

Oregon ALERT IIS

HL7 Real Time Data Exchange

Version 1.4b

Last Updated: Dec 2013

Note: HL7 2.4 is the version number Oregon assigned to its implementation of the CDC HL7 2.3.1 (June 2006) Guide

INTRODUCTION.....	3
CHECKLIST FOR PROVIDER ORGANIZATIONS AND VENDORS ESTABLISHING REAL TIME DATA EXCHANGE.....	4
WEB SERVICES AND ALERT IIS AT-A-GLANCE	6
INTRODUCTION TO SOAP WEB SERVICES IN ALERT IIS	7
SSL CERTIFICATE CREATION AND INSTALLATION.....	9
Description of Fields in the CSR.....	9
Generate a Private Key and Certificate Signing Request (CSR) using Microsoft Windows	11
Step 1: Generate the Private Key and CSR.....	11
Step 2: Send CSR to ALERT Staff	12
Step 3: Receive and Install Your Signed Certificate	12
Generate a Private Key and Certificate Signing Request (CSR) using OpenSSL	14
Step 1: Generate Private Key	14
Step 2: Generate the CSR.....	14
Step 3: Send CSR to ALERT Staff	14
Step 4: Backup the private key	14
Step 5: Receive and Install Your Signed Certificate	14
TESTING THE CERTIFICATE INSTALLATION.....	16
TROUBLESHOOTING THE CERTIFICATE INSTALLATION	16
GETTING THE WSDL	19
IMPLEMENTING THE WEB SERVICE CALLS FROM YOUR ENVIRONMENT.....	21
SOAP Faults.....	21
APPENDIX A: WSDL AND XSD FILE CONTENTS.....	22
APPENDIX B: COMPLETE SAMPLE SOAP MESSAGES.....	26
FindHistory Sample	26
UpdateHistory Sample	27

Introduction

Thank you for your interest in Health Level Seven (HL7) electronic data exchange with Oregon ALERT IIS.

HL7 real time data exchange with ALERT IIS is accomplished by sending via SOAP web services. For information about sending HL7 batch, please contact the ALERT Help Desk.

The first step for a clinic and/or their vendor is to review the ALERT IIS HL7 Implementation Guide. It is important to understand the local HL7 specifications to ensure that your EHR system can prepare and send HL7 messages that meet Oregon ALERT's requirements.

Once you have reviewed the [ALERT IIS HL7 Implementation Guide](#) and the following data exchange specifications, contact the ALERT Help Desk to begin the process of setting up a data exchange relationship with ALERT IIS. Once a site has communicated to the Help Desk that they are interested in setting up HL7 real time data exchange with ALERT, a data exchange specialist will follow up.

The ALERT IIS HL7 Implementation Guide can be downloaded from our website:
https://www.alertiis.org/docs/hl7_24_gts.pdf

For questions or more information:

ALERT Help Desk
1-800-980-9431
alertiis@state.or.us

Checklist for Provider Organizations and Vendors Establishing Real Time Data Exchange

1) Planning & Design

- Review [ALERT IIS HL7 Implementation Guide](#). Please note there are a number of Oregon-specific HL7 requirements. Two of those are:
 - Clinic level code (AL#, provided by ALERT) should be sent in MSH-4, and RXA-11 (if you plan to use the inventory module).
 - Vaccine eligibility code (for VFC program or other state supplied vaccine) must be sent at the dose level (in an OBX segment), rather than at the patient visit level.
- Review current list of sites in ALERT IIS associated with your organization. Add/modify as needed. *This list will be provided by the ALERT Data Exchange Coordinator.*
- Test plans: begin writing test plans and set up test cases/patients in your test environment.

2) Development

- Per the [ALERT IIS HL7 Implementation Guide](#), map your EHR system codes to the specifications described for ALERT IIS. There are various tools available online for validating standard HL7 message format, such as <https://phinmqf.cdc.gov>.
- Develop VXU message format, per the ALERT IIS specifications.
- Develop VXQ message format, per the ALERT IIS specifications. In planning for bi-directional query functionality, please consider the following questions in your planning and development:
 - Will records be stored permanently?
 - If storing, will returned records be integrated into the EHR immunization history?
 - Will the EHR automatically query for records, or will the user initiate the query?
 - How will the returned records be displayed to the user?
- Error Message and Response File Management. Please consider the following questions in your planning and development:
 - How will the EHR manage error messages?
 - How will error messages be displayed to the user?
 - Who will review error messages?
 - How will providers be able to correct errors and re-submit?
 - If there is web service downtime for maintenance or outage, what is the plan for how records will be re-sent?

3) Check In Call with ALERT IIS Data Exchange Team

- The purpose of this call is to review design and development steps, determine readiness and review next steps for testing and deployment. Please contact Tracy Little at tracy.c.little@state.or.us

4) Testing

- Create CSR and Install SSL Certificate for access to ALERT UAT (ALERT IIS test environment), from your test environment. See detailed instructions in this document the section, "*SSL Certificate Creation and Installation.*"
- Install SOAP web services WSDL (Web Services Definition Language) for UAT environment. The WSDL file will be provided by the ALERT IIS Data Exchange Coordinator or one of the methods outlined in the section, "*Getting the WSDL.*"
- Submit one day, then one month of immunization data to the ALERT IIS UAT environment via the web service.
- Conduct internal QA and validation of test messages. Suggested review criteria are as follows:
 - Find out how many successful, partially successful, and failed VXU messages are sent.
 - Identify major issues in failed and partially successfully messages; submit another set (month) of data until issues are resolved. Repeat this step until nearly all messages are successful.
 - Check the User Interface to ensure that the immunizations reported in the messages were added to the patients' records.
 - Check the messages and UI to ensure that historical immunizations and recommended fields are being sent.
 - Run the percentage of each vaccine and group reported during the one-month test. Ensure that percentages do not vary greatly from the amount of reporting we normally expect.
- Review results of testing with ALERT IIS Data Exchange Coordinator. Schedule Go-Live.

5) Deployment

- Create CSR and Install SSL Certificate for access to ALERT Production, from your production environment. See detailed instructions in this document the section, "*SSL Certificate Creation and Installation.*" The certificates are not shared between UAT and Production except under special circumstances.
- Install the SOAP web services WSDL (web services definition language) for Production environment. The WSDL file will be provided by the ALERT IIS Data Exchange Coordinator or one of the methods outlined in the section, "*Getting the WSDL.*"
- Coordinate date for go-live with ALERT data exchange coordinator. If currently sending electronic data to ALERT plans must be made to suspend current submission by go-live date.
- Validate data going to ALERT IIS production site and review with ALERT IIS Data Exchange Coordinator.

Web Services and ALERT IIS At-A-Glance

1. The addresses for the Web Service endpoints are:

Test: <https://soa.alertiis.org/webservices/VaccinationBService>
Production: <https://soa.alertiis.org/webservices/VaccinationBService>

To retrieve the WSDL, place the text "?wsdl" (without the quotes) at the end.

2. Web Services security relies on the client (computer making a request) and the server both having 2048-bit certificates installed.
3. Asymmetric public key technology is used for encryption and decryption of messages.
4. The Certificate Authority (CA) used is HP, using a self-signed certificate.
5. If you have more than one environment (Test, Production, Staging), each one will need its own Certificate.
6. SOAP messaging is the method used to communicate with ALERT IIS Web Services. Two web methods are supported, each containing a single argument that is the corresponding HL7 message:

<u>Web Method</u>	<u>HL7 Message Sent as Argument Arg0</u>
FindHistory	VXQ
UpdateHistory	VXU

More details about the HL7 messages can be found in the ALERT IIS HL7 Implementation Guide found at https://www.alertiis.org/docs/hl7_24_gts.pdf.

7. HL7 2.4 is the version number Oregon assigned to its local implementation of CDC's HL7 2.3.1. It follows the CDC version with a few minor differences outlined in the Implementation Guide.
8. General overview of steps to establish a secure connect to ALERT IIS Web Services:
 - a. Generate a Certificate Signing Request (CSR) from the client computer
 - b. Receive back HP's root Certificate Authority certificate and your individual certificate
 - c. Install both certificates on the client computer
 - d. Using the WSDL (Web Services Definition Language) description of the Web methods available, either:
 - i. Configure your product to send messages to ALERT IIS
 - ii. Use a programming environment such as Java or .net to develop an interface to send messages to ALERT IIS
9. Primary contact for information or help with Web Services:

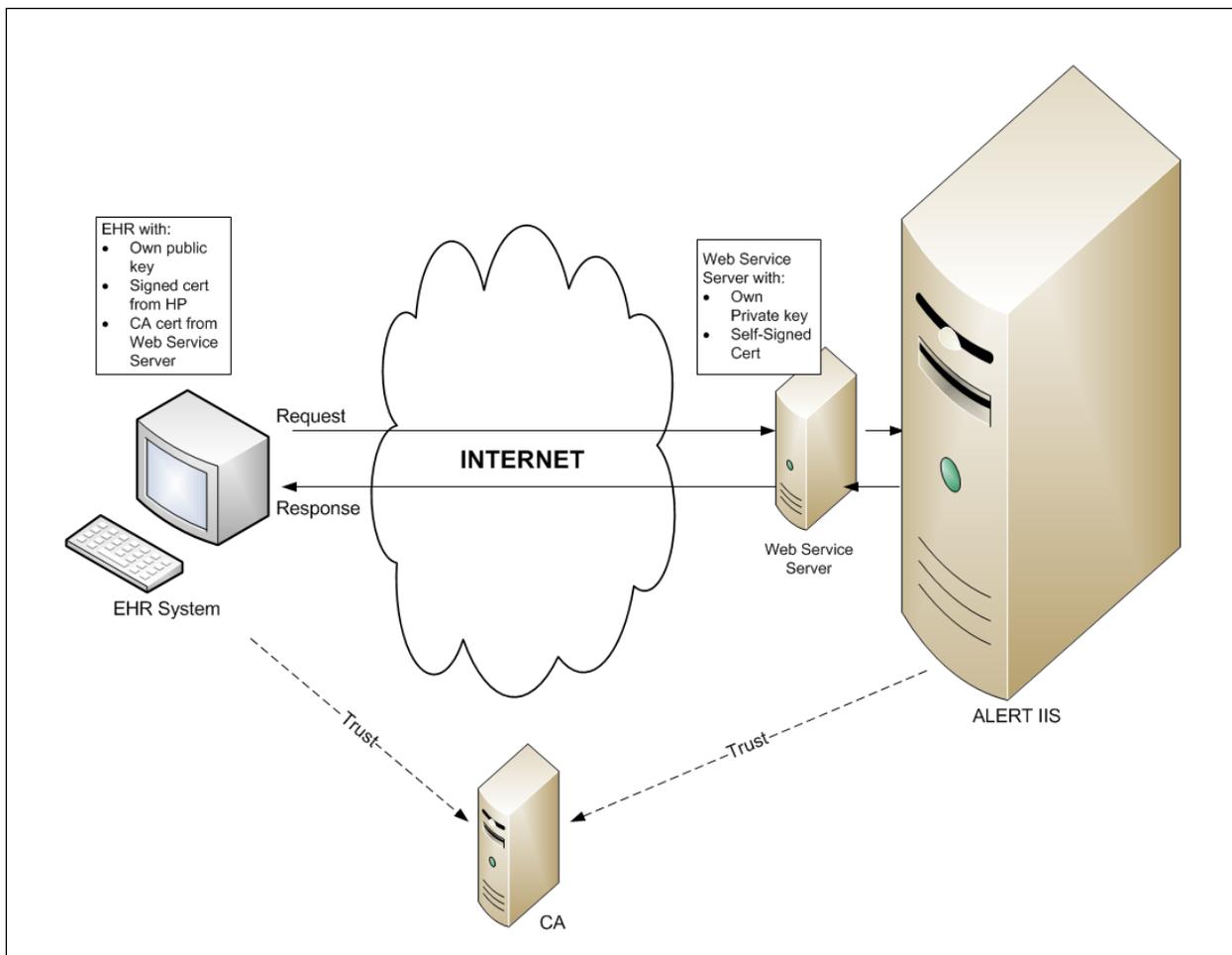
ALERT Help Desk
1-800-980-9431
alertiis@state.or.us

Introduction to SOAP Web Services in ALERT IIS

A Web Service is a standards-based method of allowing one computer to access functions in another computer through the Internet. The functions are accessed through the same ports used by Internet browsers, making them very likely to be allowed through firewalls.

Security is implemented by installing public and private key pairs (known as X.509 certificates) on both the ALERT IIS Web Services Server and the Electronic Health Record (EHR) computer requesting the information. After installation, the certificates reside within a Certificate Keystore which the system automatically accesses when trying to reach an encrypted web site such as the ALERT IIS Web Service.

Using this technique, both the client and server are identified to each other to establish trust. The communication between the computers is encrypted using a technology known as Secure Sockets Layer (SSL), a commonly-used protocol for managing the security of a message transmission on the Internet.



ALERT IIS supports the following functions through Web Services:

1. **FindHistory**, used to query the IIS for immunization data
2. **UpdateHistory**, used to send information about an administered vaccine

Web Service functions are called using Simple Object Access Protocol (SOAP) requests, which are formatted as XML (eXtensible Markup Language) messages.

Each SOAP request is made up of the following elements:

1. The envelope, which identifies the message as a SOAP request
2. The function name
3. The parameters of the function call. In most cases, the parameters are individual pieces of data, such as First Name, Last Name, etc. For the IIS Web Services, there is a single parameter which contains an entire HL7-formatted message, wrapped in a CDATA section to keep it from being misinterpreted by the parser.

Here is a simple VXU message sent to the UpdateHistory Web Service function. The pieces are numbered to match the list above.

(1) SOAP Envelope Start	<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:vac="http://vaccination.org/"> <soap:Header/> <soap:Body>
(2) Function and parameter Start	<vac:UpdateHistory> <arg0>
CDATA section Start	<![CDATA[
(3) Parameter, the HL7 message, segments vary by function	MSH ^~\& AL9999 ALERTIIS^^^ 20110201 VXU^V04 682299 P^ 2.4^^^ AL PID 79928^^^^PI A5SMIT0071^^^^^ SMITH^MAY^^^^^^ JOHN^^^^^^^ 20101212 F RXA 0 999 20110201 20110101 ^^^90701^DTP^CPT 0.5
CDATA section End]]>
(2) Function and parameter End	</arg0> </vac:UpdateHistory>
(1) SOAP Envelope End	</soap:Body> </soap:Envelope>

The definitions of the functions are specified in the WSDL (Web Services Definition Language) file. Modern development environments (such as Java and .NET) can take the WSDL and turn it into a programming interface to simplify implementation. The WSDL can be supplied by the Oregon Immunization Program or retrieved from the server once the certificates are installed.

For More Information

<http://www.w3schools.com/webservices/default.asp> is a good source for information about Web Services. The Summary page includes links to information about WSDL and SOAP.

SSL Certificate Creation and Installation

To enable the secure transmission of data between your computer and the IIS Web Service, we require that both the client and server install certificates that have been generated by the HP Immunization Services personnel. To accomplish this, you must first create a private key for each machine that will be accessing the Alert IIS web services machine. This private key is then used to create a Certificate Signing Request (CSR) which will be sent to HP. HP will create the SSL certificate which will be returned to you for installation on your client machine ("client" in this instance will most likely be the server that communicates with the Alert IIS web services servers).

To generate a CSR, a key pair must be created for the server. These two items are a digital certificate key pair and cannot be separated. If the public/private key file is lost or changed before the SSL certificate is installed, the SSL certificate will need to be re-issued. The private key, CSR, and certificate must all match in order for the installation to be successful.

The following sections outline two methods of generating the required 2048-bit key:

1. Using Internet Information Server in Microsoft Windows server products
2. Using OpenSSL software, common in most Unix-style Operating Systems and available for Windows

Make sure that any existing keys and CSR's are NOT overwritten.

Description of Fields in the CSR

Use this as a reference when filling out the information for both the Windows and OpenSSL methods.

Field	Required / Optional / Not Allowed	Description
Country Name or Country/Region	R	Use the two-letter code without punctuation for country. Example: US or CA
State or Province	R	Spell out the state completely; do not abbreviate the state or province name. Example: Oregon
Locality or City	R	The Locality field is the city or town name; do not abbreviate. (Example: Saint Louis, not St. Louis)
Company or Organization	R	If the company or department has an &, @, or any other symbol using the shift key in its name, the symbol must be spelled out or omitted. Example: XY & Z Corporation would be XYZ Corporation or XY and Z Corporation.
Organizational Unit	O	Can be used to help identify certificates registered to an organization. The Organizational Unit (OU) field is the name of the department or organization unit making the request.
Common Name	R	The Common Name is the Host + Domain Name. It looks like "my.server.com". Certificates can only be used on Web servers using the Common Name specified during enrollment. For example, a certificate for the domain "server.com" will receive a warning if accessing a site named "www.server.com" or "secure.server.com", because

		"www.server.com" and "secure.server.com" are different from "server.com".
Email Address	NA	Please do not enter anything in this field.
Challenge Password	NA	Please do not enter anything in this field.
Optional Company Name	NA	Please do not enter anything in this field.

Generate a Private Key and Certificate Signing Request (CSR) using Microsoft Windows

These steps must be completely on the web server that will be interacting with the IIS Web Services Server.

Step 1: Generate the Private Key and CSR

1. Click **Start, Settings, Control Panel**, then choose **Administrative Tools** and finally **Internet Information Services (IIS) Manager**.
2. Open the **(local computer)** entry in the treeview on the left of the window. It will be preceded by the name of the server.
3. Check which version of IIS you are running by looking at the column in the right-hand pane. In this example, it is "IIS V6.0".

Computer	Local	Version	Status
DHSC1059292 (local computer)	Yes	IIS V6.0	

4. If it indicates IIS version 6, follow these steps:
 - a. Click on the entry for **Web Sites**.
 - b. Right-click the entry for **Default Web Site**, click **New**, and then click **Site**. Create a new site and give it a temporary name such as **Cert**. Accept the defaults for ports, and set the Path to a subdirectory below your **C:\Inetpub\wwwroot** directory, such as **C:\Inetpub\wwwroot\Cert**. Accept defaults for permissions.

Windows XP Users: If you are running on Windows XP, only one Web Site is allowed. You will need to create the CSR from the existing Default Web Site, but please be careful to preserve existing certificates.

- c. Right-click this new site in the treeview, click **Properties**, click the **Directory Security** tab, and then click the **Server certificate** button. Click **Next**.
 - d. Select **Create a new certificate** and follow the wizard to create a new CSR. Select **Prepare the request now but send it later**.
 - e. Go to Step 6 for helping filling out the data elements.
5. If it indicates IIS version 7, follow these steps:
 - a. In the IIS Manager, select the server node on the top left under **Connections**.
 - b. In the **Features** pane (the middle pane), double-click the **Server Certificates** option located under the IIS or Security heading (depending on your current group-by view).
 - c. In the **Actions** pane on the top right, select **Create Certificate Request**. The dialog box opens.
 6. Fill out the data elements in the Wizard:
 - Name: Use of the domain name or IP address that will be used for the certificate as the prefix of the name
 - Bit length: Must be 2048
 - File name: Use of the domain name or IP address that will be used for the certificate as the prefix of the filename

Other fields: See the section, "Description of common fields in the CSR"

7. At this point, your server has retained your Private key and is awaiting the next step to install the Certificate.

Step 2: Send CSR to ALERT Staff

The CSR is an ASCII text file that can be attached to an email and should be sent to the ALERT IIS Data Exchange Coordinator. Please send your CSR file to the ALERT IIS Data Exchange Coordinator, Tracy Little (tracy.c.little@state.or.us)

Step 3: Receive Your Signed Certificate

Once HP has finished processing your CSR, you will receive a zip file via email which contains two files:

1. The CA certificate, **ca.crt**. This is common to all IIS users and establishes HP's server as a valid Certificate Authority, which tells your server to trust Certificates issued by HP.
2. The specific server certificate, such as **csr-site-request.crt**. This is unique to the server it was generated from.

IMPORTANT! These certificate files are not complete as emailed from HP. They must be combined with your private key and installed to become active. The Windows wizards will do this combining automatically.

Step 4: Install Your Signed Certificate

Install the signed certificate and certificate authority files mailed by HP, so the browser and Operating System can use it.

1. Extract the zip file in the email to a location on the local drive.
2. Double-click the **ca.crt** file to install it. A dialog will appear displaying Certificate Information.
3. Click the **Install Certificate** button at the bottom of the dialog, and accept all of the defaults.
4. Confirm a message appears at the end that reads, **The import was successful.**
5. Click **Start, Settings, Control Panel**, then choose **Administrative Tools** and finally **Internet Information Services (IIS) Manager**.
6. Open the **(local computer)** entry in the treeview on the left of the window. It will be preceded by the name of the server.
7. Click the **Web Sites** entry in the treeview.
8. Locate the Site you added during the CSR process.

9. Right-click the site you added in the treeview, click **Properties**, click the **Directory Security** tab, and then click the **Server certificate** button. The message should indicate a status of "You have a pending certificate request."
10. Click **Next**.
11. Choose **Process the pending request and install the certificate**.
12. Click the Browse button and point to the second .crt file you unzipped to the directory in Step 1, such as **csr-site-request.crt**.
13. Windows will combine this certificate with the private key retained from the CSR, and install it into the **Personal** category of Certificates.

Generate a Private Key and Certificate Signing Request (CSR) using OpenSSL

This option is normally available from Unix-style Operating Systems. OpenSSL can also be downloaded for the Windows environment from www.openssl.org, but the Windows method may be easier for most users. Specifically, a Windows installation file can be obtained through <http://www.openssl.org/related/binaries.html>.

Step 1: Generate Private Key

Type the following command at the prompt:

```
openssl genrsa -out my.server.com.key 2048
```

This command generates a 2048 bit RSA private key and stores it in the file, my.server.com.key

Note: For all SSL certificates, the CSR key bit length must be 2048.

Step 2: Generate the CSR

Type the following command at the prompt:

```
openssl req -new -key my.server.com.key -out my.server.com.csr
```

This command will prompt for the following attributes of the certificate as shown in the section of this document, "Description of Fields in the CSR."

A public/private key pair has now been created. The private key (my.server.com.key) is stored locally on the server machine and is used for decryption (DON'T LOSE IT). The public portion, in the form of a Certificate Signing Request (my.server.com.csr), will be for certificate enrollment.

Step 3: Send CSR to ALERT Staff

The CSR is an ASCII text file that can be attached to an email and should be sent to the ALERT IIS Data Exchange Coordinator. Please send your CSR to the ALERT IIS Data Exchange Coordinator, Tracy Little (tracy.c.little@state.or.us)

Step 4: Backup the private key

It is recommended that you back-up the .key file. A good choice is to create a copy of this file onto a diskette or other removable media. While backing up the private key is not required, having one will be helpful in the instance of server failure.

Step 5: Receive and Install Your Signed Certificate

Once HP has finished processing your CSR, you will receive a zip file via email which contains two files:

1. The CA certificate, **ca.crt**. This is common to all IIS users and establishes HP's server as a valid Certificate Authority, which tells your server to trust Certificates issued by HP.

2. The specific server certificate, such as **csr-site-request.crt**. This is unique to the server it was generated from.

IMPORTANT! These certificate files are not complete as emailed from HP. They must be combined with your private key (created and kept during the CSR generation step) and installed to become active.

The instructions for this will vary depending on your environment. A good source of reference for this information is Google (search for: importing trusted root certificates). Two common trusted stores are Public-Key Cryptography Standards #12 (PKCS#12 or PFX) and Java Key Store (JKS).

Join your private key with the signed certificate and certificate authority files mailed by HP, so the browser and Operating System can use it.

Here is an example of creating a .pfx file using openssl:

```
Openssl pkcs12 -export -out www.example.com.pfx -inkey www.example.com.key -  
in www.example.com.crt -certfile cacert.crt
```

Here is what the example file names represent:

www.example.com.pfx = this will be the output file – which you’ll install into Windows 7 so IE can use it, etc

www.example.com.key = this is the key that was generated by step 1

www.example.com.crt = this is the signed certificate provided in response to your CSR

cacert.crt = this is the CA (Certificate Authority) file which was provided, this is needed by openssl to verify we truly signed the first file.

Testing the Certificate Installation

Once the installation of the Certificate is complete, you should be able to browse to the Web Server and retrieve the WSDL information. Without the Certificates, the browser will display an error.

1. Open Internet Explorer.
2. Enter the appropriate address into the browser address bar.

Test: <https://soa.alertiis.org/webservices/trn/VaccinationBService?wsdl>

Production: <https://soa.alertiis.org/webservices/VaccinationBService?wsdl>

If you leave off the trailing “?wsdl” you will get a page of endpoints, rather than the WSDL itself.

3. The very first time, you may be asked to select the Certificate that should be used to communicate with the IIS web server. Choose the one issued by 64.73.37.141. This question may come up more than once.
4. Check the result.
 - a. Success: the WSDL appears in the web browser as shown in Appendix A and can be printed for reference.
 - b. Failure: If something is not working, you are most likely to see this message, but you may also see an error about the certificate not being correct.



Troubleshooting the Certificate Installation

Things to consider while troubleshooting a Certificate issue:

1. Confirm the Certificates were properly joined with the private key from the CSR step before they were installed into the Certificate store. Windows does this automatically when you do the steps through the Microsoft Internet Information Services Manager.
2. Test Internet Port 443 is open by connecting to another secure site, such as mail.google.com.
 - a. If you cannot reach Google Mail’s login page, your firewall or another device may be blocking the port.
 - b. If you can reach Google Mail’s login page, consider whether specific Internet addresses are being blocked by a firewall, either in the server or on a firewall device.

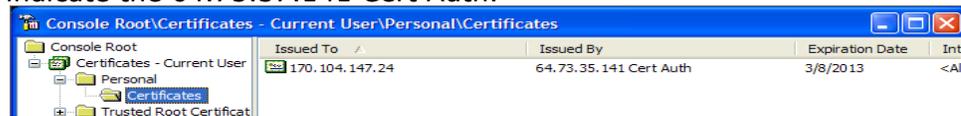
3. Confirm the Certificates are appearing in the proper area of the system. For Windows, this can be checked with the following steps.

Note: Not all Certificates appear when viewing them through the Content tab of Internet Options, so this method is suggested for Windows.

- a. Click **Start, Run**, and enter **MMC**.
- b. The Console appears.
- c. Click **File, Add/Remove Snap-In...**
- d. Click **Add...** at the bottom of the dialog box to open the Add Standalone Snap-In.
- e. Click **Certificates** and click the **Add** button.
- f. Confirm **My user account** is selected.
- g. Click **Finish**.
- h. **Close** the Add Standalone Snap-In dialog box.
- i. Click **OK** to accept the Add/Remove Snap-In dialog box. The Console should appear with an entry in the treeview for Certificates – Current User.
- j. The Root Certificate
 - i. Click the tree entry for **Trusted Root Certification Authorities**.
 - ii. Click the folder **Certificates** that appears below it.
 - iii. The “Issued By” field should indicate the 64.73.37.141 Cert Auth.

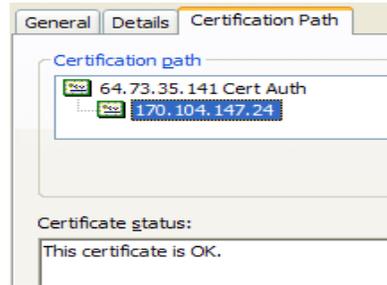


- k. Your Personal Certificate
 - i. Expand the tree for **Certificates – Current User**.
 - ii. Expand the tree for **Personal**.
 - iii. Click the folder **Certificates** that appears below it.
 - iv. The certificate should appear in the list. The Issued By field should indicate the 64.73.37.141 Cert Auth.



- v. Double-click the certificate to open the details about it.
- vi. Click the **Certification Path** tab.

- vii. Confirm the Root Certificate and your Certificate are shown. When you click each it should show a status of, "The certificate is OK."



Getting the WSDL

The WSDL (Web Services Definition Language, pronounced *whiz-dul*) is a method used to describe Web Services functions that can be accessed through the Internet from other computers.

The contents of the WSDL and related XSD file are included in Appendix A, for reference.

Using the WSDL file, modern development environments (such as Java and .NET) can turn the WSDL file into a programming interface to simplify implementation.

The WSDL can be obtained in three ways:

1. Retrieve it from the Web Service Server
2. Get a copy from the Oregon Immunization Department
3. Extract the UAT version from this document

Traditional methods such as .NET's disco command will not work with the IIS Web Service because of the additional layer of security from the certificates.

Method 1: Retrieve the WSDL from the Web Service Server

Prerequisites: This method requires that you have already requested, received and installed the certificates to reach the Web Server with full authentication and encryption.

This address can be used to retrieve the WSDL into a web browser or a development tool to generate a programming interface. The steps for working with the programming interface are beyond the scope of this documentation.

1. Enter the appropriate address into the browser address bar or the development tool.

TRN: <https://soa.alertiis.org/webservicestrn/VaccinationBService?wsdl>

Production: <https://soa.alertiis.org/webservices/VaccinationBService?wsdl>

If you leave off the trailing "?wsdl" you will get a page of endpoints, rather than the WSDL itself.

2. The WSDL appears in the web browser and can be printed for reference.
3. To get the related XSD file, do the same thing with this address. This will only be needed to view the files from a web browser, because most applications using a WSDL will automatically download and use the XSD file.

TRN: <https://soa.alertiis.org/webservicestrn/VaccinationBService?xsd=1>

Production: <https://soa.alertiis.org/webservices/VaccinationBService?xsd=1>

4. The XSD appears in the web browser and can be printed for later reference.

Method 2: Get a copy from the Oregon Immunization Department

Prerequisites: None.

Contact the ALERT Help Desk (alertiis@state.or.us) and request electronic copies of the WSDL file. This will provide the initial IIS-WSDL-Dev.xml and IIS-WSDL-Prod.xml files, which contain the Web Service Definitions. These are the same files that could be copied out of this document and pasted into text files, as shown in Method 3.

Method 3: Extract the UAT version from this document

Prerequisites: None.

1. Highlight the lines in Appendix A (the WSDL section only) from
`<?xml version="1.0" encoding="UTF-8" ?>`
to
`</definitions>`
2. Copy the highlighted text to the clipboard with Control-C.
3. Open up Notepad and paste the contents of the clipboard into the empty Notepad file.
4. Verify the contents include all of the lines you highlighted in Step 1.

Implementing the Web Service Calls from Your Environment

Once the WSDL is obtained as outlined in the previous section, the work of setting up your environment to call the Web Service can begin. For most EHR products, this means configuring the system to send HL7 messages via a SOAP message to IIS. For others, software development may be required to complete the transmission of the data to IIS.

The rest of this section outlines considerations when building the link between your EHR and IIS.

SOAP Faults

SOAP Faults are generated when an error condition occurs. There are four types of SOAP Faults in the IIS Web Service:

1. **UnsupportedOperationFault_Message** – generated if the sender attempts to request an operation that is not part of the IIS SOAP Web Service.
2. **SecurityFault_Message** – generated if the authentication credentials supplied in the submitSingleMessage operation are not validated.
3. **MessageTooLargeFault_Message** – generated if the hl7Message parameter of the submitSingleMessage operation is too large.
4. **UnknownFault_Message** – Any SOAP fault that does not fit into one of the above three SOAP Fault categories will be returned as an “unknown” fault.

Appendix A: WSDL and XSD File Contents

The following are the contents of the UAT version of the WSDL. The Production WSDL is identical, except for the location tags which show the Web Service endpoint.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.1-hudson-28-. -->
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.1-hudson-28-. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-
policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns="http://vaccination.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://vaccination.org/" name="VaccinationBService">
  <wsp:Policy xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
wsu:id="WS2007FederationHttpBinding_IVaccinationServiceBindingPolicy">
    <sp:SignedEncryptedSupportingTokens>
      <wsp:Policy>
        <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
          <wsp:Policy>
            <sp:WssUsernameToken10 />
          </wsp:Policy>
        </sp:UsernameToken>
      </wsp:Policy>
    </sp:SignedEncryptedSupportingTokens>
    <sp:TransportBinding>
      <wsp:Policy>
        <sp:AlgorithmSuite>
          <wsp:Policy>
            <sp:Basic128 />
          </wsp:Policy>
        </sp:AlgorithmSuite>
        <sp:IncludeTimestamp />
        <sp:Layout>
          <wsp:Policy>
            <sp:Lax />
          </wsp:Policy>
        </sp:Layout>
        <sp:TransportToken>
          <wsp:Policy>
            <sp:HttpsToken RequireClientCertificate="false" />
          </wsp:Policy>
        </sp:TransportToken>
      </wsp:Policy>
    </sp:TransportBinding>
  <sp:Wss10 />
</wsam:Addressing />
```

```

</wsp:Policy>
<types>
  <xsd:schema>
    <xsd:import namespace="http://vaccination.org/" schemaLocation="
https://soa.alertiis.org/webservices/trn/VaccinationBService?xsd=1" />
  </xsd:schema>
</types>
<message name="UpdateHistory">
  <part name="parameters" element="tns:UpdateHistory" />
</message>
<message name="UpdateHistoryResponse">
  <part name="parameters" element="tns:UpdateHistoryResponse" />
</message>
<message name="Exception">
  <part name="fault" element="tns:Exception" />
</message>
<message name="FindHistory">
  <part name="parameters" element="tns:FindHistory" />
</message>
<message name="FindHistoryResponse">
  <part name="parameters" element="tns:FindHistoryResponse" />
</message>
<portType name="IVaccinationService">
  <operation name="UpdateHistory">
    <input wsam:Action="http://vaccination.org/IVaccinationService/UpdateHistoryRequest" message="tns:UpdateHistory" />
    <output wsam:Action="http://vaccination.org/IVaccinationService/UpdateHistoryResponse" message="tns:UpdateHistoryResponse" />
    <fault message="tns:Exception" name="Exception" wsam:Action="http://vaccination.org/IVaccinationService/UpdateHistory/Fault/Exception"
/>
  </operation>
  <operation name="FindHistory">
    <input wsam:Action="http://vaccination.org/IVaccinationService/FindHistoryRequest" message="tns:FindHistory" />
    <output wsam:Action="http://vaccination.org/IVaccinationService/FindHistoryResponse" message="tns:FindHistoryResponse" />
    <fault message="tns:Exception" name="Exception" wsam:Action="http://vaccination.org/IVaccinationService/FindHistory/Fault/Exception" />
  </operation>
</portType>
<binding name="WS2007FederationHttpBinding_IVaccinationServiceBinding" type="tns:IVaccinationService">
  <wsp:PolicyReference URI="#WS2007FederationHttpBinding_IVaccinationServiceBindingPolicy" />
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="UpdateHistory">
    <soap12:operation soapAction="" />
    <input>
      <soap12:body use="literal" />
    </input>
    <output>
      <soap12:body use="literal" />
    </output>
  </operation>

```

```
</output>
<fault name="Exception">
  <soap12:fault name="Exception" use="literal" />
</fault>
</operation>
<operation name="FindHistory">
  <soap12:operation soapAction="" />
  <input>
    <soap12:body use="literal" />
  </input>
  <output>
    <soap12:body use="literal" />
  </output>
  <fault name="Exception">
    <soap12:fault name="Exception" use="literal" />
  </fault>
</operation>
</binding>
<service name="VaccinationBService">
  <port name="WS2007FederationHttpBinding_IVaccinationService" binding="tns:WS2007FederationHttpBinding_IVaccinationServiceBinding">
    <soap12:address location=" https://soa.alertiis.org/webservices/trn/VaccinationBService" />
  </port>
</service>
</definitions>
```

The following are the contents of the UAT XSD file, which is retrieved automatically when the main WSDL is imported into a development tool.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.1-hudson-28. -->
<xs:schema xmlns:tns="http://vaccination.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0" targetNamespace="http://vaccination.org/">
  <xs:element name="Exception" type="tns:Exception" />
  <xs:element name="FindHistory" type="tns:FindHistory" />
  <xs:element name="FindHistoryResponse" type="tns:FindHistoryResponse" />
  <xs:element name="UpdateHistory" type="tns:UpdateHistory" />
  <xs:element name="UpdateHistoryResponse" type="tns:UpdateHistoryResponse" />
  <xs:complexType name="UpdateHistory">
    <xs:sequence>
      <xs:element name="arg0" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UpdateHistoryResponse">
    <xs:sequence>
      <xs:element name="return" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Exception">
    <xs:sequence>
      <xs:element name="message" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FindHistory">
    <xs:sequence>
      <xs:element name="arg0" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FindHistoryResponse">
    <xs:sequence>
      <xs:element name="return" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

APPENDIX B: COMPLETE SAMPLE SOAP MESSAGES

The following messages are samples of the entire SOAP message. Both of these have been edited to remove the security elements, so the content-length will no longer be accurate.

FindHistory Sample

```
POST https://soa.alertiis.org/webservices/VaccinationBService HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/soap+xml;charset=UTF-8;action="http://vaccination.org/IVaccinationService/FindHistoryRequest"
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 64.73.37.134
Content-Length: 1406
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:vac="http://vaccination.org/">
  <soap:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security soap:mustUnderstand="true" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-4" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>user</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">password</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary">xxxxxxxxxxxx</wsse:Nonce>
        <wsu:Created>2011-12-21T01:01:06.171Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:Action>http://vaccination.org/IVaccinationService/FindHistoryRequest</wsa:Action>
    <wsa:MessageID>uuid:f8740f51-646c-471d-b02e-811dbe25abc3</wsa:MessageID>
  </soap:Header>
  <soap:Body>
    <vac:FindHistory>
      <arg0>
```

```
MSH|^~\&|AL9999|ALERTIIS|20111228111035-0800||VXQ^V01|122811100469|P|2.4|||ER
QRD|20111201|R|I|107327|||10^RD|01^SMITH^MARY^^|VXI^Vaccine Information^HL700048|ALERTIIS|
QRF|ALERTIIS|||~20110101~~~~^^~~~~|^
```

```
    </arg0>
  </vac:FindHistory>
</soap:Body>
</soap:Envelope>
```

UpdateHistory Sample

POST https://soa.alertiis.org/webservices/VaccinationBService HTTP/1.1
Accept-Encoding: gzip, deflate
Content-Type: application/soap+xml;charset=UTF-8;action="http://vaccination.org/IVaccinationService/UpdateHistoryRequest"
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 64.73.37.134
Content-Length: 2750

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:vac="http://vaccination.org/">
  <soap:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security soap:mustUnderstand="true" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-2" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>user</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">xxxxxxxxxxxx</wsse:Nonce>
        <wsu:Created>2011-12-22T21:55:18.593Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:Action>http://vaccination.org/IVaccinationService/UpdateHistoryRequest</wsa:Action>
    <wsa:MessageID>uuid:7e2efb25-077c-4982-b6ee-a21463437094</wsa:MessageID>
  </soap:Header>
  <soap:Body>
    <vac:UpdateHistory>
      <arg0>
MSH|^~\&|AL9999|Immalert|NW|20120322101621|6997|VXU^V04|1875186|P|2.4||AL|||||
PID|||12999996^^^001009^MR||DOE^JANE^^^^||20041031|F|||800 NE OREGON ST^^PORTLAND^OR^97232^USA^RS^^250|250|(000)000-
0000|(503)555-5555||U||MEDICAID|000000001|||33^AMERICAN US|||||N
NK1|1|SAM^^^|OTH|Test Dr^^PORTLAND^OR^97211^USA|(502)666-6666|(503)777-7777|||||
NK1|2|JOHN^^^|||||
RXA|0|999|20120322|20120322|03^MMR (MEASLES, MUMPS,
RUBELLA)^CVX|1|ML||00|6997^SMITH^SAM^^|AL1000|||0255AA|20130221|MERCK^MERCK^99232|||A|||22465253^^
RXR|SQ^SUBCUTANEOUS^99231|RA^RIGHT ARM^99230
OBX|1|CE|30963-3^Vaccine purchased with^LN^^|V02^Medicaid, OHP^ALERTIIS||||F

      </arg0>
    </vac:UpdateHistory>
  </soap:Body>
</soap:Envelope>
```