINPUT. **5** Street failed because of EWS match. Same as PBFNProcessData.cError = 4460. **7** Street found using dual address rules. Same as PBFNAddressDataDef.cDualAddressInd=Y. **9** Street not matched. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_INVALID.
FINAL-REASON-CODE6 cReasonCode[5]	Reason Code 6 - Street Range Explanation: **0** Street range correct. Same as PBFNAddressDataDef.cAdsRange = PBFN_EXACT. **1** No match on alpha portion of range. Same as PBFNAddressDataDef.cAdsRange = PBFN_RANGE_ALPHA_MISMATCH. **2** No input range. Same as PBFNAddressDataDef.cAdsRange = PBFN_NODATA. **3** Out of range. Same as PBFNAddressDataDef.cAdsRange = PBFN_INVALID. **9** Range not determined. Same as PBFNAddressDataDef.cAdsRange = PBFN_INVALID.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE7 cReasonCode[6]	Reason Code 7 - Return Suffix/Directional Explanation **0** Suffix and directional correct. Same as PBFNAddressDataDef.cAdsSuffix and cPreAdsDir and cPostAdsDir all set to PBFN_EXACT. **1** Suffix missing or incorrect. Same as PBFNAddressDataDef.cAdsSuffix = PBFN_NODATA or PBFN_INVALID. **2** Directional missing or incorrect. Same as PBFNAddressDataDef.cAdsPreDir or cAdsPostDir fields set to PBFN_NODATA or PBFN_INVALID. **3** Directional and suffix incorrect. Same as PBFNAddressDataDef.cAdsSuffix and cPreDir and cPostDir fields all set to PBFN_NODATA or PBFN_INVALID. **4** Suffix multiple choice. Same as PBFNAddressDataDef.cAdsSuffix = PBFN_COMPONENT_MULTI_CHOICE. **5** Directional multiple choice. Same as PBFNAddressDataDef.cAdsPreDir or cAdsPostDir fields set to PBFN_COMPONENT_MULTI_CHOICE. **6** Directional outside cardinal point. Same as PBFNAddressDataDef.cAdsPreDir or cAdsPostDir fields set to PBFN_CARDINAL_FAILURE. **9** Suffix/directional could not be determined. Same as PBFNAddressDataDef.cAdsSuffix or cAdsPreDir or cAdsPostDir fields set to PBFN_INVALID.
FINAL-REASON-CODE8 cReasonCode[7]	Reason Code 8 - Alias Street Description: **0** Input street is not an alias. Same as PBFNAddressDataDef.cAdsStreetType = 0 **1** Input street is a preferred alias. Same as PBFNAddressDataDef.cAdsStreetType = PBFN_PREF_ALIAS. **2** Input street is a nickname alias. Same as PBFNAddressDataDef.cAdsStreetType = PBFN_OTHER_ALIAS. **4** Input street is an alias multiple choice. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_COMPONENT_MULTI_CHOICE. **5** Input street is an alternate at delivery. Same as PBFNAddressDataDef.cAdsStreetType = PBFN_ALT_AT_DEL. **9** Input street could not be matched. Same as PBFNAddressDataDef.cAdsStreetName = PBFN_INVALID.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE9 cReasonCode[8]	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). Same as PBFNAddressDataDef.cAdsFirm = PBFN_NODATA or PBFN_EXACT. **1** The system changed and returned the firm name per your specification. Same pAdsInfo->cAdsFirm = PBFN_CORRECTED and cFirmLabel != 'I'. **2** The system changed the firm name for matching purposes but was not returned, per your specification. Same pAdsInfo->cAdsFirm = PBFN_CORRECTED and cFirmLabel = 'I'. **3** Firm record matched but the firm name was missing. Same as PBFNProcessData.cError = 5103. **9** Firm processing was not successful. There was information on the firm line but it could not be matched to the address. Same as PBFNProcessData.cError = 5104.
FINAL-REASON-CODE10 cReasonCode[9]	Reason Code 10 - Unit Descriptor Description: **0** Unit designator (if any) is correct. Same as PBFNAddressDataDef.cAdsUnit1Designator or cAdsUnit2Designator = PBFN_EXACT or PBFN_NODATA. **1** Unit designator changed or added. Same as PBFNAddressDataDef.cAdsUnit1Designator or cAdsUnit2Designator = PBFN_RTN_DATA or PBFN_CORRINPUT. **2** Unit designator abbreviated. Same as PBFNAddressDataDef.cAdsUnit1Designator or cAdsUnit2Designator = PBFN_CORRECTED. **3** Unit designator missing or no unit number. Same as PBFNProcessData.cError = 5101. **4** Unit designator invalid. Same as PBFNAddressDataDef.cAdsUnit1Designator or cAdsUnit2Designator = PBFN_INVALID. **9** Secondary address error. Same as PBFNProcessData.cError = 5102 or 5105.

COBOL Field Name/ C Field Name	Field Description
FINAL-REASON-CODE11 cReasonCode[10]	Reason Code 11 - Unit Number Description: **0** Unit number (if any) is correct. Same as PBFNAddressDataDef.cAdsUnit1Range or cAdsUnit2Range = PBFN_EXACT, PBFN_NODATA, or PBFN_CORRECTED. **1** Unit number alpha character transposed. Same as PBFNAddressIDataDef.cAdsRange or cAdsUnit1Range or cAdsUnit2Range = PBFN_RANGE_ALPHA_MISMATCH. **2** Unit number valid in multiple choice. Same as PBFNAddressDataDef, any address component may receive this disposition if the effect of the address coding was ambiguous. **3** Unit number missing. Same as PBFNProcessData.cError = 5101. **4** Unit number invalid. Same as 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. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_RR or PBFN_ADSTYPE_HC = PBFN_RANGE_ALPHA_MISMATCH. **8** PO box number alpha character transposed. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_PO_BOX = PBFN_RANGE_ALPHA_MISMATCH. **9** Unit number not determined. Same as PBFNAddressDataDef.cAdsUnit1Range or cAdsUnit2Range = PBFN_INVALID.
FINAL-REASON-CODE12 cReasonCode[11]	Reason Code 12 - Non-Conventional Address Description **0** Non-conventional address correct (if present). Same as PBFNAddressDataDef.cAddressUnchanged=Y. Also, check PBFNAddressDataDef.cAddressType. **1** RR/HC corrected. Same as PBFNAddressDataDef.cAddressUnchanged=N and PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_RR or PBFN_ADSTYPE_HC. **2** PO box corrected. Same as PBFNAddressDataDef.cAddressUnchanged=N and PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_PO_BOX. **3** General delivery corrected. Same as PBFNAddressDataDef.cAddressUnchanged=N and PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_GD. **9** Non-conventional address not determined. Same as PBFNAddressDataDef.cAddressUnchanged=N and PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_UNKNOWN.

COBOL Field Name/ C Field Name	Field Description
FINAL-ADDRESS-INFO-CODES AddrInfo	The 10 address information codes returned after each address coding attempt made by Finalist.
FINAL-INFO-CODE1 cInformationCode[0]	Address Information Code 1 - City Type Information: **0** Single ZIP Code city. Same as PBFNAddressDataDef.cCityType = PBFN_SINGLEZONE. **1** Multi-ZIP Code city. Same as PBFNAddressDataDef.cCityType = PBFN_MULTIZONE. **9** City type not determined. Same as PBFNAddressDataDef.cCityType = 0.
FINAL-INFO-CODE2 cInformationCode[1]	Address Information Code 2 - Address Type Information: **0** Conventional address. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_STREET or PBFN_ADSTYPE_HIGHRISE. **1** PO box. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_PO_BOX. **2** Rural route. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_RR. **3** Highway contract route. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_HC. **4** General delivery. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_GD. **5** Firm address. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_FIRM. **9** Address type not determined. Same as PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_UNKNOWN.

COBOL Field Name/ C Field Name	Field Description
FINAL-INFO-CODE3 cInformationCode[2]	Address Information Code 3 - Delivery Type Information: **0** City delivery. Same as PBFNAddressDataDef.sAddrMatchLevel = PBFN_ADSTYPE_STREET. **1** Address failed but the ZIP Code is unique and delivery continued. Same as PBFNAddressDataDef.sAddrMatchLevel = PBFN_ADSLVL_STREET and PBFNAddressDataDef.cZipType = PBFN_UNIQUE. **2** PO box. Same as PBFNAddressDataDef.sAddrMatchLevel = PBFN_ADSLVL_PO. **3** Rural route/highway contract route. Same as PBFNAddressDataDef.sAddrMatchLevel = PBFN_ADSLVL_RRHC_DEFAULT or PBFN_ADSLVL_RRHC_SECONDARY. **4** General Delivery. Same as PBFNAddressDataDef.sAddrMatchLevel = PBFN_ADSLVL_GENDEL. **9** Delivery type not determined. Same as PBFNAddressDataDef.sAddrMatchLevel = 0.
FINAL-INFO-CODE4 cInformationCode[3]	Address Information Code 4 - Input Address Information: **0** Firm found on firm line-delivery found on line 1 and line 2. Same as PBFNAddressDataDef.cFirmLoc=PBFN_FIRMLOC_FIRM and PBFNAddressDataDef.cAddressLoc=PBFN_ADSLOC_L1_L2. **1** Address found on address line 1. Same as PBFNAddressDataDef.cAddressLoc = PBFN_ADSLOC_L1. **2** Address found on address line 2. Same as PBFNAddressDataDef.cAddressLoc = PBFN_ADSLOC_L2. **3** Address found on both address line 1 and line 2. Same as PBFNAddressDataDef.cAddressLoc = PBFN_ADSLOC_L1_L2. **4** Firm found on firm line, no delivery address found. Same as PBFNAddressDataDef.cFirmLoc=PBFN_FIRMLOC_FIRM and PBFNAddressDataDef.cAddressLoc=PBFN_ADSLOC_UNKNOWN. **5** Firm found on address line 1, no delivery address found. Same as PBFNAddressDataDef.cFirmLoc=PBFN_FIRMLOC_L1 and PBFNAddressDataDef.cAddressLoc=PBFN_ADSLOC_UNKNOWN. **6** Address found on firm line. Same as PBFNAddressDataDef.cDualAddressInd indicates the address was found on the firm line. **7** Firm found on firm line-delivery address found on address line 1. Same as PBFNAddressDataDef.cAddressLoc = PBFN_ADSLOC_FIRM_L1. **8** Firm found on firm line-delivery address found on address line 2. Same as PBFNAddressDataDef.cAddressLoc = PBFN_ADSLOC_FIRM_L2. **9** All address lines used-address failed. Same as Return code from PBFNProcess = PBFN_FAILED.

COBOL Field Name/ C Field Name	Field Description
FINAL-INFO-CODE5 cInformationCode[4]	Address Information Code 5 - Address Lookup Information. No longer supported.
FINAL-INFO-CODE678 caCode678	Address Information Codes 6 - 8, returned after each address call to Finalist: **060** Sector/segment is not deliverable. Same as PBFNAddressDataDef.cNonDeliverableInd = Y.
FINAL-INFO-CODE9 cInformationCode[8]	Address Information Code 9 - Output Address Type: **0** Conventional address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_STREET and return code from PBFNProcess is PBFN_SUCCESS. **1** PO Box address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_PO and return code from PBFNProcess is PBFN_SUCCESS. **2** Rural route address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_RRHC_DEFAULT or PBFN_ADSLVL_RRHC_SECONDARY and PBFNAddressDataDef.cAddressType=PBFN_ADSTYPE_RR. Return code from PBFNProcess is PBFN_SUCCESS. **3** Highway contract address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_ RRHC_DEFAULT or PBFN_ADSLVL_RRHC_SECONDARY and PBFNAddressDataDef.cAddressType = PBFN_ADSTYPE_RR. Return code from PBFNProcess is PBFN_SUCCESS. **4** General deliver address coded. Same as sAddrMatchLevel = and PBFN_ADSLVL_GENDEL return code from PBFNProcess is PBFN_SUCCESS. **5** Firm address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_FIRM_PRIMARY or PBFN_ADSLVL_FIRM_SECONDARY and return code from PBFNProcess is PBFN_SUCCESS. **6** Highrise address coded. Same as sAddrMatchLevel = PBFN_ADSLVL_HIGHRISE_DEFAULT or PBFN_ADSLVL_HIGHRISE_SECONDARY and return code from PBFNProcess is PBFN_SUCCESS. **7** Firm at highrise address coded. Same as sAddrMatchLevel is a combination of (PBFN_ADSLVL_FIRM_PRIMARY or PBFN_ADSLVL_FIRM_SECONDARY and PBFN_ADSLVL_HIGHRISE_DEFAULT or PBFN_ADSLVL_HIGHRISE_SECONDARY) and return code from PBFNProcess is PBFN_SUCCESS. **8** Military address coded. Same as SAddrMatchLevel = PBFN_ADSLVL_MIL_DEFAULT or PBFN_ADSLVL_MIL_SECONDARY or PBFNAddressDataDef.cZipType = PBFN_MILITARY. **9** Cannot code output address. Same as return code from PBFNProcess = PBFN_FAILED.

COBOL Field Name/ C Field Name	Field Description
FINAL-INFO-CODE10 cInformationCode[9]	Address Information Code 10 - ZIPMove Type: **0** Match determined using ZIPMove record. Same as PBFNAddressDataDef.cZipMove = Y. **9** ZIPMove forwarding did not occur. Same as PBFNAddressDataDef.cZipMove = N.
FINAL-5D-BARCODE	Defines the five-digit barcode. This field returns with each nonfailure coded address returned by Finalist.
FINAL-5D-BEG cBeg	Defines an "!" to indicate the beginning framing character for the barcode. You may specify a different character by placing your character's hexadecimal representation in this field.
FINAL-5D-ZIP caZip	Defines the ZIP Code assigned by Finalist so the input address prints in barcode format.
FINAL-5D-CKDIGIT cChkdigit	Defines the valid one-digit modulo check digit required for printing the correct barcode.
FINAL-5D-END cEnd	Defines an "!" to indicate the ending framing character for the barcode. You may specify a different character by placing your character's hexadecimal representation in this field.
FINAL-LINE-OF-TRAVEL	Defines the LOT information.
FINAL-LOT-CODE caLotCode	Defines the LOT code.
FINAL-LOT-ASCDSC cLotAD	Defines the LOT ascending/descending code.
FINAL-LACS-FLAG cLACS	Defines the LACS processing flag to indicate whether LACS processing is available.

COBOL Field Name/ C Field Name	Field Description
FINAL-LACS-LINK-RETURN cLacsLink	Defines the LACS^{Link} return code: **A** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. **00** LACS^{Link} processing failed. No matching record found during LACS^{Link} processing. **09** LACS^{Link} processing matched the input address to an older highrise default address. The address has been converted. However, rather than provide an imprecise address, LACS^{Link} processing does not provide a new address. **14** LACS^{Link} processing failed. Match found during LACS^{Link} processing but conversion did not occur due to other USPS regulations. **92** LACS^{Link} processing successful. Record matched through LACS^{Link} processing. Unit number dropped on input.
FILLER	This area reserved for future use.
FINAL-ADVANCED-BARCODE AdvBarCode	Defines the necessary values to print the 14-digit delivery point barcode on your mail piece. This field returns with each successfully coded address returned by Finalist.
FINAL-ABC-BEG cBeg	Defines a “!” to indicate the beginning framing character for the delivery point barcode.
FINAL-ABC-ZIP caZip	Defines the ZIP Code assigned by Finalist so the input address prints in barcode format.
FINAL-ABC-ZIP4 caZip4	Defines the sector segment (ZIP + 4 Code) assigned by Finalist so the input address prints in barcode format.
FINAL-ABC-WSEQ caWseq	Defines the Advanced Barcode (ABC) digits that allow you to print the delivery point barcode.
FINAL-ABC-CKDIGIT caChkdigit	Defines the valid one-digit modulo check digit required for printing the correct delivery point barcode.

COBOL Field Name/ C Field Name	Field Description
FINAL-ABC-END cEnd	Defines an “!” to indicate the ending framing character for the delivery point barcode.
FINAL-ALT-ISOLATION AltIsol	This field is reserved for Finalist internal use only.
FINAL-ALT-RANGE caRange	This field is reserved for Finalist internal use only.
FINAL-ALT-PRE-DIR caPreDir	This field is reserved for Finalist internal use only.
FINAL-ALT STREET caStreet	This field is reserved for Finalist internal use only.
FINAL-ALT-SUFFIX caSuffix	This field is reserved for Finalist internal use only.
FINAL-ALT-POST-DIR caPostDir	This field is reserved for Finalist internal use only.
FINAL-ALT-UNIT caUnit	This field is reserved for Finalist internal use only.
FINAL-ALT-RANGE2 caRange2	This field is reserved for Finalist internal use only.
FINAL-FULL-CITY-NAME caFullCityName	Defines the full city name returned by Finalist.
FINAL-LABEL-RETURN-AREA Label	Defines the label format area where Finalist places the output address in acceptable USPS format. This process also eliminates all of the spaces between the individual fields listed earlier when creating each label line.

COBOL Field Name/ C Field Name	Field Description
FINAL-LABEL-RC cRc1	Defines the label area return code. **1** Label lines built from Finalist area. **2** Label lines built from original input.
FINAL-FIRM-LENGTH1 caLen[0]	Defines the length of the returned firm line based on the significant characters in the line.
FINAL-FIRM-LINE1 caLine[0]	Upon successful coding, this field represents possible firm or URB information. This could be blank, a firm, or a Puerto Rican URB depending on the presence of the address lines. Upon unsuccessful address coding, the same information will be presented but based on the original input fields, not the coded address information.
FINAL-LABEL-LENGTH1 caLen[1]	Defines the length of label line 1 based on the significant characters in the line.
FINAL-LABEL-LINE1 caLine[1]	Upon successful coding, this field defines the standardized address line 1. This could be a firm or a Puerto Rican URB. Upon an unsuccessful address coding, this field contains the original input address line 1.
FINAL-LABEL-LENGTH2 caLen[2]	Defines the length of label line 2 based on the significant characters in the line.
FINAL-LABEL-LINE2 caLine[2]	Upon successful coding, this field defines the standardized address line 2. This could be a conventional address, PO box, etc. Upon an unsuccessful address coding, this field contains the original input address line 2.
FINAL-LABEL-LENGTH3 caLen[3]	Defines the length of label line 3 based on significant characters in the line.
FINAL-LABEL-LINE3 caLine[3]	Upon successful coding through Finalist, this field defines the assigned city/state, ZIP Code, and ZIP + 4 Code. Upon an unsuccessful address coding, this field contains the original city/state and ZIP Code from input.
FINAL-ALT-LABEL-LENGTH2 caLen[4]	Defines the length of the alternate label line.

COBOL Field Name/ C Field Name	Field Description
FINAL-ALT-LABEL-LINE2 caLine[4]	Defines the alternate label line 2 used during alias match logic.
FINAL-MAIL-FIRM-NAME caMailFirmName	This field is reserved for future use.
FINAL-CITY-VANITY-NAME caCityVanityName	Defines the area where the standardized non-mailing city name returns if your input city was a non-mailing city name.
FINAL-URB-RETURN caURBReturn	If an address codes and contains an URB (Puerto Rico only), the output URB will be returned in this field. If a Puerto Rico address fails to code, the input URB will be returned in this field. In all other cases, this field will contain blanks.
FINAL-DEFAULT-IND cDefault	Contains "Y" if Finalist matched to a default USPS record. Otherwise, this field is blank.
FINAL-POUND-FIELD1 ca#Field1	Contains # sign information that must be returned on the output label line in accordance with USPS PMB regulations.
FINAL-POUND-FIELD2 ca#Field2	Contains # sign information that must be returned on the output label line in accordance with USPS PMB regulations.
FINAL-DPV-IND caDPVIND	Defines the Delivery Point Validation (DPV) processing flag that indicates whether the address is a confirmed delivery point. **N** Not a valid delivery point. The USPS cannot deliver mail to this address. **Y** Delivery point validated. Primary range and secondary range (when present) are valid. The USPS can deliver mail to this address. **S** Valid primary range. Secondary range is present but is not confirmed. The USPS can deliver mail to this address. **D** Valid primary range. Secondary range is missing. The USPS can deliver mail to this address.
FINAL-DPV-CMRA-IND caDPVCMRA	Contains a "Y" if the address is a valid Commercial Mail Receiving Agent (CMRA). This field contains an "N" if the address is a confirmed delivery point but is not a valid CMRA. This field is blank if the address is not a confirmed delivery point (i.e., FINAL-DPV-IND contains "N").

COBOL Field Name/ C Field Name	Field Description
FINAL-DPV-FALSE-POSITIVE caDPVFalsePositive	Contains a "Y" if the address is not a confirmed delivery point and a positive response is received from the False Positive File. This field contains an "N" if the address is not a confirmed delivery point and a negative response is received from the False Positive File. This field is blank if the address is a confirmed delivery point.
FINAL-MATCH-LEVEL-IND caMatchLevel	Indicates whether an address matched to a firm, street, high-rise, PO box, rural route, or general delivery record. **F** Firm record match **G** General delivery match **H** High-rise default match **P** PO box match **R** RR/HC match **S** Street record match
FINAL-DPV-RETURN-CODE caDPVReturnCode	This field is not used.
FINAL-LACS-SEED-HIT cLacsSeedHit	Indicates whether a LACS seed hit was encountered.
FINAL-DPV-SHA-VALUE caDPVSHA	Defines the DPV SHA value.
FINAL-DPV-EMDP-VALUE caDPVEMDP	Defines the DPV EMDP value.
FINAL-SEASONAL-IND caSeasonInd	Contains the seasonal indicators for the returned ZIP Code. These indicators identify, at the 5-digit ZIP Code level, the months in which seasonal addresses receive delivery. There are 12 monthly flags (January through December). A "Y" in one of the monthly slots indicates that seasonal addresses are delivered mail in the month indicated by that slot. This field is blank if there are no seasonal deliveries for the ZIP Code.
FINAL-SLK-RETURN-CODE caSLKReturnCode	Defines the Suite^{Link} return code.

COBOL Field Name/ C Field Name	Field Description
FINAL-SLK-MATCH-CODE caSLKMatchCode	Defines the Suite^{Link} match code.
FINAL-SLK-FIDELITY-CODE caSLKFidelityCode	Defines the Suite^{Link} fidelity code.
FINAL-DPV-SEED-HIT caSeedViolationFlag	Indicates whether a DPV seed violation has occurred.
FINAL-DPV-NCOAKEY-FLAG caDPVKeyNCOAFlag	Indicates whether the DPV key is an NCOA key.
FINAL-RDI-RETURN-CODE caRDIReturnCode	Contains a "Y" if the matched address is a residential delivery. Contains an "N" if the matched address is a business delivery. This field is blank if the address fails address lookup, or the RDI™ Option is not active.
PBNFRRTN PBFNAddressDataDef	Refer to **PBFNAddressDataDef** on page 138 for information on the PBFNAddressDataDef C structure that defines the return address information.

Tips and Techniques for the Compatibility Interface (CI)

This section provides programming tips and techniques for using the Compatibility Interface (CI).

Generic Search Criteria

You must supply search criteria to use the GETZIP, GETSTRT, GETRNGE, and GETCITY functions (for example, a target ZIP Code or target city name). You can use fully-qualified search criteria (for example, a five-digit ZIP Code) or a generic qualifier. Specify generic qualifiers from left to right only. For example, 605* is a valid generic ZIP Code, but *0137 is not a valid generic ZIP Code.

After each criteria field, such as ZIP Code or city name, there is a five-character field for you to specify how Finalist should use the search criteria. For example, if you specify 'FIRST' in this field, Finalist returns the first item on the file that matches the supplied criteria. If you specify 'NEXT' in this field and you made a previous call, Finalist returns the next sequential item.

Example

These examples use generic qualifiers with the GETZIP function.

Table 164: Generic Search Examples

Requested ZIP Code	Indicator	Action
60137	FIRST	Finalist returns ZIP Code 60137.
60137	NEXT	If this is the first request for 60137, Finalist returns that ZIP Code. If you made a previous request for 60137, Finalist returns 60138. If you make another call, Finalist returns 60139.
605*	FIRST	Finalist returns the first valid ZIP Code for SCF 605.
605*	FIRST	If this is the first request with 605* specified as the target ZIP Code, the program returns the first valid ZIP Code for SCF 605. If you made a previous request with 605* specified as the target ZIP Code, this request will return the next valid ZIP Code for SCF 605.

Note: Prime entries in callinfo or retinfo refer to the primary spelling of a city or street name rather than a variation.

Sample Database Access Method Program

The database access method gives you access to information in the Finalist Data File and City/State File. This sample program retrieves ZIP Code information from the Finalist Data File. You can use other database access method functions to retrieve street/range information, such as sector segment, carrier route, and apartment information. The database access method also includes functions to retrieve city information.

Sample Access Method Program

The LPAMDOC sample database access method program below demonstrates how to use the Finalist database access method. This program writes sequential output to the LPAMOUT file. Note that this is an MVS program. The sample database access method program interrogates the LPAM

return code and then performs the appropriate action (print a ZIP Code or error message). Numbers that appear to the left of some program statements correspond to descriptions that follow the figure.

```
     LPAMDOC  CSECT                   *S
              STM    R14,R12,12(R13) * T       L
              BAL    R14,80(R15)     *  A       I
     SAVEAREA DS     18F             *   N       N
              ST     R13,4(R14)      *    D       K
              ST     R14,8(R13)      *     A       A
              LR     R13,R14         *      R       G
              USING  SAVEAREA,R13    *       D       E
     *
              PRINT  NOGEN
              OPEN   (LPAMOUT,(OUTPUT))
              LA     R1,=CL8'LPAM'
              LOAD   EPLOC=(R1),ERRET=ENDIT
     *        CDLOAD (R1),RETPNF=YES
     *        LTR    R15,R15
     *        BNZ    OUT
              ST     R0,ALPAM
              L      R15,ALPAM
  1           CALL   (15),(FINIT,ANCHOR@,RETCODE,ERRINFO,PROFILE),VL
              MVC    PLINE+20(4),=C'INIT'
              CLC    RETCODE(2),=C'08'
              BNL    DOERROR
  2           MVC    PRODUCT(8),=C'BACCMETH'
              MVC    RESOURCE(8),=C'CBDATA'
              L      R15,ALPAM
              CALL   (15),(FOPEN,ANCHOR@,RETCODE,ERRINFO,RCB@,
      X
                     PRODUCT,RESOURCE),VL
              CLC    RETCODE(2),=C'08'
              BNL    DOERROR
  3           MVC    ZCTZIP(5),=C'60187'
              MVC    ZCTZIPI(5),=C'FIRST'
              MVC    ZCREPL(3),=C'010'
              L      R15,ALPAM
              CALL   (15),(FGETZIP,ANCHOR@,RETCODE,ERRINFO,
      X
                     RCB@,ZIPCALL,ZIPINFO),VL
              CLC    RETCODE(2),=C'08'
              BNL    DOERROR
              MVC    PLINE,SPACES
              MVC    PLINE(17),=CL17'# ZIPS RETURNED ='
              MVC    PLINE+18(3),ZR#ENT
              PUT    LPAMOUT,PLINE
              XR     R3,R3
              PACK   DLBLWORD,ZR#ENT(3)
              CVB    R3,DLBLWORD
              LA     R4,ZIPINFO+3
     DOLOOP   MVC    PLINE,SPACES
              MVC    PLINE(ZRZTLEN-3),0(R4)
              PUT    LPAMOUT,PLINE
```

```
              LA     R4,(ZRZTLEN-3)(R4)
              BCT    R3,DOLOOP
              B      CLOSE
     *
     DOERROR  MVC    PLINE,SPACES
              MVC    PLINE(13),=CL13'RETURN CODE ='
              MVC    PLINE+14(2),RETCODE
              PUT    LPAMOUT,PLINE
              MVC    PLINE,SPACES
              MVC    PLINE(80),ERRINFO
              PUT    LPAMOUT,PLINE
              B      ENDIT
     *
     CLOSE    EQU    *
              L      R15,ALPAM
  4           CALL  (15),(FCLOSE,ANCHOR@,RETCODE,ERRINFO,RCB@),VL
              L      R15,ALPAM
  5           CALL  (15),(FTERM,ANCHOR@,RETCODE,ERRINFO),VL
              CLOSE (LPAMOUT)
     *
     ENDIT    L      R13,4(R13)
              LM     R14,R12,12(R13)
              XR     R15,R15
              BR     R14
     *
     PLINE    DC     CL80' '
     SPACES   DC     CL80' '
     DLBLWORD DS     D
     ALPAM    DS     F
     *
     *  ACCESS METHOD FUNCTIONS.
     *
     FINIT    DC CL8'INIT'
     FOPEN    DC CL8'OPEN'
     FGETZIP  DC CL8'GETZIP'
     FCLOSE   DC CL8'CLOSE'
     FTERM    DC CL8'TERM'
     *
     ZIPCALL  LPAM007M  ZC                 Call area for the GETZIP
function.
     *
     ZIPINFO  LPAM008M  ZR                 Return area for the GETZIP
     *                                     function.
              DC    30CL(ZRZTLEN-3)' '     Save Room For 30 More Zip
     *                                     Returns
     *
     ERRINFO  DS    CL80                   Error Diagnostics Returned
By Acc
     *                                     Meth.
     RETCODE  DS    CL2                    Return Code From The Access
Method
     PRODUCT  DS    CL8                    Access Method Product Name
     RESOURCE DS    CL8                   Access Method Resource (File)
```

```
 Name
     PROFILE  DS    CL8                    Access Method Profile Table
Name
     *
     RCB@     DS    D                      Filled in by the ACC Meth
'OPEN'
     *                                    function.  Do not modify this
 field.
     ANCHOR@  DS    F                      Filled In By The Acc Meth
'INIT'
     *                                    function.  Do not modify this
 field
              LTORG
     *
     LPAMOUT  DCB   DDNAME=LPAMOUT,DSORG=PS,MACRF=PM,RECFM=FB,
      X
                    LRECL=80,BLKSIZE=80
     *
     R0       EQU   0
     R1       EQU   1
     R2       EQU   2
     R3       EQU   3
     R4       EQU   4
     R5       EQU   5
     R6       EQU   6
     R7       EQU   7
     R8       EQU   8
     R9       EQU   9
     R10      EQU   10
     R11      EQU   11
     R12      EQU   12
     R13      EQU   13
     R14      EQU   14
     R15      EQU   15
     *
              END   LPAMDOC
```

The numbers that appear to the left of the program statements correspond to the following descriptions.

1. This is the access method INIT call. This must be the first function issued to the access method. An 'INIT' call causes the access method to do all one-time initialization getting control areas (GETMAIN), etc. The access method needs only one INIT call. The ANCHOR@ field is filled with an address when the 'INIT' call completes successfully. The caller must not modify this field. The access method uses the field on all subsequent calls.
2. This is the access method OPEN call. Finalist uses the OPEN function to establish a path to a resource (in this case, a file). The RCB@ field is where the access method will place the address of the internal control block that manages activity against the resource. The caller must not modify the RCB@ field after the OPEN call. All subsequent access method calls against the resource (i.e., GETCNTL, GETZIP, etc.) need the RCB@ field. The product and resource field values are no longer supported and will be ignored.

3. The GETZIP function retrieves information about a ZIP Code. The RCB@ field is from the access method OPEN call. The ZIPCALL parameter points to the call area for the GETZIP function. The caller fills in the call area with search criteria information such as the target ZIP Code, how many ZIP Codes (starting with the requested one) it needs, etc. The ZIPINFO field points to an area where GETZIP will place the returned ZIP Code information.
4. This is the access method CLOSE call. The access method CLOSE function closes a path to a resource. The access method 'OPEN' call must open this path before the CLOSE call. The RCB@ field is from the OPEN call.
5. This is the access method TERM call. The TERM function performs the inverse of the INIT function. The TERM function frees all access method control blocks and releases (deletes) all programs.

Sample COBOL Driver Program

A sample calling or driver program for Finalist in COBOL is shown next. Your driver may also call LPFNSMPL (for a one-line-per-record output listing reflecting the input records used in processing). To do so, substitute LPFNSMPL for LPFNIORP.

```
IDENTIFICATION DIVISION.
 PROGRAM-ID.  'FINALDR'.
 REMARKS.   THIS IS AN EXAMPLE OF AN MVS FINALIST COBOL
            DRIVER PROGRAM.   IT BUILDS THE FINALIST CALL
            AREA FROM AN INPUT RECORD, CALLS FINALIST TO
            RETURN A FULL RETURN AREA, KEEPS COUNTERS AND
            WRITES AN OUTPUT RECORD.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
      SELECT USER-FILE-IN ASSIGN TO UT-S-SYSUT1.
      SELECT OUT-FILE     ASSIGN TO UT-S-SYSUT2.
      SELECT PRINT-FILE   ASSIGN TO UT-S-SYSPRINT.
 DATA DIVISION.
 FILE SECTION.
 FD USER-FILE-IN
      RECORD CONTAINS 220 CHARACTERS
      LABEL RECORDS ARE STANDARD
      BLOCK CONTAINS 0 RECORDS
      DATA RECORD IS RECORD-IN.
 01   RECORD-IN.
      05  FILLER                        PIC X(8).
      05  ZIP-IN                        PIC X(5).
      05  FILLER                        PIC X(43).
      05  ADDR-IN                       PIC X(30).
      05  CITY-STATE-IN                 PIC X(40).
      05  FILLER                        PIC X(94).
 FD  OUT-FILE
      RECORD CONTAINS 92620 CHARACTERS
```

```
     BLOCK CONTAINS 0 RECORDS
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS OUT-RECORD.
01   OUT-RECORD.
     05  USER-RECORD-OUT              PIC X(220).
     05  OUT-RETURN-AREA              PIC X(92400).
WORKING-STORAGE SECTION.
01   COUNTERS.
     05 RECORDS-READ                  PIC S9(9)  COMP-3 VALUE 0.
     05 RECORDS-WRITTEN               PIC S9(9)  COMP-3 VALUE 0.
01   EOF-SW                          PIC X(3)   VALUE 'NO '.
01   MAX-RECS BINARY                 PIC S9(5)  VALUE +4.
*     COPY LPFNCL01 HERE
      COPY LPFNCL01
PROCEDURE DIVISION.
      OPEN   INPUT USER-FILE-IN
             OUTPUT OUT-FILE.
      INITIALIZE FINAL-EXPANDED-INFO.
      MOVE '0' TO FINAL-FUNCTION-CODE.
      MOVE 'CNFIGAAJ' TO FINAL-CNFIG-ID.
      MOVE 'Y' TO FINAL-DPV-OPT.
      MOVE 'Y' TO FINAL-LLK-OPT.
      MOVE 'Y' TO FINAL-SLK-OPT.
      MOVE SPACES TO OUT-RETURN-AREA.
      MOVE HIGH-VALUE TO FINAL-FILLER.
      CALL 'FINAL' USING FINAL-CALL-AREA
        RETURNING FINAL-RETURN-CODE
      IF FINALIST-RC-PBFN-ERROR
         DISPLAY 'FINAL INIT FAILED, RC = '
           FINAL-RETURN-CODE
         DISPLAY 'PBFN-IERR-MESSAGECODE = '
           PBFN-IERR-MESSAGECODE OF
           FINAL-INIT-RETURN
         MOVE '9' TO FINAL-FUNCTION-CODE
         CALL 'FINAL' USING FINAL-CALL-AREA
         MOVE 16 TO RETURN-CODE
         STOP RUN
       ELSE IF NOT FINALIST-RC-PBFN-SUCCESS
         DISPLAY 'FINAL INIT WARNING, RC = '
         FINAL-RETURN-CODE
       END-IF
      CALL 'LPFNIORP' USING FINAL-CALL-AREA,MAX-RECS.
      MOVE '5' TO FINAL-FUNCTION-CODE.
      READ USER-FILE-IN
           AT END MOVE 'YES' TO EOF-SW.
      PERFORM 100-MAINLINE THRU 100-EXIT
              UNTIL EOF-SW EQUAL 'YES'.
      MOVE '9' TO FINAL-FUNCTION-CODE.
      CALL 'FINAL' USING FINAL-CALL-AREA
        RETURNING FINAL-RETURN-CODE
      CALL 'LPFNIORP' USING FINAL-CALL-AREA.
      DISPLAY RECORDS-READ.
      DISPLAY RECORDS-WRITTEN.
```

```
      CLOSE  USER-FILE-IN
             OUT-FILE.
      MOVE 0 TO RETURN-CODE
      STOP RUN.
100-MAINLINE.
      MOVE SPACES TO FINAL-PROCESS-RETURN.
      MOVE ADDR-IN       TO USER-INPUT-ADDRESS-1.
      MOVE SPACES        TO USER-INPUT-ADDRESS-2.
      MOVE CITY-STATE-IN TO USER-INPUT-CITY-STATE.
      MOVE ZIP-IN        TO USER-INPUT-ZIP.
      CALL 'FINAL' USING FINAL-CALL-AREA
        RETURNING FINAL-RETURN-CODE
      IF FINALIST-RC-PBFN-ERROR
        DISPLAY 'FINAL PROCESS FAILED, RC = '
          FINAL-RETURN-CODE
        DISPLAY 'PBFN-IERR-MESSAGECODE = '
          PBFN-IERR-MESSAGECODE OF
          FINAL-PROCESS-RETURN
        MOVE '9' TO FINAL-FUNCTION-CODE
        CALL 'FINAL' USING FINAL-CALL-AREA
        MOVE 16 TO RETURN-CODE
        STOP RUN
      ELSE IF FINALIST-RC-PBFN-FAIL
        DISPLAY 'ADDRESS DID NOT CODE:'
        DISPLAY ' ' FIRM-IN
        DISPLAY ' ' URB-IN
        DISPLAY ' ' ADDR1-IN
        DISPLAY ' ' ADDR2-IN
        DISPLAY ' ' CITY-STATE-IN
        DISPLAY ' '
      END-IF
      CALL 'LPFNIORP' USING FINAL-CALL-AREA.
*  PLACE ADDITIONAL PROCESSING HERE
      MOVE RECORD-IN                 TO USER-RECORD-OUT.
      MOVE FINAL-PROCESS-RETURN      TO OUT-RETURN-AREA.
      WRITE OUT-RECORD.
      READ USER-FILE-IN
           AT END MOVE 'YES' TO EOF-SW.
100-EXIT.  EXIT
```

Using the FINAL Subroutine

Input

- Your input address
- Processing parameters passed during the initialization call to Finalist
- Function code and input address passed during the processing calls
- Finalist City/State File and Data File
- Exceptions table file (if used)

Note: The FINAL subroutine does not alter the input address data fields.

Output

The FINAL subroutine output consists of the return data fields in the Finalist passdata area.

Driver Program Requirements

You can write the driver program in any programming language that uses standard IBM linkage. The COBOL and Assembler languages are the most commonly used languages. CALLAREA (passdata) definitions for COBOL, Assembler, and C are provided. Data areas in the C header file that refer to character strings are processed as fixed length character arrays, not C strings (these areas are not x'00' delimited).

If you write your own driver program, you must enter three fullword fields in the field reserved for that purpose in your passdata area. Set these three fields to "-1" prior to the initialization call. Do not modify on subsequent calls. If the program calling Finalist is not a driver (it is called from another program) then these fullwords must be saved in the linkage section and passed back to Finalist on subsequent calls. Failure to do this may cause unpredictable results. For each call, follow the conventions of IBM standard linkage as described next.

- Use register 13 for the address of the save area
- Use register 14 for the return address
- Use register 15 for the entry address

Most Finalist modules have been assembled and linked in 31-bit mode, and take advantage of "above the line" storage. Batch drivers that call FINAL or ADDRSCAN no longer have the limitation of running in 24-bit mode. LPAM remains in 24-bit mode. The linkage attributes for FINAL and ADDRSCAN are address mode (AMODE) 31 and resident mode (RMODE) 24. FINAL and ADDRSCAN can accept a 24-bit or 31-bit address. If your user-written driver resides above the line (31-bit enabled), the driver must dynamically LOAD the FINAL and/or ADDRSCAN modules.

Statically linking FINAL or ADDRSCAN (not recommended) may only be done if the user-written driver resides below the line (24-bit enabled). LPAM routines remain AMODE (24) and RMODE (24). All calls to LPAM must pass 24-bit addresses.

An example of linking COBOL routines with Dynamic Parameter and 31-bit addressing follows:

```
//COB      EXEC PGM=IGYCRCTL,
//         PARM=(APOST,OPT,DYN,RES,NOSEQ,DATA(31),
//         LANG(UE),XREF,MAP,OFFSET,FDUMP)
```

Your program should handle the following:

1. Set the buffer count.

   If you choose not to let the buffer count default, you must initialize the buffer count on the initialization call by moving the number of buffers you want to run with into INITBUFF. You may want to pass the number of buffers to your driver program as a parameter in your JCL, which allows you to alter the number of buffers from one run to another without recompiling each time. You can assign buffers during Finalist initialization (function code "0" call only).

   If you are using a COBOL driver and want to hard code the value into INITBUFF, define the fields of your Finalist call area as shown in the first three lines below. Include the fourth line in your procedure division.

   ```
   05  CALL-INIT.
      10  INITMOD                  PIC X(8).
      10  INITBUFF                 PIC S9(5) BINARY.
   MOVE +99 TO INITBUFF.
   ```

2. Perform input/output processing.

   Your driver program must perform the I/O logic and the actual read/write operations on your name-and-address file. This includes moving the returned data to the output area prior to writing your own record.

3. Prepare the passdata/call area.

   Your driver program must extract one or two street address lines of up to 70 characters each from the input record. If your input file contains more than two address lines, your program can access ADDRSCAN before processing the FINAL subroutine. Your driver must also extract one of these items:

- City and state for a city/state lookup — The FINAL subroutine determines all returned data including ZIP Code, carrier route code, sector segment, and delivery point barcode, based on your input city and state.
- ZIP Code for a ZIP Code lookup — The FINAL subroutine determines all returned data including city, state, sector segment, and delivery point barcode based on your input ZIP Code.

Regardless of the type of lookup you choose, you may also want to extract any ZIP Codes, sector segment numbers, or carrier route codes on your input file and pass them to the call area. The Finalist reports and return codes can help you to determine the accuracy of this information.

4. Set the function code.

   Determine the pertinent function code your driver program passes to the FINAL subroutine and set FINAL-FUNCTION-CODE in the call area. For more information, refer to **Function Codes** on page 355.

5. Define and initialize MAXRECS field. The MAXRECS field should be passed as a parameter on your initial call to LPFNIORP or LPFNSMPL. The MAXRECS field sets the maximum records printed on the Input/Output ISOL/ST Reports or the Input/Output Sample Report. For COBOL, define the MAXRECS field as:

   ```
   01  MAX-RECS BINARY   PIC S9(5).
   ```

6. Pass control to the FINAL subroutine.

   The driver program must include the actual subroutine call statement at the appropriate time in its processing logic. For sample call statements in COBOL and Assembler, refer to **Using the FINAL Subroutine** on page 350.

7. Analyze the return code fields (optional).

   The FINAL subroutine returns 23 return codes (made up of one general return code, 12 reason codes, and 10 address information codes). These codes indicate that Finalist either successfully attached data or reveal the reason why certain data couldn't be determined. You can analyze these return codes to determine the input data's accuracy.

   Based on the return codes your addresses generate, you might want to include additional processing in your driver program. For example, you might want to write only those records for which Finalist successfully assigned a ZIP Code.

8. Analyze the OUTSEL field (optional). The source members LPFNCL04 (Assembler), LPFNCL01 (COBOL), and LPFNCL0C (C) contain the three-position OUTSEL field.
   a) The first position refers to addresses that Finalist successfully coded.
   b) The second position refers to addresses Finalist could not code.
   c) The third position refers to addresses Finalist coded but changed the city name or ZIP Code. The change suggests a geographical difference between the address as entered and coded.

To quickly determine how Finalist coded each record, interrogate each field. A "Y" in a field indicates that the record fits the category above. For more information about a particular record, refer to the remaining return codes.

9. Create printed output.

   When you enter function 9 in the subroutine call area, the FINAL subroutine generates all requested reports. This call also terminates Finalist processing, frees storage and deletes loaded modules.

Using the FINAL Subroutine

1. Before you call the FINAL subroutine for the first time, set the three fullwords FINAL-FILLER ($REREGS) to "-1".
2. Move "0" to FINAL-FUNCTION-CODE (FUNCODE) to initialize the FINAL subroutine. Function code 0 indicates that Finalist should perform its own initialization processing.
3. Move the configuration value into the call area to satisfy CASS requirements. A COBOL example of moving the configuration value into the Finalist call area on a function 0 call follows:

```
MOVE 'CNFIGxxx' TO FINAL-CNFIG-ID.
```

   Note: Values moved to this area will be used during processing. If you move individual tailoring option values to this call area, these options take precedence over any CNFIGxxx parameter values previously moved to the call area. To satisfy CASS requirements, move only the desired CASS-certified configuration to the call area.

4. Call the FINAL subroutine and pass your call area to it. A sample COBOL call follows:

```
MOVE '0' TO FINAL-FUNCTION-CODE.
CALL 'FINAL' USING FINAL-CALL-AREA
  RETURNING FINAL-RETURN-CODE
IF FINALIST-RC-PBFN-ERROR
  DISPLAY 'FINAL INIT FAILED, RC = '
    FINAL-RETURN-CODE
  DISPLAY 'PBFN-IERR-MESSAGECODE = '
    PBFN-IERR-MESSAGECODE OF
    FINAL-INIT-RETURN
  MOVE '9' TO FINAL-FUNCTION-CODE
  CALL 'FINAL' USING FINAL-CALL-AREA
  MOVE 16 TO RETURN-CODE
  STOP RUN
ELSE IF NOT FINALIST-RC-PBFN-SUCCESS
  DISPLAY 'FINAL INIT WARNING, RC = '
    FINAL-RETURN-CODE
END-IF
```

5. Initialize the optional report modules. A sample call for the Input/Output ISOL/ST Report follows:

```
CALL 'LPFNIORP' USING FINAL-CALL-AREA,MAX-RECS.
```

   A sample call for the Input/Output Sample Report in COBOL follows:

```
CALL 'LPFNSMPL' USING FINAL-CALL-AREA,MAX-RECS.
```

6. Place the appropriate address lookup function in position 1 of the function code.
7. Repetitively clear the FINAL-ORIGINAL-RETURN-AREA and FINAL-EXPANDED-RETURN-AREA. Move data from your input area to the call area until end-of-file. Separate the address, city, state, and ZIP Code components using a blank (you must have at least one space between contiguous components).
8. Call the FINAL subroutine and pass your call area to it. The FINAL subroutine analyzes and parses your data. The subroutine then returns data to the Finalist return area. A sample call in COBOL follows:

```
MOVE '5' TO FINAL-FUNCTION-CODE.
CALL 'FINAL' USING FINAL-CALL-AREA
  RETURNING FINAL-RETURN-CODE
IF FINALIST-RC-PBFN-ERROR
  DISPLAY 'FINAL PROCESS FAILED, RC = '
    FINAL-RETURN-CODE
  DISPLAY 'PBFN-IERR-MESSAGECODE = '
    PBFN-IERR-MESSAGECODE OF
    FINAL-PROCESS-RETURN
  MOVE '9' TO FINAL-FUNCTION-CODE
  CALL 'FINAL' USING FINAL-CALL-AREA
  MOVE 16 TO RETURN-CODE
  STOP RUN
ELSE IF FINALIST-RC-PBFN-FAIL
  DISPLAY 'ADDRESS DID NOT CODE:'
  DISPLAY ' ' FIRM-IN
  DISPLAY ' ' URB-IN
  DISPLAY ' ' ADDR1-IN
  DISPLAY ' ' ADDR2-IN
  DISPLAY ' ' CITY-STATE-IN
  DISPLAY ' '
END-IF
```

9. Call the optional report modules to generate the Input/Output ISOL/ST Report or the Input/Output Sample Report. A sample call for the Input/Output ISOL/ST Report in COBOL follows:

```
COBOL CALL  'LPFNIORP' USING FINAL-CALL-AREA.
```

   A sample call for the Input/Output Sample Report follows:

```
CALL  'LPFNSMPL' USING FINAL-CALL-AREA.
```

Note: Make this call conditionally, on the basis of which records you want to view on the report.

A sample call to print key Input/Output ISOL/ST Reports follows:

```
CALL 'LPFNIORP' USING FINAL-CALL-AREA,MAX-RECS,FINAL-REPORT-KEY.
```

A sample call to print key Input/Output Sample Reports follows:

```
CALL 'LPFNSMPL' USING FINAL-CALL-AREA,MAX-RECS,FINAL-REPORT-KEY.
```

10. Move both the original data and the Finalist return data to your output area.
11. At the End-Of-File, set the function code field to a value of 9 and call the FINAL subroutine. This causes Finalist to print all requested reports. A sample call in COBOL follows:

```
MOVE '9' TO FINAL-FUNCTION-CODE.
CALL 'FINAL' USING FINAL-CALL-AREA
  RETURNING FINAL-RETURN-CODE
```

A sample call for keyed reports follows:

```
CALL 'LPFNIORP' USING FINAL-CALL-AREA,MAX-RECS,FINAL-REPORT-KEY.
```

Function Codes

Before calling the FINAL subroutine, your driver program must set the value for the function code field.

Table 165: Function Code 0 (Zero)

Position	Sample Entry	Description
1	0	Initialize the FINAL subroutine. Function code 0 indicates to load the module named in INITMOD.
2, 3, 4	Blank	These positions are reserved for future use.

Tailoring Options

The tailoring options are required to modify default options. Set the tailoring switches in FINAL-TAILOR-OPTS. The call area format source members LPFNCL04 (Assembler), LPFNCL01 (COBOL), and LPFNCL0C (C) have the tailoring switches. These switches cannot override the configuration parameter setting (FINAL=CNFIGxxx) which the function 0 call must also set. If you do not use the FINAL=CNFIGxxx parameter setting, the USPS Form 3553 (CASS Summary Report) does not print and you are not eligible for automation-based discounts.

Finalist evaluates these switches on a function 0 call. When evaluating the tailoring codes, Finalist looks for one of the valid values listed above and accepts it. If Finalist does not find a valid value for a given option, a default value is substituted. You may allow any or all switches to default.

Table 166: Tailoring Switches

Switch Position	Switch Title	Switch Description	Acceptable Values	Default Value
1-10		No longer used.		
11	CTYLONG	Return long city name.	Y=Yes N=No	N
12	ALSLBL	Return alias street name in label area. For more information on returning alias street names and the values shown here, refer to FINAL-ALSLBL-OPT in **Finalist Call Area - Function 0 Init Call** on page 298.	Y=Yes N=No 1 2 3 4 5 6	N
13	FIRMLBL	Return data base firm name in label area if you specify "D". Return input firm name in label area if you specify "I".	D=DB I=INP	D

Additional Function Codes

Table 167: Function Codes 4 through 9

Position	Sample Entry	Description
1	4	Code the input name and address using a ZIP lookup (that is, the FINAL subroutine determines the ZIP Code, sector segment, and carrier route on the basis of your input ZIP Code). If it cannot code an address, Finalist determines the ZIP Code, sector segment, and carrier route based on your input city and state (city/state lookup).

Position	Sample Entry	Description
	5	Code the input name and address using a city/state lookup. Finalist uses a ZIP lookup if unable to code an address.
	6	Code the input name and address using a ZIP lookup only.
	7	Code the input name and address using a city/state lookup only.
	9	End of job designation. Close the files, print the Processing Statistics Report and Input/Output Listing Reports, and return control to the driver program.
2, 3, 4	blank	These positions are reserved for future use.

Printing the Finalist Batch Report With Your Driver

The Finalist Batch Report contains six sections of Finalist processing statistics for your input file. The steps to select sections for printing are shown next.

1. Place the string explained below before the initialization call to Finalist.
2. Move characters to an eight-byte field called "FINAL-STAT-REPORT-OPTIONS" in the passdata area. When moving characters, follow one of these procedures:
   - Move a value of 'N' to the first byte of FINAL-STAT-REPORT-OPTIONS.
   - Move a value of 'N' followed by up to a six-byte numeric string. Use this numeric string to indicate which of the six sections should not print.

     Example

     Move 'N123456' to the FINAL-STAT-REPORT-OPTIONS field if you do not want to print any of the Processing Statistics Report sections. If you do not want to print section two or section five -- but want to print sections one, three, four, and six -- the program should place the string 'N25' into this eight-byte field.

Printing the Finalist Input/Output Listing Reports

Specify the Input/Output Reports to print:

1. Set the values noted below before the initialization call to Finalist.
2. Passdata area positions 70-75 contain six flags. Each flag corresponds to a report. Set a flag to "Y" to print the corresponding report. Set a flag to "N" if you do not want to print the corresponding report.

The fields FINAL-REPORT-SELECT1 through FINAL-REPORT-SELECT6 control the Input/Output Reports for Finalist. The following is a sample of the passdata area.

```
05  FINAL-EXPANDED-INFO.
10  FINAL-STAT-REPORT-OPTIONS  PIC X(8) VALUE 'Y'.
10  FINAL-REPORT-CALL-AREA.
    15  FINAL-REPORT-SELECT1    PIC X VALUE 'Y'.
    15  FINAL-REPORT-SELECT2    PIC X VALUE 'Y'.
    15  FINAL-REPORT-SELECT3    PIC X VALUE 'Y'.
    15  FINAL-REPORT-SELECT4    PIC X VALUE 'Y'.
    15  FINAL-REPORT-SELECT5    PIC X VALUE 'Y'.
    15  FINAL-REPORT-SELECT6    PIC X VALUE 'Y'.
```

Example

If you don't want to print reports one, three, and five:

3. Set FINAL-REPORT-SELECT1, FINAL-REPORT-SELECT3, and FINAL-REPORT-SELECT5 to 'N'
4. Set FINAL-REPORT-SELECT2, FINAL-REPORT-SELECT4, and FINAL-REPORT-SELECT6 to 'Y'.

   Note: The JCL must contain the corresponding DDNAME for MVS.

Example

The following is a sample COBOL call for the Input/Output ISOL/ST Report in COBOL:

```
CALL 'LPFNIORP' USING FINAL-CALL-AREA,MAX-RECS.
```

Example

The following is a sample COBOL call for the Input/Output Sample Report:

```
CALL 'LPFNSMPL' USING FINAL-CALL-AREA,MAX-RECS.
```

Example

The following is an additional sample COBOL call for the Input/Output Sample Report. Set up MAX-RECS in working storage as:

5. Set up MAX-RECS in working storage as:

```
01  MAX-RECS        PIC S9(5)  BINARY
           VALUE + nnnnn.
```

The number of records you to include on report is indicated by "nnnnn". For more information, refer to the sample user driver in this chapter.

Checking the Return Code After FINAL Initialization Call

FINAL sets a return code to indicate the success or failure of the call to FINAL. In COBOL programs, add a RETURNING keyword and a field name such as FINAL-RETURN-CODE.

Table 168: FINAL Call Return Codes

Return Code	Description
-1	Error — Processing should not continue. Check the logs for detailed information.
0	Error — Processing should not continue. Check the logs for detailed information.
1	Success — Processing completed as requested

Following an initialization call ('0'), additional returns codes may be returned.

Table 169: Initialization Call (’0’) Return Codes

Return Code	Description
4	Warning — The database will expire for CASS purposes at the end of the month.
5	Warning — The database has expired for CASS purposes and your run did not complete in CASS mode.
7	Warning — Your key will stop CASS processing key at the end of the month.
8	Warning — eLOT processing was requested but the eLOT database was not available.
9	Warning — Your software key has expired for CASS processing and CASS processing has been turned off.

Using the FINALOL Subroutine

Your driver must issue a CICS LINK to invoke the on-line subroutine. Use the following format:

```
EXEC CICS LINK PROGRAM('FINALOL') COMMAREA(FINAL-CALL-AREA) LENGTH(10000)
```

The Finalist CICS subroutine operation differs from FINAL (the batch counterpart). In a batch driver, you can pass a buffer count on a function 0 call. This count determines the buffer pool size. The on-line subroutine does not support a buffer count. The on-line subroutine ignores any value passed to INITBUFF.

If a fatal error (i.e., GETMAIN failure, I/O errors, etc.) occurs within the Finalist On-Line system, the subroutine immediately exits. FINAL-ONLINE-RETURN-CODE in the Finalist passdata area contains an "E". For debugging purposes, the passdata area fields CAERRMOD, CAERRSRC, and CAERRDSC are also set with the specific component name and return code.

User Driver Modules

To fully use Finalist CICS, you may need to write site-specific applications. As you write these applications, note the following:

- Before invoking the on-line subroutine to perform Finalist On-Line processing, your application driver must prepare the passdata area for each link.
- You can write your driver in any language supported by the CICS environment.
- The layout and content of the on-line subroutine's passdata area are the same as the batch Finalist passdata area. Three links to FINALOL must be made for each address that needs to be coded by Finalist. These links include a function zero call that initializes the tailoring options and anchor fullwords, a function 5 or 4 call that passes the user input data, and a function 9 call to terminate Finalist properly and free all acquired storage before returning to the calling program.
- Use the anchor fullwords to hold the address of Finalist dynamic storage between calls to Finalist. High values in the anchor fullwords cause Finalist to acquire new storage. Be sure to set the anchor fullwords to high values (X'FF's).
- Check FINAL-ONLINE-RETURN-CODE (COBOL) for errors or warnings that occur during processing.

Resolving Problems with User Drivers

Use the following steps to help resolve user-driver problems.

1. Test Finalist CICS using the Finalist driver, LPCT. If the same problem occurs, contact a Technical Support Representative.
2. If the problem only exists when running your driver, ask a Technical Support Representative to review your user driver and verify that the sequence and call area for each function call are correct.
3. If the calls are correct, Technical Support will obtain your approval to forward the problem to the Professional Services Department for resolution.

Checking the Return Code After FINALOL Initialization Call

FINALOL sets a return code in the Finalist passdata area to indicate the success or failure of the initialization call. In COBOL programs, check field FINAL-ONLINE-RETURN-CODE, defined in the Finalist COBOL copybook LPFNCL01. A description for possible returned values and the actions your application should take are described in the figure shown next. Please note the meanings of these values are valid only on an initialization call (function 0 call).

Table 170: On-Line Return Codes After FINALOL Initialization Call

Value	Description
0	No errors. No errors occurred during initialization. Warning: Database validation has not occurred.
E	A fatal processing error has occurred. This message indicates that an unexpected error occurred during Finalist processing. Fields CAERRMOD, CAERRSRC, and CAERRDSC (COBOL) contain further information regarding the error. The driver program should interrogate these fields for further information regarding the error.

Checking the Return Code After FINALOL Process Call

A Process call to FINALOL (function 5 or function 4 call) also returns a value in the On-Line return code field (FINAL-ONLINE-RETURN-CODE). The returned value descriptions are different from the values returned after an initialization call.

Table 171: On-Line Return Codes After FINALOL Processing Call

Value	Description
0	Product and database are current.
2	Database expires at the end of the month.
3	Product expires at the end of the month.

Value	Description
4	Warning: Database has expired. The Finalist database has expired. The database has exceeded the USPS mandated 105 day usage period. Your application may continue non-certified processing; however, the return information may no longer be accurate. You must install the current monthly or bimonthly data before any CASS-certified processing can be performed.
6	Warning: Finalist batch product has expired. The Finalist batch product has exceeded the CASS certification date. The return information may no longer be accurate. You must install the new release of Finalist.
8	Unsuccessful. Refer to fields CAERRMOD, CAERRSRC, and CAERRDSC for specific error data. The Finalist database version is invalid for this version of the Finalist software. The interface program terminates processing. Address processing is not performed. You must install a new monthly or bimonthly database.
E	A fatal processing error has occurred. This message indicates that an unexpected error occurred during Finalist processing. Fields CAERRMOD, CAERRSRC, and CAERRDSC (COBOL and ASSEMBLER) contain further information regarding the error. The driver program should interrogate these fields for further information regarding the error.

Finalist CICS On-Line Error Checking

The Finalist On-Line system provides error information in three fields. This section describes the error fields and their contents. Follow these steps to check for errors:

1. After each link to the on-line subroutine, check the FINAL-ONLINE-RETURN-CODE (COBOL) or OLRETCD1 (Assembler) in the Finalist passdata area.
2. If this byte contains an "E", a serious error occurred within Finalist. This error prevented Finalist from completing the request. You can determine the error's cause by checking these three fields in the passdata area.

- CAERRMOD — This field contains the module name that detected the error.

Table 172: CAERRMOD Values

Possible Value	Description
FINALOL	On-Line subroutine
CICS	Service routine
LPFN510C	Finalist tailoring options subroutine
LPFNCICS	CICS service request module
LPFNEWS	Early Warning System (EWS) subroutine
LPFNDIOC	DPV I/O subroutine

- CAERRSRC — This field contains resource name involved in the error (for CICS-related errors only).

Table 173: CAERRSRC Values

Possible Values	Description
CBDATA	Any resource used by Finalist
CBCTYST	
LPFNSFX	
LPFNDIR	

- CAERRDSC — Contains exceptional condition causing the error. The figure shown next describes possible values in this field. The CAERRDSC field may contain any exceptional condition that can occur in CICS. Please consult your CICS Application Programmers Reference Manual for a complete list and description of these conditions.

Table 174: CAERRDSC Values

Possible Values for non-CICS related errors	Description
01	The callarea was not passed to the on-line subroutine or the specified length on the CICS link command was too small.

Possible Values for non-CICS related errors	Description
02	An invalid function code passed in the Finalist call area.
03	An ABEND occurred within the on-line subroutine.
06	Invalid Data File type. Verify that you loaded the Data File correctly.
07	Invalid Data File version level. Verify that you loaded the correct version of the Data File.
08	Your Finalist validation key does not support the execution of CICS.

When the error is CICS-related, the CAERRSRC and the CAERRDSC fields represent the EIBRSRCE and the EIBRCODE fields of the Exec Interface Block. When the error is not CICS-related, CAERRSRC is not used and CAERRDSC contains one of the Finalist error codes.

3.

Using the Exceptions Table

To create an exceptions table in Finalist On-Line or to replace a current exceptions table with a new one, follow the steps below.

1. Execute the following z/OS JCL to build the exceptions table. This JCL includes the job stream required to build the exceptions table and place it in the Finalist On-Line executable library.

```
1   //STEP1      EXEC PGM=PBFNEXCP
2   //STEPLIB    DD   DSN=your.finalist.batch.loadlib
3   //SYSPRINT   DD   SYSOUT=*
4   //SYSLOG     DD   SYSOUT=*
5   //SYSUDUMP   DD   SYSOUT=*
6   //EXCPIN     DD   *
    L1MANE              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,NOOBJECT,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,
    //              PARM='LIST,MAP,RENT,XREF,AMODE=31,RMODE=ANY'
16  //SYSLIB     DD DUMMY
17  //SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(1,1))
18  //SYSPRINT   DD SYSOUT=*
19  //SYSLMOD    DD DISP=SHR,DSN=your.finalist.online.loadlib
20  //SYSLIN     DD DSN=*.ASMCC.SYSPUNCH,DISP=SHR
    //             DD *
    NAME PBFNEXTB(R)
    //
```

A number precedes each statement containing a user-defined variable. This number refers to a specific explanation provided below.

Statement Number	Description
1	This statement executes PBFNEXCP. This program processes the exceptions table control cards and creates an output file. The next step inputs this output file to the Assembler program. The output file contains all statements needed to create an exceptions table for use with the Finalist On-Line system.
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.
4	Include this statement if your site requires system messages 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 identical to those used for exceptions table processing in the batch environment. For more information on exceptions table processing, refer to the section "Exceptions Table Option" in your Finalist User's Guide.
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.

Statement Number	Description
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 On-Line programs. The DSN parameter in statement 20 must appear as shown. If not, the Finalist On-Line system can not reference the exceptions table.

2. If replacing an existing table with a new one, issue a 'CEMT S PROGRAM(PBFNEXTB) NEW' command in CICS to refresh the copy of the exceptions table on-line.
3. Activate the optional exceptions table feature for Finalist On-Line. If you use the LPCT transaction, set the screen's selection switch to Y. If calling the "FINALOL" subroutine, set the "CAEXCPSW" field to "Y" in the Finalist passdata area.

Using the FINALI Subroutine

It is necessary to write site-specific applications to fully use the IMS version of the Finalist On-Line Windows system at your location. You may write the driver program in any language that uses standard IBM linkage conventions. To implement Finalist processing in IMS online applications, use the FINALI interface program. To implement Finalist processing in IMS batch, use the FINALB interface program.

> **Note:** Some fields in the Finalist call area are treated differently when used in the IMS environment. These differences are detailed later in this chapter. In addition, the IMS Finalist online interface program, FINALI, sets a return code to indicate the success or failure of the interface initialization process. For more information, refer to **Checking the Return Code After Calling FINALI** on page 369.

The function calls for a Finalist IMS user driver calling FINALI are:

Table 175: Finalist IMS FINALI Function Calls

Function Call	Description
0	Unconditionally getmain work areas and perform program loads necessary to process address lookup requests. Pointer to getmained work area is stored in user's call area in the FINAL-FILLER field. This pointer value must be maintained between calls to FINALI for proper resource allocation/deallocation purposes. A corrupted pointer may result in storage-related abends such as S80A, S106, or S878. S0C4 abends may also result.
9	Frees getmained storage and issues program deletes for all loaded modules. Placement of the FUNCTION-9 call in on-line depends on the ability to maintain the pointers in the FINAL-FILLER field of the call area by the user application.

For proper resource utilization it is imperative that each FUNCTION-0 call have a corresponding FUNCTION-9 call with the FINAL-FILLER field intact from the FUNCTION-0 call. With the importance of this in mind, you are free to issue multiple 4, 5, 6, and/or 7 calls between the FUNCTION-0 and FUNCTION-9.

The following table describes the function calls for a Finalist/IMS user driver calling FINALB:

Table 176: Finalist IMS FINALB Function Calls

Function Call	Descriptions
0	Unconditionally getmain work areas and perform program loads necessary to process address lookup requests. Pointer to getmained work area is stored in user's call area in the FINAL-FILLER field. This pointer value must be maintained between calls to FINALB for proper resource allocation/deallocation purposes. A corrupted pointer may result in storage-related abends such as S80A, S106, or S878. S0C4 abends may also result.
9	Frees getmained storage and issues program deletes for all loaded modules. If the Finalist Summary Reports were requested, the reports are produced on the Function-9 call.

For proper resource utilization it is imperative that each FUNCTION-0 call have a corresponding FUNCTION-9 call with the FINAL-FILLER (COBOL) and $REREGS (Assembler) field intact from the FUNCTION-0 call. With the importance of this in mind, you are free to issue multiple 4, 5, 6, and/or 7 calls between the FUNCTION-0 and FUNCTION-9.

Using DL/I Data Files

Update the application program with changes to the entry point to receive the PCB addresses from IMS. If the application program is written in COBOL, it must call the program LPFNPCB to set the

PCB address that must be passed to the Finalist interface program, FINALI. Add the following COBOL linkage code to your application program.

```
WORKING-STORAGE SECTION.
       COPY LPFNCL01.
LINKAGE SECTION.
      *=*============================================================*=*

      *=* THIS PROGRAM USES 8 PCBS. -                                *=*

      *=* (1) AN I/O PCB TO COMMUNICATE WITH THE TERMINAL            *=*

      *=*     AND RECIEVE MESSAGES FROM THE TERMINAL.                *=*

      *=* (2) A PCB THAT POINTS TO THE DL/I DATAFILE.                 *=*

      *=* (3) A PCB THAT POINTS TO THE DL/I CITYFILE.                 *=*

      *=* (4-8) PCB'S THAT POINT TO ANCILLARY DATABASES.              *=*

      *=*============================================================*=*

       COPY LPCFIPCB.
       COPY LPCFDPCB.
       COPY LPCFCPCB.
      *=*============================================================*=*
      *=*                                                            *=*
      *=* THE LENGTH OF THE FOLLOWING PCB'S ARE NOT EXACT NOR         *=*
      *=* DO THEY NEED TO BE.                                         *=*
      *=*                                                            *=*
      *=*============================================================*=*
       01 FNEWS-PCB            PIC  X(42).
       01 FNLOT-PCB            PIC  X(42).
       01 DPV-PCB              PIC  X(42).
       01 LLK-PCB              PIC  X(42).
       01 SLK-PCB              PIC  X(42).
       01 RDI-PCB              PIC  X(42).
       01 PBK-PCB              PIC  X(42).
PROCEDURE DIVISION.
       STARTING SECTION.
           ENTRY 'DLITCBL' USING I-O-PCB,   DATAFILE-PCB, CITY-PCB,
                                 FNEWS-PCB, FNLOT-PCB,
                                 DPV-PCB,   LLK-PCB,      SLK-PCB,
                                 RDI-PCB, PBK-PCB.
```

Prior to calling Finalist, make a call to LPFNPCB. This call is necessary to get the PCB addresses to be passed to Finalist. Execute the following calling sequence to PSBs one time prior to each initialization (Function-0) call to FINALI for Finalist processing.

```
CALL 'LPFNPCB' USING DATAFILE-PCB IMS-INITDAT.
           CALL 'LPFNPCB' USING CITY-PCB     IMS-INITCITY.
           CALL 'LPFNPCB' USING FNEWS-PCB    PCBFNEWS.
```

```
           CALL 'LPFNPCB' USING FNLOT-PCB    PCBFNLOT.
           CALL 'LPFNPCB' USING DPV-PCB      PCBFNDPV.
           CALL 'LPFNPCB' USING LLK-PCB      PCBFNLLK.
           CALL 'LPFNPCB' USING SLK-PCB      PCBFNSLK.
           CALL 'LPFNPCB' USING RDI-PCB      PCBFNRDI.
           CALL 'LPFNPCB' USING PBK-PCB      PCBFNPBK.
           MOVE '0' TO FINAL-FUNCTION-CODE.
           CALL 'FINAL' USING FINAL-CALL-AREA
               RETURNING FINAL-RETURN-CODE
```

The PCB fields are initialized by calling program "LPFNPCB". These fields are defined in the supplied source member, LPFNCL01.

Checking the Return Code After Calling FINALI

The Finalist interface program, FINALI, sets a return code in the Finalist callarea to indicate the success or failure of the interfaces initialization status. For COBOL programs, check field FINAL-ONLINE-RETURN-CODE, defined in the supplied COBOL copybook, LPFNCL01. The following table lists the possible return values.

Table 177: FINALI Return Codes

Value	Description
0	No errors.
2	Warning: database expires end-of-month. This message is only a warning and indicates that the Finalist database will expire soon. Processing may continue normally.
3	Product expires at the end of the month.
4	Warning: database has expired. This message indicates that the Finalist database has expired. It has exceeded the USPS mandated 105 day usage period. While your application may continue non-certified processing, be aware that the returned information may no longer be accurate. You must install the current monthly or bimonthly database before any CASS-certified processing can be performed.
6	Warning: Finalist batch product has expired. This is a warning to alert the driver program that the Finalist batch product has exceeded the CASS certification date. This indicates that the returned information may no longer be accurate. You must install the most current version of Finalist.

Value	Description
8	Invalid database version. This message indicates that the Finalist database version is invalid for the version of Finalist software accessing it. The interface program terminates processing. Address processing will not be performed, since undesirable results would occur. This message is also used to indicate you are not authorized to use the IMS Finalist On-Line software. You must install the current monthly or bimonthly database.
E	A fatal processing error has occurred. This message indicates that an unexpected error occurred during Finalist processing. Fields CAERRMOD, CAERRSRC, and CAERRDSC (COBOL and ASSEMBLER) contain further information regarding the error. The driver program should interrogate these fields for further information regarding the error
16	Invalid anchor pointer passed to interface. The interface program, FINALI received a non-zero function call and the call area field FINAL-FILLER (COBOL) or $REREGS (Assembler) has a value of either Spaces, zeroes or high-values. This message indicates that the driver has overlaid the anchor field. Repetitive calls to the interface under these circumstances would ultimately result in out-of-storage related problems. Finalist processing is terminated. To resolve this problem, review your driver program to locate the code that is overlaying the anchor field after the initialization (Function-0) call.

Incorporating AddrScan in Your IMS applications.

Calling the AddrScan product in IMS uses the same process as the batch version of AddrScan.

Example

Below is a sample COBOL call statement to invoke AddrScan from your IMS Finalist On-Line transactions, as well as your IMS BMP applications.

```
CALL 'ADDRSCAN' USING ADDR-PASS-AREA.
```

Incorporating Exceptions File Processing in IMS Applications

As in Finalist batch processing, exceptions handling can be performed by converting your Finalist batch exceptions file, EXCPIN, to load module format. This module is loaded during IMS on-line and IMS batch processing and substitutions may be made.

Using the Exceptions Table

To create an exceptions table in Finalist On-Line or to replace a current exceptions table with a new one, follow the steps below.

1. Execute the following JCL. This JCL includes the job stream required to build the table and place the table in the Finalist On-Line executable library.
2. Activate the optional exceptions table feature for Finalist On-Line. If you use the S56LPCH transaction, set the screen's selection switch to Y. If calling the "FINALI" subroutine, set the "CAEXCPSW" field to "Y" in the Finalist passdata area.

z/OS JCL to Build the Exceptions Table

```
1   //STEP1      EXEC PGM=PBFNEXCP
2   //STEPLIB    DD   DSN=hlq.BATCH.LOAD
3   //SYSPRINT   DD   SYSOUT=*
4   //SYSLOG     DD   SYSOUT=*
5   //SYSUDUMP   DD   SYSOUT=*
6   //EXCPIN     DD   *
    L1MANE              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,NOOBJECT,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,
    //           PARM='LIST,MAP,RENT,XREF,AMODE=31,RMODE=ANY'
16  //SYSLIB     DD DUMMY
17  //SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(1,1))
18  //SYSPRINT   DD SYSOUT=*
19  //SYSLMOD    DD DISP=SHR,DSN=hlq.IMS.LOAD
20  //SYSLIN     DD DSN=*.ASMCC.SYSPUNCH,DISP=SHR
    //           DD *
```

```
    NAME PBFNEXTB(R)
    //
```

A number precedes each statement containing a user-defined variable. This number refers to a specific explanation provided next.

Statement Number	Description
1	This statement executes PBFNEXCP. This program processes the exceptions table control cards and creates an output file. The next step inputs this output file to the Assembler program. The output file contains all statements needed to create an exceptions table for use with the Finalist On-Line system.
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.
4	Include this statement if your site requires system messages 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 identical to those used for exceptions table processing in the batch environment. For more information on exceptions table processing, refer to the section "Exceptions Table Option" in your Finalist User's Guide.
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.
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 On-Line programs. The DSN parameter in statement 20 must appear as shown. If not, the Finalist On-Line system can not reference the exceptions table.

IMS Finalist Batch Processing Using DL/I Databases

In addition to running Finalist in an IMS on-line environment, Finalist is capable of running in an IMS batch mode. This method is used when IMS drives the batch process. With this method, a user driver receives control from IMS and calls Finalist. The user driver program calls the FINALB module. JCL STEPLIB or JOBLIB concatenation is required to place the Finalist IMSBATCH.LOAD library in front of the Finalist IMS.LOAD library.

> **Note:** Depending on your use of DFSMDA, DL/I processing may not require the DDs for Finalist databases.

Sample JCL for Using DL/I Databases

```
//DLI      EXEC PGM=DFSRRC00,PARM='DLI,user-driver,psbname'
//STEPLIB  DD DISP=SHR,DSN=hlq.IMSBATCH.LOAD
//         DD DISP=SHR,DSN=hlq.IMS.LOAD
//         DD DISP=SHR,DSN=ims.reslib
//IMS      DD DISP=SHR,DSN=ims.psblib
//         DD DISP=SHR,DSN=ims.dbdlib
//SYSFINAL DD SYSOUT=*
//*******  Finalist files
//CBDATA   DD  DISP=SHR,DSN=hlq.FINALIST.SHISAM.DATAFILE
//CBCTYST  DD  DISP=SHR,DSN=hlq.FINALIST.SHISAM.CITYFILE
//*******  EWS file
//CBEWS    DD  DISP=SHR,DSN=hlq.FINALIST.SHISAM.CBEWS
//*******  eLOT file
//LTDATA   DD  DISP=SHR,DSN=hlq.FINALIST.SHISAM.LOTFILE
//*******  DPV files
//FDPVDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.DPVDB
//FDPVHDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.DPVHDB
//FDPVSDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.DPVSDB
//FDPVSUD DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.DPVSUD
//*******  LACSLink files
//FLLKDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.LLKDB
//FLLKSUD DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.LLKSUD
//*******  SuiteLink files
//FSLKDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.SLKDB
//..remainder of Finalist and optional DD's..
//FRDIDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.RDIDB
//FPBKDB DD DISP=SHR,DSN=hlq.FINALIST.SHISAM.PBKDB
```

Error Handling

This IMS version of Finalist On-Line Windows issues status codes to inform you of errors that may unexpectedly occur while processing the Finalist DL/I databases. This section provides the possible

status codes and descriptions as well as common system errors that may occur while performing the initial load of a DL/I database, or during input/output processing in the DL/I I/O.

Troubleshooting

The following table describes the possible error conditions.

Table 178: Status Codes

Status Code	Description
AB	This code indicates you did not set up the call properly. You did not specify the segment I/O area.
AD	This code indicates you did not set up the call properly. The function parameter is invalid.
AI	This code indicates an open error. You may be missing a DD or have an empty data set.
AJ	This code indicates you did not set up the call properly. The SSA qualification format is invalid.
AK	This code indicates you did not set up the call properly. The SSA field name is invalid.
AM	This code indicates the processing does not match the call. You may have used the wrong PSB.
GE	This code indicates the segment requested was not found. The passed record area may include a record from a previous call.
Spaces	This code indicates a successful call.

Common System Errors

Type	Error	Probable Cause
ABEND	S80A S106 S878	GETMAIN failure. Verify that user drivers make one termination (Function-9) call for each initialization (Function-0) call. S80A and S106 abends occur when a user application calls FINALI and has not retained the anchor pointer in the FINAL-FILLER field from the call area. If the user application cannot ensure the integrity of the pointer values in the FINAL-FILLER field from the call area, modify the application to make three calls per address: 1. Function-Zero call for initialization 2. Function-Four (or Five) call 3. Function-Nine call for termination
	U16	Invalid function-code sequence (user driver) (i.e., call FINALI with function 4 without preceding function 0).
	U17	Invalid function-code sequence (user driver) (i.e., call FINALI with function 4 without preceding function 0).
Message	DFSO64	No such transaction code. Verify that you entered a trailing blank after the transaction name.

Generating CI Return Codes Using Native API Structures

The FNSOURCE installation library includes the FNCIRCAC sample program that demonstrates how to generate the Compatibility Interface (CI) style Finalist return codes using the native API structures. You can use this sample C program to learn the correlation between the native structures and the CI Return, Reason, and Address Info codes. As a sample program, no support is provided for this routine. However, you are free to modify or incorporate this into your own processing. To rewrite this sample program into COBOL, review **Using the FINAL Subroutine** on page 350 for information on the PBITBYTE routine. FNCIRCAC is available as a callable load module in z/OS environments (z/OS, CICS, and IMS).

4 - Programming Tips and Techniques

In this section

Programming Techniques Overview

This chapter provides programming techniques for a variety of languages. This chapter includes information on:

- Making calls to PBFN from programs written in various languages
- Code fragments that let you access a particular API, namely the PBFNInit function

When Mixed Language Programming is Needed

If your application needs to call programs written in other languages, you need to implement mixed-language programming techniques. You might need to use mixed language programming if:

- A commercial language other than your own offers a subprogram
- A different language describes algorithms more naturally

Coding Notes for All Languages

You must set the version number in all structures to PBFN_CURRENT_VERSION. For C, there are defines set up in the header file to accomplish this step.

When initializing strings in structures, do not set the strings to blanks. Initialize the strings with a null terminating character in position 1. Blanks in a string can be considered valid addressing input and may affect the way an address codes.

PBFN Postal Coding Library

The PBFN postal coding library was designed for easy integration into your existing applications. The PBFN postal coding library is:

- A set of functions (or APIs) that are callable from a variety of programming languages that include C, C++, and Visual C++™
- Written in C language

- Distributed as a Dynamic Link Library (DLL) for Windows, an archive library for UNIX users, and as a multi-entry load module for z/OS

C Language

PBFN is written in C language. Therefore, if your application programming language is C, call the PBFN API as you would any other library function.

Function Call Definition

```
short __stdcall PBFNInit(char *, PBFNExtendedErrorDef*,PBFNSetupDef*,
                                    void*,
                                    void*,
                                    void*,
                                    void*,
                                    void*,
                                    void):
```

Note: The header file contains all the data structure definitions and function prototypes for C and C++ applications.

C Code Example

A C code example is provided as part of the Finalist installation.

C++ Language

If you have written your applications in the C++ language, you can call the PBFN APIs using the same calling conventions as in the C language. However, in C++, you include the header file differently. Include the header file in your application files as demonstrated in the C code example.

Using DLLs With Other Programming Languages

This section provides guidelines on using DLLs with other programming languages.

Dynamic-Link Library Guidelines

Many programming languages specifically made for the 32-bit Microsoft Windows environment call a routine written in a Dynamic-Link Library (DLL). The calling language (for example, Visual Basic) must have the ability to call a DLL and meet the requirements described next.

- Allow two separate kinds of routine calls
  - Function — A call that returns a value
  - Subroutine or procedure — A call with no return value
- Allow passing a variable by reference and by value
- Have supported data types

Table 179: Supported Data Types

Data Type	Description
short	16-bit integer
integer	32-bit integer
single	32-bit floating point

Data Type	Description
double	64-bit floating point
long	32-bit integer
character	8-bit character

- Have a data type that resembles a structure or class. For example, in Visual Basic this data type is called a Type.

Passing Data to a C struct

Data is often stored as a type referred to as a structure. The C language keyword to define such a data type is struct. An entire structure can be passed to a routine by reference. However, other programming languages may call this data type by another name. For example, in Microsoft Visual Basic, a structure is called a Type. Regardless of the name of the data type, the data type must be passed by reference instead of by value so that the called routine may return data in its proper positions to the calling routine. For example, refer to the following C structure.

```
struct {
     unsigned int  lVersion;
     char          szFirstName[30];
     char          szLastName[40];
     int           nBalance;
} strCStruct;
```

Note: Character arrays in C always end with a null ('\0') but will be the same length as an identical string defined in another programming language. By contrast, the BSTR data type (a length-prefixed string) stores the string length in its definition. However, you should check with the documentation for your compiler to confirm that the null terminator is ignored.

Limitations of the Language

Not all programming languages support the use of pointers. Even though you may be able to pass arguments by reference to a routine, that does not guarantee support of pointers. Without pointer support, you cannot pass a pointer to a function. This may limit the level of communication from the calling language to the DLL because it cannot take advantage of accessing memory directly.

Additionally, a language that does not support pointers cannot access pointers to data elements inside a structure individually. For example, the structure shown next contains a nested structure. It also contains a pointer to a function entry point.

```
struct strStruct {
     long lLongDataElement;
     int  nIntegerDataElement;
     char szCharacterArrayElement[255];
     long (*FunctionEntryPoint)();
};
struct {
     char   szCharacterArrayTwo[127];
     long   lLongValue;
     struct strStruct strInside;
} strStructInStruct;
```

A language that does not support pointers cannot make a call to the address of the function entry pointer stored in FunctionEntryPoint. The value of FunctionEntryPoint will be a useless number to such a language.

Without having the ability to pass a pointer to a variable within a structure, the address of the variable shown next is inaccessible. It is impossible to dereference a pointer to it.

```
strStructInStruct.strInside.szCharacterArrayElement
```

This means that you cannot obtain the address like you could in C:

```
&strStructInStruct.strInside.szCharacterArrayElement
```

Normally, C manipulates data in character arrays by reference to its pointers to the array. In doing so, it directly addresses memory.

> **Note:** Many routines in a DLL require the use of pointers to functions. A programming language that does not have a pointer data type cannot access these functions.

Calling DLLs from Microsoft Visual Basic Programs

Visual Basic .NET

Visual Basic .NET must first declare a routine within the program. A sample Declare Function command is listed below:

```
Declare Function FunctionName Lib "PBFN.dll" ( _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr, _
    ByRef SetupDef As PBFNSetupDefinition, _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr, _
    ByVal initType As IntPtr _
        ) As Integer
```

Note: The type PBFNExtendedErrorDef and PBFNSetupDef types are user-defined types which correspond to a data structure necessary for passing data to the specific DLL function.

By default, Visual Basic passes all arguments by reference. This means that it passes the actual address of the data rather than the data itself. Much of the data that is retrieved from a DLL is stored as a reference to a structure. For additional information on declaring external subroutines and functions, see the chapter, "Calling Procedures in DLLs", Programmer's Guide, Microsoft Visual Basic.

Microsoft Visual Basic .NET Code Examples

Microsoft Visual Basic .NET code examples are provided as part of the Finalist installation.

Calling the Finalist APIs from COBOL Programs (z/OS)

The Finalist executable is distributed as a dynamic link library (DLL). A DLL is a load module or a program object that can be accessed from other separate load modules. A DLL differs from a traditional load module in that it exports definitions of programs, functions, or variables to DLLs, DLL applications, or non-DLLs. You do not need to link the target routines into the same load module as the referencing routine. When an application references a separate DLL for the first time, the system automatically loads the DLL into memory. Calling a program in a DLL is similar to calling a load module with a dynamic call.

For applications (for example, Language Environment enclaves) that are structured as multiple, separately bound modules, use exclusively one form of linkage between modules. You can use dynamic call linkage or DLL linkage. DLL linkage refers to a call in a program that is compiled with the DLL and NODYNAM options where the call resolves to an exported name in a separate module.

Calling the Finalist DLL APIs Using Standard Linkage

You can call the Finalist product APIs using standard (non-DLL) linkage without using ProgramName(LongMixed). For example:

1. You generate the following call to a Finalist API:

```
CALL 'PBFNProcess' USING(…) RETURNING XXX
```

2. The COBOL compiler truncates the name to eight characters and forces the entire name to upper case:

```
CALL 'PBFNPROC' USING(…) RETURNING XXX
```

Finalist supports the use of standard linkage and upper case names for the these product APIs only.

Standard linkage	DLL/LongMixed
PBFNINIT	PBFNInit
PBFNPROC	PBFNProcess
PBFNTERM	PBFNTerminate
PBFNCASE	PBFNCase
PBFNSTAT	PBFNStats
PBFNDECR	PBFNDecrypt
PBFNINFO	PBFNInfo

The Finalist database APIs (for example, PBFNCreateSuggestionList, PBCSGetZipList) are not supported with standard linkage and require DLL Linkage as described in **Calling the Finalist APIs Using DLL Linkage** on page 384 or the use of the calling method described in **Calling the Finalist APIs Using Dynamic Call Linkage** on page 385.

Calling the Finalist APIs Using DLL Linkage

To call the Finalist APIs using DLL linkage, you must compile your COBOL programs with the DLL, NODYNAM, PGMNAME(LONGMIXED), and RENT options. When you compile a COBOL program with the DLL option, the program is enabled for DLL support. Applications that use DLL support must be re-entrant. You must compile them with the RENT compiler option and link them with the RENT binder option.

You need to use the Language Environment prelinker before standard linkage editing. After compiling your COBOL DLL program(s), prelink the object decks to form a single object module. Include the Finalist definition side files (created when you prelinked the Finalist objects) together with the object decks that are input to this prelink step. The side files instruct the prelinker to resolve the symbolic references to the Finalist APIs in your COBOL program to the symbols exported from the Finalist DLLs.

Use the linkage editor or binder as usual to create the load module from the object module produced by the prelinker. Specify the RENT option of the linkage editor or binder.

> **Note:** All components of a DLL application must have the same AMODE. The automatic AMODE switching normally provided by COBOL dynamic calls is not available for DLL linkages.

Calling the Finalist APIs Using Dynamic Call Linkage

It is possible to dynamically call the Finalist APIs from a COBOL program that is compiled with the NODLL option. There are some restrictions for using this method. You can use one of the methods described below.

- Include the Finalist object modules in the load module that calls the Finalist APIs. Use static calls from the COBOL non-DLL program(s) to call the Finalist APIs.
- Call the Finalist APIs from your COBOL non-DLL program(s) with CALL procedure-pointer, where the procedure-pointer is set to a function descriptor of the Finalist API to be called. You can obtain the address of the function descriptor for the Finalist API in the DLL by calling the C routine PBFNCOB.

The following is sample code that shows how a COBOL non-DLL program can call the Finalist PBFNInit API.

```
   WORKING-STORAGE SECTION.
   01  WS-DLL-INFO.
       05  WS-DLL-LOAD-MOD         PIC X(012).
       05  WS-DLL-FUNCTION         PIC X(160).
       05  WS-DLL-FUNC-PTR         USAGE POINTER.
   01  WS-DLL-RC                   PIC S9(09)  BINARY.
   01  WS-PBFN-INIT                USAGE PROCEDURE-POINTER.
   PROCEDURE DIVISION.
  **** Move in the load module (DLL) and function names. They
  **** must be null terminated.
       MOVE Z'PBFN'                TO WS-DLL-LOAD-MOD.
       MOVE Z'PBFNInit'            TO WS-DLL-FUNCTION.
  **** Call the C routine to load the DLL and to get the
  **** function descriptor address.
       CALL 'PBFNCOB' USING BY REFERENCE WS-DLL-INFO,
                            BY REFERENCE WS-DLL-RC.
       IF WS-DLL-RC = ZERO
  ****     Move the function pointer to a procedure pointer
  ****     so that we can use the CALL statement to call the
  ****     function in the DLL.
           SET WS-PBFN-INIT        TO WS-DLL-FUNC-PTR
  ****     Call the function in the DLL.
           CALL WS-PBFN-INIT USING BY VALUE 0,
                                   BY REFERENCE PBFN-IERR-EXTERROR,
                                   BY REFERENCE PBFN-GCFG-SETUP,
                                   BY VALUE 0,
                                   BY VALUE 0,
                                   BY VALUE 0,
                                   BY VALUE 0,
                                   BY VALUE 0,
                                   BY VALUE 0,
                             RETURNING WS-FINAL-RC
       END-IF.
```

Sample COBOL Driver Program

A sample COBOL driver program is provided below to help you develop a COBOL driver program for your environment if needed. This is a sample intended for illustration purposes only. It is not intended for use in a production environment and no support will be provided for such usage.

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. 'FINALDR8'.
      *      REMARKS.    THIS IS AN EXAMPLE OF AN MVS FINALIST COBOL
      *                  DRIVER PROGRAM. THIS PROGRAM GETS AN ADDRESS
FROM
      *                 AN INPUT RECORD, CALLS FINALIST TO CLEANSE THE

      *                  ADDRESS, CHECKS FOR ERRORS RETURNED, AND THEN
      *                  WRITES AN OUTPUT RECORD.
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT USER-FILE-IN ASSIGN TO UT-S-SYSUT1.
           SELECT OUT-FILE ASSIGN TO UT-S-SYSUT2.
      *     SELECT PRINT-FILE ASSIGN TO UT-S-SYSPRINT.
       DATA DIVISION.
       FILE SECTION.
       FD  USER-FILE-IN
           RECORD CONTAINS 600 CHARACTERS
           LABEL RECORDS ARE STANDARD
           BLOCK CONTAINS 0 RECORDS
           DATA RECORD IS RECORD-IN.
       01  RECORD-IN.
           05  FILLER                          PIC X(277).
           05  FIRM-IN                         PIC X(040).
           05  URB-IN                          PIC X(028).
           05  ADDR1-IN                        PIC X(064).
           05  ADDR2-IN                        PIC X(064).
           05  CITY-STATE-IN                   PIC X(042).
           05  FILLER                          PIC X(084).
       FD OUT-FILE
            RECORD CONTAINS 660 CHARACTERS
            BLOCK CONTAINS 0 RECORDS
            LABEL RECORDS ARE STANDARD
            DATA RECORD IS OUT-RECORD.
       01 OUT-RECORD.
        05 USER-RECORD-OUT PIC X(600).
        05 OUT-FINALIST-DATA.
           10 RECORDOUT-FIRM PIC X(071).
           10 RECORDOUT-URB PIC X(032).
           10 RECORDOUT-ADDRESS1 PIC X(080).
           10 RECORDOUT-ADDRESS2 PIC X(080).
```

```
           10 RECORDOUT-UNIT1 PIC X(020).
           10 RECORDOUT-UNIT2 PIC X(020).
           10 RECORDOUT-CITY PIC X(070).
           10 RECORDOUT-STATE PIC X(002).
           10 RECORDOUT-ZIP PIC X(005).
           10 RECORDOUT-ZIP4 PIC X(004).
           10 RECORDOUT-CRRTE PIC X(004).
           10 RECORDOUT-PMBUNIT PIC X(017).
           10 RECORDOUT-FIPSCODE PIC X(005).
           10 RECORDOUT-COUNTY-NAME PIC X(028).
           10 RECORDOUT-ADVANCEBARCODE PIC X(014).
           10 RECORDOUT-5-DIGIT-BARCODE PIC X(008).
      * 600 bytes + 460 bytes = 1060-byte records
      * ************************************************************
       WORKING-STORAGE SECTION.
       01 COUNTERS.
        05 RECORDS-READ PIC S9(9) COMP-3 VALUE 0.
        05 RECORDS-WRITTEN PIC S9(9) COMP-3 VALUE 0.
       01 MISC-VARIABLES.
        05 EOF-SW PIC X(3) VALUE 'NO '.
      * ============================================================
      * The users can work either with the actual numeric values
      * that are returned from calls to the Finalist functions,
      * or they can use the meanings associated with those values.
      * Since the logic of the return code values in release 8.0 is
      * the REVERSE of those values in release 7.60, I think that
      * using the descriptive meanings defined below could help reduce
      * confusion. For example, coding
      *
      * IF FINALIST-RETURN-INT < PBFN-SUCCESS
      * instead of
      * IF FINALIST-RETURN-INT < 1
      *
      * makes what is being tested extremely clear.
      * ============================================================
      * ************************************************************
      * PB FINALIST RETURN CODES
      * ************************************************************
        05 FINALIST-RETURN-INT PIC S9(9) COMP.
        05 PBFN-ERROR PIC S9(9) VALUE -1.
      * System Error
        05 PBFN-FAIL PIC S9(9) VALUE 0.
      * Address did not code - can't do any more
        05 PBFN-SUCCESS PIC S9(9) VALUE 1.
      * Address Successfully coded
        05 PBFN-HAVE-WARNING PIC S9(9) VALUE 4.
      * DB expires at end of month
        05 PBFN-DB-EXPIRED PIC S9(9) VALUE 5.
      * DB has expired. CASS processing turned off.
        05 PBFN-ENGINE-WARNING PIC S9(9) VALUE 7.
      * CASS engine expires at end of month
        05 PBFN-NOLOT PIC S9(9) VALUE 8.
      * LOT database not available
```

```
        05 PBFN-ENGINE-EXPIRED PIC S9(9) VALUE 9.
      * CASS engine has expired, CASS processing turned off.
      * ============================================================
      * Where in release 7.60 we had a single copybook for the only
      * Finalist callarea that could be passed on a call to FINAL,
      * with the new Finalist we have a separate copybook for each data

      * structure that could be passed to one or more of the new
      * functions. There are many individual copybooks that are
      * now included, but any given program will only use
      * a subset of them. This program only uses four of them.
      *
      * We are presuming here that all needed processing options and
      * desired reports are defined in the pbfn.cfg file. If there
      * are options we want to use for this specific program, we
      * would include the PBFNSetup structure here. Before the
      * program issues the PBFNInit call, we would set appropriate
      * values into the fields of the PBFNSetup structure, and then
      * include that structure in the call to PBFNInit.
      *
      * NOTE: Finalist structures must be aligned on a 4-byte
      * boundary. The FILLERs below accomplish this.
      *
      * ============================================================
      * ************************************************************
      * COPYBOOK FOR PBFNSetup STRUCTURE
      * ************************************************************
      * 05 FILLER PIC S9(08) BINARY.
        COPY PBFNGCFG.
      * Note: not needed if all parms are handled in pbfn.cfg file.
      * ************************************************************
      * COPYBOOK FOR PBFNAddressDataDef STRUCTURE
      * ************************************************************
        05 FILLER PIC S9(08) BINARY.
        COPY PBFNADRS.
      * ************************************************************
      * COPYBOOK FOR RETURN PBFNAddressDataDef STRUCTURE
      * ************************************************************
        05 FILLER PIC S9(08) BINARY.
        COPY PBFNRRTN.
     * ************************************************************

       PROCEDURE DIVISION.
           OPEN INPUT  USER-FILE-IN
                       OUTPUT OUT-FILE.
           INITIALIZE PBFN-GCFG-SETUP.
      * ============================================================
      * Important coding note:
      * --------- ------ ----
      * By default, in a call to a subroutine COBOL passes the memory
      * addresses of the data items listed in the USING clause. With
      * these "pointers," the subroutine has access to the contents
      * of the memory locations, so that any changes it makes there
```

```
      * are available to the calling program. This is known as passing
      * by reference. The functions in Finalist 8.0, however, expect
      * to receive the actual VALUES of some parameters rather than
      * pointers to those values. Thus it is necessary to explicitly
      * specify BY VALUE (or BY CONTENT) for such parameters. While
      * it is not necessary, then, to specify BY REFERENCE in the
      * call, being specific for each type of parameter eliminates a
      * potential source of error.
      *
      * Also note that calls to Finalist 8.0 are calls to FUNCTIONS
      * that return a value -- so they need a RETURNING clause to
      * received that value returned from the function.
      * ================================================================
           CALL 'PBFNInit' USING
                           BY VALUE 0,
                           BY REFERENCE PBFN-GCFG-SETUP,
                           BY VALUE 0,
                           BY VALUE 0,
                           BY VALUE 0,
                           BY VALUE 0,
                           BY VALUE 0,
                           BY VALUE 0,
                           BY VALUE 0,
                           RETURNING FINALIST-RETURN-INT.
      * ================================================================
      * Instead of making an initial call to FINAL with a function
      * code of zero, we call PBFNInit. We pass the "extended error"
      * data structure so that we will have more information in the
      * event of a problem.
      * ================================================================
           IF FINALIST-RETURN-INT < PBFN-SUCCESS THEN
               DISPLAY ' ERROR ON PBFNInit -- ERROR: '
                       PBFN-GCFG-ERRORNUM
               DISPLAY ' MESSAGE: ' PBFN-GCFG-MESSAGECODE
               CLOSE USER-FILE-IN
                     OUT-FILE
               STOP RUN
           END-IF.
           DISPLAY ' SUCCESSFUL INIT CALL: ' FINALIST-RETURN-INT.
           READ USER-FILE-IN AT END MOVE 'YES' TO EOF-SW.
           DISPLAY ' INITIAL READ, EOF-SW IS ' EOF-SW.
           PERFORM 100-MAINLINE THRU 100-EXIT
                   UNTIL EOF-SW EQUAL 'YES'.
      * ================================================================
      * When done processing the input, we call PBFNTerminate to shut
      * down Finalist and produce the reports.
      * ================================================================
           DISPLAY ' NUMBER OF RECORDS READ IS: ' RECORDS-READ.
           DISPLAY ' NUMBER OF RECORDS WRITTEN: ' RECORDS-WRITTEN.
           CALL 'PBFNTerminate' USING
                                BY VALUE 0,
                                BY VALUE 0,
                                BY VALUE 0,
```

```
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              RETURNING FINALIST-RETURN-INT.
           CLOSE USER-FILE-IN
                 OUT-FILE.
           STOP RUN.
       100-MAINLINE.
           PERFORM 1000-INITIALIZE-STRUCTURES THRU 1000-EXIT.
           ADD 1 TO RECORDS-READ.
           MOVE ADDR-IN TO PBFN-ADRS-ADDRESS1.
           MOVE CITY-STATE-IN TO PBFN-ADRS-CITY.
           MOVE ZIP-IN TO PBFN-ADRS-ZIP.
           CALL 'PBFNProcess' USING
                              BY REFERENCE PBFN-ADRS-ADDRESSDATA,
                              BY REFERENCE PBFN-RRTN-RRTN,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              BY VALUE 0,
                              RETURNING FINALIST-RETURN-INT.
      * PLACE ADDITIONAL PROCESSING HERE
           MOVE RECORD-IN TO USER-RECORD-OUT.
           IF FINALIST-RETURN-INT >= PBFN-SUCCESS THEN
               MOVE PBFN-RRTN-FIRM     TO RECORDOUT-FIRM
               MOVE PBFN-RRTN-URB      TO RECORDOUT-URB
               MOVE PBFN-RRTN-ADDRESS1 TO RECORDOUT-ADDRESS1
               MOVE PBFN-RRTN-ADDRESS2 TO RECORDOUT-ADDRESS2
               MOVE PBFN-RRTN-UNIT1    TO RECORDOUT-UNIT1
               MOVE PBFN-RRTN-UNIT2    TO RECORDOUT-UNIT2
               MOVE PBFN-RRTN-CITY     TO RECORDOUT-CITY
               MOVE PBFN-RRTN-STATE    TO RECORDOUT-STATE
               MOVE PBFN-RRTN-ZIP      TO RECORDOUT-ZIP
               MOVE PBFN-RRTN-ZIP4     TO RECORDOUT-ZIP4
               MOVE PBFN-RRTN-CRRTE    TO RECORDOUT-CRRTE
               MOVE PBFN-RRTN-PMBUNIT  TO RECORDOUT-PMBUNIT
               MOVE PBFN-RRTN-FIPSCODE TO RECORDOUT-FIPSCODE
               MOVE PBFN-RRTN-COUNTY-NAME
                                       TO RECORDOUT-COUNTY-NAME
               MOVE PBFN-RRTN-ADVANCEBARCODE
                                       TO RECORDOUT-ADVANCEBARCODE
               MOVE PBFN-RRTN-FIVEDIGITBARCODE
                                       TO RECORDOUT-5-DIGIT-BARCODE
           ELSE
               MOVE PBFN-RRTN-ERROR TO RECORDOUT-URB
```

```
            MOVE PBFN-RRTN-MESSAGECODE TO RECORDOUT-FIRM
        END-IF.
        WRITE OUT-RECORD.
        ADD 1 TO RECORDS-WRITTEN.
        READ USER-FILE-IN
             AT END MOVE 'YES' TO EOF-SW.
 100-EXIT. EXIT.
 1000-INITIALIZE-STRUCTURES.
* ================================================================
* We use separate structures for input and output here. A single
* ADRS structure could be used for both input and output, but
* the input data would be over-written by the call to Finalist.
* ================================================================
        INITIALIZE OUT-FINALIST-DATA.
        INITIALIZE PBFN-ADRS-ADDRESSDATA.
        INITIALIZE PBFN-RRTN-RRTN.
* ================================================================
* The lengths for the fields in the input structure could be
* set to the actual length of the corresponding items in the
* input data as the input data loaded into the structure --
* or you can just set them to the defaults, as shown here.
* ================================================================
        ADD 71 TO PBFN-ADRS-FIRM-LEN.
        ADD 32 TO PBFN-ADRS-URB-LEN.
        ADD 80 TO PBFN-ADRS-ADDRESS1-LEN.
        ADD 80 TO PBFN-ADRS-ADDRESS2-LEN.
        ADD 20 TO PBFN-ADRS-UNIT1-LEN.
        ADD 20 TO PBFN-ADRS-UNIT2-LEN.
        ADD 72 TO PBFN-ADRS-CITY-LEN.
        ADD 4 TO PBFN-ADRS-STATE-LEN.
        ADD 16 TO PBFN-ADRS-PMBUNIT-LEN.
        ADD 30 TO PBFN-ADRS-COUNTYNAME-LEN.
        ADD 80 TO PBFN-ADRS-ALTERNATESTREET-LEN.
        ADD 30 TO PBFN-ADRS-FULLCITYNAME-LEN.
        ADD 30 TO PBFN-ADRS-ABBRCITYNAME-LEN.
        ADD 30 TO PBFN-ADRS-NONMAIL-CITYNAME-LEN.
        ADD 20 TO PBFN-ADRS-DPVFOOTNOTE-LEN.
* ================================================================
* Set the lengths for the members of the output structure to
* their default max values. Any fields defined with zero length
* will be populated by zero bytes on the return.
* ================================================================
        ADD 71 TO PBFN-RRTN-FIRM-LEN.
        ADD 32 TO PBFN-RRTN-URB-LEN.
        ADD 80 TO PBFN-RRTN-ADDRESS1-LEN.
        ADD 80 TO PBFN-RRTN-ADDRESS2-LEN.
        ADD 20 TO PBFN-RRTN-UNIT1-LEN.
        ADD 20 TO PBFN-RRTN-UNIT2-LEN.
        ADD 72 TO PBFN-RRTN-CITY-LEN.
        ADD 4 TO PBFN-RRTN-STATE-LEN.
        ADD 16 TO PBFN-RRTN-PMBUNIT-LEN.
        ADD 30 TO PBFN-RRTN-COUNTYNAME-LEN.
        ADD 30 TO PBFN-RRTN-ALTERNATESTREET-LEN.
```

```
            ADD 30 TO PBFN-RRTN-FULLCITYNAME-LEN.
            ADD 30 TO PBFN-RRTN-ABBRCITYNAME-LEN.
            ADD 30 TO PBFN-RRTN-NONMAIL-CITYNAME-LEN.
            ADD 20 TO PBFN-RRTN-DPVFOOTNOTE-LEN.
       1000-EXIT. EXIT.
```

Sample COBOL Driver Program JCL

The following is sample z/OS JCL to compile, pre-link, and link the COBOL user driver above. This is a sample intended for illustration purposes only. It is not intended for use in a production environment and no support will be provided for such usage.

```
//jobname JOB 'FINALIST 77 DRIVER',REGION=4M,MSGCLASS=X,
//            CLASS=C,NOTIFY=D966DES
//COMPILE EXEC PGM=IGYCRCTL,PARM=('DLL,PGMN(LM),LIB')
//*********************************************************
//*   Note that PGMN(LM) stands for PGMNAME(LONGMIXED). This
//*   parm is required since the new Finalist C functions
//*   have names like PBFNProcess, PBFNTerminate, etc.
//*********************************************************
//SYSIN    DD DISP=SHR,DSN=yoursourcelib(COBOL77H)
//SYSLIB   DD DISP=SHR,DSN=hlq.FINALIST.COBCOPYL
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT2   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT3   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT4   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT5   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT6   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSUT7   DD UNIT=VIO,SPACE=(CYL,(2,2))
//SYSLIN   DD DSNAME=&&LOADSET,UNIT=VIO,DISP=(MOD,PASS),
//            SPACE=(CYL,(3,1),RLSE)
//*********************************************************
//*   A pre-link step is needed to enable a COBOL program to
//*   call C functions with long, mixed-case program names.
//*********************************************************
//PLKED   EXEC PGM=EDCPRLK,PARM='MAP,NOER',
//             COND=(8,LT,COMPILE),
//             REGION=2048K
//STEPLIB  DD  DSNAME=CEE.SCEERUN,DISP=SHR
//SYSMSGS  DD  DISP=SHR,DSN=CEE.SCEEMSGP(EDCPMSGE)
//SYSLIB   DD  DISP=SHR,DSN=CEE.SCEECPP
//SYSIN    DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD  DISP=SHR,DSN=hlq.BATCH.IMPORT(PBFN)
//SYSMOD   DD  DSNAME=&&OBJLIB(COBOL77P),DISP=(NEW,PASS),
//             SPACE=(TRK,(15,15,15),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
```

```
//*
//LKED   EXEC PGM=HEWL,COND=(8,LT,COMPILE),REGION=1024K,
//        PARM='AMODE=31,RMODE=ANY,MAP,RENT,REUS,XREF,LIST,CALL'
//SYSLIB   DD  DSNAME=CEE.SCEELKED,DISP=SHR
//         DD  DSNAME=&&OBJLIB,DISP=(OLD,DELETE)
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DDNAME=SYSIN
//SYSLMOD  DD  DSNAME=your.output.loadlib,
//             SPACE=(TRK,(10,10,1)),
//             UNIT=VIO,DISP=(MOD,PASS),DSNTYPE=LIBRARY
//SYSUT1   DD  UNIT=VIO,SPACE=(TRK,(10,10))
//SYSIN    DD  *
 INCLUDE SYSLIB(COBOL77P)
 ENTRY FINALDR8
 NAME COBOL77H(R)
//*
//
```

Interrogating Bits in a Byte in COBOL

COBOL does not provide a method for interrogating bits in a byte. Finalist provides a utility program, PBITBYTE, that expands a bit into eight characters to enable a COBOL program to interrogate the bits. The output value is a series of character 1s and 0s representing the bit values of the input field. The following is a sample of a program calling PBITBYTE.

```
WORKING-STORAGE SECTION.
       77  PBITBYTE   PICTURE X(08) VALUE 'PBITBYTE'.
       01  WORK-AREA.
           03 BYTE-IN                      PIC  X(01).
           03 BYTE-8-OUT                   PIC  X(08).
...
       PROCEDURE DIVISION.
       CALL PBITBYTE USING  BYTE-IN
                            BYTE-8-OUT.
```

The first parameter is an input field and points to a single byte. The second parameter is an output parameter that is eight (8) characters in length and contains the byte expanded in character format. For example, if the input field is X'F3', the output field contains '11110011'. If the input field is X'A8', the output field contains '10101000'. This allows the COBOL program to interrogate fields in Finalist structures like PBFN-ADRS-ADDRESS-TYPE.

2 Blue Hill Plaza, #1563
Pearl River, NY 10965
USA

www.precisely.com