Medicity

A Healthagen Business

Http Post/Soap Forwarder (DHIN Specific Edition)

HIE Immunizations Proxy

12/8/2014

Version 1.0 / Final

Medicity

A Healthagen Business

Medicity has prepared this document for use by Medicity personnel, licensees, and customers. The information contained herein is the property of Medicity and shall not be copied, photocopied, translated, or reduced to any electronic or machine readable form, either in whole or in part, without prior written approval from Medicity.

© 2014 by Medicity, A Healthagen Business. All Rights Reserved. Printed in the USA.

Trademarks and registered names:

ProAccess is a trademark of Medicity

All other trademarks or registered trademarks are the property of their respective owners.

Medicity
257 E. 200 S.
Suite 1300
Salt Lake City, UT 84111
Phone 801.322.4444 Fax 801.322.4413
www.medicity.com

Table of Contents

1 Overview 2

1.1 Document Version 2

2 Configuration 3

2.1 AppSettings Section 3

2.2 Sample Configuration 4

3 WSDL 7

4 Soap Message Format 8

4.1 Operation ExecuteHL7Message 8

4.2 Operation ExecuteHL7MessageByte 10

1 Overview

The Http Post/Soap Forwarder (proxy) service forwards Http Post and Soap messages to the Delaware Immunizations Registry and returns the response back to the sender. This version of the proxy service accepts SOAP messages specific to the Delaware Immunization Information System, DelVAX. The purpose of this document is to explain the format for sending SOAP messages to the proxy service.

1.1 Document Version

Author	Version	Date	Comments	Status
Bryan Wagstaff	1.0	12/2/2014	Initial Document	Final

2 Configuration

This service runs under the WCFServiceHost.exe, therefore, the configuration will be found in the file WCFServiceHost.exe.config file. You will need to refer to other documentation on setting up and configuring the WCFServiceHost. For this application specific setup, it will require a Medicity.syslog section and an appSettings section. In the system.serviceModel section, you will need to add a mex endpoint and in the service Configuration you will need to set the httpGetEnabled to true so the wsdl can be produced. All this is usually included in the sample config files in the deployment directory. There is also a sample configuration included at the end of this section.

NOTE: When this service will be available over the public internet, it should be secured using TLS protocol.

2.1 AppSettings Section

The appSettings section contains the following:

```
<appSettings>
  <add key="PostTo"       value="https://ws.delaware.gov:1403/dhss/HL7Service" />
  <add key="Content-Type" value="soap" />
  <add key="namespace"    value="http://tempuri.org/" />
  <add key="soapaction"    value="http://tempuri.org/ExecuteHL7Message" />
  <add key="operation"     value="ExecuteHL7Message" />
</appSettings>
```

Explanation of each setting is:

- PostTo — This value is set to the URL that processes the immunizations for the HIE.
  DHIN PROD: https://ws.delaware.gov:1403/dhss/HL7Service
  DHIN CERT: https://testws.delaware.gov:1403/dhss/HL7Service
- Content-Type — This value should be set to “soap” always.
- namespace — This is the namespace for the soap message. For Delvax systems it should be set to “http://tempuri.org/”
- soapaction — This value needs to be set to one of the two action types available for this service.
  - http://tempuri.org/ExecuteHL7Message
  - http://tempuri.org/ExecuteHL7MessageByte
- operation — This value needs to be set to one of the two operations available for this service that is related to the soapaction setting.
  - ExecuteHL7Message
  - ExecuteHL7MessageByte

NOTE: namespace, soapaction, and operation does not define the message being received, it defines the message that will be sent to the PostTo URL. If any of these settings are incorrect, you will see the error in the response back from the immunization URL in syslog.

2.2 Sample Configuration

Below is a complete example of a configuration.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="WcfServiceHost"
type="Medicity.WcfServiceHost.Configuration.ServiceHostConfigSectionHandler, WcfServiceHost"
/>
    <section name="medicity.syslog"
type="Medicity.Syslog.Configuration.MedicitySyslogSection, Medicity.Syslog" />
  </configSections>
  <WcfServiceHost>
    <!—
      The name of the service. This must be unique on the system where the service is
      installed.
    -->
    <ServiceName value="DHIN_PROD_HttpPostForwarder" />

    <!--
      The number of seconds to wait for startup.  This allows the service to
      play nicely with others, and allow other startup tasks to succeed
      before performing potentially system-intensive startup tasks.
    -->
    <SoftStartWait value="30" />

    <!--
      Services hosted by the instance.  Multiple WCF services may be
      hosted by a single instance of the WcfServiceHost.  While there
      is no hard-coded limit to the number of services hosted, there
      may be a practical limit.
    -->
    <Services>
      <!--
        Defines a single service.
          assembly  : the long form of the assembly name containing the service type
          type      : the fully-qualified type name of the WCF service
      -->
      <Service assembly="Medicity.HttpPostForwarder"
type="Medicity.HttpPostForwarder.HttpPostForwarder" />
    </Services>
  </WcfServiceHost>
  <medicity.syslog>
    <client>
      <clientConfig name="Medicity.HttpPostForwarder" throwOnFail="true" async="false">
        <udp address="10.8.8.2" port="17777" />
      </clientConfig>
    </client>
  </medicity.syslog>
  <appSettings>
    <add key="PostTo" value="https://ws.delaware.gov:1403/dhss/HL7Service" />
    <!--
      Current supported Content-Types: urlencoded, multipart/form-data and Soap
```

```
      (default: multipart/form-data)
    -->
    <!-- The configured Content-Type is for the outbound POST to the HIE -->
    <add key="Content-Type" value="soap" />
    <!-- Soap data below -->
    <add key="namespace" value="http://tempuri.org/" />
    <add key="soapaction" value="http://tempuri.org/ExecuteHL7Message" />
    <add key="operation" value="ExecuteHL7Message" />
  </appSettings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>

  <system.web>
    <compilation debug="false" />
  </system.web>

  <!--
    When deploying the service library project, the content of the config file must be added
    to the host's app.config file.  System.Configuration does not support config files for
    libraries.
  -->
  <system.serviceModel>
    <services>
      <service name="Medicity.HttpPostForwarder.HttpPostForwarder"
behaviorConfiguration="wsdl">
        <host>
          <baseAddresses>
            <add baseAddress="http://immunizations.dhin.net:30100/" />
          </baseAddresses>
        </host>
        <endpoint address="HttpPostForwarder"
                  contract="Medicity.HttpPostForwarder.IHttpPostForwarder"
                  binding="webHttpBinding"
                  behaviorConfiguration="webby">
          <identity>
            <dns value="immunizations.dhin.net" />
          </identity>
        </endpoint>
        <endpoint address="SoapForwarder"
                  binding="basicHttpBinding"
                  contract="Medicity.HttpPostForwarder.IHttpPostForwarder">
          <identity>
            <dns value="immunizations.dhin.net" />
          </identity>
        </endpoint>
        <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
      </service>
    </services>
    <behaviors>
      <endpointBehaviors>
        <behavior name="webby">
          <webHttp />
        </behavior>
      </endpointBehaviors>
```

```
      <serviceBehaviors>
        <behavior name="wsdl">
          <serviceMetadata httpGetEnabled="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <bindings>
      <basicHttpBinding>
        <binding name="basic_no_security_binding" maxReceivedMessageSize="2147483647">
          <readerQuotas maxArrayLength="2147483647" />
          <security mode="None" />
        </binding>
      </basicHttpBinding>
      <webHttpBinding>
        <binding transferMode="Streamed" />
      </webHttpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

3 WSDL

In this section you will see references that use “http” and others use “https”. Whenever you see the “http” used in the URL, it is referencing the URL for the service that is located behind the firewall and therefore does not need to be secure. When you see the “https” used in the URL, it is referencing the URL for the load balancer that is available to all external customers. The load balancer URL is a secure URL that acts as a relay between external customers and the actual service located behind the firewall. Remembering this will help with understanding the explanations below.

The wsdl will explain the service to which you will be sending messages to as well as the format of the message. The wsdl can be obtained by using the base address of the service and include the value ?singlewsdl at the end. For example: https://certimmunizations.dhin.net/?singlewsdl. In the PROD version, ?singlewsdl is not available so you will need to use just ?wsdl. I.E. https://immunizations.dhin.net/?wsdl.

NOTE: You will need to remove the port number and add the https to all the secondary addresses included in the wsdl.

An example of this is:

Public URL: https://immunizations.dhin.net/SoapForwarder/
Wsdl URL: http://immunizations.dhin.net:30100/SoapForwarder/

You will notice in the example above, the difference in the wsdl address and the public address. If these are different, always use the public URL.

You can also obtain the wsdl by using ?wsdl at the end of the base address. For example: https://immunizations.hie.net/?wsdl. The problem with this method, is if the service is placed behind a firewall, you will have a section in the wsdl like:

```
<xsd:schema targetNamespace="http://tempuri.org/Imports">
  <xsd:import schemaLocation="http://immunizations.dhin.net:30100/?xsd=xsd1"
namespace="http://tempuri.org/"/>
  <xsd:import schemaLocation="http://immunizations.dhin.net:30100/?xsd=xsd0"
namespace="http://schemas.microsoft.com/Message"/>
  <xsd:import schemaLocation="http://immunizations.dhin.net:30100/?xsd=xsd2"
namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
</xsd:schema>
```

In order to get all the definitions you will need to change the schemaLocation value to that of the public URL. For example:

`http://immunizations.dhin.net:30100/?xsd=xsd1` will need to be changed to this:
`https://immunizations.dhin.net/?xsd=xsd1`

Because of this, it is better to use the ?singlewsdl when retrieving the wsdl.

4 Soap Message Format

The soap message format for sending messages to the proxy is a standard form of SOAP. It consists of a set of HTML headers and a soap envelope. The soap envelope is an xml document that contains all the information that is to be passed to the proxy. Two types of messages can be sent. The one discussed in section 4.1 is the preferred method. The one in section 4.2 takes extra care in converting the message to a 64 bit binary string prior to sending the message.

4.1 Operation ExecuteHL7Message

HTML Headers that are expected to be sent with the message are:

```
Method: POST
SOAPAction: http://tempuri.org/ExecuteHL7Message
ContentType: text/xml; charset=utf-8
Content-Length: <Length of complete Soap Envelope>
```

The Soap Envelope should contain the following (Spaces and returns are optional except the HL7 message is required to have carriage return as a delimiter for segments):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ExecuteHL7Message xmlns="http://tempuri.org/">
      <userName>[Username Assigned]</userName>
      <password>[Password Assigned]</password>
      <flatWire>
        <![CDATA[<HL7 Message>]]>
      </flatWire>
    </ExecuteHL7Message>
  </s:Body>
</s:Envelope>
```

The tags <![CDATA[…]]> surrounding the HL7 message is to keep the message free of any html encoding is required.

The Username and Password will be passed as entered to the Immunization Registry so make sure they are correct. If you do not have a username and password, then contact the Delaware Immunizations Program at Cheryl.oliver-knight@state.de.us

Example of Sent Message (Spaces and returns added for readability):

```
Http Method: POST
SOAPAction: "http://tempuri.org/ExecuteHL7Message"
Connection: Keep-Alive
Content-Length: 1083
```

```
Content-Type: text/xml; charset=utf-8
Expect: 100-continue
Host: ltp-bwagstaff:8080

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ExecuteHL7Message xmlns="http://tempuri.org/">
      <userName>User.Name</userName>
      <password>Password</password>
      <flatWire>

<![CDATA[MSH|^~\&|PWX|DEXXXX|DelVAX|DE0000|20141010130909||VXU^V04^VXU_V04|Q870957023T1341981
32|P|2.5.1|||AL|NE
PID|1||1234^^^MRN^MR^||Star^Silly^R^^^^L||19460809000000|F||2106-3^White^HL70005|123 Any
Rd^^Dover^DE^19904^^M||^PRN^PH^^^999^9999999|^WPN^^^^999^9999999|ENG|||||||2186-5^Not
Hispanic or Latino|||0|||||N
NK1|1|Star^Silly^R^^^^L|SEL^Self|^^^^^^H|^PRN^PH^^^999^9999999
PV1|1|R||||||||||||||||||V01
IN1|1|999999^BCBShield DE|9999999|Betty Ford Immunization Clinic - Insurance
Organization||||10006450||0||20110731|21001231|||Star^Silly^R^^^^C|SEL^Self^HL70063^SEL^Self^
HL7 v2^^^Self|19460809000000|123 Any
Rd^^Dover^DE^19904|||2||||||||||||||||||||F||||||9999999999
ORC|RE||9654D549046A76FAEFAA54||||||||9999999999^Ford^Bettyl^^MD^^^^^P^^^NPI
RXA|0|1|20141010130800|20141010130800|135^influenza virus vaccine,
inactivated^CVX^^^^^^influenza virus vaccine, inactivated|0.5|mL^Milliliter^ISO+||00^NEW
IMMUNIZATION RECORD^NIP001|1111111111^Greenjeans^Mister|||||1411901|20150501|NOV^Novartis
Pharmaceutical Corporation^MVX|||CP||20141010130907
RXR|IM^im^HL70162|LD^left deltoid^HL70163
OBX|1|CE|64994-7^Vaccine funding program eligibility category^LN|1|V01^Not VFC Eligable
B^HL70064||||||F||||||VXC40^Eligibility captured at the Immunization level^CDCPHINVS
OBX|2|CE|30963-3^Vaccine Funding Source^LN|1|PHC70^Private Funds^CDCPHINVS^PHC70^Private
Funds^HL7 v2^^^Private Funds||||||F
OBX|3|TS|29768-9^Date vaccine information sheet published^LN|1|20140819||||||F
OBX|4|TS|29769-7^Date Vaccine Information sheet Presented^LN|1|20141009||||||F
]]>
      </flatWire>
    </ExecuteHL7Message>
  </s:Body>
</s:Envelope>
```

The Response you receive will come back with the same headers as the sending message. The response envelope will have the following format:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ExecuteHL7MessageResponse xmlns="http://tempuri.org/">
      <ExecuteHL7MessageResult>[Response Text or HL7Message]</ExecuteHL7MessageResult>
    </ExecuteHL7MessageResponse>
  </s:Body>
</s:Envelope>
```

4.2 Operation ExecuteHL7MessageByte

HTML Headers that are expected to be sent with the message are:

```
Method: POST
SOAPAction: http://tempuri.org/ExecuteHL7MessageByte
ContentType: text/xml; charset=utf-8
Content-Length: <Length of complete Soap Envelope>
```

The Soap Envelope should contain the following (Spaces and returns are optional except the HL7 message is required to have carriage return as a delimiter for segments):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ExecuteHL7MessageByte xmlns="http://tempuri.org/">
      <userName>[Username Assigned]</userName>
      <password>[Password Assigned]</password>
      <flatWireByte>
        <HL7 Message changed into 64 byte string>
      </flatWireByte>
    </ExecuteHL7MessageByte>
  </s:Body>
</s:Envelope>
```

The Username and Password will be passed as entered to the Immunization Registry so make sure they are correct. If you do not have a username and password, then contact the Delaware Immunizations Program. Cheryl.oliver-knight@state.de.us

The Response you receive will come back with the same headers as the sending message. The response envelope will have the following format:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ExecuteHL7MessageByteResponse xmlns="http://tempuri.org/">
      <ExecuteHL7MessageByteResult>
        [Response Text or HL7Message in 64 byte string format]
      </ExecuteHL7MessageByteResult>
    </ExecuteHL7MessageByteResponse>
  </s:Body>
</s:Envelope>
```