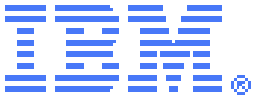

SupportPac CA0A –

CICS Channels and Containers Support Utility for WebSphere ILOG Rules for COBOL

Installation and User's Guide
Version 1.0 – October 2009

Author:
Daniel Donnelly

IBM United Kingdom Limited.
Hursley Park, Winchester, SO21 2JN. UK.

Note! Before using this information and the product it supports, be sure to read the general information under “Notices” on page 14.

First edition (October 2009)

This edition applies to Version 1.0 of SupportPac CA0A – CICS Channels and Containers Support Utility for WebSphere ILOG Rules for COBOL and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

1. Contents

1. *Contents* 3
2. *Introduction* 4
3. *SupportPac files* 5
4. *Overview* 6
5. *Preconditions and notes* 8
6. *Installing the SupportPac* 9
7. *Running the SupportPac* 10
8. *Using the generated code* 12
9. *Troubleshooting* 13
10. *Notices* 14
11. *Trademarks* 15

2. Introduction

This SupportPac provides a command line driven Java application that generates a COBOL program capable of accepting data from CICS channels and containers, and passing the data to a COBOL program generated by WebSphere ILOG Rules for COBOL. This enables a Rules for COBOL generated program to be called using an EXEC CICS LINK command. By allowing the Rules for COBOL program to be called in the way, the generated program can be run in a separate CICS system to the calling program. It may be desirable to run a Rules for COBOL program in a separate CICS system because:

- You want to centrally manage all your Rules for COBOL programs in a single CICS system (a Rules Owning Region).
- You want to use your Rules for COBOL programs in conjunction with CICSPlex System Manager Workload Management to provide best possible throughput and availability of Rules for COBOL generated programs.

This document outlines how to install and run the Java application on your workstation.

3. SupportPac files

This SupportPac contains the following artefacts:

License Directory	Directory of license information.
SupportPac CA0A.pdf	This document.
GenerateCICSWrapper.jar	JAR that contains the compiled Java application.
CallGenerateWrapper.bat	Sample batch script for calling the GenerateCICSWrapper application.
CallGenerateWrapper.sh	Sample shell script for calling the GenerateCICSWrapper application.

4. Overview

Today many business rules are locked away in application code, this has many disadvantages:

- It is not obvious what business rules are used by applications.
- A change to business rules requires a change to the application. The lifecycle of business rules can be different to the lifecycle of an application. This can mean changes to business rules are held up by the application lifecycle. This inhibits an organization's ability to respond to changing market conditions.
- Changes to business rules must be implemented by application developers. This introduces the possibility of misinterpretation of the requirements by the developers. If the business rules are externalized then it is possible for business stakeholders to make modifications to them. This removes the possibility of miscommunication between the business and IT groups.
- Multiple business rules are changed on an application lifecycle boundary. This can make it difficult to track who requested a business rule change and when. By externalizing business rules with a Business Rule Management System (BRMS) it is possible to provide an audit trail of changes down to the level of an individual business rule.

WebSphere ILOG Rules for COBOL allows you to externalize business rules locked inside your COBOL applications. It allows business rules to be authored using business terminology and then deployed into your application as a COBOL sub-program that can be called from CICS, IMS or batch environments. Used in conjunction with WebSphere ILOG Rules Team Server, business users can manage these rules without needing to understand COBOL or the underlying application infrastructure. The IT department then only have to manage the deployment of these sub-programs into production.

Currently, the sub-programs generated by Rules for COBOL must be either statically or dynamically linked with their calling program as shown in the diagram below.

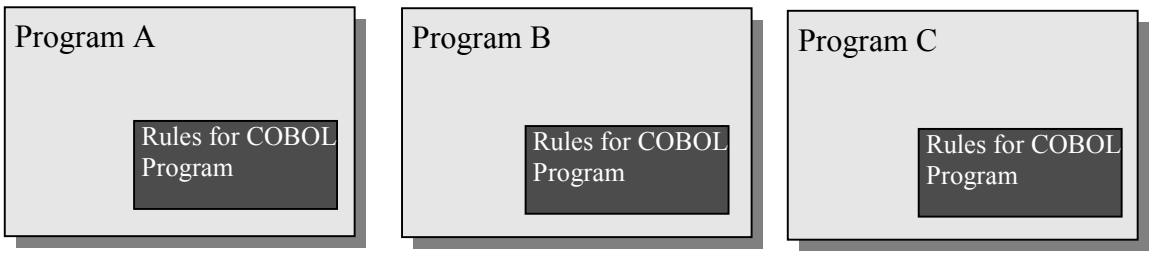

Figure 1 – Rules for COBOL generated program statically linked with its calling programs

Users who want to host their Rules for COBOL programs in a separate CICS system to their application code must manually update the Rules for COBOL generated code in order to correctly pass data from the application to the generated code. This SupportPac automatically creates a wrapper program for the Rules for COBOL code to allow users to host their code in a separate CICS system to their application code as shown below.

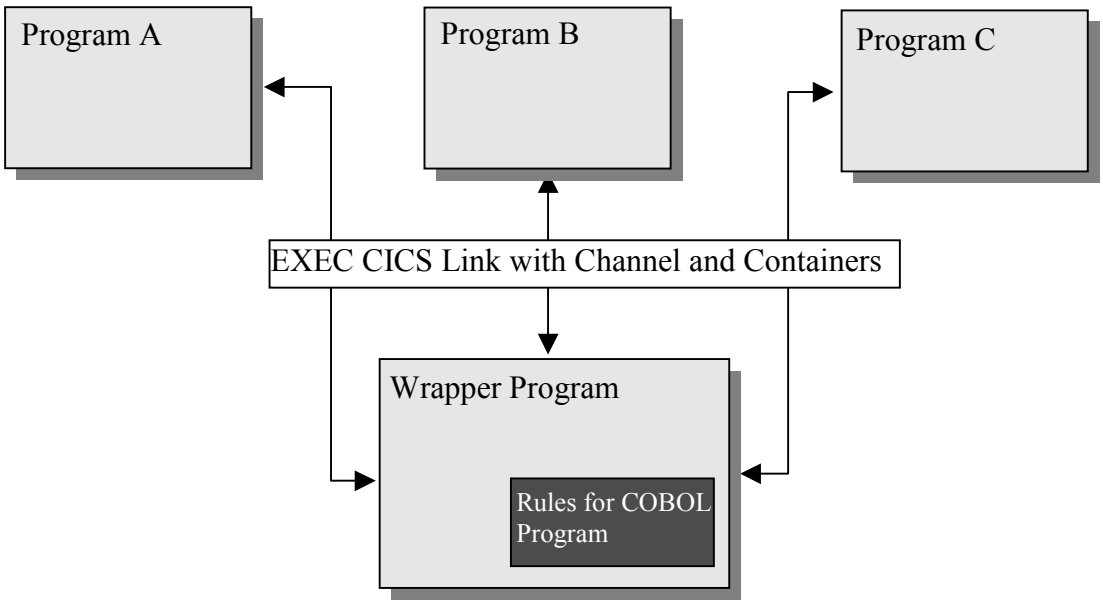

Figure 2 – Rules for COBOL generated program linked with a wrapper program

5. Preconditions and notes

The following is assumed in order to use this SupportPac:

- You have a Java 5 JRE correctly installed and set up on your workstation.
- WebSphere ILOG JRules and WebSphere ILOG Rules for COBOL 6.7.3 or higher is installed on your workstation.
- The Rules for COBOL generated file that you want to create a wrapper program for is local to your workstation.

6. Installing the SupportPac

- Extract all the files from the SupportPac archive to an appropriate folder on your workstation.
- To verify the installation perform the following:
  1. Launch a command prompt (e.g. the DOS Command Prompt in Windows).
  2. Change your directory to point at the folder you extracted the SupportPac archive to (e.g. cd C:\SupportPacCA0A).
  3. Type in the following command:

     java –jar GenerateCICSWrapper.jar

  4. If you see output similar to the following the installation has been successful:

     Incorrect number of arguments supplied.
     Usage:
     GenerateCICSWrapper r4cFile outcopyFile outputFile outcodeFile
     Where:

r4cFile	*is the name of the generated Rules for COBOL program to be analyzed.*
outcopyFile	*is the name of a new copybook that will hold the channel and container constants.*
outputFile	*is the name of a new file that will contain the CICS wrapper program.*
outcodeFile	*is the name of a new file that will hold a snippet of code to be used in the calling program.*

- If you do not see the message listed above first check that your command prompt is pointing at the directory that holds CICSWrapperFile.jar. If this does not solve the problem check that the Java JRE has been set up correctly on your system.

7. Running the SupportPac

Supplied in the SupportPac are a Windows batch file (GenerateCICSWrapper.bat) and a UNIX Shell Script (GenerateCICSWrapper.sh). It is recommended that these are used to run the SupportPac. The shell script/batch file should reside in the same directory as the GenerateCICSWrapper JAR file.

In the shell script/batch file there are four variables that must be given values to allow the SupportPac to run correctly. The variable names and there descriptions are as follows:

R4CFILE - This is the file name (including the fully qualified path name) of the file generated by Rules for COBOL that we want to create a wrapper for. Note that you should not have performed any modifications to this file after it has been generated by Rules for COBOL. If any modifications have been performed, and these modifications result in invalid COBOL, then the SupportPac may fail to generate any files.

OUTCOPYFILE – This is the file name (including the fully qualified path name) of a COBOL copybook file that will be generated by the SupportPac. This copybook will define constants that contain the names of the channel and containers that will pass data to and from a CICS wrapper program.

The SupportPac determines these names from two sources:

- The channel name is derived from the name of the copybook contained in the LINKAGE section of the Rules for COBOL generated program (R4CFILE).
- The container names are derived from data structure names on the PROCEDURE DIVISION USING line of the Rules for COBOL generated program (R4CFILE).

OUTPUTFILE – This is the file name (including the fully qualified path name) of the CICS Wrapper program that will be generated by this SupportPac. The SupportPac adds the following to the file:

1. Adds an EXEC CICS GET CONTAINER statement for each container name contained in the channels and containers copybook (OUTCOPYFILE).
2. Adds a COBOL call statement to the Rules for COBOL generated sub-program.
3. Adds an EXEC CICS PUT CONTAINER statement for each container name contained in the channels and containers copybook (OUTCOPYFILE).
4. Adds an EXEC CICS RETURN statement at the end of the code to pass control back to the calling program.

OUTCODEFILE – This is the file name (including the fully qualified path name) of a file that contains a snippet of sample code for calling the CICS wrapper program. You can use this code as a basis for modifying your application to enable it to call the Rules for COBOL code (through the wrapper program). The SupportPac adds the following to the file:

5. Adds an EXEC CICS PUT CONTAINER statement for each container that must be passed to the wrapper program.
6. Adds an EXEC CICS LINK PROGRAM statement to invoke the CICS wrapper program.
7. Adds EXEC CICS GET CONTAINER statements for each container originally passed to the calling program

Note: OUTCODE file only contains a code snippet and is not a complete COBOL program. It will not compile.

8. Using the generated code

Once the code has been successfully generated you can move it to System z. If you have Rational Developer for z you may want to use its FTP function to move the R4CFILE, OUTCOPYFILE, and OUTPUTFILE onto the mainframe (note that rather than moving outcode file to the mainframe, its contents should be used to guide changes to your application code).

If Rational Developer for z is not available then an FTP client that can reliably convert from ASCII to EBCDIC should be used to move the code to the mainframe (on Windows the native DOS FTP client will do this).

Once you have moved the code onto System z, you can start to use your standard COBOL compilers to compile and link-edit the code.

Note: The generated code does not contain any RESP or RESP2 handling.

Note: If your Rules for COBOL program causes the size of a data structure related to a container to change (for example OCCURS DEPENDING ON is used), then the code in your application program should be changed to use COBOL pointers on the EXEC CICS GET CONTAINER commands. If this is not done an ABEND may occur.

9. Troubleshooting

Below are listed the error messages returned by the SupportPac and their meanings:

- ***Error: Input file cannot be opened*** – this message indicates that the file specified by the R4CFILE parameter either does not exist or is not a valid file. Additional messages are generated stating the reason for the failure and the value of the parameter that failed.

  Check that you have correctly specified the R4CFILE value.

- ***Error: Input file is in an invalid format*** – this indicates that the contents of R4CFILE are not valid. Additional messages are generated stating the reason for the failure.

  Check that the file specified by the R4CFILE parameter is a Rules for COBOL generated file. Check that the Rules for COBOL program has a correctly formatted LINKAGE SECTION and PROCEDURE DIVISION USING section.

- ***Error: Copybook has failed to generate*** – this indicates a problem with trying to create, open, or write to a file specified by the OUTCOPYFILE parameter. Additional messages are generated stating the reason for the failure and the value of the parameter that failed.

  Check that you have correctly specified the OUTCOPYFILE parameter.

- ***Error: CICS Wrapper file has failed to generate*** – this indicates a problem with trying to create, open, or write to a file specified by the OUTPUTFILE parameter. Additional messages are generated stating the reason for the failure and the value of the parameter that failed.

  Check that you have correctly specified the OUTPUTFILE parameter.

- ***Error: Sample code file has failed to generate*** – this indicates a problem with trying to create, open, or write to a file specified by the OUTCODEFILE parameter. Additional messages are generated stating the reason for the failure and the value of the parameter that failed.

  Check that you have correctly specified the OUTCODEFILE parameter.

10. Notices

The provisions set out in the following two paragraphs do not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

Information contained and techniques described in this publication have not been submitted to any formal IBM test and are distributed on an "AS IS" basis.

The use or implementation of any information contained and/or of any technique described in this document is the user's responsibility and depends on the user's ability to evaluate and integrate the information and/or technique into the user's operational environment. While IBM has reviewed each item for accuracy in a specific situation, IBM offers no guarantee or warranty that the same or similar results will be obtained elsewhere. Users attempting to adapt any technique described in this document to their own environments do so at their own risk.

The information contained in this publication could include technical inaccuracies or typographical errors.

Changes are periodically made to the information contained herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any reference in this publication to an IBM licensed program or another IBM product is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe applicable intellectual property rights may be used instead of the referenced IBM licensed program or other IBM product.

The user is responsible for evaluating and verifying the operation of the material supplied in conjunction with this publication in conjunction with other products, except those expressly designated by IBM.

International Business Machines Corporation may have patents or pending patent applications covering subject-matter described in this document. The furnishing of this document does not give you any license to any such patent. You can send license inquiries, in writing, to:

The IBM Director of Licensing
International Business Machines Corporation
North Castle Drive

Armonk, NY 10504-1785
U.S.A.

11. Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICS	IBM	SupportPac
System z	z/OS	Rational

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a trademark of The Open Group in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.