Web Service Integration

SOAP Service

Web Services Element

Web services are a common way for any kind of application to communicate with externally hosted servers to retrieve information or send notification events in a standard manner. Voice applications that access a web service can use the *Web Services* element.

- **Web Services Element—**A special action element used to interface with a web service.

The Web Services element is an action element so it has the same features as the action element: it does not affect the call flow and has a single exit state. The Web Services element, however, has a more complex configuration than a standard action element. Call Studio renders this configuration with its own special interface.

One unique feature of the Web Services element is its ability to configure itself at design time. This is done by loading a Web Services Description Language (WSDL) file. A WSDL file is an XML file that defines the operations supported by the web services server. It is necessary in order to define the inputs required by the service that must be entered by the designer and the outputs returned by the service that can then be stored for use later in the application.

For much more detailed information about how to use the Web Services element, refer to the Call Studio online help.

Rest Service

Rest_Client Element

Cisco Unified Call Studio includes a new element called the Rest_Client element. The Rest_Client element provides a flexible interface in order to interact with REST endpoints. The communication between the REST client and server is made completely secure using two-way Secure Sockets Layer (SSL). The Rest_Client element permits users to send GET, POST, PUT, or DELETE requests to application servers.

Set Flag to False

REST uses the boolean flag **Ignore Certificate Validation** to validate the certificate. The flag can be set to True or False. If the flag is set to False, the client checks for a valid server certificate in its keystore. If the certificate is not found, an error message appears.

The **Ignore Certificate Validation** flag checks for the availability of a valid certificate in the following key stores:

- Call Studio in debug mode: `C:\Cisco\CallStudio\eclipse\jre\lib\security\cacerts`
- Call Studio in VXML Server: `C:\Cisco\CVP\jre\lib\security\cacerts`

Note Before you validate, ensure that the required certificate is in the respective keystore.

Import Certificate in Debug Mode

Procedure

Step 1 Copy the server certificate file manually to the client machine.

Step 2 From the command prompt, navigate to `C:\Cisco\CallStudio\eclipse\jre\bin`.

Step 3 Run the following command to import the server certificate to the client keystore:

```
keytool.exe -importcert -file <manually copied server certificate file> -keystore
c:\Cisco\CallStudio\eclipse\jre\lib\security\cacerts
```

The certificate is imported to the client keystore with the default alias name **mykey** and password **changeit**.

Step 4 Run the following command to check whether the certificate is imported.

```
keytool.exe -list -keystore c:\Cisco\CallStudio\eclipse\jre\lib\security\cacerts.
```

Import Certificate in VXML Server

Procedure

Step 1 Copy the server certificate file manually to the client machine.

Step 2 From the command prompt, navigate to `C:\Cisco\CallStudio\eclipse\jre\bin`.

Step 3 Run the following command to import the server certificate to the client keystore:

```
keytool.exe -importcert -file <path to manually copied server certificate file> -keystore
C:\Cisco\CVP\jre\lib\security\cacerts
```

The certificate is imported to the client keystore with the default alias name **mykey** and password **changeit**.

Step 4 Run the following command to check whether the certificate is imported.

```
keytool.exe -list -keystore C:\Cisco\CallStudio\eclipse\jre\lib\security\cacerts
```

Step 5 Restart the VXML Server after importing the certificate and then run the call flow.

Create One-Way Communication Between VXML and REST Server

One-way secure communication imports the REST Server Certificate Authority (CA) certificate into the VXML server trust store, if CA is not available by default.

Perform the following steps to import the REST Server CA certificate into the VXML server:

Procedure

Step 1 Use the Java key tool to export the CA certificate from the REST Server.

Step 2 Copy the exported CA certificate file from the REST Server to the VXML Server.
For example: <RESTServer_ca_cert>

Step 3 From the command prompt, run the following command to import the REST Server CA certificate into the VXML truststore:

```
..\..\bin\keytool -importcert -keystore <path to the VXML Truststore> -alias <alias name>
-file <Path to RESTServer_ca_cert>
```

File path to VXML truststore: `%CVP_HOME%\jre\lib\security\cacerts`. The default password is changeit.

Note For a self- signed certificate, export the ca_cert from the REST Server and the self-signed certificate. Then, import this self-signed certificate in the VXML Server trust store.

Step 4 Restart the Cisco Unified CVP VXML Server service running in VXML Server.

Note Do not import a server certificate signed with a standard CA to the VXML Server trust store, as it contains standard CA details.

Create Two-Way Communication Between VXML and REST Server

Two-Way secure communication between VXML and REST Server involves importing the VXML Server CA certificate into the REST Server trust store.

Perform the following steps to import the VXML Server CA certificate on the REST Server:

Procedure

Step 1 Retrieve the keystore password from the security.properties file on the VXML Server.

Step 2 Use the Java key tool to export the VXML Server CA certificate from the keystore.
File path to VXML keystore: `%CVP_HOME%\conf\security\.ormKeystore`.

Step 3 Copy the exported certificate file from the managed Cisco Unified CVP VXML Server to the REST Server.
For example: <VXMLca_cert_file>

Step 4 Use the following Java key tool command to import the certificate into the REST Server truststore

```
keytool –importcert -keystore <Path to REST server Truststore> -alias <Alias_name> -file
<path to VXMLca_cert_file >
```

Note
- For a self-signed certificate, export the ca_cert from the VXML Server and import the ca_cert to the REST Server truststore.
- For a VXML standard trusted CA, do not import the CA certificate on the REST Server truststore.

XPath Expression

Cisco Unified Call Studio includes a new utility that allows you to use XPath expressions in JavaScript to return values from the XML. You can specify an XPath expression in the element setting. If the REST response is an XML, then the nodes which are returned are available as element data. Based on the XML result from the GET method, you can add XPath expression to get the value of a specific row.

For example, consider the following XML you get when you query WSM SNMP public:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<results>
    <communities>
        <community>
            <name>Hello</name>
            <snmpversion>V1</snmpversion>
            <acceptfromanyhost>true</acceptfromanyhost>
            <accessprivilege>readWrite</accessprivilege>
            <servers>
                <server>IP address</server>
            </servers>
        </community>
    </communities>
    <pageinfo>
        <resultsPerPage>25</resultsPerPage>
        <startIndex>0</startIndex>
        <totalResults>1</totalResults>
    </pageinfo>
</results>
```

To get the value from one specific row, use the following XPath expression: /results/communities/community/snmpversion.

The output of the expression is **V1**.

If you use the following XPath expression:/results/communities/community/name.

The output of the expression is **Hello**.

JSONPath Expression

Cisco Unified Call Studio includes a new utility that allows you to specify a JSONPath expression in the element setting. The nodes which are returned are available as element data if the REST response is a JSON.

For example, consider the following REST response:

```
{"community":
{"name":"public","snmpversion":"V1","acceptfromanyhost":"true","accessprivilege":"readOnly","servers":{"server":"IP
 address}}"
}
```