

XML Transformer

This topic explains how to configure and apply the XML Transformer tool, which gives you XSLT-like functionality to transform any XML. Sections include:

- [Understanding XML Transformer](#)
- [Configuring XML Transformer](#)
- [Adding Regression Controls](#)
- [Parameterizing XPath](#)
- [Handling Empty Elements to Maintain the Integrity of the XML Response](#)

Understanding XML Transformer

SOAP messages often contain a large XML payload. You may only be interested in only a small portion of that payload and wish to create regression controls using only a few elements of the response or request. Ignoring all of these elements one at a time can become very tedious. In this case, it would be more efficient to use the XML Transformer tool.

The XML Transformer gives you XSLT-like functionality to transform any XML. This is very useful if you would like to create a regression control using only a few elements of the SOAP response or request.

Configuring XML Transformer

To configure the XML Transformer, complete the following:

1. Select the main test suite node and click the **Add test or output** button.
2. Select **Standard Test> XML Transformer** from the Add Test wizard, and then click the **Finish** button. A **XML Transformer** node displays in the test suite.
3. Double-click the **XML Transformer** node.

You can configure the following options:

- **Name:** Specifies the name of the Transformer tool.
- **Tree/Literal/Element:** Specifies the expected XML response. A **Literal** tab, a **Tree** tab, and an **Element** tab are available for viewing the expected XML in literal, tree, or simple element form.
- **Selected Element:** Specifies the element you would like to add as a parameterized value.
- **Extract Element:** To add an element, select a value from the **Selected Element** list and click the **Extract Element** button. The value you added displays in the **Selected Element** list with a Data Source Column name corresponding to the tool the value was chosen from and the value name.
- **Remove:** To remove an XPath, select a value from the **Selected Element** list and click the **Remove** button.
- **Modify:** To modify an XPath, select a value from the **Selected Element** list and click the **Modify** button. The **Modify** dialog displays and contains the following options:
 - **XPath:** Displays the selected XPath. If you are looking for a more general XPath, you can easily type in a different number into the list indices. For example [1] can be changed to [2] if you are interested in the 2nd occurrence only. After editing the XPath text, click the **Validate** button to validate the XPath format, click **OK**.
 - **Options:** Allows you to extract either the Entire Element, or the Content Only.
 - **Entire Element:** Selecting **Entire Element** will output the entire XPath. For example, `xPath/Parent` will output `<parent>VALUE</parent>`. You can configure the following **Entire Element Options**:
 - **Index to extract:** Default value is 1. This controls which element is extracted in the case where the element occurs more than one time.
 - **Content Only:** Selecting **Content Only** will output only the value. For example, `xPath/Parent` will output `VALUE`. You can configure **Text Content:** Extracts the text content of the element selected, or **All Child Nodes:** Extracts all child nodes of the element selected.
 - **XPath Evaluation:** Clicking the **Evaluate** button displays the result of applying the XPath expression against the expected XML.
 - **Save Expected XML:** Specifies whether or not to save the expected XML.
 - **Canonicalize XML Output:** Specifies whether or not an extracted element is canonicalized. If this option is selected, and if an entire element is extracted, any necessary namespace declarations are added to the extracted element if the element contains prefixes referencing namespaces that are not declared in the same element.
 - **Allow Alteration:** Specifies whether to allow the alteration of an XPath. When this option is selected, an **Extract** tab and an **Alter** tab display beneath the **Selected Element** list. To alter an XPath, select the **Allow Alteration** check box, select the **Alter** tab, add an XPath by clicking the **Add** button, and then modify the XPath by clicking the **Modify** button. The **Modify** dialog displays and contains the following options:
 - **XPath:** Displays the selected XPath. To edit and validate a selected XPath, edit the XPath text, click the **Validate** button to validate the XPath format, click **OK**.
 - **Alteration Type:** Allows you to select how the **Value** you enter alters the XML.
 - Selecting **Append** will add the altered value to the end of the XPath.
 - Selecting **Prepend** will add the altered value to the beginning of the XPath.
 - Selecting **Replace With** will replace the entire XPath with the altered value you specify.
 - **Alteration Value:** Allows you to enter the value you would like to alter the XPath with.
 - **Extract Empty Elements As:** Specifies whether or not empty XML elements will be extracted. When this option is enabled, you can use the adjacent text field to specify a text string that indicates what "placeholder" value should be added for every empty extracted element.

Note that additional configuration is required if you use this option; see [Handling Empty Elements to Maintain the Integrity of the XML Response](#).

- **Wrap Output in XML:** Specifies whether the output of the XML Transformer tool will be well-formed XML.

If this option is selected, the content will be output as well-formed XML. For example, the XML encoded value of:

```
&lt;foo
```

would be output as:

```
<root> <ElementName>&lt;foo&gt;</ElementName> </root>
```

Adding Regression Controls

You can add a regression control using the XPath values you extracted or altered. Right-click the SOAP Client node that contains an XML Transformer and select **Create Regression Control** from the shortcut menu. A **Transformed XML > Diff** node appears underneath the XML Transformer node. If you select this node, you will see the XPaths you added to the **Selected XPaths** list in the XML Transformer tool.

Parameterizing XPaths

You can parameterize XPaths to reference test or Responder suite variables, environment variables, and data source values. The syntax to reference variables is `${myVariableName}`. The syntax to reference XML Data Bank values and Data Source Values is: `${myColumnName}`.

For example, if `${XPath Key}` is a data source column name, you could use

```
//*[local-name(.)="bookstore" and namespace-uri(.)=""]//*[local-name(.)="book" and namespace-uri(.)=""][child::node()[local-name(.)="title" and text()="${XPath Key}"]]
```

Handling Empty Elements to Maintain the Integrity of the XML Response

By default, empty elements will not be extracted. This could impact the integrity of the XML response for use in a Writable Data Source. For instance,

```
<e>5</e>
<e />
<e>6</e>
```

would produce a Writable Data Source that looks like

```
ROW 1 = 5
ROW 2 = 6
```

If you prefer to have all nodes—including empty elements—represented in the extraction:

1. Enable **Extract empty elements as**.
2. (Optional) If you want a "placeholder" value to be added for every empty extracted element, indicate the desired placeholder string in the adjacent text field.

Options

- Save expected XML
- Canonicalize XML output
- Allow alteration
- Extract empty elements as:

3. Select one of the XPaths that specifies what to extract, then click **Modify**.
4. Ensure that the **Extract** option is set to **Content only**.
5. Delete the `/text()` at the end of the related XPath(s) For example, if the current XPath is

```
/*[local-name(.)="Envelope" and namespace-uri(.)="http://schemas.xmlsoap.org/soap/envelope/"]/.../*  
[local-name(.)="i" and namespace-uri(.)=""][1]/text()
```

Edit it to

```
/*[local-name(.)="Envelope" and namespace-uri(.)="http://schemas.xmlsoap.org/soap/envelope/"]/.../*  
[local-name(.)="i" and namespace-uri(.)=""][1]
```

XPath

```
/*[local-name(.)="Envelope" and  
namespace-uri(.)="http://schemas.xmlsoap.org/soap/envelope/"]/. . /*[local-name(.)="i"  
and namespace-uri(.)=""][1]
```

Options

Extract:

6. Repeat steps 3-5 for additional XPath's as needed.