Confirm®

Import Address Agent Specification
v19.20c.AM
• Proj4js, version 1+, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://trac.osgeo.org/proj4js/.

• requireJS, version 2.1.2, which is licensed under the MIT License or the New BSD License. The license can be downloaded from https://github.com/jrburke/requirejs/blob/master/LICENSE. The source code for this software is available from http://requirejs.org/.

• Apache Cordova, version 8.1.2, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://phonegap.com/download/.

July 16, 2019
# Table of Contents

**Specifications**

<table>
<thead>
<tr>
<th>Import Address Specifications</th>
<th>6</th>
</tr>
</thead>
</table>

Specifications

The following sections outline all the Specifications that exist within the Confirm functionality.

In this section

Import Address Specifications 6
Import Address Specifications

Introduction

The purpose of this document is to describe the specifications of the files for importing address records into Confirm, using 'Import Address Agent'.

The 'Import Address Agent' imports records in XML format.

A comma delimited (.csv) file can also be imported, but it needs to be transformed into XML format readable by Confirm, using a stylesheet.

The following table details the XML format applicable to this type of address import:

<table>
<thead>
<tr>
<th>XML Tag</th>
<th>Type</th>
<th>Size</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropertyRef</td>
<td>Character</td>
<td>20</td>
<td>Mandatory - A unique reference, up to 20 characters.</td>
</tr>
<tr>
<td>SiteCode</td>
<td>Character</td>
<td>10</td>
<td>Optional</td>
</tr>
<tr>
<td>SiteAddress</td>
<td>Character</td>
<td>2000</td>
<td>Optional - Truncated, if there are more than 2000 characters.</td>
</tr>
<tr>
<td>SubAddress</td>
<td>Character</td>
<td>60</td>
<td>Optional - Truncated, if there are more than 60 characters.</td>
</tr>
<tr>
<td>PropertyAddress</td>
<td>Character</td>
<td>60</td>
<td>Mandatory - Truncated, if there are more than 60 characters.</td>
</tr>
<tr>
<td>AppendSite</td>
<td>Boolean</td>
<td>True/False</td>
<td>Optional - Defaults to False, if left blank.</td>
</tr>
<tr>
<td>Locality</td>
<td>Locality</td>
<td>6,0</td>
<td>Optional. Inherits the locality of Site, if left blank. Refer to the Note below the table.</td>
</tr>
<tr>
<td>LocalityId</td>
<td>Number</td>
<td>6,0</td>
<td></td>
</tr>
<tr>
<td>PostCode</td>
<td>Character</td>
<td>10</td>
<td>Optional</td>
</tr>
<tr>
<td>AddressEastIng</td>
<td>Number</td>
<td>10,2</td>
<td>Optional</td>
</tr>
<tr>
<td>AddressNorthing</td>
<td>Number</td>
<td>10,2</td>
<td>Optional</td>
</tr>
<tr>
<td>GazTypeCode</td>
<td>Character</td>
<td>4</td>
<td>Optional - Refer to the rules below the table.</td>
</tr>
<tr>
<td>CoordTolCode</td>
<td>Character</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Boolean</td>
<td>True/False</td>
<td>Optional - Use True to delete record. Defaults to False, if left blank.</td>
</tr>
</tbody>
</table>

Locality

<table>
<thead>
<tr>
<th>XML Tag</th>
<th>Type</th>
<th>Size</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Number</td>
<td>6,0</td>
<td>Provide either Key or Name.</td>
</tr>
<tr>
<td>Name</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>
### Town

<table>
<thead>
<tr>
<th>XML Tag</th>
<th>Type</th>
<th>Size</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Number</td>
<td>6,0</td>
<td>Provide either Key or Name.</td>
</tr>
<tr>
<td>Name</td>
<td>Character</td>
<td>35</td>
<td>Provide either Key or Name.</td>
</tr>
<tr>
<td>CountyId</td>
<td>Number</td>
<td>6,0</td>
<td>Provide either CountyId or County.</td>
</tr>
<tr>
<td>County</td>
<td>County</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### County

<table>
<thead>
<tr>
<th>XML Tag</th>
<th>Type</th>
<th>Size</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Number</td>
<td>6,0</td>
<td>Provide either Key or Name.</td>
</tr>
<tr>
<td>Name</td>
<td>Character</td>
<td>35</td>
<td>Provide either Key or Name.</td>
</tr>
</tbody>
</table>

**Note:** Provide only one XML tag, either `<LocalityId>` or `<Locality>.

If `<LocalityId>` is provided, the import will search for an existing locality in the system. If `<Locality>` is provided, the import will search for the unique combination of the specified locality, town and county in the system. If this combination is not found, then the missing data will be created in order to obtain this combination.

- `<Locality>` needs to be in one of the following formats:
  1. `<TownId>` is provided, to uniquely identify (or create) locality.

```xml
<Locality>
  <Name>LOCALITY NAME</Name>
  <TownId>1</TownId>
</Locality>
```

  2. A combination of `<County>` and `<Town>` is provided, to uniquely identify (or create) locality.

```xml
<Locality>
  <Name>LOCALITY NAME</Name>
  <Town>
    <Name>TOWN NAME</Name>
    <County>
      <Name>COUNTY NAME</Name>
    </County>
  </Town>
</Locality>
```

### Rules for Gazetteer and Coordinate Tolerance:

- If no gazetteer type code is supplied, then the system setting default will be applied to the address.
- If the system setting default is applied, then the corresponding coordinate tolerance code for the system setting default will be set against the address unless the coordinate tolerance code is explicitly supplied.
- If the coordinate tolerance code supplied is not present in Confirm, then the corresponding coordinate tolerance code for the specified gazetteer type will be used instead.
Below is an example to illustrate the transformation process to be followed for importing delimited (.csv) files into Confirm.

Sample: Import File
This is a sample .csv file to import addresses into Confirm:

```
```

Sample: Stylesheet
This is the stylesheet to be supplied to transform the .csv file to a format readable by Confirm:

```xml
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes"/>
    <xsl:template match="*">
        <xsl:copy>
            <xsl:apply-templates select="table"/>
        </xsl:copy>
    </xsl:template>
    <xsl:template match="table">
        <Entities>
            <xsl:for-each select="tr">
                <CentralProperty>
                    <PropertyRef>
                        <xsl:value-of select="td[2]"/>
                    </PropertyRef>
                    <SiteCode>
                        <xsl:value-of select="td[3]"/>
                    </SiteCode>
                    <SiteAddress>
                        <xsl:value-of select="td[4]"/>
                    </SiteAddress>
                    <SubAddress>
                        <xsl:value-of select="td[5]"/>
                    </SubAddress>
                    <PropertyAddress>
                        <xsl:value-of select="td[6]"/>
                    </PropertyAddress>
                    <xsl:if test="TD[7]!=''">
                        <AppendSite>
                            <xsl:value-of select="td[7]"/>
                        </AppendSite>
                    </xsl:if>
                    <Locality>
                        <Name>
                            <xsl:value-of select="td[8]"/>
                        </Name>
                    </Locality>
                    <Town>
                        <Name>
                            <xsl:value-of select="td[9]"/>
                        </Name>
                    </Town>
                    <County>
                        <Name>
                            <xsl:value-of select="td[10]"/>
                        </Name>
                    </County>
                </CentralProperty>
            </xsl:for-each>
        </Entities>
    </xsl:template>
</xsl:stylesheet>
```
Sample: Transformed XML

Confirm internally generates an .xml file and saves it in the database:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Entities>
    <CentralProperty>
        <PropertyRef>AP101A8H5R25DT7085</PropertyRef>
        <SiteCode>47700365</SiteCode>
        <SiteAddress>SITE ADDRESS</SiteAddress>
        <SubAddress>SUB ADDRESS</SubAddress>
        <PropertyAddress>PROPERTY ADDRESS</PropertyAddress>
        <AppendSite>True</AppendSite>
        <Locality>
            <Name>LOCALITY NAME</Name>
            <Town>
                <Name>TOWN NAME</Name>
                <County>
                    <Name>COUNTY NAME</Name>
                </County>
            </Town>
        </Locality>
        <PostCode>POST CODE</PostCode>
        <AddressEasting>274426.40</AddressEasting>
        <AddressNorthing>192847.40</AddressNorthing>
        <GazTypeCode>ADD</GazTypeCode>
        <CoordTolCode>NA</CoordTolCode>
    </CentralProperty>
</Entities>
```