



Web Service

Along with Action and Decision elements, another way to perform backend interactions and obtain real-time data is via the Web Service element. This element leverages industry standards, such as the Web Service Definition Language (WSDL) for service definitions and SOAP for message encapsulation to provide simple, seamless interaction with remote web services.

Unlike one-off web service implementations using custom code, this element provides an intuitive graphical interface that dynamically adjusts to match each of your web services. It uses WSDL to discover required and optional settings, setting dependencies, and even valid enumerated values. Like other elements in @audiumstudio.field@, it ensures that the values you enter are of the right type, while still allowing the use of Substitution throughout.

Web Service elements provides a dynamic graphical interface for embedding web service interactions into the call flow.

This element is designed to work with the following technologies:

- WSDL 1.1 (using namespace <http://schemas.xmlsoap.org/wsdl/>)
 - Binding Styles
 - RPC/encoded
 - RPC/literal
 - Document/literal
 - Document/literal (wrapped)
- SOAP 1.1 encoding (using namespace <http://schemas.xmlsoap.org/soap/encoding/>)
 - Includes built-in support for 1-dimensional SOAP-encoded arrays that do not use href references for array items.
 - To parse n-dimensional SOAP-encoded arrays (where n is greater than 1) or href references in web service response messages, use the "Store Full Response XML" option and process the response with custom code.
- XML schemas (using namespace <http://www.w3.org/2001/XMLSchema>)
 - Includes built-in support for 1-dimensional arrays (that is, sequences).
 - To parse n-dimensional arrays (where n is greater than 1) in web service response messages, use the "Store Full Response XML" option and process the response with custom code.

Note

The earlier application that contains Web Service element has to be imported again to Call Studio latest version before deploying in new VXML server.

- [Exit States, on page 2](#)
- [Element Data, on page 2](#)
- [Settings, on page 3](#)
- [Configuring Request Parameters, on page 8](#)
- [Configuring Response Parameters, on page 9](#)

Exit States

Name	Description
done	This exit state is followed when the web service was successfully invoked at runtime, and responded within the time specified in the "Connection Timeout" setting.
Java Exception-error	This exit state is followed when the element encounters any error at runtime. Some examples include a web service that cannot be reached, the web service taking too long (more than the value specified in the "Connection Timeout" setting) to respond, or receiving unexpected data from the service. If this exit state is followed, refer to the @audiumcallservices.field@ logs for additional information about the cause.
fault	This exit state is only present when the loaded WSDL specifies a possible fault message for the selected operation. This exit state is followed when the web service is successfully contacted at runtime, but it responds with its fault message.

Element Data

response_xml	Only created if the "Store Full Response XML" checkbox has been checked. Holds the full XML response from the web service at runtime, for later processing by custom code or for debugging purposes.
--------------	--



Note This element may also create numerous other element or session data variables (with user-specified names), depending on the settings specified in the "Configure Response Parameters" dialog.

Settings

The Web Service element has just one Element Configuration tab, named "General". Refer to the image below and description of each setting for more information.

Figure 1: Element Configuration Tab

The screenshot shows the 'Element Configuration' dialog box for a 'Web Service' element. The 'General' tab is selected. The dialog is titled 'Web Service' and has a sub-tab 'General'. The 'Name' is 'Web_Service_01'. The 'Load WSDL' section includes a 'URI' dropdown menu, a text input field, a 'Browse' button, and a 'Load' button. The 'Configure Web Service Call' section includes 'Service:', 'Port:', and 'Operation:' labels with corresponding text input fields. Below these are 'Request:' and 'Response:' labels, each with a 'Configure' button. There is also a checkbox for 'Store Full Response'. The 'Runtime Settings' section includes a 'Connect Timeout:' label with a text input field containing the value '3'. Below this is a checkbox for 'Requires HTTP Authentication'. The 'Username:' label has a text input field and a refresh icon. The 'Password:' label has a text input field, a 'Uses Substitution' checkbox, and a refresh icon. Below these are a checkbox for 'Use Proxy', and 'Proxy Host:' and 'Proxy Port:' labels with corresponding text input fields.

Group	Name	Description
Load WSDL	WSDL Location	<p>In order for the Web Service element to be configurable, a WSDL file defining the desired web service must first be loaded. First, choose either "URI" or "File" from the drop-down, then either browse for a local file or enter a remote URI where the WSDL can be retrieved, the URI can be HTTP or HTTPS. Then, click the "Load" button to initiate @audiumstudio.field@'s download, caching, and parsing of the WSDL. Once WSDL is loaded, the other configuration options become available.</p>

Group	Name	Description
Configure Web Service Call	Service	This drop-down allows you to select which service you would like this element to invoke at runtime. Generally, WSDL files only define a single service so this list may have just one item. Each service's namespace is listed alongside it in parenthesis.
	Port	This drop-down allows you to specify which port you would like to use to connect to the web service at runtime. Each port has a name, and may define completely different connection properties than other ports. Please refer to your web service's documentation, or the WSDL file, for information about what each port represents. Note that this port list is dependent on which service is selected, and so it will update as the service is changed.
	Operation	This drop-down allows you to specify which operation you would like to run against the previously-selected service. Note that this operation list is dependent on which port is selected, and so it will update as the port is changed.
	Request	Click the "Configure" button next to the "Request" label to bring up the "Configure Request Parameters" dialog. Using that dialog, you can specify which values to send to the web service as inputs at runtime.
	Response	Click the "Configure" button next to the "Response" label to bring up the "Configure Response Parameters" dialog. Using that dialog, you can specify in which element or session data variable each potential return value from the web service should be stored at runtime.
	Store Full Response XML	

Group	Name	Description
		<p>Check this box if you would like the full XML response from the web service to be stored in element data at runtime, for later processing by your own custom code, or for debug purposes. Note that checking this box may be memory intensive if the response XML documents are large. Even if this checkbox has been selected, response parameter storage settings from the "Configure Response Parameter" dialog will still be used.</p>

Group	Name	Description
Runtime Settings	Connect Timeout	<p>This setting allows you to specify how many seconds @audiumcallservices.field@ should wait for the web service socket connection to get established at runtime, before timing-out and following the "error" exit state.</p> <p>Note This is the tcp connect timeout (sun.net.client.defaultConnectTimeout), which specifies the timeout (in milliseconds) to establish the connection to the host. Depending on which services uses it (http or ftp), it is used for the connection set up and not for the service timeout.</p>
	Requires HTTP Authentication	Check this box if you would like HTTP authentication to be used when accessing the web service at runtime.
	Username	Only available if the "Requires HTTP Authentication" checkbox has been selected. This field allows you to specify the username to use for HTTP authentication when accessing the web service at runtime.
	Password	Only available if the "Requires HTTP Authentication" checkbox has been selected. This field allows you to specify the password to use for HTTP authentication when accessing the web service at runtime.
	Use Proxy	Check this box if you would like a proxy to be used when accessing the web service at runtime.
	Proxy Host	Only available if the "Use Proxy" checkbox has been selected. This field allows you to specify the proxy host to use to access the web service at runtime.
	Proxy Port	

Group	Name	Description
		Only available if the "Use Proxy" checkbox has been selected. This field allows you to specify the proxy port to use to access the web service at runtime.

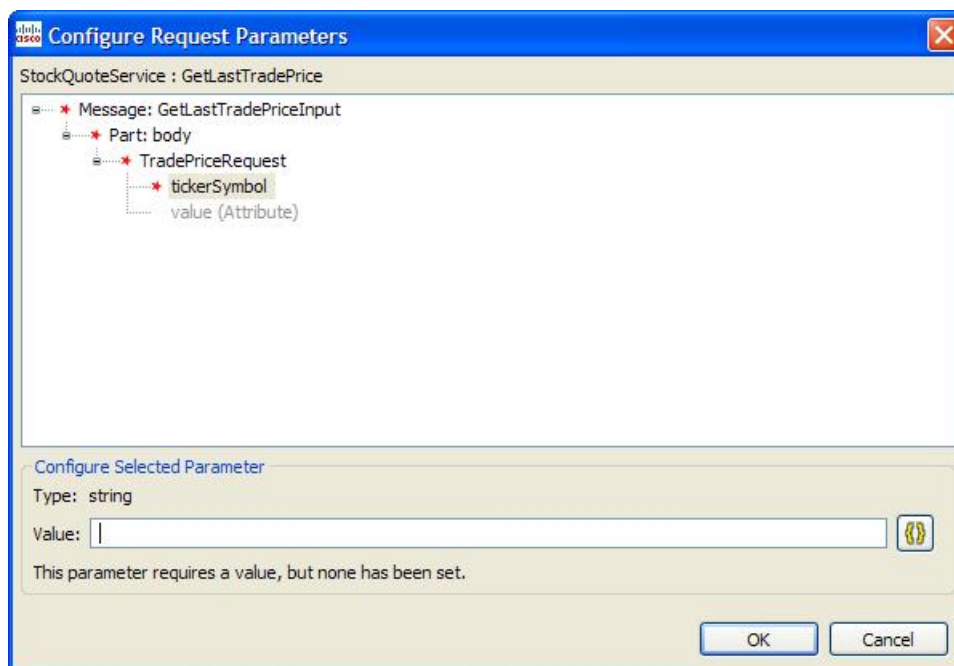
Configuring Request Parameters



Note Unified CVP Call Studio does not support SOAP Encode Schema. For all request and response parameters use the XMLSchema namespace format as listed in the [XML Schema document](#).

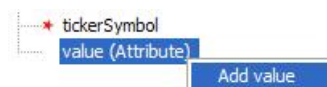
When the "Configure" button for Request Parameters is clicked, the following dialog is displayed:

Figure 2: Configure Request Parameters



Its contents are pre-populated with parameters that the loaded WSDL specifies. These parameters are displayed in a tree format, and use the same symbols for required and repeatable that the settings of other elements use. If a setting is optional it is greyed-out by default (like "value" in the image above), and can be added by right-clicking on it and choosing "Add PARAM_NAME":

Figure 3: Add Parameter



Each parameter has a type, such as string, integer, or float. Some parameters cannot hold a value (they will show "N/A" as their type), because they are intended to either only contain child parameters, or to act as markers. An example of a marker parameter might be "disable_logging"; if it is defined, then no logging will be performed on the service end. Only variables with a type can hold a value. The value you enter will be validated as you type it (a warning message may be displayed below the value field), and also when you validate the entire project before deploying.

If a setting is repeatable it will have its index in the list in brackets, such as the "item" parameter in the following example:

Figure 4: Repeatable Parameters



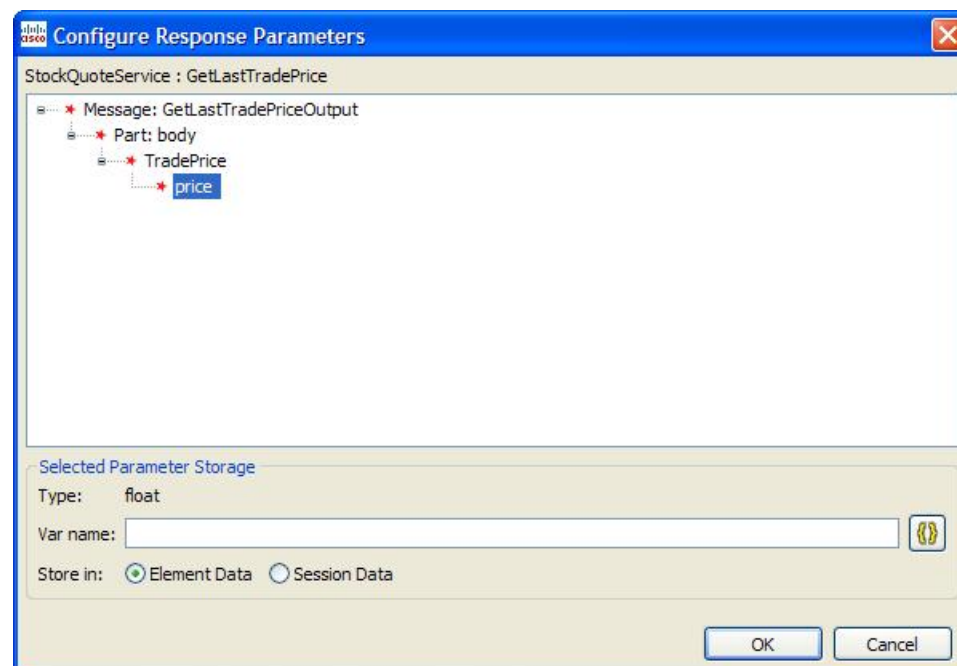
You can add additional parameters to the list by right-clicking on any list item and choosing "Add PARAM_NAME". To remove a parameter from the list, right-click on it and choose "Delete PARAM_NAME". This same functionality can be used to disable (gray-out) an optional parameter, regardless of whether it is repeatable or not.

Similar to element settings, all required parameters must be configured with a value in order for the voice application project to pass validation.

Configuring Response Parameters

Response parameters (data sent back by a web service) are handled in much the same way as request parameters. The "Configure Request Parameters" dialog is also tree-based, and it allows for parameters to be added or deleted as desired.

Figure 5: Configure Response Parameters



However, there are a few differences. First, you must specify whether each parameter should be stored in Element or Session data. Additionally, the text input field is used to specify the variable name to create, rather than a value to pass to the service.

No type-checking is performed in this dialog; the response parameter type is listed only for convenience.

The most significant difference between this dialog and the "Configure Request Parameters" dialog is that parameters marked as required do not need to be configured. Any parameter not configured in this dialog will simply not be stored in element or session data at runtime; if it is present in the web service's response, it will be ignored.