

---

# **Address Validator**

## **for Oracle Warehouse Builder**

---

## Copyright

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Melissa Data Corporation. This document and the software it describes are furnished under a license agreement, and may be used or copied only in accordance with the terms of the license agreement.

© April, 2005. Melissa Data Corporation. All rights reserved.

Information in this document is subject to change without notice. Melissa Data Corporation assumes no responsibility or liability for any errors, omissions, or inaccuracies that may appear in this document.

## Trademarks

Address Validator is a trademark of Melissa Data Corporation. Windows is a registered trademark of Microsoft Corp. ZIP Code and ZIP+4 are registered trademarks of the United States Postal Service (USPS). All other brands and products are trademarks of their respective holder(s).

Document number: AV0505UG

### **MELISSA DATA CORPORATION**

22382 Avenida Empresa  
Rancho Santa Margarita, CA 92688

Phone: 1-800-MELISSA (1-800-635-4772)

Fax: 949-589-5211

E-mail: [info@MelissaData.com](mailto:info@MelissaData.com)

Web site: [www.MelissaData.com](http://www.MelissaData.com)

**Dear Oracle User,**

I would like to take this opportunity to introduce you to Melissa Data Corp. Founded in 1985, Melissa Data provides data quality solutions, with emphasis on address and phone verification, postal encoding, and data enhancements.

We are a leading provider of cost-effective solutions for achieving the highest level of data quality for lifetime value. A powerful line of software, databases, components, and services afford our customers the flexibility to cleanse and update contact information using almost any language, platform, and media for point-of-entry or batch processing.

This online manual will guide you through the properties and methods of our easy-to-use programming tools. Your feedback is important to me, so please don't hesitate to email your comments or suggestions to [ray@MelissaData.com](mailto:ray@MelissaData.com).

I look forward to hearing from you.

Best Wishes,

A handwritten signature in black ink, appearing to read "Ray Melissa". The signature is fluid and cursive, with a long horizontal stroke at the end.

Raymond F. Melissa  
President

# Address Validator for Oracle Warehouse Builder

Address Validator for Oracle Warehouse Builder is a plug-in for OWB which powers the address cleansing functions, using the same address checking technology behind MAILERS+4.

## Installing Address Validator

To install Address Validator:

- 1 Close all applications that are open, including any anti-virus and e-mail software.
- 2 Place the Address Validator CD in your CD-ROM drive.
- 3 Click the **Start** button, and then select **Run** from the pop-up menu. The Run dialog box opens.
- 4 Type your CD drive letter followed by : \SETUP.EXE in the Command Line box. (for example, D:\SETUP.EXE). After typing the path, click OK.
- 5 Read the license agreement, and then click **Yes** if you agree to the terms. (To proceed with the Address Validator installation, you must accept this agreement.)

## Uninstalling

To uninstall Address Validator:

- 1 Open the Windows Control Panel.
- 2 Double-click **Add/Remove Programs**.
- 3 Select Address Validator for Oracle Warehouse Builder.
- 4 Click **Change/Remove**.

## Adding the License String

By default, Address Validator runs in Demo Mode, which will verify and standardize only Nevada addresses. Before you can use a registered copy of Address Validator, you must add the license string to the MDNameAddr.properties file, located in the Address Validator installation directory.

Open this file using a text editor and add the following line.

```
addr_license = XXXXXXXXXX
```

Replace the X's with the license string issued when you purchased Address Validator from Melissa Data.

## Stopping and Starting Address Validator

To stop the Address Validator:

- 1 Open a DOS command prompt.
- 2 Navigate to the `owb/bin/win32` sub-directory of the OWB Server installation directory.
- 3 Type `NASTOP` and press ENTER.

To re-start Address Validator:

- 1 Open a DOS command prompt.
- 2 Navigate to the `owb/bin/win32` sub-directory of the OWB Server installation directory.
- 3 Type `NASTART` and press ENTER.

## Filtering Out Non-U.S. Addresses

Address Validator currently works on addresses located within the United States and its possessions. Submitting non-U.S. address can have unpredictable effects on your data.

Therefore, you should insert a filter between your input data and the NameAddr operator that only allows U.S. addresses to pass through to Address Validator. You can filter on the `IN_COUNTRY_CODE` input role, only passing through records where this field equals "US" or "USA". If your database uses a different field or values to indicate country, use that value instead.

## Options

Options control certain aspects of how Address Validator handles each address.

To open Address Validator options:

- 1 Right-click on the title bar of the NameAddress operator.
- 2 Select **Operator Properties** from the pop-up menu.

### Parsing Type

Parsing Type controls which part of the input data is processed. Using this option can save time by excluding unnecessary input roles.

**Name:** Address Validator processes only those input roles that pertain to a person's name.

**Address:** Address Validator processes on those input roles that pertain to the delivery address.

**Name and Address (Default):** Address Validator processes all submitted input roles.

## Primary Country

This option selects the country to use to determine the rules for processing address information. The current version of Address Validator processes only U.S. addresses, therefore selecting a different country will have no effect.

## Dual Address Assignment

This option has no effect in the current version of Address Validator.

# Input Roles

Address Validator processes data from the following input roles.

## IN\_FIRM\_NAME

The name of the company associated with the input address.

If a company name is supplied for a company that has been assigned a specific Plus4 by the USPS, the address checking logic will return a more accurate Plus4 code.

If a company name is not supplied, the address checking logic will still be able to code the address but it will supply a more generic "street level default" Plus4.

## IN\_PERSON\_NAME

Contains the full name of the person associated with the address data being cleansed.

If this role contains a null value, then Address Validator examines the next two input roles:

## IN\_NAME\_FIRST\_PART

Contains the first and middle names of the person associated with the address data being cleansed.

## IN\_NAME\_LAST\_PART

Contains the last name of the person associated with the address data being cleansed.

If the these two input roles both contain null values, Address Validator examines the following seven input roles:

## IN\_PRE\_NAME

Contains the prefix to the name of the person associated with the address data being cleansed, such as "Dr." or "Ms."

## IN\_FIRST\_NAME

Contains the first or given name of the person associated with the address data being cleansed.

**IN\_MIDDLE\_NAME1**

Contains the first middle name of the person associated with the address data being cleansed, if any.

**IN\_MIDDLE\_NAME2**

Contains the second middle name of the person associated with the address data being cleansed, if any.

**IN\_MIDDLE\_NAME3**

Contains the third middle name of the person associated with the address data being cleansed, if any.

**IN\_LAST\_NAME**

Contains the last name of the person associated with the address data being cleansed.

**IN\_POST\_NAME**

Contains the suffix to the name of the person associated with the address data being cleansed, such as "Jr." or "III".

**IN\_ADDRESS or IN\_PRIMARY\_ADDRESS**

The delivery address portion of the address to be cleansed.

**IN\_ADDRESS2 or IN\_SECONDARY\_ADDRESS**

The second part of the delivery address portion of the address to be cleansed or an alternate address, such as a P.O. Box.

**IN\_CITY**

The city or municipality associated with the address to be cleansed. City and state are optional if a valid ZIP Code is provided.

**IN\_STATE**

The state or province associated with the address to be cleansed. City and State are optional if a valid ZIP Code is provided.

**IN\_COUNTRY\_CODE**

The 2-character abbreviation for the country an address is associated with. The current version of Address Validator is only for U.S. addresses, so this should always be "U.S."

**IN\_COUNTRY\_NAME**

The name of the country an address is associated with. Using IN\_COUNTRY\_CODE is preferred.

## IN\_NEIGHBORHOOD

The urbanization name associated with the input address (applies to Puerto Rican addresses only).

## IN\_POSTAL\_CODE

The five or nine digit ZIP Code or the six character Postal code, associated with the address to be cleansed. This field is optional if a valid City and State combination are input.

## IN\_LAST\_LINE

A string value containing the city, state and ZIP Code of the address to be cleansed. Can be used if those values are not available as individual fields.

## IN\_LINE1 through I\_LINE10

If the address data is not broken into the discrete fields listed above, an address can be input as up to ten lines of text.

**Note:** If *any* text is entered in these fields, the other input roles will be ignored.

## IN\_PERSON\_NAME2, IN\_PERSON\_NAME3

Address Validator does not handle multiple names per address record, so these input roles are ignored.

# Return Components

After cleansing the data, Address Validation makes available the following components.

Components that return a true or false value will return a string value of "T" if true, "F" otherwise.

## FIRM\_NAME

Passes through the IN\_FIRM\_NAME input role.

## PERSON

The full name of the person associated with the returned address.

## PRE\_NAME

Contains the prefix to the name of the person associated with the returned address record, such as "Dr." or "Ms.".

## FIRST\_NAME

Contains the first or given name of the person associated with the returned address record.



## **FIRST\_NAME\_STD**

Contains a standardized version of the first name of the person associated with the returned record. For example, "Bill" will return "William".

## **MIDDLE\_NAME**

Contains the middle name of the person associated with the returned record, if any.

## **MIDDLE\_NAME\_STD**

Contains a standardized version of the middle name of the person associated with the returned record. For example, "Bob" will return "Robert".

## **LAST\_NAME**

Contains the last name of the person associated with the returned record.

## **GENERATION**

Contains the suffix to the name of the person associated with the returned record, such as "Jr." or "III".

## **GENDER**

Contains a one letter-code for the gender information derived from the returned name record.

Possible return values from the Gender Property:

'F' = Female

'M' = Male

'N' = Neutral (Male or Female)

'U' = Unknown

## **NAME\_DESIG**

Address Validator will populate this component with a zero-length string.

## **MIDDLE\_NAME2, MIDDLE\_NAME3**

### **MIDDLE\_NAME2\_STD, MIDDLE\_NAME3\_STD**

Address Validator does not parse or standardize multiple middle names. Multiple middle names are combined in the MIDDLE\_NAME component.

## **OTHER\_POST\_NAME**

Address Validator will populate this component with a zero-length string. All name suffixes are returned to the GENERATION component.

## **RELATIONSHIP**

Address Validator will populate this component with a zero-length string.

## **FIRM\_CNT**

If a firm name is passed by an input role, this component will return a value of 1, otherwise it will be 0. Address Validator does not detect multiple firm names in an address record.

## **PERSON\_CNT**

If a person's name is passed by an input role, this component will return a value of 1, otherwise it will be 0. Address Validator does not detect multiple person names in an address record.

## **ADDRESS**

Address will contain the STREET address along with any corrections or standardizations performed by the address checking logic, including any suite or private mail box information.

Address corrections may include fixing misspelled street names or inserting missing suffixes and directionals. Address standardization involves the conversion of suffixes and directionals to preferred postal abbreviations (for example, "Street" to "St").

## **PRIME\_ADDR**

Contains the street address portion of the returned delivery address.

## **PRIM\_RANGE**

Contains the delivery number from the returned address record.

## **PRE\_DIR**

Contains any directional that precedes the street name in the returned delivery address. Spelled-out directions will be return as one-letter abbreviations. For example: "North" would be returned as "N."

## **PRIM\_NAME**

Contains the name of the street returned with the delivery address.

For street names that begin with one of the following Spanish words, that first word will be moved to the suffix field and the next word in the street name will be returned as the street name. This is because spanish street names have the suffix at the front of the address. If the suffix was not moved to the suffix field, it would appear that every street in some Puerto Rican ZIP codes began with the same letter. The spanish words that get moved to the suffix field are: "Avenida", "Calle", "Camino", "Paseo" and "Via"

## **STREET\_TYPE**

Contains the suffix portion of returned delivery address.

When a street names are used more than once within a city or area, street suffixes are often used to distinguish them. For example, if there is a Main Street but also a Main Avenue, this field will hold either the "St" or "Ave" suffix, depending on which one is being referred to.

If the word "Street" is found, it will be converted to "St" and stored in this property.

## **POST\_DIR**

Contains any directional that follows the street name in the returned delivery address. Spelled-out directions will be return as one-letter abbreviations. For example: "North" would be returned as "N."

## **SEC\_ADDR**

Contains any suite or private mail box information associated with the returned address record.

## **UNIT\_DESIG**

Contains the name of the secondary unit of a returned address.

Possible return values are: "#", "APT", "BLDG", "BOX", "BSMT", "DEPT", "FL", "FRNT", "HNGR", "LBBY", "LOT", "LOWR", "OFC", "PH" (Penthouse), "PIER", "REAR", "RM", "SIDE", "SLIP", "SPC", "STE", "STOP", "TRLR", "UNIT", "UPPR".

## **UNIT\_NBR**

Contains the range of any secondary unit of an address, such as the suite or private mailbox.

## **BLDG\_NAME**

Address Validator will populate this component with a zero-length string.

## **BOX\_NAME**

Address Validator will populate this component with a zero-length string.

## **BOX\_NBR**

Address Validator will populate this component with a zero-length string.

## **RTE\_NAME**

Address Validator will populate this component with a zero-length string.

## **RTE\_NBR3**

Address Validator will populate this component with a zero-length string.

## **CITY**

Contains the name of the city returned by the address checking logic.

## **COUNTY\_NAME**

Countains the name of the county associated with the returned address record.

## FIPS

Contains the five-digit FIPS code for the county associated with the returned address record.

The Federal Information Processing Standard (FIPS) is a 5-digit code defined by the U.S. Bureau of Census. The first two digits are the state code and the last three indicate the county within the state.

For Example: "06037" is the County FIPS for Los Angeles, CA ("06" is the state code for California and "037" is the county code for Los Angeles).

The CountyFips Property is accurate to the 9-digit level.

## FIPS\_COUNTY

Contains the three-digit county portion of the FIPS component.

## STATE

Contains the two-character abbreviation for the state or province associated with the returned address record.

## LAST\_LINE

Contains a string value with the City, State or Province and the ZIP or Postal Code associated with the returned address record.

## LACS

This indicator notifies you if the input address has undergone a LACS conversion.

Locatable Address Conversion Service (LACS) is a process where some rural route addresses are modified to city-style addresses to allow emergency services (for example, ambulance, police, fire, and so on) to find these addresses more efficiently.

There are two possible values for this property. An empty space indicates that the address has not undergone a LACS conversion. A value of "L" in the LACS field indicates that the address has undergone a conversion. After a conversion, the old address is retained in the ZIP+4 file for a period of one year. After the one year period, the old addresses will be dropped from the ZIP+4 file and the address checking logic will not assign a +4 for this address.

The new addresses are not contained within the ZIP+4 file. To change the old addresses to the new addresses, you will need to send these addresses to a company that does LACS processing. Melissa Data or your local post office will be glad to assist you with this process.

## URB\_NAME

The urbanization name associated with the returned address record (Puerto Rican addresses only).

## NEIGHBORHOOD

Passes through the contents of the IN\_NEIGHBORHOOD input role.

## **POSTAL**

Contains the full Postal Code or nine-digit ZIP+4 Code. The ZIP+4 code is returned without a dash after the fifth digit.

## **POSTAL\_FMT**

Contains the full Postal Code or nine-digit ZIP+4 Code. The ZIP+4 code is returned with the dash after the fifth digit.

## **ZIP5**

The five-digit ZIP Code associated with the returned address record.

## **ZIP4**

The four-digit suffix of the full ZIP+4 Code associated with the returned address record.

## **CTRY**

Contains the two-digit abbreviation for the country associated with the returned address record, such as "US" for the United States.

## **CTRY3**

Contains the three-digit abbreviation for the country associated with the returned address record, such as "USA" for the United States.

## **PARSING\_COUNTRY**

This component returns the three-letter country code for the country used to determine the rules for validating the input data. In the current version of Address Validator, this will always be "USA".

## **CTRY\_NAME**

Returns the unabbreviated name for the country passed by IN\_COUNTRY\_CODE. Currently, the Address Validator works with U.S. addresses only, so this will always be "United States".

## **LATITUDE**

Contains the latitude of the geographic center of the five-digit ZIP code associated with the returned address record.

## **LONGITUDE**

Contains the longitude of the geographic center of the five-digit ZIP code associated with the returned address record.

## **MSA**

Contains the four-digit MSA number associated with the returned address record.

The Office of Management and Budget defines the Metropolitan Statistical Area (MSA) as one or more counties forming a large population with adjacent communities and having a high degree of social and economic integration.

## CART

Contains the carrier route associated with the returned address data.

The first character of this property is always alphabetic, and the last three characters are numeric. For example, "R001" or "C027" would be typical carrier routes. The alphabetic letter indicates the type of delivery associated with this address.

B = PO Box  
C = City Delivery  
G = General Delivery  
H = Highway Contract  
R = Rural Route

## DPBC

Returns the 10th and 11th positions of the 12-digit POSTNet barcode.

In 12-digit POSTNet barcodes, the ZIP Code is used for positions 1 to 5, the Plus4 code for positions 6 to 9, the delivery point code for positions 10 and 11, and this check digit for position 12.

## CHK\_DIG

This 1-digit string makes up the 12th position of a 12-digit POSTNet barcode.

## MISC\_ADDR

This component returns any miscellaneous data from the input that did not correspond to a known address component. If there are none, this would return the contents of the first unused IN\_LINE input role, if any.

## ADDR\_TYPE

Contains a 1 or 2-character string value which indicates the type of address that was returned:

Code	Type
F	Firm or company address
G	General Delivery address
H	Highrise or business complex
P	PO Box address
R	Rural route address
S	Street or residential address
FD	Firm Default
HD	High-Rise Default
RS	Rural Route Default
SD	Street Default

Code	Type
UD	Unique Default

## IS\_PARSED

The Address Validator successfully parsed the name and/or address information from the input roles above. This does not mean that the name or address were found in the data, only that they were correctly formatted and recognized as a name or address.

## IS\_FOUND

A true value the name and address were successfully matched to the Address Validator's database. This does not automatically mean that the address is a real delivery address, but that the address falls with a valid range for the street and ZIP Code.

## IS\_GOOD\_ADDR

When verifying address information, a true value means that a U.S. address was successfully verified, meaning that it was parsed and found in the Postal database. For non-U.S. addresses, true indicates that the address was successfully parsed.

## IS\_GOOD\_NAME

A true value means that the value of LAST\_NAME was successfully matched with the input address, if name information is being verified.

## IS\_GOOD\_GRP

A true value means that both IS\_GOOD\_NAME and IS\_GOOD\_GROUP returned a value of "T". If either one did not, this field returns a false.

## IS\_ADDRESS\_VERIF

A true value means that the country specified in the IN\_COUNTRY\_CODE or IN\_COUNTRY\_NAME input roles is one covered by the Address Validator's database. Presently, that encompasses U.S. addresses only. If no country is specified in either of those input roles, the U.S. is automatically assumed.

## NAME\_WARNING

A true value indicates that the Address Validator could not find a parsable person's name in any input role.

## CITY\_WARNING

A true value indicates that the ZIP/Postal Code was not and could not be determined by the city/municipality and state/province.

## STREET\_WARNING

A true value indicates that the street portion of the input address could not be found in the database in the specified city and zip code.

## PARSE\_STAT

This component returns a one-character code indicating the level of verification that the Address Validator achieved with the input address.

Code	Detail
E	Expired Database
S	The address was standardized but not coded. Standardization means that some conversion was done on the address (for example, changing Post Office Box to PO Box or abbreviating street suffixes).
X	Address was not coded.
7	There were multiple matches for the address but they were all in the same ZIP Code and carrier route. The returned ZIP Code and carrier route will be correct but you will not get any +4 information.
9	The address was fully coded.

If "S" or "X" is returned, check the contents of the PARSE\_STAT\_DESC for the reason why the address was not coded.

## PARSE\_STAT\_DESC

Contains a short description of the reason why an address was not successfully coded.

Text	Detail
Multiple Matches	More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing information, such as city/municipality names or urbanization names, can help reduce the number of multiple match errors.
No Street Data for ZIP/Postal Code	The ZIP/Postal Code exists but no streets begin with the same letter in that ZIP/Postal Code.
Address Out of Range	The address was found but the street number in the input address was not between the low and high range of the post office database.
Component Mismatch	Either the directionals or the suffix field did not match the post office database, and there was more than one choice for correcting the address. For example, if the given address was "100 Main St" and the only addresses found were "100 E Main St" and "100 Main Ave", this value would be returned to PARSE_STAT_DESC because we do not know whether to add the directional "E" or to change the suffix to "Ave."
Unknown Street	An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.



Text	Detail
Early Warning System	This address has been identified in the Early Warning System (EWS) data file, and should be included in the next national database update. The EWS data is not included with the Address Validator but it can be downloaded from this address: <code>ftp://www.melissadata.com/Updates/ews.txt</code> . The EWS data file should be saved to the same directory as the Address Validator's data files.
Non-Deliverable Address	The physical location exists but there are no homes on this street. One reason might be railroad tracks or rivers running alongside this street, as they would prevent construction of homes in this location.
ZIP/Postal Code Error	The ZIP/Postal Code does not exist and could not be determined by the city/municipality and state/province.

## ADDR\_CORRECTED

A true value indicates that the information in one or more elements of the input address was changed. This includes only changes to the actual content, such as a spelling correction, but not standardization, such as changing "North" to "N." in the directional.

## POST\_CORRECTED

A true value indicates that the ZIP Code information has been changed. This could include adding or correcting a ZIP Code based on city and state or adding a Plus 4 extension to a five-digit zip code.

## CITY\_CORRECTED

A true value indicates that the city information has been changed, including adding a city and state based on ZIP, if the IN\_CITY input role was empty.

## STREET\_CORRECTED

A true value indicates that the street name or range was changed.

## STREET\_COMP\_CORRECTED

A true value indicates that one of the street name components, either the street type or a directional has changed. This does not include standardization (changing "Street" to "ST" or "North" to "N") but changes to the actual information (changing "ST" to "RD" or "N" to "S").

## CITY\_MATCH

A true value indicates that the input address is a valid street address within the specified city. The ZIP Code may not be correct but the address does exist within the city.

## STREET\_NAME\_MATCH

A false value indicates that an exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.

## **STREET\_NBR\_MATCH**

A false value indicates that the address was found but the street number in the input address was not between the low and high range of the post office database.

## **STREET\_COMP\_MATCH**

A false value indicates that the address was found, but not match the directionals or street type provided. In other words, if "1234 N. Main St." was entered and either "1234 S. Main St." or "1234 N.Main Pl." were found, this component would return false.

## **NON\_AMBIGUOUS\_MATCH**

A true value indicates that one and only one record was found matching the input address. A false value indicate that more than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing information, such as city/municipality names or urbanization names, can help reduce the number of multiple match errors.

## **MCD**

Address Validator will populate this component with a zero-length string.

## **CENSUS\_ID**

Address Validator will populate this component with a zero-length string.

## **LOCALITY\_CODE**

Address Validator will populate this component with a zero-length string.

## **LOCALITY\_NAME**

Address Validator will populate this component with a zero-length string.

## **INSTALLATION\_CODE**

Address Validator will populate this component with a zero-length string.

## **INSTALLATION\_NAME**

Address Validator will populate this component with a zero-length string.

## **DELIVERY\_OFFICE**

Address Validator will populate this component with a zero-length string.

## **BEAT\_NUMBER**

Address Validator will populate this component with a zero-length string.

## **LOT**

Address Validator will populate this component with a zero-length string.

## **LOT\_ORDER**

Address Validator will populate this component with a zero-length string.