CICS® TS custom EP adapter for Dynamic Scripting applications

CICS SupportPac CA1W - Installation and User's Guide

Version 1.0 – September 2010

Author:

Adam Rice – IBM CICS System Verification Test

Key contributor:

James O'Grady - IBM CICS System Verification Test

IBM United Kingdom Ltd.

Hursley Park. Winchester. SO21 2JN. UK

Contents

Chapter 1. Overview

This SupportPac enables the user to take advantage of the benefits offered by both the CICS Dynamic Scripting Feature Pack and event processing. These two technologies are combined by converting the data produced when an event is captured into a format that is readily consumable by a dynamic scripting application.

This allows the user to become familiar with both technologies and perhaps prototype a business solution before committing to a large scale implementation using other products such as WebSphere MQ, WebSphere Business Events, or WebSphere Business Monitor. Additional uses could include taking advantage of the fast development cycle of situational applications to quickly produce dashboards for event monitoring.

The CICS Dynamic Scripting Feature Pack provides an agile web application platform for developing and running modern web applications. You can take advantage of web technologies and improve your productivity by using CICS dynamic scripting to create, assemble, and run applications that meet your specific needs or the needs of your clients. Such applications might include dashboards and monitoring utilities.

CICS® event processing enhances business flexibility by providing a noninvasive methodology for enhancing business applications. Events are defined and controlled independently of the business logic, extending a business application without modification.

More information about event processing and the Dynamic Scripting Feature pack can be found in the Information Center.

Dynamic Scripting:

https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.smash.doc/welcome/WelcomePage.html

Event Processing:

https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.eventprocessing.doc/topics/dfhep_startHere.html

Event processing and dynamic scripting in CICS

Event processing enables detection of business situations, providing early and intelligent insight to assist you in making timely and effective business decisions. Event processing can help your business in these ways:

- Obtaining the right information at the right level of detail for the right person at the right time.
- Facilitating quick observation of exceptional business behavior and notification to the appropriate people.
- Diagnosing problems based on symptoms, and resolving them.
- Providing data for dashboard display of real-time business service availability.

Event processing can be complemented by the CICS Dynamic Scripting Feature Pack, which provides an environment for the development of situational applications, for example dashboards. Using the feature pack the user can quickly develop a situational application that can be used for monitoring events captured by the system.

This SupportPac acts as a bridge between these two technologies, allowing the user to combine them effortlessly. The data that is captured by the system is converted into the JavaScript Object Notation (JSON) format, which is then processed by a dynamic scripting application.

SupportPac CA1W

The SupportPac includes a COBOL custom EP adapter, which may need to be

compiled before use. The adapter is supplied both as a load module and as source code which you may extend if required.

Additionally, there is a sample dynamic scripting application to demonstrate the usage of the adapter. This needs to be unzipped and placed in your applications directory. This sample application contains a useful utility script for quickly creating JSON model definitions. This script is discussed in more detail later in this document.

For the purpose of the example, the dynamic scripting application runs in the same CICS region as the CICS Catalog Manager sample, however it could be running in any CICS region or as a WebSphere sMash application.

Along with the CICS Catalog Manager sample, the application produces a simple stock monitoring facility, allowing the user to quickly see the items that may need reordering and take appropriate action.

Software requirements

To use SupportPac CA1W, you must have the following versions of software and service installed:

- CICS TS for z/OS v4.1
- IBM CICS Dynamic Scripting Feature Pack v1.0
- IBM CICS Explorer v1.0.1.0

Restrictions

This SupportPac has the following restrictions:

- When you are using the COBOL2JSON utility, all copybooks referenced by the event specification copybook must be present in the directory from which the command is issued, in order for the conversion to work.

- The resource handler and the model definition in the dynamic scripting application consuming events must have the same name as the event binding responsible for the events being captured.

Chapter 2. Installing the SupportPac

Audience: experienced CICS system programmers who are responsible for installing and customizing CICS. Knowledge of downloading files from the internet, uploading files to zFS and MVS, and UNIX commands is assumed. For information about the options for the ftp command, see the *z/OS UNIX System Services Command Reference, SA22-7802.*

The following tasks guide you through downloading, installing, and configuring the SupportPac. These tasks assume that the software requirements are installed.

Downloading and copying the SupportPac files

From a workstation:

1. Download the SupportPac.

   a) Go to the CICS SupportPacs website at
      http://www.ibm.com/support/docview.wss?rs=1083&uid=swg27007241

   b) Click **View all: By Category**.

   c) Click **CA1W**.

   d) Download `CA1W.zip` by clicking the appropriate link under **Download package**.

   e) Providing you agree to the terms and conditions shown, click **I agree**.

   f) Save the file `CA1W.zip` to your workstation.

2. Extract the contents of `CA1W.zip` to a local directory. The directory contains the following files:

   ```
   CA1W.pdf              This Installation and User's Guide
   CA1W.COBOL.XMIT       SupportPac CA1W code
   CA1W.LOAD.XMIT        SupportPac CA1W load module
   CA1W.JCL.XMIT         JCL for defining CICS resources
   EPAdapter.zip         Dynamic Scripting sample application
   Licenses/Lic_*.txt    License files in various languages
   ```

3. Upload the `*.XMIT` files in binary to your z/OS system. For example, enter the following commands, where `zos.mysite.com` is your z/OS system TCP/IP host name and `INSTALL_HLQ` is the data set to be used for the SupportPac.

   ```
   ftp zos.mysite.com
   ftp> quote site lrecl=80 recfm=fb
   ftp> binary
   ftp> put CA1W.COBOL.XMIT 'INSTALL_HLQ.CA1W.COBOL.XMIT'
   ftp> put CA1W.LOAD.XMIT 'INSTALL_HLQ.CA1W.LOAD.XMIT'
   ftp> put CA1W.JCL.XMIT 'INSTALL_HLQ.CA1W.JCL.XMIT'
   ```

4. Using the same ftp session, upload the sample dynamic scripting application to your z/OS system. For example, enter the following commands where `apps_dir` is the directory for your dynamic scripting applications that you created during installation of the feature pack.

   ```
   ftp> cd apps_dir
   ftp> put EPAdapter.zip
   ftp> quit
   ```

5. Extract the dynamic scripting application into your dynamic scripting applications directory, `apps_dir`.

   a) From a z/OS Unix shell extract the `EPAdapter.zip` file; for example enter:
      ```
      cd apps_dir
      jar -xvf EPAdapter.zip
      ```

b) Ensure that the application directory and files have the correct permissions.

The z/OS UNIX user ID that the CICS® region uses must have write access to the directory and files. If your z/OS UNIX user ID is in a common group with the CICS region user ID, you can use the following command to make the full directory tree writable, where *samp_dir* is the directory of the sample dynamic scripting application supplied with the SupportPac:

```
chmod -R g+w /samp_dir
```

c) Check that the configuration of the application is correct. Open the following file for editing:

samp_dir/config/`zero.config`

Change the port that the application will run on as appropriate.

d) The sample application may require the following command to be run in order to ensure that you have the most recent modules from the repository. From a z/OS Unix shell issue the following command:

```
zero update
```

6. From a TSO session use the **RECEIVE** command to receive the XMIT files that you transferred earlier in this section. For example, use the following command:

a) Enter the **RECEIVE** command; for example:

```
RECEIVE INDATASET('INSTALL_HLQ.CA1W.JCL.XMIT')
```

b) When prompted for restore parameters enter the following where `INSTALL_HLQ` is the data set used for the SupportPac:

```
DATASET('INSTALL_HLQ.CA1W.JCL')
```

c) Repeat for the remaining two XMIT files.

Adding resources in the CICS CSD

The SupportPac must have resources installed in the CICS region so that it can function correctly. These resources are supplied in a group called CA1W.

To create group CA1W update and submit the JCL job `INSTALL_HLQ.CA1W.JCL(CSDDEF)`:

1. Change all instances of *<CA1WINST>* to the SupportPac `INSTALL_HLQ`, where *INSTALL_HLQ* is the data set to be used for the SupportPac.

2. Change *<YOUR_CSD>* to the data set name of your CICS CSD.

3. Review the JCL job card and STEPLIB statements and change them as appropriate.

4. Save your changes and submit the job. The job should complete with a return code of 4. Warning message DFH5510 is issued, because the definition of transaction CA1W starts with the letter C.

The resources that you have just defined must be installed.

1. From within the IBM CICS Explorer use the following menu item:

   Administration > Resource Group Definitions

2. From the Resource Group Definitions view right-click the CA1W group.

3. Click **Install...**

4. Select the target region and click **OK**.

Chapter 3. Getting started

The following sections in this chapter show you the steps required to set up the infrastructure so that your system generates events, and the events are displayed in a dynamic scripting application.

Setting up the Catalog Manager Sample

SupportPac CA1W uses the CICS Catalog Manager example application to demonstrate its usage. Instructions on setting up this application can be found in the Information Center:

http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.exampleapplication.doc/topics/dfhxa_t230.html

After the example application has been installed, verify that it is working correctly. Instructions on how to use the example application can be found here:

http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.exampleapplication.doc/topics/dfhxa_t330.html

Installing a URIMAP CICS resource

The URL of the target dynamic scripting application is specified to the EP adapter via a URIMAP resource in CICS. The same dynamic scripting application can consume events captured from multiple capture specifications, so only one URIMAP resource might be required for each dynamic scripting application.

1. Define the resource. This resource can be defined using the CEDA transaction as follows:

```
CEDA DEF URIMAP(CATAPP) G(CA1WEXMP) USAGE(CLIENT)
PORT(PORT) HOST(zos.mysite.com) PATH(/)
```

Where *<PORT>* is to be substituted for the port number on which the example dynamic scripting application is running and *<zos.mysite.com>* is your z/OS system TCP/IP host name.

2. Install the URIMAP resource. The resource can be installed using the CEDA transaction as follows:

```
CEDA INSTALL G(CA1WEXMP) URIMAP(CATAPP)
```

Installing the sample event binding

To demonstrate the usage of SupportPac CA1W, the sample event binding that can be found in the CICS Explorer is required. The following instructions describe how to install the event binding.

1. Create a new CICS bundle project.

   a) Within the IBM CICS Explorer, from the Resources perspective, use the following menu selection:

   Explorer > New Wizards > CICS Bundle project

   b) Provide a name for the project and click **Finish**.

2. Create an event binding for the catalog sample.
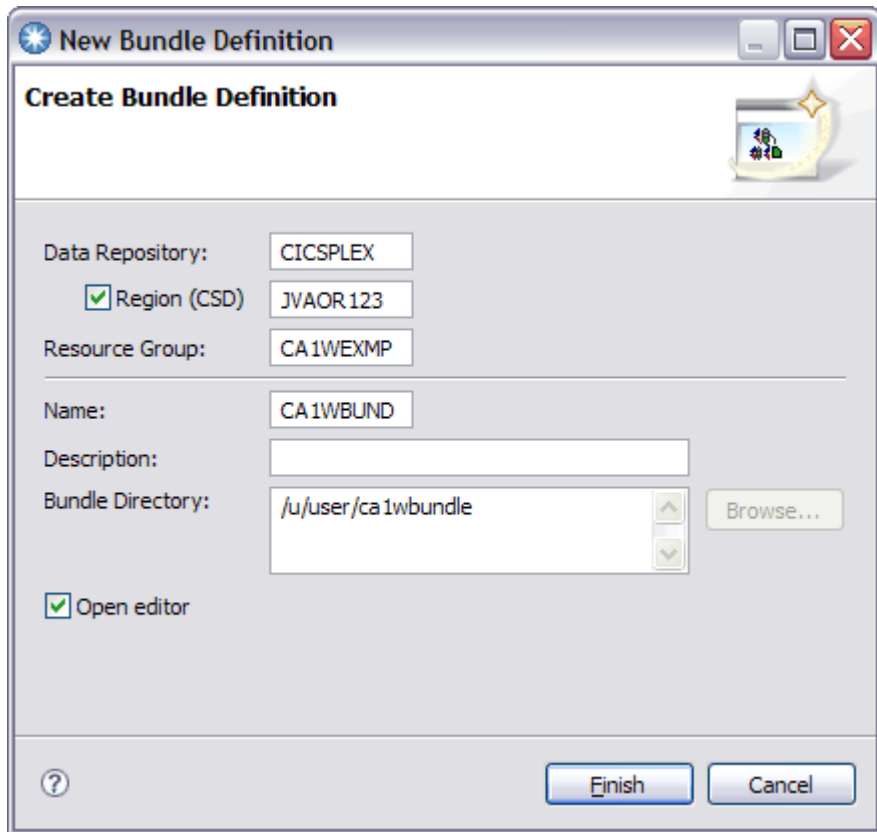
   a) Use the following menu selection:

   Explorer > New Wizards > Catalog Sample Event Binding

   b) Call the event binding **ca1wbind**.

   c) Select the CICS Bundle project created in step 1 as the parent folder.

   d) Click **Finish**.

3. Edit the event binding.

   a) Expand the CICS Bundle project that contains the new event binding and double-click the event binding file to open it in an editor.

   b) Select the Adapter tab.

   c) Click the **Export Event Specifications...** button.

   d) Select a destination directory on your workstation and click **Export**. Make a note of this location as the exported copybook will be needed later.

   e) From the Adapter tab, make the following changes:

      i. From the drop-down list set the Adapter to **Custom (User Written)**.

      ii. Set the Transaction ID to CA1W.

      iii. In the text area labeled **Data passed to the Custom Adapter** enter the name of the URIMAP resource that is to be used to provide the URL of the dynamic scripting application. In this case enter CATAPP.

      iv. In the Advanced section of the Adapter tab, select the **Use Context User ID** check box.

   f) Save the changes you have made.

4. Export the bundle to z/OS Unix.

   a) Right-click the CICS Bundle project and select **Export to z/OS UNIX File System...**

   b) Select a destination folder for the Bundle project. Make a note of this location as it will be needed in the next step.

5. Create a bundle definition in CICS.

   a) Switch to the CICS SM perspective.

   b) Use the following menu selection:

      Explorer > New Wizards > Other...

   c) In the filter text box type bundle.

   d) Select **Bundle Definition** and click **Next**.

   e) Complete the fields as in the diagram below, substituting the correct values for your environment, then click **Finish**.

f) To show the bundle definitions in your CICS regions use the following menu selection:

Administration > Bundle Definitions

6. Install the bundle definition.

   a) Right-click on the bundle previously defined and select **Install...**.

   b) Select the target region and click **OK**.

7. Verify the bundle installation.

   a) Open the Bundles view by using the following menu selection:

Operations > Bundles

   b) Verify that the new bundle is installed and enabled.

Configuring the dynamic scripting application

The sample dynamic scripting application provided with this SupportPac utilizes the ZRM functionality of the Dynamic Scripting Feature Pack. This means that the application uses a simple resource handler containing the line `ZRM.delegate()`

to handle all of the RESTful requests. However, content of the events must be defined to the application.

This is achieved by creating a model definition. Model definitions are JSON files that reside in the `app/models` folder of the application. If there is a lot of data being produced in the events, creating a model definition manually can be tedious and prone to error. To prevent these issues the sample application contains a script that automates this process for you.

These next steps show you how to quickly create a model definition from the COBOL copybook that you created from the event specification in the previous section.

1. Rename the copybook to match the event binding name. In this example the event binding is called **ca1wbind**.

2. Open the copybook in an editor and remove or comment out the following lines, then save the changes:

   ```
        05 ContextData.
          COPY DFHEPFEO.
   ```

3. Upload the copybook file to your z/OS system. For example, enter the following commands, where `zos.mysite.com` is your z/OS system TCP/IP host name and `apps_dir` is your dynamic scripting applications directory.

   ```
   ftp zos.mysite.com
   ftp> cd apps_dir/EPAdapter
   ftp> put ca1wbind.cpy
   ftp> quit
   ```

4. Generate the JSON model definition from the COBOL copybook.

   a) Navigate to the `apps_dir/EPAdapter` directory.

   b) Run the COBOL2JSON script against the copybook. For example:

      ```
      zero COBOL2JSON ca1wbind.cpy
      ```

   c) Check the output in the `app/models` folder of the application. There will be a JSON model definition describing the data that was in the copybook.

5. Build the Derby database tables using the new model definition:

   ```
   zero model sync
   ```

6. Ensure that your CICS region is up and running. Then start the application using the following command:

   ```
   zero start
   ```

Running the sample

This section describes the steps required to have the system generate an event and then view it on the web page provided by the sample dynamic scripting application.

1. From a CICS 3270 terminal, run transaction EGUI. This runs the catalog manager example application.

2. Select option 1 to list the items.

   To trigger an event an item must be purchased that has the following attributes:

   - Its new stock level is less than 24

- Its on order value is 0

A suitable item to order would be item 10, 'Ball Pens Black 24pk'.

3. Place a '/' next to the item and press Enter.

4. Input the required data and press Enter.

5. View the generated event on the dashboard web page in the dynamic scripting application.

One of the advantages of using ZRM is that it automatically appends a time stamp to the record so that the user can see when the event was created, which may be useful in circumstances where events are time sensitive.

Troubleshooting

If you experience issues when using the SupportPac, the following list of items should be checked:

- Ensure that the custom adapter program and transactions are correctly defined in CICS.

- Check that the URIMAP resource has the correct host name and port for the target dynamic scripting application.

- Check that the JSON model in the `/app/models` folder of the dynamic scripting application correctly represents the content of the data being received.

- Check that both the model definition and resource handler in the dynamic scripting application have the same name as the event binding.

- Ensure that the event binding was configured correctly prior to deployment. Check that the adapter was set to use the CA1W transaction and that the name of the URIMAP resource was included.

- Depending on the security settings of your region, using transaction name CA1W may cause security violations. If this is the case define a new transaction for the EP2JSON program and update the event binding to use the new transaction.

Feedback

What do you think of this SupportPac? We welcome your comments by email to suppacs@uk.ibm.com.

Chapter 4. References

- IBM CICS Transaction Server
  ibm.com/software/htp/cics/tserver/v41/

- Project Zero, the community website for WebSphere sMash.
  http://www.projectzero.org

- Documentation for JavaScript Object Notation (JSON)
  http://http://www.json.org/

- CICS Dynamic Scripting Feature Pack download and information
  http://www-01.ibm.com/software/htp/cics/scripting/

- CICS Dynamic Scripting Feature Pack Information Center
  http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.smash.doc/welcome/WelcomePage.html

Chapter 5. 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.

License

The international license agreement for the SupportPac is available as a set of language-specific text files in the directory `Licenses` in the `CA1W.zip` file. The SupportPac is provided "as-is" and does not include defect correction.

Trademarks

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

CICS	MVS	WebSphere
IBM	Tivoli	z/OS
DB2	System z	

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.