Form Input View Options

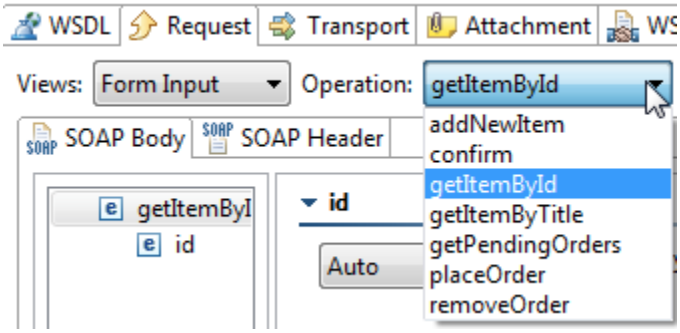
This topic covers the Form Input view, which allows you to enter parameters in UI form fields.

Sections include:

> **Populating the Form Input View from the Literal XML view**
>
> If you want a simple way to view and work with an existing XML message (e.g., a sample from developers, traced from the server, from legacy testing tools, etc.), you can copy it into the Literal view, then open the Form Input view. Once a message is copied into the Literal view, the Form Input view is populated in a schema-aware and schema-constrained manner that is much simpler to edit, manage, and parameterize.

General Options

The top of the Form Input View provides an operation drop-down menu that lists the available methods/operations you can call and work with.



The operation list is created automatically when a valid WSDL URI is entered. You can select which operation you want to work with by selecting an operation from the list.

The main panel contains an expandable representation of the message. SOAP-based tools have sub-tabs for the **SOAP Body** and the **SOAP Header**. For information on the **Header** sub-tab, see Adding SOAP Headers. The **SOAP Body** and **SOAP Header** tabs are not available if you chose the Plain XML message type in the WSDL tab.The available controls let you specify how you want to specify the parameter values for each node in the tree. Options include:

- **Fixed:** Use this option if you want to enter a literal value (such as a string) by completing form fields. If you select **Fixed** in the **Value** box, enter the literal value for your tool parameters in the control that opens. The nature of the control will vary depending on the method. For more information, see Specifying Literal Values.
- **Parameterized:** Use this option to use values specified in a data source. You must first add the appropriate data source to theresponder or test suite.
- **Auto** Use this option to automatically generate parameter values. Automatic parameter generation is particularly useful when you want to assess the service's constraints and determine what type of data will fail. If you selected **Auto**, you do not need to enter anything to the right of the control. Values will be generated automatically. *This option is not applicable to Form JSON*.
- **Script:** Use this option if you want a script to generate parameters at runtime. If you select **Script**, click **Edit Script**, then enter the method's location (or the method itself) in the dialog box that opens. The method should be a static Java method that returns an object (or a Java class that contains a default constructor) and that meets the signature the Web service is expecting. Parasoft will transform the method into a SOAP parameter and send it as a SOAP call when the tool is executed. For general guidelines on adding methods, see Extensibility and Scripting Basics.
- **Input:** Use this option if this tool is chained to another tool and you want the result from the first tool to be used as the parameter for the second tool.

Specifying Literal Values

You can send a literal value as part of the message if **Fixed** was chosen in the **Value** box Depending on the method, a number of different options may be available.

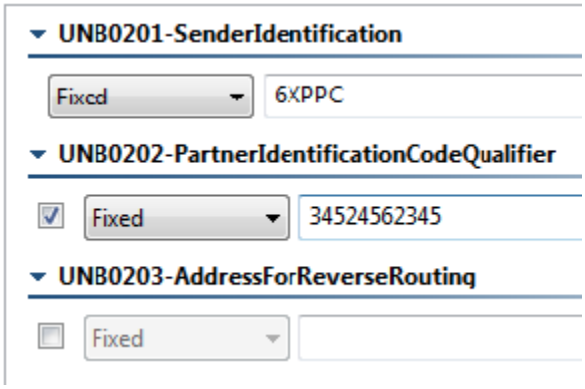
You can either configure options for sending simple parameters as the literal value, or you may configure options for sending nested parameters within complex types of objects (such as an array).

Configuring Simple Parameters

When configuring a simple parameter, the options will vary depending on whether a data source was specified in the tool, and if the XML schema of the WSDL contains any nillable elements.

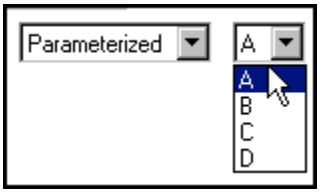
The following options may be available when configuring values for simple parameters:

- **nil:** Determines whether the nil attribute is sent by the tool. If this box is checked, the nil attribute will be sent by the tool. If this option is not enabled, the nil attribute will not be sent. Note that for Form JSON, this field is **null** rather than **nil**.
- **Fixed/Parameterized Value:** (Only available if the nil check box is not selected and the "optional" check box [if present] is selected) Determines the value to be sent by the tool. If a data source is specified in the tool, you will have a choice to select either a **Fixed** or a **Parameterized** value. If a data source is not available, then only a Fixed value can be entered.
  - **Fixed** values are the literal values that you specify by entering an input into the available text field.



Note that in the example above, the checkbox to the left of the second and third Fixed items indicate that the item is optional. When that box is checked, that element will be sent in the message. When it is not checked, it will be omitted.

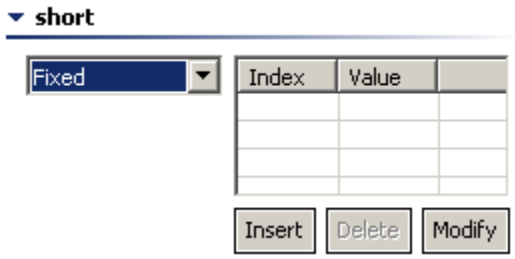
- **Parameterized** values are the values from a data source column. When configuring Parameterized values, a drop-down box containing column names from a data source will become available. These column names correspond to the data source specified in the tool. All values in the data source column that you select will be sent as the literal value by the tool.



If you are parameterizing from a repository data source, be sure to use record list columns to parameterize complex elements and use primitive or primitive list columns to parameterize simple elements.

Configuring Nested Parameters

If the WSDL specified in the tool contains an array of objects, the GUI beneath the Value box displays an **Index** column and a **Value** column. You will be able to nest parameter values that the tool will send as part of the message.



Parameterizing with Hierarchical Data from a Data Repository Data Source

You can parameterize complex XML elements from a Parasoft Data Repository data source column, which may reference zero, one, or multiple hierarchical record values.

1. Add a Parasoft Repository Data source as described in Creating a Repository Data Source.
2. Parameterize the desired column(s) as described in Hierarchical Parameterization with Form Views.

Advanced Options

The following advanced options can be performed in the Form Input view:

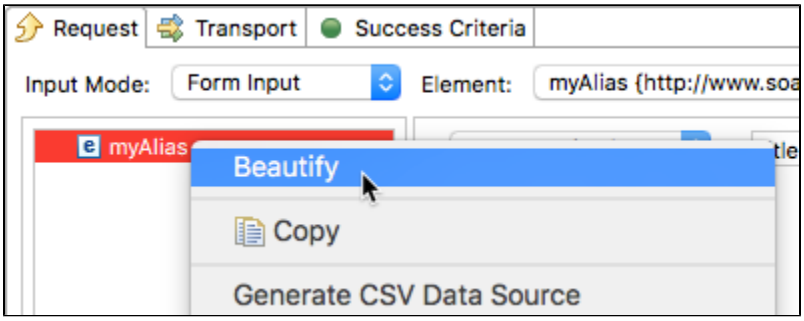
Using Excel Spreadsheets to Generate Dynamic Array or Sequence Values

Manually creating a data source for parameterizing large, complex messages can be time-consuming and tedious. You can configure SOAtest and/or Virtualize to automatically generate a CSV data source template based on the structure of the message that you want to parameterize.

Details on how to do this are provided inGenerating a Data Source Template for Populating Message Elements in SOAtest or Generating a Data Source Template for Populating Message Elements in Virtualize.

Sending Compact and Beautified Messages

You can right-click the root element in the messaging tool and choose **Beautify** to format messages for readability. You can also send compact messages (messages formatted to fit on a single line) by by right-clicking the node and disabling the **Beautify** option.



Schema Type Enforcement

By default, SOAtest and/or Virtualize checks to ensure that the parameter value entered conforms to its specified schema type (shown when you hover your cursor over the element). If you want to use a value that does not conform to the schema type, disable the schema type enforcement by right-clicking the element, then choosing **Enforce Schema Type**. To disable schema enforcement for child elements, right-click the parent element, then choose **Ignore Schema Type of Children**. If you later want to re-enable schema type enforcement, choose the appropriate right-click option.

minoccurs/maxoccurs Enforcement

By default, SOAtest and/or Virtualize checks to ensure that the parameter value entered conforms to minoccurs and maxoccurs constraints. If you want to use a value that is beyond these limits, right-click that element, then choose **Enforce Occurrences**. If you later want to re-enable schema type enforcement, repeat the same action.

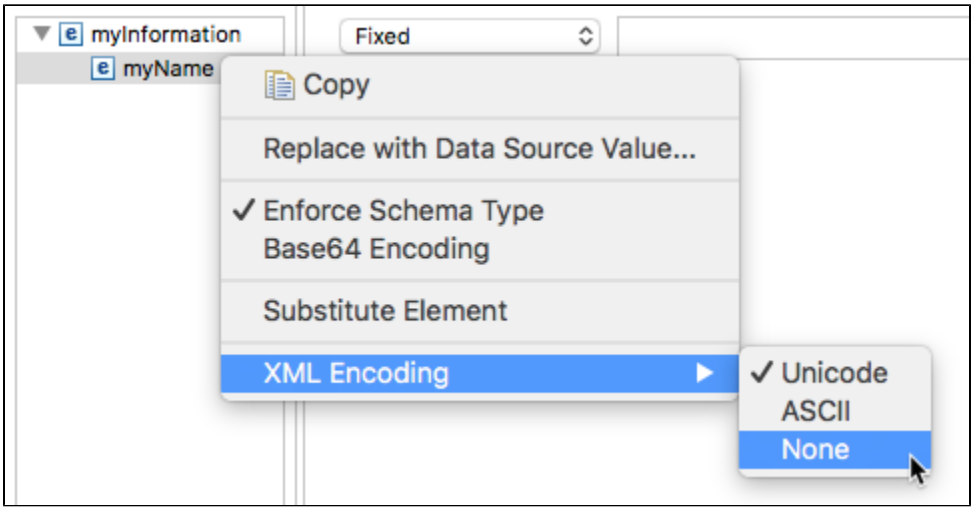
Base 64 Encoding

Base 64 encoding is not enabled by default. To use base 64 encoding, right-click the element you want encoded, then choose **Base 64 Encoding** from the shortcut menu. If you later want to disable base 64 encoding, repeat the same action.

XML Encoding

By default, XML values are encoded as unicode characters, but you can right-click an element and choose **XML Encoding> Unicode** or **ASCII** or **None** to change the encoding.

- Choose **Unicode** to preserve all unicode characters supported by XML specifications. Only restricted XML characters, such as "<" and "&", will be encoded.
- Choose **ASCII** to preserve all ASCII characters. Non-ASCII unicode characters, as well as restricted XML characters, will be encoded.
- Choose **None** to disable XML encoding. Characters from the field value, including restricted XML characters, will be represented as-is in the final document.



Adding Multiple Values to an Array

To quickly add multiple values to an array, right-click the related element, then choose **Insert Multiple**. In the dialog that opens, use the available controls to specify the values that you want to use.

Replacing Specific Elements with Data Source Values

You can configure SOAtest and/or Virtualize to replace an entire element (or just its content) with a value from a data source or XPath when the tool is executed. To do this:

1. Right-click that element, then choose **Replace with Data Source Value**.
2. In the Replacement Settings dialog that opens, specify the following, then click **Next**.
    - The replacement mode (entire element or content only). *Note that this option is not applicable to Form JSON, where it replaces an entire object or array*.
    - Whether you want to use a data source value or an XPath.
3. In the next page, specify the appropriate data source column or XPath, then click **Finish**.

If you later want to stop using the data source or XPath value, right-click the element and choose **Remove Replacement Setting**.

Populating a Set of Elements from a Data Source Values

The Populate feature allows you to automatically fill a set of form fields using values stored in an existing data source (as opposed to specifying values manually). For example, you can create a data source with one column per element then automatically add the corresponding values to all of the available elements.

To populate an element's fields using data source values, right-click that field in the Form Input view, then choose **Populate**. Leave **Map parameter values to data source columns** enabled, then specify exclusion and nillable settings if applicable.For additional details, see Populating and Parameterizing Elements with Data Source Values.

Populating a Set of Elements with Automatically-Generated Values

The Populate feature can also automatically generate simple values for a set of form fields.

To populate an element's fields using simple automatically-generated values, right-click that field in the Form Input view, then choose **Populate**. Clear **Map parameter values to data source columns**, then specify exclusion and nillable settings if applicable.For additional details on the options available in the Populate wizard (Element Exclusion, Nillable Elements, Attribute Exclusion), see Populate Wizard Options.

Substituting Elements

If you want to replace the original element with another element defined in a schema (for example, to use a foreign-language equivalent of the original element name, or to use a different variation of the original element), you can use the Substitute Element feature. Substituted elements need to have the same type as the original—unless the original element's type is anyType. In that case, you can replace it with any element that is a valid substitution (any element that belongs to the 'substitution group' defined by the abstract element that's being substituted/replaced).

To substitute an element, right-click the original element, then choose **Substitute Element** from the shortcut menu. In the dialog that opens, specify the schema that contains the new element, then add the desired element to the substitution list.

Using Data Source Values to Determine if Optional Elements or Attributes Are Sent

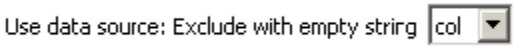
(Only available if a data source is included in the responder or test suite and the WSDL or schema specifies that minOccurs=0 for that parameter—Does not apply to repository data sources)

When you are working with a message with optional elements or attributes, you can use a data source to configure whether or not the optional elements or attributes are sent as part of the message. After the proper configuration (described below), an empty string in the designated data source column will tell SOAtest and/or Virtualize NOT to include the optional elements or attributes in the message. If the data source has an actual value, that value will be sent as part of the message.

Using Data Source Values to Configure if Optional Elements Are Sent

To have values stored in a data source dictate whether a value is sent for an optional element:

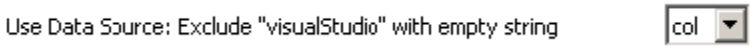
1. Right-click the element's tree node.
2. Choose **Use Data Source> Exclude Element with Empty String**.
3. In the tool configuration panel, ensure that the appropriate data source column is selected under **Use data source: Exclude with empty string**.



Using Data Source Values to Configure Optional Attributes Are Sent

To have values stored in a data source dictate whether a value is sent for an optional attribute:

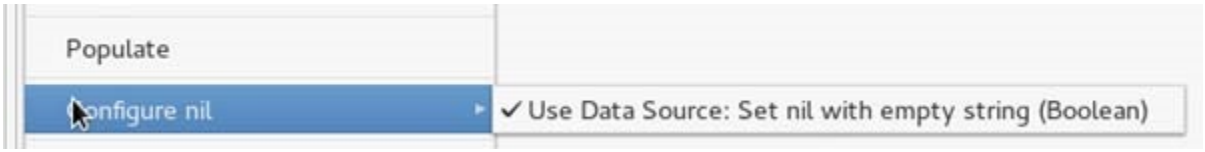
1. Right-click the parent element's tree node.
2. Choose **Use Data Source> Exclude Attribute with empty string> [attribute you want to configure]**.
3. In the tool configuration panel, ensure that the appropriate data source column is selected under **Use data source: Exclude [attribute] with empty string**.



Using Data Source Values to Configure if Nil or Null Attributes Are Used

(Only available if a data source is included in the responder or test suite and the nil check box for the given element is not selected — Does not apply to repository data sources)

To have values stored in a data source dictate whether a nil attribute or an actual value is used for various elements, right-click the related tree node, then choose **Configure nil> Use Data Source: Set nil with empty string [element]** (for JSON, the label is **Configure null> Use Data Source: Set null with empty string [element]**).



When the data source has an empty string, the nil attribute will be sent as part of the message. If a value is specified, the nil attribute will not be sent; instead, the specified value will be used.

For instance, assume you have the following data source:

Element Value	Nil with Empty String
value	value
value	
	value

Nil with Empty String takes precedence over Element Value. Consequently, `xsi:nil="true"` will be sent regardless of the value in the Element Value column whenever the Nil with Empty String column has an empty string.

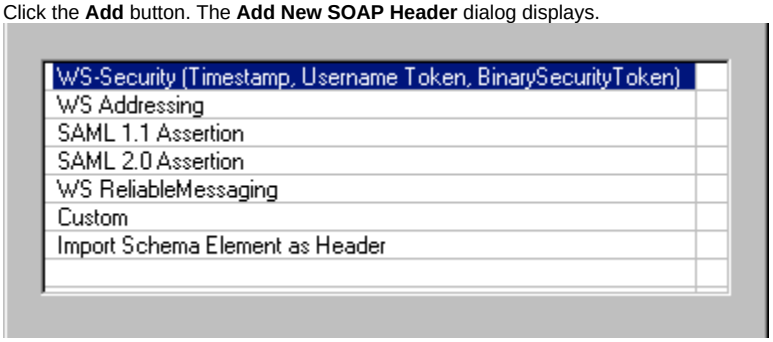
Using Nil with Repository Data Sources

This description does not apply to repository data sources. For details on how to to set a respository data source value to nil, see Setting Values to Null or Exclude.

Adding SOAP Headers

The **SOAP Header** sub-tab in the Form Input view allows you to specify header parameters. To add a header, perform the following from the Header sub-tab of the Form Input view:

1. Click the **Add** button. The **Add New SOAP Header** dialog displays.



2. Select **Custom**, **WS-Security**, **SAML 1.1 Assertion**, **SAML 2.0 Assertion**, **WS Addressing**, **WS ReliableMessaging**, or **Import Schema Elementas Header** from the Available Header types and click **OK**. For details about the available options, see:
   - WS-Security Options
   - WS Addressing Options
   - SAML 1.1 Assertion Options
   - SAML 2.0 Assertion Options
   - WS Reliable Messaging Options

WS-Security Options

If you selected **WS-Security** from the Available Header Types, the following options are available via the **Timestamp**, **Username Tokens**, **BinarySecurity Token**, and **Options** tabs in the right GUI panel:

Timestamp Tab

The following options are available in the Timestamp tab:

Option	Description
Send Timestamp	Select to add a timestamp value.
Dynamically generate timestamp	Select to generate a timestamp value each time a message is generated.
wsu:Created	(Only available if Dynamically generate timestamp is not selected) Manually enter a timestamp value. This value will be the same for each message generated.
Send Expires	Select to add an expiration value.
Time Interval between Created and Expires	(Only available if Send Expires is selected) Manually enter a time interval to take place between the created and expires timestamps.

Username Tokens Tab

The following options are available in the Username Tokens tab:

Option	Description
Send Username Token	Select to add a username value.
wsse: Username	Enter a username.

wsse: Password	Enter a password.
Add Nonce	Select to add a nonce value.
Dynamically generate Nonce	Select to randomly generate a nonce value each time a message is generated.
wsse:Nonce	(Only available if Dynamically generate Nonce is not selected) Manually enter a Nonce value. This value will be the same for each message generated.
Add Timestamp	Select to add a timestamp value.
Dynamically generate timestamp	Select to generate a timestamp value each time a message is generated.
wsu:Created:	(Only available if Dynamically generate timestamp is not selected) Manually enter a timestamp value. This value will be the same for each message generated.

BinarySecurityToken Tab

The following options are available in the Username Tokens tab:

Option	Description
Tokens	Select a BinarySecurityToken from the **Tokens** drop-down list.
Add	Click to add a new BinarySecurityToken. In the **Add BinarySecurityToken** dialog that displays, specify the **wsuID** of the token.
Remove	Click to remove a BinarySecurityToken.

Options Tab

The following options are available in the Options tab:

Option	Description
WS-Security URI	Enter or select from the drop-down menu, the namespace of the Username token header. The default corresponds to the latest WS-Security spec from OASIS. Selecting a **WS-Security URI** also changes the **WS-Security Utility URI** correspondingly. However, you can change the WS-Security Utility URI so that it does not correspond to the WS-Security URI.
WS-Security Utility URI	Enter or select from the drop-down menu, the utility namespace of the Username token header. The default corresponds to the selection in the **WS-Security URI** menu.

WS Addressing Options

If you selected **WS Addressing** from the Available Header Types, the following four Header types are sent by default: **Action, To, MessageID,** and **Reply /To**. The following headers are not by default: RelatesTo, From, FaultTo. These Header types can be configured via the **Action/To**, **MessageID/ReplyTo**, and **RelatesTo/From/FaultTo** tabs in the right GUI panel. Also by default, the 2004/08 namespace is used (i.e. http://schemas.xmlsoap.org/ws/2004/08 /addressing).

Action/To Tab

The following options are available in the Action/To tab:

Option	Description
wsa:Action	Specifies the WS Addressing Action value.
wsa:To	Specifies the WS Addressing To value.
WS-Addressing URI	Enter or select from the drop-down menu, the WS-Addressing URI.

MessageID/ReplyTo Tab

The following options are available in the MessageID/ReplyTo tab:

Option	Description

wsa:MessageID	Specifies the WS Addressing MessageID. The default automatically generates a unique value.
wsa:ReplyTo	Specifies the WS Addressing ReplyTo endpoint reference to which the return message will be sent. • **wsa:Address:** You can configure the wsa:Address as **Fixed**, **Automatic**, **Script**, **Anonymous URL**, or **Call Back URL**. • **wsa:ReferenceParameters and wsa:ReferenceProperties:** Allows you to modify the given parameters and properties.

RelatesTo/From/FaultTo Tab

The following options are available in the RelatesTo/From/FaultTo tab:

Option	Description
Send wsa:RelatesTo	Specifies the WS Addressing RelatesTo value.
Send wsa:From	Specifies the WS Addressing From endpoint reference from which the return message is sent. • **wsa:Address:** You can configure the wsa:Address as **Fixed**, **Automatic**, **Script**, **Anonymous URL**, or **Call Back URL**. • **wsa:ReferenceParameters and wsa:ReferenceProperties:** Allows you to modify the given parameters and properties.
wsa:FaultTo	• Specifies the WS Addressing FaultTo endpoint reference to which the fault message will be sent. • **wsa:Address:** You can configure the wsa:Address as **Fixed**, **Automatic**, **Script**, **Anonymous URL**, or **Call Back URL**. • **wsa:ReferenceParameters and wsa:ReferenceProperties:** Allows you to modify the given parameters and properties.

SAML 1.1 Assertion Options

If you selected **SAML 1.1 Assertion** from the Available Header Types, see Adding SAML Headers.

SAML 2.0 Assertion Options

If you selected **SAML 2.0 Assertion** from the Available Header Types, various options will be available that correspond to the OASIS SAML Token Profile. For more information, see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

WS Reliable Messaging Options

If you selected **WS ReliableMessaging** from the Available Header Types, the following options are available:

Option	Description
wsrm: Sequence	Specifies the wsrm:Sequence parameters such as the Identifier and MessageNumber.
wsrm: AckRequested	Specifies the wsrm:AckRequested parameters such as the Identifier and MaxMessageNumberUsed. These options are only available if the **Send AckRequested** check box is selected.

Custom Options

If you selected **Custom** from the Available Header Types, the following options are available:

Option	Description
Views	Select the desired view from the drop-down menu and configure accordingly.
Populate	Fills SOAP array and element parameters. Clicking this button also sets any element nils to false and expands them out. This button is only enabled if the tool is created from a WSDL.

Import Schema Element as Header Options

If you selected **Import Schema Element as Header** from the Available Header Types, a dialog appears from which you can load declared elements from a schema location. After loading elements, you can select multiple elements for the SOAP header. Once you click **OK**, a new header with the chosen element's structure will be added to the tool.

Adding SAML Headers

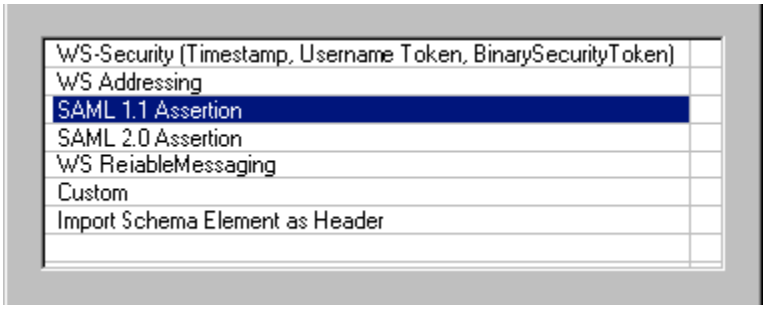
The **Header** sub-tab in the Form Input view allows you to specify header parameters. When adding SAML 1.1 headers, you can choose from four types of SAML confirmation methods:

- **Sender-Vouches: Unsigned:** Contains minimal sender-vouches SAML assertion with no optional elements included in the request message creation. There are no signatures or certificates required for the SAML assertion.
  - See Sender-Vouches: Unsigned for details.
- **Sender-Vouches: Unsigned: SSL:** Contains minimal sender-vouches SAML assertion, whereas a signature is not required, but the transport is over SSL and therefore certificates will be required in order to support SSAL at the transport layer.
  - See Sender-Vouches: Unsigned: SSL for details.
- **Sender-Vouches: Signed:** Contains a sender-vouches SAML assertion, whereas both the assertion and the body elements are signed, while a reference to the certificate used to verify the signature is to be provided in the header.
  - See Sender-Vouches: Signed for details.
- **Holder-of-Key:** Contains a holder-of-key SAML assertion where an enveloped signature is over the Assertion element using the issuer key store, the certificate used to verify the issuer signature is contained within the assertion signature. The body is then signed with the user certificate which is to be contained within the assertion's SubjectConfirmation element.
  - See Sender-Vouches: Signed for details.

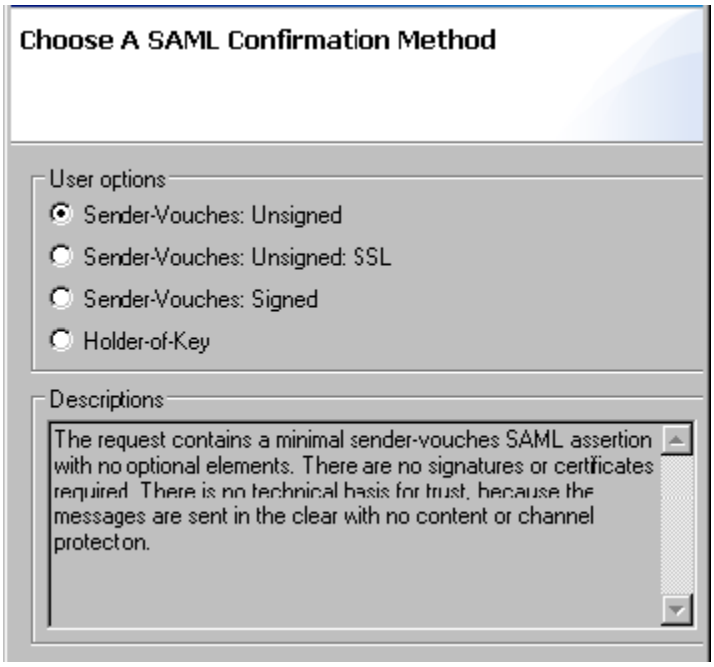
Sender-Vouches: Unsigned

To add a **Sender-Vouches: Unsigned** confirmation method, perform the following from the Header sub-tab of the Form Input view:

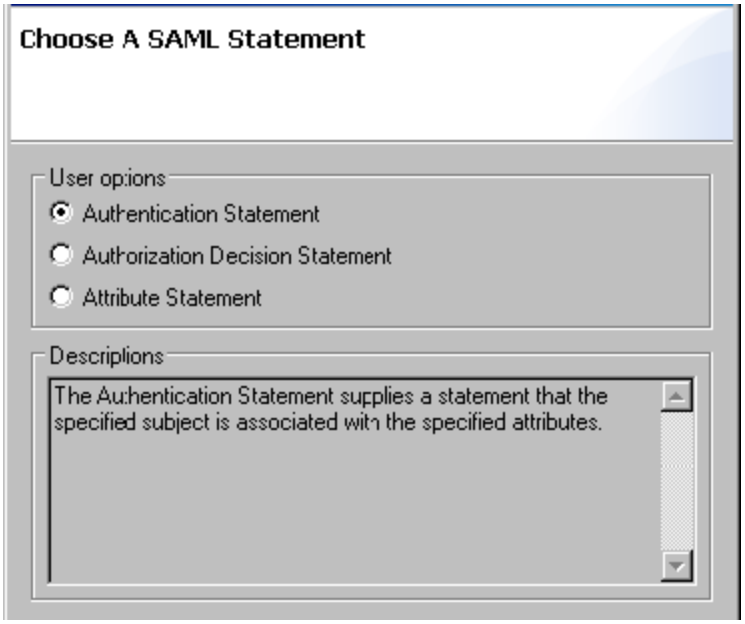
1. Click the **Add** button. The **Add New SOAP Header** dialog displays.



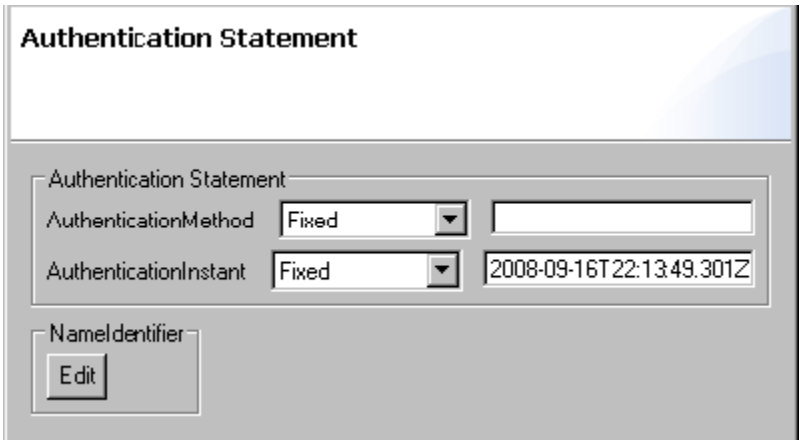
2. Select **SAML 1.1 Assertion** and click **OK**. The SAML Assertion wizard displays.



3. Select **Sender-Vouches: Unsigned** and click the **Next** button.



4. Select the desired SAML statement type and click **Next**.

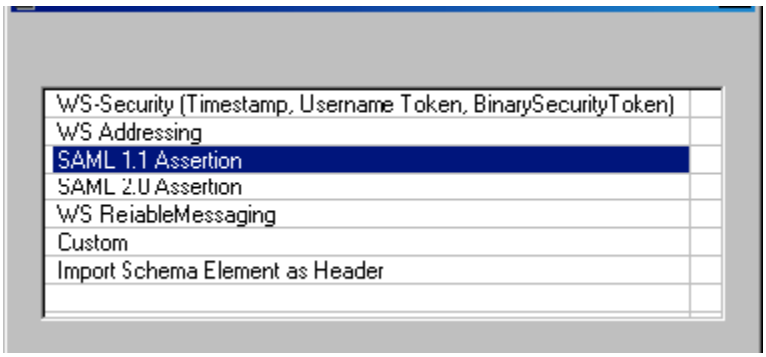


5. Complete the necessary fields for the **Authentication Statement** and click **Finish**. For more information on Authentication Statements, see Adding and Modifying SAML Statements.

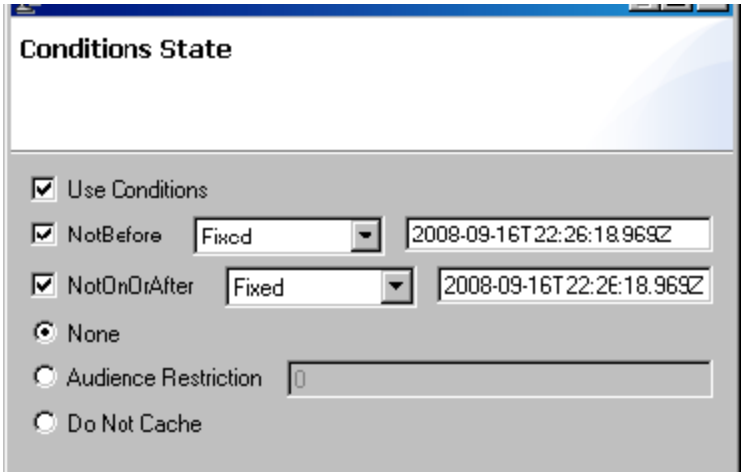
Sender-Vouches: Unsigned: SSL

To add a **Sender-Vouches: Unsigned: SSL** confirmation method, perform the following from the Header sub-tab of the Form Input view:

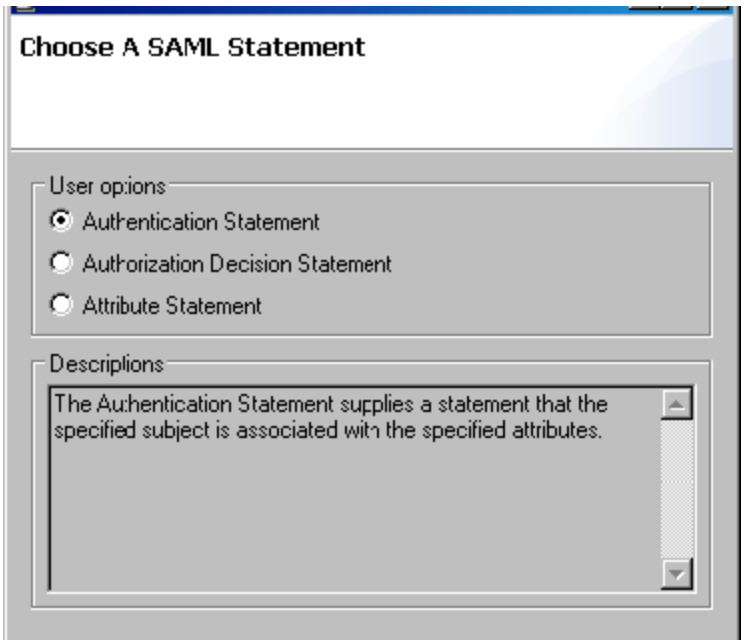
1. Click the **Add** button. The **Add New SOAP Header** dialog displays.



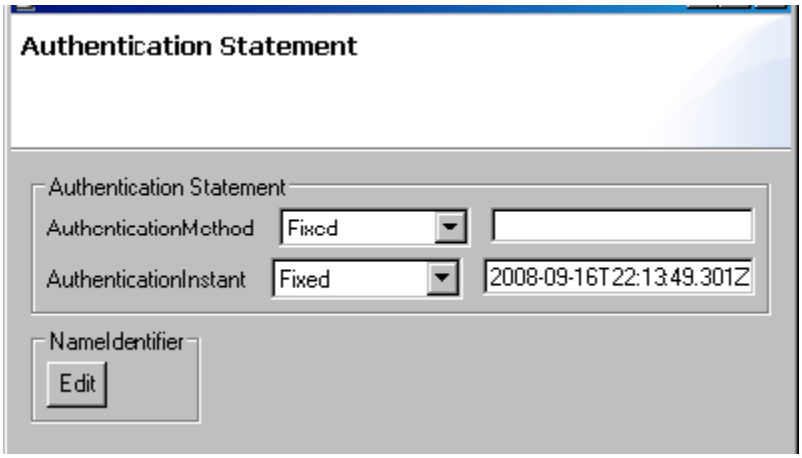
2. Select **SAML 1.1 Assertion** and click **OK**. The SAML Assertion wizard displays.
3. Select **Sender-Vouches: Unsigned: SSL** and click the **Next** button.



4. Select **Use Conditions**, if needed.
    - If **Use Conditions** was selected, then choose the desired Condition type or None.
    - If **Audience Restriction** was selected and a value greater than 0 was entered, then click **Next** to be prompted by the **Audience Restriction** control and fill each field in accordingly.
5. Click the **Next** button.



6. Select the desired SAML statement type and click **Next**.

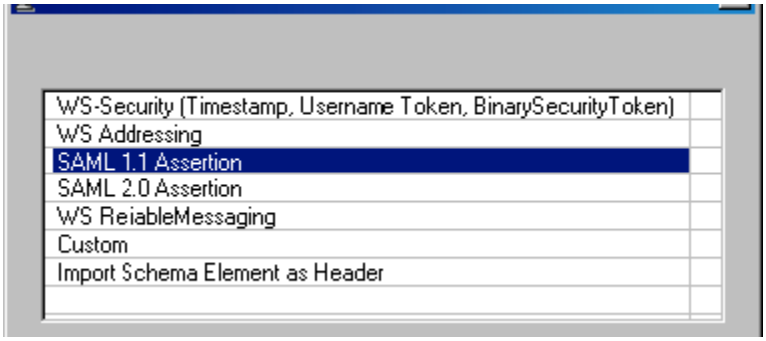


7. Complete the necessary fields for the **Authentication Statement** and click **Finish**. For more information on Authentication Statements, see Addin g and Modifying SAML Statements.

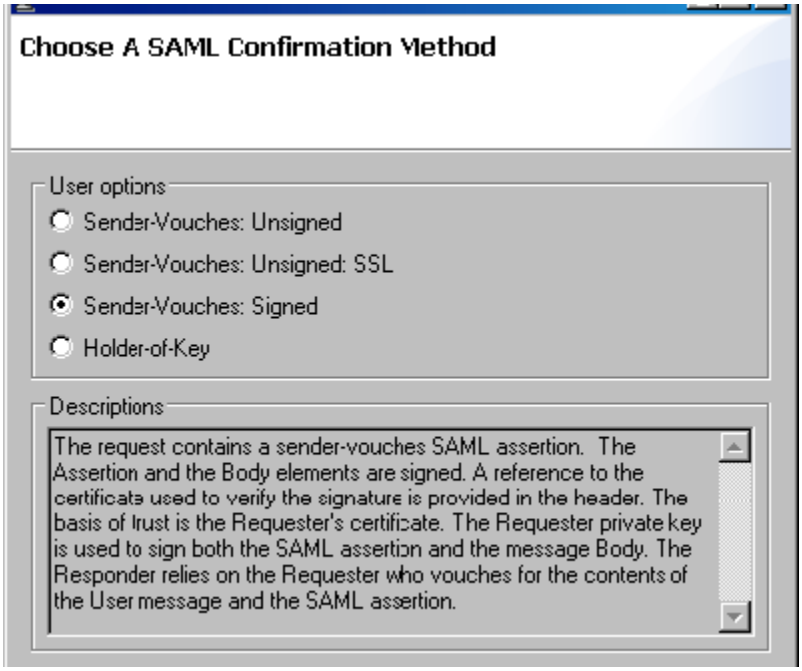
Sender-Vouches: Signed

To add a **Sender-Vouches: Signed** confirmation method, perform the following from the Header sub-tab of the Form Input view:

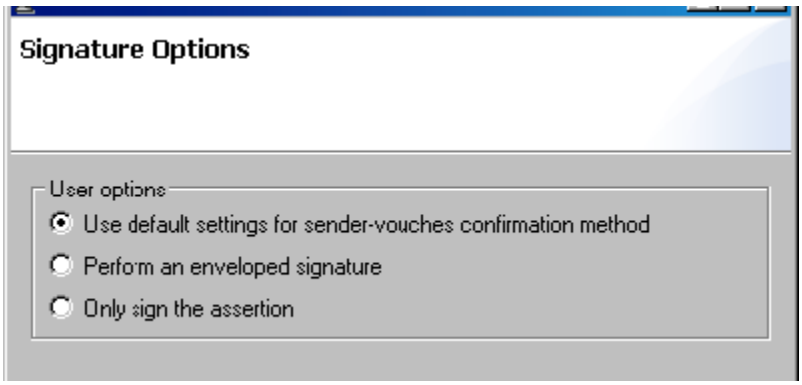
1. Click the **Add** button. The **Add New SOAP Header** dialog displays.



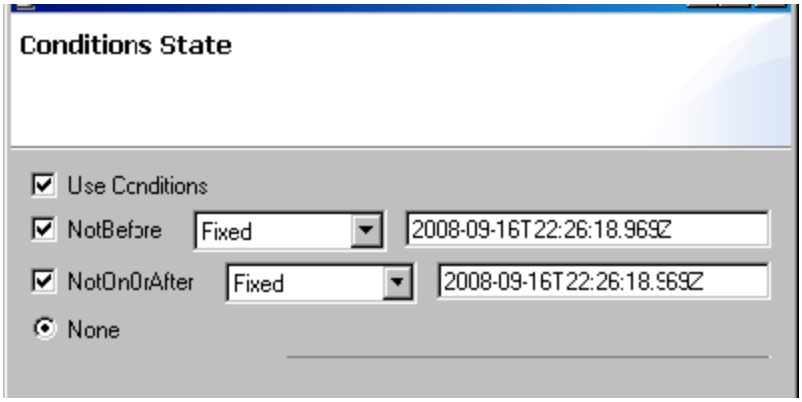
2. Select **SAML 1.1 Assertion** and click **OK**. The SAML Assertion wizard displays.



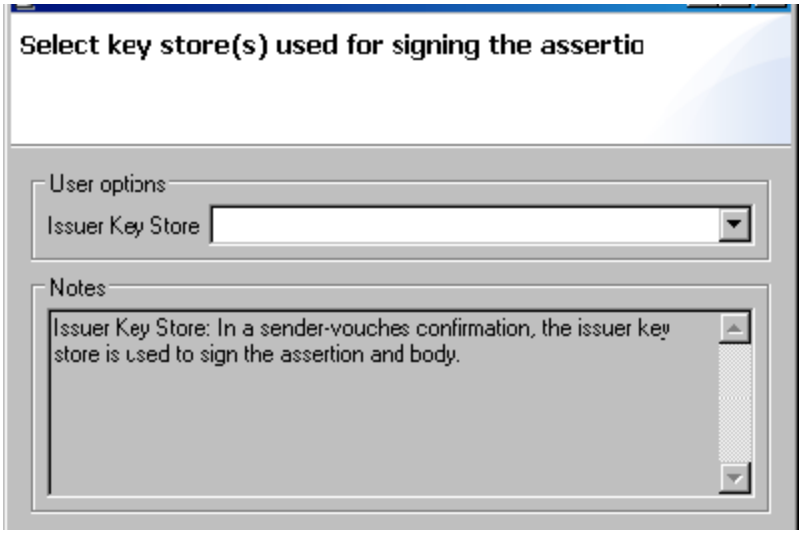
3. Select **Sender-Vouches: Signed** and click the **Next** button.



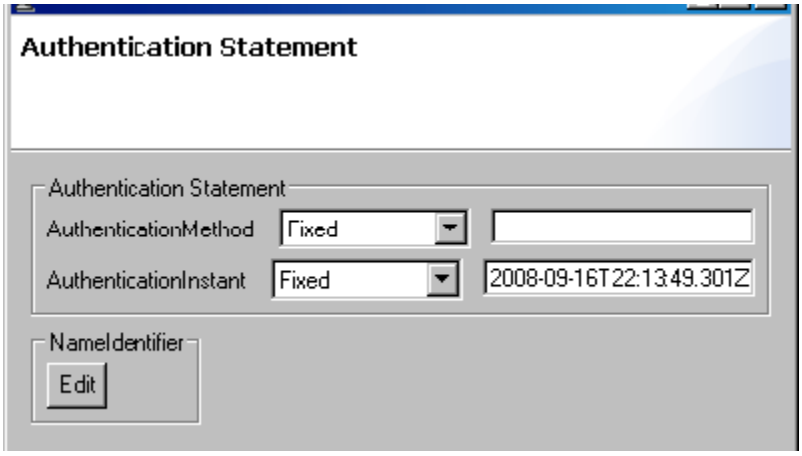
4. Select the desired **Signature Options** and click the **Next** button.



5. Select **Use Conditions**, if needed.
    - If **Use Conditions** was selected, then choose the desired Condition type or None.
    - If **Audience Restriction** was selected and a value greater than 0 was entered, then click **Next** to be prompted by the **Audience Restriction** control and fill each field in accordingly.
6. Click the **Next** button.



7. Select the key store to be used for the issuer signature on the assertion and the body from the **Issuer Key Store** drop down menu and click **Next**.
8. Select the desired SAML statement type and click **Next**.

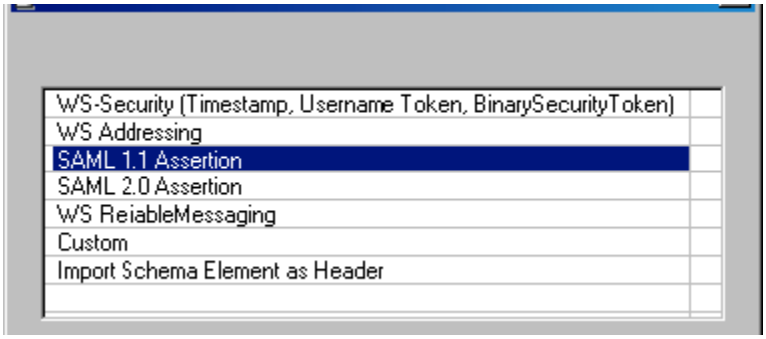


9. Complete the necessary fields for the **Authentication Statement** and click **Finish**. For more information on Authentication Statements, see Adding and Modifying SAML Statements.

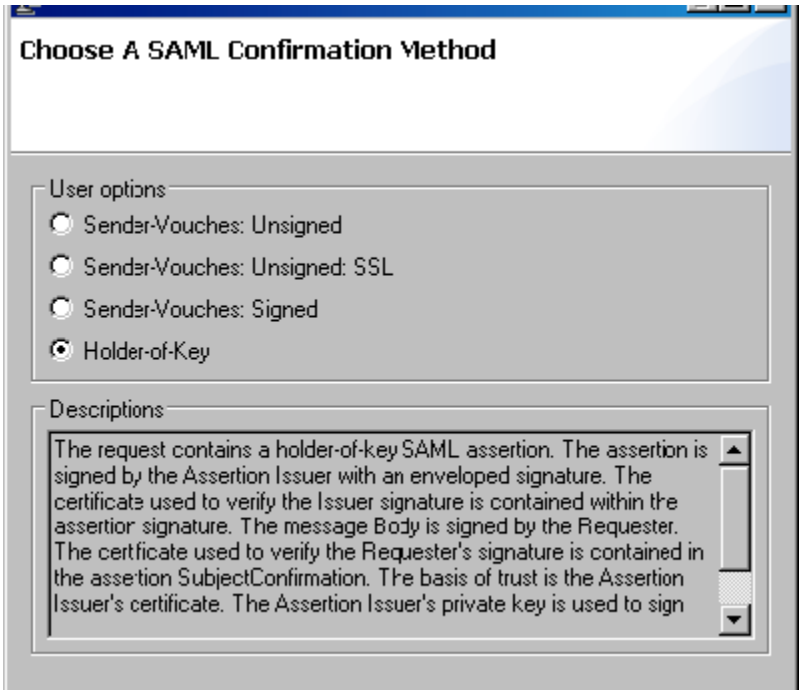
Holder-of-Key

To add a **Holder-of-Key** confirmation method, perform the following from the Header sub-tab of the Form Input view:

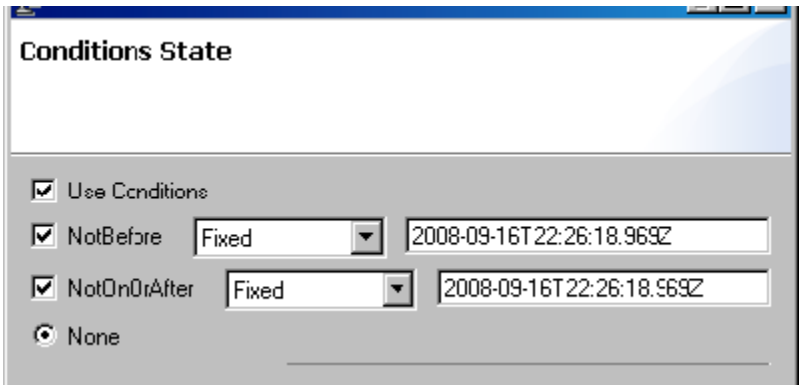
1. Click the **Add** button. The **Add New SOAP Header** dialog displays.



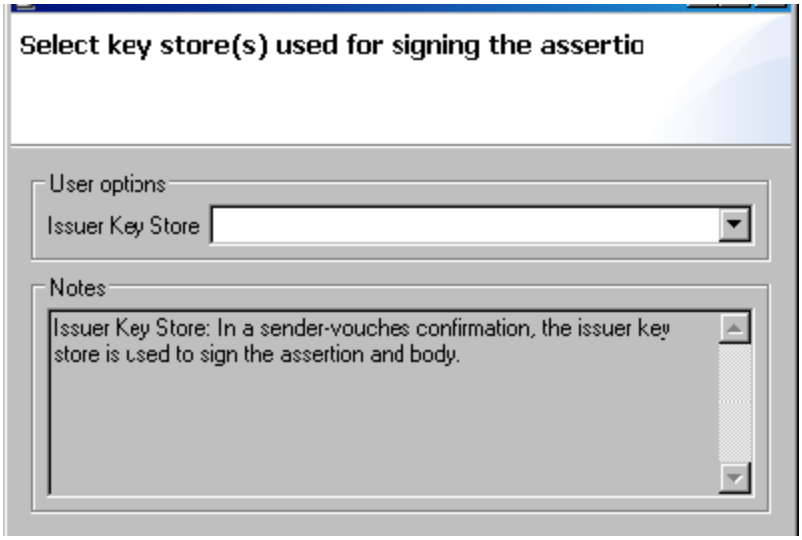
2. Select **SAML 1.1 Assertion** and click **OK**. The SAML Assertion wizard displays.



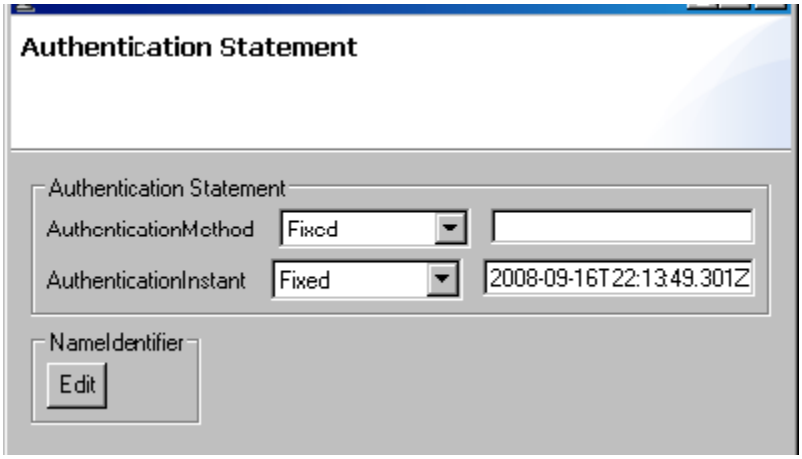
3. Select **Holder-of-Key** and click the **Next** button.



4. Select **Use Conditions**, if needed.
   - If **Use Conditions** was selected, then choose the desired Condition type or None.
   - If **Audience Restriction** was selected and a value greater than 0 was entered, then click **Next** to be prompted by the **Audience Restriction** control and fill each field in accordingly.
5. Click the **Next** button.



6. Select the key store used for the issuer enveloped signature over the Assertion element, and the key store used by the user to sign the body element, and click **Next**.
7. Select the desired SAML statement type and click **Next**.



8. Complete the necessary fields for the **Authentication Statement** and click **Finish**. For more information on Authentication Statements, see Adding and Modifying SAML Statements.

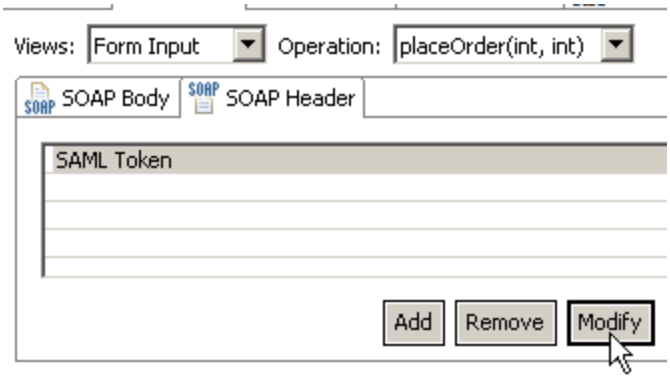
Adding and Modifying SAML Statements

SAML Statements must be added during the creation of an assertion. The following choices are available:

- Authentication Statement
- Authorization Decision Statement
- Attribute Statement

SAML Statements can be added and modified after an assertion has been created in order to extend or customize the assertion. To do this:

1. Select the SAML Assertion in the **SOAP Header** tab.
2. Click **Modify**.



3. Edit the assertion and add/edit SAML statements in the dialog that opens.
    - When adding or modifying a statement, be sure to fill out each of the enabled fields. These are the minimum requirements for the Statements. If desired, enable fields that are disabled by default and fill out the fields for them as well.
    - By default, the SAML assertion element is included under a wsse:Security element (as per the WS-Security SAML profile specification). If you would rather have it placed placed directly under the header (e.g., to allow for compatibility with systems that expect the SAML assertion element to be directly under the SOAP Envelope Header and have no awareness of WS-Security), then go to the **WS-Security Attributes** tab and disable **Wrap with a wsse:Security element**.