

# Melissa Data Address Verification Web Service

## *Introduction*

### **What is Melissa Data?**

Melissa Data is a database of addresses recognized by the US Postal Service. The record set contains all data for all addresses in the United States. Each address is properly formatted in a way consistent with a standard the postal service recognizes.

The software and database was purchased by the Division of Information Technology and is updated several times a year. At this time there is no cost associated with the use of the product although its use is limited to Rhode Island agencies to conduct official state business.

### **What isn't Melissa Data?**

The data does not match individuals to addresses. The database does not include information about who lives at a specific address – it simply determines whether the data entered is a valid address. If the address is not valid or there are multiple matches, an error message will be returned.

### **What is the Melissa Data web service?**

This is a service which enables you to pass an address, or partial address to the Melissa database. The address sent will be parsed against the database and results returned as a structured XML file.

### **What would I use Melissa Data for?**

There are three main ways to use this web service. The first is to verify the existence of an address. The second is to make sure that the address is in the proper format. The third is to provide additional information about an address such as the last four digits of an extended zip code, the county and even the time zone the address is in.

### **Why would I use Melissa Data?**

The most prominent reason to verify address data is to ensure that you are collecting the most accurate and complete data you can. Melissa Data Web service can correct incorrect zip codes and street names, add four digit codes to the zip code (zip plus four), and supply missing zip codes.

Using address verification is also important because it can help prevent mail from being returned to you as undeliverable. Properly formatting addresses can also lower postage rates and ensures prompt delivery (such as including the 4 digit zip code extension.)

The proper formatting of addresses can also help in matching user input against a database of known addresses such as a voter look-up search.

The Melissa Data object also contains information about an address ranging from time zone to congressional district.

### **Who can use Melissa Data?**

The Melissa Data Service is available to all Rhode Island state government agencies. The service is located outside the Department of Administration server firewall to provide access to all agencies.

All users must include their state mailing code in each request to the system. This code is used by Central Mail to track which agencies are utilizing the web service.

## ***Using Melissa Data***

### **Calling the web service**

To utilize the web service you will need to pass the service the following as many of the following parameters as you can: address, suite, city, providence, state, zip.

Remember to also include your state mailing code. This code is used by central mail to track which agencies are utilizing the web service.

Here is an example of a URL passing data as parameters as part of a CGI "get" request:

```
http://[MELISSA_SERVER]/cgi-bin/address/validate.cgi?address=10+Dorrance+Street&suite=Suite+610&city=Providence&state=RI&zip=02903&statepc=22932
```

The service will always return an XML object. The first part of the object will be enclosed as `<input></input>` and will contain all of the data originally sent in the request. The second part of the object will be enclosed as `<results></results>` and will vary dependent on the match of your input.

If the address is valid the service will return as much information about the address, enclosed in the `<results></results>` tag as it has available.

If the address is not valid the service will return appropriate error codes indicating what was wrong with the input address.

A complete list of results data and error codes can be found at the end of this document.

### **Examples of error codes**

#### **Example 1: Range Error**

In this example, the address submitted was 45 Main Street but the range of numbers on Main Street only goes up to 41. The error code "R" indicates that 45 Main Street is out of range and therefore an invalid address.

```
<result>
  <message>VerifyAddress Failed</message>
  <error_code>R</error_code>
  <error_string>Range Error</error_string>
  <status_code>X</status_code>
</result>
```

#### **Example 2: Unknown Street**

In this example the address submitted was 12 Blue Street in Barrington, but no such street exists. The error code "U" indicates that Blue Street is unknown and is therefore an invalid address.

```
<result>
  <message>VerifyAddress Failed</message>
  <error_code>U</error_code>
  <error_string>Unknown Street</error_string>
  <status_code>X</status_code>
</result>
```

Another error the system may generate is a Multiple Match error. An example would be an input of 12 Mathewson Street in Barrington. There is no Mathewson Street in that town but there is a Mathewson Lane and Mathewson Court. Because there is a number 12 on both the system rejected the input.

## **Correcting Addresses**

One of the most powerful aspects of the Melissa Data Web Service is the ability to correct poorly formatted addresses into the format recognized as correct by the US Postal Service.

If the service finds an address in it's dataset which on several of the parameters matches to the input data it will return the corrected address as part of the <results> object.

2 Crestwood Street in Barrington will be returned as 2 Crestwood Rd

In the same way, missing zip code information (including the 4 digit extension) will be supplied as part of the <results> object.

### **Note on Address Correction**

Users should understand that matching on incomplete addresses can be problematic. If a zip code is omitted from the <input> object, or is incorrect when the street address, city and state are matched to the record set the system may return the zip code it determines to be correct however there is no guarantee that this is the zip code matching the address intended by the user on input. For that reason it is always best to display aspects of the result back to the user before committing to the changes suggested by the system.

Users should also check input information against the result object to determine what has been changed to avoid the acceptance of wrong information.

## Appendix

### Error Codes

M	Multiple Matches
N	No Data Available For City
R	Range Error
T	Component Error
U	Unknown Street
X	Undeliverable Address
Z	Invalid ZIP Code
C	Canadian ZIP Code
D	Demo Mode Only

### Results XML

The following XML object is returned in the <results> tags. Notes have been included where the data tag may not be clear.

<address>  
<address2>  
<city>  
<state>  
<zip>  
<zip\_4>  
<suite>  
<suite\_status>

M	A suite number is required for the given street address but is missing from the submitted record.
R	A suite number is present on the submitted record but is either not required or is out of range for the given street address.
V	The suite field was verified.
X	The suite field was not coded.

<carrier\_route>  
<delivery\_point\_code>

A 1-digit string set after a successful return from the Verify Address Method. This 1-digit string makes up the 12 <sup>th</sup> position of a 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. <dpc_check_digit>
---

<city\_abbreviation>  
<country>  
<county>  
<county\_fips>

Federal Information Processing Standards Codes for states, counties, and named populated places.
--

<address\_type\_code>

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

**<address\_type\_string>**

**<zip\_type>**

**<urbanization>**

This property applies to Puerto Rican addresses only.

**<congressional\_district>**

**<lacs>**

Returns a one-character string which indicates if the input address has undergone a conversion to city-style street addresses. 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. The LACSLink service matches the old address with the updated address and corrects it as part of the Address Check process.

**<time\_zone>**

**<time\_zone\_code>**

**<msa>**

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

**<default\_flag\_indicator>**

Unknown at this time

**<status\_code>**

D	Demo Mode
E	Expired database
S (US)	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).
V	Street number validated to DPV level.
X	Address was not coded.
6	A Canadian address was fully coded.
7 (US)	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 (US)	The address was fully coded.

**<error\_code>**

M	Multiple Matches
N	No Data Available For City

R	Range Error
T	Component Error
U	Unknown Street
X	Undeliverable Address
Z	Invalid ZIP Code
C	Canadian ZIP Code
D	Demo Mode Only

**<error\_string>**

**<cmra>**

Returns a one character (Y/N) string which indicates whether or not an address is actually a private mailbox at a Commercial Mail Receiving Agency (CMRA) like the UPS Store

**The following return information about the parameters sent as part of the input object**

<parsed\_address\_range>

<parsed\_pre\_direction>

<parsed\_street>

<parsed\_suffix>

<parsed\_postdir>

<parsed\_suite\_name>

<parsed\_suite\_range>

<parsed\_box\_name>

<parsed\_box\_num>

<parsed\_garbage>

<private\_mailbox>