ACHWorks-Verification SOAP API 2.0
# Table of Contents

1. INTRODUCTION ........................................... 3

2. DESCRIPTION OF METHODS .......................... 4
   2.1 CHECKING CONNECTION AND ACCOUNT STATUS ... 4
       2.1.1 connectionCheck .......................... 4
       2.1.2 checkCompanyStatus ..................... 5
   2.2 VERIFICATION .................................. 5
       2.2.1 verify ................................. 5

3. DATA DEFINITIONS ................................ 8
   3.1 CompanyInfo ................................ 8
   3.2 CustomerInfo ............................... 8
   3.3 AcctInfo ................................ 9
   3.4 ServiceCode ................................ 10
   3.5 verifyResp ................................ 10
   3.6 Field Types And Lengths .................... 14

4. CONTACT INFORMATION .............................. 14

APPENDICES .............................................. 15
   A1. Service Description .......................... 15
   A2. Sample Client codes ....................... 21
       A2.1 PHP .................................. 21
       A2.2 C# (.NET) ............................ 24
       A2.3 Java ................................ 26
1. Introduction

The use of web services like SOAP, both in theory and by experience, is said to be one of the most suitable applications for easy and seamless integration with existing or even new systems regardless of development platform and/or operating system. This is proven to be true for web services involving complex data structure and applications which requires frequent connections (data updates) like the Verification system. The diagram in Figure 1 briefly describes the process.

This system features a data structure that employs full object-oriented (OO) design that easily translates to non-fully OO formats like XML. Consequently, the generated classes from the web service WSDL present data as objects. This feature facilitates easier and more intuitive consuming of the web service i.e. creation of a SOAP client application.

![Figure 1. ACHWorks-Verification SOAP Schematic Diagram](image-url)
2. Description of Methods

This section lists the methods and description for using the ACHWorks-Verification web service. A brief description is provided for each method to complement the service description (test WSDL url) which can be found on this link: http://tstsvr.achworks.com/vnet/verws.asmx?WSDL. A different link for WSDL to send live verification transactions is provided for registered merchants.

For details of the data structure and definitions, please refer to Section 3 of this guide.

Viewing the sample codes, Appendix A2 in this guide, also prove useful in understanding on how a method is called in an application and to easily get started quickly.

2.1 Checking Connection and Account Status

2.1.1 connectionCheck

This is the basic method for accessing the web service to determine validity of company credentials. This method may be used for initial testing of the service only. To view the Request/Response description of this operation go to this link: http://tstsvr.achworks.com/vnet/verws.asmx?op=connectionCheck.

Format:
connectionCheckResult = connectionCheck(InpCompanyInfo),
   where ConnectionCheckResult is of type String
   InpCompanyInfo is of type CompanyInfo (Section 3.1)

Example:
The InpCompanyInfo can have the following value,
   InpCompanyInfo.SSS="TST"
   InpCompanyInfo.LocID="9505"
   InpCompanyInfo.Company="MYCOMPANY"
   InpCompanyInfo.CompanyKey="SASD_DGL_RRDG23DF9862"

and the ConnectionCheckResult object may have the following value after the method call,
connectionCheckResult -> “SUCCESS:Valid Account”

or if rejected,
connectionCheckResult -> “REJECTED:Invalid Account”
2.1.2 checkCompanyStatus

This method checks the status of the company credentials whether it’s an active or an inactive account. It is possible that an account is valid (i.e. connectionCheck) but not yet activated or already been deactivated or closed. An account must be active to initiate methods for sending verification transactions. To view the Request/Response description of this operation go to this link: http://tstsvr.achworks.com/vnet/verws.asmx?op=checkCompanyStatus.

Format:
checkCompanyStatusResult = checkCompanyStatus(InpCompanyInfo),
where CheckCompanyStatusResult is of type String
InpCompanyInfo is of type CompanyInfoVer (Section 3.1)

Example:
The InpCompanyInfo can have the following value,
InpCompanyInfo.SSS="TST"
InpCompanyInfo.LocId="9505"
InpCompanyInfo.Company="MYCOMPANY"
InpCompanyInfo.CompanyKey="SASD_DGL_RRDG23DF9862D"

and the CheckCompanyStatusResult object may have the following value after the method call,
checkCompanyStatusResult -> "SUCCESS:Valid And Active Account"

or if rejected,
checkCompanyStatusResult -> "REJECTED:Invalid And/Or Inactive Account"

2.2 Verification

2.2.1 verify

This is the main method used for initiating the Verification process. To view the Request/Response description of this operation go to this link: http://tstsvr.achworks.com/vnet/verws.asmx?op=verify

Format:
verifyResult = verify(InpServiceCode, InpCompanyInfoVer, InpCustomerInfoVer, InpAcctInfoVer),
where verifyResult is of type verifyResp (Section 3.5)
InpServiceCode is a String to specify service options (Section 3.4)
InpCompanyInfoVer is of type CompanyInfoVer (Section 3.1)
InpCustomerInfoVer is of type CustomerInfoVer (Section 3.2)
InpAcctInfoVer is of type AcctInfoVer (Section 3.3)

Example:
The input InpCompanyInfoVer, InpCustomerInfoVer and InpAcctInfoVer objects may have the following values,
InpCompanyInfoVer.SSS="TST"
InpCompanyInfoVer.LocID="9502"
InpCompanyInfoVer.Company="MYCOMPANY"
InpCompanyInfoVer.CompanyKey="SASD_DGL_RRDG23DF9862D"

InpCustomerInfoVer.FirstName="JOHN"
InpCustomerInfoVer.LastName="SMITH"
InpCustomerInfoVer.SSN="123124567"
InpCustomerInfoVer.DateOfBirth="12/24/1978"

InpAcctInfoVer.Amount = 100.25;
InpAcctInfoVer.AcctType = "C"
InpAcctInfoVer.AcctNo = "000023217926"
InpAcctInfoVer.RoutingNo = "987654320"

Then, the **verifyResult** object may have the following value after the method call,
verifyResult.RefNo->"TQ-123"
verifyResult.RequestCode->"1111"
verifyResult.SuccessCode->"1111"
verifyResult.Remarks->"Valid Account"
verifyResult.CKVXML->"<CKV>
  <amount>100.25</amount>
  <accttype>C</accttype>
  <routingno>987654320</routingno>
  <acctno>000023217926</acctno>
  <code>XB4</code>
  <result>NEG</result>
  <meaning>Invalid Trn</meaning>
  <description>Account has invalid routing number</description>
</CKV>"
verifyResult.SSNXML->"<SSN>
  <Response>
    <custdateofbirth>12/24/1978</custdateofbirth>
    <responsecode>211</responsecode>
    <responseoother>PRIOR TO 7/1/83; US.RS</responseoother>
    <responsedescription>Active</responsedescription>
    <ssnstatus>
      <ssn>123124567</ssn>
      <status>Active</status>
      <issued>U.S RS</issued>
      <period>PRIOR TO 7/1/83</period>
    </ssnstatus>
  </Response>
</SSN>
verifyResult.BTRNXML->"<BTRN>
  <description>RDFI not in database</description>
  <code>1</code>
  <name></name>
  <address></address>
  <city></city>
  <state></state>
  <zip></zip>
  <zipext></zipext>
  <phone></phone>
</BTRN>
verifyResult.OFACXML->"<OFAC>
  <description>False</description>
  <code>1</code>
  <row>1</row>
  <identifier/></identifier>
  <sdnname></sdnname>
  <sdntype></sdntype>
  <program></program>
  <title></title>
  <vesselcallsign></vesselcallsign>
  <vesseltype></vesseltype>
  <vesseltonnage></vesseltonnage>
  <grossregtonnage></grossregtonnage>
  <vesselflag></vesselflag>
  <vesselowner></vesselowner>
  <remarksonsdn></remarksonsdn>
</OFAC>

Or if rejected,
verifyResult.Remarks->"Invalid Account"
3. Data Definitions

This lists the data types for input/output in using the ACHWorks-Verification SOAP web service. A brief description is provided for each method to complement the service description (WSDL url) which can be found on this link: http://tstcc.achworks.com/cnet/ccws.asmx?WSDL.

Please see Section 3.8 for required field types and lengths.

3.1 CompanyInfoVer

The CompanyInfoVer serves as the basic company credential for accessing the ACHWorks-Verification SOAP web service. This is a required input for each method or operation. It consists of SSS, LocID, Company and CompanyKey which will be described below.

3.1.1 LocID
LocID is a 4-digit alphanumeric code furnished to registered merchants. For sample codes and for using the test server, the value ‘9502’ or ‘9505’ is commonly used.

3.1.2 Company
Company is an alpha-numeric user ID (used as username for the service) furnished to registered merchants. For sample codes and for using the test server, the value ‘MY COMPANY’ or ‘THAT COMPANY’ is commonly used.

3.1.3 CompanyKey
CompanyKey is an alpha-numeric user KEY (used as password for the service) furnished to registered merchants.

3.2 CustomerInfoVer

The CustomerInfoVer represents the basic or single Customer. It consists of FirstName, LastName, Social Security Number and DateOfBirth fields which will be described below. For OFAC matching, FirstName and LastName are the only required fields. For Social Security Number validation all these fields are required. For details on the field types and format please refer to Table 3.8.1.

3.2.1 FirstName
The First Name of the customer in a customer record.
3.2.2 **LastName**  
The Last Name of the customer in a customer record.

3.2.3 **SSN**  
The 9 digit social security number (without dash).

3.2.4 **DateOfBirth**  
The date of birth in this format mm/dd/yyyy.

### 3.3 **AcctInfoVer**

The AcctInfoVer represents an object for the bank account information. It consists of Amount, AcctType, AcctNo and RoutingNo which will be described below. For details on the field types and format please refer to Table 3.8.1.

3.3.1 **Amount**  
Amount of the transaction in US dollars e.g. 100.25.

3.3.2 **AcctType**  
This is bank account type which can have a value of C for checking or S for savings.

3.3.3 **AcctNo**  
This is the bank account number (16 digits maximum).

3.3.4 **RoutingNo**  
This is the bank routing number (ABA).
3.4 InpServiceCode

Table 3.4.1 Combination of service options and values for InpServiceCode

<table>
<thead>
<tr>
<th>Check Verification</th>
<th>SSN Validation</th>
<th>Routing Number ODFI Info</th>
<th>OFAC Match</th>
<th>InpServiceCode*</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1111</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>1110</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>1100</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>0001</td>
</tr>
</tbody>
</table>

* e.g. if verification request requires (✓) all four service options, the InpServiceCode value is 1111

3.5 verifyResp

The TransactionResp is the result object for sending credit card transaction commands (Section 2.2) such as sendCCTrans, sendCCAuth, sendCCCapture and sendCCVoid. It consists of TransResult, RefNo, BatchNo, AddressVerResult, CardResult, Status, Remarks and Transaction which will be described below. For details on the field types and format please refer to Table 3.8.1.

3.5.1 RefNo

This is the unique reference number assigned by ACHWorks verification system. The reference number can be used for tracking a specific verification transaction.

3.5.2 RequestCode

This is the same as the InpServiceCode during the time of the verification request (see Section 3.4).

3.5.3 SuccessCode

This is the approved InpServiceCode after issuing the request (see Section 3.4).

3.5.4 CKVXML

The result of Check Verification in XML format. The result follows this structure:

```
<CKV>
    <amount></amount>
    <accttype></accttype>
    <routingno></routingno>
```
Values of codes, meanings and descriptions are provided in this document ACHWorks Account Verification Scope of Services ([http://achworks.com/files/achworks-accountverificationscope.pdf](http://achworks.com/files/achworks-accountverificationscope.pdf)).

### 3.5.5 SSNXML

The result of SSN validation in XML format. The result follows this structure:

```
<SSN>
  <Response>
    <custdateofbirth/></custdateofbirth>
    <responsecode/></responsecode>
    <responseother/> //Issue Date Range; State of Issue
    <responsedescription/></responsedescription> //SSN Validate, SSN Invalid, etc
    <ssnstatus>
      <ssn/></ssn>
      <status/></status>
      <issued/></issued>
      <period/></period>
      <sourceid/></sourceid>
    </ssnstatus>
  </Response>
</SSN>
```

Please see Table 3.5.2 below for possible values of SSN response codes and descriptions.

<table>
<thead>
<tr>
<th>RESPONSE CODE</th>
<th>RESPONSE DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>SSN Valid or Active</td>
</tr>
<tr>
<td>-211</td>
<td>SSN has not been issued</td>
</tr>
<tr>
<td>-212</td>
<td>SSN associated with a deceased person</td>
</tr>
<tr>
<td>-213</td>
<td>SSN issued within 5 years of inquiry</td>
</tr>
<tr>
<td>-214</td>
<td>SSN issued prior to date of birth</td>
</tr>
<tr>
<td>-215</td>
<td>Data missing or invalid</td>
</tr>
</tbody>
</table>

### 3.5.6 BTRNXML

The result of Routing Number Validation in XML format. The Routing Number result follows this structure:
3.5.7 OFACXML
The result of OFAC matching in XML format. The OFAC result follows this structure:

<OFAC>
   <description>//True or False</description>
   <row/>
   <identifier/>
   <sdnnname/>
   <sdntype/>
   <program/>
   <title/>
   <vessell Callsign/>
   <vesselt ype/>
   <vesseltonnage/>
   <grossregtonnage/>
   <vesselflag/>
   <vesselowner/>
   <remarksonsdn>
</OFAC>

3.5.8 ServerTime
This is server time after a successful verification request.

3.5.9 Remarks
This indicates validity of verification request and may contain additional information pertaining to the transaction.

3.8 Field Types and Length
Listed in the table below are the fields with the required types and lengths for data entry/output. The web service allows entry of some of the string types e.g. Customer’s FirstName to exceed the required length but the value is trimmed to the required length during processing and final storage.

**Table 3.8.1 Fields with required types and lengths.**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Length</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>InpServiceCode</td>
<td>String</td>
<td>4</td>
<td>Values are 1111, 1000, 1100 etc. see Table 3.4.1</td>
</tr>
<tr>
<td>SSS</td>
<td>String</td>
<td>3</td>
<td>Use the value assigned by T$$ (TST for test server)</td>
</tr>
<tr>
<td>LocID</td>
<td>String</td>
<td>4</td>
<td>Use the value assigned by T$$ (e.g. 9505 for test server)</td>
</tr>
<tr>
<td>Company</td>
<td>String</td>
<td>&gt;=8</td>
<td>Use the value assigned by T$$</td>
</tr>
<tr>
<td>CompanyKey</td>
<td>String</td>
<td>&gt;=8</td>
<td>Use the value assigned by T$$</td>
</tr>
<tr>
<td>FirstName</td>
<td>String</td>
<td>-</td>
<td>Customer’s First Name (Section 3.2.1)</td>
</tr>
<tr>
<td>LastName</td>
<td>String</td>
<td>-</td>
<td>Customer’s Last Name (Section 3.2.2)</td>
</tr>
<tr>
<td>SSN</td>
<td>String</td>
<td>9</td>
<td>Social Security Number</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>String</td>
<td>10</td>
<td>Date of Birth in this format mm/dd/yyyy</td>
</tr>
<tr>
<td>Amount</td>
<td>Double</td>
<td>-</td>
<td>Amount in dollars, e.g. 100.75, MAX = 999999.99</td>
</tr>
<tr>
<td>AcctType</td>
<td>String</td>
<td>1</td>
<td>Checking (C) or Savings (S)</td>
</tr>
<tr>
<td>AcctNo</td>
<td>String</td>
<td>16</td>
<td>Account Number</td>
</tr>
<tr>
<td>RoutingNo</td>
<td>String</td>
<td>9</td>
<td>Bank Routing Number</td>
</tr>
</tbody>
</table>

**4. Contact Information**

For more information and/or questions please email ACHWorks support: support@achworks.com or call us at +1 916 638 8811.
APPENDICES

A1. Service Description

Test WSDL URL: http://tstsvr.achworks.com/vnet/verws.asmx?WSDL
(A Live WSDL URL will be provided to registered merchants)

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdll/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://ver.achworks.com/">
<wsdl:documentation
<wsdl:types>
<s:schema elementFormDefault="qualified" targetNamespace="http://ver.achworks.com/">
<s:element name="connectionCheck"/>
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="InpCompanyInfo" type="tns:CompanyInfoVer"/>
</s:sequence>
</s:complexType>
</s:element>
s:complexType name="CompanyInfoVer">
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="SSS" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="LocID" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="Company" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="CompanyKey" type="s:string"/>
</s:sequence>
</s:complexType>
</s:complexType>
<s:element name="connectionCheckResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="connectionCheckResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="checkCompanyStatus">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="InpCompanyInfo" type="tns:CompanyInfoVer"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="checkCompanyStatusResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="checkCompanyStatusResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="verify">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="InpServiceCode" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="InpCompanyInfo" type="tns:CompanyInfoVer"/>
      <s:element minOccurs="0" maxOccurs="1" name="InpCustomerInfo" type="tns:CustomerInfoVer"/>
      <s:element minOccurs="0" maxOccurs="1" name="InpAcctInfo" type="tns:AcctInfoVer"/>
    </s:sequence>
  </s:complexType>
</s:element>

<s:complexType name="CustomerInfoVer">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="FirstName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="LastName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="SSN" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="DateOfBirth" type="s:string"/>
  </s:sequence>
</s:complexType>

<s:complexType name="AcctInfoVer">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Amount" type="s:double"/>
    <s:element minOccurs="0" maxOccurs="1" name="AcctType" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="AcctNo" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="RoutingNo" type="s:string"/>
  </s:sequence>
</s:complexType>
<s:sequence>
  </s:complexType>
</s:element>
</s:complexType>
<s:complexType name="VerifyResp">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="RefNo" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="RequestCode" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="SuccessCode" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="CKVXML" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="SSNXML" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="BTRNXML" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="OFACXML" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="ServerTime" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="Remarks" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="Status" type="s:string"/>
  </s:sequence>
</s:complexType>
</s:schema>
</wsdl:types>
</wsdl:message>
<wsdl:message name="checkCompanyStatusSoapIn">
  <wsdl:part name="parameters" element="tns:checkCompanyStatus"/>
</wsdl:message>
<wsdl:message name="checkCompanyStatusSoapOut">
  <wsdl:part name="parameters" element="tns:checkCompanyStatusResponse"/>
</wsdl:message>
<wsdl:message name="verifySoapIn">
  <wsdl:part name="parameters" element="tns:verify"/>
</wsdl:message>
<wsdl:message name="verifySoapOut">
  <wsdl:part name="parameters" element="tns:verifyResponse"/>
</wsdl:message>
<wsdl:portType name="ACHWorksVerWSSoap">
  <wsdl:operation name="connectionCheck"/>
<wsdl:input message="tns:connectionCheckSoapIn"/>
<wsdl:output message="tns:connectionCheckSoapOut"/>
</wsdl:operation>
<wsdl:operation name="checkCompanyStatus">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Checks status of user/company account</wsdl:documentation>
<wsdl:input message="tns:checkCompanyStatusSoapIn"/>
<wsdl:output message="tns:checkCompanyStatusSoapOut"/>
</wsdl:operation>
<wsdl:operation name="verify">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Main verification method</wsdl:documentation>
<wsdl:input message="tns:verifySoapIn"/>
<wsdl:output message="tns:verifySoapOut"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ACHWorksVerWSSoap" type="tns:ACHWorksVerWSSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="connectionCheck">
<soap:operation soapAction="http://ver.achworks.com/connectionCheck" style="document"/>
<wsdl:input>
<soap:body use="literal"/> </wsdl:input>
<wsdl:output>
<soap:body use="literal"/> </wsdl:output>
</wsdl:operation>
<wsdl:operation name="checkCompanyStatus">
<wsdl:input>
<soap:body use="literal"/> </wsdl:input>
<wsdl:output>
<soap:body use="literal"/> </wsdl:output>
</wsdl:operation>
<wsdl:operation name="verify">
<wsdl:input>
<soap:body use="literal"/> </wsdl:input>
<wsdl:output>
<soap:body use="literal"/> </wsdl:output>
</wsdl:operation>
</wsdl:binding>
ACHWorksVerWS (ACHWorks-Ver SOAP 2.0) is a web service (web API) by [ACHWorks](http://www.achworks.com) that provides an API for initiating Verification methods (Check Verification, OFAC, Routing No Validation and SSN Validation). The complete API Guide can be downloaded [here](http://www.achworks.com/files/achworksversoapver1guide.pdf). Please [contact us](http://www.achworks.com/contactus) for more information. Last updated on: 10/02/2014. Release Version 2.0TSTSVR.
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
A2. Sample Client Codes

The sample codes below are written for the most popular and recent development platforms namely PHP, C#, Visual Basic (VB.NET), ASP.NET and JAVA.

Brief descriptions are provided to provide ease in understanding the required input and output and their formats.

Please contact us if you want a digital copy of the sample codes. See Section 4 for contact information.

A2.1 PHP (Version 5)

connectioncheck.php

```php
<?php
//ACHWORKS-Ver ConnectionCheck (Connection Check)
//company info
class CompanyInfoVer {
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

$mycompanyinfo = new CompanyInfoVer;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";


//Important: use InpCompanyInfo in the soap stru not CompanyInfo
$myresult = $client->connectionCheck(array("InpCompanyInfo"=>$mycompanyinfo));

print($myresult->connectionCheckResult);
?>
```
verify.php

<?php

//ACHWORKS-Ver SOAP verify (Sends a single verification request)
//08.01.2014 - rico pamplona, rpamplona@achworks.com

//Company info
class CompanyInfoVer
{
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

class CustomerInfoVer
{
    public $FirstName;
    public $LastName;
    public $SSN;
    public $DateOfBirth;
}

class AcctInfoVer
{
    public $Amount;
    public $AcctType;
    public $AcctNo;
    public $RoutingNo;
}

//CompanyInfo
$mycompanyinfo = new CompanyInfoVer;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";

//Customer
$myCustomerInfo = new CustomerInfoVer;
$myCustomerInfo -> FirstName = "JOHN";
$myCustomerInfo -> LastName = "SMITH";
$myCustomerInfo -> SSN = "239567045";
$myCustomerInfo -> DateOfBirth = "04/23/1984";

//AcctInfo
$myAcctInfo = new AcctInfoVer;
$myAcctInfo -> Amount = 540.75;
$myAcctInfo -> AcctType = "C";
$myAcctInfo -> AcctNo = "0002322322768";
$myAcctInfo -> RoutingNo = "987654320";

$myServiceCode = "1111";

//SOAP call - test server
$myresult = $myclient->verify(array("InpServiceCode"=>$myServiceCode,
    "InpCompanyinfo"=>$mycompanyinfo,"InpCustomerinfo"=>$myCustomerInfo,"InpAcctInfo"=>$myAcctInfo))->verifyResult;

//print status and details
print($myresult->Status " , "$myresult->Remarks "<br">");
?>
A2.2 C# (.NET)

Connectioncheck
private void ConnectionCheck_Click(object sender, EventArgs e)
{
    //instance
    ver.ACHWorksVerWS myVer = new ver.ACHWorksVerWS();

    //Company Info
    myCompanyInfo.LocID = "9505";
    myCompanyInfo.Company = "MYCOMPANY";
    myCompanyInfo.CompanyKey = "PASSWORD123";
    myCompanyInfo.SSS = "TST";

    //Connection Result
    try
    {
        String myConnectResult = myVer.connectionCheck(myCompanyInfo);
        //Add result to ListBox
        listBox1.Items.Clear();
        listBox1.Items.Add(myConnectResult);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}

verify
private void Verify_Click_1(object sender, EventArgs e)
{
    //instance
    ver.ACHWorksVerWS myVer = new ver.ACHWorksVerWS();
    ver.VerifyResp myResult = new ver.VerifyResp();

    //Company Info
    myCompanyInfo.LocID = "9505";
    myCompanyInfo.Company = "MYCOMPANY";
    myCompanyInfo.CompanyKey = "PASSWORD123";
    myCompanyInfo.SSS = "TST";

    //Account Info
    ver.AcctInfoVer myAcctInfo = new ver.AcctInfoVer();
    myAcctInfo.AcctType = "C";
    myAcctInfo.AcctNo = "0012362986";
myAcctInfo.RoutingNo = "987654320";
myAcctInfo.Amount = 100;

//Customer Info
ver.CustomerInfoVer myCustomerInfo = new ver.CustomerInfoVer();
myCustomerInfo.FirstName = "John";
myCustomerInfo.LastName = "Doe";
myCustomerInfo.DateOfBirth = "12/31/1978";
myCustomerInfo.SSN = "728128765";

//Service Code
String myOption = "1111";

//Verification
try {
    myResult = myVer.verify(myOption, myCompanyInfo, myCustomerInfo, myAcctInfo);

    //Add result to ListBox
    listBox1.Items.Clear();
    listBox1.Items.Add("RefNo: " + myResult.RefNo);
    listBox1.Items.Add("Remarks: " + myResult.Remarks);
    listBox1.Items.Add("");

    listBox1.Items.Add("Routing No Validation XML: " + myResult.BTRNXML);
    listBox1.Items.Add("Check Verification XML: " + myResult.CKVXML);
    listBox1.Items.Add("SSN Validation XML: " + myResult.SSNXML);
    listBox1.Items.Add("OFAC XML: " + myResult.OFACXML);
} catch (Exception ex) {
    MessageBox.Show(ex.ToString());
};
A2.3 Java (1.5/1.6)

connectioncheck

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        //instances
        ACHWorksVerWS myVerWS = new ACHWorksVerWS();
        CompanyInfoVer myCompanyInfo = new CompanyInfoVer();
        //assign values for myCompanyInfo
        myCompanyInfo.sss = "TST";
        myCompanyInfo.locID = "9505";
        myCompanyInfo.company = "THAT COMPANY";
        myCompanyInfo.companyKey = "RICO";

        //put value of ConnectionCheck call to jTextField
        jTextField1.setText(myVerWS.getACHWorksVerWSSoap().connectionCheck(myCompanyInfo));
    } catch (Exception e) {
        jTextField1.setText("Exception caught: "+ e);
    }
}

verify

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        //instances
        ACHWorksVerWS myVerWS = new ACHWorksVerWS();
        CompanyInfoVer myCompanyInfo = new CompanyInfoVer();
        CustomerInfoVer myCustomerInfo = new CustomerInfoVer();
        AcctInfoVer myAcctInfo = new AcctInfoVer();
        verifyResp myVerifyResp = new verifyResp();

        //assign values for myCompanyInfo
        myCompanyInfo.sss = "TST";
        myCompanyInfo.locID = "9505";
        myCompanyInfo.company = "THAT COMPANY";
        myCompanyInfo.companyKey = "RICO";
//Customer
myCustomerInfo.firstName="JOHN";
myCustomerInfo.lastName="SMITH";
myCustomerInfo.SSN="613453456";
myCustomerInfo.DateOfBirth="12/24/1978";

//Acct
myAcctInfo.amount=540.75;
myAcctInfo.accttype="C";
myAcctInfo.acctno="00002321260527";
myAcctInfo.routingno="987654320";

//Service Code
myServiceCode="1111";

//call verify method
myTransactionResp = myVerWS.getACHWorksVerWSSoap().verify(myServiceCode,myCompanyInfo,
        myCustomerInfo, myAcctInfo);
        jTextField1.setText(myTransactionResp.status + ", " + myTransactionResp.remarks);

} catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}