

# Multiple Response Generator 1.1

The Multiple Responses Generator is a powerful tool that allows you to easily handle hierarchical data scenarios, one-to-many asynchronous messaging, and more.

In this section:

- [Overview](#)
- [Requirements](#)
- [Installation](#)
- [Configuration](#)
- [Usage](#)
- [Example Scenarios](#)

## Overview

For a piece of XML (`input`), the name of an XML element (`elementName`), and the URL of a backend asset (`backendURL`), the Multiple Responses Generator finds each occurrence of `elementName` in `input` and sends the occurrences separately to `backendURL`. The backend asset is a virtual asset configured to respond to messages with `elementName` as their root element and sends back responses to the Multiple Responses Generator. Once it has these responses, the Multiple Responses Generator aggregates them together into a single XML document for use in responses.

## Requirements

- Parasoft Virtualize

## Installation

1. Extract the Multiple Responses Generator JAR file from the archive downloaded from the Parasoft marketplace.
2. Open Virtualize and choose **Parasoft > Preferences > System Properties** from the main menu.
3. Click **Add JARs...** and browse to the Multiple Responses Generator JAR file
4. Click **OK** and restart Virtualize.

After performing this setup, you will be able to see the Multiple Responses Generator tool in the list of tools available to attach to a message responder.

## Configuration

The Multiple Responses Generator is controlled using two main sets of options. The options in the Required section are required for the Multiple Responses Generator to function while the options in the Optional section grant more granular control for advanced scenarios.

### Required Configuration

The following options must be configured in order for the Multiple Responses Generator to function.

<b>Name of element to send to backend asset</b>	The name of an XML element, including the namespace prefix if applicable. Each occurrence of an element in the incoming XML that matches this name will be split and sent to the virtual asset specified in Backend Asset URL. In addition, the tool will create combinations of all list elements if there are multiple lists in the request. Exclude any other lists if you want to use a single list.
<b>Backend Asset Path</b>	Specify the HTTP path on which the backend virtual asset is listening. If the full URL is specified, only the path segment will be used. The virtual asset should have a responder that handles XML messages whose root element is the element defined in <b>Name of element to send to backend asset</b> setting.

### Optional Configuration

<b>Elements excluded from splitting process</b>	A comma-separated list of element names, including the namespace prefix if applicable. By default, if the XML specified in the Name of element to send to backend asset option contains any sibling elements that share the same name (i.e., a list), then the Multiple Responses Generator will create a separate request for each repetition of the sibling elements.
<b>Response Type</b>	If the virtual asset specified in Backend Asset URL returns XML, leave this field blank. If the virtual asset specified in Backend Asset URL returns a non-XML payload, enter <code>Other</code> .

<b>Output XML Root Tag Name</b>	The name of the root tag for the Response XML output. Default is <code>responses</code> .
<b>Output XML</b>	The name of the child elements of Output XML Root Tag Name. Default is <code>response</code> .
<b>SOAPAction List</b>	For each SOAPAction specified in this comma-separated list, every occurrence of the element specified in Name of element to send to backend asset will be sent to Backend Asset URL with that SOAPAction. This is useful if each occurrence of the element requires multiple responses.

## Performance Configuration

<b>Use thread pool</b>	Enable this option (set to <code>true</code> ) to send concurrent requests to the backend virtual asset. Default is <code>false</code> .
<b>Thread pool type</b>	Specify the type of pool to use for threads that make concurrent requests to the backend virtual asset. Options are <code>FIXED</code> or <code>CACHED</code> . If the <code>FIXED</code> option is used, the max number of threads is bound by the <b>Number of threads</b> option. If the <code>CACHE</code> option is used, a new thread will be created any time a request needs to be sent and all available thread are currently busy. Default is <code>CACHED</code> .
<b>Number of threads</b>	Specify the maximum number of concurrent requests to the backend asset that can be executed at a given time. This option is applicable then the <b>Thread pool type</b> option is set to <code>FIXED</code> . Default is <code>100</code> .

## Usage

The Multiple Responses Generator can be attached to any XML output of a Parasoft Virtualize message responder. This encompasses the **Incoming Request > Payload** for any XML-based responders or the **Incoming Request > Payload Converted to XML** for any non-XML based responders. The basic procedure for using the Multiple Responses Generator is described below. A more detailed description is provided in [Example Scenarios](#).

1. Create a front end asset with one or more responders.
2. For each responder that needs the asset, attach a Multiple Responses Generator to either the **Incoming Request> Payload** output of the message responder (if the responder will handle XML input) or **Incoming Request> Payload Converted to XML** output of the message responder (if the responder will handle non-XML input).
3. Configure the Multiple Responses Generator with at least the **Name of element to send to backend asset** and **Backend Asset URL**.
4. Create a backend asset to handle the messages that the Multiple Responses Generator will send. Be sure the backend asset is deployed at the URL specified in **Backend Asset URL** in the Multiple Responses Generator.
5. Attach an XML Data Bank or Text Data Bank to the **Response XML** output of the Multiple Responses Generator and configure it to pull out the aggregated responses.
6. Use the aggregated responses in your response.

## Example Scenarios

While there are many scenarios that the Multiple Responses Generator can handle, this section provides examples for those used most frequently.

### Handling Lists with a Dynamic Number of Elements

The most common scenario handled by the Multiple Responses Generator is messages that contain lists with a dynamic number of elements. For example:

```
<root>
  <items>
    <item>
      <name>Name1</name>
      <id>1</id>
    </item>
    <item>
      <name>Name2</name>
      <id>2</id>
    </item>
  </items>
</root>
```

This request can contain N sub-requests, where  $N \geq 1$ . The response we will expect looks like this:

```

<root>
  <itemResponses>
    <itemResponse>
      <balance>10</balance>
    </itemResponse>
    <itemResponse>
      <balance>20</balance>
    </itemResponse>
  </itemResponses>
</root>

```

Using the process described in [Usage](#), the following instructions will help you set up an asset with the above requests and responses.

1. Create a frontend asset with one or more responders:
  - a. Right-click the **VirtualAssets** project and choose **Add New> Virtual Asset (.pva) File**.
  - b. Enter `Example1-Frontend` in the File Name field when prompted and click **Next**.
  - c. Choose **Empty** from the list of options on the next screen and click **Finish**.
  - d. Right-click **Responder Suite** in the newly created `Example1-Frontend.pva` and choose **Add New > Responder**.
  - e. In the window that appears, choose **Literal Message Responder** and click **Finish**.
2. For each responder that needs it, attach a Multiple Responses Generator to either the **Incoming Request > Payload** output of the message responder (if the responder will handle XML Input) or **Incoming Request > Payload Converted to XML** output of the message responder (if the responder will handle non-XML input).
  - a. Right-click the **Literal Message Responder** and choose **Add Output...**
  - b. In the window that appears, choose **Payload** in the Incoming Request section and choose **Multiple Responses Generator** under Common Tools.
  - c. Click **Finish**.
3. Configure the Multiple Responses Generator with at least the **Name of element to send to backend asset** and **Backend Asset URL**.
  - a. In the **Name of element to send to backend asset field**, enter `item`.
  - b. In the **Backend Asset URL field**, enter `http://localhost:9080/Example1-Backend`.
  - c. In the **Optional** section, enter `itemResponses`.
  - d. Create a backend asset to handle the messages that the Multiple Responses Generator will send. Be sure the backend asset is deployed at the URL specified in Backend Asset URL in the Multiple Responses Generator.



For simplicity, we will use a fixed response on the backend. However, the `Example1-Backend.pva` in the archive downloaded from the Marketplace uses a dynamic response.

4. Right-click on the **VirtualAssets** project and choose **Add New> Virtual Asset (.pva) File**.
5. Enter `Example1-Backend` in the File Name field when prompted and click **Next**.
6. Choose **Empty** from the list of options on the next screen and click **Finish**.
7. In the newly created `Example1-Backend.pva`, right-click the **Responder Suite** and choose **Add New > Responder**.
8. Choose **Literal Message Responder** in the next screen and click **Finish**.
9. Enter the following XML:

```

<itemResponse>
  <balance>10</balance>
</itemResponse>

```

10. Attach an XML Data Bank or Text Data Bank to the Response XML output of the Multiple Responses Generator and configure it to pull out the aggregated responses.
  - a. In the `Example1-Frontend.pva`, right-click **MultipleResponsesGenerator** and choose **Add output...**
  - b. Choose **Response XML** on the left-hand side and choose **XML Data Bank** on the right-hand side.
  - c. Click **Add XPath**
  - d. Click the line that appears in the Selected XPath column and click **Modify**.
  - e. Change the XPath field to `/*`
  - f. Choose **Entire Element** from the Extract dropdown menu.
  - g. Choose **Data source column** on the left-hand side of the editor and enter responses in the **Custom column name** field.
  - h. Click **OK**.
11. To use the aggregated responses in your response, double-click **Literal Message Responder** and enter the following into the text field:

```

<root>
  ${responses}
</root>

```

The following diagram describes the complete scenario:

