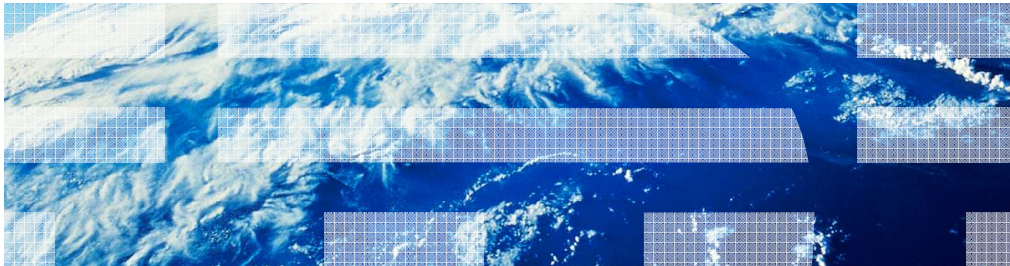

CICS Transaction Server for z/OS V4.2

Event processing

Synchronous event emission



This module provides information about synchronous event emission and how it is supported in CICS Transaction Server V4.2 event processing.

Agenda

- **Overview of event emission modes**
- **When to use synchronous event emission**
- **How to configure synchronous event emission**
- **Considerations and limitations**

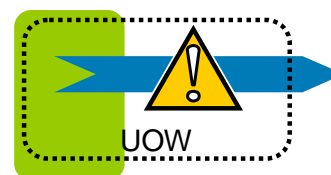
Synchronous event emission is a new event emission mode in CICS V4.2 event processing.

This module gives an overview of the event emission modes available in V4.2 before looking more closely at synchronous event emission.

It explains when it should be used, how to configure it, and any considerations and limitations that should be taken into account.

Overview of event emission modes

- Asynchronous
 - After capture, events are formatted and emitted asynchronously to the capturing application's unit of work
 - Failure to format or emit events does not affect the capturing application.
 - All events in CICS TS V4.1 are emitted this way
- Synchronous
 - Event formatting and emission is performed as part of the capturing unit of work
 - If the event fails to be emitted, the capturing units of work are backed out at sync point and its transaction is abended with ASP7
 - New to CICS TS V4.2



CICS V4.2 event processing provides two event emission modes, asynchronous and synchronous.

Asynchronous emission is the default and is how all events in CICS V4.1 were emitted.

When using asynchronous emission, events are formatted and emitted asynchronously to the capturing application's unit of work.

This has the least impact on the application's response time and means that a failure to emit the event does not affect the capturing application in anyway.

Synchronous emission has been introduced in CICS V4.2.

When using synchronous emission, events are formatted and emitted within the capturing application's unit of work.

This means that the capturing application is now dependant on the successful emission of the event. If it fails, the capturing application is backed out at sync point and it's transaction abended with an ASP7 abend.

When to use synchronous event emission

- When using synchronous emission event emission is assured

- Use synchronous event emission when:
 - *The capturing application is not considered to be successful if the event is not emitted (for example: in event driven processes)*
 - *You want the capturing application to be dependant on the successful emission of the event.*

When using synchronous emission, event emission is assured. Events cannot be lost due to emission failures. With asynchronous emission, the failure of a synchronous event emission causes the unit of work that caused the event to be backed out, effectively, making it as if the event never happened.

You should use synchronous event emission when events are to be considered part of the application from which they were captured.

Typically this means that the events are being used for more than just information or data gathering, they are actively being used to drive further processing that is critical to the business.

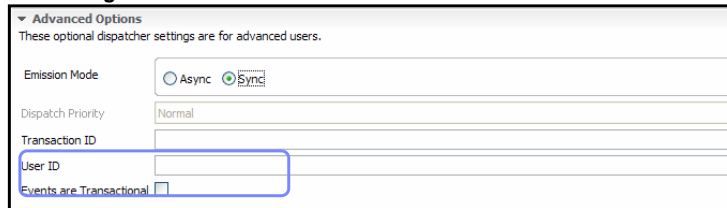
For example, in an order processing system, an event can be captured every time an order is placed in the system.

If the event was being used to track how many orders the business was getting over a period of time then synchronous emission is not necessary. The order can still be placed and processed, even in the rare case that the event fails to emit. The only consequence is that the company's statistics can be inaccurate by one or two orders, due to lost events. In this case asynchronous emission should be used.

However, if the order placed event was being used to drive the dispatch processing for that order, then the order cannot be placed without successfully emitting the event, or it will not be dispatched. In this case, the event is critical to the business and so synchronous emission should be used to assure event emission.

How to configure synchronous event emission

- Select the sync emission mode in the advanced options section of the EP adapter tab in the event binding editor



The screenshot shows the 'Advanced Options' section of the EP adapter tab. It contains the following fields:

- Emission Mode:** Radio buttons for 'Async' and 'Sync'. The 'Sync' button is selected.
- Dispatch Priority:** A dropdown menu set to 'Normal'.
- Transaction ID:** A text input field.
- User ID:** A text input field.
- Events are Transactional:** A checkbox that is currently unchecked.

- All events emitted using this adapter will use synchronous emission

The emission mode is a property of an EP adapter.

The emission mode is selected in the advanced options section of the EP adapter tab in the event binding editor. By default, this is set to asynchronous.

Events are emitted using the emission mode of the EP adapter specified in their event binding.

Considerations and limitations

- **Not available for the transaction start EP adapter**
- **Not available for system event capture points**
- **Greater performance impact on capturing application.**

There are some considerations and limitations to take into account when deciding to use synchronous emission.

Firstly, synchronous emission is not available for use with the transaction start EP adapter or with events captured from system event capture points.

Secondly, synchronous emission has a greater impact on capturing application response time, when compared to asynchronous emission. This is due to the formatting and emission of the event being performed on the application thread.

Considerations and limitations

- When used with transactional TRANSMODE, event transport must be recoverable
 - *Supported EP Adapters: WMQ, TSQ, Custom**

- When used with non-transactional TRANSMODE, transport must be non-recoverable
 - *Supported EP Adapters: WMQ, HTTP, TSQ, Custom**

*where all actions are recoverable if the EPAP_RECOVER flag is set (and not recoverable if not set) in the DFHEP.ADAPTPARM container

Due to the way synchronous emission supports event transactionality, the transports used must be recoverable if transactional is used, and non-recoverable, if non-transactional is used. Since HTTP is a non-recoverable transport, the HTTP EP adapter cannot be used when events are required to be both synchronous and transactional.

TSQ recoverability can be configured through the use of a TSMODEL resource. This allows a synchronous TSQ EP adapter to be used with both transactional and non-transactional, providing its target TSQ has been configured with the correct recoverability.

Custom EP adapter programs should honor the EPAP_RECOVER flag in the DFHEP.ADAPTPARM container, which tells them whether their actions need to be recoverable or not.

Summary

- CICS TS V4.2 introduces synchronous event emission
- Allows the applications to be dependant on the events they emit
- Has more impact on capturing application performance

In Summary, synchronous event emission is new to CICS V4.2.

It provides a way for applications to be dependant on the successful emission of the events captured from them.

Although synchronous event emission has more impact on the capturing application's performance, it provides a way to assured event emission for those events which are used to drive business critical processes.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback about IEA CICS synchronous events.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20IEA%20CICS%20synchronous%20events.ppt)

This module is also available in PDF format at: [../IEA_CICS_synchronous_events.pdf](..../IEA_CICS_synchronous_events.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, CICS, 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 other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.