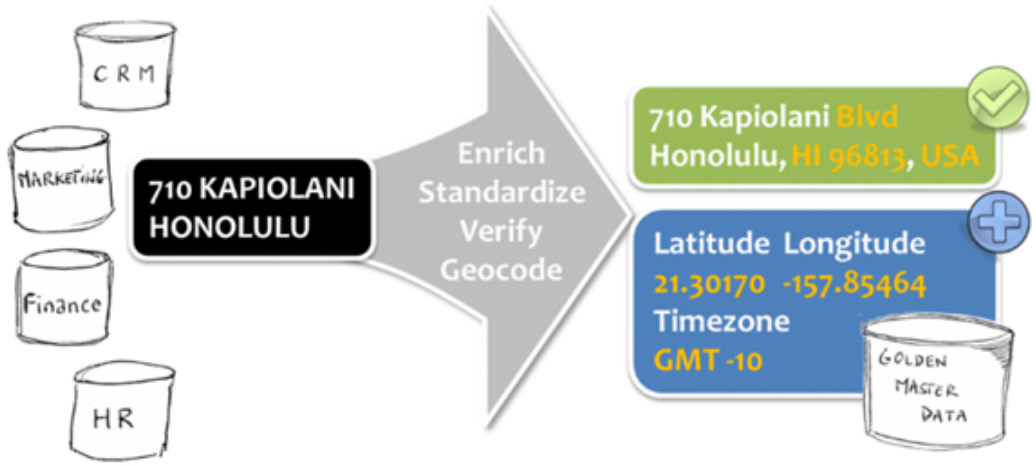
Address Standardization

Quick Guide: CPM Address Correction & Standardization (CPM ACS) API

1. Overview

Verifying and geocoding address information is a critical activity for consistent business operations throughout the enterprise. Accurate and valid address information guarantees lower mailing, shipping and billing costs as well as better customer relations (sales, support, etc.) and experience.

In the context of Master Data Management, address information represents the most important piece of information in a variety of domains such as *Customers*, *Suppliers*, *Sites*, *Locations*, *Parties*, *Citizens*, etc. It is a prerequisite for accurate duplicate detection.



Business Case

Build capabilities to

1) Standardize heterogeneous address data from various sources before it gets into CPM in an effort to maintain clean and reliable address information.

2) Continuously keep the address information clean and reliable over day to day maintenance cycles it would go through.

The scope right now is limited to address standardization and not address verification and/or correction even though usage of google API implicitly provides some of these functions as well.

Solution

Google geocoding API was found to be a good match to build the above capabilities. While it provides strong foundational capabilities, some level of custom pre and post processing rules have to be built to make it work for our specific ask and data variety.

2. Deployed and availability:

CPM ACS API is built on a .NET application server to perform address standardization and address correction

CPM ACS API is deployed in CSI DEV & PROD servers.

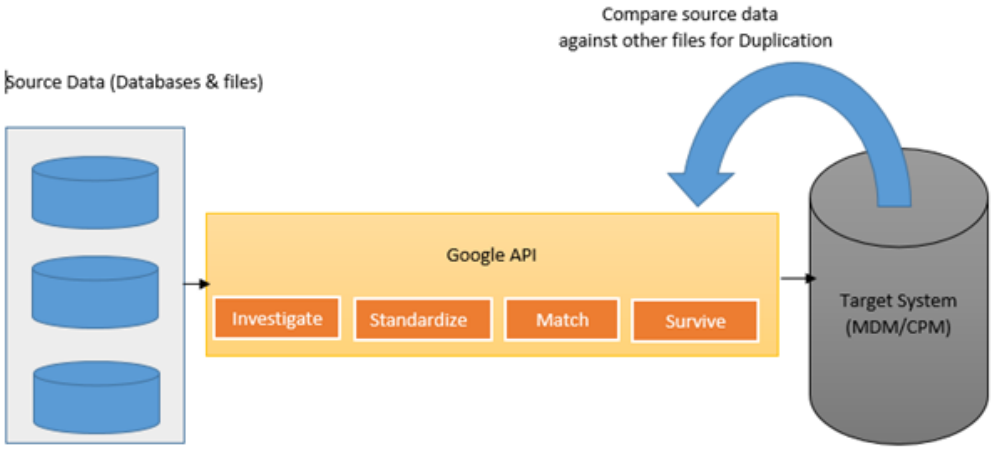
CSI DEV URL:

http://142.122.222.233:81/GetStandardizedAddress/MDMStandardizeAddressService.asmx

CSI PROD URL:

http://TOROONDC27X-039.on.bell.ca/GetStandardizedAddress/MDMStandardizeAddressService.asmx

3. CPM Flow:



4. Request and Response formats example in CPM

Sample Input Request

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap="http://MDMQSService.MDMQS.isd.ibm.com/soapoverhttp/">

<soapenv:Header/>

<soapenv:Body>

<soap:standardizeAddress>

<soap:addresslineone>16 brookers lane toronto</soap:addresslineone>

<soap:addresslinetwo>unit 1806</soap:addresslinetwo>

<soap:addresslinethree/>

<soap:city>etobicoke</soap:city>

<soap:state>ON</soap:state>

<soap:postalcode>M8V0A5</soap:postalcode>

<soap:country>CA</soap:country>

</soap:standardizeAddress>

</soapenv:Body>

</soapenv:Envelope>

Sample output Response received from CPM ACS to MDM:

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Body>

<standardizeAddressResponse xmlns="http://MDMQSService.MDMQS.isd.ibm.com/soapoverhttp/">

<standardizeAddressResult>

<standardizeIndicator>Y</standardizeIndicator>

<addresslineone_formatted>16 BROOKERS LANE</addresslineone_formatted>

<addresslinetwo_formatted></addresslinetwo_formatted>

<addresslinethree_formatted/>

```
        <residencenumber_formatted>UNIT 1806</residencenumber_formatted>
        <subcityname_formatted>ETOBICOKE</subcityname_formatted>
        <cityname_formatted>TORONTO</cityname_formatted>
        <stateabbreviation_formatted>ONTARIO</stateabbreviation_formatted>
        <statetype_formatted>108</statetype_formatted>
        <fullpostalcode_formatted>M8V0A5</fullpostalcode_formatted>
        <streetnumber_formatted>16</streetnumber_formatted>
        <streetname_formatted>BROOKERS LANE</streetname_formatted>
        <countryname_formatted>CANADA</countryname_formatted>
        <countrytype_formatted>31</countrytype_formatted>
      </standardizeAddressResult>
    </standardizeAddressResponse>
  </soap:Body>
</soap:Envelope>
```


5. Standardization rules in CPM ACS

Below are the set of rules followed **before** making the Google Geocode API call to get back a proper response from google.

1. In the request xml in the beginning if there are numbers, followed by string in address line 1, address line 2 & address line 3 then the string will become the input parameter for google API call.

2. In the request xml, if the strings “UNIT”, “ APT”, “SUITE” appears in address line 2 & address line 3 followed by numbers, it will be concatenated to input parameter for google API call.

3.In the ACS request xml,

If below values are in address line 1, address line 2 and address line 3, then remove them

- City name,
- State,
- Country

4. City, State and Country values from ACS request are concatenated to the input parameter before making the google API call

5. In ACS request xml, rest of the strings in address line 1, address line 2 & address line 3 are ignored for google API call.

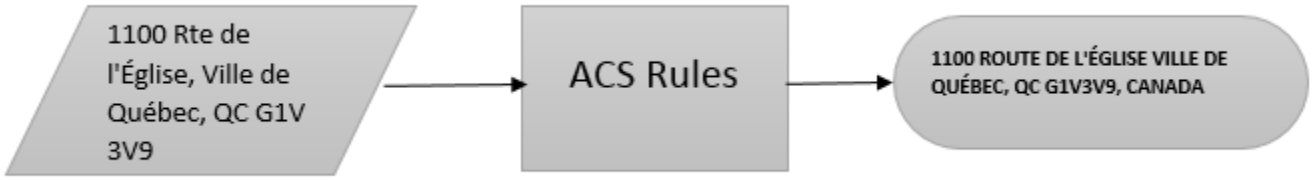
6. Postal code is not passed to google API. But the postal code passed in ACS request xml is validated against CANADA and US postal code standards. If found valid, the input postal code will take preference over google returned postal code. For postal code validations Refer to https://www.canadapost.ca/tools/pg/manual/PGaddress-e.pdf

7. Extra spaces are removed in Postal code and all other fields.

8. UTF-8 encoding is used handle all character set including French.

9. ACS request and response xml response are stored in internal Database so as to avoid multiple calls to google API for the same address

10. ACS handles french address and does all the above mentioned validations.



Once the response is received from Google Geo code API, it goes through the below standardization rules to form the ACS response XML

1. "longest common substring" (LCS) algorithm is used to determine the standardizeIndicator with "Y" or "N". This algorithm returns the longest string which is common to both input string and output string.
   Example:

   ```
   " coles supermarket " and " coles ", LCS = " coles ", length 7
   " coles supermarket " and " dandenong markets ", LCS = "market", length 6
   ```

   Effectively, this process bumps up the length of an LCS by two characters if both input strings match with word breaks, rather than a portion of a word.

   In the full address supplied in ACS input and google api output, address is split to
   a. Street number and Street name
   b. City, Province and Country

   If match(Streetnumber + Streetname) > 50 % **AND** match (City + Province + Country) > 50% then

   standardizeIndicator= "Y"

   Else

   standardizeIndicator = "N"
2. All the characters are converted to upper case.
3. If postal code is not passed in the ACS request, pick the one google provides. If postal code is passed and is valid, hardcode the same in the ACS response.
4. For any string – FLOOR alias FLR details is set to addresslinethree_formatted field.

5. For any string UNIT, APT, SUITE details are set into residencenumber_formatted field.
6. For any string with company names or other strings is set to addresslinetwo_formatted
7. MDM code look up types for Country & State value is set to countrytype_formatted & statetype_formatted
8. AddressLine two and addressLine three are compared with addressLine one, if the match percentage is >60, then AddressLine two / three values are removed from the response
9. Street Number, Street Name are captured.
10. Look up table for Street Types is maintained and logic to capture Street Suffix from the response
    Refer to Street Type look up table in https://www.canadapost.ca/tools/pg/manual/PGaddress-e.pdf
11. Streetsuffix are set in predirectional_formatted or postdirectional_formatted fields

    if EAST, WEST, NORTH, SOUTH strings are found in Streetsuffix, then it is removed and set to predirectional_formatted or postdirectional_formatted fields

    example 1: (if NORTH comes after street name)

    Street Suffix = VICTORIA NORTH

    new format :

    Street Suffix = VICTORIA

    postdirectional_formatted = NORTH

    example 2: (if NORTH comes before street name)

    Street Suffix = NORTH VICTORIA

    new format : Street Suffix = VICTORIA

    predirectional_formatted = NORTH
12. If StreetType / StreetSuffix values (short name & long name) are present in addressLine 2 or addressLine 3, then those strings are removed
13. City Name & Sub-city Name are captured.
14. In the ACS response, City, State and Country values are removed from address Line 1, address Line 2 and address Line 3
15. Post standardization rules are applied even on the **non-standardized address with Indicator "N"** so as to keep the data clean (after calling google api and if zero results returned)
16. Google api returns the short names for every address field, however CPM only picks and stores the full name
    <long_name>Saint Georges Avenue</long_name>
    <short_name>St Georges Ave</short_name>

Note:

if <standardizeIndicator>**Y**</standardizeIndicator>

Address is valid and in good format

if <standardizeIndicator>**N**</standardizeIndicator>

Address is not valid or the format is incorrect.

6. MDM ACS for Address Standardization

In MDM, OOTB Address Standardizer class is extended to have the custom logic to invoke this CPM ACS API instead of calling Quality Stage.

It's a configuration switch between CPM ACS & QS. Execute the below SQLs to enable MDM to use CPM ACS API.

MDM will pass the address object in a web-service call (SOAP call) to CPM ACS application and will receive SOAP xml response with the standardization indicator as 'Y' or 'N'

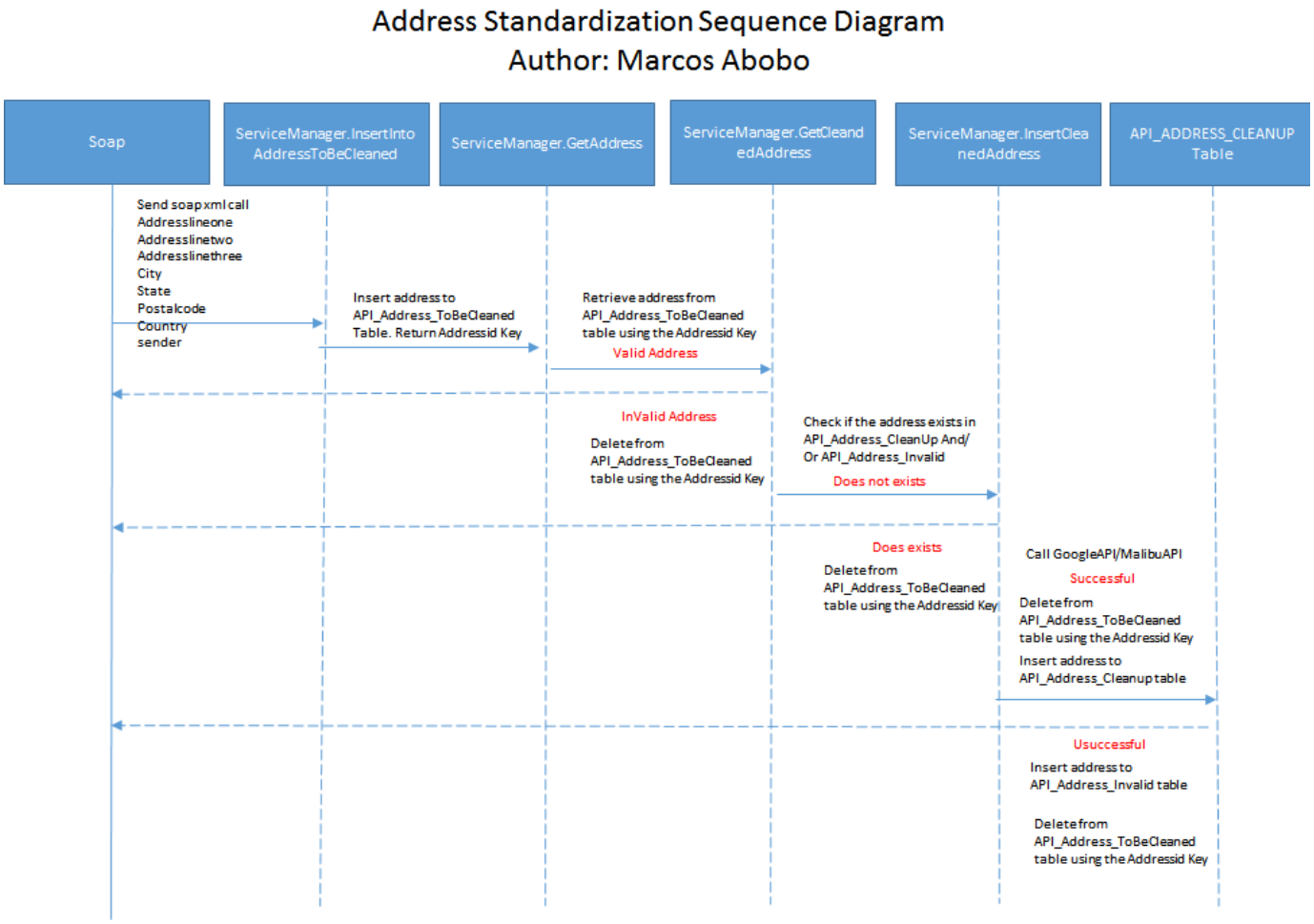
MDM Configuration SQLs:

update CONFIGELEMENT **set value**='com.bbm.mdm.standardizer.standardization.InfoServerStandardizerAdapterExt',last_update_dt=current_timestamp **where** name='/IBM/Party/Standardizer/Address/className';

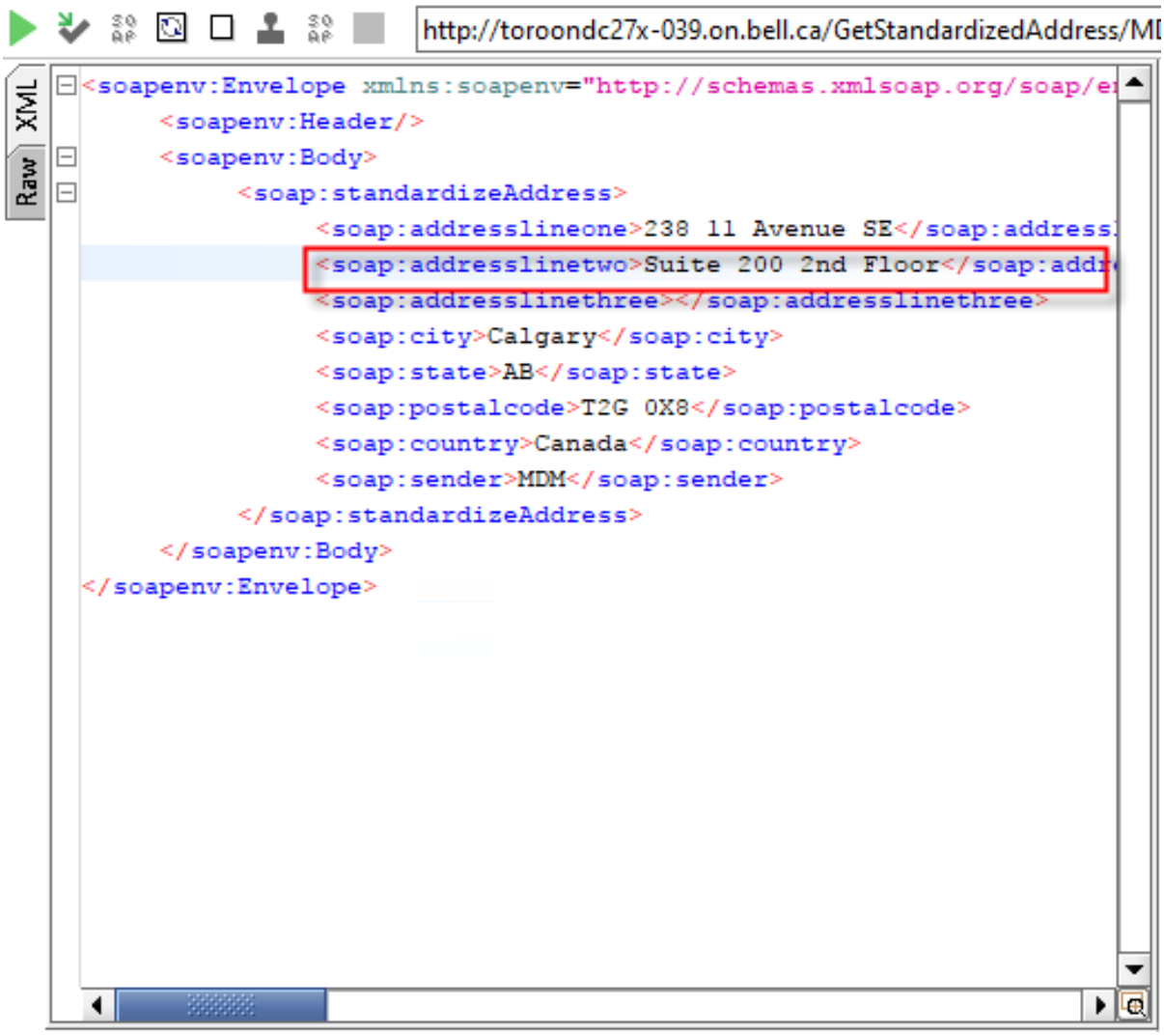
Insert **into** CONFIGELEMENT **values** (1000849, 1000,'/CPM/AddressCorrection/Enabled','true','false',**null**,current_timestamp,'mdmadmin',**null**,**null**);

Insert **into** CONFIGELEMENT **values** (1000850, 1000,'/CPM/AddressCorrection/URL',**null**,'http://142.122.222.233:81/GetStandardizedAddress/MDMStandardizeAddressService.asmx',**null**,current_timestamp,'mdmadmin',**null**,**null**);

Sequence Diagram:



Example of Soap XML call:


http://toroondc27x-039.on.bell.ca/GetStandardizedAddress/MI
XML
Raw
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/e
<soapenv:Header/>
<soapenv:Body>
<soap:standardizeAddress>
<soap:addresslineone>238 11 Avenue SE</soap:address
<soap:addresslinetwo>Suite 200 2nd Floor</soap:addi
<soap:addresslinethree></soap:addresslinethree>
<soap:city>Calgary</soap:city>
<soap:state>AB</soap:state>
<soap:postalcode>T2G 0X8</soap:postalcode>
<soap:country>Canada</soap:country>
<soap:sender>MDM</soap:sender>
</soap:standardizeAddress>
</soapenv:Body>
</soapenv:Envelope>