Adding Global Test Suite Properties

This topic explains how to create JMS, XPath, SOAP Header, and Database, properties in SOAtest and Virtualize that can be shared and referenced globally across a test or action suite. This section includes:

- Global JMS Connection Properties
- Global Ignored XPath Properties
- Global SOAP Header Properties
- Global Database Account Properties
- Global Key Stores
- Global Tools
- Global WS-Policy Banks

Global JMS Connection Properties

When creating a large test suite with multiple tools, there may be instances where some tools (i.e. SOAP Client, Messaging Client, and Call Back Tool) will use the same JMS Connection Properties.Rather than manually entering the same information into each tool or copying and pasting settings between tools, it may be easier to create JMS settings that each tool can reference. In this case, you can create global JMS Connection Properties at the test or action suite level.

To create global JMS Properties, complete the following:

1. Select the desired node and click the **Add Property** button.
2. In the Add Global wizard, choose **Global Property> JMS Connection Properties** and click **Finish**. A **Properties** node displays and the JMS Connection Properties panel displays in the right side of the GUI.
3. Specify the settings in the JMS Connection Properties panel:
   1. (Optional) Enter the new name in the **Name** field. The name will appear in the tools that reference these properties. You can create more than one global reference for JMS Connection Properties, so the names you specify should be intuitive to how it's used.
   2. Click the **Add Property to All** button. If you don't click this button, the global properties you add will be ignored by the tools in the suite. If **Use Shared Property Only** is selected from the drop-down list, the corresponding tools in the suite will be able to only use the global property you added.
   3. In the **Provider URL field**, specify the location of the JMS Administered Objects.
   4. In the **Initial Context** field, specify the Java class that contains all the JMS properties mappings.
   5. In the **Connection Factory** field, specify the key used to look up the MOM-specific factory from the initial context. This can be either a Queue Connection Factory or a Topic Connection Factory.
   6. In the **Authentication** area, select the **Perform Authentication** check box and enter the **Username** and **Password** to authenticate the request. If the correct username and password are not used, the request will not be authenticated.

> **Only the SOAP Client, Messaging Client, and Call Back Tools can reference global JMS Connection Properties**
>
> After specifying the global JMS Connection Properties, you can share these properties across multiple instances of these SOAtest tools.

Global Ignored XPath Properties

As with global JMS properties, there may be instances when you have multiple Diff tools that use the same XPath settings.Rather than manually entering the same information into each Diff tool or copying and pasting settings between Diff tools, it may be easier to create XPath settings that each Diff tool can reference. In this case, you can create global XPath Properties at the action or test suite level.

To create a global list of ignored XPaths, complete the following:

1. Select the desired suite node and click the **Add Property** button.

   The **Add Global** wizard displays.
2. Select **Global Property> Ignored XPaths** from the **Add Global** wizard and click **Finish**. A **Properties** node displays and the Ignored XPaths panel displays in the right side of the GUI.
3. Specify the settings in the Ignored XPaths panel as follows:
   1. If you want to change the default name, enter the new name in the **Name**. This will be the name that appears in the Diff tools from which you will reference these XPaths. Because you can create more than one global reference list for XPaths, the name you enter should be intuitive to its use.

2. Click the **Add Property to All** button (if you don't click this button, the global properties you add will be ignored by the tools in the suite). Depending on what you select from the drop-down list, one of the following will occur:

   - If **Use Shared Property Only** is selected from the drop-down list, the corresponding tools in the suite will be able to only use the global property you added.
   - If **Use Local and Shared Properties** is selected from the drop-down list, the corresponding tools in the suite will be able to use the global property you added and any properties configured within the individual tool itself.

3. Click the **Add** button. An empty field displays in the **XPath** column of the **Ignored XPaths List**. By default, the **Settings** column is automatically filled in with all XPath operations specified, meaning that the entire XPath you add will be ignored.

4. Using the Ignored XPath setting dialog that opens when you double-click in the **XPath** column, specify an XPath position. The XPath you enter can be shared by multiple diff tools within the suite. If you want to ignore more than one attribute at an element's XPath location, leave the attribute name empty or use the wild card * (e.g. myAttribute*).
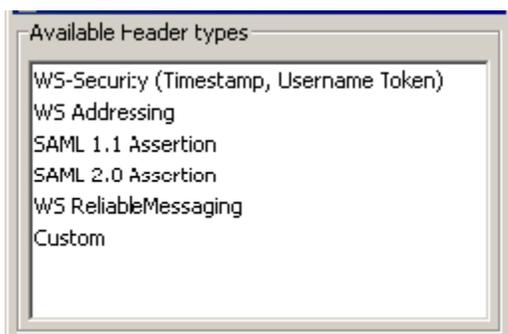
Global SOAP Header Properties

When creating a large test suite with multiple tools, there may be instances where SOAP Client tests will use the same SOAP Header Properties. Rather than manually entering the same information into each tool or copying and pasting settings between tools, it may be easier to create SOAP Headers that each tool can reference. In this case, you can create global SOAP Header Properties at the test or action suite level.

To create a global SOAP Header, complete the following:

1. Select the desired suite node and click the **Add Property** button.

   The **Add Global** wizard displays.

2. Select **Global Property> SOAP Headers** from the **Add Global** wizard and click **Finish**. A **Properties** node displays and the SOAP Headers panel displays in the right side of the GUI.

3. Specify the settings in the SOAP Headers panel as follows:
   1. If you want to change the default name, enter the new name in the **Name** field.
   2. Click the **Add Property to All** button (if you don't click this button, the global properties you add will be ignored by the tools in the suite). Depending on what you select from the drop-down list, one of the following will occur:

      - If **Use Shared Property** Only is selected from the drop-down list, the corresponding tools in the suite will only be able to use the global property you added.
      - If **Use Local and Shared Properties** is selected from the drop-down list, the corresponding tools in the suite set will be able to use the global property you added and any properties configured within the individual tool itself.

   3. Click the **Add** button. The **Choose Header Type** dialog displays.

   ```
   ┌─ Available Header types ──────────────────┐
   │ ┌──────────────────────────────────────┐  │
   │ │ WS-Security (Timestamp, Username Token) │
   │ │ WS Addressing                          │  │
   │ │ SAML 1.1 Assertion                     │  │
   │ │ SAML 2.0 Assertion                     │  │
   │ │ WS ReliableMessaging                   │  │
   │ │ Custom                                 │  │
   │ └──────────────────────────────────────┘  │
   └───────────────────────────────────────────┘
   ```

   4. Select a SOAP Header type from the **Available Header types** list and click OK.
   5. Configure the SOAP Header parameters as needed. For more information on each SOAP Header, see Adding SOAP Headers in SOAtest.

Global Database Account Properties

When creating a large test suite with multiple tools, there may be instances where DB tools will use the same Database Properties. Rather than manually entering the same information into each tool or copying and pasting settings between tools, it may be easier to create a Database Account that each tool can reference. In this case, you can create global Database Account Properties at the suite level.

To create a global Database Account, complete the following:

1. Select the desired suite and click the **Add Property** button.

2. In the Add Global wizard, choose **Global Property> Database Account** and click **Finish**. A **Properties** node displays and the Database Account panel displays in the right side of the GUI.
3. Specify the settings in the Database Account panel as follows:
    1. If you want to change the default name, enter the new name in the **Name** field.
    2. Click the **Add Property to All** button (if you don't click this button, the global properties you add will be ignored by the tools in the suite). Depending on what you select from the drop-down list, one of the following will occur:

        - If **Use Shared Property Only** is selected from the drop-down list, the corresponding tools in the suite will be able to only use the global property you added.
    3. Configure the rest of the Database Account settings as needed.
        - If the account settings are stored in a file, select **File** then specify the path to that file.
            - To refresh/reload the file (e.g., if you edited it externally), click **Refresh Configuration Settings**.
        - If you want to specify the settings in this panel, enable **Local** and specify the driver settings. See

        Database Configuration Parameters for additional information.

            - To export these values to a file, click **Export Configuration Settings**. Once the values are exported to a file, you can import the file through the **File> Input File** control (described above). This way, you won't have to re-type the same values if you want to add this same account to a different suite.

            Note that exported properties files contain the following properties:

                - driver
                - url
                - username
                - password
                - close.connection

            For example:

            ```
            version=1
            driver=org.hsqldb.jdbcDriver
            url=jdbc:hsqldb:hsql://localhost/parabank username=sa
            password=dGVzdA==
            close.connection=true
            ```

Global Key Stores

Key Stores contain the necessary certificates and private keys needed to perform secure Web service through means such as server/client authentication, XML encryption, and XML digital signatures. The values you specify in a Key Store will be available to use with the SOAP Client, XML Encryption, and XML Signer tools.

The SOAP Client Tool can use a Key Store certificate to complete the handshake with a server. The XML Encryption Tool can use a Key Store certificate to encrypt XML documents, and the XML Signer tool can use a Key Store certificate and private key to sign and verify your identity in an XML document.

Important: In order to use Key Stores, you will need to download and install the Unlimited Strength Java Cryptography Extension. For details, see JCE Prerequisite.

To add a key store

1. Select the desired test suite node and click the **Add Property** button.
   The **Add Global** wizard displays.
2. Select **Global Key Store** from the **Add Global** wizard and click **Finish**. A **Properties** node displays in the Test Case Explorer tree and the Key Store panel displays in the right side of the GUI.
3. If you want to change the default name, enter the new name in the **Name** field.
4. Specify the settings in the Key Store panel's Certificate tab as follows:
    1. Select **Use same keystore for private key** if the Key Store contains private keys for the certificate.
    2. In the **Key Store File** field, specify the key store file by clicking the **Browse** button and using the file chooser that opens. If you want the path saved as a relative path (for example, to facilitate project sharing), check the **Persist as Relative Path** option.
    3. In the **Key Store Password** field, specify the Key Store password.
    4. In the **Key Store Type box**, select the type of Key Store being used (e.g. JKS, PKCS12, BKS, UBER, PEM).
    5. Click **Load** to populate the aliases with the available certificates/keys (if the path, type, and key store password are valid), then choose the certificate alias in the **Certificate Alias**.box.
5. Specify the settings in the Key Store panel's Private Key tab as follows:

1. In **Key Store File**, specify the key store file by clicking the **Browse** button and using the file chooser that opens. If you want the path saved as a relative path (for example, to facilitate project sharing), check the **Persist as Relative Path** option.
   - This field is only available if the **Key Store Contains Keys** option is unselected in the Certificate tab.
2. In **Key Store Password**, specify the Key Store password.
   - This field is only available if the **Key Store Contains Keys** option is unselected in the Certificate tab.
3. In **Key Store Type**, specify the type of Key Store being used (e.g. JKS, PKCS12, BKS, UBER, PEM).
   - This field is only available if the **Key Store Contains Keys** option is unselected in the Certificate tab.
4. Click **Load** to populate the aliases with the available certificates/keys (if the path, type, and key store password are valid), then choose the private key alias in the **Private Key Alias**.box.
5. In **Private Key Password**, specify the private key password.

Global Tools

If you expect to use certain specialized tools (for example a particular XSLT tool or a "chained" tool that operates on a request or response, then sends its output to additional tools, which can send their out-put to additional tools, and so on) *only within the context of the current test suite*, you can add them to the test suite tool repository, then add them to the test suite without recreating them each time. (If you plan to use a specialized tool for multiple test suites, you should add it to the program via the **Tools** panel available when you choose **Tools> Customize**.

To add a tool to a test suite's tools repository:

1. Select the desired test suite node and click the **Add Property** button.
   The **Add Global** wizard displays.
2. Select **Global Tool> [Tool_name]** from the **Add Global** wizard and click **Finish**. A new tool node displays in the Test Case Explorer tree (under the **Tools** branch, which will be added if it did not already exist) and a tool configuration panel displays in the right side of the GUI.
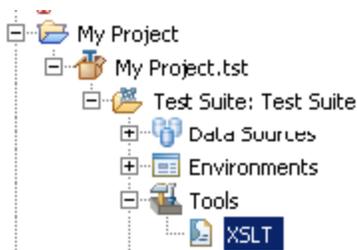
3. Customize that tool's settings in the tool configuration panel that opens.
4. You can chain additional tools to that tool as described in Adding a Single Output.

To use a repository tool in a test, select it from the **Existing Tools** list that is available when you add a test or output.

Global WS-Policy Banks

One of the biggest aspects of web services is interoperability. web services rely on a standardized interface to declare what requirements must be met in order for a service consumer to interact with a service provider. The basic WSDL specification does not have the capacity to declare complex client-side requirements. To accommodate for this, WSDL is extended with WS-Policy and WS-PolicyAttachment, allowing a service provider to define additional requirements within the WSDL. WS-Policy leaves it up to other WS-* specifications to define their own set of policies. One such specification is WS-SecurityPolicy which defines policies related to WS-Security.

When reading a WSDL with SecurityPolicy extensions, SOAtest automatically generates the test cases with all the necessary policy related configurations. There are some attributes of the test case that still require manual configuration, but SOAtest will automatically set up the foundation.

> **Note**
>
> WS-Policy is a lightweight specification. It delegates policy design to the WS-* specifications; in addition, there is a large set of proprietary policies. Since the WS-* space is large, SOAtest only sup-ports WS-SecurityPolicy assertions but will continue to extend the processor to handle other common assertion sets.

To add a global WS-Policy Bank, complete the following:

1. Select the desired test suite node and click the **Add Property** button.
   The **Add Global** wizard displays.

2. Select **WS-Policy Bank** from the **Add Global** wizard and click **Finish**. A new WSDL Policies node displays in the Test Case Explorer tree (under the **WS-Policy Banks** branch, which will be added if it did not already exist) and a WSDL Policies configuration panel displays in the right side of the GUI and various WS-security tests will be chained to the SOAP client tools.
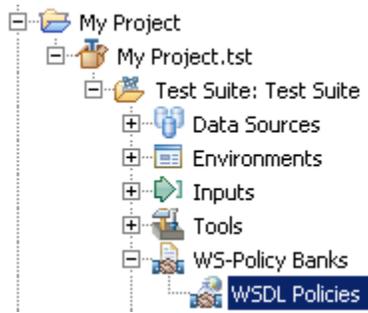
3. Specify the settings in the WSDL Policies panel as follows:
    1. If you want to change the default name, enter the new name in the **Name** field.
    2. In **WSDL URI**, specify the WSDL URI where this Web service can be accessed. You can either enter a WSDL or click the **Browse** button.
    3. Click **Refresh from WSDL** to refresh the WSDL from the given location URL and reparses it.
    4. In the **Global Policies** area, review the policy definitions in non-XML format. as well as the policy alternatives implied by your WSDL. Each section in the left hand tree represents a global policy element in the WSDL