

## 3.1.11 Get Customer Details Web Service - CENSSAWEBSVCLIB.GET\_CUST\_INFO\_XML

This web service is available as of AA version 6.5.1

ACGI has developed a web service that allows permitted third-party sites to gather customer details about a given customer but does not require an ODBC connection.

### Web service Details

The "get customer information" web service with internal IP white list validation of the referring site(s) can be referenced as http or https. ACGI recommends implementing this with SSL to avoid risk of transferring plain text or cost and complexity of encryption. ACGI also recommends only requesting the necessary data, to avoid unnecessary overhead. The example below shows how to pull all currently supported data, but removing any of the child elements of <details> will prevent that type of data from being returned.

See APPENDIX A for Error Conditions and Error Codes

### ***censsawebstvclib.get\_cust\_info\_xml* passing arguments as an xml document with name "p\_input\_xml\_doc"**

#### **Input Document Changes**

Added ability to filter by directoryId in AA 7.1.1 release

Added bio contents in AA 7.1.6 release

Added companyAdmins in AA 7.3 release; It is applicable only for "Company" customer types.

Added employmentAttributes in AA 7.3.1 release

Added directory list for addresses, customer create date, aliases and bulkRequest in AA 7.4.1 release. There is a section on bulkRequest near the bottom of the page that explains it in more detail.

Added certifications in AA7.5.

Added dimensioned attributes in AA7.5.2

Add company employees in AA8.0.6

Added files & referralInfo in AA8.1.2

#### **Input Document Example (including all data)**

```
<?xml version="1.0" encoding="UTF-8"?>
<custInfoRequest>
  <custId></custId><!-- this element can repeat if bulkRequest is true.
see bulk request details section -->
  <integratorUsername></integratorUsername>
  <integratorPassword></integratorPassword>
  <directoryId></directoryId> <!-- if provided, returned data will
respect customer preferences for indicated directory ID -->
  <bulkRequest>true</bulkRequest><!-- true/false - if "true", multiple
customers' info can be requested at once. see bulk request details
```

section -->

```
<details includeCodeValues="true">
  <roles include="true" />
  <addresses include="true" includeBad="true" />
  <phones include="true" />
  <emails include="true" includeBad="true" />
  <websites include="true" includeBad="true" />
  <jobs include="true" includeInactive="true" />
  <employmentAttributes include="true" includeAll="true">
```

<!--

To pull employment attributes, you must include jobs in your request.  
The following are examples of querying attributes by type or

type+code.

Do not include the attr children if includeAll is set to "true"

```
<attr type="GENDER" />
```

```
<attr type="EDUCATION" code="COLLEGE" />
```

-->

```
</employmentAttributes>
```

```
<committeePositions include="true" includeInactive="true" />
```

```
<memberships include="true" includeInactive="true" />
```

```
  <subscriptions include="true" includeExpired="true" />
```

```
<communicationPreferences include="true" />
```

```
<customerAttributes include="true" includeAll="true">
```

<!--

The following are examples of querying attributes by type or

type+code.

Do not include the attr children if includeAll is set to "true"

```
<attr type="GENDER" />
```

```
<attr type="EDUCATION" code="COLLEGE" />
```

-->

```
</customerAttributes>
```

```
<custDimAttrs include="true" includeAll="true">
```

<!--

The following are examples of querying attributes by type or

type+code.

Do not include the attr children if includeAll is set to

"true"

```
<attr type="VOLUME" dimensionType="YEAR"
```

```
dimensionType="2015" />
```

```
<attr type="EDUCATION" code="COLLEGE" />
```

-->

```
</custDimAttrs>
```

```
<bio include="true" />
```

```
<aliases include="true" />
```

```
<companyAdmins include="true" />
```

```
<certifications include="true" />
```

```
<employees include="true" />
```

```
<referralInfo include="true" />  
<files include="true" />
```

```
</details>
</custInfoRequest>
```

#### Output Document Changes

Output element bio (includes text and images (full and thumbnail)) was added to the AA 7.1.6

Output element companyAdmins was added in the AA 7.3

Output elements committeeGrpDescr and [committeeGroup](#) (code value) were added in AA 7.3.4

Output elements [createDate](#), aliases, and [showInDirectory](#) (child of address element) were added in AA 7.4.1

Output element certifications was added in AA 7.5.

Output element custDimAttrS was added in AA 7.5.1

Output element employees was added in AA 8.0.6

Output element inheritedFromCustId was added in AA 8.1.0.

Added <lockCode> in AA8.1.0 for use in conjunction with censsawebsvclib.update\_cust\_mast\_xml.

Added files & referralInfo in AA 8.1.2

Added file tags in AA 8.1.3

#### Output Document XSD (single customer - successful)

```
<?xml version="1.0"?>
<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="custInfo">
    <xs:complexType>
      <xs:all>
        <xs:element name="custId" type="xs:string" minOccurs="0" />
        <xs:element name="lockCode" type="xs:string" minOccurs="0" />
        <xs:element name="createDate" type="xs:string" minOccurs="0" />
        <xs:element name="custType" type="xs:string" minOccurs="0" />
        <xs:element name="loginId" type="xs:string" minOccurs="0" />
        <xs:element name="name" minOccurs="0">
          <xs:complexType>
            <xs:all minOccurs="0">
              <xs:element type="xs:string" name="prefixName"/>
              <xs:element type="xs:string" name="firstName"/>
              <xs:element type="xs:string" name="middleName"/>
              <xs:element type="xs:string" name="lastName"/>
              <xs:element type="xs:string" name="suffixName"/>
              <xs:element type="xs:string" name="degreeName"/>
              <xs:element type="xs:string" name="informalName"/>
              <xs:element type="xs:string" name="displayName"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="roles" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
```

```

        <xs:element type="xs:string" name="role"
maxOccurs="unbounded" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="addresses" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="address" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element type="xs:int" name="addressSerno"/>
                        <xs:element type="xs:string" name="addressType"/>
                        <xs:element type="xs:string"
name="addressTypeDescr"/>
                        <xs:element type="xs:string" name="preferred"/>
                        <xs:element type="xs:string" name="street1"/>
                        <xs:element type="xs:string" name="street2"/>
                        <xs:element type="xs:string" name="street3"/>
                        <xs:element type="xs:string" name="city"/>
                        <xs:element type="xs:string" name="state"/>
                        <xs:element type="xs:string" name="province"/>
                        <xs:element type="xs:string" name="postalCode"/>
                        <xs:element type="xs:string" name="countryCode"/>
                        <xs:element type="xs:string" name="countryDescr"/>
                        <xs:element type="xs:string" name="county"/>
                        <xs:element type="xs:string" name="latitude"/>
                        <xs:element type="xs:string" name="longitude"/>
                        <xs:element type="xs:string" name="badAddress"/>
                        <xs:element type="xs:string" name="regionCode"/>
                        <xs:element type="xs:string" name="regionDescr"/>
                        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
                    <xs:element name="showInDirectory">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="directory" type="xs:string"
minOccurs="0" maxOccurs="unbounded" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:all>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="phones" minOccurs="0">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="phone" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element type="xs:int" name="phoneSerno"/>
                        <xs:element type="xs:string" name="phoneType"/>
                        <xs:element type="xs:string" name="phoneTypeDescr"/>
                        <xs:element type="xs:string" name="preferred"/>
                        <xs:element type="xs:string" name="number"/>
                        <xs:element type="xs:string" name="ext"/>
                        <xs:element type="xs:string" name="remark"/>
                        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="emails" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="email" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element type="xs:int" name="emailSerno"/>
                        <xs:element type="xs:string" name="emailType"/>
                        <xs:element type="xs:string" name="emailTypeDescr"/>
                        <xs:element type="xs:string" name="preferred"/>
                        <xs:element type="xs:string" name="address"/>
                        <xs:element type="xs:string" name="remark"/>
                        <xs:element type="xs:string" name="badAddress"/>
                        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="websites" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="website" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="websiteSerno" type="xs:string"

```

```

minOccurs="0" />
        <xs:element name="websiteType" type="xs:string"
minOccurs="0" />
        <xs:element name="websiteTypeDescr" type="xs:string"
minOccurs="0" />
        <xs:element name="preferred" type="xs:string"
minOccurs="0" />
        <xs:element name="address" type="xs:string"
minOccurs="0" />
        <xs:element name="remark" type="xs:string"
minOccurs="0" />
        <xs:element name="badAddress" type="xs:string"
minOccurs="0" />
        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="communicationPreferences" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="globalOptOut" type="xs:string"
minOccurs="0" maxOccurs="1" />
            <xs:element name="preference" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="categoryDescr" type="xs:string"
minOccurs="0" />
                        <xs:element name="subcategoryDescr" type="xs:string"
minOccurs="0" />
                        <xs:element name="preferredEmail" type="xs:string"
minOccurs="0" />
                        <xs:element name="receiveEmail" type="xs:string"
minOccurs="0" />
                        <xs:element name="categoryCode" type="xs:string"
minOccurs="0" />
                        <xs:element name="subcategoryCode" type="xs:string"
minOccurs="0" />
                        <xs:element name="preferredEmailSerno"
type="xs:string" minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element name="memberships" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="membership" minOccurs="0"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:all>
                                <xs:element name="subgroupId" type="xs:string"
minOccurs="0" />
                                <xs:element name="subgroupName" type="xs:string"
minOccurs="0" />
                                <xs:element name="subgroupTypeDescr"
type="xs:string" minOccurs="0" />
                                <xs:element name="member" type="xs:string"
minOccurs="0" />
                                <xs:element name="statusDescr" type="xs:string"
minOccurs="0" />
                                <xs:element name="statusCode" type="xs:string"
minOccurs="0" />
                                <xs:element name="directOrInherited"
type="xs:string" minOccurs="0" />

                                <xs:element name="relationship" minOccurs="0"
maxOccurs="1">
                                    <xs:complexType>
                                        <xs:element name="relationshipStartDate"
type="xs:string" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="relationshipEndDate"
type="xs:string" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="relationshipTypeDescr"
type="xs:string" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="inheritedFromCustId"
type="xs:string" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="relationshipType"
type="xs:string" minOccurs="0" maxOccurs="1" />
                                    </xs:complexType>
                                </xs:element>
                                <xs:element name="subgroupType" type="xs:string"
minOccurs="0" />
                                <xs:element name="classCode" type="xs:string"
minOccurs="0" />
                                <xs:element name="subclassCode" type="xs:string"
minOccurs="0" />
                                <xs:element name="classSubclassDescr"
type="xs:string" minOccurs="0" />
                                <xs:element name="joinDate" type="xs:string"
minOccurs="0" />
                                <xs:element name="expirationDate" type="xs:string"
minOccurs="0" />

```



```

        <xs:element name="paidThroughDate" type="xs:string"
minOccurs="0" />
        <xs:element name="reinstateDate" type="xs:string"
minOccurs="0" />
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

</xs:element>
<xs:element name="subscriptions" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="subscription" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="packageCode" type="xs:string"
minOccurs="0" />
                        <xs:element name="packageName" type="xs:string"
minOccurs="0" />
                        <xs:element name="beginDate" type="xs:string"
minOccurs="0" />
                        <xs:element name="expirationDate" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="jobs" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="job" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="employerName" type="xs:string"
minOccurs="0" />
                        <xs:element name="titleName" type="xs:string"
minOccurs="0" />
                        <xs:element name="titleNameLong" type="xs:string"
minOccurs="0" />
                        <xs:element name="functionDescr" type="xs:string"
minOccurs="0" />
                        <xs:element name="startDate" type="xs:string"
minOccurs="0" />
                        <xs:element name="endDate" type="xs:string"

```

```

minOccurs="0" />
        <xs:element name="hoursPerWeek" type="xs:string"
minOccurs="0" />
        <xs:element name="remark" type="xs:string"
minOccurs="0" />
        <xs:element name="employmentSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="employerId" type="xs:string"
minOccurs="0" />
        <xs:element name="functionCode" type="xs:string"
minOccurs="0" />
        <xs:element name="preferred" type="xs:string"
minOccurs="0" />
        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
        <xs:element name="employmentAttributes" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="employmentAttribute" minOccurs="0"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:all>
                                <xs:element name="type" type="xs:string" minOccurs="0" />
                                <xs:element name="typeName" type="xs:string" minOccurs="0" />
                                <xs:element name="code" type="xs:string" minOccurs="0" />
                                <xs:element name="codeDescr" type="xs:string" minOccurs="0" />
                                <xs:element name="char" type="xs:string" minOccurs="0" />
                                <xs:element name="number" type="xs:string" minOccurs="0" />
                                <xs:element name="date" type="xs:string" minOccurs="0" />
                            </xs:all>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:all>
</xs:complexType>
</xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
        <xs:element name="committeePositions" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="committeePosition" minOccurs="0"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:all>
                                <xs:element name="subgroupName" type="xs:string"
minOccurs="0" />

```

```

        <xs:element name="committeeDescr" type="xs:string"
minOccurs="0" />
        <xs:element name="committeeGrpDescr"
type="xs:string" minOccurs="0" />
        <xs:element name="positionDescr" type="xs:string"
minOccurs="0" />
        <xs:element name="startDate" type="xs:string"
minOccurs="0" />
        <xs:element name="endDate" type="xs:string"
minOccurs="0" />
        <xs:element name="positionSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="subgroupId" type="xs:string"
minOccurs="0" />
        <xs:element name="committeeType" type="xs:string"
minOccurs="0" />
        <xs:element name="committeeGroup" type="xs:string"
minOccurs="0" />
        <xs:element name="positionCode" type="xs:string"
minOccurs="0" />
    </xs:all>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="customerAttributes" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="customerAttribute" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="type" type="xs:string"
minOccurs="0" />
                        <xs:element name="typeName" type="xs:string"
minOccurs="0" />
                        <xs:element name="code" type="xs:string"
minOccurs="0" />
                        <xs:element name="codeDescr" type="xs:string"
minOccurs="0" />
                        <xs:element name="char" type="xs:string"
minOccurs="0" />
                        <xs:element name="number" type="xs:string"
minOccurs="0" />
                        <xs:element name="date" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="custDimAttrs" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="custDimAttr" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="type" type="xs:string"
minOccurs="0" />
                        <xs:element name="code" type="xs:string"
minOccurs="0" />
                        <xs:element name="dimensionType" type="xs:string"
minOccurs="0" />
                        <xs:element name="dimensionCode" type="xs:string"
minOccurs="0" />
                        <xs:element name="char" type="xs:string"
minOccurs="0" />
                        <xs:element name="number" type="xs:string"
minOccurs="0" />
                        <xs:element name="date" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="bio" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:all>
            <xs:element name="image" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="imageThumbnail" type="xs:string"
minOccurs="0" />
                        <xs:element name="imageFull" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="bioText" type="xs:string" minOccurs="0"
/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="companyAdmins" minOccurs="0">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="admin" minOccurs="0">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="id" type="xs:string" minOccurs="0"
/>
                            <xs:element name="displayNm" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="aliases" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="alias" minOccurs="0">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="aliasType" type="xs:string"
minOccurs="0" />
                            <xs:element name="aliasTypeDescr" type="xs:string"
minOccurs="0" />
                                <xs:element name="aliasValue" type="xs:string"
minOccurs="0" />
                                    </xs:all>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            <xs:element name="certifications" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="certification" minOccurs="0">
                            <xs:complexType>
                                <xs:all>
                                    <xs:element name="certificateType" type="xs:string"
minOccurs="0" />
                                        <xs:element name="certificateSpecialty"
type="xs:string" minOccurs="0" />
                                            <xs:element name="certificateLevel" type="xs:string"
minOccurs="0" />
                                                <xs:element name="certificateStatus"
type="xs:string" minOccurs="0" />
                                                    <xs:element name="certified" type="xs:string"
minOccurs="0" />
                                                        <xs:element name="certificateExpDate"
type="xs:string" minOccurs="0" />

```

```

        <xs:element name="certificateNumber"
type="xs:string" minOccurs="0" />
    </xs:all>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="employees" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="employee" minOccurs="0">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="id" type="xs:string" minOccurs="0"
/>
                        <xs:element name="firstName" type="xs:string"
minOccurs="0" />
                        <xs:element name="lastName" type="xs:string"
minOccurs="0" />
                        <xs:element name="displayName" type="xs:string"
minOccurs="0" />
                        <xs:element name="administrator" type="xs:string"
minOccurs="0" />
                        <xs:element name="lockCode" type="xs:string"
minOccurs="0" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="files" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="file" minOccurs="0">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="context" type="xs:string"
minOccurs="0" />
                        <xs:element name="docType" type="xs:string"
minOccurs="0" />
                        <xs:element name="mimeType" type="xs:string"
minOccurs="0" />
                        <xs:element name="docSize" type="xs:string"
minOccurs="0" />
                        <xs:element name="fileName" type="xs:string"
minOccurs="0" />
                        <xs:element name="shortName" type="xs:string"
minOccurs="0" />

```

```

minOccurs="0" />
    <xs:element name="descr" type="xs:string"
minOccurs="0" />
    <xs:element name="qual1" type="xs:string"
minOccurs="0" />
    <xs:element name="displayOrder" type="xs:string"
minOccurs="0" />
    <xs:element name="urlToFileThumbnail"
type="xs:string" minOccurs="0" />
    <xs:element name="urlToFile" type="xs:string"
minOccurs="0" />
    <xs:element name="tags" minOccurs="0">
        <xs:complexType>
            <xs:sequence>
                <xs:all>
                    <xs:element name="tag" type="xs:string"
minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="displayOrder"
                                type="xs:int"
                                use="optional" />
                        </xs:complexType>
                    </xs:element>
                </xs:all>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:all>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="referralInfo" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="referral" minOccurs="0">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="vendorType" type="xs:string"
minOccurs="0" />
                        <xs:element name="lastUpdatedDate" type="xs:string"
minOccurs="0" />
                        <xs:element name="status" type="xs:string"
minOccurs="0" />
                        <xs:element name="contactTxt1" type="xs:string"
minOccurs="0" />
                        <xs:element name="contactTxt2" type="xs:string"
minOccurs="0" />
                        <xs:element name="prefAddrSerno" type="xs:string"
minOccurs="0" />

```

```

minOccurs="0" />
        <xs:element name="prefEmailSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="prefWebsiteSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="prefPhoneSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="prefFaxSerno" type="xs:string"
minOccurs="0" />
        <xs:element name="publishInDirectoryFlag"
type="xs:string" minOccurs="0" />
        <xs:element name="publishAddrFlag" type="xs:string"
minOccurs="0" />
        <xs:element name="publishEmailFlag" type="xs:string"
minOccurs="0" />
        <xs:element name="publishWebsiteFlag"
type="xs:string" minOccurs="0" />
        <xs:element name="publishPhoneFlag" type="xs:string"
minOccurs="0" />
        <xs:element name="publishFaxFlag" type="xs:string"
minOccurs="0" />
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```



### Output Document (failure example)

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error code="1">Badly-formed XML</error>
</errors>
```

### Bulk Requests:

- By passing "true" in the "bulkRequest" element, you enable the service to respond with details for all custId values provided in the request.
- To enable this and maintain backwards-compatibility, only upon adding this to your request, the response will include a new **root** element of "customers" and the custInfo element (which is the root element in a non-bulk request) becomes a repeating element (one per customer requested).
- Everything about the single-customer response (XSD above) is identical to a bulk response except for the new root element and that the custInfo element can repeat.
- This can drastically reduce the overhead on ACGI and any integrator's servers by consolidating many requests into one.
- If you run into issues with large requests, please confirm that you are using POST and not GET.

### General Notes on XML Schema:

- By default, only name and customer type will be included in the response. All other sections are driven by the <details> block.
- If specifying "true" for the includeCodeValues attribute of the details element, the XML will include codes, types, and IDs that are normally suppressed. This need only be used by integrators maintaining a relational database or using conditional logic based on the data, as the descriptions may change, but the codes will not.

## Appendix A: Error Codes

The following table defines the possible error/failure codes.

error code	error message
1	Badly-formed XML
60	Your IP may not access this function
100	Bad integrator username/password combination
200	Invalid customer ID (the customer ID sent will be here)
999	Unexpected error: <i>technical error message will be included here</i>