CICS Transaction Server V4.1

CICS on System z -

Turn insight into action with CICS business events

Mark Cocker

Senior Software Engineer, CICS Development, IBM UK Ltd.
mark_cocker@uk.ibm.com

© 2009 IBM Corporation

CICS on System z -
Turn insight into action with CICS business events

Developed for: Business users, business analysts, solutions architects, application development managers

Technical level: Basic – intermediate

If you could see patterns in your business processes as they occur, you could take action quickly to resolve issues. Having this insight into business transactions would give your business the power to respond rapidly to changes in market conditions. It's now possible using event processing in CICS®.

New capability in CICS Transaction Server will enable CICS applications to be a significant source of valuable business events. This session will explain how you can realize that potential. Join us for this complimentary teleconference for an overview of event processing in general, and an introduction to emission of events from CICS – without any change to your applications. The concepts of CICS events support will be covered, including event specifications and how these are created.

The session will explain how to:

- Specify the business information required in an event
- How this can be captured from the CICS application
- How to consume events in WebSphere® Business Events (WBE) to detect patterns in events, in WebSphere Business Monitor to populate business dashboards and raise alerts, and in other CICS applications

Events from CICS can a business understanding, the ability to monitor risk and business compliance, and responsiveness to business opportunities and change.

© 2009 IBM Corporation

Topics

- **Introduction to Event Processing**
- **CICS as a source of events**
- **Example workflow & roles**
- **CICS Event Provessing**
  - Specifications, capture, adapters
  - Event Binding Editor
  - Customization
- **Summary**

© 2009 IBM Corporation

Topics – Notes

- The presentation begins with an introduction to Event Processing in general, and the reasons for its current importance and business focus.
- We look at why CICS is a significant player in the event processing space, and consider some scenarios involving events from CICS applications.
- We then go through some details of the new CICS events support in CICS TS V4.1, including event specifications, EP adapters, the Event Binding Editor and options for customization.

© 2009 IBM Corporation

Introduction to event processing

Business value of events

© 2009 IBM Corporation

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 (WBE) provides complex event processing for business users

© 2009 IBM Corporation

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. 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.
- Event emission is asynchronous to the emitting application, and the consumption of the event is decoupled from its originator.
- This slide also explains the distinction between a simple event, and complex event processing, the later being based on a pattern of simple events potentially occurring over time, and correlated together in some way.
- A business event is something that happens which is relevant to the business. 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.

© 2009 IBM Corporation

Event Processing – Why now?

- **Event processing is not new**
  - Systems management and monitoring
  - Pub/Sub messaging systems
- **The business value of event processing is decreased latency in**
  - Obtaining insights
  - Making decisions based on those insights
  - Executing the decisions
- **Multiple business factors have accelerated event processing requirements:**
  - Compliance with regulations on-line
  - Demand for cost reduction leading to more automation
  - Technology developments such as RFID
  - Desire for greater awareness of business behavior through Business Activity Monitoring, Business Performance Management

© 2009 IBM Corporation

Why Now? – Notes

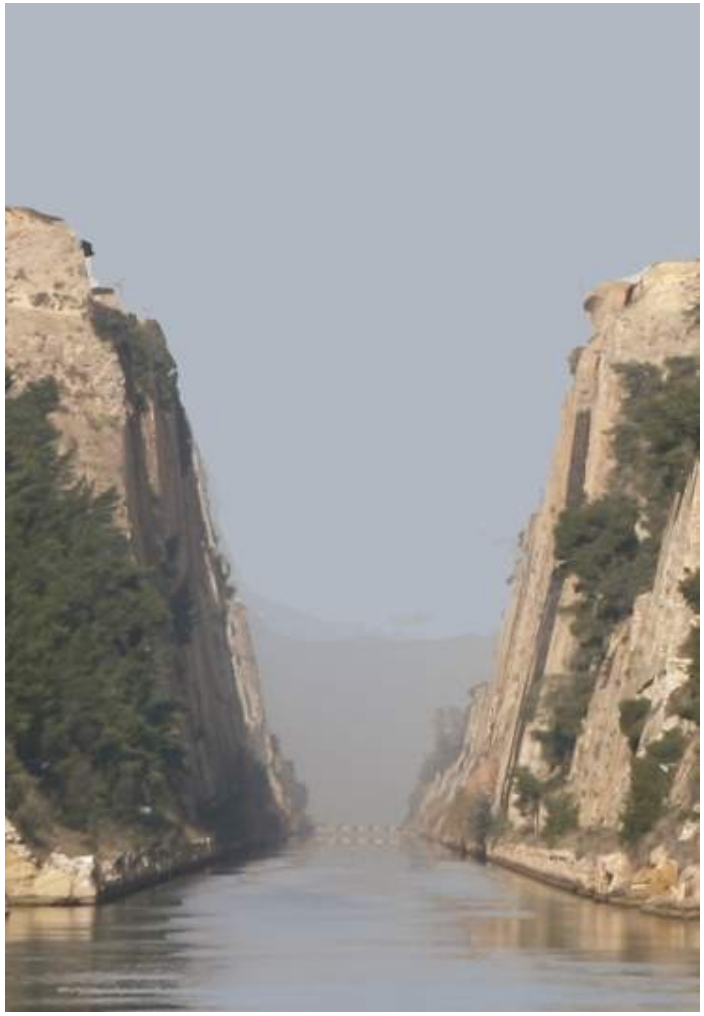
- **Event processing and event-based systems have been around for some time, used in particular in managing and monitoring IT systems, and Pub/Sub messaging is a form of event processing.**
- **An aspect of event processing which is now gaining considerable momentum is a focus on the business value which can be obtained from events, based on the growing need to react and make decisions much closer to real-time, and to gain insight into business processing.**
- **The drivers behind this momentum include the introduction of compliance regulations which require real-time responses to situations, and the desire to respond rapidly to changes in the business without entailing long development cycles.**

© 2009 IBM Corporation

The gap between LOB and IT limits agility, flexibility and responsiveness

Business users know which event patterns are relevant and what actions are required, but aren't equipped to implement themselves

Business User



IT people *may* have event-based tools and technologies but can't respond quickly enough to changing business requirements

© 2009 IBM Corporation

The Gap Between LOB and IT – Notes

- Business managers and analysts understand actionable situations; that is, the key events and the desired actions to be taken. They deal with these every day, and are directly responsible for knowing when they occur and managing the response. However, they don't have the solutions available that can enable them to identify and respond to the volume and complexity of these situations themselves.
- At the same time, although millions of potentially actionable events are flowing freely through the IT infrastructure today, supporting advanced event-driven solutions has previously required long development and test cycles

© 2009 IBM Corporation

Business Event Processing Bridges the Gap

Puts power in the hands of the business user

- No coding required for defining business event patterns
- Tasks performed via tooling

Business User

Provides unique convergence of power, flexibility, and ease of use

- IT configures end-points and payloads
- Leverages SOA infrastructure

© 2009 IBM Corporation

Business Event Processing Bridges the Gap – Notes

- The challenge of closer alignment between business and IT is addressed by what IBM refers to as ‘Business Event Processing’, which combines the event processing with capabilities to enable business users to make use of events.
- A particular Business Event Processing product to which this applies is IBM’s *WebSphere Business Events*, which provides so-called “Complex” Event Processing support.
- Business Event Processing provides IT with the functionality to support advanced event processing requirements in a high-performance, manageable, scalable environment.
- Additionally, through the inclusion of graphical, non-procedural user interfaces, Business Event Processing as provided by products such as WebSphere Business Events can equip the business user to define the event process themselves.
- The Event support in CICS is designed to significantly reduce the development time for introducing event-driven solutions, and to allow IT to respond rapidly to requests from the business.

© 2009 IBM Corporation

IBM Event Processing products – some examples

- WebSphere Process Server
- WebSphere Business Services Fabric
- CICS Transaction Server for z/OS V4.1
- WebSphere Message Broker
- WebSphere Adapters
- WebSphere Business Events
- WebSphere ESB
- InfoSphere Change Data Capture
- WebSphere Business Monitor
- WebSphere Business Modeler
- Tivoli Business Service Manager
- Cognos Now!
- Tivoli Netcool/Impact
- Tivoli Enterprise Console
- Tivoli Netcool/OMNIbus

© 2009 IBM Corporation

Event Processing Products – some examples – Notes

- This slide lists some of the IBM products which provide event processing capabilities, acting as event sources, carrying out event processing, consuming events, or in many cases a combination of these.
- For example, WebSphere Process Server and WebSphere Message Broker can act as sources of events, emitted from processes or nodes, and can also be driven as a result of events occurring. WebSphere Business Monitor uses events to monitor business processing and to measure KPIs etc., Tivoli Business Service Manager (TBSM) uses resource and system status and availability events to display information about business service availability, and Cognos Now! monitors data event streams. WebSphere Business Events provides event pattern matching and correlation capabilities.
- And also, of course, CICS TS V4.1 (the subject of this presentation) can act as a source of business events.

© 2009 IBM Corporation

CICS as a source of events

© 2009 IBM Corporation

CICS TS V4.1 is aimed at helping users to

Compete *for new opportunity 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

Event Processing

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

– IT Simplification

© 2009 IBM Corporation

CICS TS V4.1 – Notes

- This slide shows the overall themes of the CICS TS V4.1 release.
- CICS remains important to many business around the world, and of the volume of processing which takes place within CICS on a daily basis, across all industries.
  - This makes CICS a very significant source of business events.
- 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.

© 2009 IBM Corporation

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 TS 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 MQSeries, …

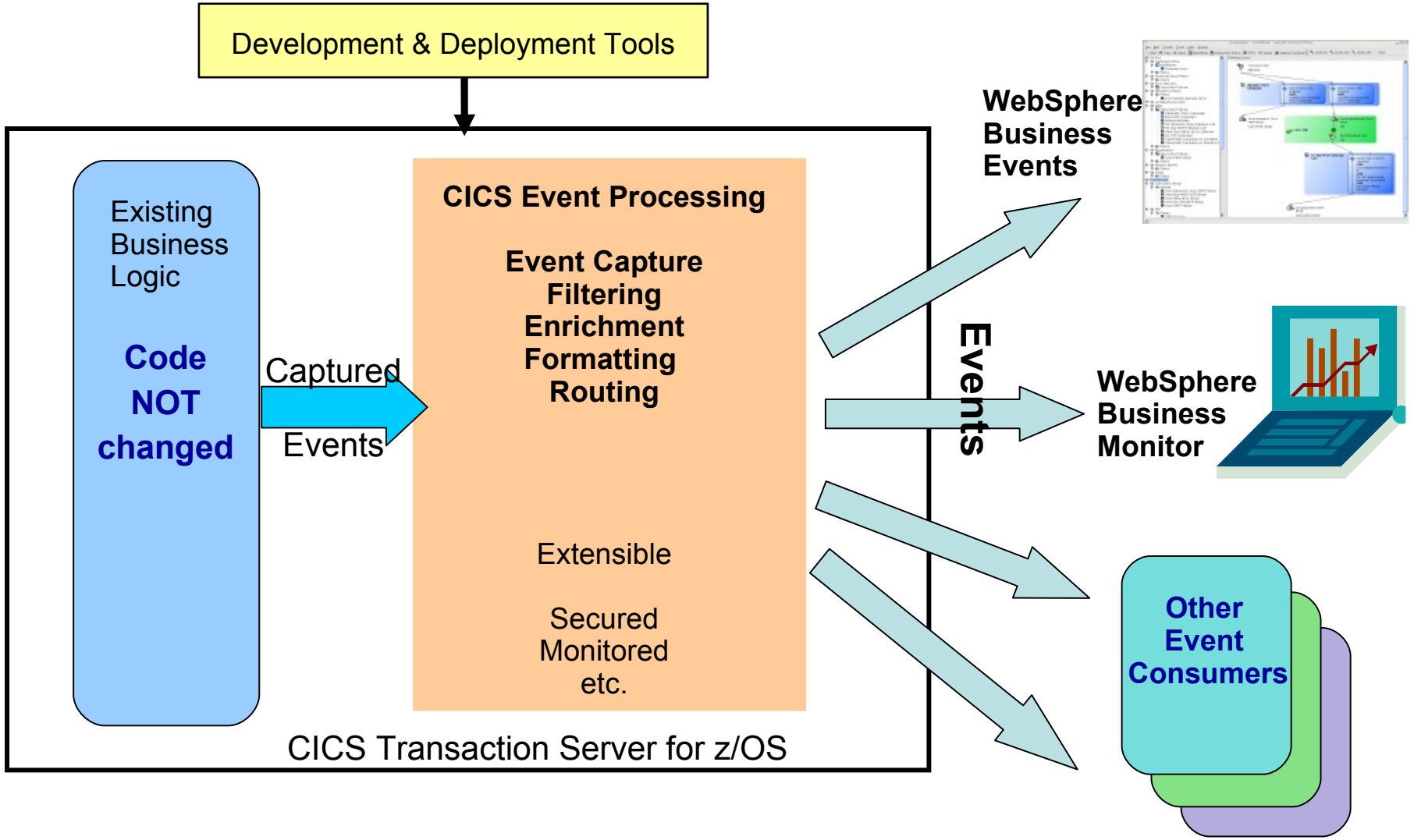
© 2009 IBM Corporation

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 help 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.

© 2009 IBM Corporation

CICS and event processing – overview



© 2009 IBM Corporation

CICS Event Processing Overview – Notes

- This gives a high-level summary of CICS as a source of business events.
- CICS event processing support will allow 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 provides 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.

© 2009 IBM Corporation

The event solution for CICS

- **Events can be emitted without change to existing business logic by defining existing EXEC CICS calls (from the supported subset) as candidates to trigger events**
- **Events triggered when the predefined conditions are met:**
  - Specified EXEC CICS Command, and
  - Specified file/program/service (for example), and/or
  - Running under specified transaction or program (for example)
- **Filter conditions or predicates relating to event data can also control the event capture (for example “account balance must be less than 100 dollars”)**
- **Once conditions are met, CICS will**
  - Capture application data specified for the event payload
  - Apply any policy relating to the event
  - Emit the event through its associated Event Processing Adapter
- **Events are specified using the Event Binding Editor tooling, under the CICS Explorer**
- **The CICS Explorer is used to deploy the events to CICS**

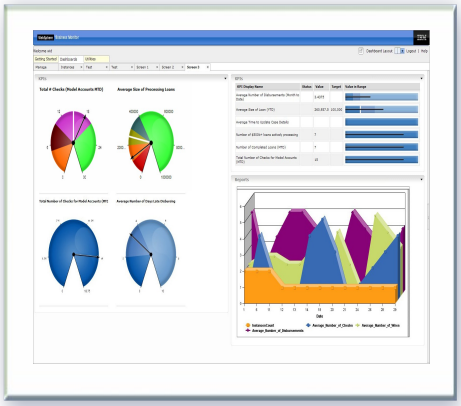
© 2009 IBM Corporation

A few example uses of CICS events

© 2009 IBM Corporation

EP Scenario 1 – Observe business processing

- **Identify key points in order processing business logic**
  - e.g. 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

© 2009 IBM Corporation

EP Scenario 1 – Observe business processing – Notes

- This is a simple instrumentation example, based on an order processing system:
  - Define event capture points 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.

© 2009 IBM Corporation

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 automatically updating an audit log or 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

© 2009 IBM Corporation

EP Scenario 2 – Non-invasive change – Notes

- 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 an MQ message or as the result of the action of a complex event processing engine.

© 2009 IBM Corporation

EP Scenario 3 – Event Combination

- **Collect events relating to credit and other bank card usage**
- **Check for unusual patterns of behaviour using WebSphere Business Events**
  - New card ordered within a week of an address change request
  - Several online purchases where none had been made before
  - 2 or more cash withdrawals in quick succession when withdrawals rare on this card, or normally for smaller amounts
  - Purchases in different geographical locations in short period of time
  - etc.
- **Specify actions to take in WebSphere Business Events e.g. confirm with cardholder that this change is expected**
- **e.g. “Red flag” policy required by US FACT (Fair and Accurate Reporting Credit Transactions) Act to detect potential instances of identity theft**

© 2009 IBM Corporation

EP Scenario 3 – Event combination – Notes

- 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.

© 2009 IBM Corporation

Example workflow for using Events in CICS

© 2009 IBM Corporation

Example Workflow for using events in CICS – Notes

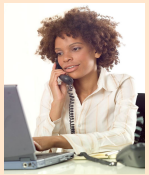
- The following set of slides show the various actors involved in specifying events to be produced by CICS, identifying where in the CICS application logic those events occur, and other aspects of using CICS events.

© 2009 IBM Corporation

LoB defines the business events

LoB Manager

Identify Business Events

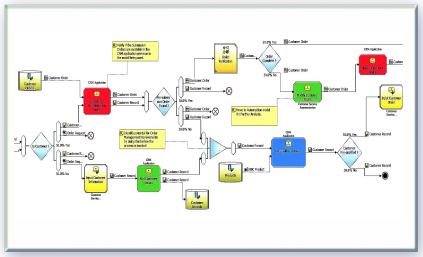


Application Analyst

Syster Progra er

Studies existing business processing and identifies business events that occur and information to be collected.

May define patterns between events and specify delivery destinations or event display using other tooling.

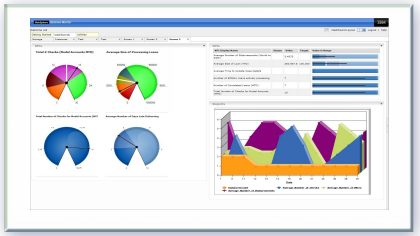


Existing Application

CICS Event Processing Support

CICS TS

Event Consumer



© 2009 IBM Corporation

LoB Defines the Business Events – Notes

This workflow looks at the roles involved in defining the events of interest to the business and enabling them in CICS, through to obtaining information as a result of the events.

It is important that business events will be definable by the business owners, Line of Business managers and Business Analysts.

They will of course need some help from the application analysts and programmers to provide the technical description of the event to CICS.

The following slides illustrate a typical CICS business event definition workflow

The LoB manager (or equivalent role) will define the CICS business application events in terms he is familiar with, together with the data that is to be passed to the event consumer with the event.

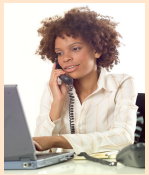
- For example
  - Every time an order in excess of €10,000 is received, capture the following fields:
    - Order Number
    - Order Date
    - Customer Ref
    - Customer Name
    - Customer Address

© 2009 IBM Corporation

Application Analyst identifies events in application

LoB Manager

Identify Business Events

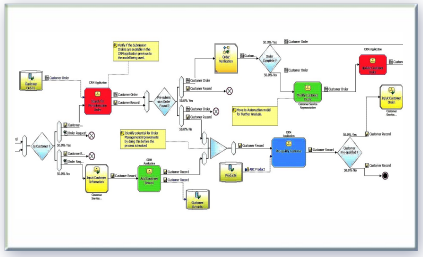


Application Analyst

Create Event Capture Specification

Inspects program(s) to determine how to capture the business events and collect the required data. Specifies where and how event is to be emitted.

Systems Programmer



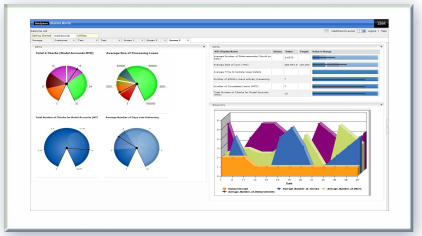
Business Modeler

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer



Business Dashboard

© 2009 IBM Corporation

Application Analyst identifies events – Notes

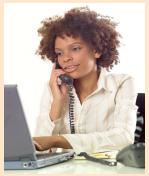
- The Application Analyst will take the LoB manager's natural language specification of the event and convert it into formal capture specifications which specify in CICS terms where the event occurs in application processing logic.
- The application analyst will also specify how the data to be included in the event payload can be obtained from data available at the event capture point and from the context in which the event occurs. This includes data which is used to filter the event (such as the order-value which needs to be > 10,000 in the above example).

© 2009 IBM Corporation

Systems Programmer configures the infrastructure

LoB Manager

Identify Business Events



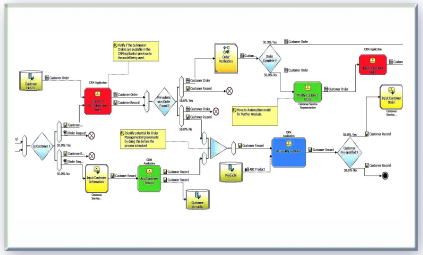
Application Analyst

Create Event Capture Specification

Systems Programmer

Specifies format and destination for the captured events. Defines CICS resources to deploy the events

Deploy Events



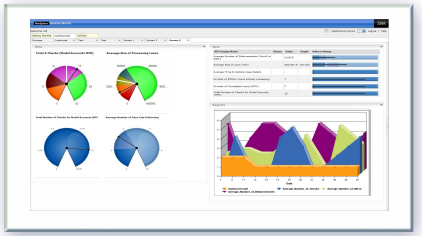
Business Modeler

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer



Business Dashboard

© 2009 IBM Corporation

Systems Programmer configures the Infrastructure – Notes

- The Systems programmer will be responsible for setting up the infrastructure in support of the event architecture. He will configure the system such that the event can be routed to the required destination, and create the CICS resources to deploy the event capture specifications into CICS.

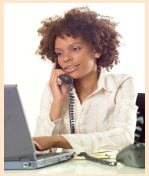
© 2009 IBM Corporation

LoB defines the business actions

LoB Manager

Identify Business Events

Specify Business Actions

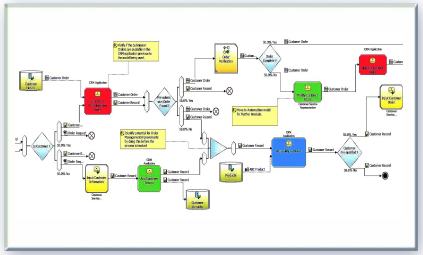


Application Analyst

Specifies the actions to take on consuming an event

Systems Programmer

Deploy Events



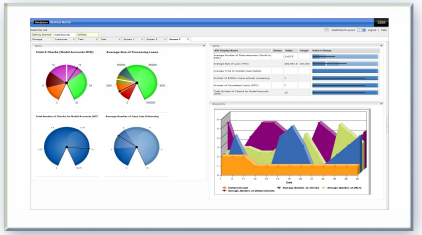
Business Modeler

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer



Business Dashboard

© 2009 IBM Corporation

LoB defines the business actions – Notes

- The event consumer will need to be configured to respond appropriately to the event
- The event consumer could include
  - A WebSphere Business Events runtime
  - WebSphere Business Monitor
  - Another CICS transaction
  - Another accessible event consumer
- The LoB manager may use tooling supplied with the chosen event consumer product (for example WebSphere Business Events) to define the actions that will occur on receiving an event from CICS. This may involve interactions between multiple events from CICS. In our example it may be that a special discount will be offered to customers placing more than 3 orders greater than 10,000 in a 30 day period.

© 2009 IBM Corporation

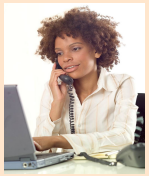
LOB optionally configures a dashboard

LoB Manager

Identify Business Events

Specify Business Actions

Optionally Configure Dashboard



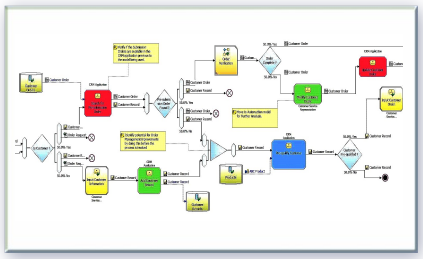
Application Analyst

Create Eve
Capture
Specificati

Optionally configures a business dashboard to display metrics based on events from CICS and other components of the application system

Events

Systems Programmer



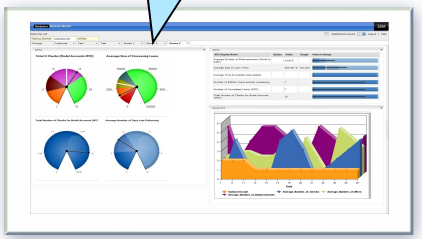
Business Modeler

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer



Business Dashboard

© 2009 IBM Corporation

LOB optionally configures a dashboard – Notes

- Optionally the LoB manager may chose to configure a business dashboard to allow an abstracted view of the events being created by the applications in the system

© 2009 IBM Corporation

CICS Event Specifications

or… how CICS knows where the events happen

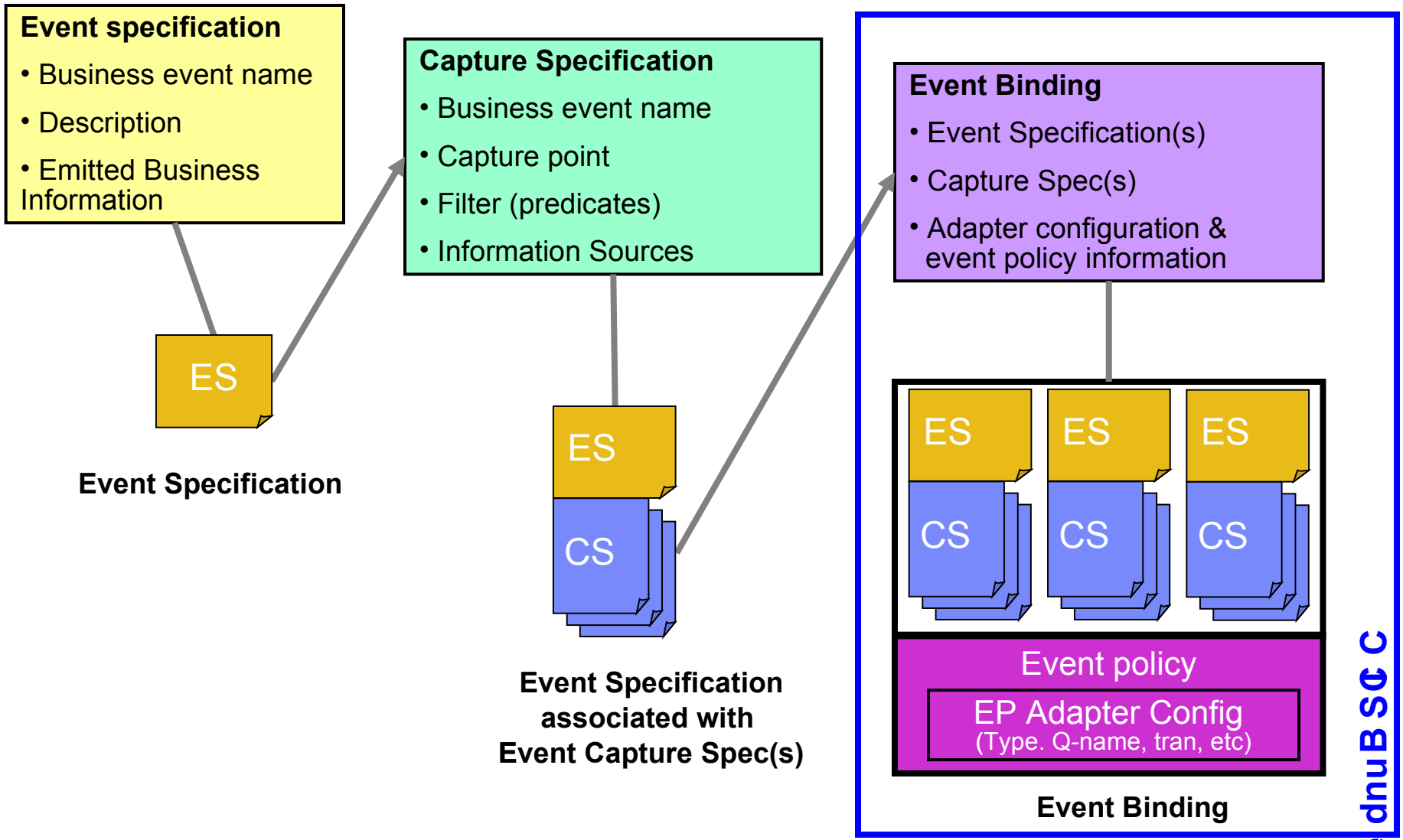
© 2009 IBM Corporation

CICS Event Specifications – Notes

- This section introduces the concept of an event specification, and how it is used to provide information to CICS about an event that is to be captured.
- This includes details of event capture, and the event processign adapters used to format and emit events.

© 2009 IBM Corporation

CICS Event Specification



© 2009 IBM Corporation

CICS Event Specification – Notes

- This slide shows the logical hierarchy of event specifications for CICS events.
- The event specification is a statement of the event required (e.g. insurance quote requested) and the business information to be emitted as part of the event (e.g. customer, insurance type).
- Associated with an event specification is normally one capture specification (although the architecture allows for more than one capture specification per event). The capture specification provides the information that CICS will use to detect the event within application processing in the system (e.g. when the insurance quote program is linked to, and data in a container passed to the program indicates that a quote is being requested). The capture specification also relates information available to the application to the business information required (e.g. data in other containers in the channel).
- Related event specifications and their associated capture specifications are grouped together into an event binding. The event binding also provides information about how (what format) and to where (e.g. which MQ queue) the event is to be emitted, in the EP adapter configuration.
- An event binding is the unit of deployment and enablement for a group of related events. It is defined to CICS and deployed as part of a Bundle, which is a new type of CICS resource.

© 2009 IBM Corporation

CICS Event Specification Example

Event Specification:

Event name: order_OverTenThousand_Received

Event description: Whenever an order is processed that is for over 10 thousand, this event is triggered for display on dashboard

Emitted business information: customer ID, OrderNumber

Capture Specification:

Before EXEC CICS LINK command ← Capture Point

to PROGRAM(OrderDB) ← Application Command Options Predicate

from current_program = OrderUI ← Application Context Predicate

where OrderVal > 10K ← Application Data Predicate

OrderVal is in the Commarea or channel passed on the LINK

How to provide event data from data available from application and context ← Information Sources (Captured Data)

Note this identifies an EXEC CICS command and some filters, it does not 'point' directly at a specific location in the application code, this can be made more or less specific by use of filtering predicates

Event Binding:

order_OverTenThousand_Received,

order_FromMajorCustomer_Received, … ← Event binding includes other related events

EP Adapter = CICSTransaction ← how events in this Binding are emitted

© 2009 IBM Corporation

CICS Event Specification Example – Notes

- This shows an example of an event specification, where we are interested in events relating to orders of value over 10,000.
- The associated capture specification contains a filtering expression made up of a number of ‘predicates’, which indicate when the event of interest occurs (in this example, when an EXEC CICS LINK command is executed to a target program OrderDB, and when the linking program is OrderUI, then an order has been received, but it is only an order for over ten thousand if the order value field passed on the LINK command has a value greater than 10000). The capture specification also contains details on how to obtain the information that is to be included in the event (the customer ID and order number in this example).
- The event specification is grouped, together with other events relating to significant orders, in an event binding which specifies how the events are to be emitted.

© 2009 IBM Corporation

The Event Binding

- **xml specification of one or more related business events**
  - **Multiple business events can be grouped together where it makes sense to deploy and manage them as a single entity**
  - **These events can be quite different from each other, but will share**
    - The same Event Processing adapter, with the same configuration
    - The same event processing policy (e.g. transactionality)
- **Each event specification may have multiple capture specifications**
  - **A particular event may have several manifestations**
    - e.g. “credit card usage” event may occur through several channels (Internet, in-store purchase, ATM machine)
  - **Each capture specification must be able to provide the data that is to be emitted in the event (e.g. credit card number and customer ID) and in the same format**
  - **More typically, each event specification will be associated with one capture specification**

© 2009 IBM Corporation

The Event Binding – Notes

- The event binding is an XML file of type .evbind, containing one or more event specifications. It allows related events to be grouped together, such as all the events relating to the insurance application, or all the order processing events which are to be monitored together.
- All the events within an event binding are handled in the same way, in that they go to the same event processing adapter with the same configuration (e.g. sent via WebSphere MQ to the same queue in the same event format), and that they all have the same event processing policy (the same priority, transactionality etc.).
- Within an event binding, each event specification is associated with one or more capture specifications. In the more normal case, there would be one capture specification per event, but the CICS event binding allows the concept of multiple capture specifications associated with an event. This is for the case where the same event can occur in different code paths, requiring different capture specifications to identify and capture all occurrences of the event. Where multiple capture specifications are used, they must be able to provide all of the data which is required in the emitted business information for the event.

© 2009 IBM Corporation

Deployment of Event Bindings

- **The Event Binding is deployed into CICS**
  - Event Bindings are deployed via inclusion in a CICS Bundle
    - A CICS Bundle resource contains a collection of related CICS resources
      - An archive file containing resources, artifacts, etc. plus a manifest
      - Used for a number of new resources, including Event Bindings
      - Installing a BUNDLE into CICS will install the included resources
    - Create bundle and export to zFS
    - Define bundle resource including the zFS location
  - Deploying the Event Binding into a particular CICS region will resolve the capture specifications
  - CICS enables the capture specifications in the Event Binding so that they can be intercepted during runtime processing

© 2009 IBM Corporation

Deployment of event bindings – Notes

- A CICS Bundle resource is a new type of resource introduced in CICS TS V4.1, and is created as an archive file with a manifest which describes its contents, and the resources and artifacts which form its contents. CICS bundles are used for a number of types of resource, one of which is the event binding.
- When a bundle is installed into CICS, the various resources that it contains are each installed as a result. To install an event binding into CICS, you therefore install the bundle in which it is included.
- When an event binding is installed, the capture specifications associated with the events in the bundle are deployed into CICS and (assuming they are enabled) the runtime will start to capture events where they match the filtering specified in the capture specifications.
- You install the event bindings into those CICS regions where you want the events that they contain to be captured and emitted. Or you could install the bindings into all the CICS regions where you might want to capture those events, and only enable them in regions where you currently want to do so. For example, there could be 3 regions that LINK from OrderUI to OrderDB, but if you are only interested in the event in regions1 and 2, then the Binding can be deployed into those regions only (or could be deployed into the third region for future use, but currently disabled).
- When an event binding is installed, the information about how and where it is to be emitted (the EP adapter information) is made available to CICS, so that when the events occur they can be handled as required.

© 2009 IBM Corporation

CICS Event Capture

© 2009 IBM Corporation

CICS Event Capture – Notes

- This subsection provides some more detail about event capture, including the explicit event API which is available in addition to non-invasive specification of events, and explains which EXEC CICS commands can be specified as ‘capture points’ i.e. included in event capture specifications.

© 2009 IBM Corporation

CICS Event Capture options

- Non-invasive
  - Declare event points in application logic without opening up the application
  - Use application knowledge to map business event onto point(s) in the logic where the event occurs

- Explicit API
  - **EXEC CICS SIGNAL EVENT**
    - EVENT supplies an event identifier
    - Data can be supplied as either FROMCHANNEL or Data area and length
    - Identifier to be used in event specification
  - Explicit way of adding a capture point to an application
    - Allows exact pinpointing of the event point, and exact selection of relevant data
    - Use to “event-enable” the application
    - Once this is done, the instrumentation can be used for different purposes
  - Define as event within an event binding
    - Allows filtering and selection of data to use for different business events
    - Allows event to be enabled and disabled
    - ‘fast path’ in tooling to simplify specification of explicit events

© 2009 IBM Corporation

CICS Event Capture Options – Notes

- The main focus for CICS event processing support is on the ability for CICS to capture your events without the need to change the application code. This is referred to as 'non-invasive' event capture.
- As we will see shortly, the subset of the CICS API supported for event capture has been selected to give the best chance that you will be able to specify where events occur in your applications in this non-invasive manner.
- However, there will be some situations where you want explicit control over capturing of events in your applications, and there is a new EXEC CICS SIGNAL EVENT API introduced to allow you to do this. This might be because the best place to detect an event is not associated with an EXEC CICS command, or not with the supported commands, or it could be because data you would like to include in the event is not available on an API command, but can be extracted from information known to the program.
- Note that the SIGNAL EVENT command does not cause the event to be automatically emitted. Instead, it allows you to include this command within an event specification, which gives you the flexibility to include data from the event in filtering expressions, and to extract only relevant parts of the data. In this way, having added SIGNAL EVENT calls to your application, you can regard it as event-enabled, and can enable and disable those event points, and alter how they are used without further change to the application.

© 2009 IBM Corporation

Eventable CICS Commands – Principles

- Focus is on events of interest in business terms, so commands relating to system activity not eventable
  - e.g. not ABEND, DUMP TRANSACTION, HANDLE CONDITION, SPI commands
- Anything that starts work has a good likelihood of mapping to business events
  - e.g. START, START TRANSID, LINK, INVOKE WEBSERVICE
  - Also enable event capture for program initiation via whatever means (e.g. Web services pipeline, entering tranid at a terminal)
- Getting data into or out of CICS can be a good way of finding out about business events
  - e.g. RECEIVE MAP, RECEIVE, SEND MAP
- Writes to CICS data resources (files, queues) may often occur when processing business events
  - e.g. WRITE FILE, WRITEQ TS
- Reads of CICS data resources are also interesting, as events do not only occur when data is updated
  - e.g. READ FILE, READQ TD
- Do not (initially) plan to event enable data-oriented commands which can be evented in other ways (such as via the database)
  - e.g. not RMI (DB2, MQ)
  - In a future release, might event enable these commands to get additional application context
    - **but** only limited information about the command could be available to CICS & included in capture specs

© 2009 IBM Corporation

Eventable CICS commands – Principles – Notes

- This slide describes the basis on which the set of 'Eventable' CICS commands were selected. An 'eventable' command is one which can be included in an event capture specification as the capture point for the event.
- Commands which are likely to relate to events which will be of interest to the business are:
  - Commands which initiate work in CICS
  - Commands which receive data into or send data out from CICS
  - Commands which access CICS data resources, either as updates to the resources, or as read accesses.
- Commands which are not regarded as likely candidates for events of business interest are commands relating to the system and its availability.
- Commands which access non-CICS resources will not be eventable in CICS TS V4.1. There are often other ways of obtaining events relating to these data accesses.

© 2009 IBM Corporation

Eventable CICS Commands

- Channel commands
  - PUT CONTAINER, START (TRANSID)
- File Control
  - WRITE, REWRITE, DELETE
  - READ, READNEXT, READPREV
- Interval Control
  - START, RETRIEVE
- Program Control
  - LINK, RETURN, XCTL
- Scheduling Services
  - START (ATTACH)
- Temporary Storage
  - WRITEQ TS, READQ TS, DELETEQ TS
- Transient Data
  - WRITEQ TD, READQ TD, DELETEQ TD
- Web support
  - INVOKE (WEB)SERVICE
  - WEB READ, WEB READNEXT
- BMS
  - RECEIVE MAP
  - SEND MAP
  - SEND TEXT
- Terminal Control
  - CONVERSE, RECEIVE, SEND
- New APIs
  - SIGNAL EVENT, INVOKE SERVICE
- Program initiation
  - Enable event when program starts

© 2009 IBM Corporation

Eventable CICS Commands– Notes

- This slide lists the commands which can be specified in event capture specifications in CICS TS V4.1.
- This list might be extended in the future, as we learn more about the places in applications where events occur.
- There is also one event capture point which is not an EXEC CICS command – this is program initiation, which allows you to specify that an event is to be captured when a specified program starts, however that program was initiated.

© 2009 IBM Corporation

Filterable and Capturable Data

- Application Context – applies to all commands
  - Filterable (can be included in a predicate):
    - Tranid, Current program, Userid, Command response (OK/not OK)
  - Captured automatically:
    - UOWid, Network applid qualifier & CICS applid, Date & time
  - Capturable (can be information source for an item of emitted business information):
    - Tranid, Current program, Userid
- Application command options and application data – Command-specific
  - e.g. For RECEIVE MAP
    - Filterable and capturable: MAP (*primary predicate*), MAPSET, EIBaid, EIBCposn
      - ***Primary Predicate*** for each command is the data item on which filtering is *strongly recommended* for performance
  - e.g. For LINK
    - Filterable & Capturable: Program (*primary predicate*), Data from channel or Commarea
- Most commands will be captured *after* they occur, some offer the option to capture *before* e.g. LINK

© 2009 IBM Corporation

Filterable and Capturable Data – Notes

- This slide explains how the data available when an EXEC CICS command is executed can be used in filtering statements (predicates) and how it can be captured (act as an information source)
- There is some application context which is available at any capture point. This is the transaction ID, current program, userid, UOW id, network-qualified CICS applid, date & time, and the response from the command. Some of this application context can be used to filter whether the event is to be captured, some is automatically captured and will (for events formatted by a CICS-provided EP adapter) be included as contextual information in the event, and some can be specifically captured as part of the emitted business data, as shown on the slide.
- The application command options and application data are specific to a command, and can similarly be used in filtering expressions or captured as event data as appropriate. For example, on a LINK command, the program name can be specified as a predicate in a filter or captured as an emitted business item. Also, any of the data passed on the LINK in the channel or commarea can be filtered on or captured.
- Each command has an identified primary predicate, which is the application command option for which you are recommended to provide a predicate expression, to optimize performance.

© 2009 IBM Corporation

EP Adapters

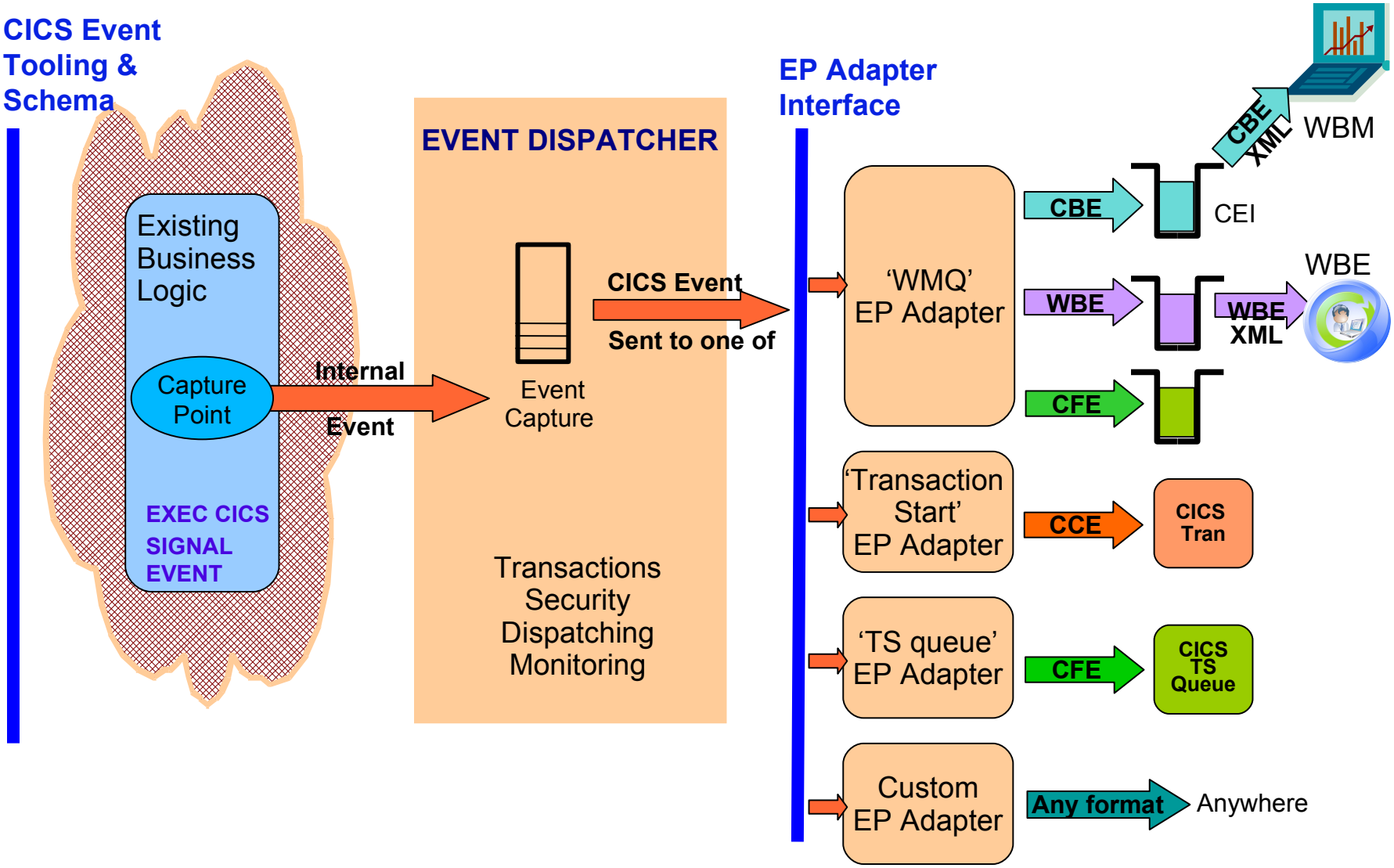
© 2009 IBM Corporation

Introduction to EP Adapters – Notes

- This section introduces the Event Processing Adapters, which carry out the formatting and routing of events after they have been captured by CICS.

© 2009 IBM Corporation

CICS Event Processing Adapters



© 2009 IBM Corporation

CICS Event Processing Adapters – Notes

- When an event is captured, CICS directs it to an EP adapter, based on what has been specified in the event binding. All adapters are invoked using a standard EP Adapter interface.
- The EP adapters format the event and route it to the potential consumers, which include WebSphere Business Monitor, WebSphere Business Events, and a CICS transaction. More details are given on the next slide.
- This slide also illustrates the customization options for CICS Event Processing support, discussed in more detail later:
  - The standard EP adapter interface allows custom EP adapters to be written, to support formats and/or transports not provided by CICS.
  - The event binding schema files are shipped with CICS, providing opportunities to create pre-built event bindings, or different tooling interfaces.
  - The explicit SIGNAL EVENT API allows applications to be event-enabled.

© 2009 IBM Corporation

EP Adapters and Event Formats

EP Editor Tooling: / EP Adapter Type	EP Adapter specified	Event format emitted	Transport or Emission Mechanism	Intended consumer(s)
Message Queue	WMQ Queue EP Adapter	CBE (XML)	WebSphere MQ	WebSphere Business Monitor or any application which processes a CBE
		WBE (XML)		WebSphere Business Events or any application which processes a WBE connector
		CFE (structure)		Application written to get the CFE from the message queue
CICS Transaction	Transaction Start EP Adapter	CCE (containers in channel)	Program start, with channel	A CICS application which processes the event data from the containers (or just takes action as a result of being driven.
CICS TS Queue	TS Queue EP Adapter	CFE (structure)	Temporary storage queue write	Test and debug
Custom	User-written			

© 2009 IBM Corporation

EP Adapters and Event Formats – Notes

- The EP adapters provided by CICS TS V4.1 support the following transports and formats:
  - WMQ transport, CBE (Common Base Event) XML format – primarily for sending events to WebSphere Business Monitor
  - WMQ transport, WBE XML format – primarily for sending events to WebSphere Business Events
  - WMQ transport, CFE ("CICS Flattened Event") text-based format – can be read from the queue by consuming application
  - CICS transaction start 'transport', CCE ("CICS Channel-based Event") format – to drive new work in this or another CICS region
  - Temporary storage queue 'transport', CFE format – primarily to test that events are emitted when expected and contain the correct data
- You can deploy 2 or more Event Bindings containing the same capture specification, to allow the same event to be directed to more than one EP adapter (i.e. for consumption in different ways) or to the same type of EP adapter but with different configuration options (e.g. to 2 different WMQ queues).

© 2009 IBM Corporation

Event Transactionality

- **Transactional option on the event definition**
  - Part of the Advanced Adapter Options on an event binding
  - When set, causes CICS to wait for syncpoint completion before either emitting or discarding event (depending on syncpoint outcome)
  - For many events will not want transactionality e.g. attempt to write to file could be as interesting as succeeding
- **Note**
  - Transactional events are not emitted until the UOW reaches syncpoint – for a long-running transaction, this could mean the events are not very close to real-time

© 2009 IBM Corporation

Event Transactionality – Notes

- Although the default for an event binding is for the events to be non-transactional, as CICS is a transaction processor, the capability to make events transactional is provided.
- If ‘Transactional’ is specified (i.e. the box for ‘Events are Transactional’ is checked in the Event Binding Editor), then events will be captured as they take pace, but are emitted if and when the unit of work in which the event took pace is committed. If the unit of work is rolled back or abends, then no transactional events in that unit of work will be emitted.
- There are performance implications associated with defining events as transactional; most notably that these events will be kept in the system until the unit of work commits, so they can be less timely than non-transactional events.

© 2009 IBM Corporation

Other event processing policy attributes

- **Dispatch Priority**
  - Specify priority of events in the event binding as Normal or High
- **Userid the EP Adapter is to run under**
  - Specify a userid under which the EP adapter will run
    - e.g. might be needed to allow access to required WMQ queue, or for actions carried out by custom EP adapter
  - 'Use context userid' will run EP adapter under the same userid as that running when the event was captured
  - By default, EP adapters run under CICS region userid
- **Transaction ID the EP adapter is to run under**
  - Normally runs under a default tranid
- **Can specify a different tranid or userid for charging**
- **Some performance implications to specifying userid or tranid**

© 2009 IBM Corporation

Other event processing Policy – Notes

- It is also possible to specify whether the event is to be processed at normal or high priority. High priority events will be processed by CICS in preference to Normal priority events that are captured at around the same time.
- By default, EP adapters run under a default transaction id, and using the CICS region userid. For access to resources or accounting purposes, for example, a specified userid and/or tranid can be used. This includes an option to run under whichever userid was running when the event was captured.
- Note that there is some overhead associated with specifying a userid or tranid other than the default, as the EP adapter must then be run under a separately attached task.
- The transaction ID and user ID options are not available for all EP adapters; for example, the transaction start EP adapter always runs under the defaults (as the transaction it starts will run under a separate transaction ID and optionally a separate user ID).

© 2009 IBM Corporation

Event Binding Editor

The CICS event specification tooling

© 2009 IBM Corporation

Event Binding Editor – Notes

- This section describes the Event Binding Editor, which is the tooling in which you create event specifications within an event binding.

© 2009 IBM Corporation

CICS EP Tooling

- **Event Binding Editor**
  - Eclipse editor feature, operating on event binding files (type .evbind)
- **Event Binding Editor is a part of the CICS Explorer**
  - Create event binding(s) within a Bundle Project
    - CICS Explorer provides support to deploy bundles containing event bindings
- **Event binding can be built up in stages**
  - Validated against schema each time it is saved, other validation as information added
  - A problems pane keeps a running record of all errors

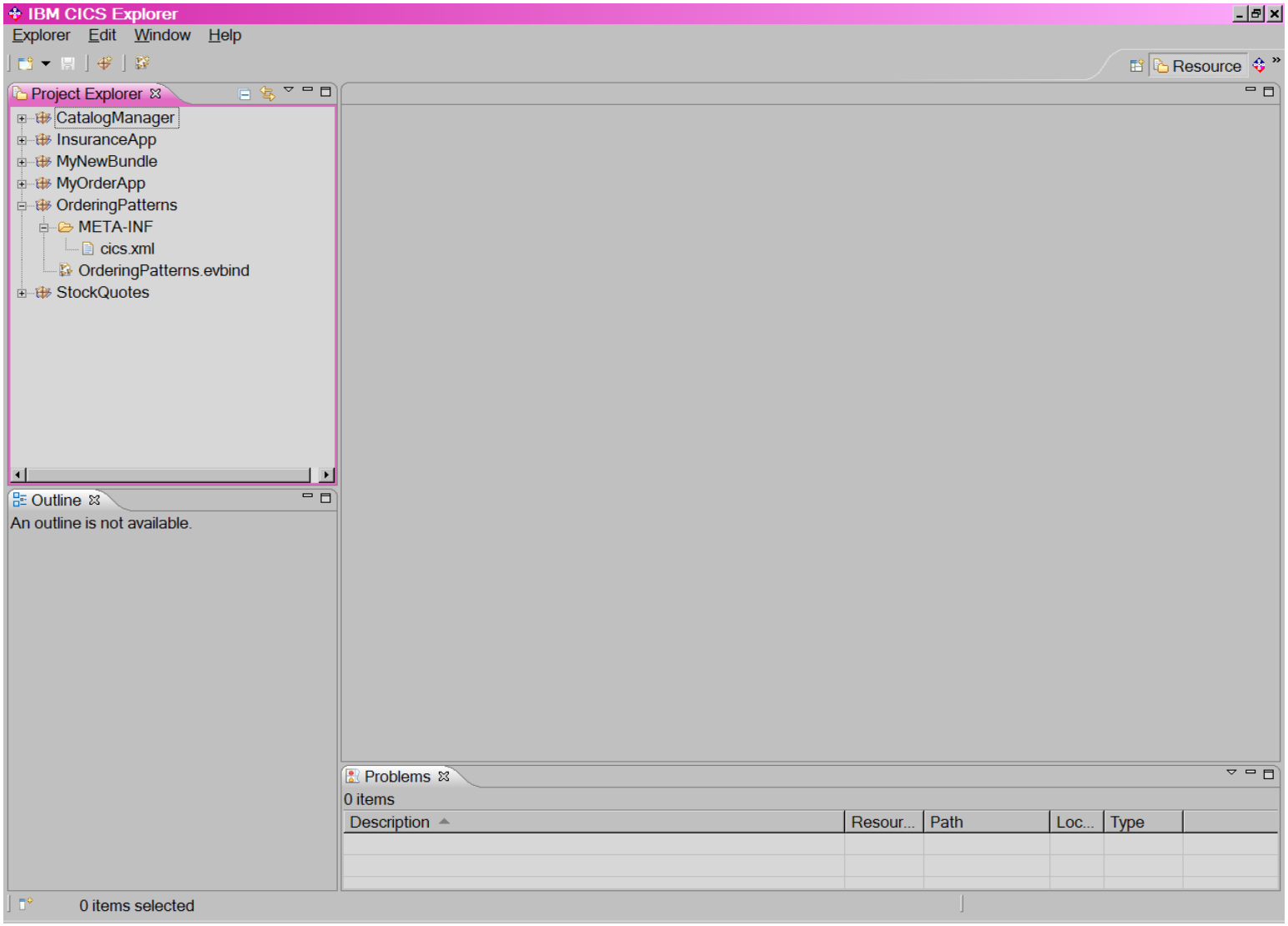
© 2009 IBM Corporation

CICS EP Tooling – Notes

- The Event Binding Editor is an Eclipse-based editor. It is part of the CICS Explorer ("the new face of CICS") and support for creation and deployment of CICS Bundles is provided by the CICS Explorer.
- The starting point for creating event specifications with the Event Binding Editor is a Bundle Project.
- Event Bindings can then be added to the Bundle, to which in turn event specifications are added, and their associated capture specifications. The EP Adapter configuration and event policy information are also specified for the events in the event binding.

© 2009 IBM Corporation

CICS Explorer Resource Perspective



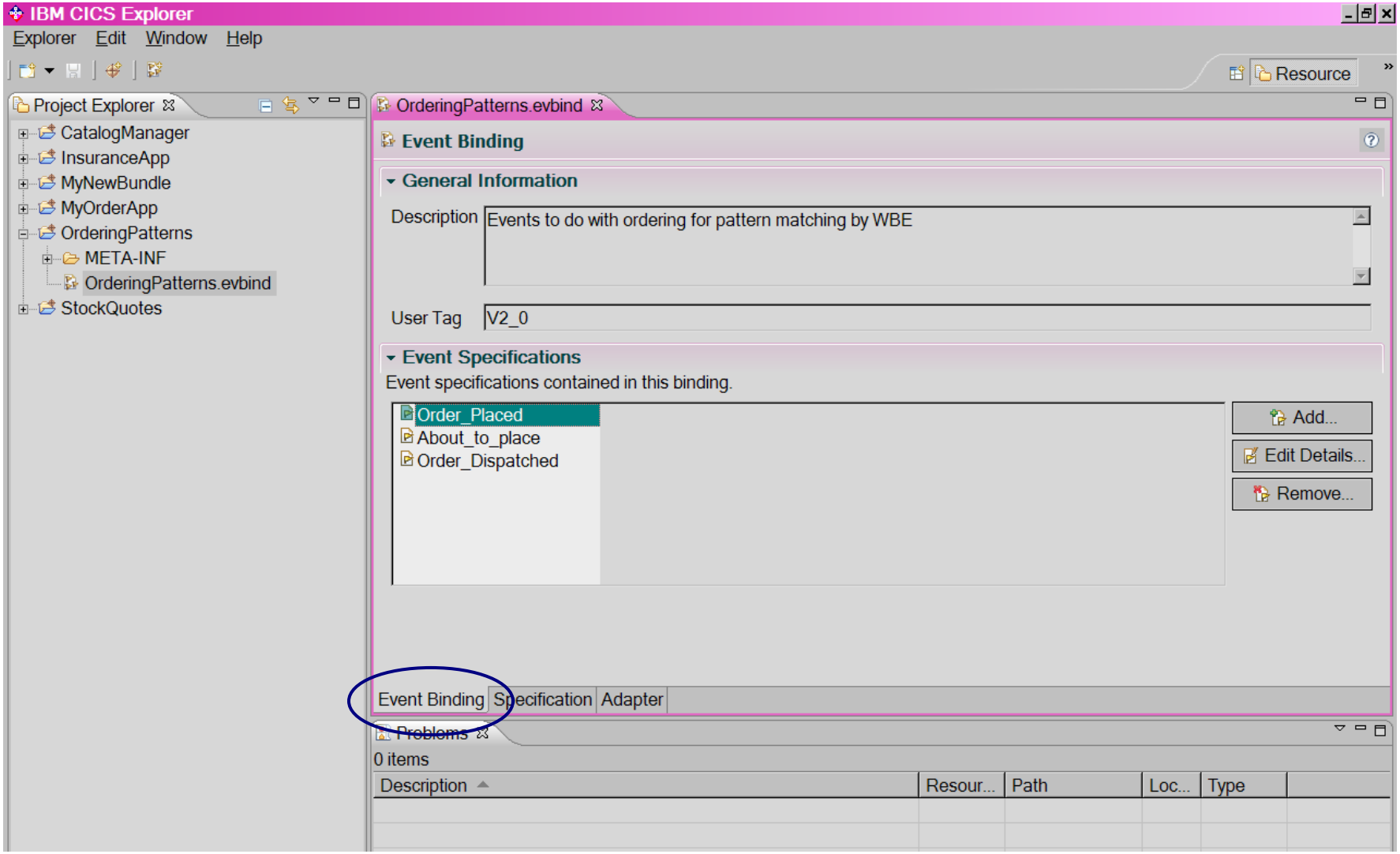
© 2009 IBM Corporation

CICS Explorer Resource Perspective – Notes

- You create event bindings within CICS Bundle Projects using the CICS Explorer.
- The starting point for this is the Resource Perspective (Window->Open Perspective)
- Within the Resource Perspective you can create a new CICS Bundle Project (Explorer -> New Wizards ->). The first time you do this, you need to select Other, expand the CICS resources, and select CICS Bundle Project. The CICS Bundle Project requires a name.
- An event binding can now be created; for example, right click on the bundle project (under Project Explorer) and select New -> Event Binding.
- The slide shows a number of Bundle Projects which have been defined, with one expanded. This Bundle contains the bundle manifest (cics.xml) and an event binding (OrderingPatterns).

© 2009 IBM Corporation

Event Binding Editor – Event Binding



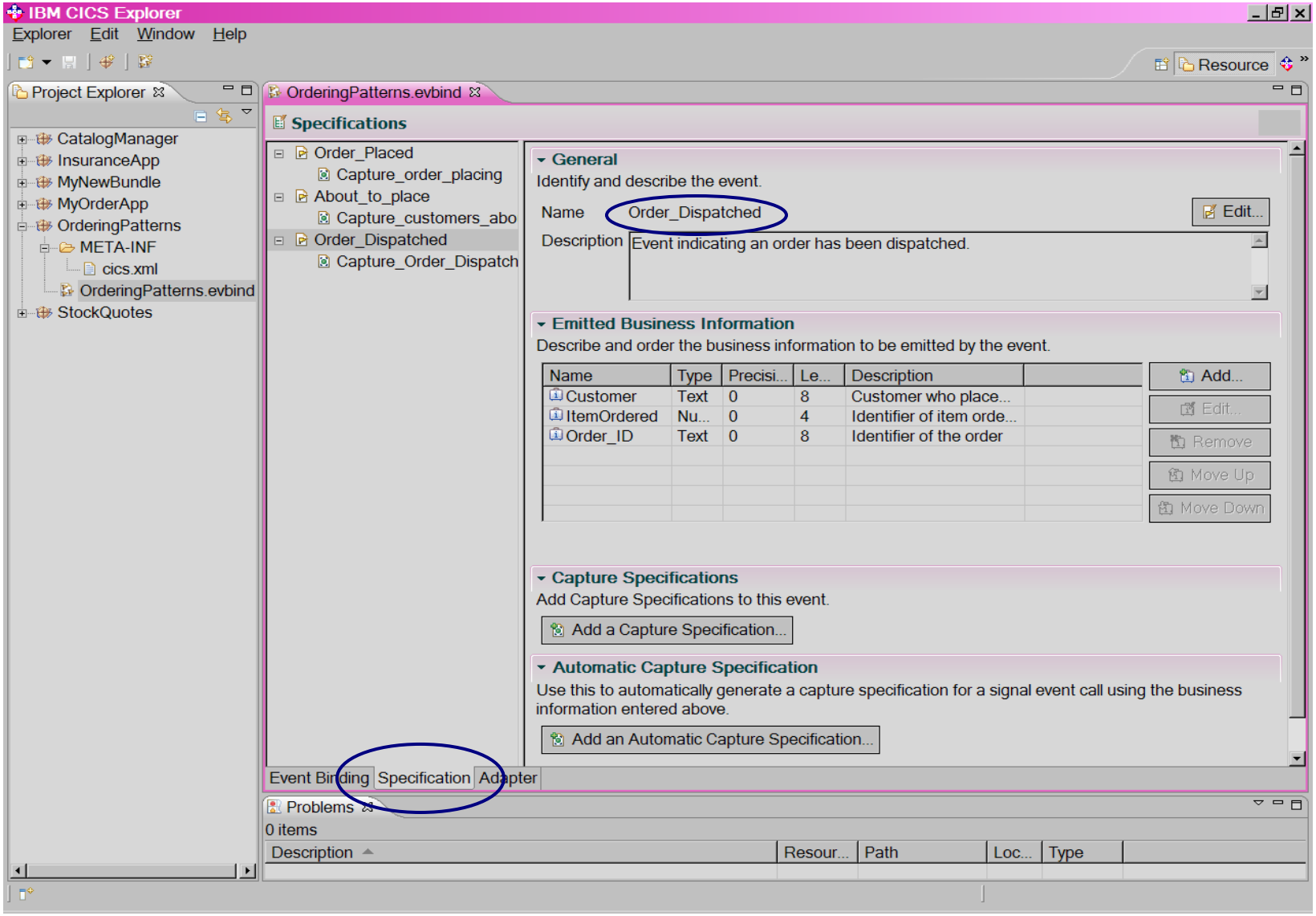
© 2009 IBM Corporation

Event Binding Editor – Event Binding – Notes

- This slide shows an event binding open in the Editor.
- The slide displays the General Information for the Event Binding, which is its description, a User Tag which can be used in any way you wish to identify this iteration of the event binding, and the event specifications contained in the event binding.
- There are three aspects to an event binding, as shown by the tabs at the bottom of this screen shot: the overall event binding, the specifications within the binding, and the EP adapter configuration.

© 2009 IBM Corporation

Event Binding Editor – Event Specification



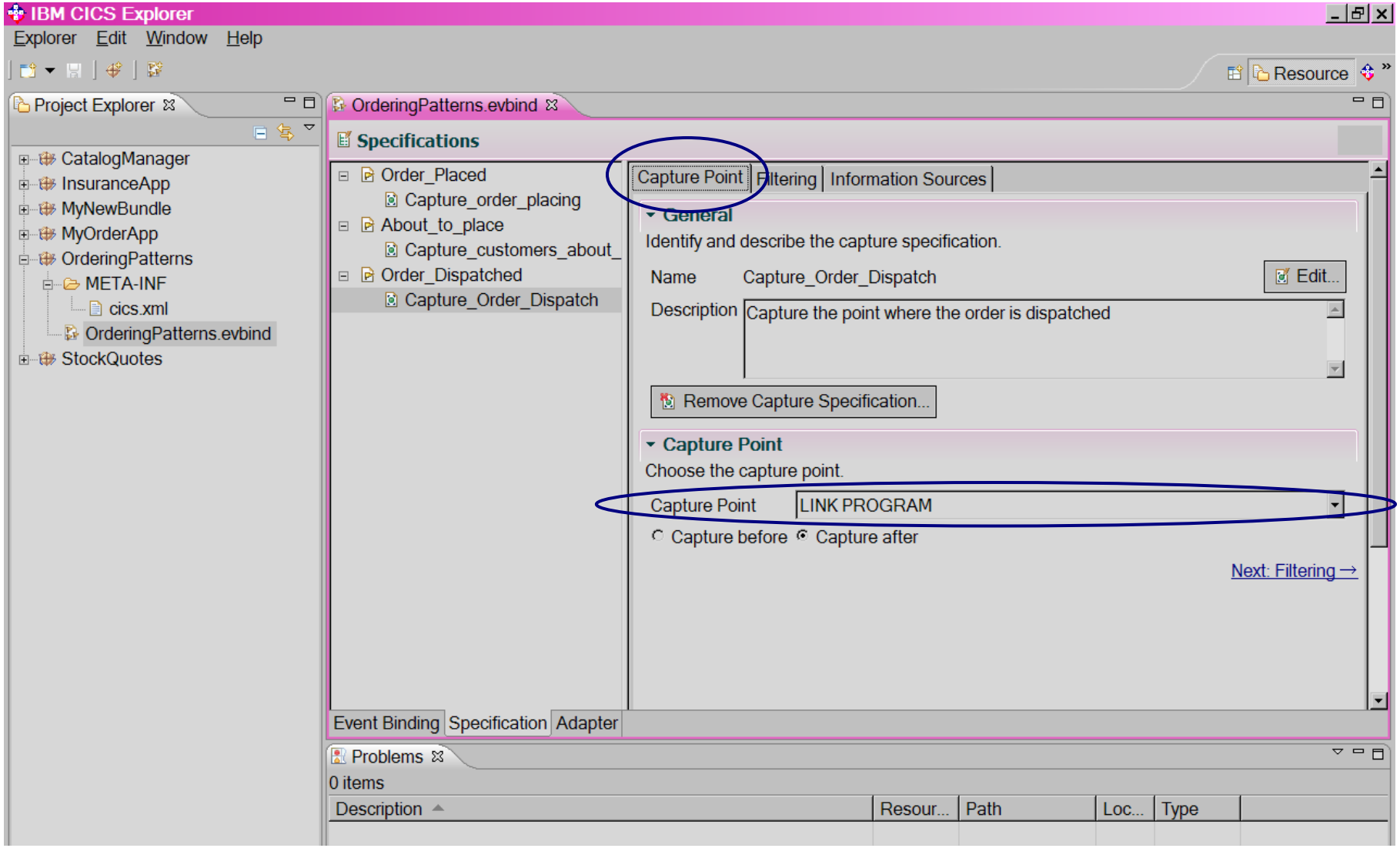
© 2009 IBM Corporation

Event Binding Editor – Event Spec – Notes

- To move to the specifications portion of the editor, you can Add a new event, Edit details of an existing event, or select the ‘Specification’ tab and the event to work with.
- This slide shows an event Order_Dispatched within an event binding called OrderingPatterns. This view of the event specification shows the business-oriented aspects of the event: its name, which will indicate the interesting thing that happened, a description (which is used to describe the event when working with it in the Event Binding Editor), and the items of information to be included in the event.
- In this example, the event indicates that an order has been dispatched, and the business information that is to be emitted as part of the event is the customer to whom the order is being dispatched, the item that was ordered, and an order identifier for the order. When specifying the emitted business information, you also specify whether it is text or numeric data, the length to be emitted, and a precision to be used (for numeric data).
- Having defined the external view of the event, we need to ‘Add a capture specification’, which will indicate to CICS how to detect this event when it occurs at runtime.
- (The Automatic Capture Specification option is for adding a simple capture specification to match explicit EXEC CICS SIGNAL EVENT calls.)

© 2009 IBM Corporation

Event Binding Editor – Capture Specification



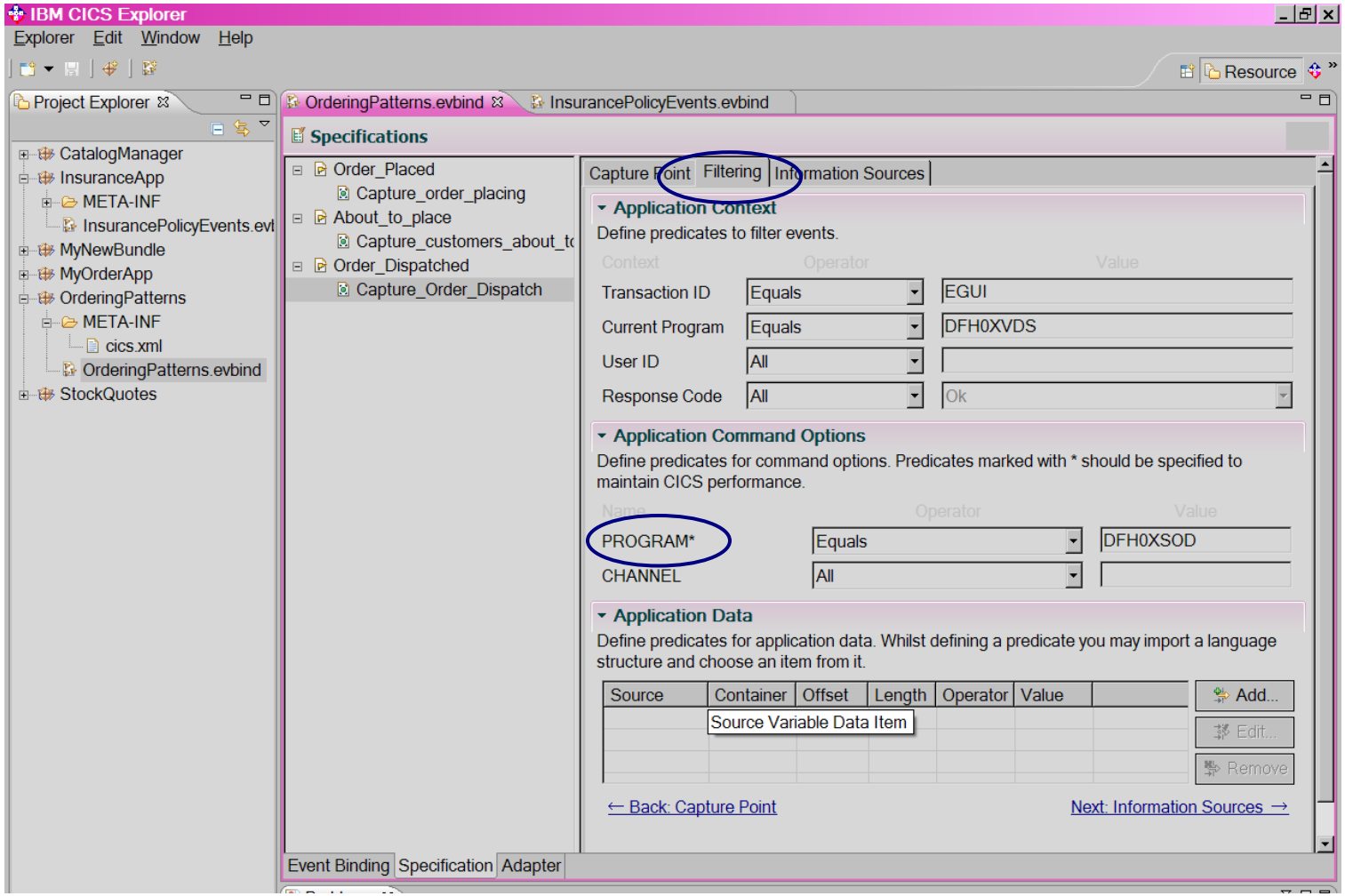
© 2009 IBM Corporation

Event Binding Editor – Capture Specification – Notes

- Adding a capture specification, or selecting a previously added capture specification, takes you to the Capture Specification dialog, which has 3 parts (indicated by the tabs for ‘Capture Point’. ‘Filtering’ and ‘Information Sources’. This screen shot shows the Capture Point tab.
- The General information is the name of the capture specification and a description of it. The name does not need to be related to the event name, but something which indicates the interaction between the two is recommended.
- The Capture Point section is where you indicate which of the supported subset of EXEC CICS commands it to be used to detect that this event has happened. The screen shot shows a LINK PROGRAM command selected from the pull down list of supported commands (which also includes ‘Program Initiation’). This is one of a few commands which have an option to capture the event either before or after the command executes; for others (for example, REWRITE to a FILE) the capture is always after the command.
- Without adding any filtering, this event would be emitted every time any program in the CICS region into which it was deployed issued a LINK PROGRAM, so some filtering is advisable. This is the next tab in the capture specification.

© 2009 IBM Corporation

Event Binding Editor – Capture Specification Filtering



© 2009 IBM Corporation

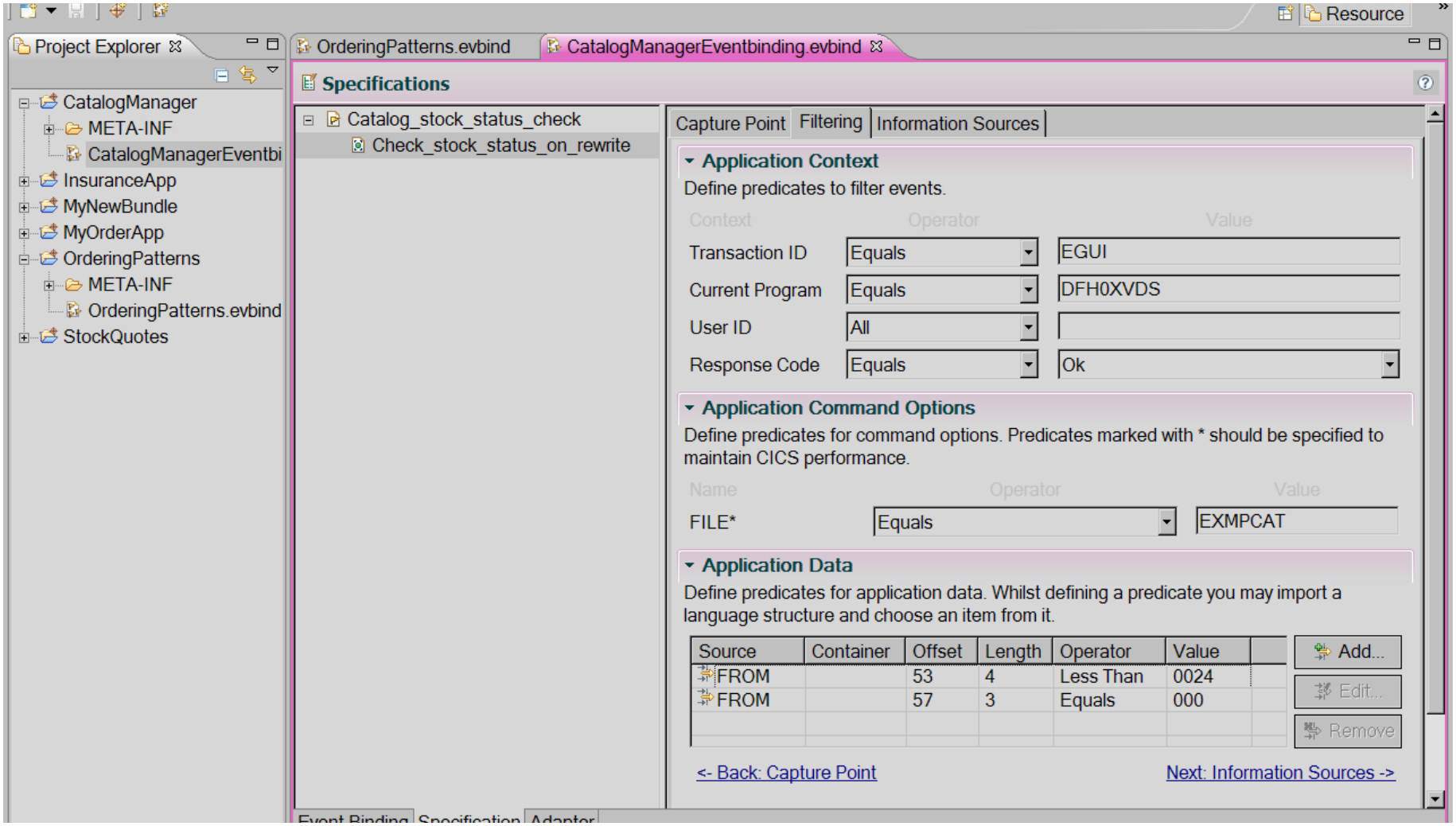
Event Binding Editor – Capture Specification Filtering – Notes

- The Filtering tab allows you to specify in greater detail when the event should be captured.
- There are three types of information which can be used in the filtering expresssions, or *predicates*:
  - Application Context data is information available from the context in which the command occurs, such as the current program or transaction.
  - Application Command Data covers values that are specified on the API command
  - Application Data is data within variables on the command.
- Each command has a 'primary predicate' which is indicated in the event binding editor with an asterisk. For a LINK command, it is the name of the PROGRAM being linked to, and in this example the program name has been specified. The other predicates provided to filter this event are the transaction ID and the current program (i.e. the program that issued the LINK) in the application context. Other data that would be available for filtering is set to 'All', meaning that all values will match.

© 2009 IBM Corporation

Event Binding Editor – Filtering (2)

Example showing more predicates used to filter an event



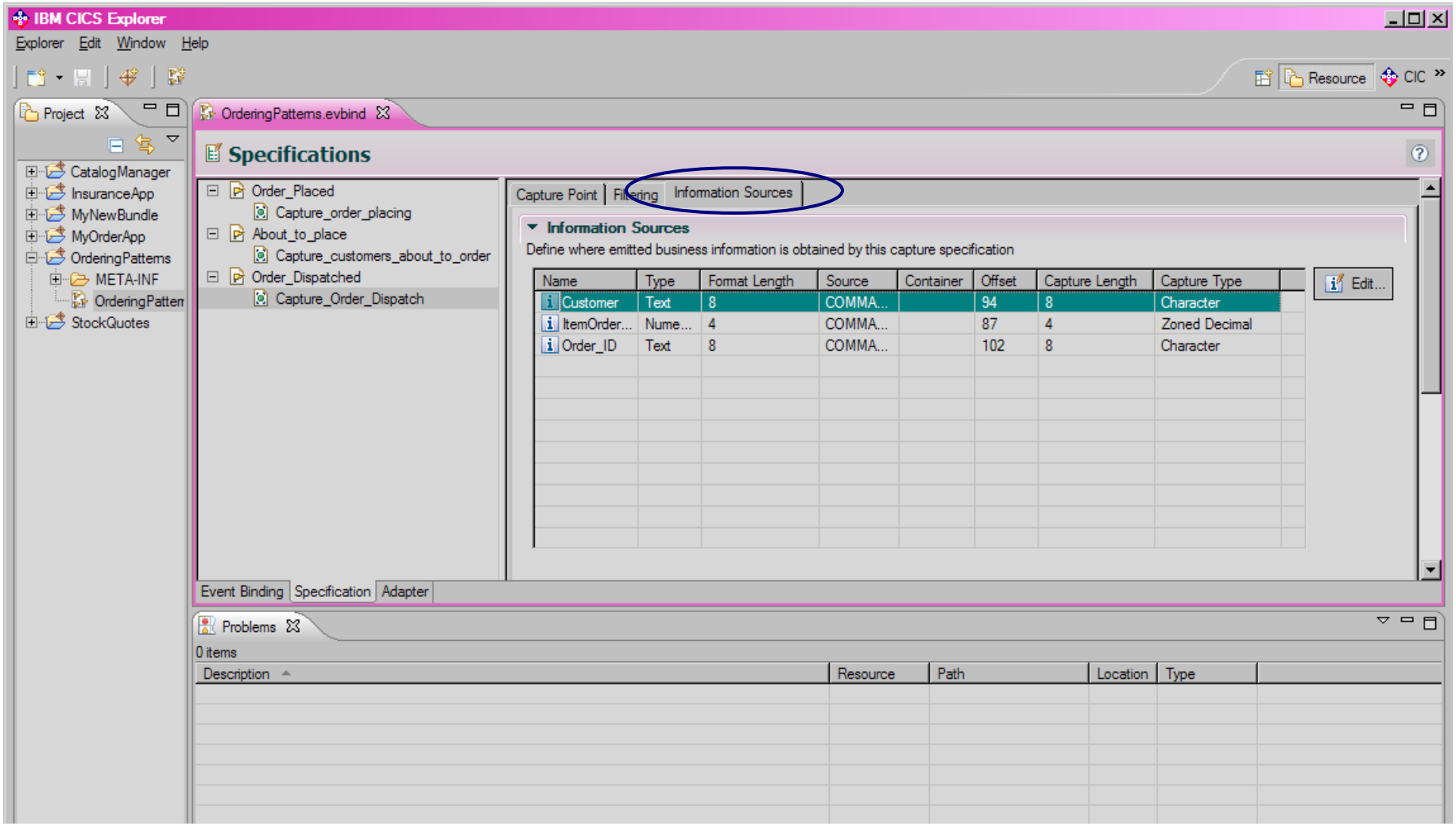
© 2009 IBM Corporation

Event Binding Editor – Filtering (2) – Notes

- This is a more advanced filtering example, and is taken from the CatalogManager bundle that is shipped with CICS to get you started with CICS events.
- This event is the Catalog_Stock_Status_Check_Event, and the capture specification (as its name implies) has REWRITE as the capture point.
- In this example, we check not only that the rewrite is to a file called EXMPCAT with response code of Ok, but also that the rewrite is from program DFH0XVDS and running under transaction EGUI.
- This example also shows filtering on values within the data passed on the command. This event is to be emitted if an order is placed when the current stock levels for the item have fallen below 24 and the amount of stock on order for the item is zero. These pieces of information are available in the record that is rewritten to the file, i.e the FROM area.

© 2009 IBM Corporation

Event Binding Editor – Information Sources



© 2009 IBM Corporation

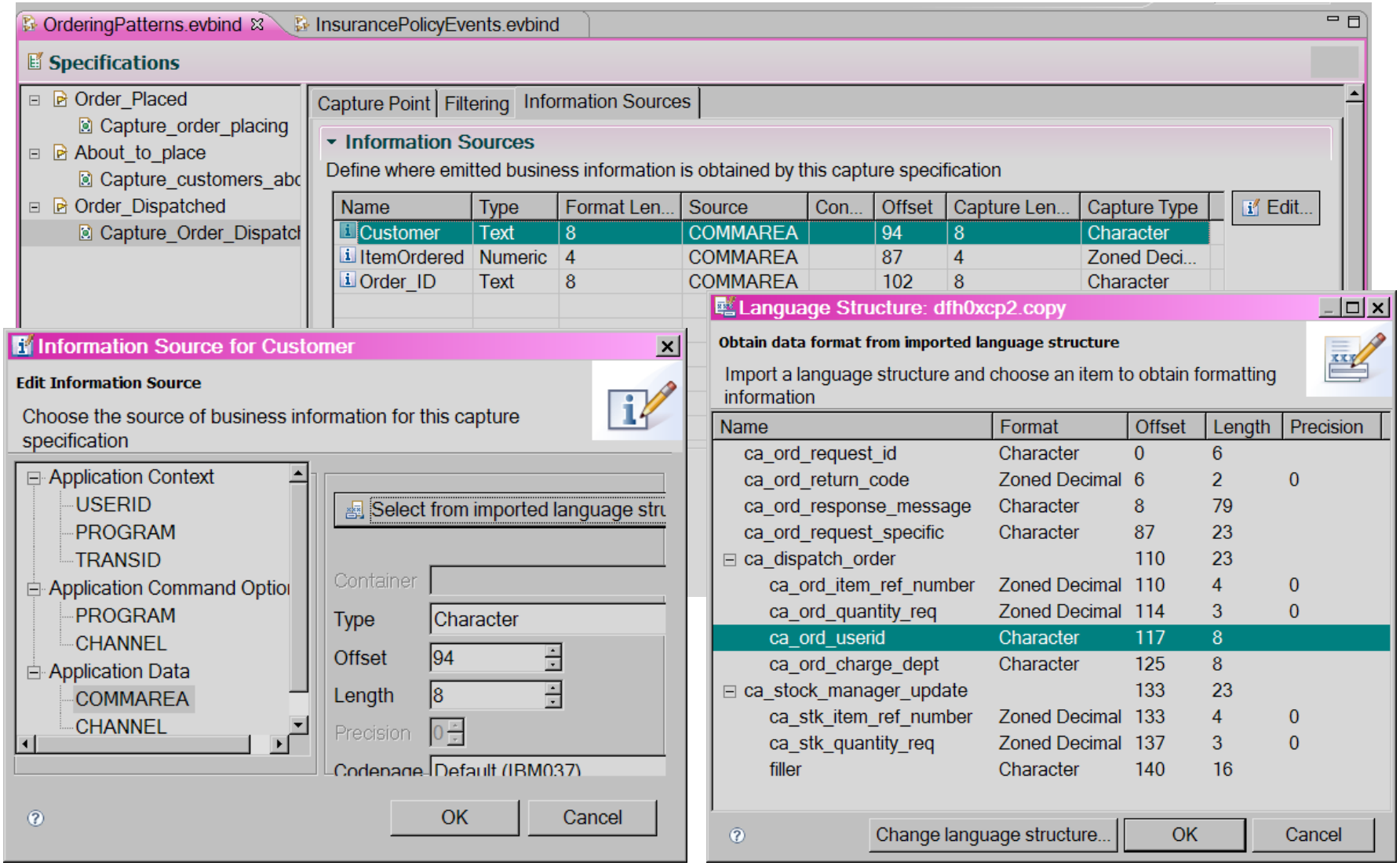
Event Binding Editor – Information Sources – Notes

- The third tab in the capture specification is for defining the source of the information which is to be emitted in the event. The table is pre-filled with the business information defined in the event specification, including its Format Length (the length it is to be formatted as in the output), but this table needs to be completed with the source of this information from data available to the application, including its data type and Capture Length.
- The source can be from application context data, such as the userid, application command data, or application data.
- In this example, all three items of information (Customer, ItemOrdered and Order_Id) are obtained from the COMMAREA passed on the LINK command, at the various offsets at which they are available.
- It is possible in the event specification to give a length of zero for an item of emitted business information. This length is known as the format length, and a value of zero indicates that the length to be emitted should be taken from that specified in the capture specification (or the first capture specification if there is more than one).
- It is also possible in the Information Sources table to specify a Capture Length of zero, which indicates that the entire data area (such as a container) is to be captured regardless of its length.

© 2009 IBM Corporation

Event Binding Editor – Information Sources Import

Showing use of imported source code (copybook) to fill in the Information Sources



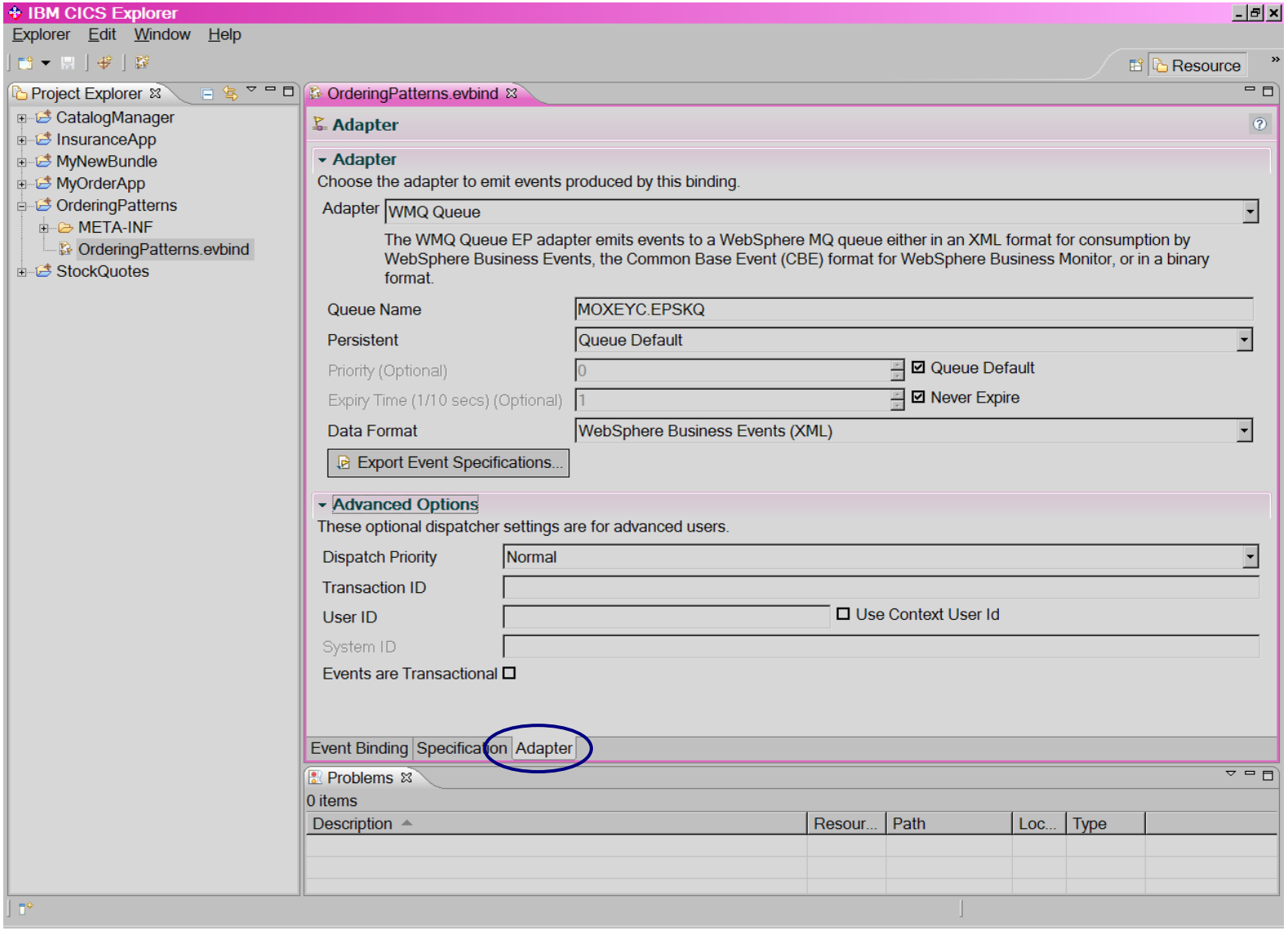
© 2009 IBM Corporation

Event Binding Editor – Information Sources Import – Notes

- To fill in the Information Source detail, you select the item and use the Edit button. This presents an 'Edit Information Source' dialog which shows the available data sources for the command used as the capture point.
- The example shows how the value for the 'Customer' can be obtained. The 'Information Source for Customer' dialog allows the value to be obtained from application context data, the command options (PROGRAM or CHANNEL names in the case of an EXEC CICS LINK command) and application data, which can be in either a COMMAREA or CHANNEL for a LINK.
- In this example, the COMMAREA data has been selected, and then the option to 'Select from imported language structure' has been used. The slide shows that a language structure (in this case, a Cobol copybook) has been imported into the tooling, and the required data item from this language structure has been selected. This fills in the offset, type, and length of the item in the table of Information Sources.
- The option to select from imported language structure can be very convenient, and it can even be worth creating a copybook in order to use this, if the application does not define the data in a copybook (for example, the structure is just defined within the program which uses it).

© 2009 IBM Corporation

Event Binding Editor – EP Adapter



© 2009 IBM Corporation

Event Binding Editor – EP Adapter – Notes

- Returning to the tabs relating to the overall event binding, we need to fill in the EP Adapter details.
- On this panel, the EP adapter which is to emit the event is selected (WMQ Queue is shown), the event format to be used (the XML-based WebSphere Business Events format is shown), and configuration relevant to this EP adapter, such as the WMQ queue name and properties.
- There are also some Advanced Options, which are initially collapsed in the event binding editor, as you do not always need to specify anything other than the defaults.
- If the 'Events are Transactional' box is checked, then events in this binding will not be emitted until the unit of work in which they occurred reaches syncpoint and commits. If the unit of work rolls back or abends then the events will not be emitted.

© 2009 IBM Corporation

Deploying an Event Binding

- **Export bundle from CICS Explorer Resource Perspective to zFS, where it should be unpacked**
- **Create BUNDLE resource definition with the location of the bundle directory on zFS as BUNDLEDIR**
  - via CICS Explorer, CICSPlex SM, CEDA etc.
- **Install BUNDLE resource**
  - Will install the event binding(s) in the bundle
  - Error is reported if the event binding fails validation checks carried out at install
    - Most validation errors will be prevented by Event Binding Editor

© 2009 IBM Corporation

Deploying an Event Binding – Notes

- This slide summarises the steps involved in deploying event bindings that have been created using the Event Binding Editor into CICS.
- Event bindings are created within a CICS Bundle Project in CICS Explorer, and the CICS Explorer provides support for exporting the bundle to an expanded directory on zFS.
- The bundle is defined as a BUNDLE resource to CICS, with a BUNDLEDIR parameter which specifies the location of the top-level directory for this bundle on zFS.
- When a BUNDLE resource is installed into CICS, any event bindings which it contains will be installed, and (providing the STATUS of the BUNDLE resource is ENABLED) will be enabled.
- It is possible for the install of an event binding to fail due to some validation which is carried out by CICS at install time. The main reason for this validation is for event binding files which have not been created using the event binding editor, since the event binding editor tooling carries out validation checks, including a check against the event binding schemas each time the event binding is saved in the editor.

© 2009 IBM Corporation

Customization

Customization options with CICS events

© 2009 IBM Corporation

Custom EP Adapter

- **CICS transaction – intended to be supplied by user or 3rd party**
- **Attached whenever CICS captures an EP event using an event binding that specifies 'Custom (User Written)' and names the transaction in the EP adapter specification**
- **Takes a newly captured CICS Event and performs the initial processing to prepare it for consumption, such as:**
  - Formatting and routing to an event consumer outside CICS
  - Formatting and routing to an event consumer inside CICS
  - Enriching or filtering event with information known to custom EP adapter
- **Should be threadsafe**
- **Should not carry out extensive processing such as would be the role of the event consumer**

© 2009 IBM Corporation

Custom EP Adapter – Notes

- It is possible to emit events via a custom EP adapter. This facility makes it possible to create events in formats other than those supported by the CICS-provided EP adapters, and to route events over other transports (such as a different messaging provider from WebSphere MQ).
- A custom EP adapter might also be used to augment the event with information which is not available at the point the event is captured.
- The custom EP adapter is specified in the Adapter properties of an event binding, so that all the events within the binding will be processed by that custom EP adapter. It is possible to pass configuration information to the custom EP adapter by specifying it in the adapter properties.
- A custom EP adapter is purely intended to prepare the event for emission from (or within) CICS. The actual consumption of that event would not be the role of the custom EP adapter.
- It is anticipated that custom EP adapters will be provided by ISVs to support additional event formats and transports, and also potentially by application providers for use with events that can be captured in their application.

© 2009 IBM Corporation

Custom EP Adapter – Handling of context data

- **CICS-provided EP Adapters process data from the application context as follows:**
  - Included as standard part of event (location depends on event format)
    - Business Event Name
    - Event binding name, Event binding user tag, Capture Specification Name
    - Network unit of work ID, Network qualified applid and CICS applid
    - Date and Time of Day
  - Included in event data if used as an information source
    - Tranid, Current Program, Userid
  - Not included in emitted event
    - Command response
- **Custom EP adapter can follow the same convention, but also has opportunity to use context information in a different way**

© 2009 IBM Corporation

Custom EP Adapter – handling of context data – Notes

- Writing a custom EP adapter provides an opportunity to process the standard application context information (passed in the DFHEP.CONTEXT container) in a different way from the CICS-provided EP adapters.
- The CICS-provided EP adapters follow the convention that some of this application context is automatically included in the emitted event. The location of that information depends on the format of the event; for example, in the CBE format, some of this information is included in standard CBE elements, and the rest is in the CICS 'static' portion of the xs:any slot in the CBE.
- There is also some application context information which the CICS-provided EP adapters only include in the event if it has been selected in the capture specification as part of the event payload; for example, the userid is used to provide a customer identifier.
- Some application context information (e.g. the command response) is never included in an event formatted by a CICS-provided EP adapter.
- A custom EP adapter could, for example, include the tranid automatically in events it emits, rather than only when selected as an item of business information.

© 2009 IBM Corporation

Event Processing SPI and Identifying Event Capture points

© 2009 IBM Corporation

Enabling and Disabling Events

- **Events can be enabled and disabled**
  - Via enable/disable of individual Event Bindings
    - INQUIRE/SET/DISCARD EVENTBINDING
  - Disable or enable event processing globally (enabled by default)
    - SET EVENTPROCESS EPSTATUS(STARTED ¦ DRAIN ¦ STOPPED)
    - INQUIRE EVENTPROCESS EPSTATUS(STARTED ¦ DRAIN ¦ STOPPED)
  - Also via Explorer, CICSPlex SM, CEMT, CECI

© 2009 IBM Corporation

Enabling and Disabling Events – Notes

- Related groups of events can be enabled and disabled by enabling and disabling installed event bindings.
- Event bindings can also be inquired upon and discarded.
- Event processing can be enabled or disabled globally using the EVENTPROCESS status.
- An EPSTATUS of STARTED enables event processing in the system. DRAIN causes event capture to stop but allows events which have already been captured to be processed through the system and emitted (transactional events will not be emitted if the unit of work had not reached syncpoint at the time the DRAIN request is received). STOPPED causes event processing to stop immediately, with no further events being captured or emitted.
- In addition to the SPI, management of eventbindings and of the EPSTATUS for eventprocessing is available via the normal mechanisms, including using the CICS Explorer.

© 2009 IBM Corporation

INQUIRE CAPTURESPEC

- **EVENTBINDING can contain several CAPTURESPECs**
- **INQUIRE CAPTURESPEC() EVENTBINDING()**
  - Can browse all the capturespecs in an eventbinding
  - Returns:
    - CAPTUREPOINT()
    - CAPTUREPTYPE (PRECOMMAND ¦ POSTCOMMAND ¦ PROGRAMINIT)
    - EVENTNAME()
- **Also via Explorer, CICSPlex SM, CECI**

© 2009 IBM Corporation

Inquire CaptureSpec – Notes

- Each EventBinding contains one or more CaptureSpecs. The CaptureSpec specifies the event capture criteria (conditions under which an event is to be captured and data required when the event is captured). Each CaptureSpec relates to a capture point e.g LINK_PROGRAM or PROGRAM_INITIATION and is of a particular type e.g precommand, postcommand, programinit.
- Capture specifications are not standalone resources, they only exist in the context of an eventbinding, and their names are only unique within the scope of an eventbinding name. The eventbinding name is required when inquiring on a capturespec.
- The SPI allows the user to query these attributes:
  - INQUIRE CAPTURESPEC() EVENTBINDING() CAPTUREPOINT() CAPTUREPTYPE() EVENTNAME().
- Browse is supported.
- This is also supported by CECI, the CICSPlex SM WUI and the CICS Explorer.

© 2009 IBM Corporation

Finding Capture Points – CICS Interdependency Analyzer

- **Use CICS IA to discover**
  - Which transactions update which resources
    - e.g. where is my PIN number file updated?
    - where are order numbers read from the TS queue?
  - The runtime flow
    - e.g. which program is linked to when an insurance quote is requested?
- **Ensure capture specification filters are sufficiently specific**
  - How many programs use this resource?
- **Can also use CICS IA Plugin to CICS Explorer**

© 2009 IBM Corporation

Finding Capture Points - CICS IA – Notes

- The CICS Interdependency Analyzer can be used standalone, or as a plugin to CICS Explorer, to investigate current processing and help identify potential events.
- The slide discusses some of the information provided by CICS IA which can be used to achieve this.

© 2009 IBM Corporation

Finding Capture Points – IBM WebSphere Studio Asset Analyzer

- **WSAA**
  - Scans and analyzes source code
- **Can for example**
  - List CICS programs
  - List transactions and their flow
  - View the transaction flow
- **IBM Rational Asset Analyzer is also available**
  - Scans mainframe and distributed software assets that are downloaded to Windows or Linux system

© 2009 IBM Corporation

Finding Capture Points - WSAA – Notes

- WSAA provides source code scanning and analysis. It can provide a static view across all relevant source, and complements CICS IA which provides a runtime view of how programs and resources are actually used, but only during the time when the data was collected.

© 2009 IBM Corporation

CICS and Events 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 consumers including** WebSphere Business Events **and** WebSphere Business Monitor
  - CICS-provided EP adapters plus capability for custom EP adapters

© 2009 IBM Corporation

CICS and Events Summary – Notes

- 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, without requiring any changes to the application programs.
- A new SIGNAL EVENT API is also provided, to add explicit event-enabling points into applications, where such flexibility is required.
- 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.

© 2009 IBM Corporation

References

- **CICS TS V4.1 Announcement letter**

  http://www.ibm.com/software/htp/cics/tserver/v41/

- **CICS TS V4.1 Information Center**

  http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp

- **CICS Events with WebSphere Business Events High-level**

  http://www.youtube.com/watch?v=S0orwDxSOvM

- **CICS Events 5 minute demo**

  http://www.youtube.com/watch?v=-wQhxFfmd9U

© 2009 IBM Corporation

Notices

© IBM Corporation 2009. All Rights Reserved.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. 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. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. 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 the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ibm.com/legal/copytrade.shtmlAIX, CICS, CICSPlex, DataPower, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS/ESA, Power Systems, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, Redbooks, Sametime, SMART SOA, System z , Tivoli, WebSphere, and z/OS.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce

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

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

© 2009 IBM Corporation