

# IBM CICS and Events: Concepts and Event Specifications

Pradeep Gohil  
CICS Software Engineer  
gohilpr@uk.ibm.com



IBM CICS® User Conference 2009

## Introduction to CICS and Event Processing – Notes

- This material provides an introduction to new CICS capability in CICS TS V4.1 to participate in Event Processing solutions, and for CICS applications to act as a source of business events to consumers including IBM event processing products, with a focus on the use of WebSphere MQ to emit events to consumers outside CICS TS.
- The following are thanked for their help with this material: Chris Backhouse, Brian Jones, Mark Cocker, Ann Collins, Adam Coulthard, Gillian Curwen, Tom Grieve, Ken Porter, Dan Millwood, Anna Maciejkovicz, and many others in the CICS event processing team. Also Latha Sivakumar and Yuan Yuan from the WebSphere Business Monitor team, and Peter Crocker, James Taylor and Mark Hiscock from the WebSphere Business Events team.
- The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM trademarks, see [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)  
AIX, CICS, CICSplex, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS, iSeries, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, RCAF, Redbooks, Sametime, Smart SOA, System i, System i5, System z , Tivoli, WebSphere, and z/OS.

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.

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.

Other company, product, or service names may be trademarks or service marks of others

# IBM CICS® User Conference 2009

© 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.shtml](http://ibm.com/legal/copytrade.shtml)AIX, 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](http://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.

# Topics

- **CICS and Event Processing**
  - Introduction to event processing
  - Why and How
  - A few example scenarios
- **Event Specifications**
  - Event Capture
- **Event Processing Adapters**
- **Event Binding Editor**
  - CICS EP Tooling
- **Using CICS Events**
  - Integration with WebSphere Business Monitor
  - Integration with WebSphere Business Events
- **Summarizing scenario, Summary and Q&A**

# CICS and Event Processing

An introduction to CICS as a source of business events

# CICS and Event Processing – Notes

- This section introduces the core concepts of CICS event processing support.

# What is an event?

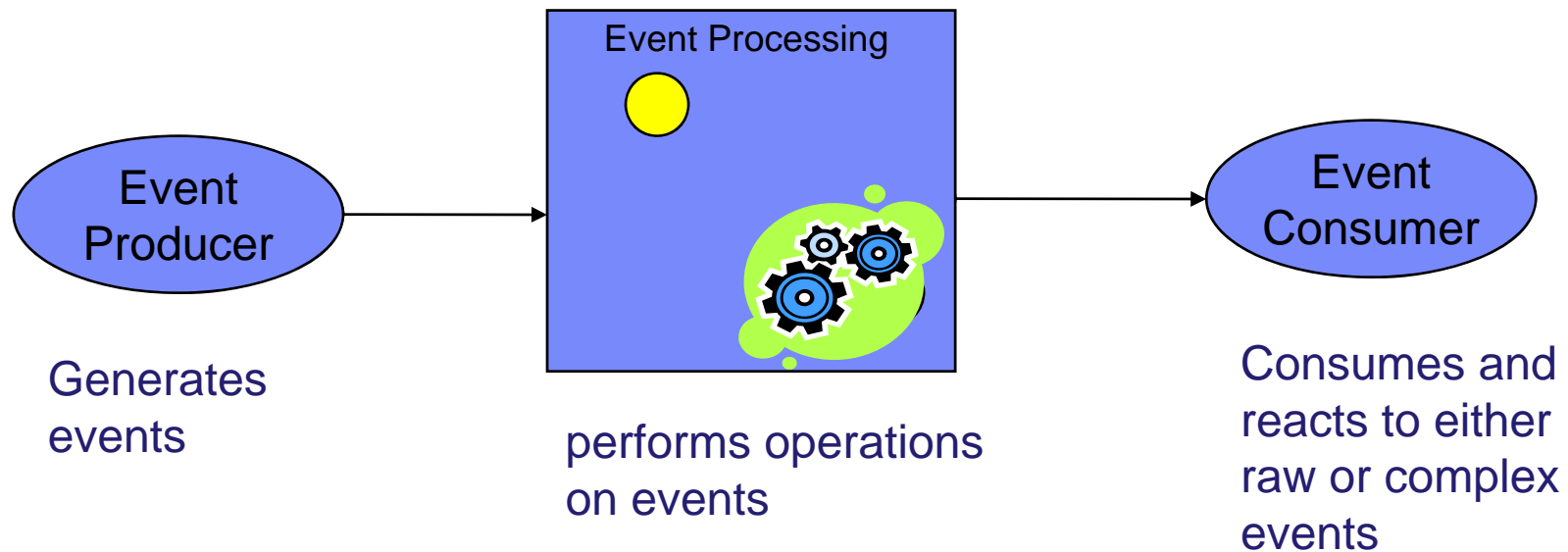
- **An event is**
  - *Anything that happens (or is contemplated as happening)*
  - An event has a name and usually some data (its payload)
  - Produced and responded to asynchronously
- **Simple event**
  - A single event, meaningful in itself
  - e.g. order placement; bank account update; stock trade
- **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. 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.
- “Business events processing is becoming more important as it enables companies of all sizes and in all industries to proactively analyze and respond to minute market changes that can significantly impact their businesses” – Tom Rosamilia



# Event Processing in a Nutshell



# Event Processing in a Nutshell – Notes

- 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. date/timestamp it, add information about the source of the event (producer, etc.)), 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 may be very simple and just update a database or a visual dashboard with the data carried with the event, or it may 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

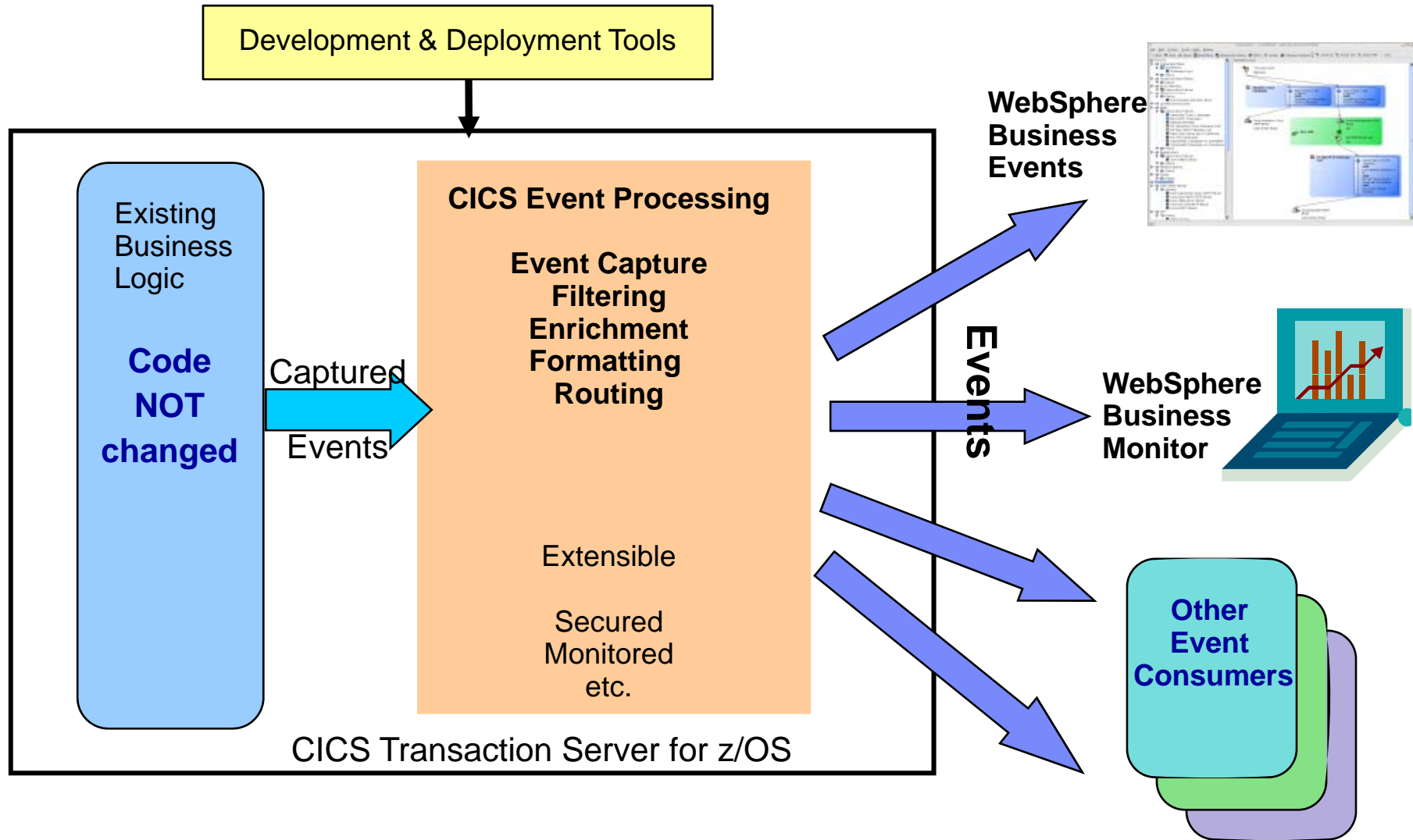


- **Event processing addresses the need for agility**
  - Modern businesses must react quickly to circumstances
  - Decision makers need reliable, timely information
- **CICS systems run an enormous amount of existing business logic**
- **Using an Event-based approach, there is potential to gain insight 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 and event processing Overview



# 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.

# A few example uses of CICS events

## A few example uses of CICS events – Notes

- This section shows a few of the ways in which CICS events can be used.



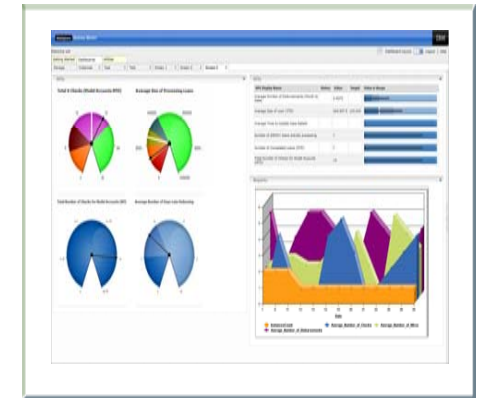
### EP Scenario 1 – Observe business processing

- **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

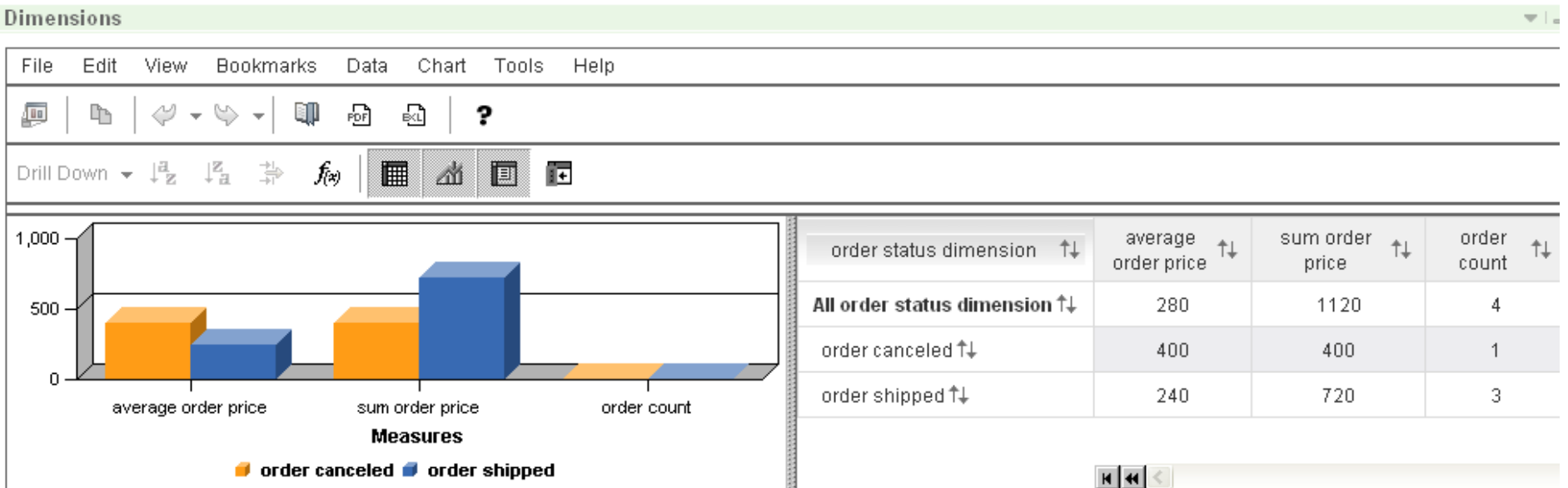
### EP Scenario 1 – Observe business processing – Notes

- 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.

## EP Scenario 1 – KPIs and Dimensions in Business Space

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

| KPIs                          |        |                |               |   |   |
|-------------------------------|--------|----------------|---------------|---|---|
| KPI Name                      | Status | Value          | Target        | Actions   | Value in Range  |
| average time on order shipped |        | 23 m, 18.601 s | 1 h, 0 m, 0 s |    |  |
| percent of order cancelled    |        | 25.00%         | 30.00%        |    |  |



## 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 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

## 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.

## 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**

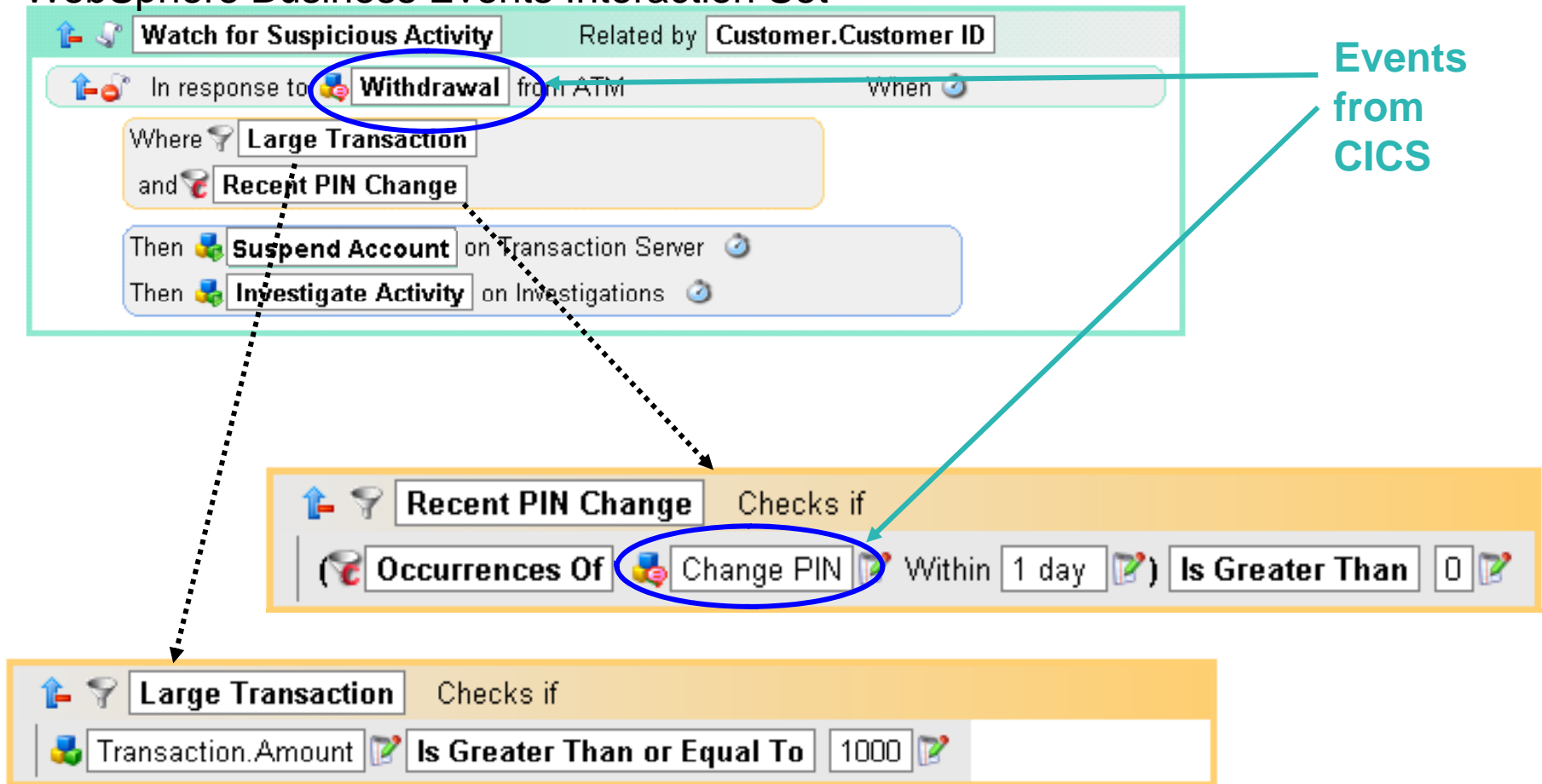
## EP Scenario 3 – Event combination – Notes

- 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.



# EP Scenario 3 – Detecting event pattern using WebSphere Business Events

## WebSphere Business Events Interaction Set



## 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 to watch for a suspicious activity.

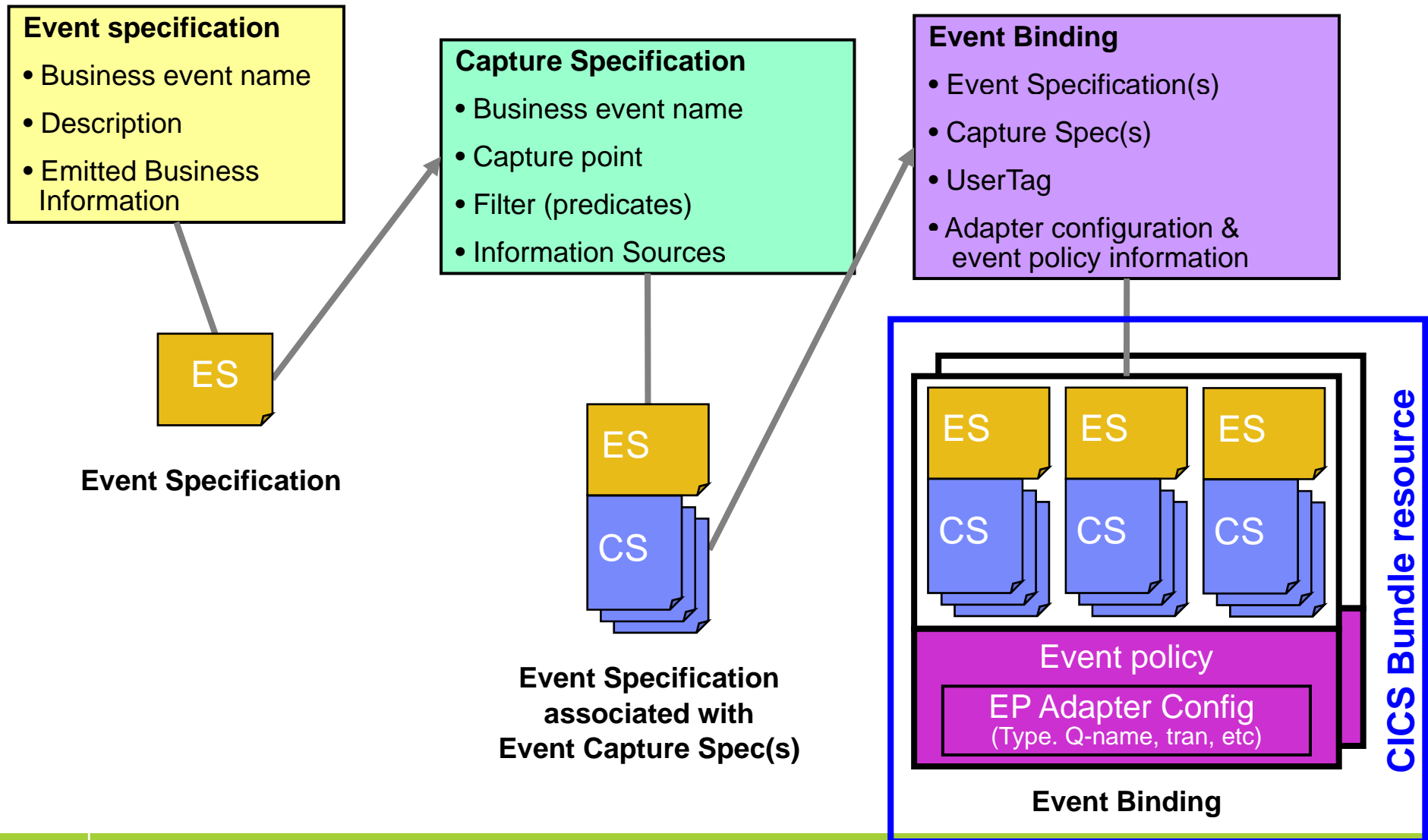
# CICS Event Specifications

or... how CICS knows where the events occur

# 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.

# CICS Event Specification



## 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). A bundle can contain several event bindings (for example, all relating to an application) along with other resources.

# 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

*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; the capture spec 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

## 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.



# CICS Event Capture

## CICS Event Capture – Notes

- This section 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.

# 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

## 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.

# 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 transid 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 the additional application context
    - **but** only limited information about the command could be available to CICS & included in capture specs

## 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.

## 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

## 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.



## 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-qualified CICS applid, Date & time
    - also: Event name, Event binding name, User tag, Capture spec name
  - 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\*, 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\*, Data from channel or Commarea
- **Most commands captured *after* they occur, some offer option to capture *before* e.g. LINK**

## 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. Also included in the standard contextual information are the event name, the name of the capture specification which caused the event to be captured, the name of the event binding containing that capture specification, and the event binding user tag.
- 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.

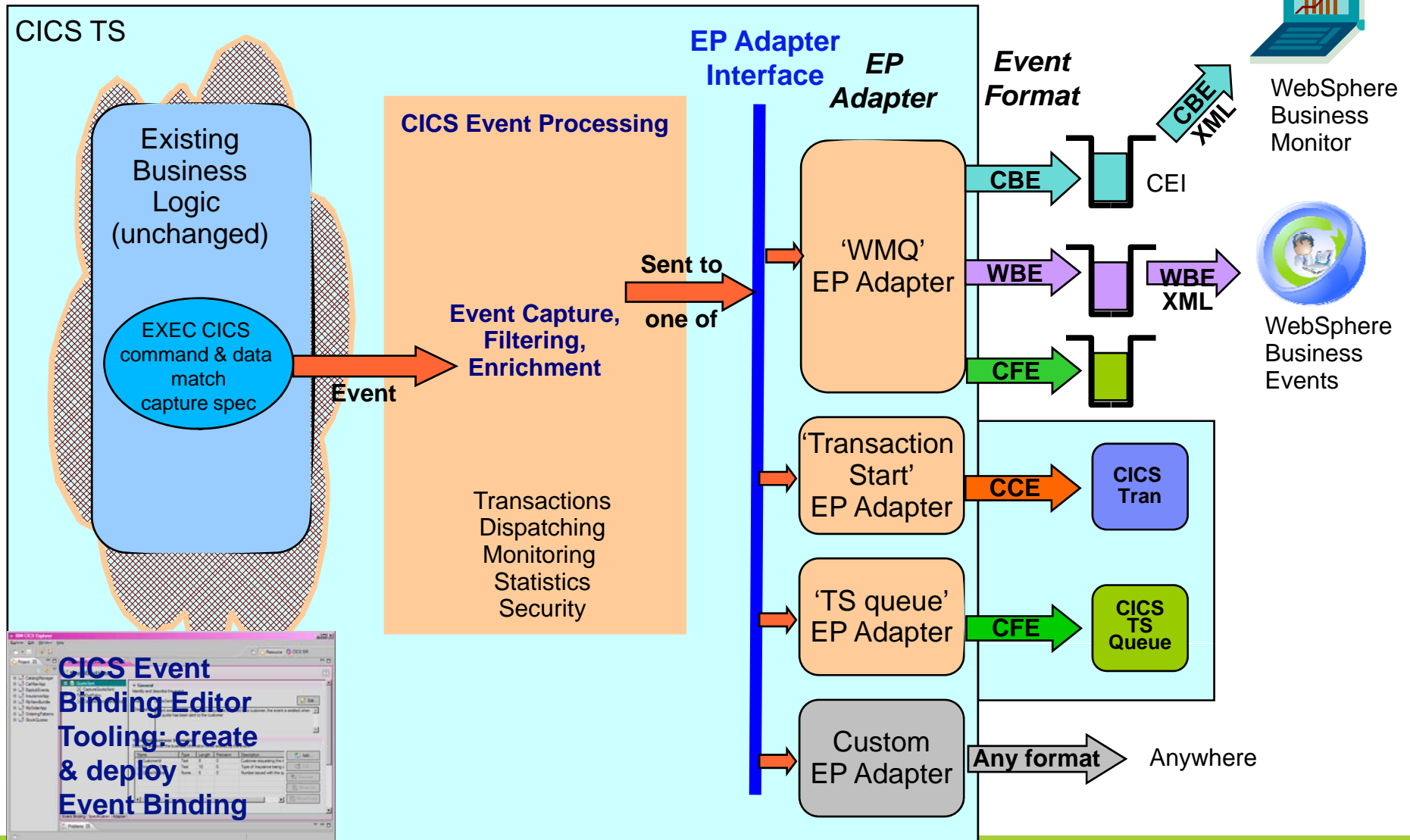
# CICS Event Processing Adapters

Formatting and Transport of CICS events

## Event Processing 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. The EP adapter configuration applies to all of the events within an event binding, as do the event processing policy options which are also described in this section. The event binding provides a way to group together events which are to be formatted and routed in the same way (e.g. because they are to be consumed by the same event consumer).

# CICS Event Processing Adapters



## 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 (note this is a system programmer interface, which may change from release to release).
- 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.
- 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, or other products which use the Common Event Infrastructure (or which can receive the event in CBE format)
  - WMQ transport, WBE XML format – primarily for sending events to WebSphere Business Events (or program which can consume the WBE event format)
  - WMQ transport, CFE (“CICS Flattened Event”) text-based format – to be read from the queue by a consuming application, either within CICS or outside
  - 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 events are emitted when expected and contain the correct data

# 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

## 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 place, but are emitted if and when the unit of work in which the event took place 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.



## 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 of specifying tranid or userid**

## 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).

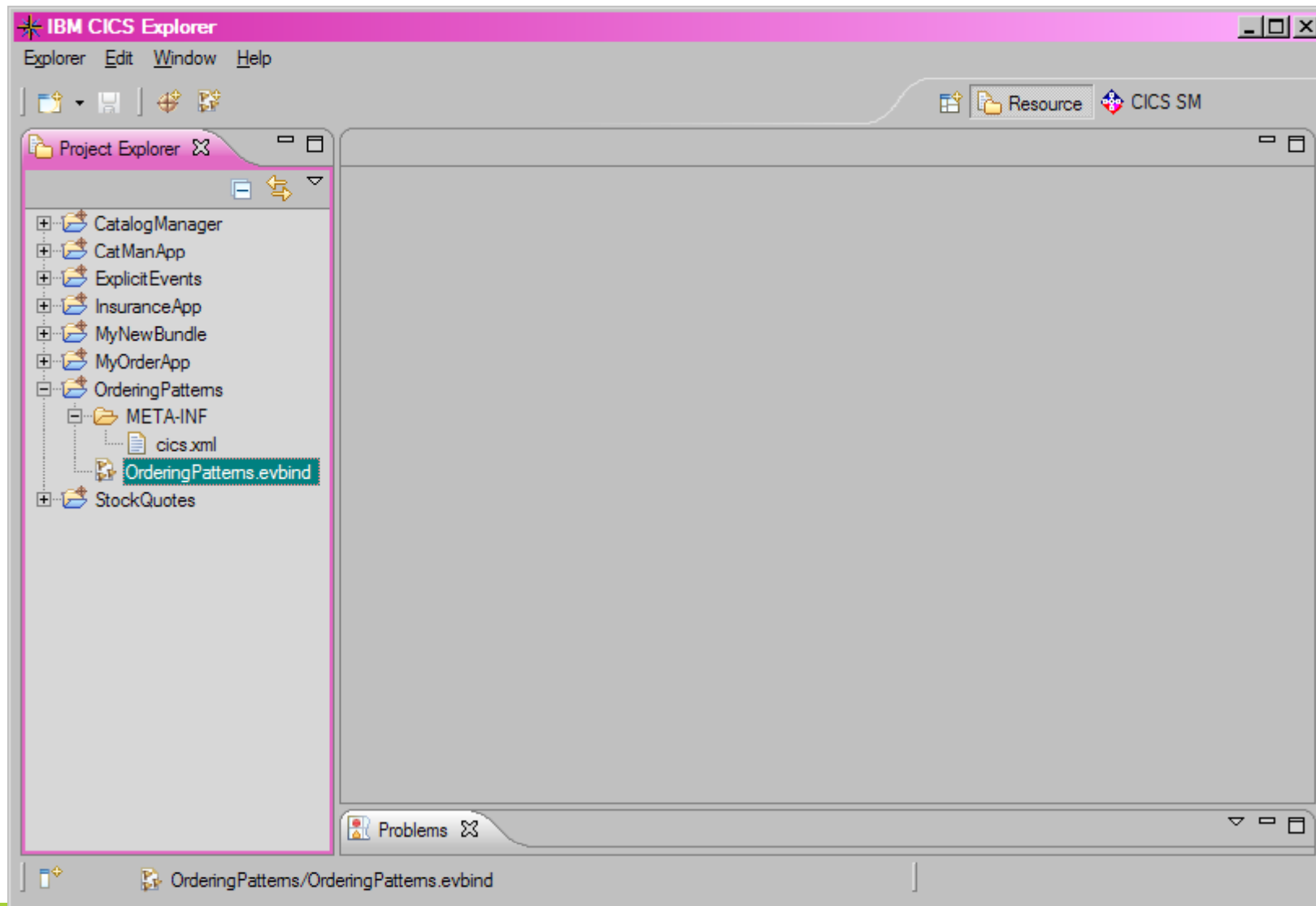
# Event Binding Editor

The CICS event specification tooling

# 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.

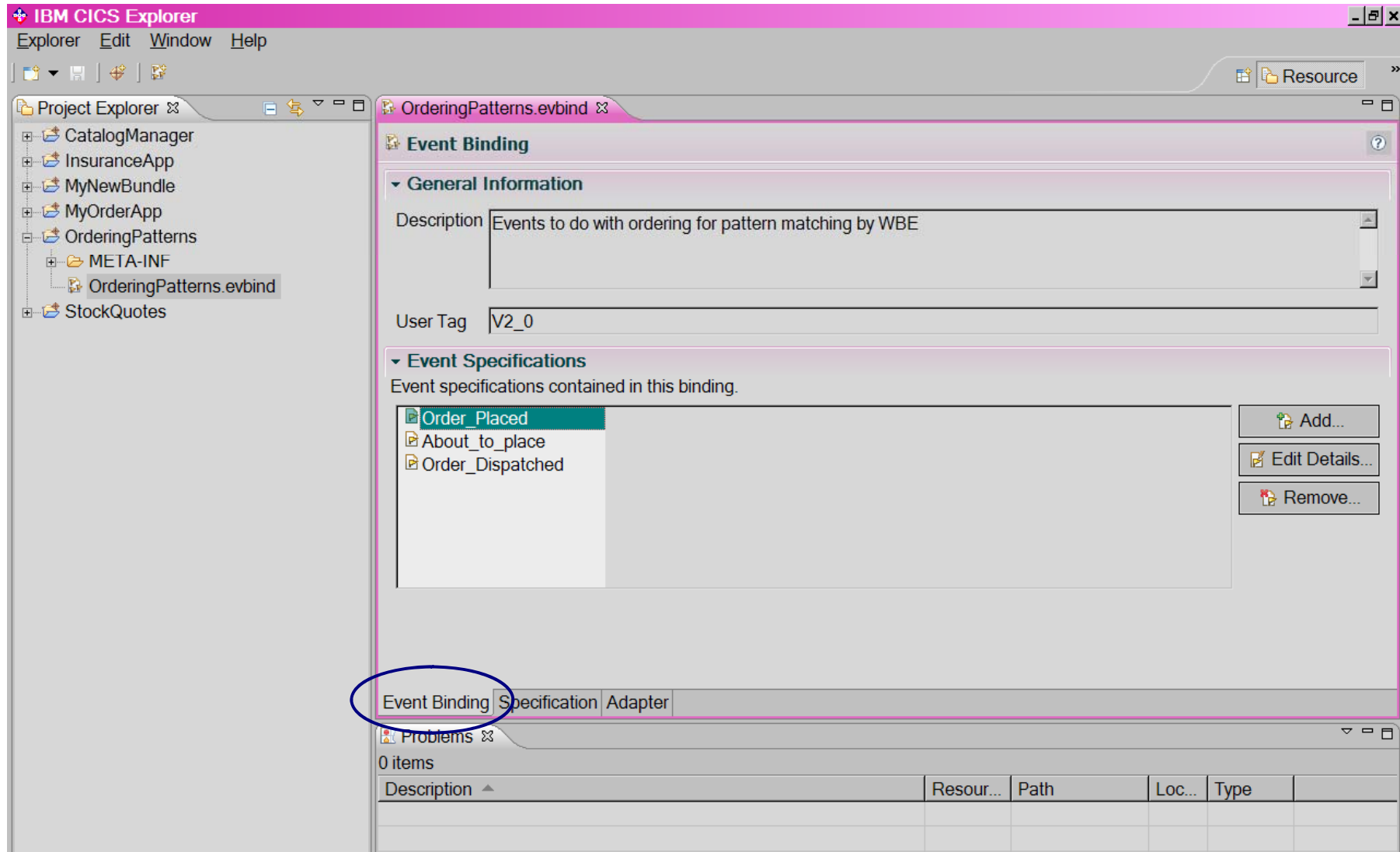
# CICS Explorer Resource Perspective



# 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).

## Event Binding Editor – Event Binding



# 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.



## Event Binding Editor – Event Specification

The screenshot displays the IBM CICS Explorer interface for editing an event binding. The main window shows the 'OrderingPatterns.evbind' file with the following specifications:

- Specifications:**
  - Order\_Placed
    - Capture\_order\_placing
  - About\_to\_place
    - Capture\_customers\_abo
  - Order\_Dispatched
    - Capture\_Order\_Dispatch

The 'General' tab for the selected event is shown below:

**General**  
Identify and describe the event.

Name: Order\_Dispatched [Edit...]

Description: Event indicating an order has been dispatched.

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

| Name        | Type  | Precisi... | Le... | Description                |
|-------------|-------|------------|-------|----------------------------|
| Customer    | Text  | 0          | 8     | Customer who place...      |
| ItemOrdered | Nu... | 0          | 4     | Identifier of item orde... |
| Order_ID    | Text  | 0          | 8     | Identifier of the order    |

**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...]

At the bottom of the window, the 'Specification' tab is selected, and the 'Problems' pane shows 0 items.

# Event Binding Editor – Event Specification – 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.)

## Event Binding Editor – Capture Specification

The screenshot displays the IBM CICS Explorer interface. On the left, the Project Explorer shows a tree view with 'OrderingPatterns' expanded to 'META-INF' and 'OrderingPatterns.evbind'. The main editor window, titled 'OrderingPatterns.evbind', shows a list of specifications under the 'Specifications' tab. The 'Capture\_Order\_Dispatch' specification is selected. The 'Capture Point' tab is active, showing the following configuration:

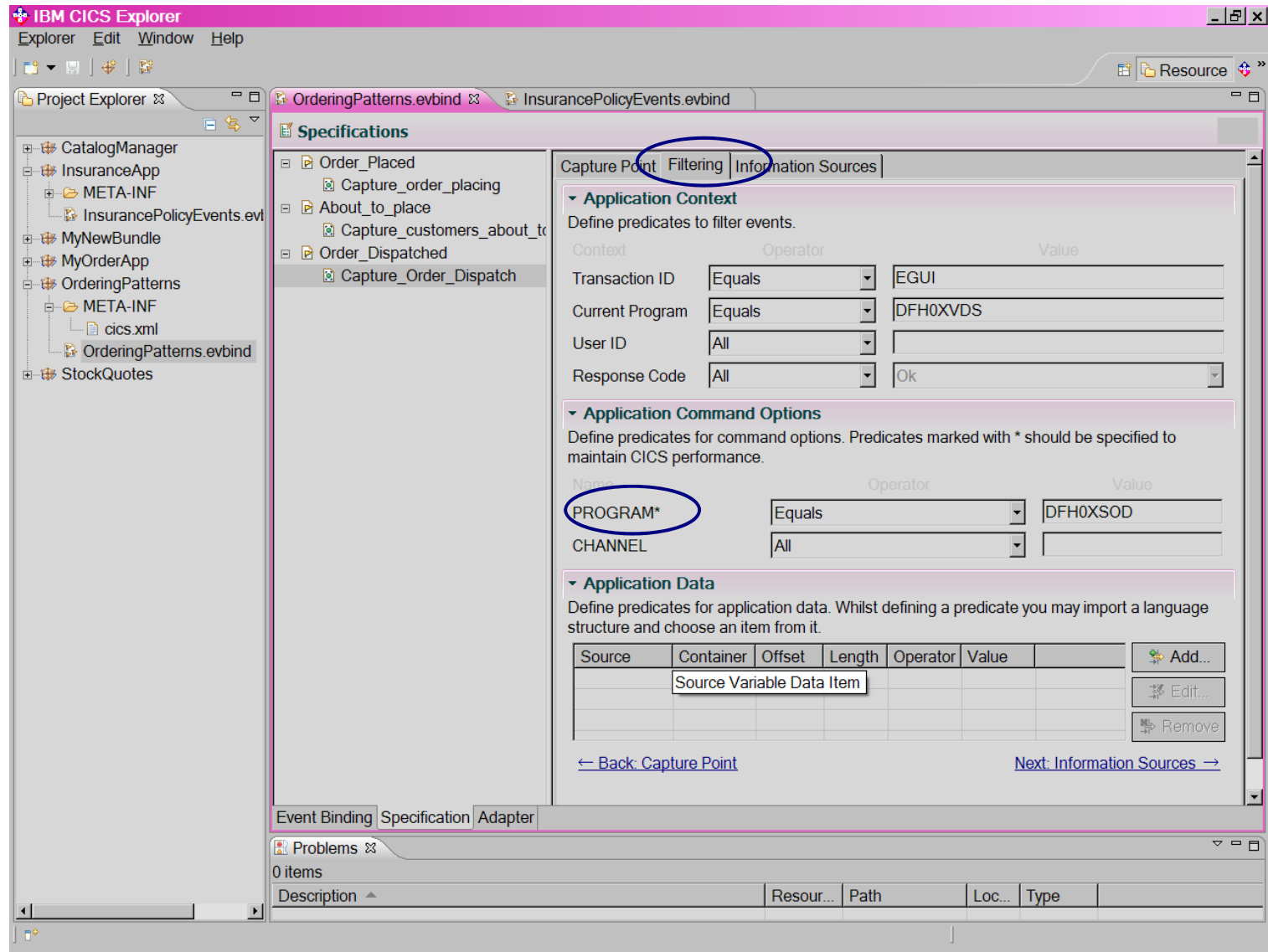
- Name:** Capture\_Order\_Dispatch
- Description:** Capture the point where the order is dispatched
- Capture Point:** LINK PROGRAM
- Options:** Capture before (unselected), Capture after (selected)

The 'Capture Point' dropdown menu and the 'LINK PROGRAM' value are circled in blue. The 'Remove Capture Specification...' button is also visible. The bottom of the editor shows the 'Problems' tab with 0 items and a table with columns: Description, Resour..., Path, Loc..., Type.

## 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 an 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.

## Event Binding Editor – Capture Specification Filtering



## Event Binding Editor – Capture Spec 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 expressions, 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.

## Event Binding Editor – Information Sources

The screenshot shows the IBM CICS Explorer interface. The main window is titled 'OrderingPatterns.evbind' and displays the 'Specifications' view. The 'Information Sources' tab is selected, showing a table of data sources for the 'Capture\_Order\_Dispatch' specification. The table has the following data:

| Name         | Type    | Format | Length | Source   | Container | Offset | Capture Length | Capture Type  |
|--------------|---------|--------|--------|----------|-----------|--------|----------------|---------------|
| Customer     | Text    |        | 8      | COMMA... |           | 94     | 8              | Character     |
| ItemOrder... | Nume... |        | 4      | COMMA... |           | 87     | 4              | Zoned Decimal |
| Order_ID     | Text    |        | 8      | COMMA... |           | 102    | 8              | Character     |

The 'Information Sources' tab is circled in blue. Below the table, there is an 'Edit...' button. The bottom of the window shows a 'Problems' view with 0 items.

# 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.



## Event Binding Editor – Information Sources Import

The screenshot shows the Event Binding Editor interface. The main window is titled 'Specifications' and has tabs for 'Capture Point', 'Filtering', and 'Information Sources'. The 'Information Sources' tab is active, showing a table of information sources for the 'Customer' capture specification.

| Name         | Type  | Format | Len... | Source   | Con... | Offset | Capture Len... | Capture Type  |
|--------------|-------|--------|--------|----------|--------|--------|----------------|---------------|
| Customer     | Text  |        | 8      | COMMAREA |        | 94     | 8              | Character     |
| ItemOrder... | Nu... |        | 4      | COMMAREA |        | 87     | 4              | Zoned Decimal |
| Order_ID     | Text  |        | 8      | COMMAREA |        | 110    | 4              | Zoned Decimal |

An 'Edit...' button is visible next to the table. A dialog box titled 'Language Structure: dfh0disp.copy' is open, showing a table of data fields from the language structure. The 'ca\_ord\_userid' field is highlighted.

| Name                    | Format       | Offset | Length | Precision |
|-------------------------|--------------|--------|--------|-----------|
| ca_ord_request_id       | Character    | 0      | 6      |           |
| ca_ord_return_code      | Zoned Dec... | 6      | 2      | 0         |
| ca_ord_response_message | Character    | 8      | 79     |           |
| ca_dispatch_order       |              | 87     | 23     |           |
| ca_ord_item_ref_number  | Zoned Dec... | 87     | 4      | 0         |
| ca_ord_quantity_req     | Zoned Dec... | 91     | 3      | 0         |
| ca_ord_userid           | Character    | 94     | 8      |           |
| ca_ord_charge_dept      | Character    | 102    | 8      |           |

The dialog box also includes a 'Select from imported language' section with fields for Container, Type (Character), Offset (94), Length (8), Precision (0), and Codepage (Default (IBM037)).

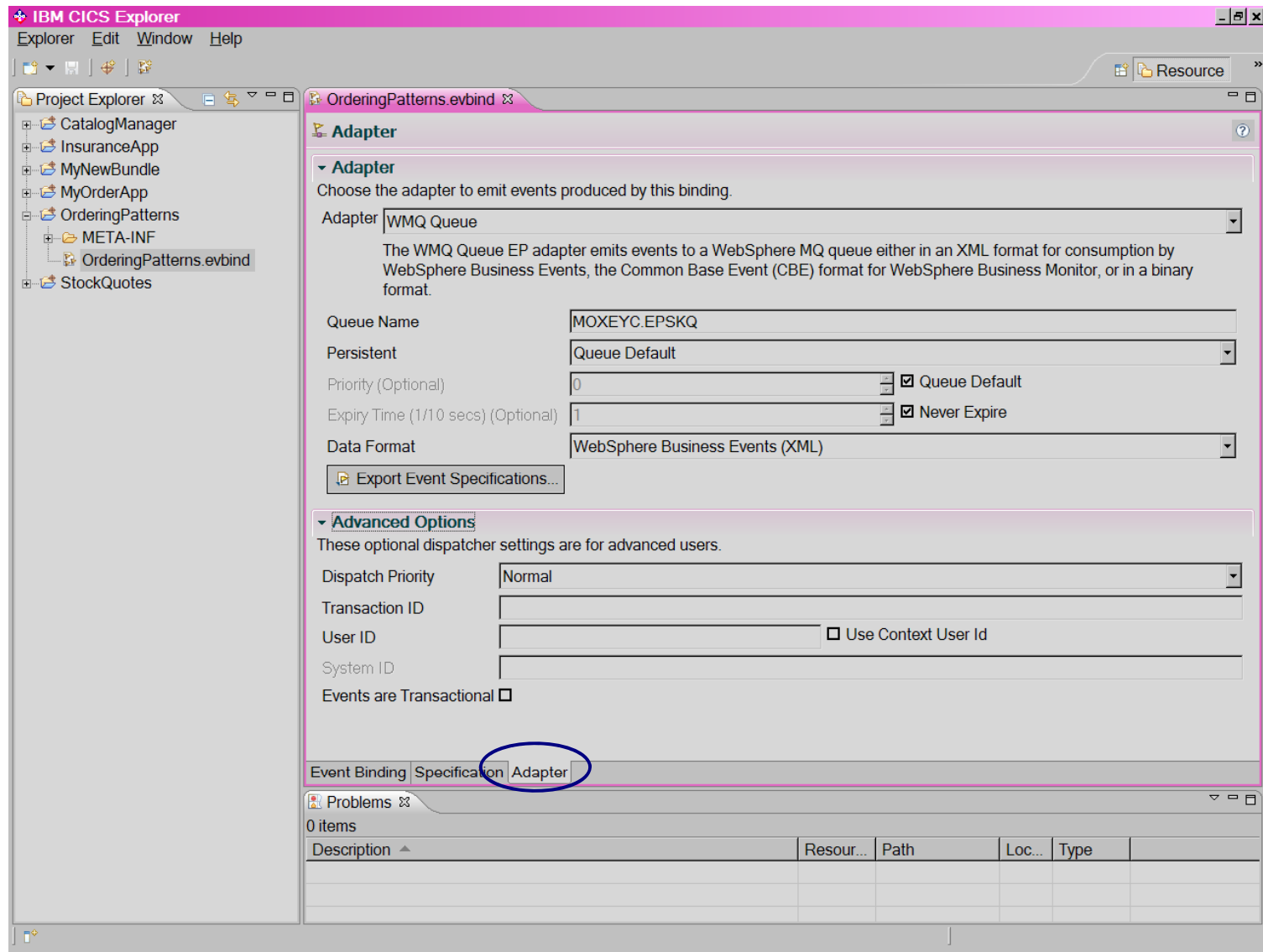
## Event Binding Editor – Information Sources Import – Notes

- 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).

## Event Binding Editor – EP Adapter

EP Adapter specification, showing the configuration details for WMQ Queue EP adapter, with Queue name and other WMQ parameters.

The Advanced Dispatcher Options are shown, but nothing other than the defaults have been selected.



## 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.

# Events in CICS Explorer

The screenshot displays the IBM CICS Explorer interface. The main window shows two views: 'Event Bindings' and 'Capture Specifications'. The left sidebar shows the 'CICSplex Explorer' tree with the server 'CIGO' and region 'IYCWZCGO (1/1)' selected.

**Event Bindings View:**

CNX02111 Context: IYCWZCGO. Resource: EVNTBIND. 5 records collected at 13-Jul-2009 17:46:46

| Region   | Name               | Status    | Bundle   |
|----------|--------------------|-----------|----------|
| IYCWZCGO | CatalogSample      | ✓ ENABLED | catbundl |
| IYCWZCGO | ExplicitEventSpecs | ✓ ENABLED | explicit |
| IYCWZCGO | EPSOLINKEvent      | ✓ ENABLED | explicit |
| IYCWZCGO | OrderingEvents     | ✓ ENABLED | MyOrderA |
| IYCWZCGO | OrderingPatterns   | ✓ ENABLED | WBEbundl |

**Capture Specifications View:**

CNX02111 Context: IYCWZCGO. Resource: EVCSPEC. 8 records collected at 13-Jul-2009 17:46:46

| Region    | Capture Specification            | Event Binding      | Capture Type | Capture Point | Event Name                 |
|-----------|----------------------------------|--------------------|--------------|---------------|----------------------------|
| IYCWZC... | Check_stock_status_on_rewrite    | CatalogSample      | POSTCOMMAND  | REWRITE       | Catalog_stock_status_check |
| IYCWZC... | CECISigEventCapture              | ExplicitEventSpecs | POSTCOMMAND  | SIGNAL_EVENT  | CECISigCommand             |
| IYCWZC... | EPSOrder_Event_Capture           | EPSOLINKEvent      | POSTCOMMAND  | SIGNAL_EVENT  | Order_Event                |
| IYCWZC... | OrderNearlyCapture               | OrderingEvents     | POSTCOMMAND  | READ          | OrderNearlyPlaced          |
| IYCWZC... | OrderPlaceCapture                | OrderingEvents     | POSTCOMMAND  | REWRITE       | OrderPlaced                |
| IYCWZC... | Capture_customers_about_to_order | OrderingPatterns   | POSTCOMMAND  | READ          | About_to_place             |
| IYCWZC... | Capture_order_placing            | OrderingPatterns   | POSTCOMMAND  | REWRITE       | Order_Placed               |
| IYCWZC... | Capture_Order_Dispatch           | OrderingPatterns   | POSTCOMMAND  | LINK_PROGRAM  | Order_Dispatched           |

## Events in CICS Explorer – Notes

- This shows the System Management perspective of the CICS Explorer, and how it can be used to look at the resources related to event processing, such as the enabled event bindings (shown in the top half of the display), the installed bundles (not shown in this display), and enabled capture specifications (in the bottom half of this display). Not shown, to the right of the capture specifications, is the number of events captured for each capture specification.

# Using CICS Events

Integration with WebSphere Business Monitor and  
WebSphere Business Events

# Using CICS Events: Integration – Notes

- This discusses integration of CICS events with other products such as WebSphere Business Monitor (WBM) and WebSphere Business Events (WBE).



## Value from CICS Events

- **Use CICS events to**
  - Monitor what is happening in the system
  - Drive additional processing as a result of events
  - Detect interesting patterns amongst events
- **to enable**
  - Business insight
  - Business flexibility and innovation
  - Regulatory compliance and management of business risk

## Value from CICS Events – Notes

- The ability to detect events happening in CICS application processing, and emit those events for consumption in a variety of ways, opens up a number of new opportunities.
- Events from CICS allow processing within CICS to be monitored, gaining insight into CICS applications and the business processes that they support, without the need to make application changes. It is also possible to extend the processing in new ways by use of CICS events, enabling flexible and timely response to business opportunities or threats. It is also possible to look for particular patterns of events, either from CICS alone or from CICS and other sources, which might indicate potential breaches of corporate, industry or government regulations or other undesirable situations.

# CICS and WebSphere Business Monitor

# CICS and WBM – Notes

- This section discusses CICS integration with WebSphere Business Monitor.

# WebSphere Business Monitor (WBM)

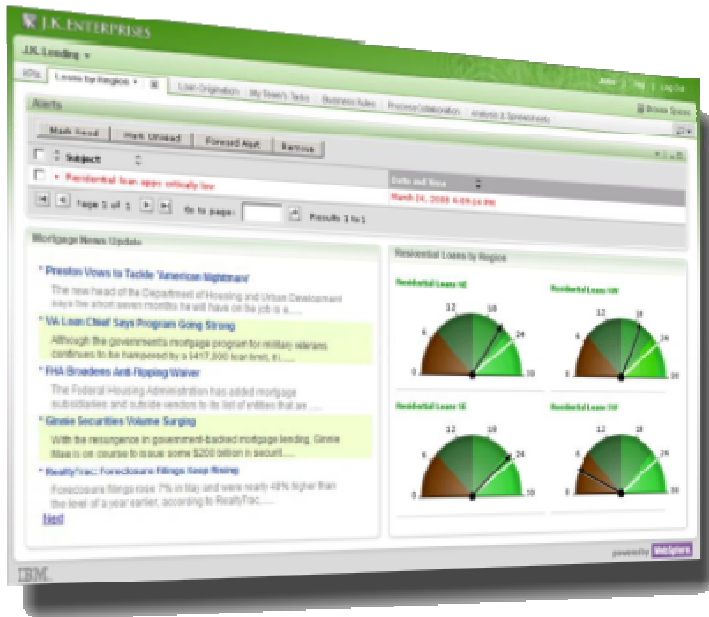
– a Business Activity Monitoring Solution

**Business activity monitoring (BAM) provides process visibility**

*Business leaders gain real-time visibility and actionable insight into processes*

**Real-time** information consolidated into customizable, role-based dashboards

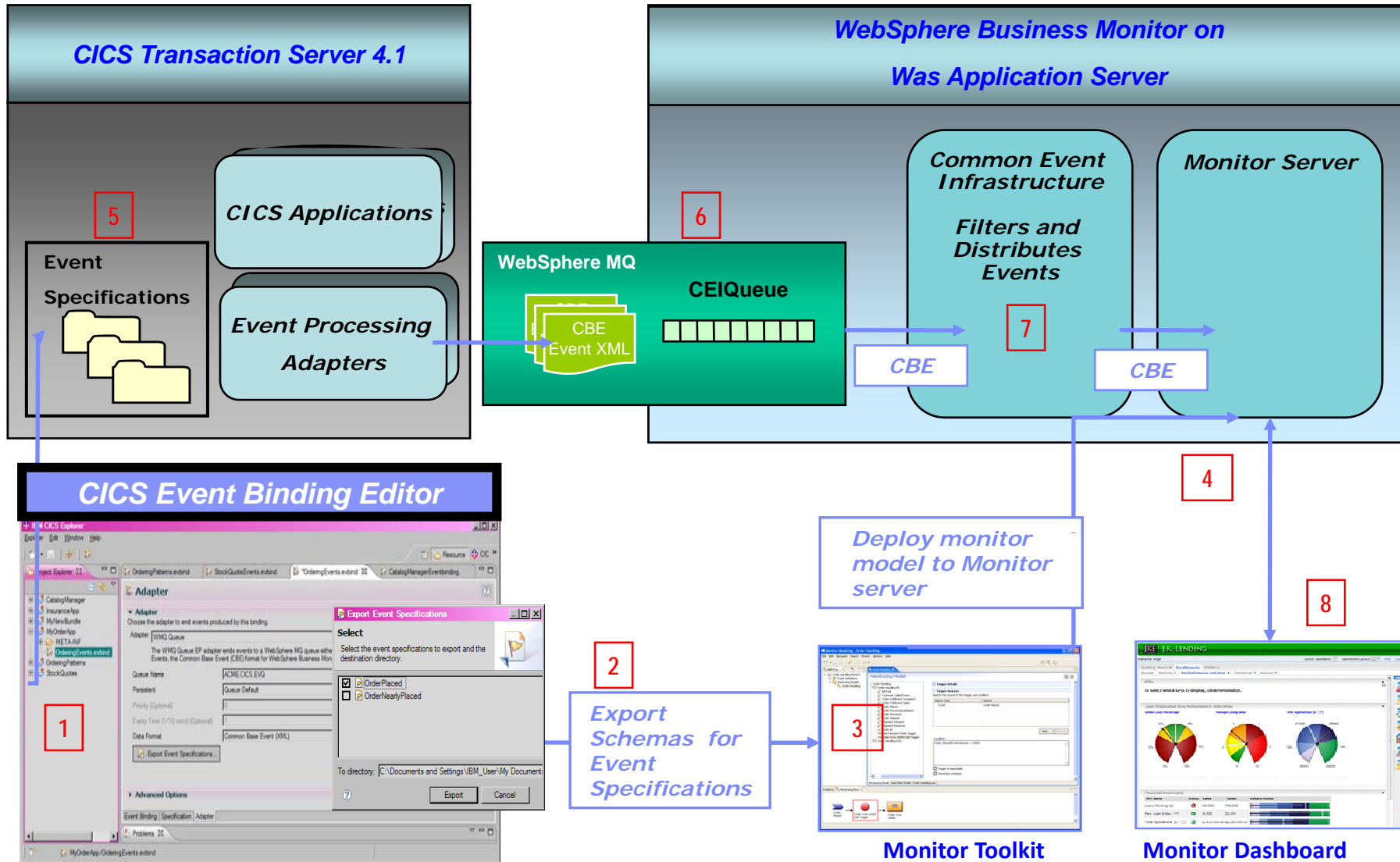
**Business leaders** monitor process KPIs and receive alerts



# WebSphere Business Monitor (WBM) – Notes

- IBM WebSphere Business Monitor is comprehensive business activity monitoring (BAM) software that provides business users and managers with a real-time and end-to-end view of business processes and operations.
- WBM is a core part of the WebSphere Dynamic Process Edition foundational offering of the IBM Business Process Management (BPM) Suite, and is also available as a standalone product.
- It provides customizable business dashboards that calculate and display key performance indicators (KPIs) and metrics derived from business processes, business activity data, and business events from a wide range of information sources.
- WBM can monitor business events from any system which can emit events in the Common Base Event format.

# CICS and Monitor Integration



# CICS and Monitor Integration – Notes

1. This slide shows how events from CICS can be integrated with WebSphere Business Monitor using WebSphere MQ and the Common Event Infrastructure (CEI).
2. The Event Binding Editor is used to create event specifications
3. Once the events have been specified in the Event Binding Editor, the dynamic schema files can be exported which describes the event payload.
4. These schemas are imported into the WebSphere Business Monitor toolkit to create Monitor Model applications that will process the incoming events.
5. The monitor models are deployed to the Monitor server.
6. Event specifications, grouped together into event bindings, are deployed to the CICS system and enable event emission.
7. When the event binding is enabled, then events will be emitted each time the set of conditions defined by the event specification is matched, and will be sent over WebSphere MQ to the Common Event Infrastructure (CEI).
8. CEI will distribute the event to Monitor server.
9. The events are processed by the monitor server and can be displayed on a web-based dashboard.



# CICS CBE format event for WBM

```

<cbe:CommonBaseEvent xmlns:cbe="http://www.ibm.com/AC/commonbaseevent1_0_1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.1"
  creationTime="2009-02-11T02:42:06+00:00"

  <cbe:sourceComponentId component="IBM CICS TS#4.1.0" componentIdType="ProductName"
    executionEnvironment="IBM z/OS"
    instanceId="GBIBMUVW.IYK1X2YZ" location="MV2C" locationType="Hostname"
    subComponent="CICS EP" componentType="http://www.ibm.com/xmlns/prod/cics/eventprocessing"/>
  <cbe:situation categoryName="OtherSituation">
    <cbe:situationType xsi:type="OtherSituation" reasoningScope="EXTERNAL">
      <CICSApplicationEvent/>
    </cbe:situationType>
  </cbe:situation>

  <cics:event xmlns:cics="http://www.ibm.com/xmlns/prod/cics/events/CBE">
    <cics:context-info>
      <cics:eventname>Order_Placed</cics:eventname>
      <cics:usertag>V0_9</cics:usertag>
      <cics:networkapplid>GBIBMUVW.IYK1X2YZ</cics:networkapplid>
      <cics:timestamp>2009-02-11T02:42:06+00:00</cics:timestamp>
      <cics:bindingname>Order_Processing</cics:bindingname>
      <cics:capturespecname>Order_PlacedCapture</cics:capturespecname>
      <cics:UOWid>1A11C7C2C9C2D4C9E8C14BC9E8C3E6E3C3F9F9BABD6C7A0D35000A</cics:UOWid>
    </cics:context-info>
    <cics:payload-data>
      <data:payload xmlns:data="http://www.ibm.com/prod/cics/V0_9/Order_Placed">
        <data:Customer>Customer1</data:Customer>
        <data:Item_Ordered>0030</data:Item_Ordered>
        <data:Number_Ordered>4</data:Number_Ordered>
      </data:payload>
    </cics:payload-data>
  </cics:event>

</cbe:CommonBaseEvent>

```

CBE Envelope

xs:any slot

# CICS CBE Format Event for WBE – Notes

- This shows the structure of a CICS event inbound to WBM.
- The CBE envelope consists of standard CBE elements and fields which are either required elements or are useful to include in a CICS event.
- The xs:any slot contains the CICS-specific elements, which are some context information about the event and the environment in which it occurred (known as the 'static' schema portion), and information that is specific to the event i.e. the emitted business information as defined in the event specification (known as the 'dynamic' schema portion).

## Example – Business process performance and KPIs

- **Order Processing application**
  - Events emitted during order processing
    - Order Placed
    - Order Shipped
    - Order Cancelled
- **Business information required**
  - How long on average does it take to dispatch orders?
    - Are we meeting our “shipped in 5 working days” SLA?
  - How often are orders cancelled?
  - How often are orders cancelled after they have been shipped?
  - etc.

## Example – Business Process Performance and KPIs – Notes

- This example is based on an order processing application.
- Business monitoring can be used to gain insight into the business processing, and see whether Key Performance Indicators (KPIs) are being met.
- This scenario gives an example of monitoring a business process using events that occur during the processing. By producing events from CICS whenever an order is placed, when it is fulfilled, when it is shipped, and when it is cancelled, it is possible to obtain insight into such things as
  - How long it is taking to dispatch the orders for shipping? Consider a situation where orders are placed via a website, and the website states that the items will be shipped to the customer within 5 working days. In such a situation, it would be very useful to be able to see how easily that KPI is being met (are most orders dispatched within 2 days?) and generate alerts when it is missed.
  - How often orders are cancelled? If the order processing system allows orders to be cancelled, then it would be useful to know how many orders are cancelled, or whether it is common for cancellation requests to come in after the order has been dispatched, incurring additional costs in arranging for return of the items.

# Exporting Dynamic Schema for CBE

The screenshot shows the IBM CICS Explorer interface. On the left is the Project Explorer showing a tree view of projects including CatalogManager, InsuranceApp, MyNewBundle, MyOrderApp, META-INF, OrderingEvents.evbind, OrderingPatterns, and StockQuotes. The main window displays the configuration for the 'OrderingEvents.evbind' resource, specifically the 'Adapter' tab. The 'Adapter' dropdown is set to 'WMQ Queue'. Below it, the 'Queue Name' is 'ACME.CICS.EVQ', 'Persistent' is 'Queue Default', 'Priority (Optional)' is '0', 'Expiry Time (1/10 secs) (Optional)' is '1', and 'Data Format' is 'Common Base Event (XML)'. A button labeled 'Export Event Specifications...' is highlighted with a blue oval. A blue arrow points from this button to a dialog box titled 'Export Event Specifications'. The dialog box has a 'Select' section with three checked items: 'Order\_Placed', 'About\_to\_place', and 'Order\_Dispatched'. At the bottom, the 'To directory:' field is set to 'C:\Documents and Settings\IBM'. The dialog box has 'Export' and 'Cancel' buttons.

# Exporting Dynamic Schema for CBE – Notes

- This screenshot illustrates the process of using the Event Binding Editor to export the dynamic schema which describes the event-specific portion of the Order\_Placed event.
- When the EP Adapter is specified with a format of Common Base Events, then the Export Event Specifications option will present a dialog from which events in the event binding can be selected, and their dynamic schema descriptions exported to the requested location.

# Authoring the Monitor Model in WID

The screenshot displays the IBM WebSphere Integration Developer 6.2 interface. The main window is titled "Monitor Details Model" and shows the configuration for an inbound event named "orderPlacedEvent".

**Monitor Details Model**

**▼ Inbound Event Details**  
 Edit the details of the inbound event, which references an event that is generated by the monitored application.

ID:  [Edit...](#)

Name:

Description:

**▼ Event Type Details**  
 Specify the event type or the XML schemas that together describe the structure of this inbound event. You can specify an extension name, event parts, or both.

Extension name:  [Browse...](#) [Clear](#)

Event parts:

| ID      | Name    | Type        | Path                                    |
|---------|---------|-------------|---|
| static  | static  | cics:event  | cbe:CommonBaseEvent/cics:event          |
| dynamic | dynamic | tns:payload | cbe:CommonBaseEvent/cics:event/cics:... |
|         |         |             |   |
|         |         |             |   |
|         |         |             |   |

[Add](#) [Remove](#)

**▼ Filter Condition**  
 Define a condition based on the event attributes to identify whether to accept an event of this type.

**▼ Correlation Expression**  
 Define an expression to identify the monitoring context instance or instances that receive the event at runtime.

If no instances are found [Create new instance](#)

The left sidebar shows a tree view of the project structure, including "cicsMM4" and its sub-elements: "itemID", "Month", "Year", "itemNumber", "OrderNumber", "orderPlacedTrigger", and "orderPlacedEvent".

The bottom status bar shows the current date and time: 4:02 下午 星期三 3/11/2009.

# Authoring the Monitor Model in WID – Notes

- This screen shot shows part of the Monitor Details Model for this scenario as modeled in the WebSphere Integration Developer (WID) tooling.
- OrderPlacedEvent is the ID for the inbound event.
- The event definition comprises different 'event parts', each of which is a schema part. In the Event Type Details, the static and dynamic parts of the CICS CBE event are the event parts.
- A Filter Condition can be defined for each inbound event, and if the condition is not met then the event is ignored. The filter condition typically uses attributes of the events, and in this example the filter condition is that the event name in the context-info within the static schema is 'Order\_Placed'.
- The Correlation Expression identifies the monitoring context instance (or instances) that receive the event; that is, if true then the event is part of this instance (e.g. it is an event relating to a particular order). In this example, if the item ordered and the month and year have not been seen before, then a new monitoring context instance will be created, to which any other events for the same item, month and year will be delivered.
- Not shown on the screen shot are the Metrics which can be defined for the event – a metric value expression can be derived from the inbound event. Also, Triggers can be defined which cause additional processing when an event arrives, such as changing a metric stopwatch or counter, or terminating the monitoring context instance. Outbound events and alerts can also be defined.



## Business Process Performance and KPIs in Business Space

### Instances

Export ...
Search for: 
Reset

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

### KPIs

| KPI Name                      | Status | Value          | Target        | Actions | Value in Range  |
|-------------------------------|--------|----------------|---------------|---------|---|
| average time on order shipped |        | 23 m, 18.601 s | 1 h, 0 m, 0 s |         | <div style="width: 100%; height: 15px; background: linear-gradient(to right, #008000 20%, #ffff00 20% 40%, #ff0000 40% 100%);"></div> |
| percent of order cancelled    |        | 25.00%         | 30.00%        |         | <div style="width: 100%; height: 15px; background: linear-gradient(to right, #008000 75%, #ff0000 75% 100%);"></div>                  |

### Dimensions

File Edit View Bookmarks Data Chart Tools Help

Drill Down

**Measures**

■ order canceled ■ order shipped

| order status dimension            | average order price | sum order price | order count |
|-----------------------------------|---------------------|-----------------|-------------|
| <b>All order status dimension</b> | 280                 | 1120            | 4           |
| order canceled                    | 400                 | 400             | 1           |
| order shipped                     | 240                 | 720             | 3           |

# Business Process Performance and KPIs in Business Space – Notes

- This shows dashboard views in Business Space for the scenario to study business process performance and KPIs. 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.
- Business Space is the unifying front end for IBM's BPM products. It is not a separate product, but a feature of many, and provides a user friendly, browser-based interface. It is included with products such as WebSphere Business Monitor, WebSphere Business Services Fabric, and WebSphere Process Server (amongst others). Business Space allows business users to build their own dashboards using widgets.

## Other monitoring opportunities

- **Follow the end-to-end processing of orders**
  - Events at all stages in the process
    - Catalog request, inquiry on item, request for price quote, place order, confirm order, order dispatched, invoice issued, payment received, stock replenished, etc.
  - Might involve a process choreographed by **WebSphere Process Server**, but with some processing carried out within CICS
    - CICS event processing support opens up the 'black box' that hitherto has been the processing in CICS

## Other Monitoring Opportunities – Notes

- There are many other monitoring possibilities, even within the realm of an order processing style of application.
- As this slide indicates, there are many other steps involved in processing of orders which could also be produced as events.
- It is also possible that the overall process does not take place within CICS. For example, there might be an end-to-end process which is managed within WebSphere Process Server, but portions of the processing (perhaps the order handling and the billing) are carried out within CICS. WebSphere Process Server already has the capability to produce events at various stages in its processing, and together with events emitted by CICS TS V4.1, it is now easy to follow the processing once it enters CICS.

# Business Flexibility with CICS Events

# Business Flexibility with CICS Events – Notes

- This section looks briefly at how the CICS support for events can be used to respond flexibly to business challenges and changes.

## Using CICS events to extend applications

- **CICS event can start new CICS transaction**
  - Transaction Start EP Adapter
  - Run new program, passing event information
- **CICS event can be written to WMQ queue**
  - WMQ Queue EP Adapter
  - Program can read event from queue and carry out new processing
  - Events can be sent to WebSphere Business Events, where they might be combined with other events, and result in actions which drive new processing

# Using CICS events to extend applications – Notes

- CICS events can be used to drive a CICS transaction, via the Transaction Start EP adapter, which can carry out processing additional to that of the emitting program.
- CICS events can also be written to a WebSphere MQ Queue in a character-based non-XML format, which can then be read from the queue in another application, either running within CICS or elsewhere, and used to drive additional processing.
- As is the nature of event-driven processing, the resulting processing will occur asynchronously to the emitting application, and in a decoupled way.



## Example of application extension

- **Copy of bank account details have fallen into wrong hands**
- **Bank needs to immediately detect activities on those accounts**
- **Events captured at all places where**
  - Money withdrawn from accounts
  - Contact details updated
  - New check (cheque) book requested
- **Drive CICS transaction which does lookup against account details in Shared Data Table**



## Example of application extension – Notes

- This is just one example from many which could be given:
- A bank discovers that the account details of large numbers of its customers have potentially fallen into the wrong hands. They need to put something in place immediately which will allow them to monitor those accounts for possible fraudulent activity. They use CICS event capture points at all places where money is withdrawn from accounts, contact details are updated, etc. The account number is a key piece of data collected when the event occurs, along with the contact details before the update in the case of a contact-detail-update event. Each event drives a CICS transaction which does a look up of the account number against (say) a shared data table. A match immediately triggers an email or phone call to the customer (using the original contact details) to verify whether they carried out the action themselves

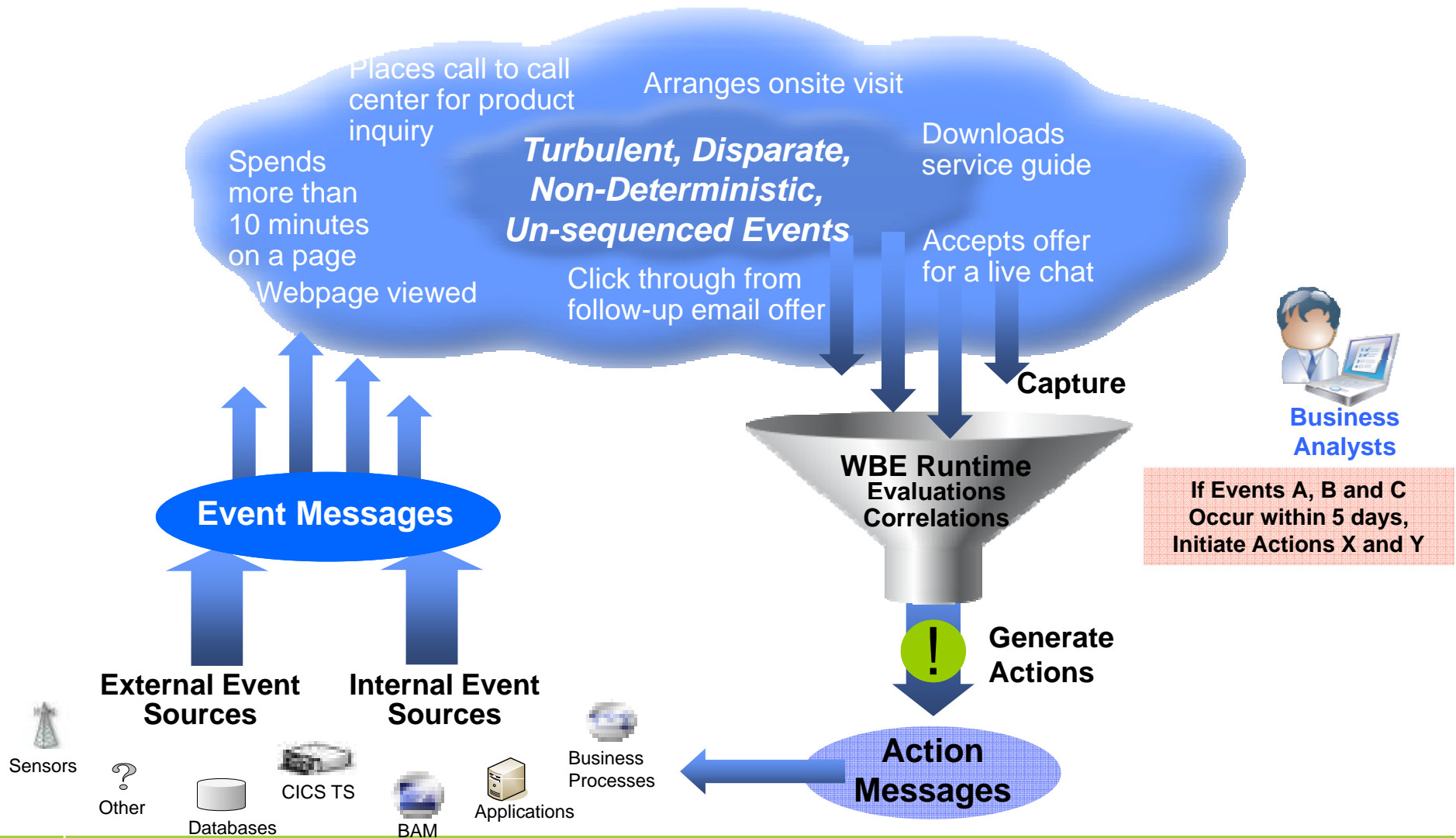
# CICS Events and WebSphere Business Events

# CICS Events and WebSphere Business Events – Notes

- This section covers integration of CICS events with WebSphere Business Events.

# WebSphere Business Events

*Correlation of business events from any source, over any time frame*



# WebSphere Business Events – Notes

- In large organizations, tens of millions of events occur every day, but not all events are of equal importance. WebSphere Business Events provides the ability to determine when a pattern of related, or seemingly unrelated, events from one or more sources has occurred and coordinate the execution of the responses to that pattern of events.
- Disparate event messages flow from virtually any source (systems, devices or human interactions), from both inside and outside the enterprise, into the IT communication network (represented here as the “event cloud”). WebSphere Business Events (WBE) determines actionable situations by detecting, evaluating and correlating the events based on user defined logic. When an actionable situation is discovered, the WebSphere Business Events runtime coordinates the appropriate follow-on processing by generating and communicating one or more actions back through the IT network.
- In the example illustrated on the slide, the business user has defined logic that instructs the runtime to detect and take specific action when a complex pattern of events indicating a customer’s level of interest in a particular product occurs.
- WebSphere Business Events detects event patterns as they happen...
  - Across different event types and disparate event sources...
  - Where events may not be ordered...
  - Where the absence of an event is significant...
  - Where the actionable event needs to be derived from physical events...
  - Where processing is composed of multiple, asynchronous steps...
  - Involving systems, humans or both...

## CB11: CICS Events for WBE SupportPac

- CICS SupportPac CB11 provides a taster of CICS acting as a source of events to WebSphere Business Events
- The SupportPac runs on CICS TS V3.1 and V3.2, and was made available December 2008 (refreshed May 2009 to run on CICS TS V4.1)
- Requires a small change to existing CICS application code
  - Invoke the program supplied by the SupportPac, passing event details in a channel
- The SupportPac formats the event information into XML conforming to WBE format and emits the event for consumption by WebSphere Business Events (or other consumer program) via the WebSphere MQ transport
- WebSphere Business Events can include the event in interaction blocks to carry out pattern matching etc.

# CB11: CICS Events SupportPac for WBE – Notes

- This gives some details about a SupportPac, CB11, which provided a taster of using events from CICS with WebSphere Business Events (but not in a non-invasive way).
- The event will be emitted by CICS over the WebSphere MQ transport. More information about configuring the system so that the event can be consumed by WebSphere Business Events are given in the documentation which accompanies the SupportPac.



## Insurance Scenario Involving a Business Opportunity

- **Customer has obtained insurance quotes for three or more of car, house, belongings and travel insurance**



- CICS application code which processes insurance quotes could emit an event when a quote is processed, including a customer name or identifier and the type of insurance policy

### **Customer has purchased only one of these insurance policies**

- CICS can emit event when purchase of insurance policy is processed, with customer identifier and policy type

- **WebSphere Business Events can detect this pattern**

- When quotes for these types of policy received but not all have matching purchase (correlated by customer), then take action



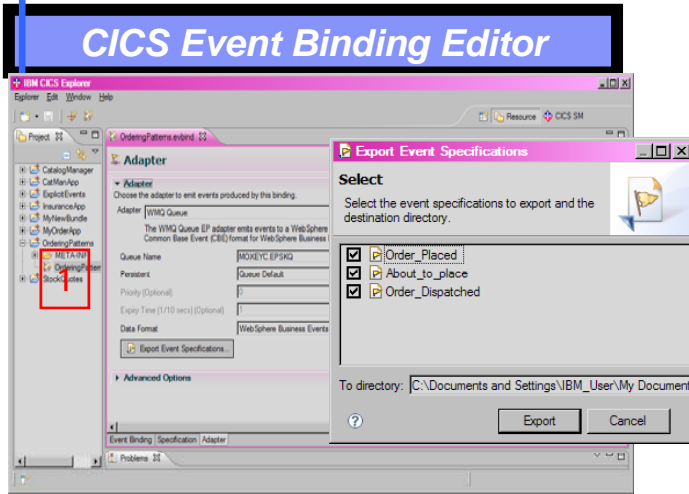
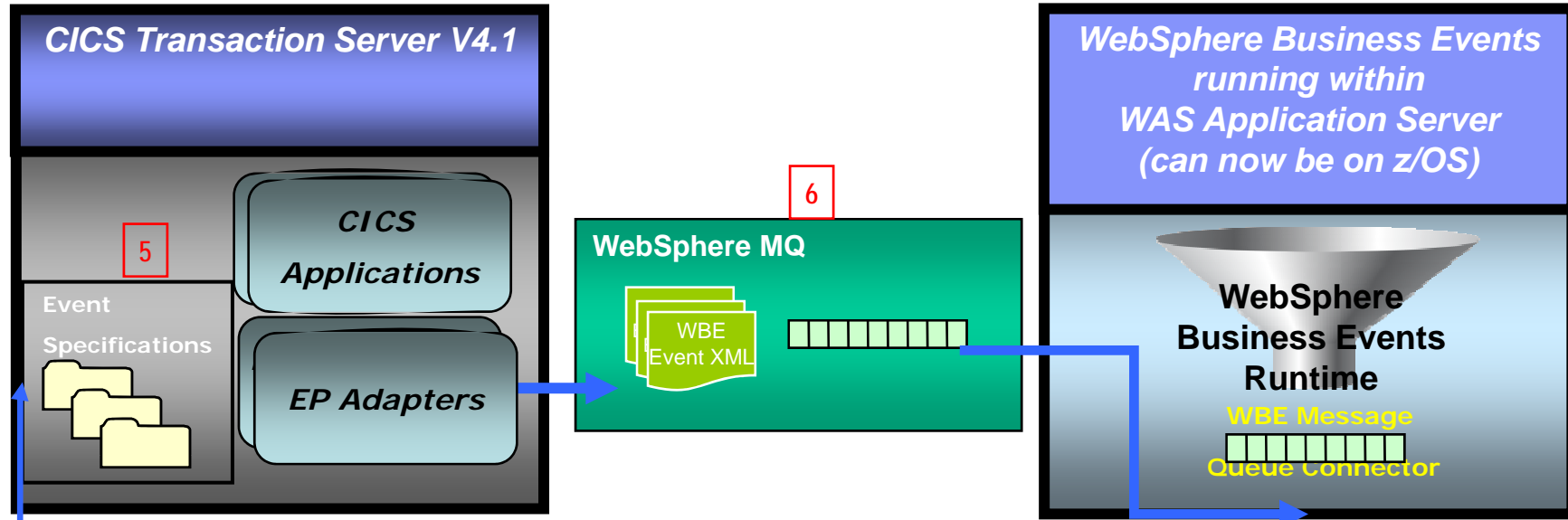
- **Send offers to the customer for the other insurance for which quotes were requested**

- Better targetting than sending offers for all types of insurance available

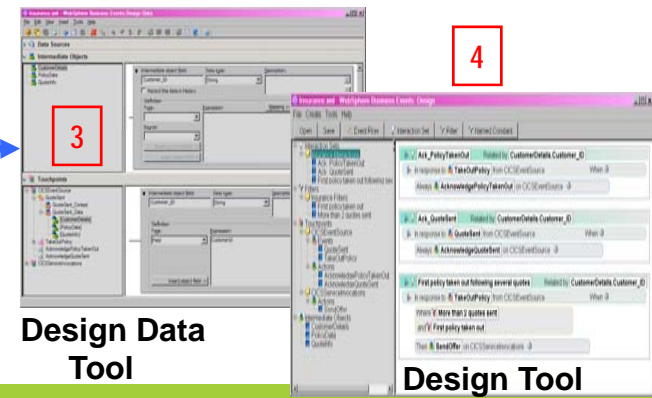
## Insurance Scenario – Notes

- This scenario shows how an insurance company could detect events occurring in CICS relating to obtaining quotes for policies, and then correlate these in WebSphere Business Events with events indicating that a policy has been taken out. This represents an opportunity to sell the other policies to the customer, which could be responded to by sending a targetted offer (for example).
- The slide shows how this could be achieved using a combination of emitting the events from CICS and defining appropriate conditions and interactions, with associated action, in WebSphere Business Events.

# CICS and WebSphere Business Events Integration



WebSphere MQ via e.g. WAS platform messaging



# CICS and WebSphere Business Events Integration

## – Notes

- This slide shows how CICS events are integrated with WebSphere Business Events, which provides the ability to detect patterns between multiple events over time.
- The figure shows the CICS Event Binding Editor [1] used to create event specifications which can then be installed in CICS TS V4.1 [5]. The WebSphere Business Events Design Data tooling can create event definitions [3] from event schemas exported from the event binding editor[2], and these event definitions can be used in the Design tooling to indicate the patterns or interactions to be detected [4]. The event definitions specify a Message Queue connection in order to receive the events over MQ.[6]
- When events are captured from CICS applications, they are processed by the appropriate Event Processing Adapter. For WebSphere Business Events, the EP adapter puts the event in the XML format recognised by WebSphere Business Events onto a WMQ queue on z/OS, which is configured to be received by the WebSphere Business Events runtime. The connection could, for example, use the WAS platform messaging provided by the WAS in which WebSphere Business Events runs. A version of WebSphere Business Events has recently been made available that runs on z/Os.

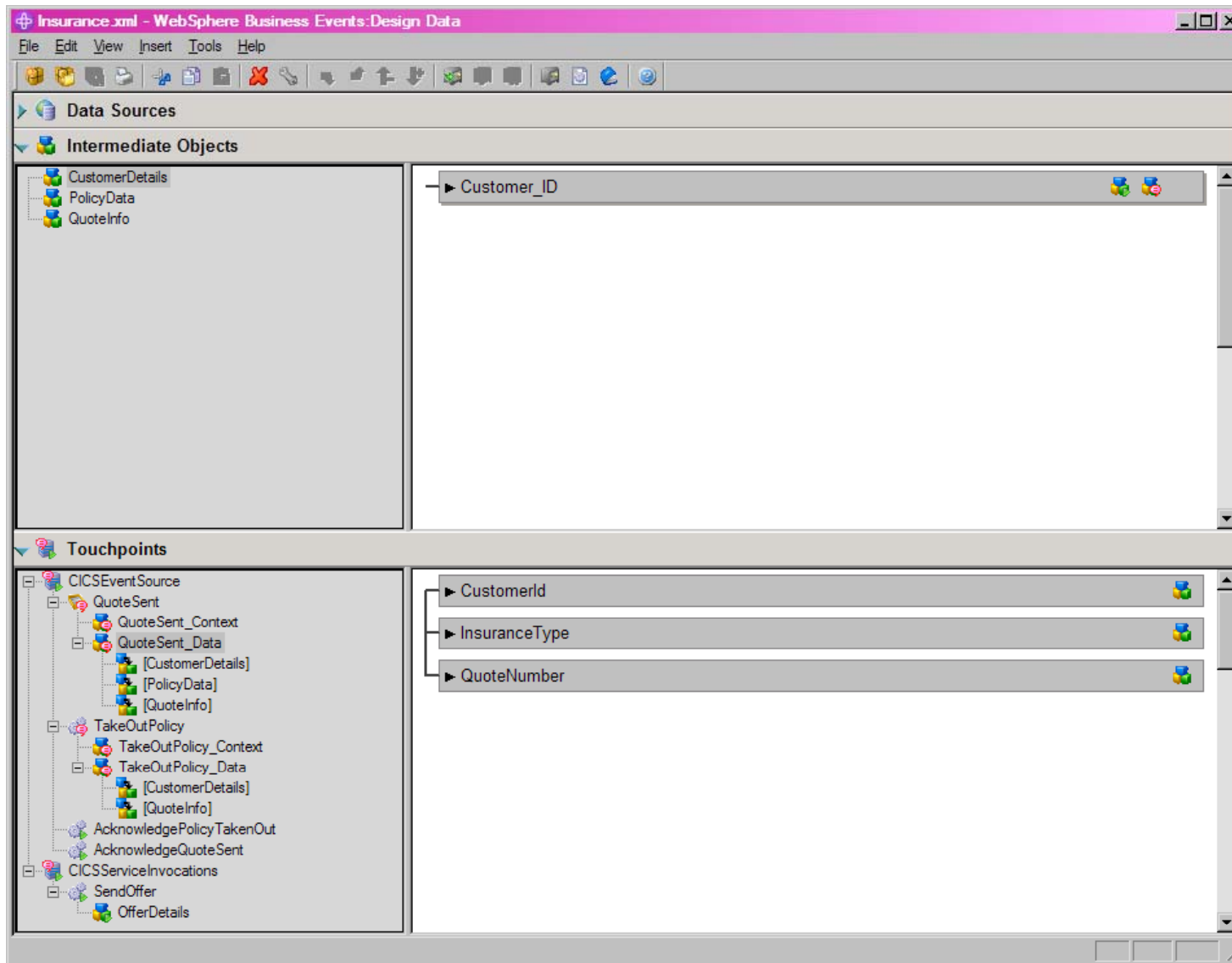
## EP Adapter and Exporting Event Specifications for WebSphere Business Events

The screenshot displays the IBM CICS Explorer interface. On the left, the Project Explorer shows a tree view of the project structure, including folders like CatalogManager, CatMan.App, ExplicitEvents, Insurance.App, MyNewBundle, MyOrderApp, OrderingPatterns, META-INF, and StockQuotes. The main window shows the configuration for the 'OrderingPatterns.evbind' binding, specifically the 'Adapter' tab. The 'Adapter' dropdown is set to 'WMQ Queue'. Below this, the 'Queue Name' is 'MOXEYC.EPSKQ', 'Persistent' is 'Queue Default', 'Priority (Optional)' is '0', and 'Expiry Time (1/10 secs) (Optional)' is '1'. The 'Data Format' is set to 'WebSphere Business Events (XML)'. A blue arrow points from the 'Export Event Specifications...' button in the main window to the 'Export Event Specifications' dialog box. The dialog box has a 'Select' section with three checked items: 'Order\_Placed', 'About\_to\_place', and 'Order\_Dispatched'. At the bottom, the 'To directory:' field is set to 'C:\Documents and Settings\IBM\_User\My Documents\Sus'. The 'Export' and 'Cancel' buttons are visible at the bottom right of the dialog.

## EP Adapter and Exporting Event Specifications for WebSphere Business Events – Notes

- This is the Adapter portion of the event binding for insurance policy events, and shows that events in this binding are emitted via the WebSphere MQ queue EP adapter, and in the XML format consumed by WebSphere Business Events. The queue on z/OS to which the event will be emitted is specified.
- When the event format is WebSphere Business Events, the Export Event Specifications option produces schema files for the events which can be imported into the WebSphere Business Events Design Data tool (used by IT users to create the events and actions and make them available to the business users in the Design tool).

## Design Data : Events and Intermediate Objects

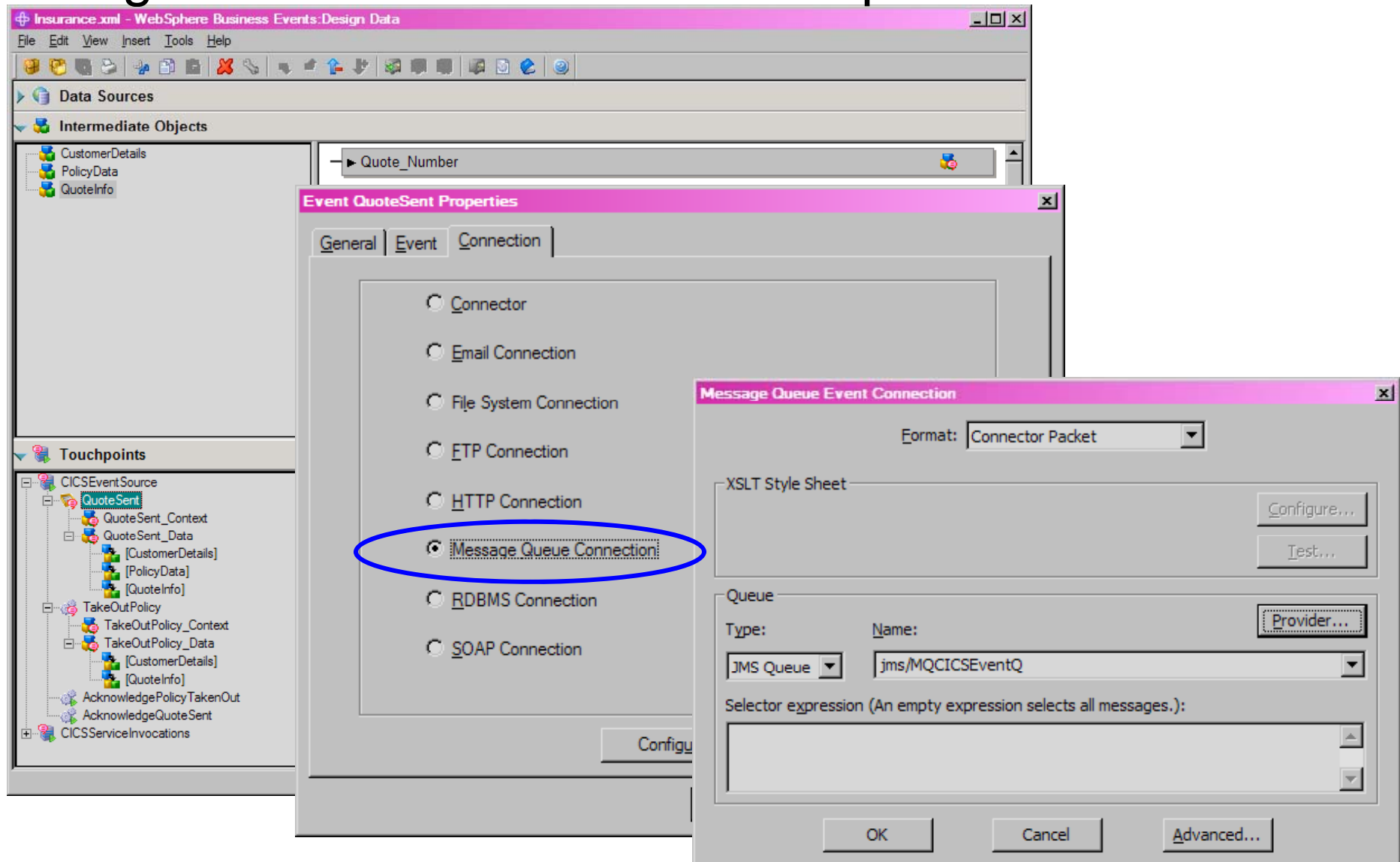


## Design Data : Event and Intermediate Object – Notes

- This screenshot is taken from the Design Data tool in WebSphere Business Events, which is the tool used by IT to define building blocks for business to use.
- The Touchpoints portion of the screen is where events received from event sources, and actions to be sent by WebSphere Business Events, are defined. An event named 'QuoteSent' has been defined under the 'CICSEventSource' touchpoint. This event contains two event objects, QuoteSent\_Context and QuoteSent\_Data, which map to the two connector-object elements included in a CICS WBE format event (the Event Binding Editor can export a schema representing the event which can then be imported into the Design Data tooling as an easy way to define the event and its data).
- The Intermediate Objects portion of the screen is used to define intermediate objects, which allow for a level of indirection between incoming events and specifications made by the business about how to respond to events. In this screenshot, the CustomerDetails intermediate object is selected, which has the customer identifier associated with it. The items of data in an intermediate object map onto information available from one or more events.



# Design Data : Event Connection Specification



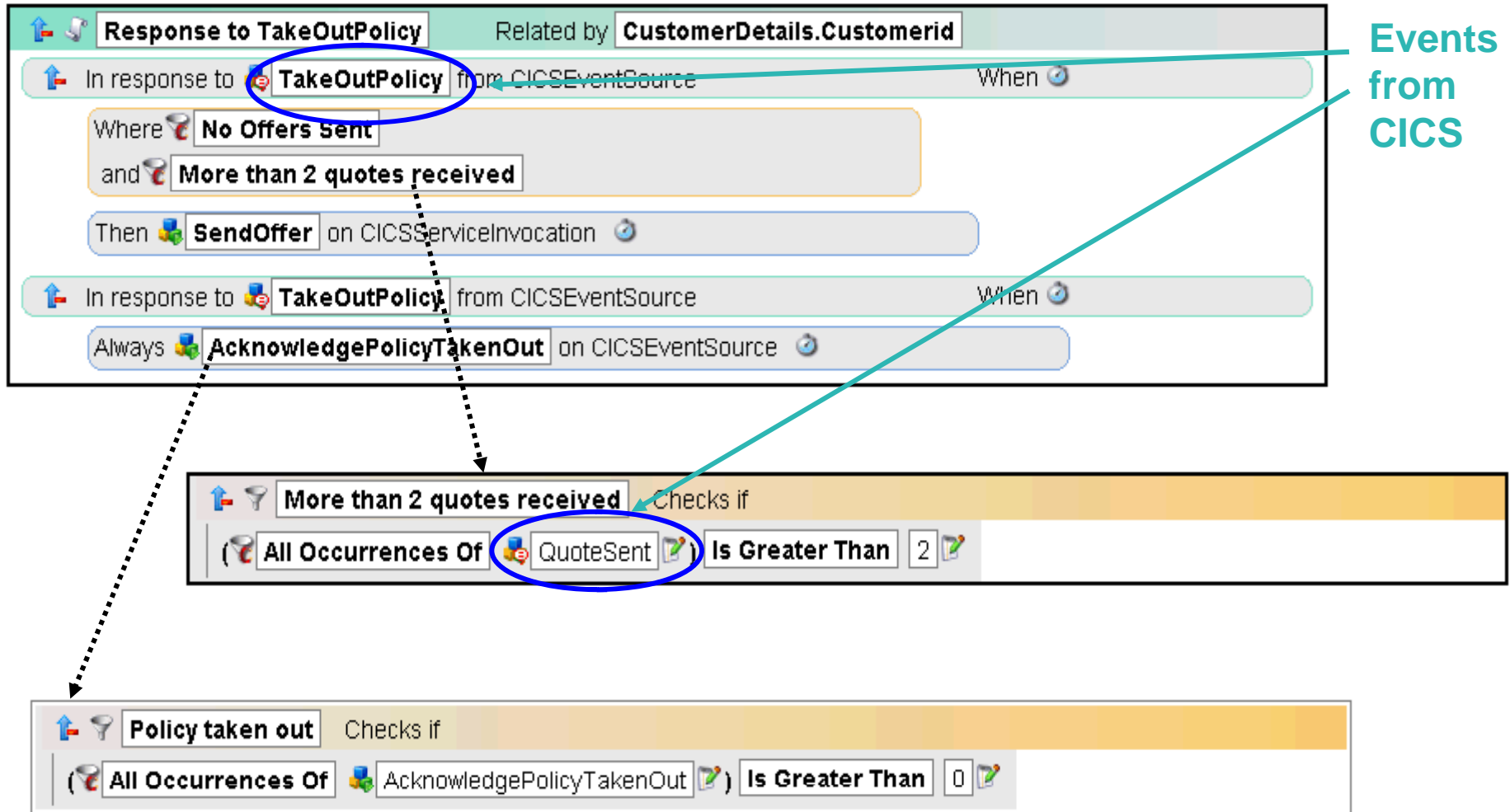
# Design Data : Event Connection Specification

## – Notes

- This screenshot shows how the connection can be set up in WebSphere Business Events to receive events from CICS.
- Under Touchpoints in the Design Data tool, the QuoteSent event has been selected, and the Event Properties dialogue opened. This is the first pop-up window shown in the screenshot, and the Connection tab has been selected.
- Under Connection, the 'Message Queue Connection' radio button option has been selected. The lower pop up window is the 'Message Queue Event Connection' configuration details, with the Format as Connector Packet, the queue type as JMS Queue, and the queue name selected.

# Insurance Quote Example – 1

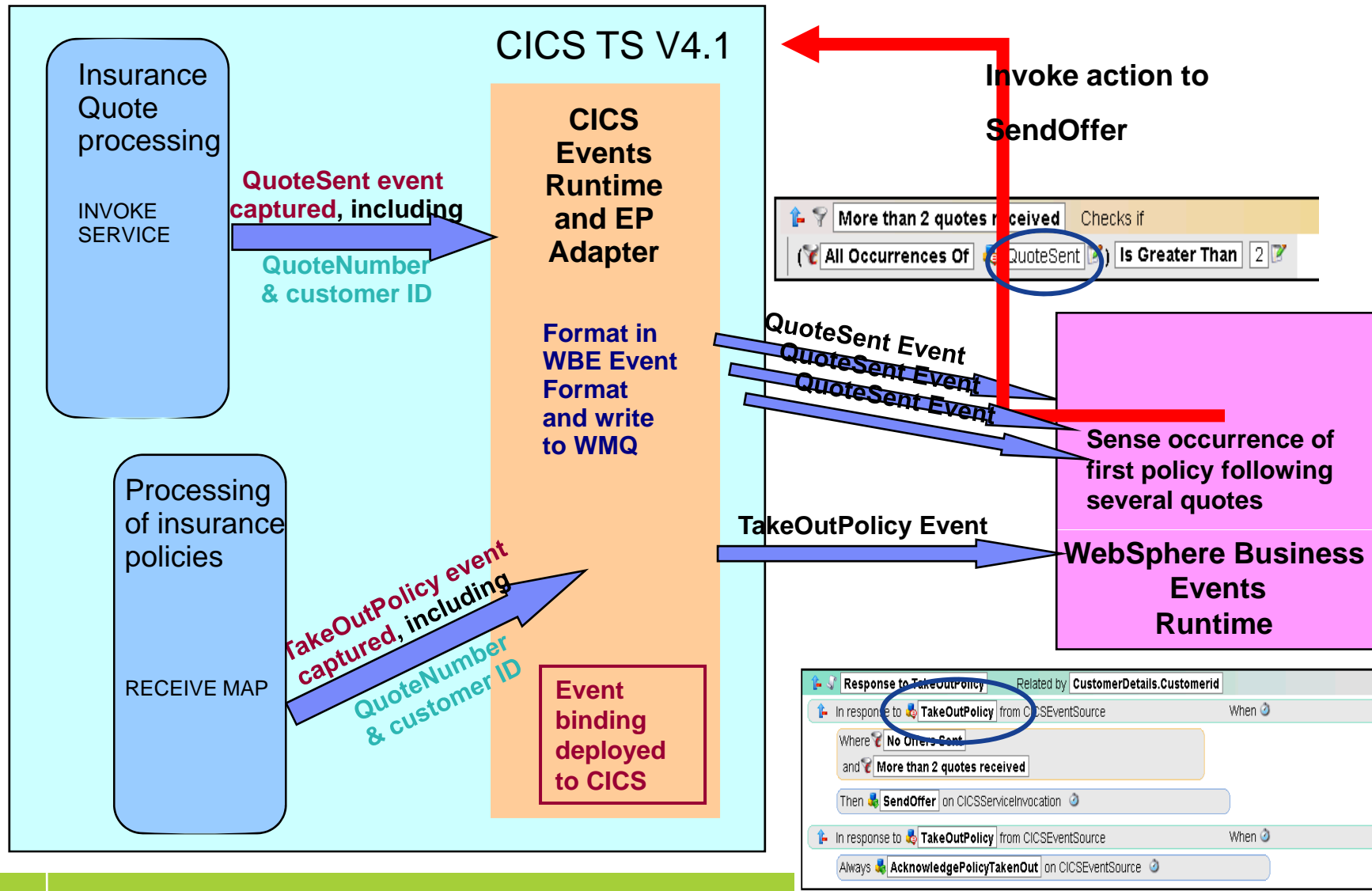
WebSphere Business Events Interaction Set



## Insurance Quote Example – 1 – Notes

- Events from CICS can be used in interaction sets and conditions (filters) defined in WebSphere Business Events
- This and the following slide illustrate this with a scenario.
- The scenario shows an interaction defined using WebSphere Business Events tooling, for 'Response to Take Out Policy'.
  - When a TakeOutPolicy event occurs, check whether an offer has been sent to this customer yet.
  - If not, check whether there have already been at least 2 instances of a QuoteSent event for this customer.
  - If there have been more than 2 quotes and no offer sent, then send an offer to the customer (to encourage them to take out more policies).
- The WebSphere Business Events Design tool has been used to specify the interactions and conditions shown on the slide.
- The events that are being checked for are events that occur in CICS. The WebSphere Business Events Design Data tool is used to define the incoming events and their data, to be used in the Design tool to define the interaction set i.e. the event-conditions-actions grouping.
- This could be made more sophisticated, adding an interaction set to cover the situation where a policy is taken out before more than 2 quotes have been sent, but we still want to send an offer to the customer, and also potentially sending out offers relevant to the quotes.

# Insurance Quote Example - 2



## Insurance Quote Example – 2 – Notes

- In this scenario, CICS carries out processing of insurance quotes, and handles taking out of insurance policies. An event binding is installed and enabled in CICS which contains capture specifications for QuoteSent and TakeOutPolicy events, and indicates that these events should be processed by the WMQ queue EP adapter and formatted in WBE format.
- The insurance quote processing writes the new QuoteNumber to a file when the quote has been calculated, and the capture specification in the event binding uses this to detect a quote being sent, so that a QuoteSent event is captured when this happens, along with information about the customer, policy, and quote.
- The WMQ Queue EP adapter formats the captured 'QuoteSent' event in WBE format, and emits the event to the WMQ queue specified in the event binding. This queue has been configured to emit the event to the WebSphere Business Events runtime.
- The program which processes insurance policies carries out the processing for a customer taking out a policy when it receives a MAP with a particular name, and this is specified in a capture specification which also specifies how to capture information about the customer making taking out the policy.
- The WMQ EP adapter formats the 'TakeOutPolicy' event in WBE format and emits it to WebSphere Business Events.
- The occurrence of the TakeOutPolicy event causes WebSphere Business Events to check for two or more previous QuoteSent events for the same customer, and if found this triggers an occurrence of the interaction.
- The action associated with this is to send an offer to the customer concerning other insurance policies. A request is made to CICS (such as a Web Service invocation or via WebSphere MQ) to carry out this action.

# Summary, References and Q&A

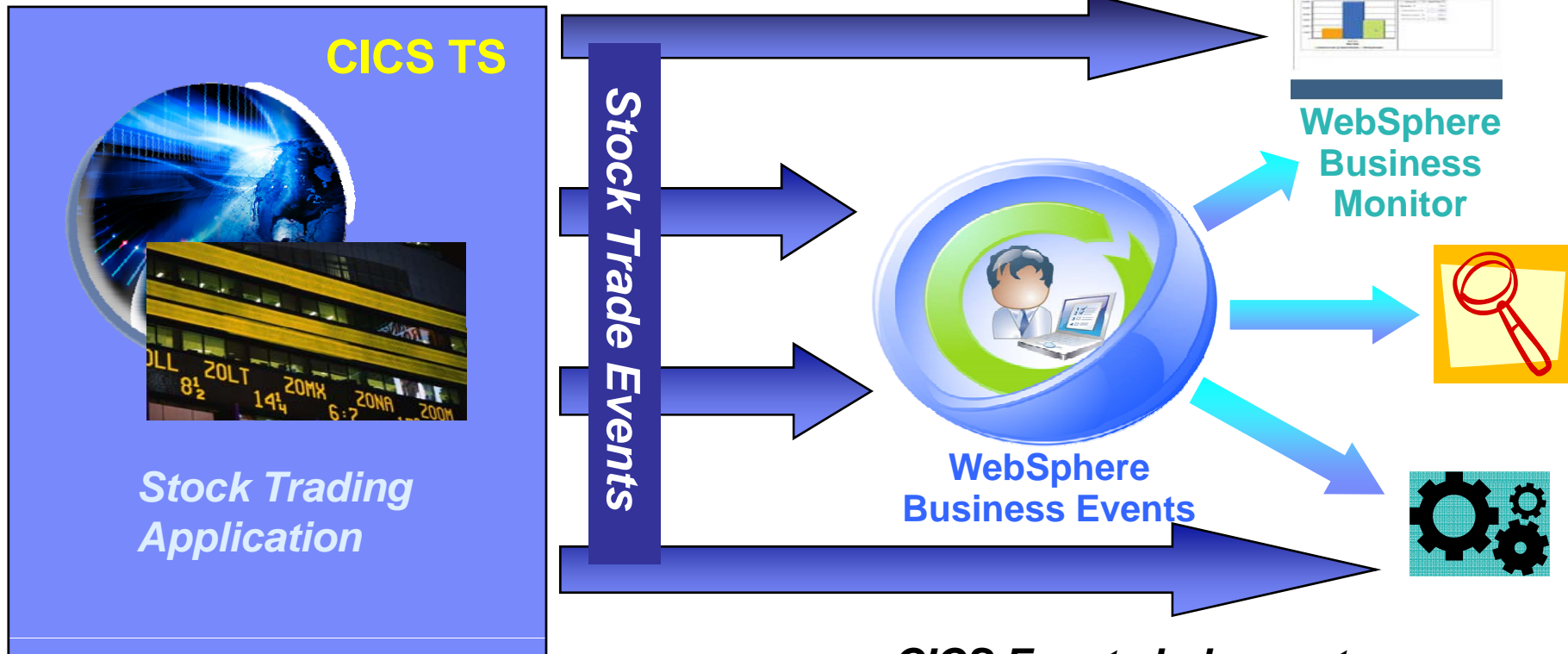
Including a summarising scenario

## 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 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](#)
- WebSphere Business Monitor
  - [WBM Introduction](#)
  - [WBM V6.2 InfoCenter](#)
- WebSphere Business Events
  - [WebSphere Business Events Introduction](#)
  - [WebSphere Business Events V6.2 InfoCenter](#)

## References – Notes

- Some references for CICS Events Support are given. These notes give the URLs behind the hyperlinks.
- CICS TS V4.1 Announcement Letter:
  - [http://www.ibm.com/common/ssi/ShowDoc.jsp?docURL=/common/ssi/rep\\_ca/5/897/ENUS209-135/index.html](http://www.ibm.com/common/ssi/ShowDoc.jsp?docURL=/common/ssi/rep_ca/5/897/ENUS209-135/index.html)
- CICS TS V4.1 Information Center
  - <http://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp>
- CB11: CICS Events for WBE
  - <http://www.ibm.com/support/docview.wss?rs=1083&uid=swg24021039>
- 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
  - <ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/wsw14043usen/WSW14043USEN.PDF>
  - <ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/zsw03120usen/ZSW03120USEN.PDF>
- WBM Introduction: <http://www-01.ibm.com/software/integration/wbimonitor/>
  - WBM V6.2 InfoCenter
    - <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.btools.help.monitor.doc/home.html>
- WebSphere Business Events Introduction: <http://www-01.ibm.com/software/integration/wbe/>
  - WebSphere Business Events V6.2 InfoCenter
    - <http://publib.boulder.ibm.com/infocenter/wbevents/v6r2m0/index.jsp>

## 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

# CICS Event Processing 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, 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.

Thank You !  
Any further questions?



# Thank you – Notes

- Thank you for your time.