

CICS Transaction Server for z/OS



# IBM CICS TS V3 and V4.1: CICS Events for WBE SupportPac CB11



CICS Transaction Server for z/OS



# IBM CICS TS V3 and V4.1: CICS Events for WBE SupportPac CB11

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 45.

This edition applies to Version 1.1 of IBM CICS TS V3 and V4.1: CICS Events for WBE SupportPac CB11 and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2009.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface</b> . . . . .	<b>v</b>	Creating WebSphere Business Events definitions for CICS events . . . . .	22
What this book is about . . . . .	v	Adding touchpoint and intermediate object definitions . . . . .	23
What you need to know to understand this book . . . . .	v	Defining a message queue connector . . . . .	25
<b>Chapter 1. CICS event processing for WBE</b> . . . . .	<b>1</b>	Defining actions . . . . .	26
<b>Chapter 2. What this SupportPac contains</b> . . . . .	<b>3</b>	Defining a file system connector . . . . .	28
<b>Chapter 3. Setting up the SupportPac files</b> . . . . .	<b>5</b>	Creating a filter . . . . .	29
<b>Chapter 4. What is event processing?</b> . . . . .	<b>7</b>	Creating an interaction set . . . . .	30
<b>Chapter 5. CICS TS V4.1 Event Processing support.</b> . . . . .	<b>9</b>	<b>Chapter 8. Errors returned by the SupportPac program.</b> . . . . .	<b>33</b>
Advantages of CICS TS V4.1 Event support . . . . .	9	<b>Chapter 9. Abends issued by the SupportPac program.</b> . . . . .	<b>35</b>
Detecting the version of SupportPac CB11 being used . . . . .	10	<b>Chapter 10. Diagnosing problems with the SupportPac program</b> . . . . .	<b>37</b>
Recommendations for emitting events on CICS TS V4.1 . . . . .	10	<b>Chapter 11. Creating CICSplex SM definitions using BATCHREP</b> . . . . .	<b>39</b>
<b>Chapter 6. CICS events with WebSphere Business Events</b> . . . . .	<b>11</b>	<b>Chapter 12. Migrating to V4.1 event support</b> . . . . .	<b>41</b>
A banking scenario . . . . .	12	Identifying the events to be migrated . . . . .	41
<b>Chapter 7. Using the SupportPac</b> . . . . .	<b>15</b>	Migrating events to use event specifications . . . . .	41
Adding events to your CICS application . . . . .	15	Specifying an EP adapter for migrated events . . . . .	42
The sample applications with CICS events . . . . .	19	An example of replacing EXEC CICS LINK by EXEC CICS SIGNAL EVENT . . . . .	42
Configuring your system to send events to WebSphere Business Events . . . . .	19	<b>Notices</b> . . . . .	<b>45</b>
		Trademarks . . . . .	46



---

## **Preface**

---

### **What this book is about**

This book provides a description of CICS acting as a source of events to WebSphere® Business Events.

---

### **What you need to know to understand this book**

This book assumes that you are familiar with CICS®, either as a system administrator or as a system or application programmer.





---

## Chapter 1. CICS event processing for WBE

You can use this SupportPac to modify your CICS applications to act as a source of events to WebSphere Business Events. This SupportPac is for use with CICS TS Version 3.1, CICS TS Version 3.2, and CICS TS Version 4.1 for migration purposes.

Use of this SupportPac on CICS TS V4.1 is for migration, where the SupportPac has previously been used on CICS TS version 3. For new events on CICS TS V4.1, you are recommended to use the more advanced event support provided by CICS TS V4.1. See Chapter 5, "CICS TS V4.1 Event Processing support," on page 9 for more details.

You can add events to existing CICS application code, by making a small change to link to the program supplied by the SupportPac, passing event details in a channel. The SupportPac program will format the event information into XML that conforms to the WebSphere Business Events format. The program then emits the event for consumption by WebSphere Business Events using the WebSphere MQ transport. WebSphere Business Events can include the event in interaction blocks to carry out pattern matching and other event processing.

### Prerequisites

To use this SupportPac, you must meet the following system requirements:

- WebSphere MQ for z/OS, Version 6.0 and above
- CICS TS Version 3.1, CICS TS Version 3.2, or CICS TS Version 4.1
- WebSphere Business Events version 6.1.0.1 and above

**Note:** This SupportPac can be used with WebSphere Business Events version 6.1.0.1, or with WebSphere Business Events version 6.2. Information on each of the versions can be found in these online information centers.

- WebSphere Business Events version 6.1 information center
- WebSphere Business Events version 6.2 information center



---

## Chapter 2. What this SupportPac contains

This SupportPac contains the zipped file `cb11.zip`, which includes all of the required files.

The following files are included in the SupportPac zip file:

**copybooks/eps0conf.copy**

A COBOL copybook for the configuration structure passed to the SupportPac program.

**copybooks/eps0err.copy**

A COBOL copybook for the error structure returned by the SupportPac program.

**copybooks/epsconfp.copy**

A PL/I copybook for the configuration structure passed to the SupportPac program.

**copybooks/epserrp.copy**

A PL/I copybook for the error structure returned by the SupportPac program.

**copybooks/epsconfh.h**

A C/C++ header file for the configuration structure passed to the SupportPac program.

**copybooks/epserrh.h**

A C/C++ header file for the error structure returned by the SupportPac program.

**copybooks/eps\$conf.copy**

An assembler language copybook for the configuration structure passed to the SupportPac program.

**copybooks/eps\$err.copy**

An assembler language copybook for the error structure returned by the SupportPac program.

**documentation/cb11.pdf**

SupportPac documentation in PDF format.

**executable/cb11.unload**

The SupportPac program executable in an unpacked MVS load library.

**executable/index.html**

Documentation for the helper classes `EPSEvent` and `EPSEventException`.

**executable/cicseps/cicseps.jar**

Classes that define the exceptions, configuration structure, and error structures, and that build containers to pass to the SupportPac.

**Licenses**

This directory contains your license to use this SupportPac. There are twelve translations of the license.

**samples/epsdefs.jcl**

DFHCSDUP JCL to add the program definition that is required for the

SupportPac program to the CICS CSD. JCL for a sample application that calls the SupportPac program and a transaction to run this sample program.

**samples/eps0link.cobol**

A COBOL sample program to illustrate how to use the SupportPac.

**samples/EPsJLink.java**

A Java sample program to illustrate how to use the SupportPac.

---

## Chapter 3. Setting up the SupportPac files

You must set up the files provided with the SupportPac to use CICS as a source of events to WebSphere Business Events.

1. Unzip the cb11.zip file to a suitable directory on your workstation.
2. Copy the samples and copybooks that you want to use to the development system that you use to edit and build your applications.

If you want to call the SupportPac program from a Java program, copy the javadoc and cicseps.jar file to an appropriate location. Chapter 7, "Using the SupportPac," on page 15 explains how to use the samples, copybooks and Java classes.

3. Use the File Transfer Program (FTP) to transfer the epsdefs.jcl file into a suitable JCL library from where you can edit and submit the JCL.
4. Re-create the MVS load library containing the SupportPac program EPSEMIT from cb11.unload by performing the following steps:
  - a. Use the File Transfer Program (FTP) to transfer the file to the z/OS UNIX system where your CICS system runs. Transfer the file as a z/OS sequential file, by issuing the following commands:

```
cd //userid
binary
quote site blksize=3120 lrecl=80 recfm=FB
put cb11.unload
```

where *userid* is replaced with the z/OS user ID under which you log on to the system.

- b. Use the TSO RECEIVE command to reload the cb11.unload file:  
RECEIVE INDATASET(CB11.UNLOAD)

By entering this command, you re-create the *userid*.CB11.LOAD data set that contains the EPSEMIT load module.

5. Create the CICS system definitions to use the SupportPac.
  - a. Define the data set containing the EPSEMIT load module to CICS. The data set is *userid*.CB11.LOAD or another data set you copied this to. You can define this data set either in a dynamic LIBRARY resource or by adding the data set to the DFHRPL concatenation. If you use a dynamic LIBRARY, you can add the SupportPac program to your system without a CICS restart.
  - b. Create a program definition for the EPSEMIT program, either by editing and submitting the supplied epsdefs.jcl or by creating a program definition in the CSD or CICSplex SM repository. See Chapter 11, "Creating CICSplex SM definitions using BATCHREP," on page 39 for BATCHREP input to do this.

You must define EPSEMIT with an execution key of CICS, so you must create a program definition with **EXECKEY(CICS)** and install this definition. Do not use program autoinstall for EPSEMIT, or define it with **EXECKEY(USER)**. If you do, EPSEMIT will abend.

The sample JCL file epsdefs.jcl shows the correct program definition.

- c. Install the program definition for EPSEMIT in the CICS systems where you will be using the SupportPac.

## Results

Your CICS system is now correctly set up to use the SupportPac. See Chapter 7, "Using the SupportPac," on page 15.

---

## Chapter 4. What is event processing?

An event is anything that happens (or does not happen) that is significant to a system. A business event is something that happens (or does not happen) that is significant to your business. An event has a name and typically some data or payload. Event processing is the capture, enrichment, formatting, and emission of business events, the subsequent routing and any further processing of emitted events (sometimes in combination with other events), for consumption by event consumers. Business event processing is an extension to complex event processing to provide the business context.

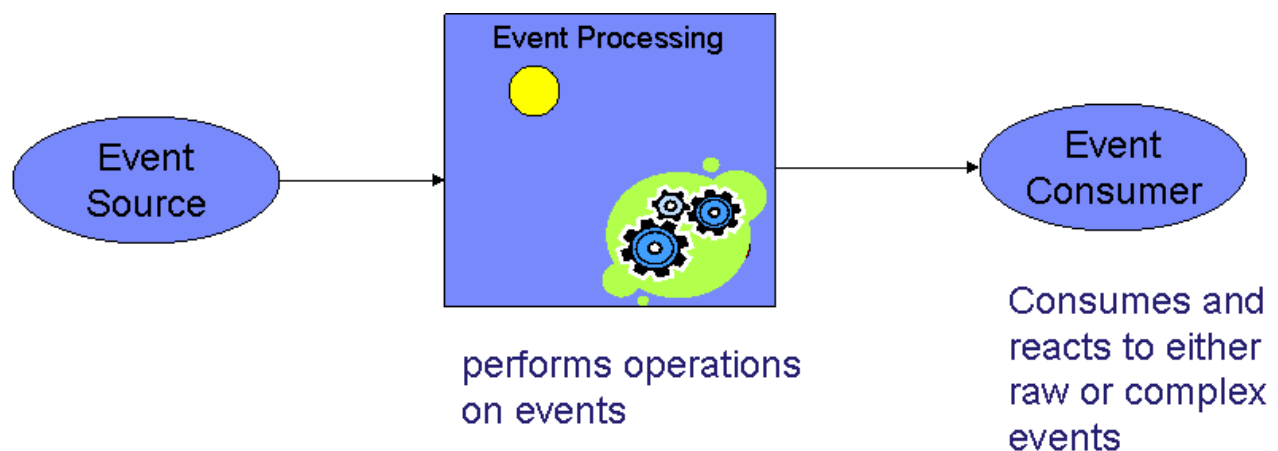


Figure 1. Components of an event processing system

As shown in Figure 1, an event processing system has three main components.

### Event producer or Event source

Emits events into the event processing system. Examples of event sources are simple Radio Frequency Identification (RFID) sensors and actuators, business processes, and CICS applications.

### Event processor

Can perform a variety of actions on events:

- Simple enriching of the event; for example, adding a time stamp to the event data.
- Adding information about the source of the event.
- Processing multiple simple events, from multiple event sources, against event patterns to produce a new derived event. Processing of this kind is often referred to as complex event processing. Here are some examples of complex events:
  - Three orders from customer A in 2 days
  - A bank withdrawal after a PIN change update
  - An interesting pattern of stock trades

### Event consumer

Reacts to the event. The event consumer might be very simple and just

update a database or a visual dashboard with the data carried with the event, or it might carry out new business processing based on the event.



---

## Chapter 5. CICS TS V4.1 Event Processing support

In the Announcement Letter for WebSphere Business Events version 6.2.0 and WebSphere Business Event eXtreme Scale version 6.2.0, in which the CICS Events for WBE SupportPac was announced, IBM issued a Statement of Direction around future enhanced capability for CICS and events.

This Statement of Direction included an invitation to Licensed CICS TS users to learn more about this capability by applying to participate in the CICS Beta program. The Statement of Direction in full is:

IBM intends, by year-end 2009:

- To integrate into CICS TS and extend the eventing capability being delivered in the new CICS TS V3 SupportPac.
- To provide new capability to enable business analysts and developers to instrument CICS business logic with WebSphere Business Events without changing the existing CICS applications.

Licensed CICS TS users wishing to gain an early insight to this intended capability may apply to participate in the CICS Beta program by sending an email to [CICSEP@uk.ibm.com](mailto:CICSEP@uk.ibm.com). With this statement of direction, all licensed users of CICS TS can plan to enable their CICS systems fully to participate in an event-enabled SOA, confident that the CICS portfolio of products continues to be well-aligned with SOA evolution.

**Note:** CICS TS and WebSphere Business Events are separately ordered and entitled products.

All statements regarding IBM plans, directions, and intent are subject to change or withdrawal without notice.

This Statement of Direction describes new capability introduced in CICS TS V4.1 for non-invasive support for emitting events from CICS applications to consumers including WebSphere Business Events. This extends and enhances the capability provided by the SupportPac.

Version 1.1 of the CICS Events for WebSphere Business Events SupportPac has been updated to enable the SupportPac to run on CICS TS V4.1, and to make it easier for events emitted using the SupportPac to move to the new capability provided by CICS TS V4.1, which is the recommended approach when running on that release.

**Note:** Users of the SupportPac should move to version 1.1 as soon as possible, and are strongly recommended to turn ON the new compatibility option passed in the configuration container.

---

### Advantages of CICS TS V4.1 Event support

There are significant advantages offered by the support for events in CICS TS V4.1 when compared with the capabilities of SupportPac CB11.

- Support for non-invasive events. Events can be emitted from CICS applications without the need to change or recompile the application code. This support

makes use of event bindings, in which events are specified. The event binding also contains information about how the specified events can be detected within the application by the CICS runtime.

- If greater control over the events than that provided by the non-invasive support is required, you can code a new EXEC CICS SIGNAL EVENT command in an application to 'event-enable' the application. After adding the command, some details of the event can be changed without requiring further changes to the application.
- Events can be dynamically enabled and disabled, by enabling or disabling the event bindings.
- There is reduced impact to the emitting application compared to the CB11 support. The overhead incurred when using CB11 is moved away from the application thread of execution.
- It is possible to emit events to consumers other than WebSphere Business Events:
  - WebSphere Business Monitor
  - Any other consumer that recognizes the Common Base Event (CBE) format
  - A program which reads the event from WebSphere MQ
  - A CICS transaction
  - A CICS TS Queue
- Support for a wider range of data types in the event data.
- Escaping of any non-valid XML characters in the event data.
- Support for different code pages.

---

## Detecting the version of SupportPac CB11 being used

The user trace entry '0013'X gives the version of the SupportPac program being used, from version 1.1 onwards.

### About this task

Examine the user trace to determine the version of the SupportPac being used.

For the initial version, 1.0, there is no user trace entry indicating a version.

All users of SupportPac CB11 are advised to install the latest version of the program. This version provides a new compatibility option in the configuration container, which when specified imposes some tighter restrictions on the names of the event and its data items, See "Adding events to your CICS application" on page 15 for more details.

Using the new compatibility option will ease migration to the CICS TS V4.1 event support.

---

## Recommendations for emitting events on CICS TS V4.1

If you are running on CICS TS V4.1, any new events should be emitted using the CICS TS V4.1 event specifications mechanism. You can do this even in programs which also use SupportPac CB11 to emit existing events.

You are recommended to migrate events which are currently emitted using SupportPac CB11 to use the CICS TS V4.1 approach, either using the non-invasive mechanism, or by adding EXEC CICS SIGNAL EVENT commands. Chapter 12, "Migrating to V4.1 event support," on page 41, gives some guidance on doing this.

---

## Chapter 6. CICS events with WebSphere Business Events

Using this SupportPac, you can emit events from a CICS application for processing by WebSphere Business Events. The SupportPac will format your supplied event information into the XML format required by WebSphere Business Events, and emit the event for consumption by WebSphere Business Events, or another consumer, using the WebSphere MQ transport.

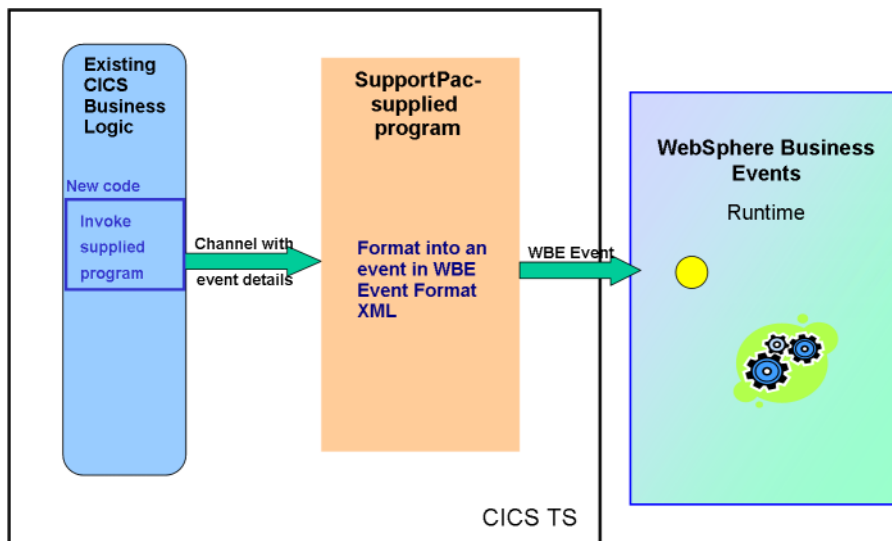


Figure 2. Using the SupportPac

The elements of the SupportPac are shown in Figure 2.

- You make small changes to existing CICS application code, to call the program supplied by the SupportPac, passing event details in a channel.
- The SupportPac program formats the event information into XML format required by WebSphere Business Events, and emits the event for consumption by WebSphere Business Events, or another consumer.
- The SupportPac program produces events with the following characteristics:
  - A time stamp of the time that the event was produced in the event data.
  - The fully qualified applid of the emitting CICS system in the event data.
  - The event has a connector name of "CICS WBE Events".
  - The event uses the supplied event name as the connector-bundle name.
  - The event contains one connector-object, which includes a set of fields, one for each of the data items with the specified data item name and data value. The connector-object has a name of the event name followed by the string Data. The way in which the string Data is appended depends on whether the compatibility option is ON. For example, if the event name is OrderEvent, then with compatibility ON the connector-object name is OrderEvent\_Data. If compatibility is not ON then the connector-object name is OrderEvent Data.
- Websphere MQ is used to send the event from the system on which your CICS system is running to the system running WebSphere Business Events.

- You can then use WebSphere Business Events to include the event in interaction blocks to carry out event processing, such as pattern matching.

## A banking scenario

This scenario demonstrates how the SupportPac might be used in a typical business application. The scenario shows an interaction defined using the WebSphere Business Events user interfaces, to watch for a suspicious activity in a customer bank account.

To understand the scenario, you must be familiar with WebSphere Business Events concepts and tools. The documentation for WebSphere Business Events can be found in the WebSphere Business Events information center.

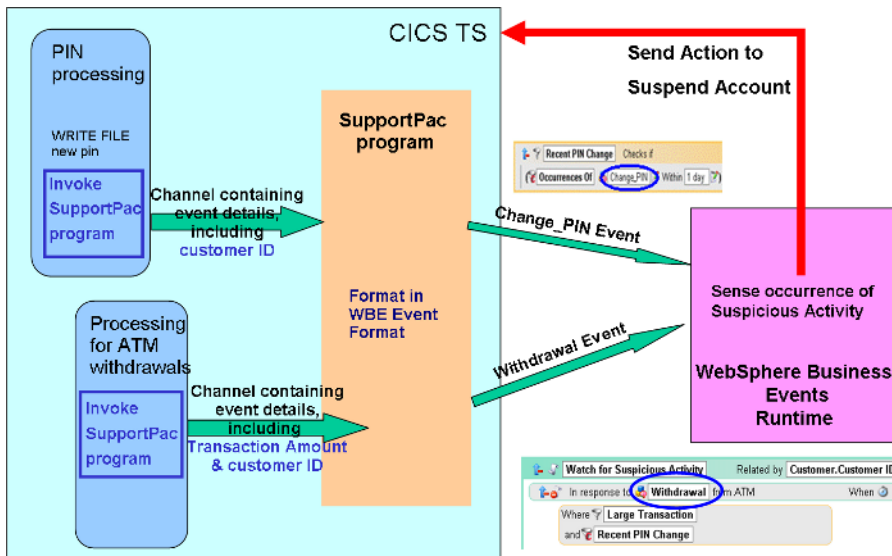


Figure 3. A banking scenario

The diagram of the scenario in Figure 3 shows processing of a withdrawal from a retail customer bank account. The activity in the system with the SupportPac installed is as follows:

1. CICS processes personal identification number (PIN) input and withdrawals from automatic teller machines (ATMs).
2. The application developer has added code to call the SupportPac program to the PIN processing transaction at the point where the program writes a new PIN to a CICS file. The event payload contains data about the customer making the PIN change. CICS Events emitted by the SupportPac program automatically include a timestamp of the date and time at which the event occurred.
3. The SupportPac program emits a **Change\_PIN** event to WebSphere Business Events. WebSphere Business Events remembers this event.
4. The application developer has added code to call the SupportPac program to the transaction that processes ATM withdrawals at the point where the program carries out withdrawal processing. The event payload contains data about the withdrawal transaction and the customer making the withdrawal.
5. The SupportPac emits a **Withdrawal** event to WebSphere Business Events.
6. The **Withdrawal** event causes WebSphere Business Events to check if this withdrawal is for a large amount. If it is, the interaction block defined in the WebSphere Business Events user interface results in a check for any recent PIN

change made by this customer. If a PIN change is found and it matches the condition of being a recent PIN change as defined in the WebSphere Business Events user interface, an occurrence of the suspicious activity is triggered.

7. The first action associated with a suspicious activity is to suspend the customer's account pending further investigation. A request is made to CICS to carry out this action; for example, by issuing a Web services request to CICS.



---

## Chapter 7. Using the SupportPac

To use the SupportPac, you add code to emit events to a CICS application, which must link to the EPSEMIT SupportPac program. You configure your system to send events to WebSphere Business Events using WebSphere MQ, and you define CICS events using the WebSphere Business Events user interfaces.

---

### Adding events to your CICS application

To emit events from your CICS application, you add code at each point at which an event is to be created. This code populates a channel with a set of containers that describe the event and the destination of the event, and links to the SupportPac program, EPSEMIT.

#### About this task

This task is illustrated by using code from the COBOL sample that is shipped with the SupportPac, EPS0LINK, which creates an event in which three items of event data are passed to EPSEMIT.

1. Determine whether to use the compatibility option.

You can specify a new compatibility option in version 1.1 of SupportPac CB11. It is specified in the configuration container passed to the SupportPac program. The recommended value of this option is ON, which indicates that the events must be compatible with the WebSphere Business Event format used by CICS TS V4.1 event support.

When you specify compatibility 'ON', the following rules apply:

- The names of the event and of the data items must conform to the naming rules which apply to the event specifications used in the new CICS TS V4.1 event support. The acceptable characters are A-Z a-z 0-9 and `_`. Embedded blank characters are not permitted. The string must not start with 0-9, `_` or `xml` in any case, for example `Xml` or `xML`.
- The connector-object is the event name with `"_Data"` appended.
- A type is not specified for the data items included in the field elements of the generated WebSphere Business Events XML. This allows the type of the data item fields to be specified using the WebSphere Business Events tooling, enabling a wider range of types to be supported.
- If a TYPE container is passed to the SupportPac program, a warning trace entry is issued and a warning code returned in the error container. The TYPE container is ignored and processing continues.

When you do NOT specify compatibility 'ON', the following rules apply:

- There is no validation of the names used for the event or event data.
- The connector-object is the event name with `"Data"` appended.
- A type is specified for the data items in the field elements of the generated WebSphere Business Events XML, either as 'String' which is the default, or as 'Integer' if a string of "numeric" has been passed in the TYPE container associated with the data item. This limits the supported data types to String and Integer.
- If a TYPE container is passed to the SupportPac program, its contents are used to specify the data type in the field element.

This behavior is the same as that provided by version 1.0 of SupportPac CB11. The compatibility option should only be set to OFF if you have a particular need for this behavior.

2. Define the channels and containers that describe the event. You create one channel, which has containers for the event name, and a set of containers for each item of event data.

You define sets of name and data containers that have a matching numeric suffix. The range of nn suffixes must have no gaps. If you do leave gaps, the following rules apply:

- If you leave out any intermediate nn suffix values, any data with higher nn suffix values is ignored.
- If you supply more *EPS.DATA* values than *EPS.NAME* values, the extra data is ignored.
- If you supply fewer *EPS.DATA* values than *EPS.NAME* values, an error is reported.

The event data containers are as follows:

- a. A mandatory container for the name of the event data item.
- b. A mandatory container for the data value of the event data item. Data must be passed to the SupportPac program as character data.
- c. An optional container for the type of the event data item. This is not used when the compatibility option is ON, and is optional when the compatibility option is OFF. When the compatibility option is OFF, the TYPE container is used as follows:

If you do not specify an *EPS.TYPE.nn* container, the data type of the item defaults to text. You can specify numeric, Numeric, or NUMERIC for a numeric type. A type container that contains any other value will cause the type to default to text.

To clarify your program, you can choose to specify text, Text, or TEXT for a text type. The COBOL sample program chooses not to do this, and only supplies an *EPS.TYPE.nn* container where a non-text type is required.

The following example shows definitions for the channel and containers for an event that has three items of data. The second item has the optional type specified.

```

01 EPS-CHANNEL                PIC X(16) VALUE 'EPS.EVENTCHANNEL'.
01 EVENT-NAME-CONTAINER      PIC X(16) VALUE 'EPS.EVENTNAME'.
01 EVENT-CONFIG-CONTAINER    PIC X(16) VALUE 'EPS.CONFIG'.
01 EVENT-ERROR-CONTAINER     PIC X(16) VALUE 'EPS.ERROR'.
01 EVENT-DATA-NAME-1         PIC X(16) VALUE 'EPS.NAME.01'.
01 EVENT-DATA-VALUE-1        PIC X(20) VALUE 'EPS.DATA.01'.
01 EVENT-DATA-NAME-2         PIC X(16) VALUE 'EPS.NAME.02'.
01 EVENT-DATA-VALUE-2        PIC X(20) VALUE 'EPS.DATA.02'.
01 EVENT-DATA-TYPE-2         PIC X(16) VALUE 'EPS.TYPE.02'.
01 EVENT-DATA-NAME-3         PIC X(16) VALUE 'EPS.NAME.03'.
01 EVENT-DATA-VALUE-3        PIC X(20) VALUE 'EPS.DATA.03'.

```

3. Include the supplied definitions for the SupportPac configuration and error containers. Your application uses these definitions to pass configuration information to the EPSEMIT program and to receive error information if an error occurs in EPSEMIT. The following code fragment from the supplied sample COBOL program EPSOLINK shows how to include the definitions:

```

COPY EPSOCONF.
COPY EPSOERR.

```

4. Put the event name into the appropriate container. The event name can be up to 32 characters in length. The code fragment shows an event name Order\_Event.



```

MOVE 'Order_Event' TO EVENT-NAME
EXEC CICS PUT CONTAINER(EVENT-NAME-CONTAINER)
                CHANNEL(EPS-CHANNEL)
                FROM(EVENT-NAME)
END-EXEC

```

5. Specify the configuration information in the configuration container. The configuration information consists of the name of the WebSphere MQ queue to which the event is to be emitted, and the compatibility indicator, for which the recommended value is ON. The code fragment shows a queue name *EP.TEST.EMITQ* on the local queue manager.

```

MOVE 'EP.TEST.EMITQ' TO CON-QUEUE-NAME
MOVE 'ON' TO CON-COMPAT
EXEC CICS PUT CONTAINER(EVENT-CONFIG-CONTAINER)
                CHANNEL(EPS-CHANNEL)
                FROM(EPS-CONFIG)
END-EXEC

```

*CON-QUEUE-NAME* and *CON-COMPAT* are defined in the *EPS0CONF* copybook. Examine the copybook for the language of your program to determine the correct field names in which to set the queue name and compatibility option.

6. For each item of event data, move the event data name and data value to the correct set of containers in the channel.
- The event data *name* can be up to 32 characters in length.
  - The event data *value* can be up to 200 characters in length.
  - The event data *type* can be up to 7 characters in length, anything else defaults to *text*. If compatibility is OFF, the event data *type* can be up to 7 characters in length, with a value of *numeric* if integer data is required; any other value defaults to *text*.

In this code fragment taken from the sample program, the second item *OrderValue* of data is moved to the set of channels with the suffix 2. Note that all data must be passed to the SupportPac program as character data. This example shows how to pass numeric data to the SupportPac program.

```

MOVE 'OrderValue' TO EVENT-DATA-NAME
EXEC CICS PUT CONTAINER(EVENT-DATA-NAME-2)
                CHANNEL(EPS-CHANNEL)
                FROM(EVENT-DATA-NAME)
END-EXEC

MOVE '10' TO EVENT-DATA-VALUE
EXEC CICS PUT CONTAINER(EVENT-DATA-VALUE-2)
                CHANNEL(EPS-CHANNEL)
                FROM(EVENT-DATA-VALUE)
END-EXEC

```

If you wanted to perform some calculation on numeric data, and then pass it to the SupportPac program, the following example shows how to do it.

- \* Given a data item ORDER-VALUE that is defined as follows:
- \* ORDER-VALUE PIC S9(7) COMP.
  
- \* You might for example carry out computation on this item:
- COMPUTE ORDER-VALUE = 20 \* 5
- END-COMPUTE.
  
- \* Put the item into a data container, ensuring you MOVE it into a
- \* variable defined as a character data type,
- \* for example defined as follows:
- \* EVENT-DATA-VALUE PIC X(20) VALUE SPACES.

```

MOVE ORDER-VALUE TO EVENT-DATA-VALUE

EXEC CICS PUT CONTAINER(EVENT-DATA-VALUE-2)
              CHANNEL(EPS-CHANNEL)
              FROM(EVENT-DATA-VALUE)

END-EXEC

```

The names of the event and of the data items must conform to the naming rules which apply to the event specifications used in the new CICS TS V4.1 event support. The acceptable characters are A-Z a-z 0-9 and `_`. Embedded blank characters are not permitted. The string must not start with 0-9, `_` or `xml` in any case, for example `Xml` or `xMl`. In addition, there are the following restrictions on the data passed to the SupportPac:

- Data must be in the default code page
- Data must not contain invalid XML characters, You must replace any invalid XML characters with their escape strings shown here in parentheses: `&` (`&amp;`), `"` (`&quot;`), `<` (`&lt;`), `>` (`&gt;`) and `'` (`&apos;`). The SupportPac does not check that the data contains only valid XML characters. If you fail to replace invalid characters, this can cause problems when the events are processed by WebSphere Business Events.

7. Link to the SupportPac program to emit the event. In this code fragment taken from the sample program, the link is to EPSEMIT, using a channel to pass the configuration and event information. The string EPSEMIT has been moved into `EPS-PROGRAM-NAME`.

```

EXEC CICS LINK PROGRAM(EPS-PROGRAM-NAME)
              CHANNEL(EPS-CHANNEL)
              RESP(RESP) RESP2(RESP2)

END-EXEC.

```

8. Check for any warnings or errors returned by the linked-to EPSEMIT program. If an error container is returned, EPSEMIT has detected an error or a warning, and EPSEMIT passes information on any problem that occurred when attempting to emit the event. See Chapter 8, "Errors returned by the SupportPac program," on page 33 for a list of the possible warning and error codes returned.

```

* Do we have an error container?
EXEC CICS GET CONTAINER(EVENT-ERROR-CONTAINER)
              INTO(EVENT-ERROR-STRUCTURE)
              CHANNEL(EPS-CHANNEL)
              RESP(RESP) RESP2(RESP2)

END-EXEC

IF RESP EQUAL DFHRESP(NORMAL) THEN
  EVALUATE TRUE
    WHEN ERR-ERROR-CODE IS EQUAL TO 0
      CONTINUE
    WHEN ERR-ERROR-CODE IS GREATER THAN 90
      MOVE ' Warning issued by EPSEMIT' TO EM-DETAIL
      MOVE ' Warning = ' TO EM-ERRINTRO
      MOVE ERR-ERROR-CODE TO EM-ERRCODE
      PERFORM WRITE-ERROR-MESSAGE
    WHEN OTHER
      MOVE ' Error occurred in EPSEMIT' TO EM-DETAIL
      MOVE ' ErrCode = ' TO EM-ERRINTRO
      MOVE ERR-ERROR-CODE TO EM-ERRCODE
      PERFORM WRITE-ERROR-MESSAGE
  END-EVALUATE
ELSE

IF RESP EQUAL DFHRESP(NORMAL) THEN
  EVALUATE ERR-ERROR-CODE

```

```

        WHEN 0 CONTINUE
        WHEN OTHER
            MOVE ' Error occurred in EPSEMIT' TO EM-DETAIL
            MOVE ' ErrCode = ' TO EM-ERRINTRO
            MOVE ERR-ERROR-CODE TO EM-ERRCODE
            PERFORM WRITE-ERROR-MESSAGE
        END-EVALUATE
    ELSE
        IF RESP NOT EQUAL DFHRESP(CONTAINERERR) THEN
            MOVE ' Unable to check for errors from EPSEMIT'
              TO EM-DETAIL
            PERFORM WRITE-ERROR-MESSAGE
        END-IF
    END-IF.

```

## The sample applications with CICS events

Two small applications are shipped with the SupportPac, one written in COBOL, the other in Java. Both applications illustrate how to add CICS events to a CICS application program.

Each sample shows how to add events to your CICS application by adding code at each point at which an event is to be created, to populate a channel with a set of containers that describe the event and the destination of the event. Each sample then links to the SupportPac program, EPSEMIT.

### **eps0link.cobol**

The COBOL sample.

### **EPSJLink.java**

The java sample. The java sample uses two helper classes contained in the jar file cicseps.jar:

#### **EPSEvent**

to define the configuration structure and the error structure, and to build containers to pass to the SupportPac program.

#### **EPSEventException**

to define exceptions thrown by the EPSEvent class methods.

The javadoc for the helper classes is supplied in the executables folder in cb11.zip.

---

## Configuring your system to send events to WebSphere Business Events

The SupportPac program EPSEMIT emits an event from CICS to a WebSphere MQ queue. The queue named in the configuration container must be set up on the z/OS system on which CICS runs and your CICS system must be set up to use WebSphere MQ. The queue configuration must allow WebSphere Business Events to receive the event on the system on which it is running

### **About this task**

This task assumes that you use the WebSphere Application Server (WAS) installed with WebSphere Business Events to receive messages from a remote queue on the z/OS system running your CICS system.

Detailed information on setting up communication between a WebSphere MQ system on z/OS and a WebSphere Application Server are in this developerWorks article: [Invoke a WebSphere Application Server Web service from CICS](#).

There are other ways in which you can configure WebSphere MQ such that events can be received by WebSphere Business Events on the system on which it is running. Details of configuring WebSphere MQ can be found in the WebSphere MQ information center. Details of configuring WebSphere Business Events can be found in the WebSphere Business Events information center.

1. Configure your CICS system to use WebSphere MQ. If your CICS system is not already set up to use WebSphere MQ, use the support appropriate to your version of CICS.
  - For CICS TS Version 4 Release 1, you must perform the following tasks:
    - You must install an MQCONN resource definition. Information on MQCONN definitions can be found in the MQCONN and MQINI resource definitions topic in the CICS TS information center.
    - You must configure your system to use the CICS-WebSphere MQ adapter supplied with CICS TS. The CICS-WebSphere MQ adapter connects a CICS subsystem to a WebSphere MQ queue manager, thus enabling CICS application programs to use the Message Queue Interface. Information on installing and using the CICS-WebSphere MQ adapter can be found in the Setting up the CICS-WebSphere MQ adapter topic in the CICS TS information center.
  - For CICS TS Version 3 Release 2, you must configure your system to use the CICS-WebSphere MQ adapter supplied with CICS TS. The CICS-WebSphere MQ adapter connects a CICS subsystem to a WebSphere MQ queue manager, thus enabling CICS application programs to use the Message Queue Interface. Information on installing and using the CICS-WebSphere MQ adapter can be found in The CICS-WebSphere MQ adapter topic in the CICS TS information center.
  - For CICS TS Version 3 Release 1, you must configure your system to use the the CICS adapter supplied with WebSphere MQ. The CICS adapter connects a CICS subsystem to a WebSphere MQ queue manager to enable CICS application programs to use the Message Queue Interface. Information on installing and using the CICS adapter can be found in the The CICS adapter topic in the WebSphere MQ information center.
2. If not already defined, define the WebSphere MQ queue that is named in the configuration container as a remote queue definition on your local z/OS queue manager. The remote queue definition causes WebSphere MQ to route messages to the remote queue manager on which WebSphere Business Events is listening. The remote queue definition contains the remote queue manager name and remote queue name.

The SupportPac uses the priority and persistence defined in the queue definition. Messages written to the queue are set with no expiry, so they remain on the queue until read by WebSphere Business Events or are cleared from the queue.
3. Start an MQ channel on your z/OS system. The channel is used to send the message to the remote queue manager.
4. Start the WebSphere Business Events Connectors program. WebSphere Business Events listens for any defined events and actions.
5. Define a JMS queue on the WebSphere Application Server that is installed with WebSphere Business Events to map to the remote CICS queue. To define a JMS queue, you must have defined a service integration bus, and your WebSphere Application must be defined as a bus member. If you use the WebSphere Application Server installed with WebSphere Business Events, a service integration bus called WbeBus is defined automatically. It contains your default

server as a bus member. The following steps assume you have used the WebSphere Application Server installed with WebSphere Business Events.

- a. Define a destination on the bus.
  - 1) In the Administrative Console for your server select your bus (WbeBus) from the **Service integration** → **Buses** panel.
  - 2) Click **Destinations** and click **New**.
  - 3) Select **Queue** as the destination type and enter an identifier for the queue. This identifier must match the **RNAME** in the z/OS CICS remote queue definition.
  - 4) Assign the destination to the bus member representing the server.
  - 5) Click **Finish** to complete the queue definition.
- b. Define a foreign bus of type MQ. This definition tells the application server that another bus exists and this bus (WbeBus) will communicate with it.
  - 1) In the Administrative Console for your server select your bus (WbeBus) from the **Service integration** → **Buses** panel.
  - 2) Click **Foreign buses** and click **New** to start the wizard that creates a new foreign bus.
  - 3) In step 1 of the wizard, enter the identifier MQ for the foreign bus, and click **Next**.
  - 4) In step 2 of the wizard, specify a routing definition type of **Direct, WebSphere MQ Link**, and click **Next**.
  - 5) Click **Finish** to complete the queue definition.
  - 6) In step 3 of the wizard, no userids are required, so click **Next**.
  - 7) Click **Finish** to complete the foreign bus definition.
- c. Define a link to WebSphere MQ. In your bus (WbeBus), you must configure a messaging engine to provide a link to a WebSphere MQ network.
  - 1) In the Administrative Console for your server select your bus (WbeBus) from the **Service integration** → **Buses** panel.
  - 2) Click **Messaging engines** and select the existing messaging engine. The **configuration** tab is displayed.
  - 3) Click **WebSphere MQ links**.
  - 4) Click **New** to start the wizard that creates a new link.
  - 5) In step 1 of the wizard:
    - a) Enter MQLink in the **Name** field.
    - b) Enter your bus name WbeBus in the **Queue Manager Name** field.
    - c) Enter MQ in the **Foreign Bus Name** field.
    - d) Click **Next**.
  - 6) In step 2 of the wizard, you define a sender channel to send messages to WebSphere MQ. The name of the sender channel must match the name of the receiver channel on your z/OS WebSphere MQ system.
    - a) Enter T0.MQ in the **Sender MQ channel name** field.
    - b) Enter the hostname of the machine running your WebSphere MQ system in the **Host name** field.
    - c) Enter the port on which your WebSphere MQ system is listening for TCP/IP connections (default 1414) in the **Port** field.
    - d) Select **OutboundBasicMQLink** from the **Transport chain** list.
    - e) Click **Next**.

- 7) In step 3 of the wizard, you define a receiver channel to receive messages from WebSphere MQ. The name of the receiver channel must match the name of the sender channel on your z/OS WebSphere MQ system.
  - a) Enter T0.WbeBus in the **Receiver MQ channel name** field.
  - b) Click **Next**.
- 8) Click **Finish** to save the definition.

## Results

You have configured your z/OS system and the system on which you are running WebSphere Business Events.

---

## Creating WebSphere Business Events definitions for CICS events

To process events emitted from your CICS system and send to a JMS queue on the WebSphere Application Server on your WebSphere Business Events system, you create definitions using the WebSphere Business Events interfaces.

### About this task

WebSphere Business Events coordinates activities, known as events and actions, between business applications known as touchpoints. WebSphere Business Events uses intermediate objects to provide a level of indirection between incoming events and their business specifications.

Further details of WebSphere Business Events concepts can be found in the WebSphere Business Events:Design Data topic in the WebSphere Business Events information center.

WebSphere Business Events provides several interfaces to define and control the components in the system. To create the definitions required for CICS events, you must be familiar with these two interfaces:

#### **WebSphere Business Events:Design Data.**

This interface supports the definition of the metadata layer required for business event processing. You use WebSphere Business Events:Design Data to create the building blocks for your application, including intermediate objects, touchpoints, events, and actions.

#### **WebSphere Business Events:Design**

This interface supports the definition of interaction sets (a collection of interaction blocks containing business logic) used at run time by the WebSphere Business Events runtime server to detect, correlate, and respond to simple and complex events. Interaction Sets can also contain Filters used to further define the conditions under which an interaction block is true, and this interface is also used to define those filters.

The interface also supports creating and managing application event flows, which describe a business process. Interaction sets defined by WebSphere Business Events:Design can be incorporated into event flows.

For further information see the WebSphere Business Events user interface help in the WebSphere Business Events information center.

You need to use the WebSphere Business Events:Design Data interface to define a touchpoint for your CICS system and an associated intermediate object. You use the WebSphere Business Events:Design interface to define filters on the intermediate object.

The following tasks show you how to use these interfaces. They use an example "order event" emitted from a CICS system. The example event has a payload of three items of data, *CustomerId*, *OrderItem*, and *OrderValue*. The example shows how definitions can be made using the WebSphere Business Events interfaces to receive and process an event from CICS according to an interaction set, and to specify a subsequent action. You do not have to use precisely these steps to process all your events from CICS; you can construct WebSphere Business Events event flows in many other ways.

The following information takes you through the steps involved in mapping data fields to an intermediate object, and later of mapping from the intermediate object fields to an action object, but note that this can also be done by dragging and dropping the relevant fields to the object where you want to map them.

## Adding touchpoint and intermediate object definitions

You add a touchpoint to represent the WebSphere MQ link to your CICS system. You define events in the touchpoint. You define an intermediate object to represent the link between the incoming event and specifications made by the business about how to respond to events.

### About this task

When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on adding, deleting, and viewing touchpoints, can be in the Touchpoints topic in the WebSphere Business Events information center.
  - Information on events, can be found in the Events topic in the WebSphere Business Events information center.
  - Information on intermediate objects, can be found in the Intermediate Objects topic in the WebSphere Business Events information center.
1. Open the WebSphere Business Events:Design Data interface.
  2. Insert a touchpoint
    - a. Click in the Touchpoint workspace, and select **Touchpoint** from the **Insert** menu.
    - b. Enter a descriptive name for the touchpoint, and click **OK**. In the example, the touchpoint has the name CICSMLink.
  3. Add an event to the new touchpoint.
    - a. Right-click the new touchpoint in the touchpoint workspace, and select **Insert Event** → **Normal** from the menu.
    - b. Enter a descriptive name for the event, and click **OK**. In the example, the event has the name *Order\_Event*. The name must match the value in the **EPS.EVENTNAME** container supplied to the SupportPac program EPSEMIT.
  4. Add an event object to the new event. An event object is an occurrence of the data that is passed in an event.
    - a. Right-click the new event in the touchpoint, and select **Insert Event Object** → **Blank** from the menu.

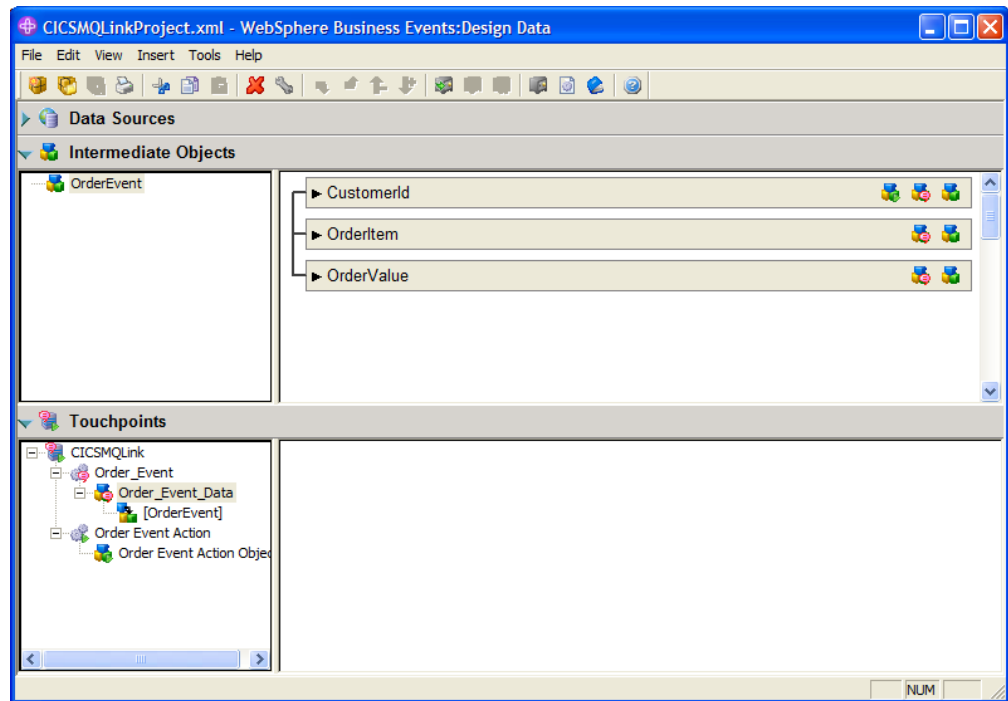


- b. Enter a descriptive name for the event object, and click **OK**. In the example, the event object has the name `Order_Event_Data`. The name must match the event name concatenated with the text `"_Data"`.
5. Add a intermediate object to represent the business data.
  - a. Click in the Intermediate Objects workspace, and select **Intermediate Object** → **Blank** from the **Insert** menu.
  - b. Enter a descriptive name for the intermediate object, and click **OK**. In the example, the intermediate object has the name `OrderEvent`
6. For each item of business data that your intermediate object contains, add an intermediate object field to the intermediate object you have just created.
  - a. Right-click the intermediate object you have just created, and select **Insert Intermediate Object Field** from the menu.
  - b. Enter the name, data type, and description of the data item. The field names must match those in the `EPS.NAME.nn` containers supplied to the SupportPac program EPSEMIT. The field names must all be present in `EPS.NAME.nn` containers for the defined action to be performed.  
 In the example, the intermediate object has three fields, *CustomerId*, *OrderItem* and *OrderValue*.
7. For each item of data in your event, map the item of data to an intermediate object field in the new intermediate object by inserting a field constructor in your event object.
  - a. Right-click the event object in the Touchpoint workspace and select **Insert Field Constructor from Object** → *Intermediate Object* → *Business Data Item* from the menu. In the example, if you wanted to map the *CustomerId* in the intermediate object **OrderEvent** to your event object, the menu item selected would be **Insert Field Constructor from Object** → **OrderEvent** → *CustomerId*.

## Example

After completing these steps, the WebSphere Business Events:Design Data interface will look similar to this display:





## Defining a message queue connector

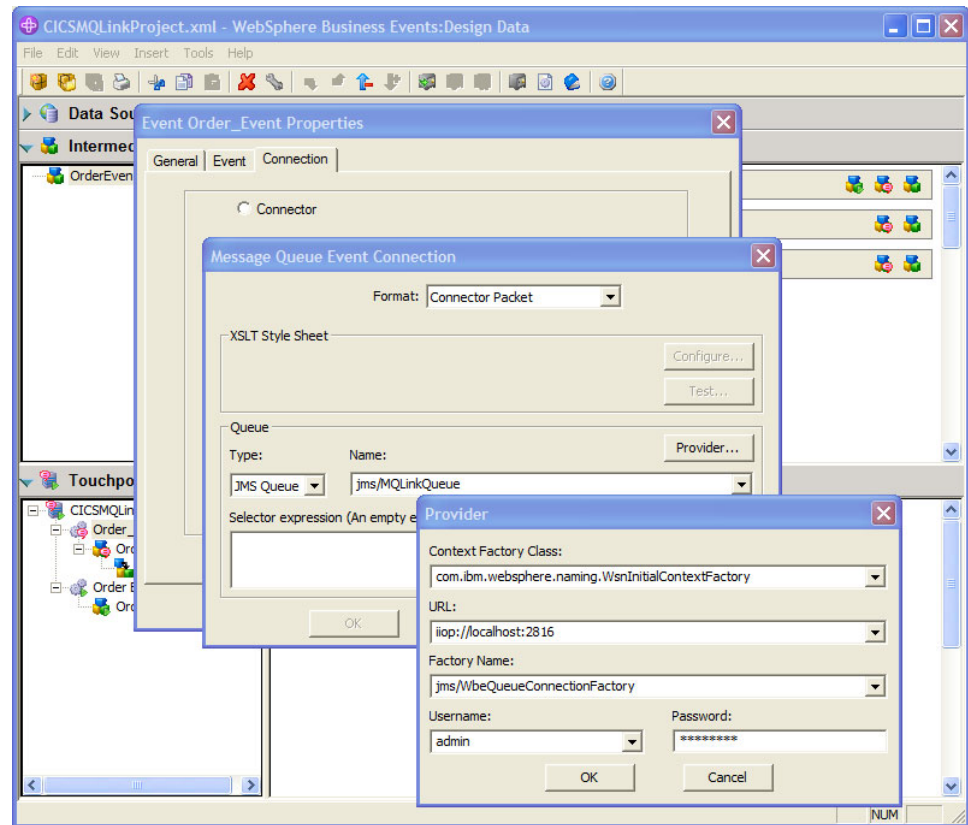
You must add a WebSphere Business Events message queue connector to the properties of your event to allow WebSphere Business Events to receive event messages from CICS on the JMS queue.

### About this task

When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on adding, deleting and viewing touchpoints, can be found in the Touchpoints topic in the WebSphere Business Events information center.
  - Information on defining an embedded message queue event connector , can be found in the Defining an Embedded Message Queue Event Connector topic in the WebSphere Business Events information center.
1. In the WebSphere Business Events:Design Data interface, right-click your event in the touchpoint and select **Event Properties** from the menu.
  2. Select the **Connection** tab and click **Message Queue Connection**. The Message Queue Event Connection window is displayed.
  3. From the **Format** list, select **Connector Packet**.
  4. From the **Type** list, select **JMS Queue**.
  5. From the **Name** list, select the name of your JMS queue.
  6. Click **Provider** to edit the provider details.
    - a. Select **com.ibm.websphere.naming.WsnInitialContextFactory** from the **Context Factory Class** list.
    - b. Select the iiop address of your WAS server from the **URL** list.
    - c. Select **Factory Name** from the **JMS QueueConnectionFactory** list.
    - d. Type the **Username** and **Password** of your WAS server.

The WebSphere Business Events: Design Data interface should look similar to this display:



7. Click **OK** in the Provider window.
8. Click **OK** in the Message Queue Event Connection window.
9. Click **OK** in the Event Properties window.

## Defining actions

An action is an occurrence of some activity on a business application. It is driven by WebSphere Business Events based on the evaluation of Interaction Blocks in the runtime system. You define an action in your touchpoint for CICS events.

### About this task

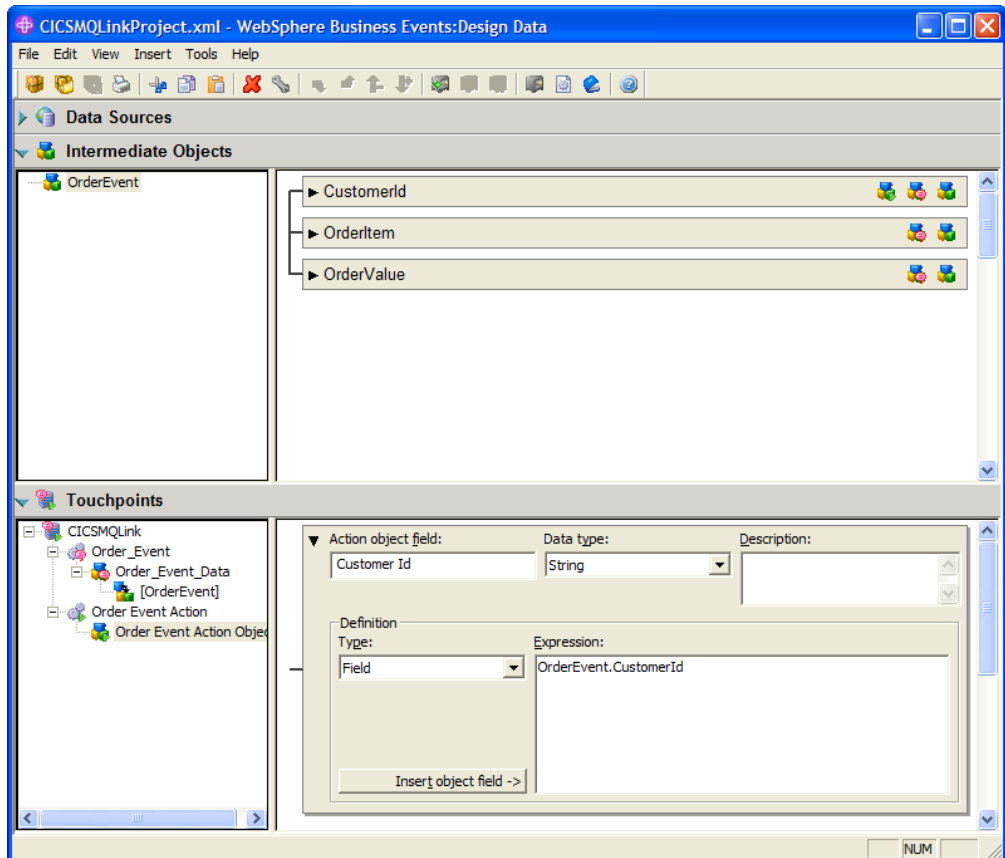
When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on adding, deleting and viewing touchpoints, can be found in the Touchpoints topic in the WebSphere Business Events information center.
  - Information on actions, can be found in the Actions topic in the WebSphere Business Events information center.
  - Information on intermediate objects, can be found in the Intermediate Objects topic in the WebSphere Business Events information center.
1. In the WebSphere Business Events:Design Data interface, right-click your event in the touchpoint and select **Insert Action** → **Normal** from the menu.
  2. Enter a descriptive name for the action, and click **OK**. In the example, the action has the name Order Event Action.

3. Add an action object to the new action. An action object is an occurrence of the data that is passed in an action.
  - a. Right-click the new action in the touchpoint , and select **Insert Action Object** → **Blank** from the menu.
  - b. Enter a descriptive name for the action object, and click **OK**. In the example, the action object has the name **Order Event Action Object**
4. For each data item from the event that you wish to include in the message generated by the action, add an action object field to the new action object. Action object fields are individual data elements in WebSphere Business Events:Design Data objects. Fields can be added to intermediate objects, events objects, and actions objects.
  - a. Right-click the new action object in the touchpoint , and select **Insert Action Object Field** from the menu. The Insert Field window is displayed.
  - b. Enter a descriptive name for the action object Field. In the example, the action object has the name **CustomerId**. The name must match the name of the field as defined in the action connector to be passed to the target touchpoint system.
  - c. From the **Type** list, select the type of the field. In the example, the action object field **CustomerId** has the type String. The type must match the type of the field as defined in the action connector to be passed to the target touchpoint system.
  - d. Click **OK**.
5. For each action object field you have defined, you must map the action object field to an intermediate object field.
  - a. Click the twistie for the action object field in the right pane of the touchpoint workspace. A detailed view of your action object field is displayed.
  - b. From the **Type** list, select the type of the definition. In the example, the action object field **CustomerId** is mapped to the **CustomerId** field in the **OrderEvent** intermediate object, and so has the same type of **Field**.  
After you have selected the type **Field**, click the **Insert object field** button and choose the intermediate object field to which you want to map.  
In the example, **Order\_Event** → **CustomerId** is selected from the menu.

## Example

After completing these steps, the WebSphere Business Events:Design Data interface will look similar to this display:



## Defining a file system connector

You can create any action properties to use with the SupportPac, not just a file system connector. This task shows an example of what you can do, using a file system connector.

### About this task

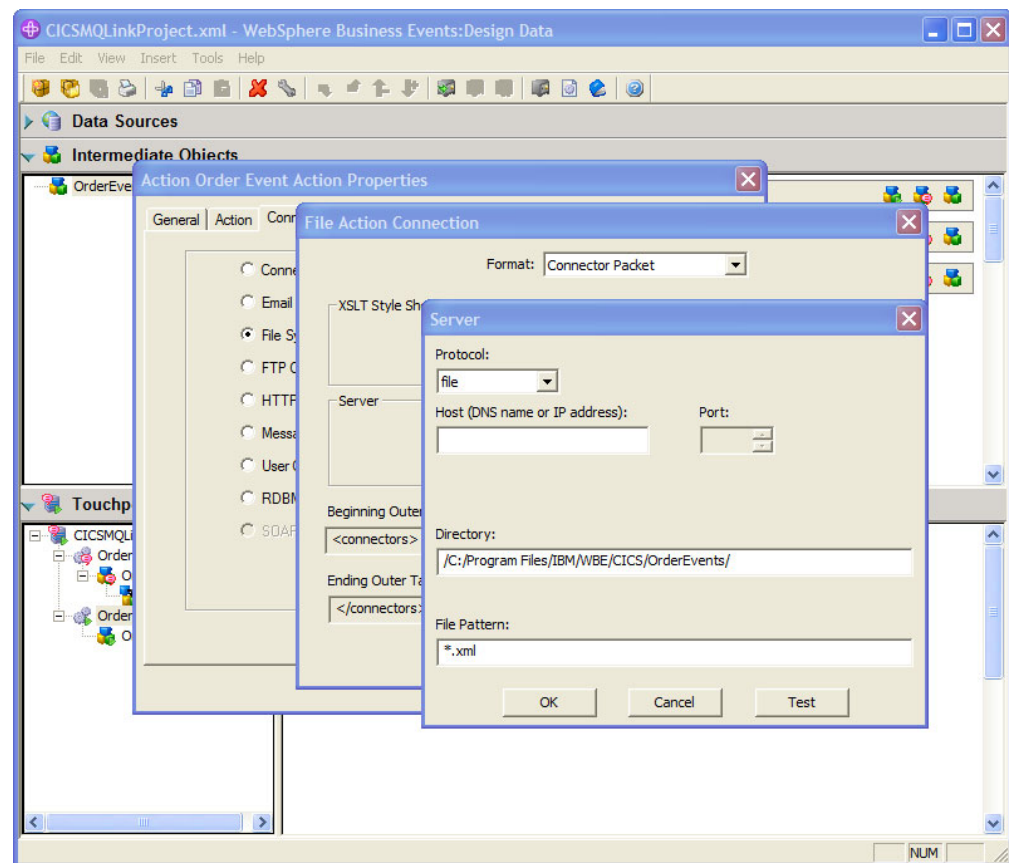
You can add a WebSphere Business Events connector to the properties of your action to enable WebSphere Business Events to perform an action in response to an event from CICS.

When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on defining an embedded file system action connector can be found in the Defining an Embedded File System Action Connector topic in the WebSphere Business Events information center.
- In the WebSphere Business Events:Design Data interface, right-click your action in the touchpoint and select **Action Properties** from the menu.
  - Select the **Connection** tab and click **File System Connection**. The File Action Connection window is displayed.
  - From the **Format** list, select **Connector Packet**.
  - Click **Configure** in the **Server** pane. The Server window is displayed.
  - In the **Directory** field, specify a directory to write files to, and specify a **File Pattern** of \*.xml. The WebSphere Business Events:Design Data interface should look similar to the example display below:

6. Click **OK** in the Server, File Action Connection and Action Properties windows.

## Example



## Creating a filter

You create a filter to represent criteria that are evaluated at run time to determine whether or not an action should be taken.

### About this task

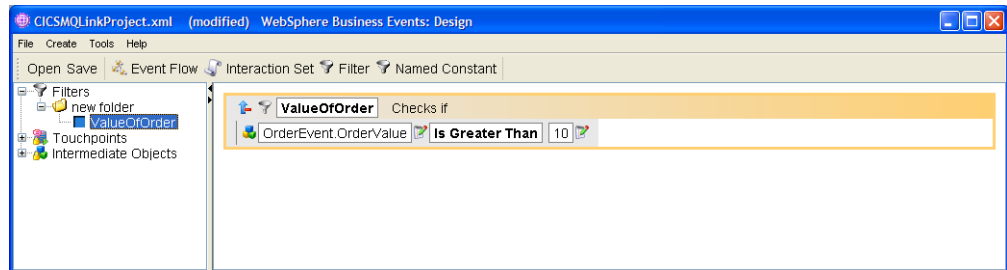
You use the WebSphere Business Events:Design interface to add filters and interaction sets to your project. When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on the WebSphere Business Events:Design interface, can be found in the WebSphere Business Events:Design topic in the WebSphere Business Events information center.
  - Information on adding, deleting and viewing filters, can be found in the Filters topic in the WebSphere Business Events information center.
1. Open the WebSphere Business Events:Design interface.
  2. Select **Filter** from the **Create** menu.
  3. Enter a descriptive name for the filter. In the example, the event has the name **ValueOfOrder**. The detail view of a new filter is displayed in the workspace.
  4. Click the left **Operand?** field and choose a field from the **Available Fields** list. In the example, the operand selected is **OrderValue**.

5. Click the **Operator?** field and choose an operator from the **Available Operators** list. In the example, the operator selected is **Is Greater Than**.
6. Click the right **Operand?** field and choose an item from the list. In the example, the **enter text** option is selected and the value 10 entered.
7. Save the project.

## Example

After completing these steps, the WebSphere Business Events:Design interface will look similar to this display:



## Creating an interaction set

You define an interaction set to process your events using filters and produce actions when the required criteria are met. An interaction set is the set of business logic, defined in one or more interaction blocks, that controls the traffic between business systems, which are also referred to as touchpoints.

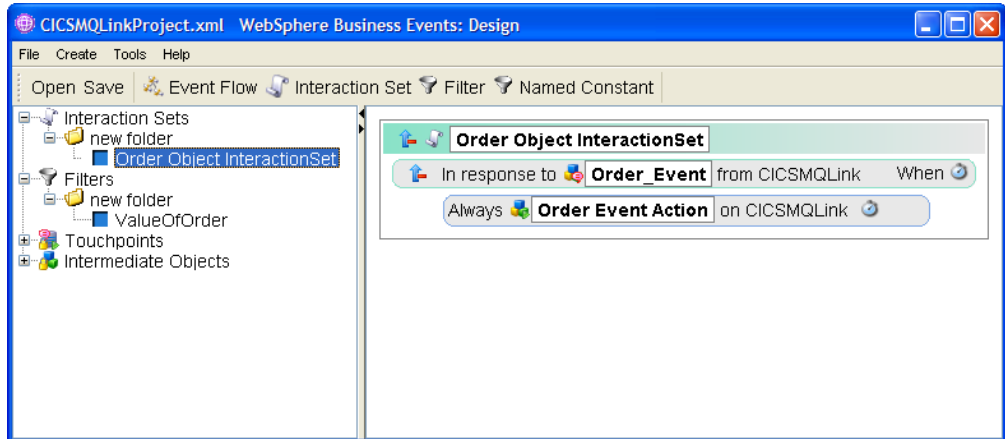
### About this task

You use the WebSphere Business Events:Design interface to add an interaction set to your project. When performing this task, you might find the following sections of the WebSphere Business Events information center useful:

- Information on the WebSphere Business Events:Design interface, can be found in the WebSphere Business Events:Design topic in the WebSphere Business Events information center.
  - Information on adding, deleting and viewing an interaction set, can be found in the Interaction Sets topic in the WebSphere Business Events information center.
1. Open the WebSphere Business Events:Design interface.
  2. Select **Interaction Set** from the **Create** menu. The detail view of a new interaction set is displayed in the workspace.
  3. Enter a descriptive name for the interaction set. In the example, the event has the name **Order Object InteractionSet**.
  4. Click the **Event?** field and choose an event from the **Available Events** list. In the example, the event selected is **Order\_Event** from **CICSMQLink**.
  5. Click the **Action?** field and choose an action from the **Available Actions** list. In the example, the action selected is **Order Event Action**.
  6. Save the project.

## Example

After completing these steps, the WebSphere Business Events:Design interface will look similar to this display:







---

## Chapter 8. Errors returned by the SupportPac program

The SupportPac program returns error information to your application in an error container.

The error container has the following structure.

- A 4-byte error code
- A 16-character field which has the following contents:
  - The name of the container for container-related errors.
  - The channel name if the name is not **EPS.EVENTCHANNEL**.
  - Spaces for any other errors.
- A 4-byte field which is the MQ reason code for WebSphere MQ errors. The MQ reason codes are described in the MQRC\_\* (Reason Codes) topic in the WebSphere MQ Version 6 information center, and in the MQRC\_\* (Reason Codes) topic in the WebSphere MQ Version 7 information center.

The following code fragment shows the structure for the error container in COBOL:

```
* CICS Events for WBE Error Structure
01  EVENT-ERROR-STRUCTURE.
*   CICS Events SupportPac Error Code
    03  ERR-ERROR-CODE          PIC S9(8) COMP VALUE +0.
*   Container name for container-related errors
    03  ERR-CONTAINER          PIC X(16) VALUE SPACES.
*   MQ Error code for MQ-related errors
    03  ERR-MQ-ERRCODE         PIC S9(8) COMP VALUE +0.
```

This list shows the error codes that the SupportPac program can return to your application in the error container.

- 21 A channel with an incorrect name was passed to the CICS events SupportPac. The channel name is returned in the ERROR container if possible.
- 22 An error occurred on getting a container. The container name is included in the structure in the ERROR container, allowing you to identify whether the error was on getting the name of the event, the name, value, or type of an item of event data, or on getting event configuration information.
- 23 A length error occurred on getting from a container. The container name is in the structure in the ERROR container, allowing you to identify whether the error was on getting the name of the event, the name, value, or type of an item of event data, or on getting event configuration information.
- 24 An error other than a container error or length error occurred on getting a container. The container name is in the structure in the ERROR container, allowing you to identify whether the error was on getting the name of the event, the name, value, or type of an item of event data, or on getting event configuration information.
- 25 An *EPS.NAME.nn* container has no corresponding *EPS.DATA.nn* container. The name of the *EPS.DATA.nn* container for which the mismatch was found is in the error data
- 26 An error occurred on putting error information into the error container.

- 27 A data value container of a data item of data type numeric is empty. A numeric data value container must contain a value. The name of the *EPS.DATA.nn* container is in the error data.
- 28 The name of the data item in container *EPS.NAME.nn* is empty, or all spaces. It is not possible to define such an item of data to WebSphere Business Events. The name of the *EPS.NAME.nn* container is in the error data.
- 30 WebSphere MQ is not configured for this CICS region.
- 31 An error occurred attempting to open a connection to WebSphere MQ. The MQ error reason code is included in the ERROR container.
- 32 An error occurred on putting the event to the specified WebSphere MQ queue. The MQ error reason code is included in the ERROR container.
- 33 An error occurred attempting to close the connection to WebSphere MQ. The MQ error reason code is included in the ERROR container.
- 40 An error occurred on getting storage.
- 41 An error occurred on freeing storage.
- 42 The name of the event or one of the event data items does not conform to the naming rules. The acceptable characters are A-Z a-z 0-9 and `_`. Embedded blank characters are not permitted. The string must not start with 0-9, `_` or `xml` in any case, for example `Xml` or `xML`. The name of the container in which the invalid name was passed is in the error container. User trace entries indicate the name or names which do not conform.
- 50 The CICS release on which the SupportPac is running is not a supported release.

This list shows the warning codes that the SupportPac program can return to your application in the error container:

- 92 A type container was specified when compatibility with CICS TS V4.1 was requested. The type container will be ignored. Processing continues.

---

## Chapter 9. Abends issued by the SupportPac program

The SupportPac program EPSEMIT can issue the following abends.

Abend code	Explanation
EPSA	The SupportPac program, EPSEMIT, was called without a CHANNEL specified.
AEXY	A purged response was returned on a request issued by the SupportPac program, EPSEMIT.
AEXZ	A severe error occurred during processing in the SupportPac program, EPSEMIT.



---

## Chapter 10. Diagnosing problems with the SupportPac program

The SupportPac program can issue user trace entries to assist you in diagnosing problems.

For diagnosis of problems using the CICS Events for WBE SupportPac ensure that you have CICS user trace enabled. If you need to report a problem to IBM, the following trace options are recommended:

- User trace enabled, for the user trace points issued by EPSEMIT
- EI level 1 trace enabled
- SM level 1 trace enabled
- PG level 1 trace enabled
- RI level 1 trace enabled

Trace identifier	Trace message contents
'0010'x	Entry to EPSEMIT
'0011'x	Exit from EPSEMIT
'0012'x	The name of the WebSphere MQ queue to which the event will be emitted.
'0013'x	The version of the SupportPac that is running.
'0014'x	The formatted WebSphere Business Events XML to be emitted. More than one trace entry can be issued to trace all of the XML.
'0020'x	Exception: No channel was passed to EPSEMIT.
'0021'x	Exception: The channel passed to EPSEMIT has the wrong name. The trace data includes the channel name.
'0022'x	Exception: An CONTAINERERR occurred on getting one of the containers used by the SupportPac program. The trace data includes the container name.
'0023'x	Exception: A LENGERR occurred on getting one of the containers used by the SupportPac program. The trace data includes the container name.
'0024'x	Exception: An error other than CONTAINERERR or LENGERR occurred on getting one of the containers used by the SupportPac program. The trace data includes the container name.
'0027'x	Exception: The <i>EPS.DATA.nn</i> container for a data item with a numeric TYPE is empty. The trace data includes the container name.
'0028'x	Exception: The <i>EPS.NAME.nn</i> container is empty. The trace data includes the container name.
'0030'x	Exception: WebSphere MQ is not set up for this CICS system.
'0031'x	Exception: Error on attempting to open the WebSphere MQ queue. The trace data includes the MQ reason code.
'0032'x	Exception: Error attempting to put to the WebSphere MQ queue. The trace data includes the MQ reason code.
'0033'x	Exception: Error attempting to close the WebSphere MQ queue. The trace data includes the MQ reason code.

Trace identifier	Trace message contents
'0040'x	Exception: Error attempting to get storage.
'0041'x	Exception: Error attempting to free storage.
'0042'x	Exception: Name of the event or an item of data does not conform to the naming rules. The invalid name is included in the trace entry, and the container in which it was passed.
'0050'x	Exception: The CICS release on which the SupportPac is running is not a supported release.
'0091'x	Warning, the CICS release is V4.1, consider moving to new style of event specification.
'0092'x	Warning, TYPE container has been specified when compatibility with CICS V4.1 is ON and will be ignored. The trace data includes the container name.

---

## Chapter 11. Creating CICSplex SM definitions using BATCHREP

You can use this BATCHREP (batch repository update) input to create program definitions for the EPSEMIT program and the COBOL sample program and transaction.

### Before you begin

Before running BATCHREP, you must edit the input to add the CONTEXT statement to identify your CICSplex.

Submit any of the following CREATE commands you require to BATCHREP to create definitions in the CICSplex SM data repository.

```
CONTEXT mycplex;
CREATE RESGROUP RESGROUP(EPSRES)
                DESCRIPTION( )
                ;
CREATE PROGDEF  NAME(EPSEMIT)
                DESCRIPTION(CICS EVENTS FOR WBE PROGRAM)
                LANGUAGE(ASSEMBLER)
                RELOAD(NO)
                RESIDENT(NO)
                USAGE(NORMAL)
                USELPACOPY(NO)
                STATUS(ENABLED)
                CEDF(NO)
                DATALOCATION(ANY)
                EXECKEY(CICS)
                CONCURRENCY(THREADSAFE)
                API(CICSAPI)
                DYNAMIC(NO)
                REMOTESYSTEM( )
                REMOTENAME( )
                TRANSID( )
                EXECUTIONSET(FULLAPI)
                JVM(NO)
                JVMCLASS( )
                JVMPROFILE(DFHJVMPR)
                RESGROUP(EPSRES)
                ;
CREATE PROGDEF  NAME(EPSOLINK)
                DESCRIPTION(CICS EVENTS FOR WBE SAMPLE PROGRAM)
                LANGUAGE(COBOL)
                RELOAD(NO)
                RESIDENT(NO)
                USAGE(NORMAL)
                USELPACOPY(NO)
                STATUS(ENABLED)
                CEDF(YES)
                DATALOCATION(ANY)
                EXECKEY(USER)
                CONCURRENCY(QUASIRENT)
                API(CICSAPI)
                DYNAMIC(NO)
                REMOTESYSTEM( )
                REMOTENAME( )
                TRANSID( )
                EXECUTIONSET(FULLAPI)
                JVM(NO)
```

```

        JVMCLASS( )
        JVMPROFILE(DFHJVMPR)
        RESGROUP(EPSRES)
        ;
CREATE TRANDEF  NAME(EPSK)
                DESCRIPTION(SAMPLE TRANSACTION TO TEST EP SUPPORTPAC)
                PROGRAM(EPSOLINK)
                TWASIZE(0)
                PROFILE(DFHCICST)
                PARTITIONSET( )
                STATUS(ENABLED)
                TASKDATALOC(ANY)
                TASKDATAKEY(USER)
                STORAGECLEAR(NO)
                RUNAWAY(SYSTEM)
                SHUTDOWN(DISABLED)
                ISOLATE(YES)
                BREXIT( )
                DYNAMIC(NO)
                ROUTABLE(NO)
                REMOTESYSTEM( )
                REMOTENAME( )
                TRPROF( )
                LOCALQ( )
                PRIORITY(1)
                TRANCLASS(DFHTCL00)
                ALIAS( )
                TASKREQ( )
                XTRANID( )
                TPNAME( )
                XTPNAME( )
                DTIMOUT(NO)
                RESTART(NO)
                SPURGE(NO)
                TPURGE(NO)
                DUMP(YES)
                TRACE(YES)
                CONFDATA(NO)
                OTSTIMEOUT(NO)
                FAILACTION(BACKOUT)
                WAIT(YES)
                WAITTIMEDD(0)
                WAITTIMEHH(0)
                WAITTIMEMM(0)
                RESSEC(NO)
                CMDSEC(NO)
                RESGROUP(EPSRES)
                ;

```



---

## Chapter 12. Migrating to V4.1 event support

You can migrate CICS events currently emitted using SupportPac CB11 to use the richer and more flexible capabilities of the events support in CICS TS V4.1.

---

### Identifying the events to be migrated

If you are not sure which events are currently being emitted using SupportPac CB11, there are a number of ways to find out.

Determine which events are currently being emitted using SupportPac CB11.

- Use the CICS Interdependency Analyzer product to identify any link requests to the program EPSEMIT (or other name if you have chosen to rename the program).
- Use a code analysis tool, such as WebSphere Studio Asset Analyzer, to scan your code base for EXEC CICS LINK requests to EPSEMIT.
- Enable user trace, and look for user trace entries issued by the SupportPac. An example is '0010'X, which indicates entry to the EPSEMIT program.

---

### Migrating events to use event specifications

Over time, events which are currently emitted from CICS using the CICS Events for WBE SupportPac should be migrated to use the new approach provided by CICS TS V4.1.

#### About this task

If you specify compatibility option ON, there should not be changes required to event definitions and other filters and interactions that have been set up in WebSphere Business Events to receive and process the events.

Migrate using one of these two methods.

- Identify an existing EXEC CICS command issued by the program which can be used as an indicator that the event has occurred. This is likely to be a command close to the point at which code was added to invoke the SupportPac program. Create and deploy an event specification and associated capture specification, as part of an Event Binding within a Bundle, which references this command and values available at the time it is issued. This causes the event to be captured and emitted automatically by CICS.

Using this option, you can delete the code which sets up EPS.EVENTCHANNEL and links to EPSEMIT when you have the alternative event capture working.

Advice on creating event bindings and capture specifications can be found in the Specifying business events using the CICS event binding editor topic in the CICS TS information center.

- Replace the EXEC CICS LINK to EPSEMIT by an explicit EXEC CICS SIGNAL EVENT call. Create and deploy an event specification and associated capture specification, as part of an Event Binding within a Bundle, which uses this command and values available at the time it is issued. This causes the event to be captured and emitted automatically by CICS.

Advice on creating event bindings and capture specifications can be found in the Specifying business events using the CICS event binding editor topic in the CICS TS information center.

An example is given below of replacing the EXEC CICS LINK in the supplied COBOL sample, EPS0LINK, with a SIGNAL EVENT command and an event binding that references it.

---

## Specifying an EP adapter for migrated events

When specifying business events using the CICS event binding editor, you must specify an EP adapter to emit events for your event specification.

### About this task

Advice on specifying an EP adapter can be found in the Specifying adapter and dispatcher information topic in the CICS TS information center.

1. Select the **WMQ Queue** adapter.
2. Enter the **Queue Name** which was passed in the configuration container.
3. Optional: Specify the persistence, priority and expiry time for WMQ messages. The CICS event binding editor provides options for persistence, priority and expiry time. However, if you do not specify these, the default values match the values used by SupportPac CB11. The defaults are that the persistence and priority are taken from the queue definition, and the expiry time is set to never expire.
4. Select the **WebSphere Business Events (XML)** data format.

### Results

You have correctly configured an EP adapter for you event specification. An additional advantage of using event specifications is that it is easy to switch the events in an event binding to use a different EP adapter; for example, you could use the **TS queue** EP adapter to test the events.

---

## An example of replacing EXEC CICS LINK by EXEC CICS SIGNAL EVENT

The supplied COBOL sample program, EPS0LINK, sets up the event and configuration details in a channel EPS.EVENTCHANNEL and specifies this channel on an EXEC CICS LINK command.

These are the values used in the sample:

- Event name in EPS.EVENTNAME container: Order\_Event
- WMQ Queue name in EPS.CONFIG container: EP.TEST.EMITQ
- Name of first item of event data in EPS.NAME.01 container: OrderItem
- Value of first item of event data in EPS.DATA.01 container: '12345 '
- Name of second item of event data in EPS.NAME.02 container: OrderValue
- Value of second item of event data in EPS.DATA.02 container: '10'
- Name of third item of event data in EPS.NAME.03 container: Customer
- Value of third item of event data in EPS.DATA.03 container: 'Someone Important'

This is the command used to link to the SupportPac program:

```
EXEC CICS LINK PROGRAM(EPS-PROGRAM) CHANNEL(EPS-CHANNEL)
```

(where EPS-PROGRAM and EPS-CHANNEL are 'EPSEMIT' and 'EPS.EVENTCHANNEL' respectively). You can replace this by the following EXEC CICS SIGNAL EVENT command:

```
EXEC CICS SIGNAL EVENT('Order_Event') FROMCHANNEL(EPS-CHANNEL)
```

The only information required in the FROMCHANNEL are the values of the three data items. The other data will be provided in the event binding, so the PUT CONTAINER commands for the other information can be removed.

The following steps describe how to create an event binding to cause this event to be emitted to WBE. You can use a similar approach for your own events. This description should be read in conjunction with the advice on creating event bindings and capture specifications found in the Specifying business events using the CICS event binding editor topic in the CICS TS information center.

1. In the CICS Explorer resource perspective, either create a new CICS bundle project and add a new event binding to it, or add a new event binding to an existing CICS bundle project. In this example, the event binding name is EPSOLINKEvent.
2. Add an event specification to the event binding with the name of the event, in this example, Order\_Event.
3. Edit the event and add the names of the items of event data to the table of emitted business information items. In this example, there are three items of emitted business information, OrderItem, OrderValue and Customer. You must provide information on their type (text or numeric), precision and length. The precision in all three cases will be 0. The types and lengths in this example are:

OrderItem	text	5
OrderValue	numeric	10
Customer	text	17

4. The CICS event binding editor has an **Add An Automatic Capture Specification** button. This is the easiest way to create a capture specification for this event, which can then be edited. Give the capture specification a name, in this example 'EPSOrder\_Event\_Capture'.
5. The automatically created capture specification is given a capture point of SIGNAL EVENT and a filtering predicate of EVENT Equals Order\_Event, which do not require changing.

The process also fills in the Information Sources with how to obtain the OrderItem, OrderValue, and Customer from the SIGNAL EVENT command, but assumes these will be in containers in the FROMCHANNEL named OrderItem, OrderValue and Customer (and that the data is at offset 0 in each container, and of the length specified in the Emitted Business Information).

You have the option either to change the program to use these container names (or others of your choosing) or to edit the Information Sources in the capture specification to use the names of the containers as used with the SupportPac. For example, EPS.DATA.01, EPS.DATA.02 and EPS.DATA.03. The prefilled Information Sources also assume a 'Capture Type' (data type within the program) of character, which matches that used by the SupportPac.

6. The EP adapter configuration in the event binding should be set up with an adapter of WMQ Queue, a queue name of EP.TEST.EMITQ , and a format of **WebSphere Business Events (XML)**.

All other values can take the defaults, but you might want to change one item under the 'Advanced Options' in the Adapter. The userid under which the

event will be written to the WMQ queue will by default be the CICS default userid, and in some circumstances you might want to select the 'Use Context User Id' option, which will cause the event to be written under whichever userid was running when the event occurred. This matches the behavior of the SupportPac.

7. Export the Bundle project to zFS using the option in the CICS Explorer, define a CICS Bundle resource to reference the location on zFS, and install this bundle to install the event binding. When the event binding is enabled, then the events will be emitted to WBE when they occur.

---

## 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.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (<sup>®</sup> or <sup>™</sup>), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

CICS  
z/OS

IBM

WebSphere

Java and all Java-based trademarks 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 company, product, and service names may be trademarks or service marks of others.



