XML Data Bank

This topic explains how to configure and apply the XML Data Bank tool in SOAtest and Virtualize. This tool that extracts XML values (e.g., from a request or response message) so that they can be used in another place. Data can also be sent to a Writable Data Source and accessed in the Extension Tool, or it can be sent to variables for easy reuse across the test suite (SOAtest) or Responder suite or Action suite (Virtualize).

Sections include:

Understanding XML Data Bank

The XML Data Bank tool enables you to extract certain XML values (e.g., from a request or response message) so that they can be used in another place. The XML Data Bank tool can be chained to any other tool that outputs XML. It can extract any information from the XML and make that information available for later use.

For example, you can configure a test suite that tests a bank's Web service transactions. Test 1 of that test suite can log on to the service using a User ID, then the SOAP response would return a session ID back to Test 1. Test 2 of that test suite can be configured to use the session ID from Test 1 to perform transactions. You can configure any of the tests in a test suite to use SOAP response parameters as SOAP request parameters.

Users typically configure an XML Data Bank by accessing the "Use Data SourceWizard" while parameterizing a value in a tool such as the SOAP Client or Messaging Client tool. This provides a quick, intuitive, and largely automated way to extract data from one tool and use it another. You simply go to the tool where you want to insert extracted data, then use a wizard to specify what data (e.g., from what tool) you want to extract. This is the usage model demonstrated in the Storing Results to Be Used in Subsequent Tests tutorial. This same method can be used to extract data that is used to set a variable. Alternatively, you can manually configure an XML Data Bank tool to extract data from one tool, then manually configure other tools to use the extracted values.

Video Tutorial

In this video, you'll learn how to extract values from XML responses and reuse them in other tests.

 <h1 id="XMLDataBank-ConfiguringXMLDataBankUsingtheDataSourceWizardConfiguringXMLDataBankUsingtheDataSourceWizard">Configuring XML Data Bank Using the Data Source Wizard</h1><h2 id="XMLDataBank-ConfiguringtheExtraction">Configuring the Extraction</h2> To use the "Use Data Source Wizard" wizard to configure an XML Data Bank: Ensure that you have an action set or test suite with at least two tools. In the configuration panel for the tool that you want use the extracted value, select one of the available Form views.
From the Operation drop-down menu, select the operation that you want to use the extracted value.
In the element view (for example, the id value), go to the message element that you want to use the extracted value, then select Parameterized and Use Data Source Wizard from the available drop-down menus.

<img class="confluence-embedded-image" src="/download/attachments/41307125/UseDataSource.png?version=1&modificationDate=1536604878075&api=v2" data-image-src="/download/attachments/41307125/UseDataSource.png?version=1&modificationDate=1536604878075&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307132" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="UseDataSource.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

In the wizard that opens:
<li style="list-style-type: none;background-image: none;"> Select the tool you want to extract a value from. The drop-down menu at the top of the panel will contain all tools in the test or Responder suite that occur before the current tool you are configuring. For example, if you are configuring Tool 4, tools 1, 2, and 3 will display in this menu along with any data sources that may be available. Using the controls on the left side of the panel, indicate what you want to extract and add it to the right side of the panel. The right panel lists the values you have configured for extraction, and shows the name of the data source column where they will be stored (if you keep the default setting).
(Optional) If you want to specify additional options (e.g., if you want to change the name of the column used to store the value, you want the value saved to a writable data source, or you want the value stored to an existing variable) —or if you want to modify advanced XPath settings—then select the appropriate element in table on the right and click Modify. Next, configure the options as needed, then click OK. Available options are described in Options for Each Extracted Element.
<h1 id="XMLDataBank-ConfiguringtheXMLDataBankManually">Configuring the XML Data Bank Manually</h1> You can also manually chain the XML Data Bank tool to a tool within the Responder, Action, or test suite. To configure the XML Data Bank as a chained tool, complete the following: Ensure that you have an action set or test suite with at least two tools. Right-click the node for the tool associated with the data you want to extract, then choose Add Output. In the Add Output wizard, indicate where you want to extract the value from (e.g., SOAP Envelope) and click the Finish button. An XML Data Bank node displays below the tool. Configure the tool as follows: Use the available controls to specify the XPaths that indicate what value you want to extract. To add an XPath, select a value from the Expected Message list and click the Extract Element button. The value you added displays in the Selected Element list with a Data Source Column name containing the name of the tool the value came from, as well as the extracted value. <img class="confluence-embedded-image" src="/download/attachments/41307125/tree.png?version=1&modificationDate=1536604878033&api=v2" data-image-src="/download/attachments/41307125/tree.png?version=1&modificationDate=1536604878033&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307130" data-linked-resource-version="

1" data-linked-resource-type="attachment" data-linked-resource-default-alias="tree.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13"> Working with the Expected Message The left panel displays the expected XML response used to create a template from which you can select elements. If this tool receives a valid XML message (e.g., from traffic or as defined by the client tool it is attached to), this panel will be populated automatically. Alternatively, you can copy a sample message into the Literal or Tree tabs. Note that the expected XML does not get saved by default; if you want to save it, enable the Save Expected XML option. If you want to further configure the XPath or customize extraction settings for this element, click Modify, then modify it as desired. See Options for Each Extracted Element for details. Repeat steps a and b as needed to configure any additional extractions you want performed.
In the bottom area of the XML Data Bank configuration panel, customize the options as desired. See Tool-Wide Extraction Options for details.
<h2 id="XMLDataBank-UsingtheExtractedValue">Using the Extracted Value</h2> After adding and/or modifying the extraction, configure the tool that you want to use the extracted data. Set the value to Parameterized, and choose the appropriate item from the drop-down. For example, if you saved the value to the "title" data source column, you would select it as follows.

<img class="confluence-embedded-image" src="/download/attachments/41307125/title2.png?version=1&modificationDate=1536604878011&api=v2" data-image-src="/download/attachments/41307125/title2.png?version=1&modificationDate=1536604878011&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307129" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="title2.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

 <h1 id="XMLDataBank-ConfigurationOptions">Configuration Options</h1> You can configure the following options when configuring an XML Data Bank: Options for Each Extracted Element
Tool-Wide Extraction Options
<h2 id="XMLDataBank-OptionsforEachExtractedElementOptionsforEachExtractedElement">Options for Each Extracted Element</h2> The following options can be set by modifying a selection listed in the right panel of the XML Data Bank configuration panel. XPath Options XPath: Displays the XPath indicating the value to extract. If you are looking for a more general XPath, you can easily type in a different number into the list indices. For example <code>[1]</code> can be changed to <code>[2]</code> 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.
Extract: Allows you to specify exactly what is extracted.
Entire Element: Selecting Entire Element will output the entire XPath. For example,<code>XPath/Parent</code> will output <code><parent>VALUE</parent></code>. You can configure Index to extract, which controls which element is extracted if the element occurs more than once.
Content Only: Selecting Content Only will output only the value. For example, <code>XPath/Parent</code> will output <code>VALUE</code>. You can configure Text Content, which extracts the text content of the element selected, or All Child Nodes, which extracts all child nodes of the element selected.
XPath Evaluation: Clicking the Evaluate XPath button displays the result of applying the XPath expression against the expected XML. Data Source Column Options Custom column name: Specifies the name of the data source column in which to store the value. Values are stored in an internal data source unless you specify otherwise (e.g., if you select Writable data source column or Variable). This is the name you will use to reference the value in other places. For example, if it is stored in a data source column named My Value, you would choose My Value as the parameterized value. You could also reference it as <code>${My Value}</code> in literal or multiple response views.
<img class="confluence-embedded-image" src="/download/attachments/41307125/myValue.png?version=1&modificationDate=1536604877978&api=v2" data-image-src="/download/attachments/41307125/myValue.png?version=1&modificationDate=1536604877978&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307127" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="myValue.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

  Writable data source column: Enables storing the value in a writable data source column (see Configuring a Writable Data Source for details). This allows you to store an array of values. Other tools can then iterate over the stored values. Write to all columns that match: Enables storing the value in all columns whose name contains the specified string. When extracting multiple values from a message, each value will be written across all the columns that match. In contrast, if you pick a single writable data source column (the above option), then the values will be written down the column across multiple rows. Variable: Enables saving the value in the specified variable so that it can be reused across the current Responder, Action, or test suite. The variable must already be added to the current suite as described in Defining Variables in SOAtest. Any values set in this manner will override any local variable values specified in the Responder, Action, or test suite properties panel. <h2 id="XMLDataBank-Tool-WideExtractionOptionsTool-WideExtractionOptions">Tool-Wide Extraction Options</h2> The following options can be configured in the lower portion of the XML Data Bank tool configuration panel. 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 Extract Element 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 Evaluate XPath button to validate the XPath format, then click OK.
Alteration Type: Allows you to select how the Value you enter alters the extraction. Selecting Append will add the altered value to the end of the extraction. Selecting Prepend will add the altered value to the beginning of the extraction. Selecting Replace With will replace the entire extraction with the altered value you specify.
Alteration Value: Allows you to specify either a fixed or parameterized value using a data source.
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.
This option applies when XPath locates a node, but that node has no text content. For example, if you had <parent><child/></parent>, the XPath //*[local-name()="child"]/text() finds the child element, but it has no text content.
Note that additional configuration is required if you use this option; see Handling Empty/Missing Elements to Maintain the Integrity of the XML Response.
Extract missing elements as: Specifies whether or not missing 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.
This option applies when XPath fails to locate any matching nodes. For example if given a simple xml document such as: <parent><child>name</child></parent>, the XPath //*[local-name()="sibling"] would find 0 nodes.
Note that additional configuration is required if you use this option; see Handling Empty/Missing Elements to Maintain the Integrity of the XML Response. <h1 id="XMLDataBank-

ViewingtheDataBankVariablesUsedDuringTestExecution">Viewing the Data Bank Variables Used During Test Execution</h1> You can configure the Console view (Window> Show View> Console) to display the data bank variables used during test execution. For details, see Monitoring Variable Usage. <h1 id="XMLDataBank-ParameterizingXPaths">Parameterizing XPaths</h1> You can parameterize XPaths to reference Responder suite or test 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: ${myColumnName}.</p><p>For example, if <code>${XPath Key}</code> is a data source column name, you could use <code>/*[local-name(.)="bookstore" and namespace-uri(.)=""]/*[local-name(.)="book"</code> and <code>namespace-uri(.)=""][child::node()[local-name(.)="title" and text()="${XPath Key}"]]</code> <h1 id="XMLDataBank-HandlingEmptyHandlingEmpty/MissingElementstoMaintaintheIntegrityoftheXMLResponse">Handling Empty/Missing Elements to Maintain the Integrity of the XML Response</h1> By default, empty and missing elements will not be extracted. This could impact the integrity of the XML response for use in a Writable Data Source. For instance, assume you have the following XML: <pre class="syntaxhighlighter-pre" data-syntaxhighlighter-params="brush: java; gutter: false; theme: Confluence" data-theme="Confluence"><?xml version="1.0" encoding="UTF-8"?> <root> <e>5</e> <e/> <e>6</e> </root></pre> Also assume that you want to create an extraction for all 'e' elements. To do this, select the first element, then click Extract Element.

<img class="confluence-embedded-image" src="/download/attachments/41307125/treeE.png?version=1&modificationDate=1536604878056&api=v2" data-image-src="/download/attachments/41307125/treeE.png?version=1&modificationDate=1536604878056&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307131" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="treeE.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

Next, select the new extraction, then click Modify. In the Modify dialog, change the XPath to extract from <code>/root/e[1]/text()</code> to <code>/root/e/text()</code>.

<img class="confluence-embedded-image" src="/download/attachments/41307125/SelectedPath.png?version=1&modificationDate=1536604877995&api=v2" data-image-src="/download/attachments/41307125/SelectedPath.png?version=1&modificationDate=1536604877995&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307128" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="SelectedPath.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

This will extract all three text nodes. Next, click Data source column and select a writable data source column. When the Data Bank is executed, the writable data source will only contain two rows because the second element is missing text: ROW 1 = 5
ROW 2 = 6 If you want to write the empty value, change the XPath from<code> /root/e/text()</code> to <code>/root/e</code> and ensure that Content Only is enabled.

<img class="confluence-embedded-image" src="/download/attachments/41307125/ContentOnly.png?version=1&modificationDate=1536604877954&api=v2" data-image-src="/download/attachments/41307125/ContentOnly.png?version=1&modificationDate=1536604877954&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="41307126" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="ContentOnly.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="41307125" data-linked-resource-container-version="13">

This way, all three elements are extracted, including their content. You can optionally enable the Extract empty element as option to change the value extracted for the empty value. The Data Bank will now append the empty value to the writable data source: ROW 1 = 5
ROW 2 =
ROW 3 = 6 <h1 id="XMLDataBank-RelatedTutorials">Related Tutorials</h1> The following tutorial lesson demonstrates how to use this tool: Scenario Testing   </body> </html>