CICS Event Processing

Ian J Mitchell,
IBM Distinguished Engineer

Thursday, March 3\textsuperscript{rd} 2011
Session 8278
Topics

- CICS and Event Processing
  - Introduction to event processing
  - CICS event processing overview, and Smarter Planet
  - Value from CICS events: a few example scenarios
- Building an Event Binding to enhance the Catalogue Example Application
  - Using the Event Binding Editor in CICS Explorer
- Deploying an Event Binding
- Managing a live Event Binding
This session provides a brief overview of the Event Processing capability introduced in CICS TS V4.1. The session will dive into a practical demonstration of the steps to create, deploy and manage event specifications in IBM CICS Transaction Server V4.1 and how to make those events available to consumers such as IBM WebSphere Business Monitor and IBM WebSphere Business Events. The session will include a demonstration of creating an event specification within an event binding using the Event Binding Editor (part of the CICS Explorer), explaining the various options and aspects of the specifications, as well as showing deployment and testing of an event binding.
CICS and Event Processing

An introduction to CICS as a source of business events
This section introduces the core concepts of CICS event processing support.
What is an event?

- An event is
  - Anything that happens (or is contemplated as happening)
  - An event has a name and usually some data (its payload)
  - Produced and responded to asynchronously
- “Simple” event
  - A single event, meaningful in itself
  - e.g. order placement; bank account update; stock trade
- “Complex event processing”
  - Detect and respond to patterns of events
  - e.g. three orders from customer A in 2 days; bank withdrawal after PIN change update; interesting pattern of stock trades
- Business Event Processing
  - Detect and respond to events that indicate business-impacting situations across the enterprise
  - Extends event processing capabilities to business users
    - e.g. IBM WebSphere Business Events provides complex event processing for business users
What is an event? – Notes

- This slide introduces the idea of an event, which is really simply something that happens. The absence of that thing happening can also be an event, but so-called ‘complex event processing’ (see below) is generally required to detect this. The definition on the slide is taken from the Event Processing Technical Society Glossary (available at http://www.ep-ts.com).

- In contrast to just sending messages, one particular characteristic of an event is that it is a named. The data associated with an event is sometimes referred to as its payload.

- Events are processed asynchronously from the emitting application, with the consumption of the event being decoupled from its originator.

- This slide also explains the distinction between a ‘simple’ event, and ‘complex event processing’, the latter being based on a pattern of simple events potentially occurring over time, and correlated together in some way. Although this is a useful distinction in the context of the type of events that CICS emits, which are simple events, these terms are less widely used than they have been.

- A business event is something that happens which is relevant to the business (this is ‘business’ in its broadest sense, not just commercial businesses). This effectively means that all events are really business events, but as we shall see, the focus for CICS events is on application events as opposed to system or “IT” events.

- CICS events aim to provide information at the level of what the application is achieving, rather than what the code or system is doing.
Event Processing in a Nutshell

Event Producer generates events. Event Processing performs operations on events. Event Consumer consumes and reacts to either raw or complex events.
Event Processing involves three main aspects:

- Event producers emit events into the event processing system. These can be simple RFID sensors and actuators, through to business flows or CICS applications. The event processor can carry out a variety of actions, ranging from simple enriching of the event in some way (e.g. adding a date and timestamp, or adding information about the source of the event), through to comparing multiple events (potentially from multiple sources) against event patterns and producing a new ‘derived’ event.

- The event resulting from processing is available for consumption.

- The event consumer will react to the event. The event consumer might be very simple and just update a database, or could populate a visual dashboard with the data carried with the event, or carry out new business processing as a result of the event. The event consumer could also carry out event processing itself.
CICS and Business Events

- Event processing addresses the need for agility
  - Modern businesses must react quickly to circumstances
  - Decision makers need reliable, timely information
- CICS systems run an enormous amount of existing business logic
- Using an Event-based approach, there is potential to gain insight into the processing in CICS and to introduce additional extensions to applications
  - In a dynamic, de-coupled fashion
  - Without the need to change the applications
- **CICS Transaction Server for z/OS V4.1** allows you to emit business events from existing applications
  - Supporting shifting corporate policies
  - Without having to modify the applications
  - And driving your choice of destination
    - WebSphere Business Monitor, WebSphere Business Events, CICS application, application through WebSphere MQ, ...
CICS and Business Events - Notes

• Events are valuable to Enterprise Systems, providing the ability to respond in real-time, or near real-time.

• Given the considerable amount of business processing which is carried out in CICS systems across the world (over 30 billion transactions a day), CICS is a very significant source of business events. This can provide enhanced business flexibility and the ability to meet governance and compliance regulations.

• Event emission is asynchronous to the emitting application, and the consumption of the event is decoupled from its originator.

• CICS TS will emit simple, single events. These may be consumed by a “complex event processing” engine where they can be combined with events from other sources in addition to CICS. They can be sent to a Business Monitor to provide insight into processing within CICS.
CICS TS V4.1 is aimed at helping users to

**Compete** for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

- **Business Flexibility and Innovation**

**Comply** with corporate, industry and government policies to manage business risk of critical business applications

- **Governance and Compliance**

**Control costs** by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

- **IT Simplification**
This slide shows the overall themes of the CICS TS V4.1 release.

CICS support for event processing helps to provide for increased business flexibility and innovation, as well as assisting with governance and compliance.

To a lesser extent, it also plays into the theme of IT Simplification, by making it easy to create events, and to enable and disable them.
CICS and event processing – overview

Development & Deployment Tools

Existing Business Logic

- Code NOT changed

CICS Transaction Server for z/OS

CICS Event Processing

- Event Capture
- Filtering
- Enrichment
- Formatting
- Routing

Extensible

- Secured
- Monitored
- etc.

WebSphere Business Events

WebSphere Business Monitor

Other Event Consumers
CICS Event Processing Overview – Notes

- This gives a high-level summary of CICS as a source of business events.
- CICS event processing support allows existing business logic to be instrumented to emit events without change to the application code.
- Tooling is used to define events and their data, to specify to the CICS runtime how to detect when the events occur, to indicate how they are to be formatted and routed, and to deploy the events to CICS.
- The CICS runtime will detect occurrences of events which are currently enabled, and capture the events without the need to make application code changes – enabling rapid, easy deployment of event-based solutions.
- CICS Event Processing is a core component of the CICS runtime, and will provide all the qualities of service you would expect of CICS. When CICS captures events, it will carry out specified filtering, enrich the event with information about the application context in which it occurred, format the event and route it such that it can be consumed by the appropriate event consumer.
- It is possible to emit events in formats suitable for consumption by WebSphere Business Events, WebSphere Business Monitor, and other consumers.
- CICS Event Processing support is extensible, with options for customization.
Smarter Planet

Instrumented, interconnected, and intelligent
This slide is a reminder that the Smarter Planet is instrumented, interconnected and intelligent.
CICS events for a Smarter Planet

Existing Business Logic

Code NOT changed

CICS Transaction Server for z/OS

Development & Deployment Tools

Instrumented

CICS Event Processing

Event Capture
Filtering
Enrichment
Formatting
Routing

Intelligent
Extensible
Secured
Monitored etc.

WebSphere Business Events

Instrumented

Intelligent

WebSphere Business Monitor

Interconnected

Other Event Consumers

Captured

Events

Instrumented

SHARE in Anaheim 2011
CICS events for a Smarter Planet – Notes

- This shows that CICS events are:
  - **Instrumented**: the tooling allows CICS applications to be ‘instrumented’ to emit events, without changing the applications
  - **Interconnected**: CICS events integrate with other event processing products and event consumers
  - **Intelligent**: CICS events can be filtered to capture the significant events, enriched with application context, formatted and routed to consumers. IBM event consumers provide intelligence such as the detection of event patterns
Value from CICS events
Value from CICS events – Notes

- This section shows some of the ways in which value can be obtained from using CICS events.
EP Scenario 1 – Observe business processing

- Monitor processing carried out by an application
  - e.g. Identify key points in order processing business logic
    - order requested, order placed, order confirmed, order dispatched, order cancelled
    - Collect relevant contextual data associated with the event, including a way to correlate events for the same order, and emit event
  - Events sent e.g. to WebSphere Business Monitor
    - Observe orders being received, processed, cancelled
    - Study KPIs – numbers of orders received per week, time to process and dispatch orders, etc.
    - Take action when thresholds exceeded, when value of a customer’s orders exceeds a certain amount, etc.

- Original application continues processing independently:
  - Event instrumentation is ‘non-invasive’ to the application
EP Scenario 1 – Observe business processing – Notes

- This is a simple instrumentation example, based on an order processing system:
  - Event capture points are defined at the key points of the business application.
  - Triggered events can update a business dashboard with both notification that the event occurred and data relevant to the event (order size, customer number etc.).
  - This can be used to observe the processing, and to see KPIs and alerts.

- In this simple example the application could also be extended by manual or automatic action taken when thresholds on the dashboard are exceeded.

- The original application continues processing without change.
EP Scenario 1 – KPIs and Dimensions in Business

Instances

<table>
<thead>
<tr>
<th>orderID</th>
<th>itemID</th>
<th>customer name</th>
<th>item price</th>
<th>item quantity</th>
<th>order duration</th>
<th>order start time</th>
<th>order end time</th>
<th>order price</th>
<th>order status</th>
</tr>
</thead>
<tbody>
<tr>
<td>+000012784 0001</td>
<td>Steve</td>
<td>20</td>
<td>20</td>
<td>24 m, 19.121 s</td>
<td>May 27, 2009 10:38:13 AM</td>
<td>May 27, 2009 11:02:32 AM</td>
<td>400</td>
<td>order canceled</td>
<td></td>
</tr>
<tr>
<td>+000012785 0002</td>
<td>Lijia</td>
<td>5</td>
<td>40</td>
<td>23 m, 25.009 s</td>
<td>May 27, 2009 10:39:24 AM</td>
<td>May 27, 2009 11:02:49 AM</td>
<td>200</td>
<td>order shipped</td>
<td></td>
</tr>
</tbody>
</table>

KPIs

<table>
<thead>
<tr>
<th>KPI Name</th>
<th>Status</th>
<th>Value</th>
<th>Target</th>
<th>Actions</th>
<th>Value in Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>average time on order shipped</td>
<td></td>
<td>23 m, 18.601 s</td>
<td>1 h, 0 m, 0 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>percent of order cancelled</td>
<td></td>
<td>25.00%</td>
<td>30.00%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dimensions

Drill Down

- Measures
  - average order price
  - sum order price
  - order count

- order status dimension
  - average order price
  - sum order price
  - order count

- All order status dimension
  - order canceled
  - order shipped

- order status dimension
  - average order price
  - sum order price
  - order count
EP Scenario 1 – KPIs and Dimensions in Business Space – Notes

• This example of dashboard views in Business Space shows how business process performance and KPIs can be monitored using events from CICS. It shows information about orders shipped and cancelled, and KPIs for the average time taken to ship an order, and the percentage of orders which get cancelled.
EP Scenario 2 – Non-invasive change to business processing

- Application extended by triggering new or existing separate program for extra, asynchronous processing

- Examples:
  - Extend governance practices by sending an alert when certain data is viewed or altered
  - Asynchronously send details of special offers or discounts when large customer orders have been placed
    - This example may be seasonal and is easily enabled/disabled without application change. The interpretation of a large order can be changed outside the application.

Original program continues processing independently

- Consumer program can run within CICS or externally
  - Flexibility to use available skills and other resources
  - Choice of processing platform depends on nature of processing, interaction with other subsystems
In this example the business application can be changed or enhanced by event processing

- Passing data relevant to the context
- External processing could be
  - Similar to base application function (extending business function)
  - Different kind of processing – typically observation (tracking activity for business or audit reasons)
  - Different processing under different conditions or times (e.g. Tue-Thu)
- Can make use of different platform, skills, tools

Application code initiated by the triggered event may be a program running within CICS or may be initiated on another system via a WebSphere MQ message or as the result of the action of a complex event processing engine.
EP Scenario 3 – Event Combination

- Respond to patterns of events
  - Track banking transactions exceeding $5000
  - Events captured from the CICS applications, generating events for large transactions via
    - *Cash deposit, cash withdrawal, account transfer (in or out), check deposit*
    - *Events sent to WebSphere Business Events*
  - Customer entering branch detected via RFID in customer’s bank card, event sent to WebSphere Business Events
  - Detect pattern of customer who has performed 5 or more large transactions then visits the branch

- Specify action to take in WebSphere Business Events e.g. alert the branch manager
This is an example of “complex” event processing with events being potentially combined from multiple sources including CICS.

A complex event processing engine (such as WebSphere Business Events) is able to collate events from multiple sources and carry out pattern matching to derive additional insight.
EP
Scenario 3

Filter: RevitalizedBank recent significant account activity

And

- Past Occurrences Of Ack_LargeTransaction Within 1 month Is Greater Than or Equal To 5
- Past Occurrences Of Inform Branch Manager Within 1 month 5 mins Is 3

Interaction Set: Customer of Interest
In response to: Customer Enters the Branch (RevitalizedBank Branch)

If: RevitalizedBank recent significant account activity
Then: Immediately Inform Branch Manager (RevitalizedBank Email Server)
Else: no action
EP Scenario 3 – Detecting event pattern using WebSphere Business Events – Notes

- Events from CICS can be used in interaction sets and conditions (filters) defined in WebSphere Business Events
- This slide illustrates EP scenario 3 by showing an interaction set and filters defined in the WebSphere Business Events Design tool.
About the demo...
What we do in the demo

• Explore the Event Binding Editor
  • Build a new Event Binding using the supplied example as a starting point

• Deploy the binding in a bundle
  • Well, sort of...

• See the Event Binding enable the extension of the Catalog Example App
  • Integrate the 3270 app with a new Dynamic Scripting App
  • Control the Event Binding via the CICS Explorer
Topology of the demo app

Order
Call-centre

CICS
SMASHAD1
Catalog Example App
EXMPCAT

3270
New Business Requirement

- Warn the procurement department when stock falls below 24 units and none are on order.
  - Unlike the Order Call Centre, Procurement do not have 3270 emulators and need a browser interface
  - Initial target is to evaluate the efficiency of the process improvement THIS WEEK
Extending the demo app with Events

Order Call-centre

CICS
SMASHAD1
Catalog App
Event Binding
Event Adaptor

CICS
JVAOR123
Dynamic Scripting

3270
HTTP
EXMPCAT

Procurement
Managing the demo app

SMASHPLX

CMCI

CICS Explorer

Order

Call-centre

SMASHAD1

3270

EXMPCAT

JVAOR123

Dynamic Scripting

Procurement

HTTP

HTTP

Catalog App

Event Binding

Event Adaptor
Building an Event Binding
Step 1

```
SMASHPLX

CICSPlex/SM
CMCI
CICS Explorer

3270
Event Binding
EXMPCAT

CICS
JVAOR123
Dynamic Scripting
HTTP

SHARE 2011
```

RAW TEXT:

```
Step 1

SMASHPLX

CICSPlex/SM
CMCI
CICS Explorer

3270
Event Binding
EXMPCAT

CICS
JVAOR123
Dynamic Scripting
HTTP
```
Use the example binding

- Fire up CICS Explorer
  - (This demo applies to version 1.0.0.7 of the Explorer – details might vary with other versions)
- Switch to the “Resource” perspective
- From the menubar... Explorer ➔ New Wizards ➔ CICS Bundle Project
- Use New ➔ Example... from the Bundle Project's context menu
- You now have a Bundle containing the Example event-binding.
A new bundle with the example event binding

Description: Several catalog items have been out of stock

User Tag: ebrefo01

Event Specifications: Catalog_stock_status_check
The Event Binding Editor

• Become familiar with the Event Binding Editor
• Notice the Tabs for the three major elements...
  • Event Binding – which contains a list of Specifications
  • Specification (multiple per binding) - edits the selected Specification in the binding
  • Adapter – there is one per binding
• Adapter needs changing for the demo, use
  • Custom Adapter type
  • Tranid ADAP
  • Pass string “DSAPP” to the adapter – it’s the name of the URI map which has the host to post the event to
Event Binding Editor

**General Information**

- **Description**: Several catalog items have been out of stock before being re-ordered causing missed sales opportunities. Generate an event when an item is about to go out of stock.

- **User Tag**: ebrief001

**Event Specifications**

- Catalog_stock_status_check

---

**Event Binding**

**Specification**

**Adapter**
Event Specifications

Identify and describe the event.

**Name**: Catalog_stock_status_check

**Description**: The stock level is low and there is no re-order in place.

### Emitted Business Information

Describe and order the business information to be emitted by the event.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Length</th>
<th>Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program_name</td>
<td>Text</td>
<td>8</td>
<td>0</td>
<td>Program Name</td>
</tr>
<tr>
<td>Item_ref</td>
<td>Numeri</td>
<td>4</td>
<td>0</td>
<td>Item reference number</td>
</tr>
<tr>
<td>Item_description</td>
<td>Text</td>
<td>40</td>
<td>0</td>
<td>Item description in catalog</td>
</tr>
<tr>
<td>in_stock</td>
<td>Numeri</td>
<td>4</td>
<td>0</td>
<td>Current number of items in stock</td>
</tr>
<tr>
<td>on_order</td>
<td>Numeri</td>
<td>3</td>
<td>0</td>
<td>Number of items on order</td>
</tr>
</tbody>
</table>

### Capture Specifications

Add Capture Specifications to this event.

- Add a Capture Specification...

### Automatic Capture Specification

Use this to automatically generate a capture specification for a signal event call using the business information entered above.

- Add an Automatic Capture Specification...
Event Adapter – for a TSQ

Choose the adapter to emit events produced by this binding.

Adapter: TS Queue

Emits events to a named CICS TS queue. Use this EP adapter to validate that the correct events are being captured with the correct data and reads from a TS queue.

Queue Name: CATMAN

System ID (Optional)

Use Auxiliary Temporary Storage

Export Event Specifications...

Advanced Options

These optional dispatcher settings are for advanced users.

Dispatch Priority: Normal

Transaction ID

User ID

Use Context User ID

Events are Transactional

Event Binding Specification Adapter
Change the Adapter for the demo

Choose the adapter to emit events produced by this binding.

- **Adapter**: Custom (User Written)

  Emits events in any format that you require. A custom EP adapter is a CICS program that you write to provide a combination of formatting and routing of an event that is not supported by the CICS-supplied EP adapters. The custom EP adapter must not carry out any other processing, such as consumption of the event.

- **Transaction ID**: ADAP

- **Data passed to the Custom Adapter**: DSAPP

**Advanced Options**

These optional dispatcher settings are for advanced users.

- **Dispatch Priority**: Normal
- **Transaction ID**
- **User ID**
- **Events are Transactional**: 🗿
- **Use Context User Id**: ✅
Back to specifications

- There can be multiple per binding.
- Select the specification to work with from the list on the first tab (“Event Binding”)
- In this case “Catalog_stock_status_check”
- An Event Specification has four elements
  - General – name and description
  - Emitted Business Information – the data that will be collected and sent with the event
  - (Explicit) Capture Specifications
  - Automatic Capture Specifications
Specification elements

```
**General**
Identify and describe the event.

Name: Catalog_stock_status_check

Description: The stock level is low and there is no re-order in place.

**Emitted Business Information**
Describe and order the business information to be emitted by the event.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Length</th>
<th>Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program_name</td>
<td>Text</td>
<td>8</td>
<td>0</td>
<td>Program Name</td>
</tr>
<tr>
<td>Item_ref</td>
<td>Number</td>
<td>4</td>
<td>0</td>
<td>Item reference number</td>
</tr>
<tr>
<td>Item_description</td>
<td>Text</td>
<td>40</td>
<td>0</td>
<td>Item description in catalog</td>
</tr>
<tr>
<td>in_stock</td>
<td>Number</td>
<td>4</td>
<td>0</td>
<td>Current number of items in stock</td>
</tr>
<tr>
<td>on_order</td>
<td>Number</td>
<td>3</td>
<td>0</td>
<td>Number of items on order</td>
</tr>
</tbody>
</table>

**Capture Specifications**
Add Capture Specifications to this event.

- Add a Capture Specification...

**Automatic Capture Specification**
Use this to automatically generate a capture specification for a signal event call using the business information entered above.

- Add an Automatic Capture Specification...
```
A capture point specification

• The example has one capture point called “Check_stock_status_on_rewrite”
  • The capture point chosen is the EXEC CICS REWRITE command
  • The capture point selection field list all the possible capture points.
• The filter tab defines predicates applied at runtime
• The information sources tab defines where the required business data for the event can be found
A capture specification

General
Identify and describe the capture specification.
Name: Check_stock_status_on_rewrite
Description: The number in stock and number on order are available in the FROM parameter of the REWRITE command.

Capture Point
Choose the capture point.
Capture Point: REWRITE

Next: Filtering ->
Filtering predicates in a capture specification

• The Filtering tab allows predicates to be defined which must match at runtime for the event to be 'interesting'.

• These apply to three types of information:
  • Context - Tranid, current program, userid, response code
  • Command options – specific to each command chosen as the capture point. In this example we can match on the filename
  • Application data – if the command has application data (commarea, container, record etc etc) then predicates can match on fields in that data
    • Here's where the stock number and order status predicates satisfy the business requirement.
      • The Source column in the table refers to the command's keyword which holds the application data – eg the FROM keyword on the REWRITE in this case
      • The row order in the table specifies the order of evaluating the predicates
Filter predicates in the editor

![Screenshot of a user interface showing filter predicates in an editor. The interface includes options for specifying application context, command options, and data filters.]
Creating predicate on application data

- An application data predicate requires knowledge of the application data structure
  - Type, offset and length
  - Can be imported from a language structure (eg from a copybook)
  - Or manually entered
The copybook importer

*****************************************************************
*                           *                                    *
* CONTROL BLOCK NAME = DFH0XCAT                                  *
*                                                               *
* FUNCTION =                                                    *
*    This copybook describes the catalog manager file           *
*    record structure.                                          *
*                                                               *
*****************************************************************

* Catalogue record structure
01 WS-CAT-RECORD.
   03 WS-CAT-ITEM.
      05 WS-ITEM-REF      PIC 9(4).
      05 WS-DESCRIPTION   PIC X(40).
      05 WS-DEPARTMENT    PIC 9(3).
      05 WS-COST          PIC 9(6).
      05 WS-IN-STOCK      PIC 9(4).
      05 WS-ON-ORDER      PIC 9(3).
      05 FILLER           PIC X(20).
Defining information sources

• The Event Specification defines what needs to be emitted with the event.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Length</th>
<th>Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program name</td>
<td>Text</td>
<td>8</td>
<td>0</td>
<td>Program Name</td>
</tr>
<tr>
<td>Item_ref</td>
<td>Numeri</td>
<td>4</td>
<td>0</td>
<td>Item reference number</td>
</tr>
<tr>
<td>Item_description</td>
<td>Text</td>
<td>40</td>
<td>0</td>
<td>Item description in catalog</td>
</tr>
<tr>
<td>in_stock</td>
<td>Numeri</td>
<td>4</td>
<td>0</td>
<td>Current number of items in stock</td>
</tr>
<tr>
<td>on_order</td>
<td>Numeri</td>
<td>3</td>
<td>0</td>
<td>Number of items on order</td>
</tr>
</tbody>
</table>

• The Information Sources tab enables you to specify where that information comes from (mainly the offset)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Format Len</th>
<th>Source</th>
<th>Contain</th>
<th>Offse</th>
<th>Capture Len</th>
<th>Capture Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>Text</td>
<td>8</td>
<td>PROG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item_ref</td>
<td>Numeri</td>
<td>4</td>
<td>FROM</td>
<td>0</td>
<td>4</td>
<td></td>
<td>Character</td>
</tr>
<tr>
<td>Item_description</td>
<td>Text</td>
<td>40</td>
<td>FROM</td>
<td>4</td>
<td>40</td>
<td></td>
<td>Character</td>
</tr>
<tr>
<td>in_stock</td>
<td>Numeri</td>
<td>4</td>
<td>FROM</td>
<td>53</td>
<td>4</td>
<td></td>
<td>Character</td>
</tr>
<tr>
<td>on_order</td>
<td>Numeri</td>
<td>3</td>
<td>FROM</td>
<td>57</td>
<td>3</td>
<td></td>
<td>Character</td>
</tr>
</tbody>
</table>
Editing Information Source data

• Again, for information obtained from Application Data the type, offset and length can be manually entered or imported from a language structure.
That's a complete Event Binding

- We have defined:
  - The business objective of the events in the binding
  - The information to be collected and sent with the event
  - The, possibly various, places the event can be captured
    - The command(s) to look at
    - The runtime conditions that mean the event is interesting
    - Where to get the interesting information from
      - Via type, offset and length possibly importing a language structure
  - The type of adapter to use to handle the event to deliver it to the intended consumer
Now for Deployment
Running the enhanced system
Orders come in...

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Cost</th>
<th>Stock</th>
<th>On Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010</td>
<td>Ball Pens Black</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0020</td>
<td>Ball Pens Blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0030</td>
<td>Ball Pens Red 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0040</td>
<td>Ball Pens Green</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0050</td>
<td>Pencil with eraser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0060</td>
<td>Highlighters As</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0070</td>
<td>Laser Paper 28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0080</td>
<td>Laser Paper 28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0090</td>
<td>Blue Laser Paper</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>Green Laser Paper</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>IBM Network Printer 24 - Toner cart</td>
<td>169.56</td>
<td>0006</td>
<td>000</td>
</tr>
<tr>
<td>0120</td>
<td>Standard Diary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0130</td>
<td>Wall Planner: Executive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0140</td>
<td>70 Sheet Hard Brochure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0150</td>
<td>Sticky Notes 3x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Order Quantity: ___
User Name: ________
Charge Dept: ________

F3=EXIT   F12=CANCEL
The event is delivered to the monitor
Summary, References and Q&A

Including a summarising scenario
Summary and Q&A – Notes

• The presentation concludes with a summarizing scenario and a summary of the main points of the presentation.
Visibility, Compliance, and Business Flexibility with CICS Events

CICS TS

Stock Trade Events

Stock Trading Application

WebSphere Business Events

WebSphere Business Monitor

CICS Events help you to
• Observe business processes
• Recognize suspicious activity
• Drive new processing
Visibility, Compliance, and Business Flexibility – Notes

• This slide pulls together the ideas from the presentation.

• CICS support for Event Processing will enable CICS as a source of events, allowing visibility into the business processes running in CICS, supporting governance, compliance and fraud detection, and providing increased business flexibility.

• The scenario on the slide shows that events emitted by a stock trading application running in CICS TS could be used to:
  • Observe the trading behaviour; for example, by displaying KPIs on a dashboard, such as WebSphere Business Monitor.
  • Spot suspicious trading activity by detecting particular patterns of events using WebSphere Business Events, and take action (which could include sending events on to WebSphere Business Monitor).
  • Drive new processing, perhaps in response to a particular type of trade, or (via WebSphere Business Events) in response to a particular pattern of trades. The new processing can be introduced to the overall application in a flexible and dynamic way without the need for long development cycles.
References for CICS Event Processing Support

- CICS TS V4.1 Announcement Letter
- CICS TS V4.1 Information Center
- CB11: CICS Events for WBE
- CICS Event Processing on YouTube
  - CICS Events with WebSphere Business Events High-level
  - CICS Events 5 minute demo
- CICS and Events white papers
  - IBM event processing solutions (CB11 introduction)
  - Gaining insight with IBM CICS and business events
- Redbook: Implementing Event Processing with CICS
- WebSphere Business Monitor
  - WBM Introduction
  - WBM V7.0 InfoCenter
- WebSphere Business Events
  - WebSphere Business Events Introduction
  - WebSphere Business Events V7.0 InfoCenter
- Smarter Banking with CICS Transaction Server Redbook
References – Notes

Some references for CICS Events Support are given. These notes give the URLs behind the hyperlinks.

- CICS TS V4.1 Announcement Letter:

- CICS TS V4.1 Information Center

- CB11: CICS Events for WBE

- CICS Events with WebSphere Business Events High-level animation
  - http://www.youtube.com/watch?v=S0orwDxSOvM

- CICS Events 5 minute demo
  - http://www.youtube.com/watch?v=-wQhxFfmd9U

- CICS and Events white papers

- Redbook: Implementing Event Processing with CICS

- WBM Introduction:
    - WBM V7.0 InfoCenter

- WebSphere Business Events Introduction:
    - WebSphere Business Events V7.0 InfoCenter
      - http://publib.boulder.ibm.com/infocenter/wbevents/v7r0m0/index.jsp

- Smarter Banking with CICS Transaction Server Redbook: SG24-7815-00
CICS Event Processing Summary

- **Non-invasive** emission of business events from CICS applications *without need to change* existing business logic
- **SIGNAL EVENT API** for explicit instrumentation of events
- **Event Binding Editor** tooling within CICS Explorer to create event specifications
- Event specifications deployed to CICS via bundles containing event bindings
  - Specifies event and the emitted business data, and how it can be detected and captured by the CICS runtime
  - Specify event capture points as EXEC CICS command (a subset of the EXEC CICS API) plus filtering on command parameters and data
- Events dispatched to specified EP adapter for formatting and emission to event consumer consumers including **WebSphere Business events** and **WebSphere Business Monitor**
  - CICS-provided EP adapters plus capability for custom EP adapters
IBM has invested in significant new Event technology that is a fully integrated part of the CICS runtime, and introduced with CICS TS version 4.1. This provides our strategic direction for integration with event processing products in the WebSphere portfolio.

CICS support for events allows CICS applications to emit business events in a non-invasive way, where such flexibility is required.

A new SIGNAL EVENT API is also provided, to add explicit event-enabling points into applications, which can give greater flexibility.

An Event Binding Editor is provided as part of the CICS Explorer, which allows event specifications to be created within event bindings, and deployed to CICS using CICS bundle resources.

The event specifications incorporate information about what data is to be included in the event and how the event can be captured by the CICS runtime. The points where events can be specified non-invasively are the EXEC CICS commands and also on program initiation.

Events are formatted and emitted using event processing adapters. A number of EP adapters are provided with CICS, supporting the most useful event formats and emission mechanisms. These include emitting events to WebSphere Business Events and WebSphere Business Monitor.

There is also the ability to write custom EP adapters to support other formats and ways of emitting events.