


## 3.1.11 Customer Details Web Service (censsawebsvclib.get\_cust\_info\_xml)

 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

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<custInfoRequest>
  <custId></custId>
  <integratorUsername></integratorUsername>
  <integratorPassword></integratorPassword>
  <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" />
    <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>
  </details>
</custInfoRequest>
```

## Output Document XSD (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="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>
              <xs:element name="prefixName" type="xs:string" minOccurs="0" />
              <xs:element name="firstName" type="xs:string" minOccurs="0" />
              <xs:element name="middleName" type="xs:string" minOccurs="0" />
              <xs:element name="lastName" type="xs:string" minOccurs="0" />
              <xs:element name="suffixName" type="xs:string" minOccurs="0" />
              <xs:element name="degreeName" type="xs:string" minOccurs="0" />
              <xs:element name="informalName" type="xs:string" minOccurs="0" />
              <xs:element name="displayName" type="xs:string" minOccurs="0" />
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="roles" minOccurs="0">
          <xs:complexType>
            <xs:element name="role" nillable="true" minOccurs="0"
maxOccurs="unbounded">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:complexType>
          </xs:element>
        <xs:element name="addresses" minOccurs="0">
          <xs:complexType>
            <xs:element name="address" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:all>
                  <xs:element name="addressSerno" type="xs:string" minOccurs="0" />
                  <xs:element name="addressType" type="xs:string" minOccurs="0" />
                  <xs:element name="addressTypeDescr" type="xs:string" minOccurs="0"
/>
                    <xs:element name="preferred" type="xs:string" minOccurs="0" />
                    <xs:element name="street1" type="xs:string" minOccurs="0" />
                    <xs:element name="street2" type="xs:string" minOccurs="0" />
                    <xs:element name="street3" type="xs:string" minOccurs="0" />
                    <xs:element name="city" type="xs:string" minOccurs="0" />
                    <xs:element name="state" type="xs:string" minOccurs="0" />
                    <xs:element name="province" type="xs:string" minOccurs="0" />
                    <xs:element name="postalCode" type="xs:string" minOccurs="0" />
                    <xs:element name="countryCode" type="xs:string" minOccurs="0" />
                    <xs:element name="countryDescr" type="xs:string" minOccurs="0" />
                    <xs:element name="county" type="xs:string" minOccurs="0" />
                    <xs:element name="latitude" type="xs:string" minOccurs="0" />
                    <xs:element name="longitude" type="xs:string" minOccurs="0" />
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:element name="badAddress" type="xs:string" minOccurs="0" />
    </xs:all>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
<xs:element name="phones" minOccurs="0">
    <xs:complexType>
        <xs:element name="phone" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:all>
                    <xs:element name="phoneSerno" type="xs:string" minOccurs="0" />
                    <xs:element name="phoneType" type="xs:string" minOccurs="0" />
                    <xs:element name="phoneTypeDescr" type="xs:string" minOccurs="0" />
                </xs:all>
            </xs:complexType>
        </xs:element>
    </xs:complexType>
</xs:element>
<xs:element name="emails" minOccurs="0">
    <xs:complexType>
        <xs:element name="email" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:all>
                    <xs:element name="emailSerno" type="xs:string" minOccurs="0" />
                    <xs:element name="emailType" type="xs:string" minOccurs="0" />
                    <xs:element name="emailTypeDescr" type="xs:string" minOccurs="0" />
                </xs:all>
            </xs:complexType>
        </xs:element>
    </xs:complexType>
</xs:element>
<xs:element name="websites" minOccurs="0">
    <xs:complexType>
        <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:all>
            </xs:complexType>
        </xs:element>
    </xs:complexType>
</xs:element>
<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:all>
</xs:complexType>
</xs:element>
</xs:complexType>

```

```

</xs:element>
<xs:element name="communicationPreferences" minOccurs="0">
  <xs:complexType>
    <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:complexType>
</xs:element>
<xs:element name="memberships" minOccurs="0">
  <xs:complexType>
    <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="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:complexType>
</xs:element>
<xs:element name="subscriptions" minOccurs="0">
  <xs:complexType>
    <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:complexType>
</xs:element>
<xs:element name="jobs" minOccurs="0">
    <xs:complexType>
        <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:all>
            </xs:complexType>
        </xs:element>
    </xs:complexType>
</xs:element>
<xs:element name="committeePositions" minOccurs="0">
    <xs:complexType>
        <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="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="positionCode" type="xs:string" minOccurs="0" />
                </xs:all>
            </xs:complexType>
        </xs:element>
    </xs:complexType>
</xs:element>
<xs:element name="customerAttributes" minOccurs="0">
    <xs:complexType>
        <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: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>
```

#### 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>