

Configuring a Data Bank

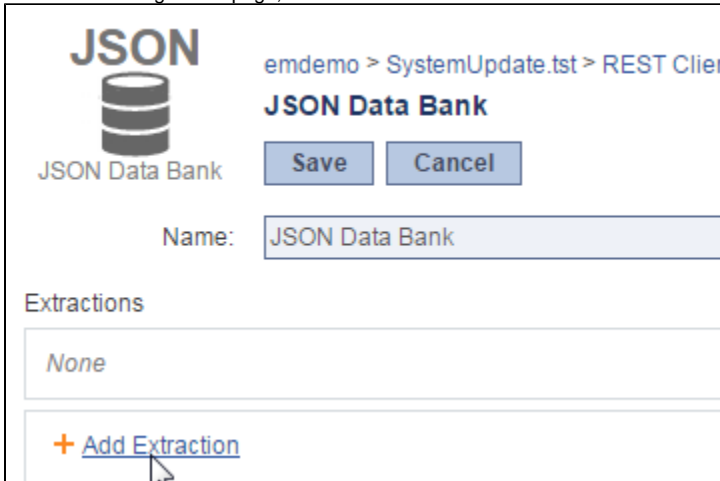
In this section:

- [Basic Data Bank Configuration](#)
- [Configuring a REST URL Data Bank](#)
- [Configuring a Data Generator](#)

Basic Data Bank Configuration

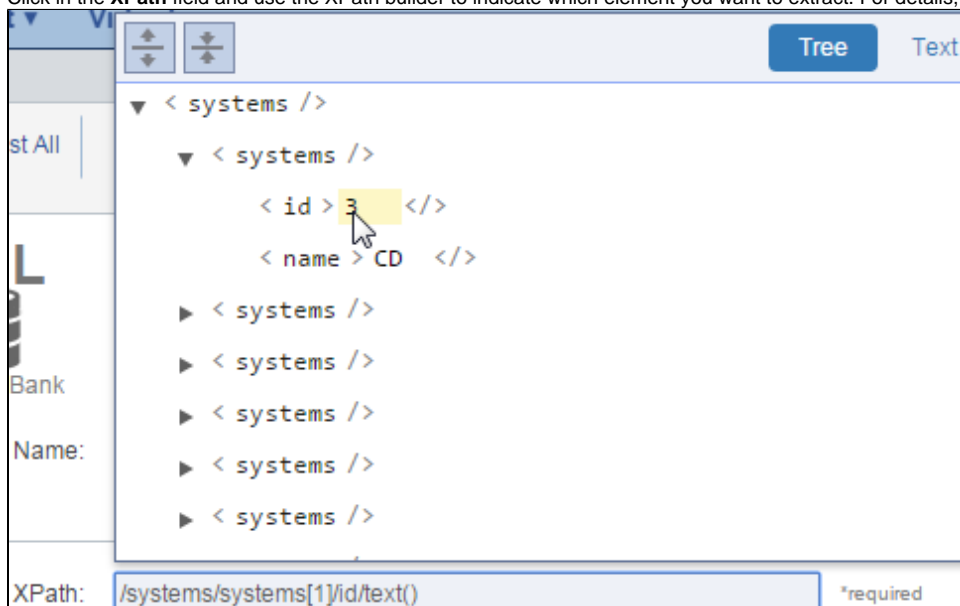
To configure a Data Bank tool:

1. In the tool's configuration page, click **Add Extraction**.



2. Specify your extraction as follows:

- a. Click in the **XPath** field and use the XPath builder to indicate which element you want to extract. For details, see [Specifying XPath](#).



- b. Leave **Extract** to set to **Custom column**. Note that if you are editing a Data Bank tool that was created in SOAtest Desktop, additional options (Writable data source column, Suite variable) may be shown; you can use CTP to change from your existing setting to **Custom column**.

- c. Under **Column name**, specify the name of the "Custom Column" to which you want the value extracted. This will be used to reference the extracted value throughout the test scenario. For example, assume you extract into a custom column named systemID:

Extractions

XPath:

Extract to:

Column name:

You could later reference that extracted value using `${systemID}`

Payload:

Format:

Content type:

Input mode:

```

1 {
2   "systemId": ${systemId},
3   "key": "Version",
4   "value": "1.0",
5   "masked": false
6 }
```

- d. Indicate whether you want to extract the text content, the entire element, or all child nodes.

i **Extraction Options**

Text content extracts the text content of the selected element. For example, `XPath/Parent` will output `VALUE`.

Entire element extracts the entire XPath. For example, `XPath/Parent` will output `<parent>VALUE</parent>`. You can configure Index to extract, which controls which element is extracted if the element occurs more than once.

All child nodes extracts all child nodes of the element selected.

3. Repeat the above process for each additional extraction you want to add. A single Data Bank can perform multiple extractions
4. Review and modify tool options as needed. Available options include:
 - **Canonicalize XML output (for XML Data Banks only):** Specifies whether extracted elements are canonicalized. It applies only if an entire element is extracted. If this is enabled and the extracted element contains prefixes referencing namespaces that are not declared in the same element, then any necessary namespace declarations will be added to the extracted element.
 - **Allow alteration:** If you want to configure XPath alterations (i.e., appending, prepending, or replacing values), open this tool in SOAtest desktop and use the **Allow alteration** controls in that UI.
 - **Extract empty elements as:** Specifies whether empty elements will be extracted. It applies when the extracted element is an empty string. 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.
 - **Extract missing elements as:** Specifies whether missing elements will be extracted. It applies when the extraction fails to locate any matching nodes. 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.
 - **Save expected** saves the message received by the associated test client. This message is used by the XPath builder.

Configuring a REST URL Data Bank

URL Data Bank only applies to Service Virtualization and requires Parasoft Virtualize 9.10.2 or higher. This tool enables you to extract the following values from a REST URL and store them in a data bank:

- Host
- Port
- HTTP Method
- Path (by index)
- Parameter (by name)

You can extract parameters that occur multiple times by concatenating the values. This tool is only available in CTP for message responders.

1. In the tool's configuration page, click **Add Extraction**.

2. Choose a value type from the drop-down menu.

URL Data Bank

APE > Documentation PVA.pva > /parabank/services/t

URL Data Bank

Save Cancel

Name: URL Data Bank

Extractions:

http://10.1.40.206:9080/parabank/services/bank

Type: Host

Custom column: Host

+ Add Extraction

Port

Method

Path

Query

3. Specify the name of the data source column in which to store the value.
4. Add additional extractions for the URL values you want to extract. You can only extract one host, port, and method, but you can add several path and parameter values to extract complex URLs.
5. Specify any additional tool options and click **Save**.

Tool Options

The following options can also be configured:

- **Extract empty elements as:** Specifies whether or not empty 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.
- **Extract missing elements as:** Specifies whether or not missing 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 missing extracted element.
- You can specify a position in the URL path to extract in the Request template field:

```
(<method>) <host>:<port>/<path 0>/<path 1>/<path . . .>/<parameter>
```

http://10.1.40.206:9080/function/newFunctions

Type: **Method**

Custom column: **METHOD** *required

+ Add Extraction

Options:

Extract empty elements as:

Extract missing elements as:

Request template:

URL: **/function/newFunctions**

Configuring a Data Generator

The Data Generator tool stores data, such as numbers, strings, and dates into data sources or test suite variables for use in other tools. For additional information, see [Data Generator](#) in the SOAtest/Virtualize documentation.

1. Select a test or virtual asset and choose **Add Data Generator** from the action menu.

docs > users > anonymous

bandcamp-check-new-arrivals-gr

Execute Create Job

Name: bandcamp-check-new-arrivals-gr

Description:

Tags: None

Filepath: /TestAssets/users/anonymous/ba

Server: docs

- Add Variable Set
- Add Test Suite
- Add DB Client
- Add REST Client
- Add SOAP Client
- Add CSV Data Source
- Add Excel Data Source
- Add Repository Data Source
- Add SQL Data Source
- Add Data Generator**
- Add Extension Tool
- Add JSON Validator
- Add Write File Tool
- Lock Test Scenario

2. (Optional) Specify a name for the tool.
3. If you are adding a data generator to a test scenario, you can add requirements (see [Associating Tests with Work Items](#)).
4. Click **+Add Generator** and choose a generator type from the drop-down menu. You can generate strings, numbers, and date/times.

Generators:

Type: String

Name: *required

Custom column: String *required

Pattern: ⓘ

Character map: !"#%&'()*+,-./0123456789;=<=>?@ABCDEFGHIJKLMNQRST *required

+ Add Generator

5. Specify the name of the data source column where the generated values will be stored in the Custom column field.
6. Specify the generator options. Each type of generator has different options. See the following sections for details on configuring each type.
 - [Configuring String Generators](#)
 - [Configuring Number Generators](#)
 - [Configuring Date/Time Generators](#)
7. Click **Save** to finish adding your tool.

Configuring String Generators

The following fields are used to configure how strings are generated.

Pattern

This field defines the structure of the string. The following special characters are used to define the pattern:

#	This character is replaced with a number 0-9 when the tool runs.
&	This character is replaced with a character as defined in the character map.
\	Escapes the # and & characters, as well as other \ characters, so that they can be included in the generated string.

Click on the information icon for a quick explanation of the characters.

Type: String

Name: *required

Custom column: *required

Pattern: ⓘ

Character map: *required

If you want to implement dynamically generate a pattern, you can click the script icon in the Pattern field and choose a language from the drop-down menu.

Pattern: ⓘ

Character map:

Generator:

Switch to Scripted Input

Enter your script into the input field when prompted.

Pattern: Groovy ▼

Method: execute(com.parasoft.api.ScriptingContext) ▼

Use data source

```

1 import com.parasoft.api.*;
2
3 boolean execute(ScriptingContext context) {
4     // TODO: Not yet implemented
5     return "";
6 }

```

Choose **Fixed** from the Pattern drop-down menu to switch back to a static pattern.

Pattern: Groovy ▲

Method: Fixed

Character Map

This field specifies how characters map to the pattern characters. The default value when set to fixed is all printable ASCII characters, 0x20 (space) through 0x7e (-).

Configuring Number Generators

The number generator enables you to randomly generate numbers between a minimum and maximum value, as well as set the number of decimal spaces. Negative numbers are supported.

Defaults

When the minimum, maximum, and decimal place values are set to Fixed and left blank, the following defaults are used:

Minimum	0
Maximum	2147483647
Decimal places	0

Additional Behaviors

The tool first generates a number that within the range specified in the Minimum and Maximum fields. Next, the number is rounded based on the number of decimal places specified.

You should enter a value in the Decimal places field equal to or greater than the largest number of decimal places specified in either the Maximum or Minimum values, otherwise the tool may generate a number outside the specified range when the number is rounded.

In the following example, the tool may generate a number outside the specified range:

Type:	Number	
Name:	Number	*required
Custom column:	Number	*required
Minimum:	0.0001	
Maximum:	1	
Decimal places:	2	*required

Configuring Date/Time Generators

1. Choose the start date from the **Input source** drop-down menu.

Type:	Date/Time	
Name:	Date/Time	*required
Custom column:	Date/Time	*required
Input source:	Current date and time	
Output format:	Current date and time	*required
Output time zone:	Current date at midnight	
Output locale:	User defined	
	-- No Selection --	

The following options are available:

- **Current date and time:** Sets the start date and time to time of the test execution according to the local machine.
- **Current date at midnight:** Sets the start date and time to midnight (00:00:00) using the output time zone. If the output time zone is blank, the local machine time zone will be used.
- **User defined:** Choose this setting to configure a specific start date, format, time zone, and locale.

i The start date and format are required when User defined is enabled

If these settings are configured to use fixed values, but the fields are blank the date/time will not be generated. The following examples show different ways to specify the start date and format:

Input start date:	1999-12-31 11:59:59 pm	*required
Input format:	yyyy-MM-dd hh:mm:ss a	*required

Input start date:	31-12-1999 AD at 11:59	*required
Input format:	dd-MM-yyyy G 'at' hh:mm:ss	*required

2. Configure the output format, time zone, and locale.

Output format:	yyyy-MM-dd hh:mm:ss a	*required
Output time zone:	Pacific Standard Time (US/Pacific)	
Output locale:	English (United States) (en_US)	

3. (Optional) Configure the tool to generate a past or future date in the Offset settings.

Offsets:	
Years:	<input type="text" value="-10"/>
Months:	<input type="text" value="1"/>
Weeks:	<input type="text"/>
Days:	<input type="text"/>
Hours:	<input type="text" value="1"/>
Minutes:	<input type="text" value="10"/>
Seconds:	<input type="text" value="30"/>
Milliseconds:	<input type="text" value="1"/>

You can specify how far off the generated date and time should be from the start date and time. You can use negative numbers to use past dates.