

Telco SmartSearch Web Service



Telco SmartSearch

Reference Guide

Copyright

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

© 2012. Melissa Data Corporation. All rights reserved.

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

Trademarks

Telco SmartSearch is a trademark of Melissa Data Corporation. Windows is a registered trademark of Microsoft Corp.

The following trademarks are owned by the United States Postal Service: ZIP, ZIP Code, and ZIP + 4.

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/telco-smartsearch-web-service.htm.htm>.

Document Code: WSTRFG
Revision Number: 120530.070
Last Update: May 30, 2012

Dear Programmer,

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

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

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

I look forward to hearing from you.

Best Wishes,

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

Raymond F. Melissa
President

1

Table of Contents

Welcome to Telco SmartSearch.....	1
Non-Published Data	2
Standardization	2
Telco SmartSearch Requests.....	4
Sending a 411 Web Service Request	4
Search Tag Attributes	5
Text Search Fields	8
Range Searches	14
Telco SmartSearch Response.....	15
Parsing the Telco SmartSearch Response	15
Listing Fields	17
Matching Column Indicators	26

1

Welcome to Telco SmartSearch

Melissa Data's Telco SmartSearch provides the ability for qualified customers to do electronic searches (or dips) into our high quality telephone listings database. This service is provided over HTTP or HTTPS and all data exchanged is an XML query and response format.

Melissa Data's Telco SmartSearch appends and verifies information to an existing business or consumer database.

Telco SmartSearch:

- Verifies and appends phone numbers.
- Verifies and appends address information.
- Distinguishes between residential and business listings.
- Retrieves the business or personal name associated with a listing.
- GeoCodes to rooftop level latitude and longitude.

Telco SmartSearch currently offers the following types of searches:

- **Straight Search** - Straight search allows you to do exact queries against the database. The exact criteria you search for will be matched exactly in the database. This is most useful for reverse number lookups, or specific business cases that require exact matching.
- **Smart Search** - Smart search takes the query parameters (name, address, city, state, ZIP Code, etc.) and through a sequence of

matching approaches, tries to find the closest matching record in the current database. For example, if your searching for “John Smith,” on “1234 Oak St,” there may not be an exact match in the database. But, it might find “John Smith” on “1233 Oak St,” which is sufficiently similar to the original search criteria to justify a match.

- **Smart Search Residential** - This search is similar to Smart Search, except that is optimized searching residential lists rather than businesses.

Non-Published Data

Data requested from the service may be flagged as non-published. Non-Published numbers will be returned with only the first name, last name and area code (NPA) populated.

The main content fields of non-published records will be masked (removed) before being returned. These fields include:

- House number
- Street name
- City
- State
- Zip5
- Zip4
- NXX (exchange code)
- Postfix (suffix)

Standardization

Throughout the country, the Regional Bell Operating Companies (RBOCs) in different geographic regions use different standards in describing and representing data. Some areas might use a fully spelled out version of “Avenue,” some areas might use “Av,” and yet others will use “Ave.”

In general, Melissa Data takes the approach of picking an intuitive standard for its data, and translating all known variations into the selected standard. For example, all the variations of Avenue above would be standardized to "Ave."

As the various RBOCs update and change their data, it's always possible that they will introduce new variations on the data. If you find a variation in the data, or something that should be standardized, please contact us with a description of the problem and two examples. Once we have been notified, we can configure our system to standardize the variations.

2

Telco SmartSearch Requests

Sending a 411 Web Service Request

Requests to the 411 Web Service are sent as XML documents via HTTP POST operation to the following URL:

```
http://api.melissadata.net/listingSearch/<MyID>/request/  
jsp/results.jsp
```

<MyID> is the unique account ID supplied by data services to each customer.

Forming the XML Request

The enclosing XML tag object for any request is the <listingsRequest>. Each listingsRequest tag will contain one or more <search> tag objects and each search will contain a <textSearch> tag object, a <rangeSearch> tag object, or both.

This sample shows a basic text search request:

```
<listingsRequest>  
  <search minMatch="0" maxMatch="15" count="10"  
    firstIndex="0">  
    <textSearch>  
      <name />
```

```
<firstName />
<lastName />
<houseNumber>22382</houseNumber>
<street>Avenida Empresa</street>
<city>Rancho Santa Margarita</city>
<state>CA</state>
<zip5>92688</zip5>
<zip4 />
<listingType />
<npa />
<nxx />
<postfix />
<flags />
<searchFlags />
</textSearch>
</search>
</listingsRequest>
```

Search Tag Attributes

The `<search>` tag has several available attributes, or options, that control how the 411 Web Service matches the request to the database and how many records are returned.

This sample shows a search tag with every possible attribute in use:

```
<search minMatch="0" maxMatch="15" Count="10"
  firstIndex="0" querytype="smartSearch">
```

- **minMatch** (required) — This is the minimum number of matches required for the search to successfully return results. If fewer than this number of matches are found, an error will be returned.
- **maxMatch** (required) — This is the maximum number of matches required for the search to successfully return results. If more than this number of matches are found, an error will be returned.
- **Count** (required) — This is the maximum number of records that will be returned from the search. This attribute is used in conjunction with `firstIndex` to gather records in chunks if the caller wants to process the

results in groups. If fewer than this number of records are found, only the available records will be returned.

- **firstIndex** (required) — - This is the zero-based index of the first element in the result set to be returned. This is used in the case where the caller wishes to process a limited number of records at a time. It enables the caller to “get next batch.” If fewer than firstIndex records are returned, an error is returned.

This allows for “pages” of records to be retrieved from the server. For instance, the request for the first would set firstIndex to zero, and count to 10, but the second request would set firstIndex to 10, and count to 10, effectively retrieving records 10 thru 19 (zero based indexing).

- **queryType** — This optional attribute indicates which type of question is to be performed. The possible values are “smartSearch” and “smartSearchResidential.” If this attribute is omitted, then a straight search is performed.

Query Types

- **Straight Search** — Straight search allows you to do exact queries against the database. The exact keywords submitted will be matched against in the database. Note that keywords do have to be found in the order submitted to generate a match. A query for “International Business Machines” would also match “Business Machines International,” if such a listing existed.

This is most useful for reverse number lookups, or specific business cases that require exact matching.

Straight Search is set by not putting the queryType attribute on the Search Field.

- **Smart Search** — Smart search takes the query parameters (name, address, city, state, ZIP, etc) and tries to find the closest matching record in the current database.

For example, if you are searching for “John Smith” on “1234 Oak St,” there may not be an exact match in the database. It might find “John Smith” on “1233 Oak St,” however, which is sufficiently similar to the original search criteria to justify a match.

Smart Search is set by setting the queryType attribute on a Search tag to “smartSearch.”

- **Smart Search Residential** — Smart Search Residential is similar to Smart Search in all other aspects except that it handles the Name input fields differently.

The main difference between Smart Search and Smart Search Residential is that Smart Search combines the last name and first name fields and searches for them together. This leads to some cases where a listing for someone named “Lawrence Summers” could match a listing for “S. Lawrence.”

The residential smart search preserves the first and last name distinctions so as to eliminate this issue.

Smart Search Residential is set by setting the queryType attribute on a Search tag to “smartSearchResidential.”

Wild Cards and Boolean Searches

The 411 Web Service supports wild card characters and boolean operations in searches.

The service supports two possible wild card characters:

- * — matches zero or more instances of any character.
- ? — matches zero or one instance of any character.

Wild card searches must be prefixed with a minimum of two characters. Here is an example of a search with wild cards:

```
<textSearch>  
  <name>Jo* Public</name>  
</textSearch>
```

This search would match “Joe,” “John,” “Jo,” “Jon,” “Joseph,” “Jonathan,” “Josephine” or “Jorge,” among others.

Change the “*” to a “?”

```
<textSearch>  
  <name>Jo? Public</name>  
</textSearch>
```

This search would only match “Joe,” “Jo” or any two or three letter name beginning with the letters “Jo.”

Boolean searches use one of the three following operators:

- **OR** — The search will match fields that contain the values on either side of this operator. “John OR Jane”
- **AND** — The search will match only fields that contain the values on both sides of this operator.
- **NOT** — The search will not match fields that contain the value that follows this operator.

Nesting of boolean operations is supported by the use of matched parentheses. Here is an example of a nested name search:

```
<textSearch>  
  <name>(Public AND (John OR Jane)) AND NOT(Joe)</name>  
</textSearch>
```

This search would match lists with the last name “Public” and a first name of either “John” or “Jane,” while excluding any listings that contain “Joe.”

The last part is unnecessary for straight searches but would work to exclude such listings from the results of smart searches.

Text Search Fields

The follow section lists and defines the available XML tags for a text search. At least one of these fields must be populated for a search to be valid.

name

This tag would contain the text of either the full name of a person or a business name to be used as a search criteria. For a residential listing, this will be last name followed by the first name.

```
<textSearch>  
  <name>Text</name>  
</textSearch>
```

firstName

The first name from a residential listing (or the first names of both people at the address, i.e. (“John & Mary”) to be used as a search criteria. This field is not populated for business listings.

```
<textSearch>  
  <firstName>Text</firstName>  
</textSearch>
```

lastName

The last name from a residential listing to be used as a search criteria. This field is not populated for business listings.

```
<textSearch>  
  <lastName>Text</lastName>  
</textSearch>
```

houseNumber

A text representation of the house or street number to be used as a search criteria. For “123 Main St,” this would be “123.”

```
<textSearch>  
  <houseNumber>Text</houseNumber>  
</textSearch>
```

street

The text of the street to be used as a search criteria.

```
<textSearch>  
  <street>Text</street>  
</textSearch>
```

city

The name of the city to be used as a search criteria.

```
<textSearch>  
  <city>Text</city>  
</textSearch>
```

state

The name of the state to be used as a search criteria.

```
<textSearch>  
  <state>Text</state>  
</textSearch>
```

zip5

The text of the five-digit ZIP Code to be used as a search criteria.

```
<textSearch>  
  <zip5>Text</5>  
</textSearch>
```

zip4

The text of second part of a nine-digit ZIP + 4 code to be used as a search criteria.

```
<textSearch>  
  <zip4>Text</zip4>  
</textSearch>
```

listingType

Indicates business, residential or government as “B,” “R,” and “G,” respectively, as well as combinations of two or three to be used as a search criteria.

Remarks

Some Regional Bell Operating Companies (RBOCs) do not distinguish between government and business listings. These listings will be flagged as businesses.

```
<textSearch>  
  <listingType>Text</listintType>  
</textSearch>
```

npa

The text of the area code portion of the phone number to be used as a search criteria. For the number “949-589-5200,” this would be “949.”

```
<textSearch>  
  <npa>Text</npa>  
</textSearch>
```

nxx

The text of the exchange code portion of the phone number to be used as a search criteria. For the number “949-589-5200,” this would be “589.”

```
<textSearch>  
  <nxx>Text</nxx>  
</textSearch>
```

postfix

The text of the suffix code portion of the phone number to be used as a search criteria. For the number “949-589-5200,” this would be “5200.”

```
<textSearch>  
  <postfix>Text</postfix>  
</textSearch>
```

flags

This field indicates which flags must be returned in the flags field of a record for the current text search to produce a match.

Remarks

See page 25 in the next chapter for list of possible values for this field.

```
<textSearch>  
  <flags>Text</flags>  
</textSearch>
```

searchFlags

This field is used only when the queryType is “smartSearch” or “smartSearchResidential.” These flags indicate which fields must match for a record to be returned.

Remarks

Flags are separated by space, and can include any combination of the following:

- LN - Must match last name
- FN - Must match first name
- HN - Must match house number
- ST - Must match street name
- CM - Must match city

- SM - Must match state
- ZC - Must match ZIP Code
- PL - only published listings are to be returned
- SN - The single best listing is to be returned. If the result contains both published and non-published listings, it will return the first published listing. If the result contains only non-published listings, it returns the first non-published.

```
<textSearch>  
  <searchFlags>Text</searchFlags>  
</textSearch>
```

Range Searches

Range searches can be performed on any numeric field and can be combined with a text search to further limit the results returned for the search.

Range searches are defined by a single tag with the following attributes:

- `startRange` — This is the lowest possible matching value.
- `endRange` — This is the highest possible matching value.
- `type` — This is the field that will be compared to the range defined by the range attributes. The possible values correspond to the field names for the same data:

HOUSE_NUMBER, ZIP5, ZIP4, NPA, NXX, POSTFIX

The following sample would match up to 10 listings for the 100 block of Main St.

```
<listingsRequest>
  <search minMatch="0" maxMatch="15" count="10"
    firstIndex="0">
    <textSearch>
      <street>Main St.</street>
    </textSearch>
    <rangeSearch startRange="100" endRange="199"
      type="HOUSE_NUMBER" />
    </search>
  </listingsRequest>
```

3

Telco SmartSearch Response

Parsing the Telco SmartSearch Response

The results of a query to the Telco SmartSearch service are returned as a single XML document. The document or root element is the `<listingResponse>` tag.

This is an example of a result returned by the Telco SmartSearch:

```
<listingResponse xmlns:xsi="http://www.w3.org/2000/10/
XMLSchema-instance"
xsi:noNamespaceSchemaLocation='metrol-dds-
listings.xsd'>
  <listingSet matchCount="1" DBSearchTime="790"
    TotalSearchTime="790" >
    <listing>
      <name>Andreychuk David </name>
      <firstName>David</firstName>
      <lastName>Andreychuk</lastName>
      <fullAddress>N Sweetzer Ave</fullAddress>
      <houseNumber>102</houseNumber>
      <houseFraction> </houseFraction>
      <preDirectional>N </preDirectional>
      <street>Sweetzer</street>
```

```
<streetType>Ave</streetType>
<postDirectional></postDirectional>
<suiteNo></suiteNo>
<latitude>34.074389</latitude>
<longitude>118.370044</longitude>
<city>Los Angeles</city>
<state>CA</state><zip5>90048 </zip5>
<zip4>3599</zip4>
<category></category>
<npa>323</npa>
<nxx>655</nxx>
<postfix>6223</postfix>
<listingType>R</listingType>
<flags>LP NC NF PA</flags>
<timezone>PST</timezone>
<RBOCEffectiveDate>20030101</RBOCEffectiveDate>
</listing>
</listingSet>
</listingResponse>
```

listingSet

All of the records returned are contained with the `<listingSet>` element. This tag has the following attributes:

- **matchCount** — This is the total number of records that matched the query. This is not necessarily the number of records returned, if the query used the `count` and `firstIndex` attributes to limit the number of records returned.

If the server returns a value of the `maxMatch` attribute plus one, this indicates that there are more matching records in the database than allowed by the `maxMatch` attribute, but does not specify how many more.

- **DBSearchTime** — This is the number of milliseconds that it took for the database engine to complete the search for this query.
- **TotalSearchTime** — This is the number of milliseconds that it took for the database engine to complete the search for this query, plus XML parsing and generation. This attribute can be useful when troubleshooting network latency issues.

Listing Fields

Each record returned is contained within a <listing> element, which will contain each of the fields in the section. Fields without data will be returned as empty elements.

name

This is the full name or business name attached to the listing. For residential records, this will be last name followed by first name.

```
<listing>
  <name>Text</name>
</listing>
```

firstName

This is the first name of the person associated with a residential listing. This element is not populated for business listings.

```
<listing>
  <firstName>Text</firstName>
</listing>
```

lastName

This is the last name of the person associated with a residential listing. This element is not populated for business listings.

```
<listing>
  <lastName>Text</lastName>
</listing>
```

houseNumber

This element returns the house number, minus any fractions.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>  
  <houseNumber>Text</houseNumber>  
</listing>
```

houseFraction

This element returns any fractional portion of the house number, such as “B,” “1/2” or so on.

```
<listing>  
  <houseFraction>Text</houseFraction>  
</listing>
```

preDirectional

This element returns any directional indicator that would precede the street name, such as “N” or “SW.”

```
<listing>  
  <preDirectional>Text</preDirectional>  
</listing>
```

street

The element returns the street name.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>
  <street>Text</street>
</listing>
```

streetType

This element returns the suffix that follows the street name, such as “Ave,” “St” or “Blvd.”

```
<listing>
  <streetType>Text</streetType>
</listing>
```

postDirectional

This element returns any directional indicator that would follow the street name, such as “N” or “SW.”

```
<listing>
  <postDirectional>Text</postDirectional>
</listing>
```

suiteNo

This element returns any suite number from the listing.

```
<listing>
  <suiteNo>Text</suiteNo>
</listing>
```

city

This element returns the name of the city from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>
  <city>Text</city>
</listing>
```

state

This element returns the name of the state from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>
  <state>Text</state>
</listing>
```

zip5

The element returns the five-digit ZIP Code from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>
  <zip5>Text</zip5>
</listing>
```

zip4

This element returns the second part of a nine-digit ZIP + 4 code from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>  
  <zip4>Text</zip4>  
</listing>
```

npa

This element returns the area code portion of the phone number from the listing.

```
<listing>  
  <npa>Text</npa>  
</listing>
```

nxx

This element returns the exchange portion of the phone number from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>  
  <nxx>Text</nxx>  
</listing>
```

postfix

This element returns the final four digits of the phone number from the listing.

Remarks

This element will be masked (blank) in non-published records.

```
<listing>  
  <postfix>Text</postfix>  
</listing>
```

latitude

This element returns the latitude portion of the geographical location of the listing.

Remarks

Geocodes in the resultant data, if present, are roof-top level, with 90% being within 100 feet of the location.

This element will be masked (blank) in non-published records.

```
<listing>  
  <latitude>Text</npa>  
</listing>
```

longitude

This element returns the longitude portion of the geographical location of the listing.

Remarks

Geocodes in the resultant data, if present, are roof-top level, with 90% being within 100 feet of the location.

This element will be masked (blank) in non-published records.

```
<listing>  
  <longitude>Text</longitude>  
</listing>
```

listingType

This element returns a code which indicates business, residential or government with “B,” “R,” or “G,” respectively.

Remarks

Some listings will be both business and residential, typically professionals working from home.

```
<listing>  
  <listingType>Text</listingType>  
</listing>
```

timezone

This element returns the timezone where the listing is located.

Remarks

The possible values for this element are:

- **PST** — Pacific Standard Time
- **CST** — Central Standard Time
- **MST** — Mountain Standard Time
- **EST** — Eastern Standard Time
- **AST** — Atlantic Standard Time
- **KST** — Alaska Standard Time
- **HST** — Hawaii/Samoa Standard Time

```
<listing>  
  <timezone>Text</timezone>  
</listing>
```

RBOCEffectiveDate

This element returns the date when the Regional Bell Operating Company last updated this listing.

Remarks

The date is returned in the format “YYYYMMDD.”

```
<listing>  
  <RBOCEffectiveDate>Text</RBOCEffectiveDate>  
</listing>
```

flags

This element returns a space-delimited list of codes that indicate the type of listing.

Remarks

The possible flags returned by this element are listed below:

- NP - non-published
- LP - listed and published record
- OA - omit address
- PA - populated address (mutually exclusive with OA)
- FX - fax line
- NF - non-fax line (mutually exclusive with FX)
- CH - children's line (only populated for residential listings)
- NC - non-children's line (only populated for residential listings, mutually exclusive with CH)
- CS - Cellular Services, indicates a wireless number
- PS - Paging Services, indicates a pager number

```
<listing>  
  <flags>Text</flags>  
</listing>
```

category

This element is reserved for future use.

Matching Column Indicators

When a smart search is executed, matching column information is returned as part of the XML response to indicate which query fields matched, and how well they matched, rated on a scale from 1 to 10. The results will look something like this:

```
<matchedColumns>  
  <columnName match="10">name </columnName>  
  <columnName match="10">firstName </columnName>  
  <columnName match="10">lastName </columnName>  
  <columnName match="10">state </columnName>  
  <columnName match="9">street </columnName>  
</matchedColumns>
```