

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

Melissa Data offers a variety of advanced data quality APIs and Web Services to verify, correct, standardize, and parse contact data including street addresses, phone numbers, full names, and email addresses. These are multiplatform APIs available in 32/64-bit versions. The product names include: Address Object, Email Object, Name Object, Phone Object, MatchUp Object, GeoCoder Object, RightFielder Object and IPLocator Object.

## File & Name Standardization

1. Explain in detail how your tool consolidates several different input files with different file formats into single output file format.

The Melissa Data APIs are essentially blind to the data source – you control the file handling, allowing for the greatest flexibility to access (and process) data in any format or structure. Using native plugins for ETL tools such as SSIS or Pentaho Data Integration, the Data Quality Components allow cleansing of most file formats and outputting the format of choice.

2. Explain in detail how your tool parses input names into pre name, first name, middle name, last name and post name and sends the data to individual fields.

When a Full Name or Multiple Names are passed into Name Object it first determines the gender of the first-names from a lookup table and then parses the specific name components as individual output properties using proprietary logic. Name Object will parse names into all the individual components. Incoming data can be in varied formats and patterns (Mr John Smith; Smith, John; Mr J L Smith Sr: etc., as well as dual names (Mr and Mrs John Smith). There are more than 190,00 first and last names from the US Census in the Name Object.

3. Explain in detail how your tool splits combined names from one record into two separate records.

Name Object has a built-in list of words that connect names (“and,” “or,” “&,” etc.). It is the presence of these connectors that tell the object that a dual name may exist in the input full name string.

4. If a file contains a name field with a “Mr. and Mrs. Tom Jones,” explain how your tool splits these into two separate records (e.g. One record for Mr. Tom Jones and a second record for Mrs. Tom Jones).

The respective parsed names would be split into these properties, one set per name.

Results for name 'Mr. and Mrs. Tom Jones':

Prefix: Mr.	Prefix2: Mrs.
First Name: Tom	First Name2: Tom
Middle Name:	Middle Name2:
Last Name: Jones	Last Name2: Jones
Suffix:	Suffix2:
Gender: M	Gender2: F

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## File & Name Standardization

**5.** Explain in detail how your tool removes extraneous information from the input field data using search and replace functionality.

The Name Object will flag vulgarities that are detected in the name, and names that contain words found on the list of nuisance names (such as “Mickey Mouse”) There are no search and replace functionalities in Name Object.

**6.** Explain in detail how your tool standardizes inconsistent firm names in the input record.

The MatchUp Object can compare company acronyms and full companies and strip out inconsistencies in the firms before comparisons are done. Name Object will recognize Company names inside a Name field and standardize them.

**7.** If a file contains a name and address in a “free formatted” data field, explain how your solution is able to identify and parse the name and address into separate data fields (e.g. Name and Address).

The RightFielder Object will parse and identify input text into the main components of an address, name, phone, and email. You can use RightFielder Object to parse customer input contact data as a block of text in any order and output it to specified properties. This allows extraction of distinct details from user input, such as Name, Address, City, State, Zip, Phone, Email, Department, etc. RightFielder Object can also be used to intelligently identify misfiled data.

**8.** Explain how your tool is able to provide match standards for the first and middle names, allowing us to overcome two types of matching problems such as alternative spellings and nicknames.

The MatchUp Object uses fuzzy matching techniques and a nickname table to recognize phonemes such as “ph” and “sh,” nicknames (Liz, Beth, Betty, Elizabeth), and alternate spellings of names (Gene, Jean, Jeanne). MatchUp Object can also handle nearly-exact strings of characters, such as “Lewis” vs. “Ewis,” and “Palacino” vs. “Al Pacino” as well as initials such as “John Smith” to “J Smith.” MatchUp Object allows you to use Nickname matching (Charles = Chuck) and ‘nearness’ algorithms (John=Jonh) when matching records. You can also determine whether to disregard middle names, so ‘John L Smith = John Smith,’ or to include them, so you can catch ‘J. Lee Smith = John L Smith.’

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Address Standardization (Online/Batch)

- |   |  |
|---|--|
| <p><b>1. Service, DPV, and geocoding directory update? How are the updates delivered?</b></p>   | <p>EWS - Weekly<br/>CASS/DPV/LACSLink/SuiteLink/AddressPlus- Monthly<br/>Canada - Monthly<br/>GeoCoder- Quarterly<br/>IpLocator - Quarterly</p> <p>The Options for delivery are download and mailed DVDs.</p>  |
| <p><b>2. What is your geocoding data source and what type(s) of information is available (i.e. state code, FIPS county code, census tract, block group, census block detail, etc.)?</b></p> | <p>TomTom and Navteq are the sources for GeoCoding data.<br/>Data fields returned include: County Name/FIPS Code/CBSAs/Census Block/Census Tract, and Lat/Long.</p>  |
| <p><b>3. Does your solution contain/support NCOA<sup>Link</sup> and LACS<sup>Link</sup> features? If so, is this an add-on or regular feature? How often are data updates provided?</b></p> | <p>LACS<sup>Link</sup> is included with Address Object at no additional charge with monthly updates. NCOA<sup>Link</sup> is available with the SmartMover Web Service as add-on pricing. 24/48-month processing is updated weekly.</p>   |
| <p><b>4. Explain how your tool is capable of performing address scrubbing in online as well as batch mode?</b></p>  | <p>The Address Object functions by populating input properties, calling a processing method VerifyAddress() and retrieving the results from output properties. This methodology works in both modes by looping through each new record retrieving the results, calling clear properties before populating the input for the next record.</p> |
| <p><b>5. Explain in detail how your tool identifies and scrubs undeliverable addresses such as vacant lots, condemned building, etc...</b></p>  | <p>The Delivery Point Validation table and UAA table are part of the National Data files that allow Address Object to recognize these types of addresses and flag them as undeliverable. Vacant addresses are flagged with the appropriate Results Code.</p>   |

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Address Standardization (Online/Batch)

- |   |   |
|---|---|
| <b>6.</b> Explain in detail how your tool identifies and scrubs vanity address/alias address.   | A vanity or alias address is used in some cities and is a holdover from the past; Examples would be 114th Ave which is locally called River Road. Address Object will convert these addresses automatically by consulting the vanity table to a deliverable street address during processing. |
| <b>7.</b> Explain how your solution handles dual address lines. (i.e. P.O. box and street address).   | Only one valid address is processed per call, Address1 or Address2. If Address1 is valid, Address Object will only return Address2 without processing. If Address2 contains a suite, it will be output in the Suite field upon processing.  |
| <b>8.</b> Explain how your tool is able to handle swapping address lines. (i.e. rearranges address lines to conform to USPS guidelines).                    | Address Object will process Address1 if valid and return back the USPS information. However, if Address1 is not valid and Address2 is valid, Address Object will swap Address2 and Address1, and validate Address2.   |
| <b>9.</b> Explain how your solution supports the ZIP move and eLOT features.  | eLOT database is invoked by Address Object and verified addresses are appended with an eLOT Lot Order/Number seamlessly if the option is set. The ZIP-Move database is part and parcel of the National Data files that are used by Address Object when verifying US addresses.                |
| <b>10.</b> Explain how your solution supports assigning latitude, longitude, census tract and fips code to the address data.                                | The MD GeoCoder Object will append the lat/long, as well as Census Tract/Block group information to the address record (at the 11 digit ZIP Code level).  |
| <b>11.</b> Explain in detail the level of geocoding options available in your tool. For example, Zip/Centroid level, Address level, parcel/point level etc. | GeoCoding is available at no charge to the 5-digit level. Add-on pricing is available for 9-digit as well as Delivery Point 11-digit level.   |

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Address Standardization (Online/Batch)

**12.** Explain in detail how your solution supports local municipal taxes that are compliant with new regulatory requirements (i.e. KY city/county taxes, MN Tri-city Fire Surcharge, WV Surcharge, Mine Subsidence, etc...).

This is not a capability of the product line.

**13.** Explain in detail how your solution supports 2010 census tract changes. What are your plans to maintain existing census track versus implementing new census tract data in 2010.

Because the GeoCoder Object is refreshed with updated data, it will automatically be up to date with 2010 Census Data. The GeoCoder Object will thereby incorporate the 2010 Census Track data, but Melissa Data will not have the history of the misc. changes.

**14.** Explain how your solution supports suggestion lists. In other words, explain how your solution provides additional “pop up” addresses that are near matches to the original input address. Is the suggestion list available as part of your base solution?

The FindSuggestion method of Address Object can be used to find a suggestion for address records that do not verify. Simply request the FindSuggestion and FindSuggestionNext iteratively until you have a list of suggestions, starting with the best guess first. Also, Address Object's StreetData interface can be used to find correct ranges. For example, if you get an address with an incorrect range, you can use StreetData to pull out all the possible ranges for that street. If you get an incorrect suite, you can pull out all the possible suite values. With this tool, if you have the street name of an apartment complex, you can pull out all the suites that are present in that complex. This interface supports the use of wildcards so that “Park\*” will return any street that begins with “Park” in that particular ZIP Code.

**15.** Explain how your solution identifies and maps natural hazard and fire protection classification zones for insurance purposes.

This is not a direct capability of the product line. However, using the accurate lat/longs from GeoCoder Object, this information can be accurately mapped in certain GIS software.

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Matching/Householding/De-duping

**1.** Explain how your solution will match the data using a set of business rules against the database.

MatchUp Object's Matchcode Editor allows you to apply 16 simultaneous match criteria (business rules) per run. Any one rule that satisfies your criteria for two records will return a match. A return status code telling which rules of these (up to 16) returned a match can be used to evaluate the quality of a matched pair.

**2.** Explain in detail how the following features are incorporated/supported within your matching tool.

- Deduping
- Set matching rules
- Householding

With up to 36 defined data types, as well as a catch-all general datatype, each with individual settings, gives you nearly unlimited flexibility in creating match rules. This can be as simple as Address Householding, to Address + Names + Companies, etc. You define the rule criteria, and MatchUp Object will return output data for each record telling you whether it matched any other record, how many records it matched, what those other records are, and assign each group a unique group number.

**3.** Explain the different matching criteria that can be used to household customer records.

MatchUp provides basic Householding matchcodes – that use Zip Code and any address data you provide, but you are free to edit, copy, or alter these rules. MatchUp Object's address splitter (used behind the scenes to build keys) allows you to match inexact records like 'Twelve N Main Street' to '12 Main St. Apt 67.'

**4.** Explain how your solution will assign household ids to each unique customer record.

MatchUp Object assigns a unique number to each group of matched records. This is your link to matched records that may be in different files and not easily seen as duplicates.

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Architecture

- |  |  |
|--|--|
| <b>1.</b> Describe the logical and physical architecture design of your solution and how the system supports Service Oriented Architecture (SOA), Web services, batch processing/scheduling, and n-tier architectures. | The Data Quality APIs are libraries and must be included in to the build process and specially utilized by programs and Web applications written in most languages. The Web Services utilize SOAP/XML/REST based client/server messaging.  |
| <b>2.</b> What application language(s) and versions of the application architecture are required to support your proposed solution? (e.g. Java, .Net, VB, C++, etc.)   | Melissa Data APIs have native support for Java, Perl, PHP, Ruby, Python, .NET, C and C++, and COM+ enabled languages on most platforms. COM+ technologies are only for Microsoft Windows®.   |
| <b>3.</b> What 3rd party software products (i.e. Websphere, Microsoft SQL Server, etc) are required to support your solution?  | Other than a host OS, no third party middleware software products are required as the Data Quality object libraries and their respective data are self-contained and only accessible by the API.   |
| <b>4.</b> What application platforms (e.g. Websphere, Oracle, .Net, etc) does the application require and support?   | The Data Quality libraries do not require any third party application platforms. However, due to the availability of wrappers for most languages, they can be called by most widely available application platforms through an extension call.   |
| <b>5.</b> What DBMS platforms (e.g. Oracle, DB2, UDB, and SQL) does the application require and support?   | The libraries do not require any installed database as the data is in a proprietary optimized structure. However, due to the availability of wrappers, the libraries can be integrated by most widely available DBMS platforms as a stored procedure call.   |
| <b>6.</b> What operating system(s) (e.g. Windows, UNIX, Linux, etc) does the application require and support?  | Melissa Data APIs ship with native libraries for Linux, AIX, HP-UX, Solaris, and Windows Operating Systems both 32/64 bit and with multiple compiler versions offered. Please see the Platform compatibility matrix for a full listing. Native 64-Bit width is the recommended architecture for using the libraries. |

# Melissa Data Fact Sheet - Overview of Data Quality API Capabilities

## Integration

**1.** Describe in detail the integration API's and messaging protocols that are supported by your solution.

The Melissa Data APIs are libraries and do not support messaging protocols directly. The Web Service solutions support SOAP/XML/REST messaging protocols.

**2.** Describe what functionality of the system is exposed through Web services

Melissa Data offers discrete web service endpoints that expose the full functionality of the Data Quality APIs within a web service format.

## Configuration

**1.** Describe how the system supports the configuration of business rules to support data quality matching processes.

The end user implements Matchcodes that are sets of business rules that tell MatchUp Object which data fields to consider when determining if two records are duplicates. MatchUp Object uses the matchcode to construct a "match key," a simplified string of characters that represents the information within the record, enough to determine if the given record is unique or a duplicate. The MatchCode editor is a visual program provided to make the design of these rules very easy.

**2.** What development and test tools are required and/or recommended to support your solution?

Melissa Data recommends test databases and a development IDE for a target programming language to test and implement our solutions.