



Address Matching System  
Version 2.80.01.M  
Release Notes

ADDRESS MANAGEMENT  
NATIONAL CUSTOMER SUPPORT CENTER  
UNITED STATES POSTAL SERVICE  
6060 PRIMACY PKWY STE 201  
MEMPHIS TN 38188-0001  
(877) 640-0724

## Contents

Purpose .....	page 3
Interface Changes .....	page 4
Added Functionality .....	page 10
Bug Fixes .....	page 11

## Purpose

This document provides information on the changes to the United States Postal Service® Address Matching System API (AMS API) for the CASS™ Cycle M release.

It is requested that you implement and evaluate this release of AMS in a test environment at your earliest convenience. You are encouraged to promptly report any problems that you may discover to our Customer Care Department at 877-640-0724, or email us at [amscd.ncsc@usps.gov](mailto:amscd.ncsc@usps.gov).

**Note:** The software should be updated prior to Cycle L expiration date of JULY 31, 2009

# Interface Changes

## Section Notes:

This section is intended to provide an overview of the changes that have been made to the interface layer of the AMS library. It is advisable for you to review the entire definition of each structure listed in the zip4.h file to better understand the context and extent of these changes.

## 1. ZIP4\_PARM structure

The changes are listed in order from top of the structure to the bottom. These changes are described in more detail below the table.		
Variable	Purpose	A/C/D
char stelnkfoot[3+1]	This field has been added to provide the status of a Suite <sup>Link</sup> lookup.	Addition
char punit2[4+1]	This field will contain the second unit designator to the right of the street name if it exist.	Addition
char psnum2[8+1]	This field will contain the second secondary number to the right of the street name if it exist.	Addition
char foot.o	This footnote identifies multiple responses that can be resolved using the lowest add-on code.	Change – Definition Only
char foot.f0 . . . foot.f5	Divided rsvdfoot[6] for future use.	Change

### NOTES:

#### A. stelnkfoot

Reserved space was used for this field.

This field will contain one of three values "", "00", or "A " to identify the status of a Suite<sup>Link</sup> lookup.

" " – Lookup was not attempted

"00" – No match

"A " – Match

#### C. punit2

Reserved space was used for this field.

This field will contain the second unit designator to the right of the street name if it exist.

#### D. psnum2

Reserved space was used for this field.

This field will contain the second secondary number to the right of the street name if it exist.

#### E. foot.o

O – Previously a reserved footnote.

Now O identifies multiple responses that can be resolved by choosing the lowest add-on code.

#### F. foot.f0 ... foot.f5

Reserved space was used for these fields.

These footnotes are for future use.

## 2. TAbbrSt structure

This structure represents a ZIP+4 database record (ADDR\_REC structure) with the addition of a delivery address line. Pass this structure to the z4ABSQuery() call to obtain abbreviated street info.

```

/*****\
*          ABBREVIATED STREET RECORD          *
*
*   Parameter list for z4ABSQuerySTD()        *
*   NOTE:  Fields names with a leading "sz"  *
*          are null terminated.              *
*
\*****/
typedef struct tagNationalDirectoryFileZip4DetailAbbreviated
{
    char psDetailCode[1];    /* COPYRIGHT DETAIL CODE          */
    char szZipcode[6];      /* ZIP CODE                       */
    char szUpdateKey[11];   /* UPDATE KEY NUMBER             */
    char psActionCode[1];  /* ACTION CODE                   */
    char psRecordType[1];  /* RECORD TYPE                   */
    char szCarrierRt[5];   /* CARRIER ROUTE               */
    char szPreDir[3];      /* PRE-DIRECTIONAL ABBREVIATED  */
    char szStreetName[29]; /* STREET NAME                   */
    char szSuffix[5];      /* SUFFIX ABBREVIATED           */
    char szPostDir[3];     /* POST-DIRECTIONAL ABBREVIATED */
    char szPrimaryL[11];   /* PRIMARY LOW RANGE             */
    char szPrimaryH[11];   /* PRIMARY HIGH RANGE            */
    char psPrimarCode[1];  /* EVEN/ODD/BOTH CODE (PRIMARY NUMBER) */
    char szFirm[41];       /* BUILDING/FIRM NAME           */
    char szUnit[5];        /* UNIT DESIGNATOR ABBREVIATED  */
    char szSecondaryL[9];  /* SECONDARY LOW RANGE          */
    char szSecondaryH[9];  /* SECONDARY HIGH RANGE         */
    char psSecondaryCode[1]; /* EVEN/ODD/BOTH CODE (SECONDARY NUMBER) */
    char szAddonL[5];      /* ADD ON LOW RANGE             */
    char szAddonH[5];      /* ADD ON HIGH RANGE            */
    char psBaseAltCode[1]; /* BASE/ALTERNATE CODE          */
    char psLACS[1];        /* LACS CONVERTED STATUS        */
    char szFinance[7];     /* FINANCE NUMBER               */
    char szState[3];       /* STATE ABBREVIATED           (NOT FILLED) */
    char szCountyNumber[4]; /* COUNTY NUMBER                */
    char szCongressDist[3]; /* CONGRESSIONAL DISTRICT       */
    char szMunicipality[7]; /* MUNICIPALITY CITY/STATE KEY  (NOT FILLED) */
    char szUrbanization[7]; /* URBANIZATION CITY/STATE KEY */
    char szLastLineKey[7]; /* LAST LINE CITY/STATE KEY     */
    char szAddress[51];    /* STANDARDIZED DELIVERY ADDRESS */
} TAbbrSt, *TPAbbrSt;
```

### 3. Z4 ENV structure

The changes are listed in order from top of the structure to the bottom.		
Variable	Purpose	A/C/D
char stelnkpath[300+1]	This field provides the location of the Suite <sup>Link</sup> files as specified during the z4opencfg() call.	Addition
char abrstpath[300+1]	This field provides the location of the abbreviated street name files as specified during the z4opencfg() call.	Addition
char rsvd1[1208]	Reserved space for future use.	Change
char stelnkflag	This field provides the Suite <sup>Link</sup> functionality as specified during the z4opencfg() call.	Addition
char abrstflag	This field provides the abbreviated street name functionality as specified during the z4opencfg() call.	Addition

#### 4. CONFIG\_PARM structure

The changes are listed in order from top of the structure to the bottom. These changes are described in more detail below the table.		
Variable	Purpose	A/C/D
char* stelinkpath	Populate this field with the location to the Suite <sup>Link</sup> data files prior to a z4opencfg() call.	Addition
char* abrstpath	Populate this field with the location to the abbreviated street name data files prior to a z4opencfg() call.	Addition
char rsvd[116]	Reserved space for future use	Change

#### **NOTES:**

##### **A. stelinkpath**

Reserved space was used for this field.

Use the field to provide the location of the Suite<sup>Link</sup> data files.

##### **C. abrstpath**

Reserved space was used for this field.

Use this field to provide the location of the abbreviated street name data files.

##### **D. rsvd**

The size of this field changed from 124 to 116 to accommodate the addition of stelinkpath and abrstpath.

#### **4. Z4OPEN\_PARM structure**

The changes are listed in order from top of the structure to the bottom. These changes are described in more detail below the table.		
Variable	Purpose	A/C/D
char dpvtypeflag	Reserved for future use.	Addition
char stelnkflag	Populate this field with a 'Y' prior to a z4opencfg() call to request Suite <sup>Link</sup> functionality.	Addition
char abrstflag	Populate this field with a 'Y' prior to a z4opencfg() call to request abbreviated street name functionality.	Addition
char rsvd2[492]	Reserved space for future use	Change

#### **NOTES:**

##### **A. dpvtypeflag**

Reserved space was used for this field.

This field is reserved for future use.

##### **A. stelnkflag**

Reserved space was used for this field.

This field is used to enable Suite<sup>Link</sup> functionality. This field should be populate with a 'Y' prior to a z4opencfg() call to enable Suite<sup>Link</sup> otherwise it is disabled.

##### **C. abrstflag**

Reserved space was used for this field.

This field is used to enable abbreviated street name fuctionality. This field should be populate with a 'Y' prior to a z4opencfg() call to enable the abbreviated street name otherwise it is disabled.

##### **D. rsvd2**

The size of this field changed from 495 to 492 to accommodate the addition of dpvtypeflag, stelnkflag and abrstflag.

## **5. Z4DPV devkits**

The sampledp binary and example source code have been removed from the dev\_kits folders. Reference the sample binary and source code found in the AMS dev\_kits folders if a test binary or coding examples are needed.

## Added Functionality

### 1. CASS Significant zero Rule:

DPV validates addresses with a secondary value of zero if the ZIP+4 match is to a highrise exact record showing a significant leading zero in the secondary number field.

### 2. CASS DPV tie-breaking Rule:

DPV will no longer break ties if it violates any existing CASS rule. Examples are the cardinal rule and overlapping ranges.

### 3. CASS Firm Rule:

When a Firm ZIP+4 code is given on input the ZIP+4 code, suite number, and firm name are retained if the ZIP+4 code is a match for the input address. Note that this match can still be made even if the suit number and firm name are invalid.

### 4a. CASS Unique ZIP Rule:

When the input ZIP+4 code consist of a unique 5 digit. The input address and ZIP+4 code are retained.

### 4b. CASS Unique ZIP Rule:

When the input ZIP code is a unique 5 digit an exact match can be made outside the ZIP code.

### 4c. CASS Unique ZIP Rule:

When the input is missing the ZIP code an exact match can be made to a unique ZIP code.

### 4d. CASS Unique ZIP Rule:

When the input ZIP code in not unique an exact match can be made to a unique if an exact match does not exist outside the unique.

5. A multiple response will be returned for ZIP+4 matches that are equal and except carrier route and +4 add-on and ZIP4\_PARAM.foot.o will be set to flag this situation.

6. z4DpvResolvMulti() has been modified to break-ties between unique and non-unique ZIP codes, and addresses that are equal except carrier route and +4 add-on.

7. Calls can be made to z4SLNKQuery() to make use of the Suite<sup>Link</sup> product. Suite<sup>Link</sup> will append secondary information on select group of highrise default address. These addresses generally receive a high volume of default mail.

8. Calls can be made to z4ABSQuery() to make use of 30 character address lines. If an abbreviated alias exist on the ZIP+4 database this call will return the abbreviated alias and the 30 character address line otherwise the input ZIP+4 record and address line are returned.

# Bug Fixes

## 1. Incorrect Firm matches (#2470)

The AMS API was making incorrect firm matches when a partial address was presented on input.

EXAMPLES:

HUNTINGTON DIST CORPS  
HUNTINGTON WV 25701

HUNTINGTON STATE HOSPITAL (INVALID MATCH)  
1530 NORWAY AVE  
HUNTINGTON WV 25705-1358

This has been fixed.

## 2. Fractions as Street Name (#2568)

The AMS API was placing fractional street names as part of the primary number.

EXAMPLE:

775 1/2 ST S (PRIMARY NUMBER = 775 ½)  
SAUK RAPIDS MN 56393

This has been fixed.

## 3. Extraneous Secondary Address Information (#2472)

The AMS API was unable to match to some addresses in which an exceptional secondary element was given with the additional secondary information.

EXAMPLE:

424 C ST NE LOWER LEVEL (IGNORES LEVEL AS A SECONDARY NUMBER)  
WASHINGTON DC 20002

This has been fixed.

## 4. Incorrect PO Box match with input ZIP+4 (#2773)

The AMS API was using only the add-on to identify if a match was made, this caused an incorrect match if multiple ZIP codes within the city-state file contain this add-on.

EXAMPLE:

PO BOX 1348 (SHOULD BE A MULT-RESPONSE)  
NEW YORK NY 10116-1348

This has been fixed.