Real-Time Inquiry Connectivity Specifications

Highmark, Inc.
2008
1. Overview

Real-time transactions utilize Simple Object Access Protocol (SOAP). SOAP is a simple XML based protocol to let applications exchange information over HTTP. Since the Internet is being utilized to transport the data, encryption will be utilized to secure messages in the same way financial transactions are secured over the Internet. Access to Highmark’s networks will follow the same security model in place today, which requires a Login/Password.

**HIPAA Real Time**

[Diagram showing the transaction process between a Trading Partner and Highmark's networks.]

- **Authentication:**
  - Logon ID
  - Password
  - Role (Real Time role)

- **Message Formats:**
  - HTTPS – Secure HTTP connection (encrypted)
  - SOAP/XML – Message Format
  - HIPAA X12 – Message Payload Format
In order to understand the lifecycle of the transaction, processes have been outlined below:

(1) Transaction Initiation
Highmark Trading Partner’s Transaction Management System will initiate a Real-time X12 HIPAA transaction.

(2) Establish Connection
The Trading Partner’s Transaction Management System will establish a secure Internet connection (HTTPS) to Highmark and send an encrypted SOAP message that contains a HIPAA X12 transaction payload, along with the Trading Partner logon id, and password assigned by Highmark.

(3) Receive Transaction
Highmark receives the Real-time request on its Web Server.

(4) Authentication/Authorization
When the SOAP message is received by Highmark’s WebSphere application, the SOAP message will be validated and the Trading Partner’s logon id, password and defined role is authenticated using the Directory Smart LDAP (Lightweight Directory Access Protocol). Only Trading Partners that have signed a Highmark Trading Partner Agreement are granted a logon id’s, passwords and defined role.

If the Trading Partner is not authorized to submit a Real-time request, the WebSphere application will return a SOAP invalid security/unauthorized message to the Trading Partner via the secure Internet connection (HTTPS).

(5) Process Transaction
Trading Partners authorized to submit real time requests will have their transactions routed through the WebSphere application to the target system. The target system will generate the Real-time response.

(6) Format Response
The WebSphere Application Server will envelope the response in a SOAP response message.

(7) Send Response
The responses will be encrypted, and returned to the Trading Partner via the secure Internet (HTTPS) connection.

(8) Receive Response
The Trading Partner’s Web Server will return the response message to the Trading Partner’s Transaction Management System that initiated the request.
2. Trading Partner Requirements for Real-Time

- Trading Partners must submit Inquiry transactions using HTTPS over a public line.

- Trading Partners must be able to connect to https://webservices.highmark.com/edirpc/servlet/TPSOAPServlet

- Trading Partner must ensure that only authorized persons and/or applications will be able to submit requests to Highmark with their logon id and password.

- Highmark Real-Time transactions (Request and Response) are based on standard SOAP formats. However, due to Highmark system requirements Real-Time transactions must adhere to Highmark's Model SOAP Messages (see Section 3).

- The SOAP message should be submitted in a continuous data string without line feeds.

- The SOAP message must not contain spaces between data tags.

- The SOAP message “header” must contain the following required data elements for all Highmark Real-Time transactions:
  
  Username = (7 positions, Upper Case) Highmark assigned login id.
  
  Password = (8 positions)

- The SOAP message “body” must contain the following required data elements for all Highmark Real-Time transactions:

  X12TypeVersion = same value as GS08 in the X12 request
  - ‘004010X092A1’ (270 Eligibility Request)
  - ‘004010X093A1’ (276 Claim Status Request)
  - ‘004010X094A1’ (278 Referral/Authorization Request)

  SenderId = (7 position, Upper Case) Highmark assigned login id.
  Same value as Username in the SOAP Header.

  RequestTarget = (16 positions) must contain ‘HighmarkMercator’

  ClientUserld = (1 to 30 positions) Trading Partner defined (used if Trading Partner wants to return data in the SOAP Response Message).
**ClientStateData** = (1 to 50 positions) Trading Partner defined (used if Trading Partner wants to return data in the SOAP Response Message).

- Although **Client UserId** and **ClientStateData** are required fields, Highmark **will not** authenticate/validate content of the data in these fields.

- The Trading Partner must use a ‘~’ as the segment terminator, the ‘^’ element delimiter and the ‘:’ Component Element Separator.

- The Trading Partner will be responsible to evaluate the response returned and to resubmit the request with corrections required as indicated by the SOAP “faultcode”.

- No XML exception characters (&, <, >, “) or non-printable characters will be used as a delimiter or contained within the data of the message. NOTE: Highmark requires the CDATA tag to handle special characters. (See examples of SOAP Message and XML Message below)

**DISCLAIMER**

Real-time transactions are designed to respond to individual end-user requests for eligibility, claim status or authorization information. For typical requests (requests with a single patient), the average response time should be within 15 seconds. Actual response time will be dependent upon Real-time transaction activity. Batched inquiries should not be submitted through the Real-time process as it may impact Real-time response time.

**3. MODEL SOAP MESSAGES** – The following are models of valid Highmark Real-time transactions (Request and Response) with properly formatted SOAP envelopes.

**Sample 270 Request Message:**

```xml
<?xml version='1.0' encoding='UTF-8'?>
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
  <wsse:UsernameToken>
   <wsse:Username>someuser</wsse:Username>
   <wsse:Password>somepasswd</wsse:Password>
  </wsse:UsernameToken>
 </wsse:Security>
</SOAP-ENV:Header>
```
4. EDI Real-Time SOAP Faults – When a Real-Time transaction fails validation for the format or content of the SOAP message; the following error codes will be used when responding to the Trading Partner.

<table>
<thead>
<tr>
<th>Fault Message</th>
<th>FAULTCODE</th>
<th>ClientStateData returned *</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP message request contains no content</td>
<td>SOAP-ENV:Client</td>
<td>No</td>
</tr>
<tr>
<td>HTTPS SOAP message request contains invalid content type of ‘##1##’.</td>
<td>SOAP-ENV:Client</td>
<td>No</td>
</tr>
<tr>
<td>Content type should be text/xml</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User authentication/authorization failed for the request [####]</td>
<td>SOAP-ENV:Client</td>
<td>No</td>
</tr>
<tr>
<td>Invalid request target [####]</td>
<td>SOAP-ENV:Client</td>
<td>Yes</td>
</tr>
<tr>
<td>Unable to parse SOAP</td>
<td>SOAP-ENV:Client</td>
<td>No, for first</td>
</tr>
<tr>
<td>request message [#####]</td>
<td>ENV:Client example; Yes, for the second</td>
<td></td>
</tr>
</tbody>
</table>
Unable to parse SOAP request message because it is not text | SOAP-ENV:Client | No
---|---|---
X12 type version #### not supported | SOAP-ENV:Client | Yes
Request target could not process request [####] | SOAP-ENV:Client | Yes
Unable to authenticate user | SOAP-ENV:Server | No
Unable to parse SOAP response from RequestTarget [####] | SOAP-ENV:Server | No
Unable to process request | SOAP-ENV:Server | No
Request target transport type of [####] not implemented for request target of [####] | SOAP-ENV:Server | No
Request timed out | SOAP-ENV:Server | Yes

An example of a SOAP response with `<ClientStateData>`:
```xml
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
  <SOAP-ENV:Fault>
   <faultcode>
    SOAP-ENV:Client
   </faultcode>
   <faultstring>
    <![CDATA[X12 type version xxxxxxx not supported [xxxxxxx type version not supported]]]>
   </faultstring>
   <detail
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <![CDATA[use for client to put any key data to return in response]]>
   </ClientStateData>
  </detail>
 </SOAP-ENV:Fault>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
5. Highmark EDI WebServices Certificate

This Section will explain how to save to a file the certificate used by the Highmark Web Services Gateway. Highmark offers the use of web services to perform EDI transactions. Since these transactions require the utmost security, all data is encrypted and transmitted over Secure Sockets Layer Protocol (SSL). The document will provide some links to information about SSL and instructions for downloading to a file the Highmark certificate that would be required to be setup as a Truststore to establish a SSL connection with the web services gateway server. Note: This document is meant for individuals whom have information technology experience and a understanding of SSL and web services.

A. Introduction

Prior to obtaining the Highmark Public Certificate for the enablement of the EDI web services one should have an understanding of SSL. Here's a link to a SSL Introduction.

Introduction to SSL

B. Downloading the Digital Certificate.

These instructions are for Windows Internet Explorer. First, open a web browser to the https://webservices.highmark.com home page. You should notice a “Security Alert” window on this window select the View Certificate button. (Note: if you do not see the “Security Alert” window skip to the next screen example.
If you did not see the Security Alert window, double click on the gold "LOCK" icon at the bottom right status bar of your Windows Internet Explorer Browser.
Next, you’ll see the Certificate Window which displays the general information about the Certificate. Select the Details tab.
On the Details tab, select the “Copy to File…” button at the lower right corner.
You should now see the Welcome window for the Certificate Export Wizard for windows. Select the Next Button.

Select the File Format. In this example, the default DER encoded binary X.509 is selected. Click Next.
Enter the file name to save the certificate under. In this example C:/downloadwsgwyhighmark.cer was used. Click Next.

Select Finish Button...
C. What to do next.

In order to establish a SSL connection via a web services client usually one needs to load the certificate to the “Truststore” file configured for the client’s platform in which the web service will be invoked. Since, there are to many web services client platforms to mention in this guide, one will need to review the documentation associated with the web services client platform. For example, to obtain information for a client platform like IBM’s WebSphere Application Server one could perform a search on Google(www.google.com) for: “ibm websphere Truststore how to” or examine the Websphere Information Center Documentation.

D. Renewing the Certificate

Digital Certificates have a specific expiration date and will need to be renewed. In the example below the General Tab identifies the Valid To and From dates for the Certificate. Each Real-Time Trading Partner will receive an email 4 weeks before the certificate expires. (Note: Email addresses should be supplied by the EDI Trading Partner upon applying for Real-Time access.) The email will include the date the new certificate will be available for download and the date the new certificate will be activated. Repeat the above steps to incorporate the renewed certificate in your application.