Copybook Builder 2.0

In this section:

Introduction

This extension provides support for COBOL copybook—enabling you to use Parasoft solutions to configure, send, validate, and virtualize copybook messages. The artifact you download from the Parasoft Marketplace is a builder executable. Run the executable to generate a JAR file that will be installed into SOAtest and/or Virtualize. *The builder jar file that you downloaded is not the copybook jar that you will install into Virtualize/ SOAtest*. The builder can be used to produce any number of copybook jar files. You can install multiple copybook jar files into Virtualize/SOAtest.

When the builder executes, it produces a consumable jar file that includes::

- An XML schema of each copybook.
- Java classes to manipulate copybook data.
- XML files to integrate copybooks as message types in Parasoft SOAtest and Virtualize.

After generating and installing the copybook JAR, you will be able to create Copybook Responders (for Virtualize) and Copybook Clients (for SOAtest)—both of which allow you to work with copybook messages in Parasoft's standard graphical message trees, as well as in "literal" (plain text) mode.

This tool supports syntactically correct copybooks and includes:

- Support for all COBOL numeric types including COMP-3 (Packed Decimal).
- Support for multiple occurrences on elementary and group level fields (OCCURS statement).
- Support for nested copy statements.

The tool does not support OCCURS DEPENDING ON statements or COPY REPLACING statements.

Requirements

- Java 6 or 7 (not Java 8)
- An appropriate JDK
- Parasoft Virtualize or Parasoft SOAtest 9.9.4 or higher

Preparing Copybook Files

The Copybook tool searches for copybook files in a list of directories provided at runtime. Within each directory, the tool searches a list of filenames and wildcards for copybook files to process.

We recommend that your copybook files have an extension that distinguishes them from COBOL source code (.cpy vs. .cob) or that you store the copybook files in separate directories from any COBOL source code.

Copybook files from Z/OS systems are limited to eight characters, but the Copybook tool does not require this.

If a copybook file contains a COPY statement, the Copybook tool resolves the name by checking for an identical file with the same extension as one provided at run time.

For example, if the copybook contains a `COPY ACCTDATA` statement and the filename list was `*.cpy,*cbl`, the copybook tool will attempt to find `ACCTDATA.cpy` or `ACCTDATA.cbl` in the list of directories provided at runtime.

Usage

SOAtest and Virtualize support various types of requests (SOAtest) and Responders (Virtualize). These types can include XML, JSON, CSV, plain text and fixed length. When the output of Parasoft Copybook Builder is integrated into these products a new Copybook Message type is created. Each copybook becomes a Message Format for the Copybook format request and responder. Perform the following steps to use the Copybook Message type:

Generating the Copybook JAR

Run the com.parasoft.soavirt.messages.copybook-<version>.jar file using the following command to build read the copybook files and generate the JAR file that you will integrate into the UI:

```
java -jar <path to executable copybook.jar> -copybookinput <path to copybook directory> -filenames <filename1,
filename2, etc.> -name <name> -apijar <path to com.parasoft.api.jar>
```

The command uses the following arguments:

-jar	Specify the Copybook extension from the Parasoft Marketplace
-copybookinput	Specify a copybook directory or a comma-separated list of copybook directories. If not a specified, the current directory will be used. The builder does not recurse sub-directories. Specify subdirectories as a separate entries and use quotation marks if the directories contain spaces or special characters.
-filenames	Specify a comma separated list of copybook files (wildcards allowed).
-name	Specifies the name of the copybook jar extension that the builder creates. This name will also be used as a UI label in Virtualize /SOAtest. If `–name` is not specified, the jar file will be named copybook.jar.
-apijar	Specify the absolute or relative path to the com.parasoft.api.jar file. This is required if you use the `-name` argument. You can access this jar file at [INSTALL]/plugins/com.parasoft.xtest.libs.web_[version]/root/com.parasoft.api.jar.
-log	Sets the log level for console messages. You can use `DEBUG`, `INFO`, `WARN`, or `ERROR`.
-preprocess	Tells the builder to stop execution after preprocessing is completed.
-help	Prints documentation.

Example

```
java -jar parasoft_copybook_builder.jar -copybookinput "C:\My Projects\proj_a\cobol_src","C:\My
Projects\proj_a\cobol_src\copy books" -filenames *.cbl,*.cpy -name Ps01CommArea -apijar com.parasoft.api.jar
```

The example searches the following folders:

- C:\My Projects\prog_a\cobol_src
- C:\My Projects\proj_a\cobol_src\copy books.

Any file that matches the wildcards *.cbl or *.cpy wil be processed as a copybook.

When the Copybook tool finishes, there will be two files created in the directory: copybook.jar and velocity.log. Assuming there are no errors, velocity.log can be deleted. Integrate the copybook.jar file as described in Integrating the Copybook Extension.

Integrating the Copybook Extension

The Copybook tool is implemented in the SOAtest and Virtualize UI as a system JAR file, which is output by executing the JAR in Generating the Copybook JAR.

1. Choose **Parasoft> Preferences**.
2. Choose **System Properties** and remove any existing copybook.jar entries.
3. Click **Add JARs** and browse to the **copybook.jar** file created after executing the JAR file in Generating the Copybook JAR.
4. Click **Open** and apply the changes
5. Restart SOAtest/Virtualize.

You can also install extensions from the command line by adding the copybook.jar file to the system.properties.classpath property in your localsettings properties file. For example:

`system.properties.classpath=<path to jar>/copybook1.jar`

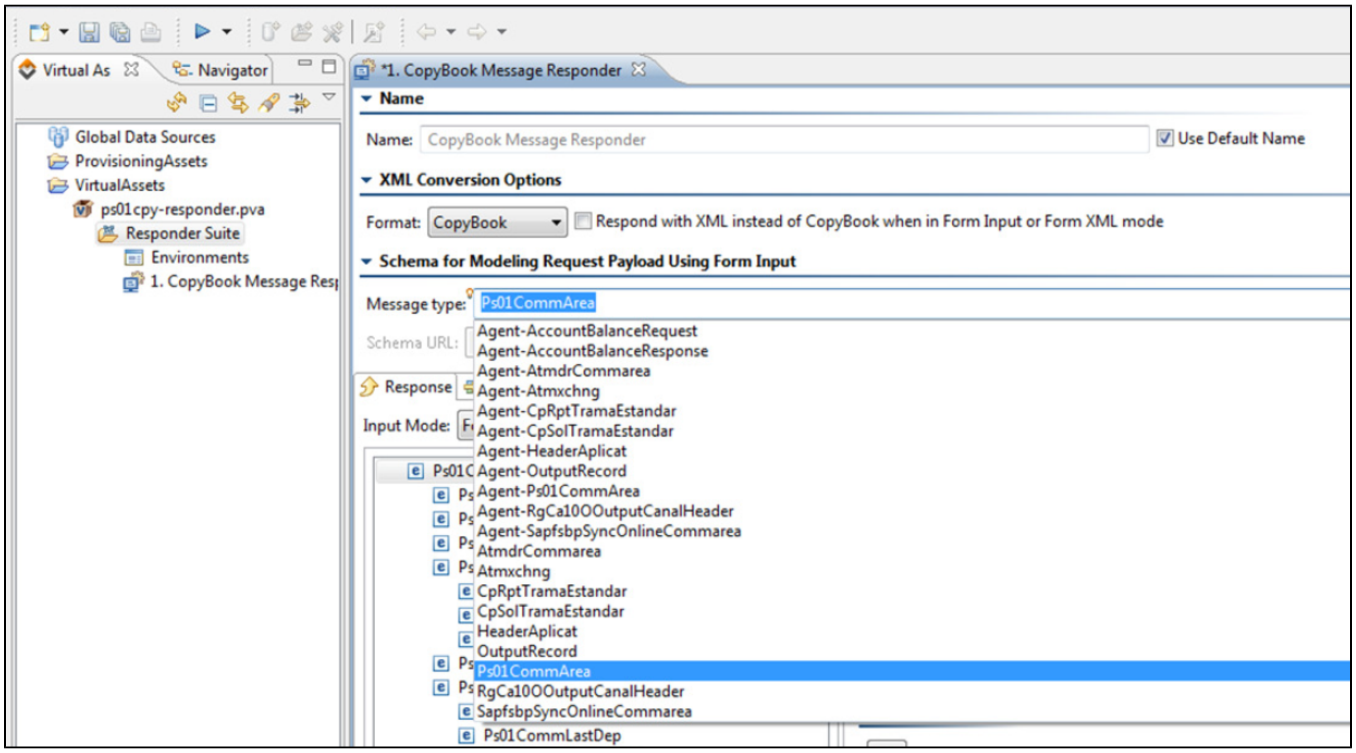
You also add the copybook.jar file to the VirtualAssets/system_jars folder (if this does not already exist, create it now). Make the following API call to reload the jar:

```
http://<virtualize_server_host>:<virtualize_server_port>/soavirt/api/<version>/preferences/systemProperties
/reload
```

Using a Copybook in a Responder

You can generate Copybook Responders and Copybook Clients from traffic, or you can add them manually. If you are adding them manually, be sure to select the desired copybook from the Message Type drop-down. Once a copybook is selected, the tree will be populated based on that copybook. This example demonstrates how to create a Virtualize responder from a copybook.

1. In the Virtualize perspective, choose **Parasoft> Show View> Virtual Asset Explorer**
2. Right-click the VirtualAssets folder and choose **Add New> Virtual Asset (.pva) File…**
3. Enter a file name and click **Next**.
4. Choose **Empty** and click **Finish**. A new .pva file is created in the VirtualAssets folder.
5. Right-click the **Responder Suite** under the new .pva and choose **Add New> Responder**.
6. Choose **CopyBook Message Responder** and click **Finish**.
7. Open the Copybook Message Responder you just created.
8. All available copybooks are presented in the Message Type drop-down menu. Choose a copybook and the fields for the copybook will be displayed as responder fields.



Name

Name: CopyBook Message Responder ✓ Use Default Name

XML Conversion Options

Format: CopyBook ☐ Respond with XML instead of CopyBook when in Form Input or Form XML mode

Schema for Modeling Request Payload Using Form Input

Message type: Ps01CommArea

Schema URL:

Agent-AccountBalanceRequest
Agent-AccountBalanceResponse
Agent-AtmdrCommarea
Agent-Atmxchng
Agent-CpRptTramaEstandar
Agent-CpSolTramaEstandar
Agent-HeaderAplicat
Agent-OutputRecord
Agent-Ps01CommArea
Agent-RgCa10OOutputCanalHeader
Agent-SapfsbpSyncOnlineCommarea
AtmdrCommarea
Atmxchng
CpRptTramaEstandar
CpSolTramaEstandar
HeaderAplicat
OutputRecord
Ps01CommArea
RgCa10OOutputCanalHeader
SapfsbpSyncOnlineCommarea
Ps01CommLastDep

You can repeat these steps to create a copybook request in your SOAtest environment.