Reference Guide

Data Quality Components
for Pentaho Data Integration

Contact Verification Component

MELISSA DATA®

Data Quality Components for Pentaho

Contact Verification Component (CVC)

Reference Guide

Copyright

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

MELISSA DATA CORPORATION
22382 Avenida Empresa
Rancho Santa Margarita, CA 92688-2112
Phone: 1-800-MELISSA (1-800-635-4772)
Fax: 949-589-5211
E-mail: info@MelissaData.com
Web site: www.MelissaData.com

For the latest version of this Reference Guide, visit
http://www.MelissaData.com/tech/tech.htm.

Document Code: CVPRFG
Revision Number: 110603.001
Last Update: June 3, 2011

Introduction

The Contact Verification Component makes it easy to integrate Melissa Data's suite of Data Quality tools into an existing Pentaho installation. It provides a simple interface which enables you to easily select the fields from your input data and direct them to the correct fields in your output data.

The Component gives you the ability to correct, parse and standardize addresses, parse out full names and detect vulgarities, verify phone numbers all the way down to at the 10-digit level, verify email domains and correct email syntax errors, and append census and geographic information such as the latitude and longitude. The Component makes it easy to retrieve data from your input database, setup the component to perform only the validation you want, and output enrichment data that you need.

Name Parsing

The Component can take a single name field, containing up to two individual names, and break both of the names into their parts:

- Prefix ("Mr.," "Ms.," or "Dr.")
- First name
- Middle name or initial
- Last name
- Suffix ("Jr.," "III," or "PhD")

The Component can also, in many cases, assign gender information based on the first name.

In addition, the Component can also correct misspellings of common first names, correct syntax errors and also catch vulgar words and obviously false names.

Address Verification and GeoCoding

The Contact Verification Component can verify that an address is a properly formatted address that matches a true and valid point of delivery using the DPV® database. It also matches the address to the LACS$^{Link®}$ file to determine if the address has been converted (usually from a rural route to a standard city-street address) and automatically updates the address.

The CASS Certified™ address checking logic includes the DPV, LACS^{Link}, and Suite^{Link}™ processing mandatory to qualify your mailings for postal discounts by appending the + 4 codes only to validated addresses. The Component can also generate the CASS™ Form 3553 required by the Postal Service™ to claim the discounts.

The Component also enhances your data by providing additional information:

- Congressional district

- County name
- County FIPS code
- Time zone
- Longitude
- Latitude
- Census information (Census block, Place Codes, and CBSA data).

If you have the optional GeoPoint feature installed, the latitude and longitude can be accurate to the rooftop level in many cases.

Email and Phone Standardizing

The Contact Verification Component can take email addresses and phone numbers, standardize, parse and verify them.

With email addresses, the Component can:

- Parse the address into mailbox, domain and top level domain.
- Correct minor syntax errors.
- Fix misspelled domain names.
- Update domain names that have changed.
- Verify domain names against our own database.
- Verify domain names via DNS lookup (slower but more complete and up-to-date than any database can be).

With phone numbers, the Component can:

- Parse the phone number into area code, prefix, suffix and extension.
- Verify phone numbers to the 10-digit or 7-digit level.
- Flag numbers as landline, wireless or VOIP.
- Update area codes that have split.
- Detect phone numbers as business, residential or SOHO (Small Office/Home Office)

Output Filters

After processing, the Contact Verification Component will direct records to either a valid table or an invalid table. The valid table contains records that match the output filters while the invalid table contains all other records.

The Component ships with several pre-built rules (filters) that should cover many common situations. You can build custom rules using the Result codes returned by the Component to tailor the product to your exact needs.

Reference

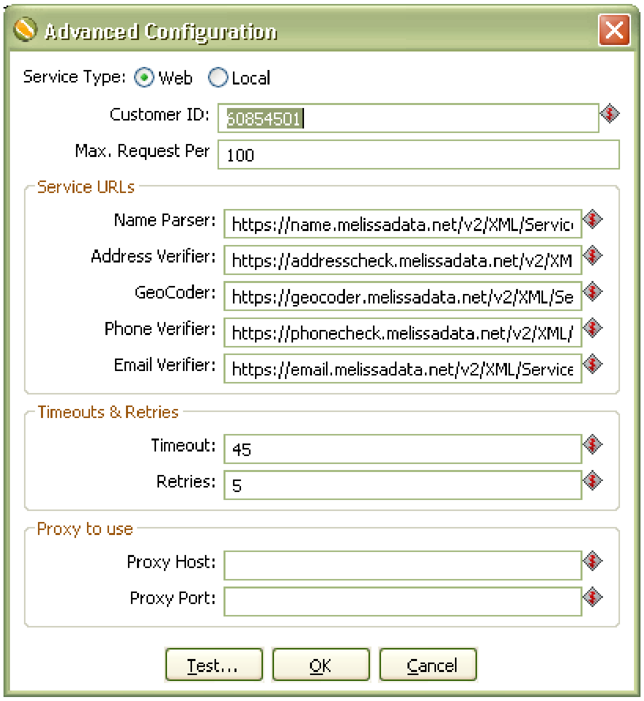
Advanced Configuration

The advanced configuration dialog is where you enter the basic information required for the Contact Verification Component to connect to the underlying Melissa Data libraries.

The Contact Verification Component allows you to select between 2 service types: Web and Local. Selecting the Web radio button will send your records through the Melissa Data Web Services for verification. Selecting the Local radio button will send your records though the Melissa Data compiled libraries which reside locally on that server.

Web Solutions

These settings apply when using the component with the web services.



Service Type: ● Web ○ Local

Customer ID: 60854501

Max. Request Per 100

Service URLs

Name Parser: https://name.melissadata.net/v2/XML/Servic

Address Verifier: https://addresscheck.melissadata.net/v2/XM

GeoCoder: https://geocoder.melissadata.net/v2/XML/Se

Phone Verifier: https://phonecheck.melissadata.net/v2/XML/

Email Verifier: https://email.melissadata.net/v2/XML/Service

Timeouts & Retries

Timeout: 45

Retries: 5

Proxy to use

Proxy Host:

Proxy Port:

Customer ID

Enter the customer ID provided by Melissa Data when you purchased the Component. Without the customer ID, the Component will not be functional.

Max Requests Per

Enter the number of records to be sent through the Web Service for each request (Maximum value of 100). Increasing this number will speed up the throughput for batch processing.

Name Parser URL

Enter the URL of the Name Web Service. Default value is https://name.melissadata.net/v2/XML/Service.svc/doNameCheck

Address Verifier URL

Enter the URL of the Address Web Service. Default value is https://addresscheck.melissadata.net/v2/XML/Service.svc/doAddressCheck

GeoCoder URL

Enter the URL of the GeoCoder Web Service. Default value is https://geocoder.melissadata.net/v2/XML/Service.svc/doGeoCode

Phone Verifier URL

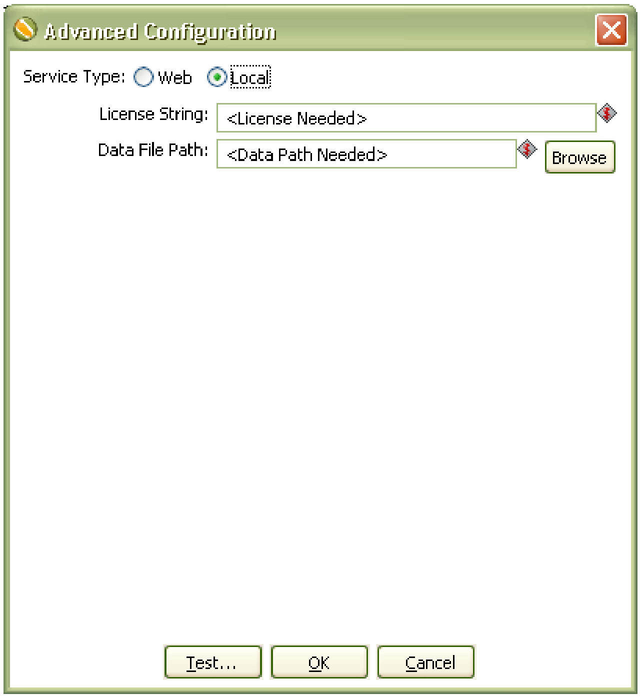
Enter the URL of the Phone Web Service. Default value is https://phonecheck.melissadata.net/v2/XML/Service.svc/doPhoneCheck

Email Verifier URL

Enter the URL of the Email Web Service. Default value is https://email.melissadata.net/v2/XML/Service.svc/doEmailCheck

Local Solutions

Use the local settings when using the Component with the Melissa Data objects instead of the web services.



Service Type: ○ Web ● Local

License String: <License Needed>

Data File Path: <Data Path Needed>

License String

Enter the license string provided by Melissa Data when you purchased the Component. Without the license string, the Component will not be fully functional.

Data File Path

Enter the file path of the folder containing the data files used by the Melissa Data libraries. Alternately, click the folder button and browse to the folder.

Test Configuration

The test configuration screen returns basic information about the object libraries being used by the Contact Verification Component. Used for local installation only.

All Objects

License Expiration

Displays the expiration date of the current license string. If this date has passed, the current license string will not work.

Name Object

Initialization

Displays any error messages generated by initializing the Name Object.

Build Number

Displays the current development release build number of the Name Object. This is usually a three or four-character string.

Database Date

Displays the date of the Name Object data files. This date confirms that the data files are the latest available.

Database Expiration Date

Displays the date when the Name Object data files will expire. This date confirms that the data files are the latest available.

Address Object

Initialization

Displays any error messages generated by initializing the Address Object.

Build Number

Displays the current development release build number of the Address Object. This is usually a three or four-character string.

Database Date

Displays the date of the U.S. address data files. This date confirms that the data files are the latest available.

Database Expiration Date

Displays the date when the U.S. data files will expire. This date confirms that the data files are the latest available.

Canadian Database Date

Displays the publication date of the Canadian data files. This date confirms that the data files are the latest available.

Canadian Database Expiration Date

Displays the date when the Canadian data files expire. This date confirms that the data files in use are the latest available.

RBDI Database Date

Displays the date of the RBDI data files. This date confirms that the data files are the latest available.

GeoCoder Object

Initialization

Displays any error messages generated by initializing the GeoCoder Object.

Build Number

Displays the current development release build number of the GeoCoder Object. This is usually a three or four-character string.

Database Date

Displays the date of the GeoCoder Object data files. This date confirms that the data files are the latest available.

Database Expiration Date

Displays the date when the GeoCoder Object data files will expire. This date confirms that the data files are the latest available.

Phone Object

Initialization

Displays any error messages generated by initializing the Phone Object.

Build Number

Displays the current development release build number of the Phone Object. This is usually a three or four-character string.

Database Date

Displays the date of the Phone Object data files. This date confirms that the data files are the latest available.

Email Object

Initialization

Displays any error messages generated by initializing the Email Object.

Build Number

Displays the current development release build number of the Email Object. This is usually a three or four-character string.

Database Date

Displays the date of the Email Object data files. This date confirms that the data files are the latest available.

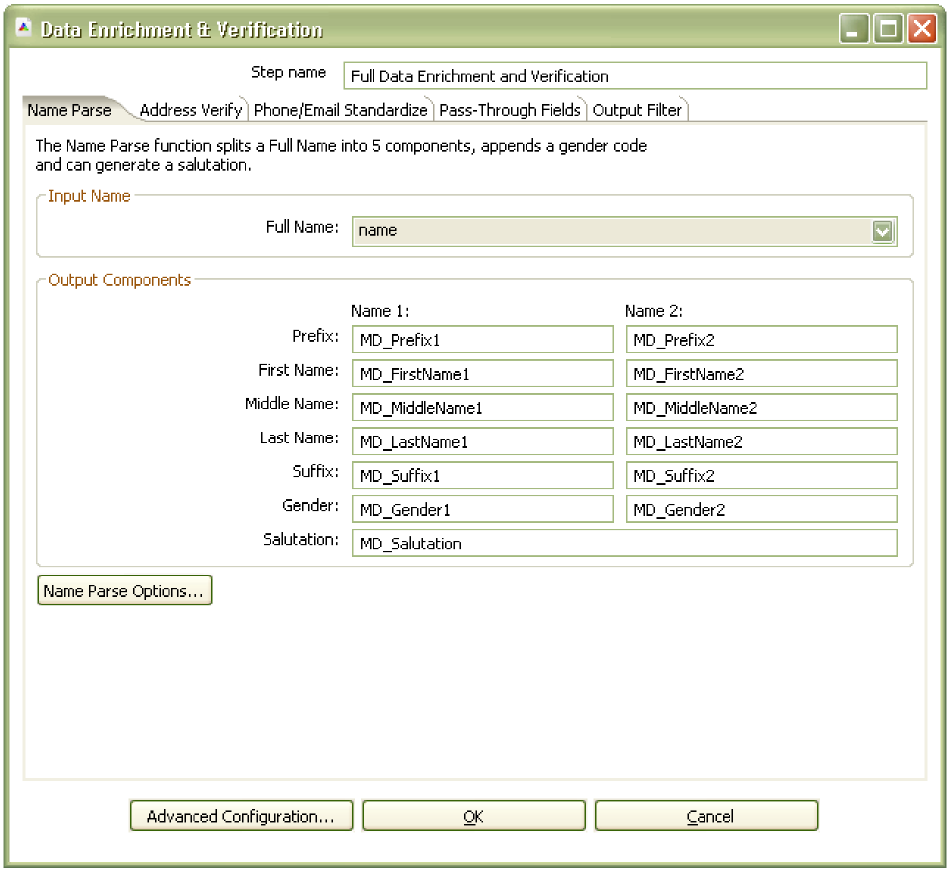
Database Expiration Date

Displays the date when the Email Object data files will expire. This date confirms that the data files are the latest available.

Name Parse Tab

The Name Parse tab configures the fields that will be used for the name parsing functionality of the Contact Verification Component.

Existing field names can be selected using the dropdown boxes. New field names can be created by typing the name into the box.



Step name: Full Data Enrichment and Verification

The Name Parse function splits a Full Name into 5 components, appends a gender code and can generate a salutation.

Input Name

Full Name: name

Output Components

	Name 1:	Name 2:
Prefix:	MD_Prefix1	MD_Prefix2
First Name:	MD_FirstName1	MD_FirstName2
Middle Name:	MD_MiddleName1	MD_MiddleName2
Last Name:	MD_LastName1	MD_LastName2
Suffix:	MD_Suffix1	MD_Suffix2
Gender:	MD_Gender1	MD_Gender2
Salutation:	MD_Salutation	

Input

The Name Parse tab requires data from a single field in order to populate its output fields.

Full Name

Select or enter the field name that will contain the name information to be parsed. This field can contain one or two full names, such as “Mr. John Q. Smith, Jr.” or “John Q. and Mary S. Smith.”

If you do not set a field for Full Name, the Component will not parse name information.

Output

The output section consists of two sets of return fields. The first name detected will be returned by the first set of fields. If a second name is detected, it will be returned by the second set of fields.

Prefix

This field returns any part of the name that precedes the given name, such as “Mr.,” “Ms.” or “Dr.,” for each full name detected.

First Name

This field returns the given name for each full name detected.

DEVC can attempt to correct misspelled first names. To enable this feature, see the Name Parse Options screen.

Middle Name

This field returns the middle names or initials for each full name detected.

Last Name

This field returns the family name for each full name detected.

Suffix

This field returns any part of the name that follows the family name, such as degrees (“MD” or “PhD”) and generational indicators (“IV” or “Jr.”), for each full name detected.

Gender

This field returns a gender indicator for each full name detected.

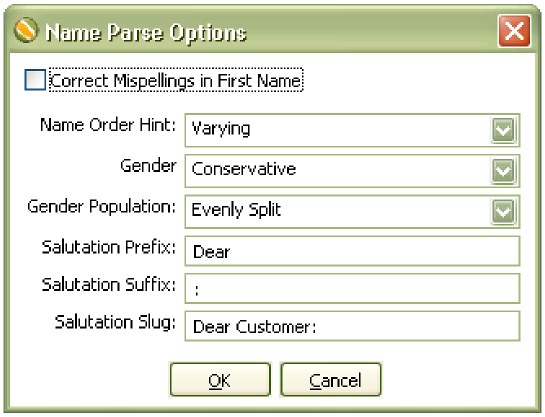
Gender is based on the first name. See the Name Parse Options screen for how to adjust how the Component assigns gender to a name.

Salutation

This field returns a salutation constructed from the first full name detected. To control how this salutation is formatted, see the Name Parse Options screen.

Name Parse Options

To access this screen, click the Name Parse Options button on the Name Parse tab.



☐ Correct Mispellings in First Name

Name Order Hint: Varying

Gender: Conservative

Gender Population: Evenly Split

Salutation Prefix: Dear

Salutation Suffix: :

Salutation Slug: Dear Customer:

Correct Misspellings in First Name

The Component uses a database of common given names to correct obvious misspellings. To enable this feature, check the box.

Name Order Hint

The Name Order Hint tells the Component in what order the name components will be found in the input full name, normal name order, last name first, or a mixture.

The default is “Varying.” The options are:

Option	Description
DefinitelyFull	Name will always be treated as normal name order, regardless of formatting or punctuation.
VeryLikelyFull	Name will be treated as normal name order unless inverse order is clearly indicated by formatting or punctuation.
ProbablyFull	If necessary, statistical logic will be employed to determine name order, with a bias toward normal name order.
Varying	If necessary, statistical logic will be employed to determine name order, with no bias toward either name order.
ProbablyInverse	If necessary, statistical logic will be employed to determine name order, with a bias toward inverse name order.
VeryLikelyInverse	Name will be treated as inverse name order unless normal order is clearly indicated by formatting or punctuation.

Option	Description
DefinitelyInverse	Name will always be treated as inverse name order, regardless of formatting or punctuation.
MixedFirstName	Name field is expected to only contain prefixes, first, and middle names.
MixedLastName	Name field is expected to only contain last names and suffixes.

Gender Aggression

This option controls how the Component assigns gender to a name, based on the first name.

First names are rated on a 1 to 7 scale on the likelihood that they are a male or female name, with 7 being “always male” and 1 being “always female.”

The Gender Aggression setting controls how the Component treats names that fall between those two extremes.

Gender Population

This options controls the gender assumed for the input data: predominantly female, predominantly male, or an even mixture.

The effect of the Gender Aggression and Gender Population settings is shown on this chart.

		Male				Female		
	Aggression	**Always (7)**	**Often (6)**	**Normally (5)**	**Neutral (4)**	**Normally (3)**	**Often (2)**	**Always (1)**
	Conservative							
Bias	Neutral	M	N	N	N	N	N	F
	Male	M	M	N	N	N	N	F
	Female	M	N	N	N	N	F	F
	Neutral							
Bias	Neutral	M	M	N	N	N	F	F
	Male	M	M	M	N	N	F	F
	Female	M	M	N	N	F	F	F
	Aggressive							
Bias	Neutral	M	M	M	N	F	F	F
	Male	M	M	M	M	N	F	F
	Female	M	M	N	F	F	F	F

Salutation Prefix

The Component will begin every salutation with the text entered in this box. The default setting is “Dear.”

Salutation Suffix

The Component will end every salutation with the text entered in this box. The default setting is a semicolon.

Salutation Slug

The Component will use this text for a salutation if the input data did not contain enough information to construct a salutation from the parsed data. The default setting is “Dear Customer;.”

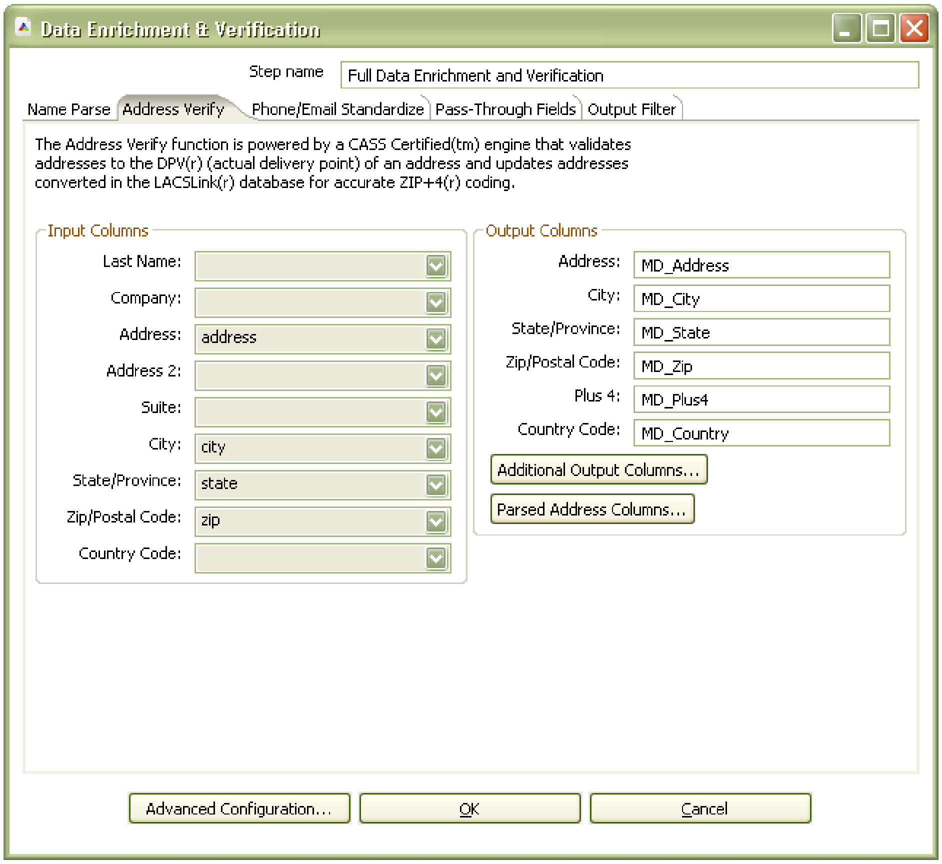
Salutation Preference

Use this box to change the order of preference for salutation formats. The highest format will be used if possible, followed by the second, until all possibilities are exhausted. If you do not wish a format to be used, place it below the selection for “Blank.”

To change the order of preference, select the items on the list and click the arrow buttons to move the selection up and down the list.

Address Verify Tab

The Address Verify Tab configures the fields that will be used for data enrichment and validation for addresses. It can correct, standardize, parse and provide additional information such as the plus4, delivery point, congressional district, etc. The Address Verify Tab will also provide Geographical information (such as the Latidute and Longitude) when requested.


Data Enrichment & Verification
Name Parse
Address Verify
Phone/Email Standardize
Pass-Through Fields
Output Filter
Additional Output Columns...
Parsed Address Columns...
Advanced Configuration...
OK
Cancel

Step name: Full Data Enrichment and Verification

The Address Verify function is powered by a CASS Certified(tm) engine that validates addresses to the DPV(r) (actual delivery point) of an address and updates addresses converted in the LACSLink(r) database for accurate ZIP+4(r) coding.

Input Columns

Last Name:	
Company:	
Address:	address
Address 2:	
Suite:	
City:	city
State/Province:	state
Zip/Postal Code:	zip
Country Code:	

Output Columns

Address:	MD_Address
City:	MD_City
State/Province:	MD_State
Zip/Postal Code:	MD_Zip
Plus 4:	MD_Plus4
Country Code:	MD_Country

Input

Use this column to select the database fields that will contain the address data to be verified and standardized.

For address verification, you must map fields for street address, city, and state, or street address and ZIP Code™.

For geocoding, you must map fields for ZIP Code and the Plus 4. If you are not doing address verification, you must map these as input fields. Otherwise, they must be mapped as output fields for Address Verify.

Geopoint information requires that you also map a delivery point code in addition to ZIP Code and Plus 4. If you are not doing address verification, you must map these as input fields. Otherwise, they must be mapped as output fields for Address Verify.

Last Name

The last name is required for the AddressPlus feature. You have the option of using the contents from one of the Last Name output fields from the Name Parse tab or select an existing field in the input database.

Company Name

Select the field in the input database that contains the company name information.

Some companies have a assigned unique ZIP Code™ for certain facilities, so the Company Name field will allow the Component to assign the correct ZIP + 4® in these cases.

Address

Select the field in the input database that contains the first line of street address information. It may also contain secondary address information such as suite and private mailbox numbers.

This field must be mapped for address verification to occur.

Address2

Select the field in the input database that contains the second line of a street address information, if any.

For more information on how the Component handles the Address and Address2 fields, see **Address Handling** on page 56.

Suite

Select the field in the input database that contains the secondary address information, if any.

City

Select the field in the input database that contains the city information.

State

Select the field in the input database that contains the state information.

Zip/Postal Code

Select the field in the input database that contains the postal code information.

If no address, city or state fields are mapped, this field must be mapped for the GeoCoder output fields to be populated with five-digit ZIP Code accuracy.

Country Code

Select the field in the input database that contains the abbreviation for the country, United States or Canada, where the address is located.

Output

Use this column to map the field names that will receive the verified and standardized address information. You may select existing fields or create new fields by entering the field names in the boxes.

Address

Map the field that will receive the standardized street address that was used for verification. This may be the contents of the Address2 input field, if address swapping has occurred.

City

Map the field that will receive the city name. The information returned may be different than the contents of the City input field, if the preferred city name has been substituted for a vanity city name or if the address was coded to a different zip code instead.

State

Map the field that will receive the two-character state abbreviation.

Zip/Postal Code

Map the field that will receive the five-digit U.S. ZIP Code or the six-character Canadian Postal Code.

Plus 4

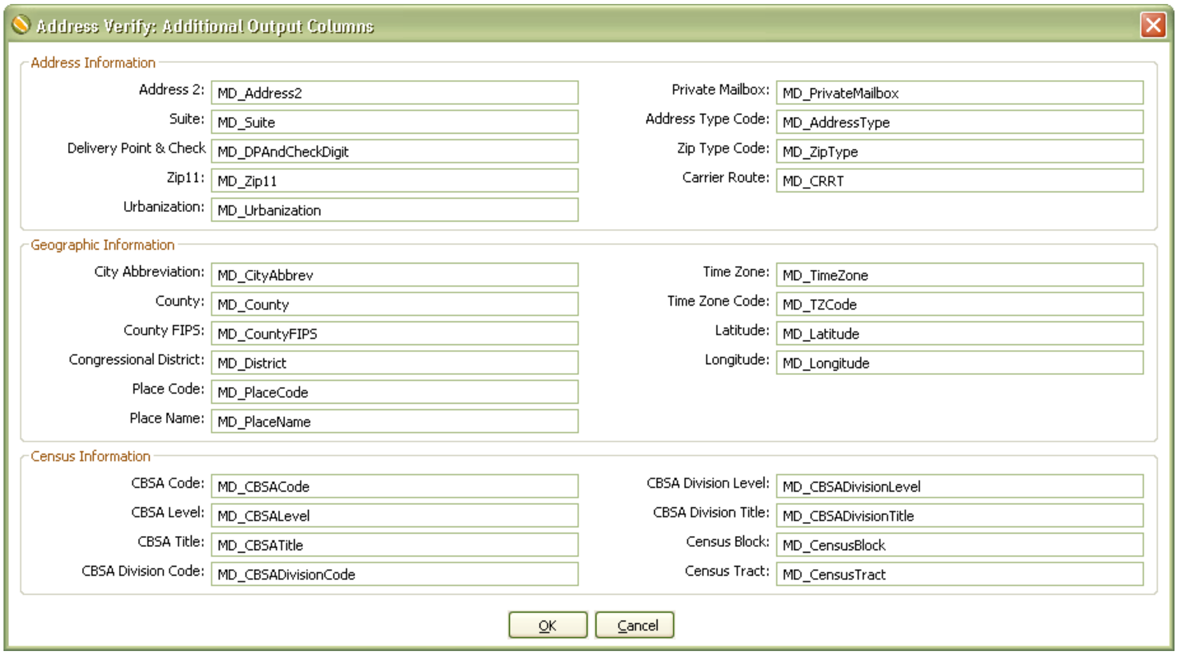
Map the field that will receive the four-digit extension from the ZIP + 4.

Country Code

Map the field that will receive the abbreviation for the country, United States or Canada, where the address is located.

Address Verify: Additional Information

The Additional Information allows you to map fields for returned data that goes beyond simply coding or validating the address.



Address Information

Address 2:	MD_Address2	Private Mailbox:	MD_PrivateMailbox
Suite:	MD_Suite	Address Type Code:	MD_AddressType
Delivery Point & Check	MD_DPAndCheckDigit	Zip Type Code:	MD_ZipType
Zip11:	MD_Zip11	Carrier Route:	MD_CRRT
Urbanization:	MD_Urbanization		

Geographic Information

City Abbreviation:	MD_CityAbbrev	Time Zone:	MD_TimeZone
County:	MD_County	Time Zone Code:	MD_TZCode
County FIPS:	MD_CountyFIPS	Latitude:	MD_Latitude
Congressional District:	MD_District	Longitude:	MD_Longitude
Place Code:	MD_PlaceCode		
Place Name:	MD_PlaceName		

Census Information

CBSA Code:	MD_CBSACode	CBSA Division Level:	MD_CBSADivisionLevel
CBSA Level:	MD_CBSALevel	CBSA Division Title:	MD_CBSADivisionTitle
CBSA Title:	MD_CBSATitle	Census Block:	MD_CensusBlock
CBSA Division Code:	MD_CBSADivisionCode	Census Tract:	MD_CensusTract

Address Information

These fields return additional information about the input data related to mail delivery to that location.

Address Type Code & Address Type String

These two fields return a one-character code indicating the type of address that was processed and a text description of the address type.

For U.S. addresses, the Component will return one of the following codes.

Code	String
F	Firm or Company address
G	General Delivery address
H	High Rise or Business complex
P	PO Box address
R	Rural Route address
S	Street or Residential address

For Canadian addresses, the Component will return one of the following codes.

Code	String
1	Street
2	Street Served By Route and GD
3	Lock Box
4	Route Service
5	General Delivery
A	Building
B	LVR Street
C	Government Street
D	LVR Lock Box
E	Government Lock Box
L	LVR General Delivery

ZIP Code Type

The type of ZIP Code can indicate a specialized address, such as military or PO Boxes complexes.

Code	Explanation
P	A ZIP Code used only for PO Boxes.
U	Unique: A ZIP Code assigned to an organization or government institution such as the IRS.
M	Military: A ZIP Code assigned to an APO/FPO.
Empty	A standard ZIP Code.

Address2

Map the field that will receive the standardized street address that was not used for verification, if two address lines were present.

Suite

Map the field that will receive the formatted suite information, if any.

Urbanization

Map the field that will receive the Urbanization name, if any.

See the entry for the **Urbanization** input field for more information on this field.

Private Mailbox

Map the field that will receive the private mailbox number. This field is populated if the primary address was identified as a Commercial Mail Receiving Agency (CMRA) and the secondary address information is present. This field would also be populated if the secondary address information contains the string “PMB,” thus identifying the address as a private mailbox.

Delivery Point

The deliver point is a two-digit number that combined with the ZIP + 4, generates a unique number for each address in that ZIP + 4. This number is used to generate barcodes, so if you plan you use this address data for commercial mailing, it would be necessary to map an output field for this information.

Carrier Route

The Carrier Route is a four-character code that indicates which mail carrier delivers to the input address.

The first character of this Carrier Route 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

CMRA

If this field returns the string “Y,” the primary input address is a Commercial Mail Receiving Agency (CMRA), such as the UPS Store® or similar private mailbox business location.

Delivery Point Check Digit

The delivery point check digit is an additional one-digit number that is combined with the Delivery Point when generating barcodes for mailing.

eLot® Number

This number indicates where the current address falls in the delivery order within the ZIP + 4.

The eLot fields may be required if you are sending the data on to be presorted for mailing using USPS Standard Mail® rates.

eLot Order

The eLot Order returns the letters “A” or “D” to indicate whether the Post Office™ delivers mail within the ZIP + 4 in ascending eLot Number order or descending.

Therefore if the eLot Number is 1 and the eLot Order is “D,” then the address is typically the last delivery of the day.

Geographic Information

The geographic information fields return information regarding the physical location of the address.

City Abbreviation

If the name returned by the City field is longer than 13 letters, the City Abbreviation field will return the official abbreviation the Post Office has associated with that city or municipality name. For example, the City Abbreviation field will return the abbreviation “Rcho Sta Marg” for “Rancho Santa Margarita.”

If the value returned by the City field is 13 letters or shorter, the City Abbreviation field will return the full city or municipality name.

County Name

This field returns the name of the county where the input address is located.

County FIPS

This field returns the six-digit Federal Information Processing Standard (FIPS) code for the county where the input address is located.

Congressional District

This field returns the standard postal abbreviation for the state, followed by a two-digit number that indicates the congressional district where the input address is located.

Place Code

This field returns the Census Bureau Place Code for the physical location of the input address. This information is useful when the boundaries of the ZIP + 4 overlap city limits.

Place Name

This field returns the official Census Bureau name for the location indicated by the Place Code.

Time Zone

This field returns the name of the time zone where the verified input address is located.

Time Zone Code

This field returns a one- or two-digit number code for the time zone where the verified input address is located. The number also indicates the number of hours that the times zone is behind UTC/GMT. In other words, Eastern Standard Time has a time zone code of 5, indicating that the Eastern time zone is five hours behind UTC/GMT.

This number does not indicate differences due to daylight savings time.

Latitude

This field returns a number indicating the latitude of the input address as the number of degrees north of the equator.

If a Delivery Point field is not mapped, this latitude will be accurate to the centroid of the ZIP + 4.

If a Delivery Point field has been mapped, this latitude will be accurate to the 11-digit rooftop level, if possible.

Longitude

This fields returns a number indicating the longitude of the input address as the number of degrees east of the 0 degrees (the Greenwich meridian). Because North America is west of the Greenwich Meridian, this number will always be negative.

If a Delivery Point field is not mapped, this longitude will be accurate to the centroid of the ZIP + 4.

If a Delivery Point field has been mapped, this longitude will be accurate to the 11-digit rooftop level, if possible.

Census Information Fields

The following fields return information useful in determining the demographic characteristics for the location of the input address.

CBSA Code

Metropolitan and micropolitan statistical areas (metro and micro areas) are geographic entities defined by the U.S. Office of Management and Budget (OMB) for use by Federal statistical agencies in collecting, tabulating, and publishing Federal statistics. The term “Core Based Statistical Area” (CBSA) is a collective term for both metro and micro areas. A metro area contains a core urban area of 50,000 or more population, and a micro area contains an urban core of at least 10,000 (but less than 50,000) population. Each metro or micro area consists of one or more counties and includes the counties containing the core urban area, as well as any adjacent counties that have a high degree of social and economic integration (as measured by commuting to work) with the urban core.

The CBSA Code is a five-digit code for the specific CBSA of the input address.

CBSA Level

This field returns the level of the CBSA for the submitted address: micropolitan or metropolitan.

CBSA Title

This field returns the official U.S. Census Bureau name for the Core Based Statistical Area (CBSA) of the input address.

CBSA Division Code

This field returns the numeric code for the division within the Core Based Statistical Area (CBSA), if any.

Some CBSA's are broken into parts known as divisions. In this case, the CBSA Division fields will also be populated. If not, these fields will be empty. Each division also has a Code, Level and Title.

CBSA Division Level

This field returns the level of the CBSA division for the submitted address: micropolitan or metropolitan.

CBSA Division Title

This field returns the official U.S. Census Bureau name for the CBSA division of the input address.

Census Block

This field returns the Census Block number for the input data.

Census blocks, the smallest geographic area for which the Bureau of the Census collects and tabulates decennial census data, are formed by streets, roads, railroads, streams and other bodies of water, other visible physical and cultural features, and the legal boundaries shown on Census Bureau maps.

A Census Block Group is a cluster of blocks having the same first digit of their 3-digit identifying numbers within a Census Tract or Block Numbering Area (BNA). For example, Census Block Group 3 within a Census Tract or BNA includes all blocks numbered between 301 and 397. In most cases, the numbering involves substantially fewer than 97 blocks. Census Block Groups never cross Census Tract or BNA boundaries, however, they may cross the boundaries of county subdivisions, places, American Indian and Alaskan Native areas, urbanized areas, voting districts, and congressional districts. Census Block Groups generally contain between 250 and 550 housing units, with the ideal size being 400 housing units.

Census Blocks are small areas bordered on all sides by visible features such as streets, roads, streams, and railroad tracks, and by invisible boundaries such as city, town, township, county limits, property lines, and short, imaginary extensions of streets and roads.

The Census Block function returns a 4-character string. The first digit is the Block Group and the last three characters (if any) are the Block Number.

Census Tract

This field returns the Census Tract number for the input data.

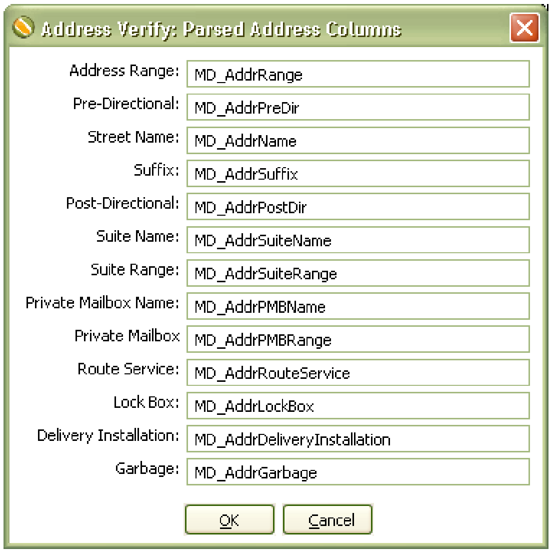
Census Tracts are small, relatively permanent statistical subdivisions of a county. Census Tracts are delineated for all metropolitan areas (MA's) and other densely populated counties by local census statistical areas committees following Census Bureau guidelines (more than 3,000 Census Tracts have been established in 221 counties outside MA's).

This field returns a four or six-character string value.

The Census Tract is usually returned as a 4-digit number. However, in areas that experience substantial growth, a Census Tract may be split to keep the population level even. When this happens, a 6-digit number will be returned.

Parsed Address Fields

These fields return street address information, split into its components.



Address Verify: Parsed Address Columns

Address Range:	MD_AddrRange
Pre-Directional:	MD_AddrPreDir
Street Name:	MD_AddrName
Suffix:	MD_AddrSuffix
Post-Directional:	MD_AddrPostDir
Suite Name:	MD_AddrSuiteName
Suite Range:	MD_AddrSuiteRange
Private Mailbox Name:	MD_AddrPMBName
Private Mailbox	MD_AddrPMBRange
Route Service:	MD_AddrRouteService
Lock Box:	MD_AddrLockBox
Delivery Installation:	MD_AddrDeliveryInstallation
Garbage:	MD_AddrGarbage

Address Range

This field returns the street number.

Pre-directional

This field returns any directional indicators that precede the street name. The directionals are returned as one- or two-character abbreviations. If the input data contained “Southwest,” it would be shortened to “SW.”

Street Name

This field returns only the street name, minus the suffixes or directionals. For “SW Main Street,” this would return “Main.”

Suffix

This field returns the standard abbreviations for the street types: “RD,” “ST,” “AVE,” “BLVD,” etc.

Post-directional

This field returns any directional indicators that follow the street name. The directionals are returned as one- or two-character abbreviations. If the input data contained “Southwest,” it would be shortened to “SW.”

Suite Name

This field returns the descriptive portion of the secondary address: “SUITE,” “APT,” “UNIT,” etc.

Suite Number

This field returns the numeric portion of the secondary address.

Private Mailbox Name

This field returns the descriptive portion of the private mailbox number. A private mailbox, or PMB, is a secondary address associated with a Commercial Mail Receiving Agency (CMRA).

Because the CMRA may itself be located in a suite, such an address may have both a suite number and a PMB number.

The name portion of the PMB will normally be either "#" or "PMB."

Private Mailbox Number

This field returns the numeric portion of the private mailbox number.

Route Service

This field returns the route service number for a Canadian address. Route Service is typically used to designate an address in a rural area.

Lock Box

This field returns the lock box number for a Canadian address. Lock boxes are similar to PO Boxes and the terms are often used interchangeably.

Delivery Installation

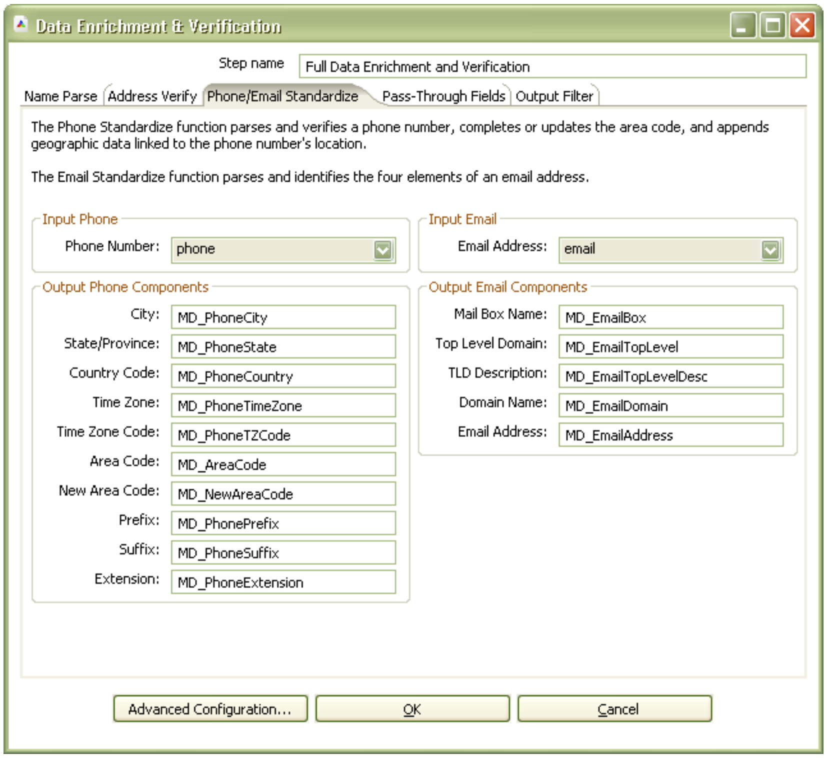
This field returns the delivery installation information for a Canadian address. The delivery installation is the Canada Post facility that services the address.

Garbage

This field returns any text from the street address field that does not fit into one of the above categories.

Phone/Email Standardize Tab

The fields on this tab are used to take a phone and/or an email address, and then parse and standardize this data.



Step name: Full Data Enrichment and Verification

The Phone Standardize function parses and verifies a phone number, completes or updates the area code, and appends geographic data linked to the phone number's location.

The Email Standardize function parses and identifies the four elements of an email address.

Input Phone

Phone Number: phone

Input Email

Email Address: email

Output Phone Components

City: MD_PhoneCity
State/Province: MD_PhoneState
Country Code: MD_PhoneCountry
Time Zone: MD_PhoneTimeZone
Time Zone Code: MD_PhoneTZCode
Area Code: MD_AreaCode
New Area Code: MD_NewAreaCode
Prefix: MD_PhonePrefix
Suffix: MD_PhoneSuffix
Extension: MD_PhoneExtension

Output Email Components

Mail Box Name: MD_EmailBox
Top Level Domain: MD_EmailTopLevel
TLD Description: MD_EmailTopLevelDesc
Domain Name: MD_EmailDomain
Email Address: MD_EmailAddress

Input Phone/Email

This is where you map the input fields containing the original phone numbers and email addresses.

Phone Number

The Phone input field requires a 10-digit phone number (for example: 1-(800)-800-6245)

Email Address

This string value must, at the minimum, containing the basic components of an email address: two strings of text separated by a “@” character.

Output Phone Components

Use this column to map output fields for the geographical and parsed phone number data. Because of number portability, the geographic information may not reflect the actual location of the phone number’s owner for wireless or VOIP numbers.

City

This field returns the city associated with the phone number area code and prefix.

State

This field returns the two-character state abbreviation associated with the phone number area code and prefix.

County

This field returns the county name for the location associated with the phone number area code and prefix.

County FIPS

This field returns the five-digit county FIPS code associated with the phone number area code and prefix.

Country Code

This field returns the country code associated with the input phone number. This is the two-character abbreviation for the United States or Canada and not the numeric international dialing code.

Time Zone

This field returns the name of the time zone where the input area code and prefix are located.

Time Zone Code

This field returns a one- or two-digit number code for the time zone where the area code and prefix are located. The number also indicates the number of hours that the time zone is behind UTC/GMT. In other words, Eastern Standard Time has a time zone code of 5, indicating that the Eastern time zone is five hours behind UTC/GMT.

This number does not indicate differences due to daylight savings time.

Area Code

This field returns the Area Code portion of the parsed phone number.

New Area Code

If the Area Code associated with the input phone number has undergone a recent change due to an area code split, this field will return the new, updated area code.

Prefix

This field returns the three-digit prefix portion of the parsed phone number.

Suffix

This field returns the four-digit suffix portion of the parsed phone number.

Extension

If the input phone number contained any extension information, that would be returned by this field.

Output Email Components

These fields return the parsed components of the input email address.

Mailbox Name

This field returns the portion of the email address that precedes the “@” character. For “ray@melissadata.com,” this field would return “ray.”

Domain Name

This field returns the domain name from the parsed email address, minus the top level domain. For “ray@melissadata.com,” this field would return “melissadata” (without the “.com”).

Top Level Domain

This field returns the top level domain (TLD) indicator from the input email address. For “ray@melissadata.com,” this would return the “dot com” portion.

Top Level Domain Description

This field returns the official text description associated with the top level domain. Not all TLDs have a description.

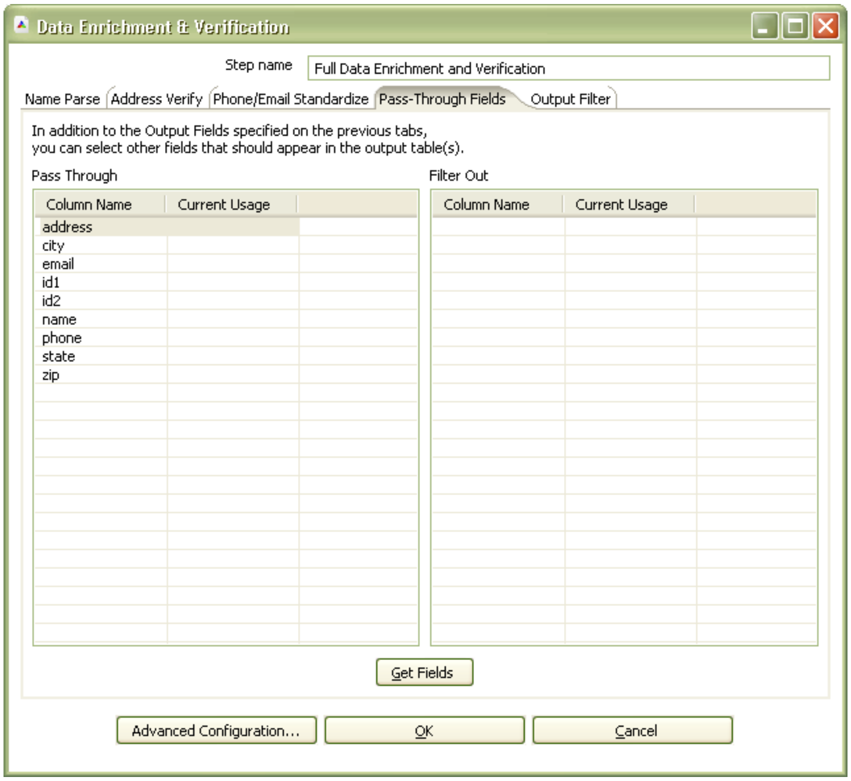
Email Address

This field returns the complete email address, standardized and corrected according to the options selected in the Email Standardize Options.

Pass-Through Fields

This shows a list of the fields in your input table that will be passed to the output table without modification. If you wish to filter out some of the pass through columns, drag and drop the columns from the pass-through table to the filter out table.

- **Column Name** — The column names from the original input table
- **Current Usage** — Where the field is currently being used as input
- **Data Type** — the data type of each column
- **Length** — Maximum number of bytes returned
- **Precision** — Maximum Number of digits
- **Scale** — Maximum number of digits to the right of the decimal point



Step name: Full Data Enrichment and Verification

In addition to the Output Fields specified on the previous tabs, you can select other fields that should appear in the output table(s).

Pass Through

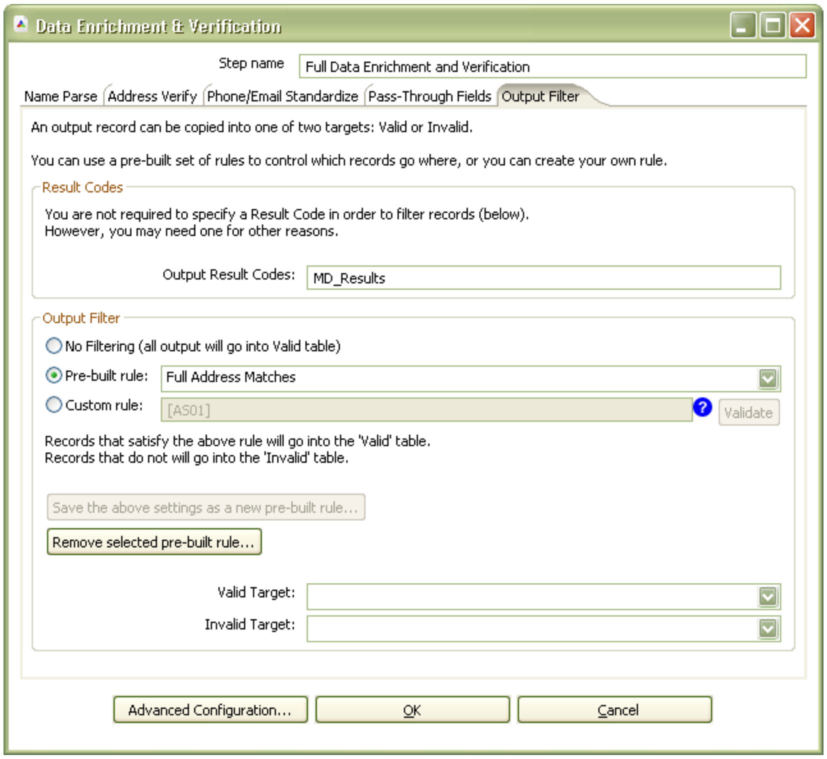
Column Name	Current Usage
address	
city	
email	
id1	
id2	
name	
phone	
state	
zip	

Filter Out

Column Name	Current Usage

Output Filter

The Contact Verification Component will output records to one of two tables: the valid table and the invalid table. The Output Filter tab allows you to control which records go to which table.



Data Enrichment & Verification
Name Parse
Address Verify
Phone/Email Standardize
Pass-Through Fields
Output Filter
Result Codes
Output Filter
Validate
Save the above settings as a new pre-built rule...
Remove selected pre-built rule...
Advanced Configuration...
OK
Cancel

Step name: Full Data Enrichment and Verification

An output record can be copied into one of two targets: Valid or Invalid.

You can use a pre-built set of rules to control which records go where, or you can create your own rule.

You are not required to specify a Result Code in order to filter records (below).
However, you may need one for other reasons.

Output Result Codes: MD_Results

○ No Filtering (all output will go into Valid table)

● Pre-built rule: Full Address Matches

○ Custom rule: [AS01]

Records that satisfy the above rule will go into the 'Valid' table.
Records that do not will go into the 'Invalid' table.

Valid Target:

Invalid Target:

Result Codes

The valid/invalid determination is made according to result codes returned by the underlying objects. The Component will combine them into a single comma-delimited string and write them to the field you map here.

Output Filter

There are three options for the output filter:

No Filter

All records are written to the valid table.

Pre-built Rule

The Component comes with several pre-built rules for common applications:

1. **Full Address Matches** — The record is valid only with a valid and deliverable mailing address.
2. **Street Address Matches** — The record is valid if the street address is deliverable, even if the suite information is missing.
3. **Full Address Match and Deliverable by all carriers** — The record is valid only with a valid and deliverable mailing address that is also served by alternate delivery services such as Fed Ex and UPS.
4. **Non-Questionable Names** — The record is valid if the name information was complete and contains no questionable information (vulgarities or common nuisance names).
5. **Valid Email Domains** — The record is valid if the domain for the email address was either confirmed to be valid or, at the very least, is not known to be invalid.
6. **Valid and Allowable Email Domains** — The record is valid if the domain for the email address was either confirmed to be valid or is not known to be invalid, and is not a mobile email address (which are not deliverable per FCC regulations).
7. **Full Phone Matches** — The record is valid only if the phone number was verified to the 10-digit level.
8. **Full and Partial Phone Matches** — The record is valid if the phone number was verified to either the 10-digit or 7-digit level.

Custom Rule

This option allows you to create custom rules for filtering based off the returned result codes.

Custom rules are an advanced subject. You should be comfortable with Boolean operations before using custom rules in a production environment. For more information on the available result codes, see **Appendix A: Custom Output Filters** on page 38.

Save the Above Settings as a New Pre-Built Rule

Clicking this button allows you to save your custom rule and save it for future referencing.

Remove Selected Pre-Built Rule

Clicking this button removes the currently selected custom rule from the list.

Valid Target

This specifies the ouput destination for the valid records based off the selected filtering. This field will be automatically populated after setting the mapping for the Valid Target in your pallet.

Invalid Target

This specifies the ouput destination for invalid records based off the selected filtering. This field will be automatically populated after setting the mapping for the Invalid Target in your pallet.

Custom Output Filters

The ability of the DEVC to create custom filters for discriminating between valid and invalid records depends on result codes. The following section contains a list of the possible return codes for each section and their meaning, to help you determine which ones are necessary for your purposes.

Name Parsing

Code	Meaning	Description
NS01	Parsing successful.	Name parsing was successful.
NS02	Error while parsing.	There was error. Check error codes below
NS03	First name spelling corrected	The spelling of the FirstName field was corrected.
NS04	Second name spelling corrected	The spelling of the FirstName2 field was corrected.
NS05	First Name 1 was found in the first name lookup table.	The return value of the Name 1 First Name output field was verified against Contact Verification Component's table of first names.
NS06	Last Name 1 was found in the last name lookup table.	The return value of the Name 1 Last Name output field was verified against Contact Verification Component's table of last names.
NS07	First Name 2 was found in the first name lookup table.	The return value of the Name 2 First Name output field was verified against Contact Verification Component's table of first names.
NS08	Last Name 2 was found in the last name lookup table.	The return value of the Name 2 Last Name output field was verified against Contact Verification Component's table of last names.
NE01	Unrecognized format.	Two names were detected but the FullName string was not in a recognized format.
NE02	Multiple first names detected	Multiple first names — could not accurately genderize.
NE03	Vulgarity detected	A vulgarity was detected in the name.
NE04	Suspicious word detected	The name contained words found on the list of nuisance names (such as "Mickey Mouse")
NE05	Company name detected.	The name contained words normally found in a company name.
NE06	Non-alphabetic character detected.	The name contained a non-alphabetic character.

Address Verification

Code	U.S. Only	Meaning	Details
AS01		Address Matched to Postal Database	Street Address is valid and deliverable. Check AE08 and AE09 for full deliverability.
AS02		Street Address Match	Address street matched to USPS database but a suite was missing or invalid.
AS09		Foreign Postal Code Detected	Postal Code from a non-supported foreign country detected. A US or Canadian Postal Code can also return this error if the US or Canadian data files are not initialized.
AS10	Y	Address Matched to CMRA	Address belongs to a Commercial Mail Receiving Agency (CMRA) like the UPS Store®.
AS13	Y	Address has been Updated by LACSLink.	Address has been converted by LACSLink from a rural-style address to a city-style address.
AS14	Y	Suite Appended by SuiteLink	A suite was appended by SuiteLink using the address and company name.
AS15	Y	Suite Appended by AddressPlus	A suite was appended by AddressPlus using the address and last name.
AS16		Address is vacant.	Address has been unoccupied for 90 days or more.
AS17		Alternate delivery	Individual address does not receive mail but delivery is made to an alternate address or has potential for delivery in the near future.
AS18	Y	DPV Locked Out	DPV processing was terminated due to the detection of what is determined to be an artificially created address. No address beyond this point has been DPV validated.
AS20	Y	This address is deliverable by USPS only.	Alternate carriers such as UPS and FedEx do not deliver to this address.
AE01		Zip Code Error	The Postal Code does not exist and could not be determined by the city/municipality and state/province.
AE02		Unknown Street Error	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.

Code	U.S. Only	Meaning	Details
AE03		Component Mismatch Error	Either the directionals or the suffix field did not match the post office database, or there was more than one choice for correcting the address.
AE04		Non-Deliverable Address Error	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.
AE05		Multiple Match Error	Address matched to multiple records. 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.
AE06	Y	Early Warning System Error	This address has been identified in the Early Warning System (EWS) data file and should be included in the next postal database update.[1]
AE07		Missing Minimum Address Input Error	Minimum required input of address/city/state or address/zip not found.
AE08		Suite Range Invalid Error	The input street address was found but the input suite number was not valid.
AE09		Suite Range Missing Error	The input street address was found but a required suite number is missing.
AE10		Primary Range Invalid Error	The street number in the input address was not valid.
AE11		Primary Range Missing Error	The street number in the input address was missing.
AE12	Y	PO, HC, or RR Box Number Invalid Error	The input address PO, RR or HC number was invalid.

[1] EWS (Early Warning System) data is composed of new addresses (for example, new housing developments) that are scheduled for inclusion in the USPS national database. The USPS updates this data every week to reflect new construction and home ownership. When verifying an address, DEVC consults the EWS file as well. Having an updated EWS file prevents DEVC from miscoding new address information with data in the older USPS national database file. For example, an input of 44 Legacy Drive would be changed to 44 Legacy Street if Legacy Drive does not yet exist in the USPS national database. If Legacy Drive is in the EWS file, however, DEVC would recognize it as a new street and not code the record. The next database update should contain the new information and enable DEVC to properly code the record.

Be sure to frequently download the latest EWS file in order to have the most current address records. However, using the EWS file is optional. Current EWS files are available every Thursday. The EWS file can be downloaded from ftp://ftp.melissadata.com/Updates/ews.txt.

Code	U.S. Only	Meaning	Details
AE13	Y	PO, HC, or RR Box Number Missing Error	The input address is missing a PO, RR, or HC Box number.
AE14	Y	CMRA Secondary Missing Error	Address Matched to a CMRA Address but the secondary (Private mailbox number) is missing.
AC01		ZIP Code Change	The five-digit ZIP Code™ was added or corrected based on the city and state names.
AC02		State Change	The state name was corrected based on the combination of city name and ZIP Code.
AC03		City Change	The city name was added or corrected based on the ZIP Code.
AC04		Base/Alternate Change	Some addresses have alternate names, often chosen by the owner or resident for clarity or prestige. This change code indicates that the address from the official, or "base," record has been substituted for the alternate.
AC05		Alias Change	An alias is a common abbreviation for a long street name, such as "MLK Blvd" for "Martin Luther King Blvd." This change code indicates that the full street name has been substituted for the alias.
AC06		Address1/Address2 Swap	The value passed to SetAddress could not be verified, but SetAddress2 was used for verification. The value passed to the SetAddress function will be returned by the GetAddress2 function.
AC07		Address1/Company Swap	A Company name was detected in address line 1 and moved to the GetCompany function.
AC08		Plus4 Change	A non-empty Plus4 was changed.
AC09		Urbanization Change	The Urbanization was changed.
AC10		Street Name Change	The street name was changed due to a spelling correction.
AC11		Suffix Change	The street name suffix was corrected, such as from "St" to "Rd."
AC12		Street Directional Change	The street pre-directional or postdirectional was corrected, such as from "N" to "NW."
AC13		Suite Name Change	The unit type designator for the secondary address was changed, such as from "STE" to "APT."

GeoCoding

Code	Details
GS01	Record was coded to the ZIP + 4 centroid.
GS02	Record was coded to the ZIP + 2 centroid.
GS03	Record was coded to the 5-digit ZIP Code centroid.
GS05	Information was coded to rooftop level. (Local Only)
GS06	Information was coded to an interpolated rooftop level. (Local Only)
GE01	ZIP Code Error. An invalid ZIP Code was entered.
GE02	ZIP Code not found. The submitted ZIP Code was not found in the database.
GE03	Demo Mode. GeoCoder Object is in Demo Mode and a ZIP Code outside the demo range was detected. (Local Only)
GE04	The GeoCoder Object data files are expired. Please update with the latest data files. (Local Only)

Phone Numbers

Code	Meaning	Details
PS01	10-digit match	The phone number was verified to the full 10-digit level.
PS02	7 digit match	Phone Object was able to verify the area code, the exchange, and the first digit of the suffix.
PS03	Corrected Area Code	Corrected area code that was changed according to the ZIP Code it falls into.
PS04	Outside Demo Range	Phone number is outside the demo range.
PS05	Expired Database	The database has expired. Contact Melissa Data. (Local Only)
PS06	Updated Area Code	The area code has split and the new area code is in the NewAreaCode function. This area code was changed using area code/prefix data. Applies to new splits only.
PS07	Exchange Type - Cell Phone	The phone number belongs to a wireless phone.
PS08	Exchange Type - Not Special	The phone number belongs to a standard land line.
PS09	Exchange Type - VOIP	The phone number is assigned to a Voice Over IP (VOIP).

Code	Meaning	Details
PS10	Phone Type: Residential	The phone number belongs to a residence.
PS11	Phone Type: Business	The phone number is assigned to a business.
PS12	Phone Type: SOHO	The phone number is assigned to a small business or home office.
PE01	Bad Area Code	The area code does not exist in our database or contains non-numbers.
PE02	Blank Phone Number	Phone number was not populated.
PE03	Bad Phone Number	Too many or too few digits.
PE04	Multiple Match	Multiple Match (could not choose between 2 or more area codes as a bad or missing area code was encountered and the distance between the area codes was too close to choose one over the other).
PE05	Bad Prefix	The prefix does not exist in the database.
PE06	Bad ZIP Code	An invalid ZIP Code was entered.

Email Addresses

Code	Meaning	Details
ES01	Valid Email Domain	The domain name of the submitted email address was confirmed as valid by either the DatabaseLookup or MXLookup
ES02	Invalid Email Domain	The domain name of the submitted email address was either not located by MXLookup or was located on the list of invalid domains.
ES03	Unverified Email Domain	The domain name of the submitted email address was not confirmed as valid by either DatabaseLookup, but was not found on the list of invalid domain names.
ES04	Mobile Email Address	The domain name of the submitted email was identified as a mobile email address, classified as not deliverable by FCC regulations.
ES10	Syntax was changed	The syntax of the submitted email address was changed.

Code	Meaning	Details
ES11	Top level domain changed.	The top level domain of the submitted email address was changed.
ES12	Domain Changed (Spelling)	The domain of the submitted email address was corrected for spelling.
ES13	Domain Changed (Update)	The domain of the submitted email address was updated due to a domain name change.
EE01	Syntax Error	There is a syntax error in the submitted email address.
EE02	Top Level Domain Not Found	The top level domain of the submitted email address was not found.
EE03	Mail Server Not Found	The mail server (domain) of the submitted email address was not found.
EE04	Invalid Mailbox Name	An invalid mailbox name was detected (IE: noreply). To configure invalid mailbox names, review mdEmailConfig.ini.
EE05	Email Object Not Initialized	Email Object was not initialized properly. Please review your code and data files. (Local Only)

Customizing Settings for Name Parsing

Customizing Name Parsing settings is available with local solutions only.

The main data folder for the Contact Verification Component includes a plain text file named mdName.cfg, which allows you to add to and override some of the default settings for the name parsing functions in the Component.

You can add entries to and remove entries from the installed tables of:

- Prefixes
- First names
- First name spelling corrections
- Casing for last name prefixes
- Proper casing of last names
- Name suffixes
- Dual name connectors
- Suspect names and vulgarities
- Dual name patterns

Each section of the file begins with the name of the table enclosed in square brackets. To add an entry to the table, simply add a line to that section. If that entry duplicates one in the default table, the Component uses the entry in the configuration file, allowing you to override the original settings.

To remove an existing entry from the default table, begin the line with a dash or minus sign ("-").

An entry consists of a single line of text containing several values separated by commas.

For example, a line in the Prefix section of the file must follow this format:

```
<First Name>, <Sex>, <Misspelling>, <Rank>, <Case>
```

Replacing each placeholder with a real value, the actual entry might look like this:

```
William, 7
```

Unused values at the end of a line are optional. Empty values in the middle of a line must have the necessary comma.

Prefix

```
[Prefix]
```

Modifies the list of name prefixes.

```
<Prefix>, <Sex>, <Dual Expansion>, <Case>
```

Field	Contents
`<Prefix>`	Word/phrase to look up.
`<Sex>`	The gender associated with this prefix, if any (M or F).
`<Dual Expansion>`	Replacement for this prefix when building dual names (for example, “Dr. & Mrs.” for “D/M”).
`<Case>`	Proper Casing for this word/phrase.

First Name

```
[FirstName]
```

Modifies the list of first names used for parsing and genderizing.

```
<First Name>, <Sex>, <Misspelling>, <Rank>, <Case>
```

Field	Contents
`<First Name>`	The name to look up.
`<Sex>`	The gender associated with the name: 1 = Always female 2 = Usually female 3 = Often female 4 = Neutral 5 = Often male 6 = Usually male 7 = Always male.
`<Misspelling>`	‘X’ if the name is misspelled (if the correction is known, add it to [FirstNameFix]).
`<Rank>`	Name's rank in U.S. Census data. Leave blank.
`<Case>`	Proper casing for this name.

First Name Fix

```
[FirstNameFix]
```

A list of misspelled first names and the correct spelling. For a correction to take place, the misspelled name must also exist in the First Name section, with an “X” in the `<Misspelling>` field.

```
<Misspelling>, <Correction>
```

Field	Contents
`<Misspelling>`	A misspelled first name.
`<Correction>`	The correction (properly cased).

Last Name Prefix

```
[LNPrefix]
```

List of the proper casing for last name prefixes, such as “von” in “von Beethoven.”

```
<Last Name Prefix>, <Case>
```

Field	Contents
`<Last Name Prefix>`	Last name prefix.
`<Case>`	Proper Casing for this word.

Last Name

```
[LastName]
```

List of last names, including proper casing.

```
<Last Name>, <Rank>, <O-Name>, <Case>
```

Field	Contents
`<Last Name>`	Name to look up.
`<Rank>`	Name's rank in the U.S. Census data. Leave this field blank.
`<O-Name>`	‘X’ if the name is an Irish O'Name (such as “O’Kelly”).
`<Case>`	Proper Casing for this name.

Suffix

```
[Suffix]
```

Modifies the list of name suffixes.

```
<Suffix>, <Prefix>, <Salutation Remove>, <Dual Name Remove>, <Case>
```

Field	Contents
`<Suffix>`	Word/phrase to look up.

Field	Contents
`<Prefix>`	Prefix associated with this suffix (ie, “Dr.” for “MD”).
`<Salutation Remove>`	‘X’ if this suffix should be removed when generating salutations.
`<Dual Name Remove>`	‘X’ if this suffix should be removed when dual name parsing.
`<Case>`	Proper Casing for this word/phrase.

Dual Name Connector

```
[DualIndicator]
```

Modifies the list of dual name connecting words.

```
<Dual Name Connector>, <Delete>
```

Field	Contents
`<Dual Name Connector>`	Word/phrase to look up.
`<Delete>`	‘X’ if the connector should be removed (the name will not be considered a dual name).

Suspect Words

```
[Suspect]
```

Modifies the list of suspicious words, phrases, and possible vulgarities.

```
<Word/Phrase>, <Indicator>, <Case>
```

Field	Contents
`<Word/Phrase>`	Word/phrase to look up.
`<Indicator>`	Word indicator: V — Vulgarity C — Company indicator S — Suspicious name indicator
`<Case>`	Proper Casing for this word/phrase.

Dual Name Patterns

```
[DualPattern]
```

Modifies the list of patterns used for parsing dual names.

```
<Pattern>, <Counts>, <Name Types>, <Split Type>
```

Field	Contents
`<Pattern>`	Pattern: P — Prefix F — First name & — Connector word/phrase ? — Unknown (not found in lookups)
`<Counts>`	Counts (each position corresponds to position in <Pattern>: > — More than one word of that type = — Exactly one word of that type (space) — Any number of words of that type
`<Name Types>`	Name Types: 1 — Full name 2 — Inverse name 4 — Mixed first name 8 — Mixed last name
`<Split Type>`	How the pattern should be split: 1 — Mr. John Smith & Ms. Mary Jones 2 — John & Jane Smith 3 — Mr. & Ms. John & Jane Smith 4 — Mr. & Ms. John Smith 5 — Mr. John & Ms. Jane Smith 6 — Smith, John & Smith, Jane 7 — Smith, John & Jane 8 — Smith, Mr. John & Ms. Jane 9 — Mr. John & Ms. Jane 10 — Mr. & Ms. John & Jane 11 — Mr. & Ms. John 12 — John Smith & Jane

Customizing Settings for Standardizing Email

Customizing Email settings is available with local solutions only.

The Email standardizing function of the Contact Verification Component uses a database of valid domain names and top-level domains, plus corrections for common misspellings of popular domain names, updated every two months.

You can add to or override this information by modifying a file called **mdEmailConfig.ini**, which must be located in your main data path specified in the Advanced Configuration dialog. Changes to this file override any settings found in the default database and are not overwritten when Component is updated.

The file contains several sections, one for each type of change that the Component can perform.

To add any item to the list, add a line beginning with a "+" followed by the new item. To remove an item from the database, use the "-" character.

These changes do not make any permanent changes to the default database.

Comment lines can be added. Just begin the line with the "#" character.

Top Level Domains

This section begins with the line containing "[TLD]."

Optionally, you can add a space, followed by a description of the TLD. For example:

```
+.new A New Top Level Domain
```

To remove a top-level domain:

```
-.old
```

Valid Domains

This section begins with the line containing "[valid domains]." These domains will be assumed to be good and no MX Lookup will be performed if that option is enabled.

Adding a valid domain:

```
+validdomain.com
```

Removing a valid domain:

```
-validdomain.com
```

Invalid Domains

This section begins with the line containing "[invalid domains]." These domains will be assumed to be invalid and all email addresses with these domains will be considered invalid.

Adding an invalid domain:

```
+invaliddomain.com
```

Removing an invalid domain:

```
-invaliddomain.com
```

Domain Updates

This section begins with the line containing “[updates].” All old domains found in this section will be updated to the new, correct domain name.

Adding a domain update:

```
+olddomain.com->newdomain.com
```

Removing a domain update:

```
-olddomain.com->newdomain.com
```

Top Level Domain Misspellings

This section begins with the line containing “[TLD misspellings].” All TLDs matching the misspelled forms will be replaced with the corrected spelling.

Adding a TLD misspelling:

```
+bad->good
```

Removing a TLD misspelling:

```
-bad->good
```

Domain Misspellings

This section begins with the line containing “[domain misspellings].” All domains matching the misspelled forms will be replaced with the corrected spelling.

Adding a domain misspelling:

```
+800nails.com->800mail.com
```

Removing a domain misspelling:

```
-800nails.com->800mail.com
```

Invalid Mailboxes

This section begins with the line containing “[invalid mailboxes].” These mailboxes will return an error of “invalid mailbox”.

Adding an invalid mailbox:

```
+NoReply
```

Removing an invalid mailbox:

```
-NoReply
```

Address Handling

A key concept for address verification is understanding how the Component handles address data. The Component can accept two lines of street address information via the Address and Address 2 input fields.

The value passed to the Address input field will typically contain the primary street address (Street number, street name, plus any directionals and street suffixes). It may or may not contain also a secondary address, such as a suite number, address number, unit number, or a private mailbox located at a commercial mail receiving agency (CMRA).

The value passed to the Address 2 input field will often contain the secondary address information, if it is not submitted via Address or Suite input fields. the Address 2 input field may also be used if there is another primary address, either a Post Office Box or a separate street address, that is part of the same record.

If secondary address information is submitted via the Address 2 input field, the Component will append this information to the value passed to the Address input field before processing the record.

Example #1

If the following were submitted:

```
Address Line 1: 1234 Main Street
Address Line 2: Suite #101
```

This is the address that would actually be verified:

```
Address Line 1: 1234 Main Street Suite #101
Address Line 2: <empty>
```

If an additional primary address was sent to the Address 2 input field, such as a P.O. Box or a second street address, and the address sent to the Address input field cannot be verified, then the Component will attempt to verify the second address line.

Example #2

If the following were submitted:

```
Address Line 1: 1234 Main Street
Address Line 2: P.O. Box 101
```

Assuming that the first address didn’t contain a verifiable address, this is when the Component would consider the contents of the second address line.

If the second address line contains a verifiable address, then the values submitted via the Address 2 field will be used. In that case, the Component will swap the values submitted via the Address and Address 2 input fields and return the standardized contents of these fields in this order:

```
The Address output field returns “P.O.Box 101.”
The Address 2 output field returns “1234 Main St.”
```

If the contents of neither field can be successfully verified, the Component will flag the record as unverified and the values submitted to the Address and Address 2 input fields will be returned in their original order:

```
The Address output field returns "1234 Main Street."
The Address 2 output field returns "P.O.Box 101"
```

Secondary Addresses

Secondary addresses include suite numbers, unit numbers and residential apartment numbers. It could also refer to a private mailbox (PMB) at a Commercial Mail Receiving Agency (CMRA).

The secondary address can be passed at the end the first address line, as the address 2 input field or via the Suite input field.

The National Postal database identifies certain primary addresses as high rises, business parks and apartment buildings. Therefore, the Component would be able to assign the correct secondary address designator to the following address:

```
1234 Main St #101
```

For example, if the primary address were an apartment complex, the Component would return "Apt 101" as the suite information.

CMRAs like the UPS Store® and other mailbox stores are a special case. These are often located at shopping centers and the store itself will have a suite number. This means that addresses located at a CMRA will often have both a suite number and a PMB number, like this:

```
1234 Main Street
Suite C1 PMB#101
```

CMRAs are identified in the national address database as this kind of business.

Example:

```
1234 Main Street #101
```

Assuming that 1234 Main Street is identified in the national database as a CMRA, the "#101" portion would be returned by the GetPrivateMailBox function rather than the GetSuite function.

Be aware that the Component will always treat a second item of secondary address information as a PMB number.

Example:

```
1234 Main Street
Suite C1 #101
```

Whether or not the primary address is identified in the database as a CMRA, the Component will identify the "#101" portion of the second line as a PMB number and return this number in the Private Mailbox field.

Even if an address is identified as a CMRA, if a secondary address is explicitly identified as a suite or anything other than private mailbox, this information will be treated as a suite and not a PMB.

Example:

```
1234 Main Street
Suite 101
```

Assuming that the primary address belongs to a CMRA, because the secondary address was supplied as “Suite 101,” this will be treated as a suite number and not a PMB number. If the number 101 was meant to refer to a PMB and there is not Suite 101 at this address, this would probably cause the address to fail verification.

City or Postal Code Information

In order to verify an address, the Component requires information on the city, the state or province, and a ZIP or Postal code. With the city and the state/province, the Component will determine the correct ZIP/Postal code and use this to verify the address.

Conversely, if supplied with a correct ZIP/Postal code, the Component can look up the city and the state/province.

Therefore either the ZIP/Postal code or the city plus the state/province are required. If the supplied city and state/province do not match the ZIP/Postal code, the Component will use the city and state/province to look up the ZIP/Postal code.