

Understand how the new features  
in CICS event processing can help  
your business

July 2011



## **IBM CICS event processing: New features in V4.2**

*Jenny He, Developer, CICS Transaction Server  
Catherine Moxey, Senior Developer, CICS Transaction Server*



CICS icon of progress: <http://www.ibm.com/ibm100/us/en/icons/cics/>

## ***Introduction***

IBM® CICS® Transaction Server for z/OS® Version 4.1 introduced capability to support event processing for business applications. Event processing can assist the business community in gaining insight into how their business processes are performing. Furthermore, event processing can help companies to take advantage of new business opportunities by providing a non-invasive methodology for enhancing existing business applications.

With the potential offered by event processing support, CICS Transaction Server for z/OS Version 4.2 enhances this capability with a number of new features. In this paper we are going to talk about some of the most significant of these new features, such as assured event emission, separate event processing (EP) adapters, and search facilities to help you understand the effect of application changes. For more information about the new feature which provides CICS system events, see the paper 'Gaining insight into IBM CICS systems with events' in this series.

## ***Assured event emission***

Business events carry business importance to users. In many situations users can tolerate the potential for occasional loss of events, for example, because you are looking for trends and patterns over a period of time. However in situations where the user needs to respond to each and every event, it is vital to make sure that every event that is captured will be emitted successfully.

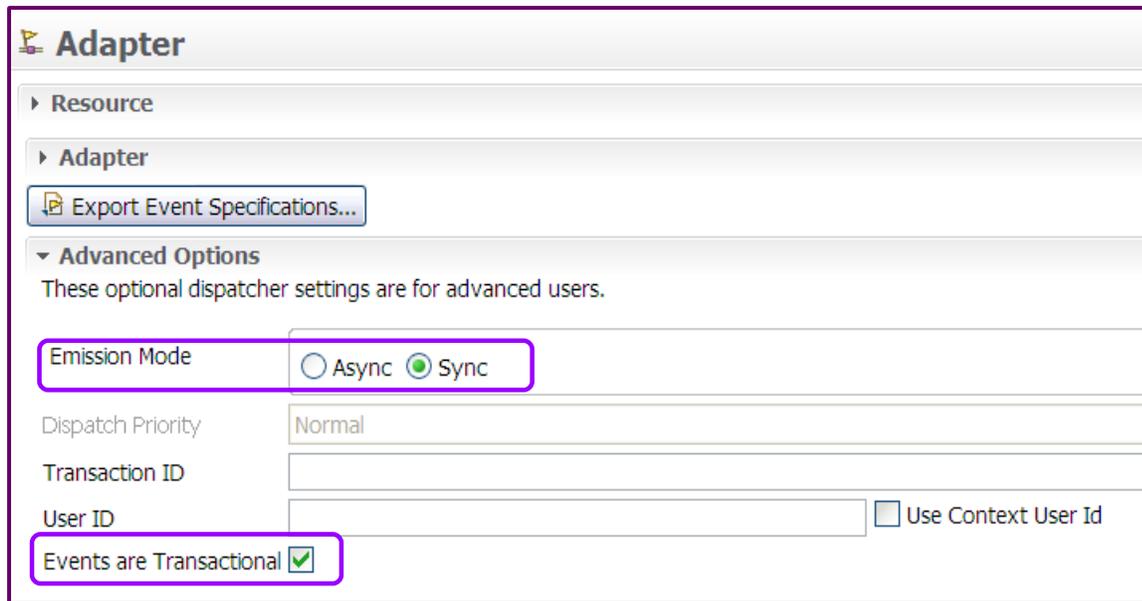
CICS TS 4.2 provides assured event emission for your event-critical business solutions. Assured event emission is achieved by selecting synchronous event emission in the CICS event binding editor.

With synchronous event emission, event formatting and emission processing is completed synchronously within the unit of work of the capturing transaction. The unit of work completes successfully only if the event is emitted. If the event emission fails, or CICS itself fails, the capturing transaction will be abended. Synchronous emission mode is used where the capturing unit of work can only be regarded as successful if it can be assured that the event has been emitted. With asynchronous event emission, the event is queued for asynchronous processing by a separate EP dispatcher thread, which reduces the overhead to the capturing transaction, but means that in those rare situations where the event fails to be emitted the capturing transaction will still complete successfully.

If a synchronous event fails to be emitted, messages are produced to explain why, statistics are updated, and the unit of work for the capturing transaction is backed out. Without assured event emission, if an event fails to be emitted CICS still provides information and statistics about the failure but the capturing unit of work continues successfully and the event is discarded.

Emission mode for the event is specified in the advanced options when defining an EP adapter using the CICS event binding editor. Figure 1 shows synchronous emission mode

being specified. You can also specify whether an event is transactional or not. For more information about emission and transactional modes, see the information on event processing adapters in the CICS TS 4.2 Information Center.



The screenshot shows the 'Adapter' configuration window. Under the 'Advanced Options' section, the 'Emission Mode' is set to 'Sync' (radio button selected). The 'Events are Transactional' checkbox is checked. Other options include 'Dispatch Priority' set to 'Normal', 'Transaction ID' (empty), 'User ID' (empty), and 'Use Context User Id' (unchecked).

Figure 1. Specifying emission mode and transactional mode in the CICS event binding editor

The following examples illustrate situations in which either synchronous or asynchronous events, would be used.

### Example for asynchronous events

One example where asynchronous events would be appropriate is when an event for bank account transactions which exceed \$1,000 is used to monitor the frequency of large transactions in a day, or to recognize high-value customers. The emission of such events is entirely separate from the application which processes the bank account transactions, which is not dependent on whether or not the events are emitted successfully.

### Example for synchronous events

Using an example of withdrawal of money from a bank account, a synchronous event could be set up so that, if the balance is lower than the amount to be withdrawn, an event is emitted immediately within the same money withdrawal transaction to trigger a balance transfer process. In this example, the event is used to drive additional processing, so if the event emission fails, the withdrawal of money should not be allowed to complete.

With assured event emission and a suitable choice of EP adapter, the user can choose the level of integrity between the business application and the emission and transportation of the event. Assuring your event emission provides the opportunity to build business critical event-based applications, and extend existing applications in as reliable a way as if you had, for example, added an MQPUT statement to an application to write an event, but without actually changing the application. The trade-off is that the synchronous processing that is essential to assuring that events are emitted might have an impact on your application response time. The synchronous event emission can also change the behavior of a transaction because an event emission failure causes the application to back out. A judicious use of synchronous event emission minimizes the application impact.

It is worth noting that all system events (as discussed in another paper in this series) are emitted asynchronously to minimize the impact to the system.

## ***Search facility***

With the growing number of event bindings being created in your enterprise, it becomes essential to be able to understand the impact of proposed changes to your CICS applications and systems on the event specified within these event bindings. Some of the questions that may get asked are "We emit events from a number of our applications. If we ever needed to make a change to any of those applications how would we know which event capture specifications are affected and might need updating?" To help you answer that question, the event processing search facility was made available in CICS TS 4.2.

The event processing search facility is a custom search that is provided in the CICS Explorer™. This facility searches event bindings and EP adapters in the CICS Explorer workspace or those installed in a CICSplex that Explorer is connected to. You can search either by resource name or by variable, structure, and copybook name of an imported language structure. The search facility will find all capture specifications which match the searched string. Based on the search result, you can decide which event bindings must change.

For example, if you need to change a program named EVPROC02, as shown in Figure 2, using the EP search facility, you can search for the string of EVPROC02 with the resource type of PROGRAM. The scope for the search is the installed event bindings in a CICSplex SDAYPEG which the CICS Explorer is connected to.

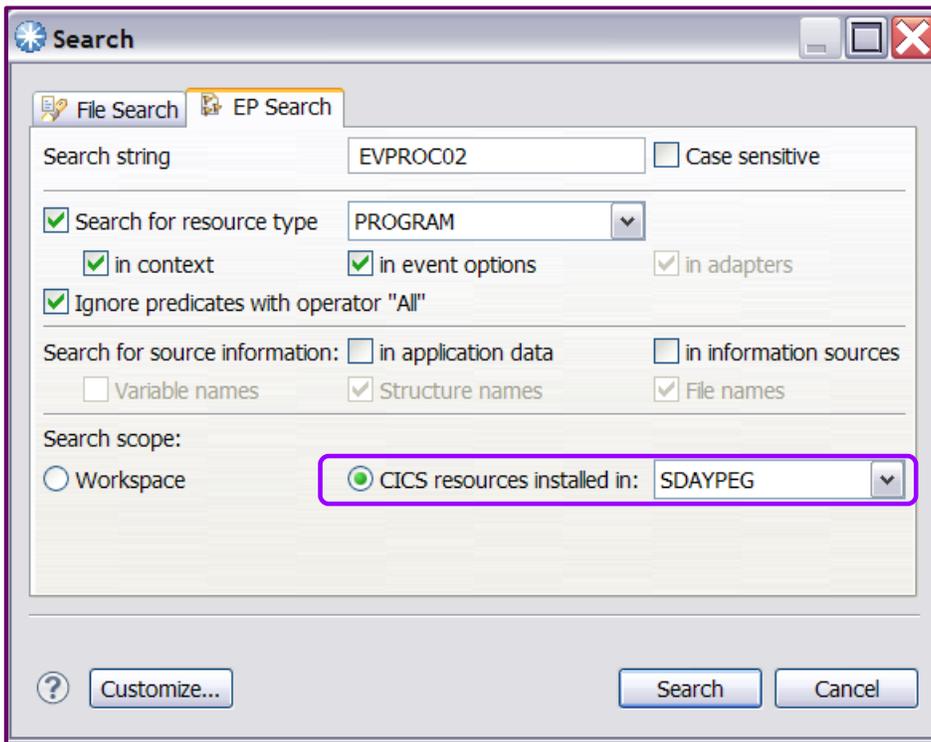


Figure 2. EP search facility

Figure 3 shows the event bindings that contain 'EVPROC02' after performing the search. Expanding each event binding, you can see the details.

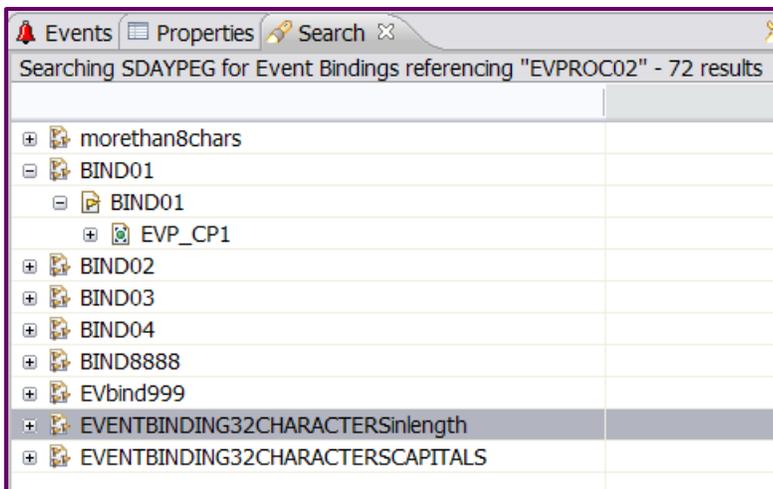


Figure 3. EP search results

## Separate EP adapters

Another new feature provided by CICS TS 4.2 is the separation of EP adapter definitions from event bindings. This makes it easier to change the configuration details for how events are to be formatted and emitted, and to reuse the same configuration details for events in multiple event bindings.

With this feature, in the CICS event binding editor, you can define a CICS Event Processing Adapter resource, which is separate from a CICS Event Binding resource, as shown in Figure 4.

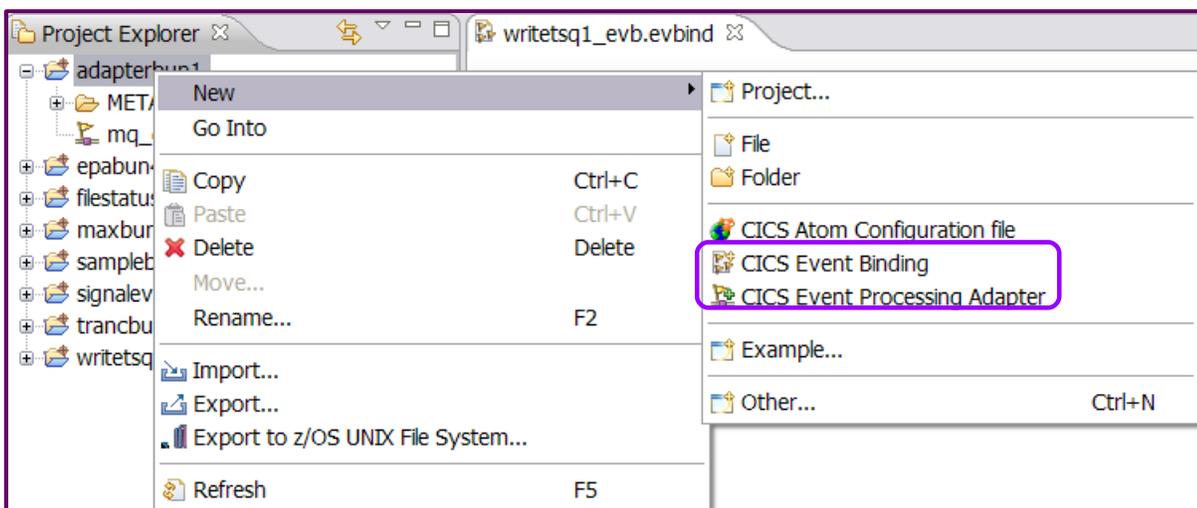


Figure 4. Separate EP adapter and event binding definition menu options

For example, you can define a WebSphere MQ® EP adapter resource, called `mq_queue` as shown in Figure 5 under the resource bundle `adapterbun1`. In an event binding `writetsq1_evb`, you can then select the option 'Use a predefined EPADAPTER resource' and provide the name of `mq_queue` as the adapter definition. The same EP adapter can be specified for as many other event bindings as required.

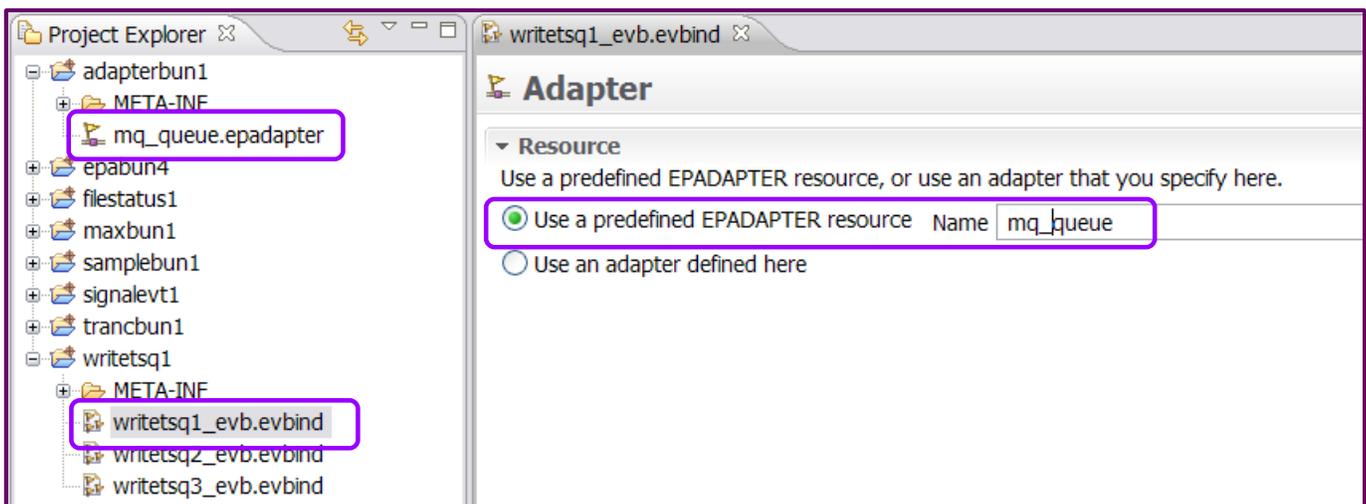


Figure 5. Choosing a separate EPADAPTER resource for an event binding

With the separate EP adapter configuration you can share the same EP adapter resource among a number of event bindings whilst having only one EP adapter resource to manage. You can also create several types of EP adapter resources, and use each of them for a specific emission and delivery requirements. For example, you can specify one WebSphere MQ EP adapter destined for WebSphere MQ queue A with emission mode of asynchronous and non-transactional, another WebSphere MQ EP adapter destined for WebSphere MQ queue B with emission mode of synchronous and transactional. According to the business requirements of the events, you can then choose the appropriate EP adapter for each event binding.

### ***Other event processing features in CICS TS 4.2***

Although we have described three of the new EP features available in CICS TS 4.2, other new features are also available.

There are a number of additional data types now supported by event processing, such as short and long floating point numbers and COBOL zoned decimal data types. For a complete list of the supported data types, see the CICS TS 4.2 Information Center.

The CICS Temporary storage (TS) queue EP adapter now supports the XML event formats of common base event (CBE), common base event REST (CBER), and WebSphere® Business Events (WBE), adding to its value in providing a way of quickly testing to ensure that the right events are being produced.

The EXEC CICS INQUIRE CAPTURESPEC command provides new options for system programmers and developers to determine information about any primary predicate or application context filters. The new EXEC CICS INQUIRE CAPOPTPRED, CAPDATAPRED and CAPINFOSRCE commands can be used to determine information about application command options, application data predicates, and information sources that are specified in a given capture specification.

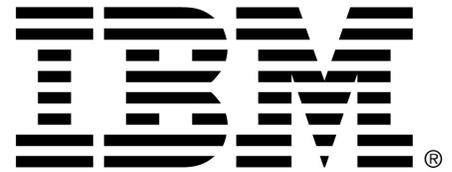
You can now define capture specifications to emit events from the file and temporary storage commands issued by the CICS Atom support and the EXEC CICS LINK commands that are issued by the CICS-WebSphere MQ bridge program.

## ***Conclusion***

In conclusion, CICS TS 4.2 enhances event processing support through both runtime offerings and improved tooling. All of these enable you to make the best use of event processing and move towards event-driven business in a fast changing world.

## ***Further reading***

1. "Gaining insight into IBM CICS systems with events", IBM, July 2011:  
<http://www.ibm.com/software/htp/cics/tserver/v42/library/index5.html>
2. Event processing enhancements in CICS TS 4.2:  
[http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.whatsnew.doc/ep/dfhe4\\_overview.html](http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.whatsnew.doc/ep/dfhe4_overview.html)
3. Event processing adapters:  
[http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.eventprocessing.doc/concepts/dfhep\\_event\\_processing\\_adapters.html](http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.eventprocessing.doc/concepts/dfhep_event_processing_adapters.html)
4. Supported data types:  
[http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.eventprocessing.doc/concepts/dfhep\\_event\\_editor\\_informationsourcstab.html](http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.eventprocessing.doc/concepts/dfhep_event_editor_informationsourcstab.html)



© Copyright IBM Corporation 2011

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. IBM, the IBM logo, ibm.com, CICS, CICS Explorer, WebSphere, WebSphere MQ, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. 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). All statements regarding IBM plans, directions, and intent are subject to change or withdrawal without notice.