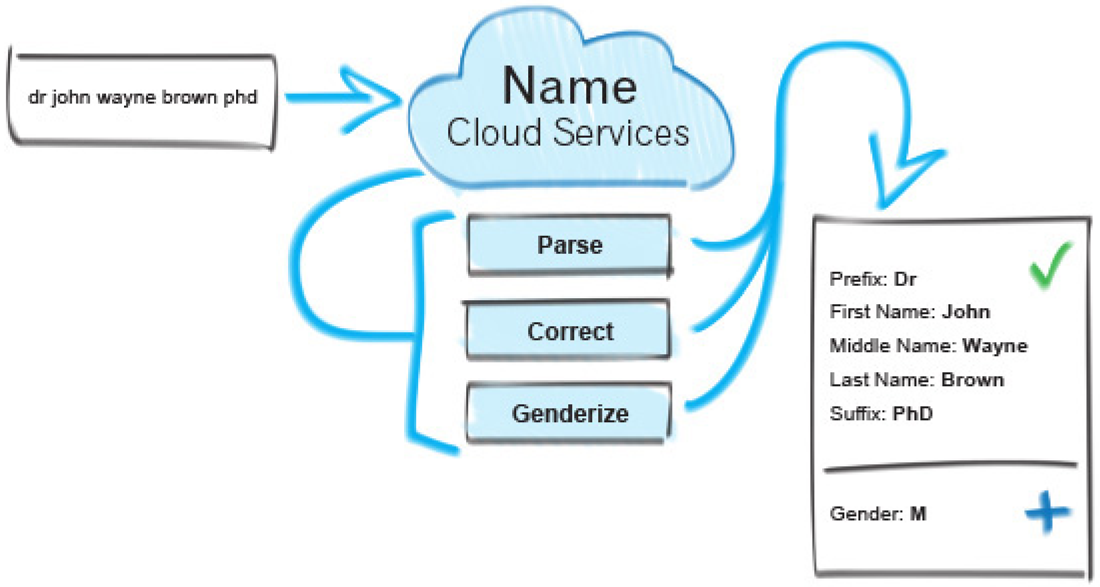
Global Name Web Service: Programmer's Quick Start

Overview

The Global Name service features real-time name parsing and gender identification allows for a cleansed, standardized contact database, which can help improve marketing campaigns according to gender, and minimize waste on vulgar/suspicious names.

The Global Name Web Service gives you the ability to parse full and multiple names, standardize common company indicators, detect dual names, identify gender, and flag vulgar/suspicious names.



You can use Global Name to:

- Recognize over 530,000 last names and 132,000 first names
- Flag inappropriate words
- Detect company names accidentally entered as a full name
- Add casing for company names

Global Name has the ability to detect:

- Vulgar or suspicious names, such as Mickey Mouse.
- Gender based off the first name
- Split multiple names
- Parse names out into their components (prefix, first, middle, last, and suffix)
- Recognize common German prefixes and suffixes

FIELDS INPUT AND OUTPUT FROM THE SERVICE

INPUTs	Description
Transmission Reference	A unique string value identifying the request
Customer ID	License String from Melissa Data
Options (CorrectFirstName, NameHint, GenderPopulation, GenderAggression, MiddleNameLogic)	Control Options for the Web Service (For more detailed explanation of each, refer to the Global Name Options Wiki Page)
Record ID	Unique ID if processing multiple records
Company	Company name to be standardized (Optional; Required if FullName is not inputted)
FullName	Full name to be genderized, standardized, and parsed (Optional; Required if Company is not inputted)

OUTPUTs	Description
Transmission Reference	A unique string value identifying the request
Transmission Results	Results of the transmission (success/fail)
Record ID	Unique ID if processing multiple records
Results	Results (status of validation)
Company	Standardized company name
NamePrefix	Prefix
NameFirst	First name
NameMiddle	Middle name
NameLast	Last name
NameSuffix	Suffix
Gender	Gender (M = Male, F = Female, U = Unknown first name or no first name present, N = Neutral)
NamePrefix2	Prefix of second name if multiple names entered
NameFirst2	First name of second name if multiple names entered
NameMiddle2	Middle name of second name if multiple names entered
NameLast2	Last name of second name if multiple names entered
NameSuffix2	Suffix of second name if multiple names entered
Gender2	Gender of second name if multiple names

	entered (M = Male, F = Female, U = Unknown first name or no first name present, N = Neutral)

License String

You should have been provided an encrypted and unique license string or Customer ID from Melissa Data. It is necessary to include this string with each request to the Global Name Web Service. This value should be put into the CustomerID element in each Web Service request.

If you do not have a license string, please contact your Melissa Data sales representative at 1-800-MELISSA (1-800-635-4772).

Sample REST Requests

https://globalname.melissadata.net/v3/WEB/GlobalName/doGlobalName?t=GlobalNamelV3_REST&id=XXXXXXXXX&opt=CORRECTFIRSTNAME:TRUE,NAMEHINT:VARYING,GENDERPOPULATION:MIXED,GENDERAGGRESSION:NEUTRAL,MIDDLENAMELOGIC:PARSELOGIC&comp= 志冰 申&full=Dr. Jane doe and jon smith sr%20&format=JSON

https://globalname.melissadata.net/v3/WEB/GlobalName/doGlobalName?t=GlobalNamelV3_REST&id=XXXXXXXXX&opt=CORRECTFIRSTNAME:TRUE,NAMEHINT:VARYING,GENDERPOPULATION:MIXED,GENDERAGGRESSION:NEUTRAL,MIDDLENAMELOGIC:PARSELOGIC&comp= 志冰 申&full=Dr. Jane doe and jon smith sr%20&format=XML

Sample JSON Response

{"Version":"3.0.0.12","TransmissionReference":"GlobalNamelV3_REST","TransmissionResults":"
","TotalRecords":"1","Records":[{"RecordID":"1","Results":"NE08,NS01,NS05,NS06,NS07,NS08","Company":"å¿—å†° ç”³","NamePrefix":"Dr.","NameFirst":"Jane","NameMiddle":"
","NameLast":"Doe","NameSuffix":" ","Gender":"F","NamePrefix2":"
","NameFirst2":"Jon","NameMiddle2":" ","NameLast2":"Smith","NameSuffix2":"Sr.","Gender2":"M"}]}

Sample XML Response

```
<Response>
	<Version>3.0.0.12</Version>
```

```
    <TransmissionReference>GlobalNamelV3_REST</TransmissionReference>
    <TransmissionResults> </TransmissionResults>
    <TotalRecords>1</TotalRecords>
    <Records>
        <ResponseRecord>
            <RecordID>1</RecordID>
            <Results>NE08,NS01,NS05,NS06,NS07,NS08</Results>
            <Company>志冰 申</Company>
            <NamePrefix>Dr.</NamePrefix>
            <NameFirst>Jane</NameFirst>
            <NameMiddle> </NameMiddle>
            <NameLast>Doe</NameLast>
            <NameSuffix> </NameSuffix>
            <Gender>F</Gender>
            <NamePrefix2> </NamePrefix2>
            <NameFirst2>Jon</NameFirst2>
            <NameMiddle2></NameMiddle2>
            <NameLast2>Smith</NameLast2>
            <NameSuffix2>Sr.</NameSuffix2>
            <Gender2>M</Gender2>
        </ResponseRecord>
    </Records>
</Response>
```

Single vs. Batch

Melissa Data's cloud services are capable of both single record real-time processing and batch processing. The difference is simply in the number of records sent in each request. Melissa Data cloud services take an array of records. This array can contain a single record or 100 records. For a real-time process like a Web form entry or a call center application, send in a request with one record. For a batch processing scenario like a database, send requests of up to 100 records until all the records are processed. Note: Make sure each record in the request has a unique Record ID.

Sample Batch XML Request

```
<Request xmlns="urn:mdGlobalName">
    <TransmissionReference>Batch Sample</TransmissionReference>
    <CustomerID>XXXXXXXXX</CustomerID>
    <Options>CORRECTFIRSTNAME:TRUE,NAMEHINT:VARYING</Options>
    <Records>
        <RequestRecord>
            <RecordID>1</RecordID>
            <Company>The Company Inc.</Company>
            <FullName>Jane Smith, M.D.</FullName>
        </RequestRecord>
        <RequestRecord>
```

```
                        <RecordID>2</RecordID>
                        <Company></Company>
                        <FullName>Jim philip smith</FullName>
                </RequestRecord>
                ...
                <RequestRecord>
                        <RecordID>100</RecordID>
                        <Company>Melissa Data corporation</Company>
                        <FullName>John Doe</FullName>
                </RequestRecord>
        </Records>
</Request>
```

Global Name Web Service URLS

Cloud Service Endpoint URLs

Choosing a Web Service Protocol

The Melissa Data Global Name Service supports REST, JSON, XML, and SOAP. For the undecided, here are some Pros and Cons of one Web Service protocol over the other.

REST

Pros: REST is lightweight and relies upon HTTP to do its work. If you don't need a strict API definition, this is the way to go. REST is also format-agnostic so you can use XML or JSON as response formats.

Cons: REST can only be used to send a single record and doesn't support strict contracts or more involved security. The Response is an XML or JSON document.

XML

Pros: XML allows record set structures of more than one record at a time and has very good support with most languages and browsers. Supports namespaces.

Cons: Developers need to use tools to initialize/de-serialize the XML structure.

JSON

Pros: JSON relies on simple object serialization based on JavaScript's object initialization. IT is very simple to use with JavaScript and is easily parsed and understood by developers.

Cons: No support for formal definitions. No namespace support. Not much support in Web Service clients with some platforms.

SOAP

Pros: Soap (using a WSDL) is a heavy-weight XML standard that is centered around document passing. The advantage with this is that your requests and responses can be very well structured.

Cons: SOAP documents are very verbose and hard to consume without a SOAP toolkit and generally carry more overhead.

Basic Order of Operations (Pseudo Code)

1. Choose SOAP, XML, or the REST protocol.
2. Create an instance of the request object.
3. Populate the request element 'Customer ID' with your Product License.
4. Set the desired options for the Web Service
5. Add input company name and full name to the <Records> array with anywhere from 1 to 100 <RequestRecord> items. (SOAP, XML)
6. Call the method and pass the request to the service using the SOAP endpoint for SOAP requests and the WEB endpoint for XML or JSON requests.
7. Examine and parse the response from the reply object back from the service.
8. Interpret the results.

Interpreting Results

Melissa Data's Global Phone Service uses Results Codes to determine if a phone number was or was not matched with one in our database. The Melissa Data Cloud Services use the following Results conventions:

1. CLOUD SERVICE ERRORS: SExx
2. CLOUD TRANSMISSION ERRORS: GExx
3. NAME STATUS CODES: NSxx
4. NAME ERROR CODES: NExx

For example: An NS01 result code indicates that the name was successfully parsed and an NS02 indicates there was an error parsing the number.

NE01 – NE08 and NE99 Result Codes can indicate what specific part of the name has been detected as having an issue. Please check the documentation for any additional information on Results.

Sample Code

Fully working examples are available on the Melissa Data Wiki pages:
Click here to go to the Global Name Web Service Wiki Page

Wiki Page

A product support Wiki is available for your convenience. In the Wiki, you will find documentation about the service in more detail.
Click here to go to the Global Name Web Service Wiki Page

Misc. Considerations

Firewall

If you are behind a firewall, you may need to allow specific IP addresses access in order to communicate with the service. For a full list of IP Addresses, see IP Address information.

Result Codes

The service returns a series of results codes to tell you the status of the Name and any changes or errors found during the verification process.

For a full list of the result codes returned by Global Name, see Global Name Results Codes.