Advanced
Technology
Group

IBM

CICS Event Processing

Steve Fowlkes – fowlkes@us.ibm.com

Leigh Compton – lcompton@us.ibm.com

Introduction to CICS and Event Processing – Notes

- This material provides an introduction to new CICS capability in CICS TS V4.1, TS 4.2, and TS 5.1 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.
- This presentation utilizes charts developed by Catherine Moxey, IBM Hursley.
- Catherine Moxey thanked the following 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 and James Taylor from the WebSphere Business Events team.
- This presentation has been updated to include enhancements from CICS TS V5.1.

This material has been prepared by IBM or the 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.

Topics

- Introduction to Event Processing
- CICS and Event Processing
  - Why and How
  - A few example scenarios
  - Example workflow & roles
  - Event specifications
    - Event Capture
    - EP Adapters
- CICS Policies

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 Operational Decision Manager (ODM) 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.

Event Processing – Why now?

- Event processing is not new
  - Systems management and monitoring
  - Pub/Sub messaging systems

- The *business* value of event processing is decreased latency in
  - Obtaining insights
  - Making decisions based on those insights
  - Executing the decisions

- Multiple business factors have accelerated event processing requirements:
  - Compliance with regulations on-line
  - Demand for cost reduction leading to more automation
  - Technology developments such as RFID
  - Desire for greater awareness of business behavior through Business Activity Monitoring, Business Performance Management

Why Now? – Notes

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

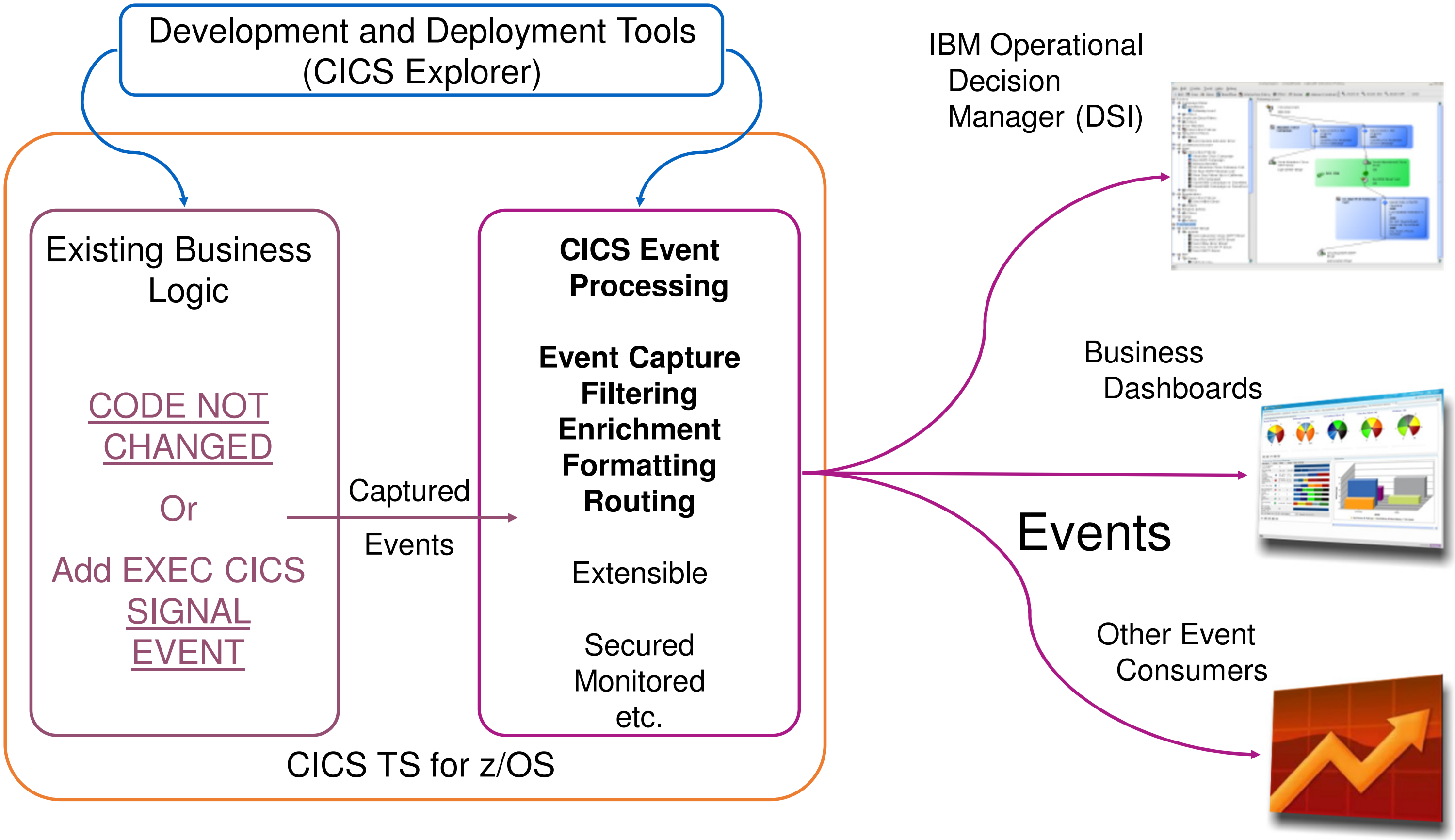
CICS and Business Events

- Event processing addresses the need for agility
  - Modern businesses must react quickly to circumstances
  - Decision makers need reliable, timely information
- CICS systems run an enormous amount of existing business logic
- Using an Event-based approach, there is potential to gain insight into the processing in CICS and to introduce additional extensions to applications
  - In a dynamic, de-coupled fashion
  - Without the need to change the applications
- CICS TS allows you to emit business events from existing applications
  - **CICS TS V5.1 added static data and multiple EP adapters**
  - Supporting shifting corporate policies
  - Without having to modify the applications
  - And driving your choice of destination
    - IBM Business Monitor, IBM Operational Decision Manager, CICS application, application through IBM MQ, …

CICS and Business Events – Notes

- Events are valuable to Enterprise Systems, providing the ability to respond in real-time, or near real-time.
- Given the considerable amount of business processing which is carried out in CICS systems across the world (over 30 billion transactions a day), CICS is a very significant source of business events. This can provide enhanced business flexibility and the ability to meet governance and compliance regulations.
- Event emission is asynchronous to the emitting application, and the consumption of the event is decoupled from its originator.
- CICS TS will emit simple, single events. These may be consumed by a “complex event processing” engine where they can be combined with events from other sources in addition to CICS. They can be sent to a Business Monitor to provide insight into processing within CICS.
- CICS TS V4.1 focused on business events; CICS TS V4.2 adds 6 new system events
- CICS TS V5.1 added the EXEC CICS WRITE OPERATOR command, you can capture and emit events when your system issues a DFHxxnnnn or EYUxxnnnn message. Additionally, you can add static data to the event and the EP search has been expanded to include system messages. Plus, you can send events to multiple adapters.

CICS and Event Processing Overview



Development and Deployment Tools (CICS Explorer)

Existing Business Logic

CODE NOT CHANGED

Or

Add EXEC CICS SIGNAL EVENT

Captured Events

CICS Event Processing

Event Capture
Filtering
Enrichment
Formatting
Routing

Extensible

Secured
Monitored
etc.

CICS TS for z/OS

Events

IBM Operational Decision Manager (DSI)

Business Dashboards

Other Event Consumers

CICS Event Processing Overview – Notes

- This gives a high-level summary of CICS as a source of business events.
- CICS event processing support 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 IBM Operation Decision Manager, WebSphere Business Monitor, and other consumers.
- CICS Event Processing support is extensible, with options for customization.

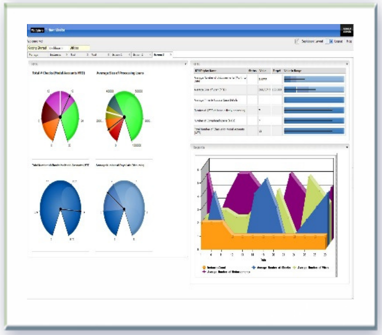
EP Scenario 1 – Observe business processing

- 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 a Business Dashboard
    - 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 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

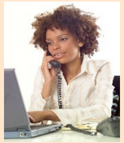
- Collect events relating to credit and other bank card usage
- Check for unusual patterns of behaviour using ODM / DSI
  - New card ordered within a week of an address change request
  - Several online purchases where none had been made before
  - 2 or more cash withdrawals in quick succession when 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 ODM/DSI
  - 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 IBM Operational Decision Manager) is able to collate events from multiple sources and carry out pattern matching to derive additional insight.

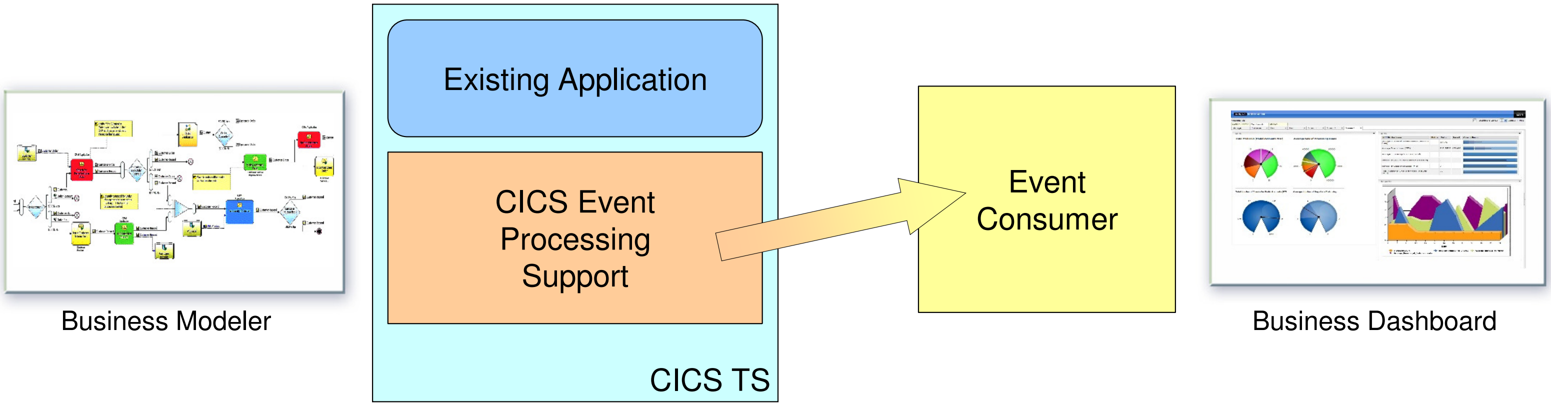
The Role-Based CICS Event Workflow

LoB Manager



Application Analyst

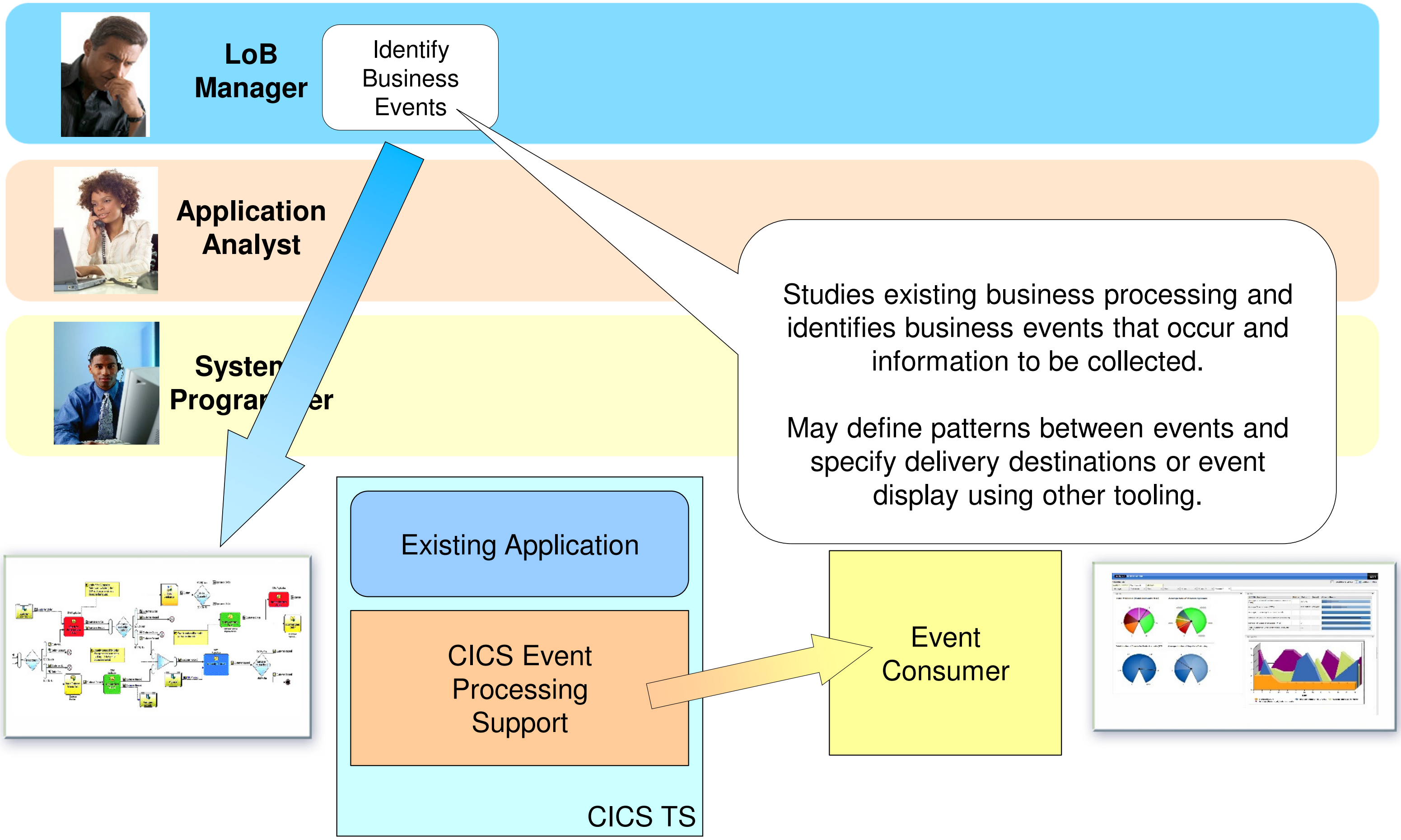
Systems Programmer



The Role-based CICS event workflow – Notes

- This workflow looks at the roles involved in defining the events of interest to the business and enabling them in CICS, through to obtaining information as a result of the events.
- It is important that business events will be definable by the business owners, Line of Business managers and Business Analysts.
- They will of course need some help from the application analysts and programmers to provide the technical description of the event to CICS.
- The following slides illustrate a typical CICS business event definition workflow

LoB defines the business events



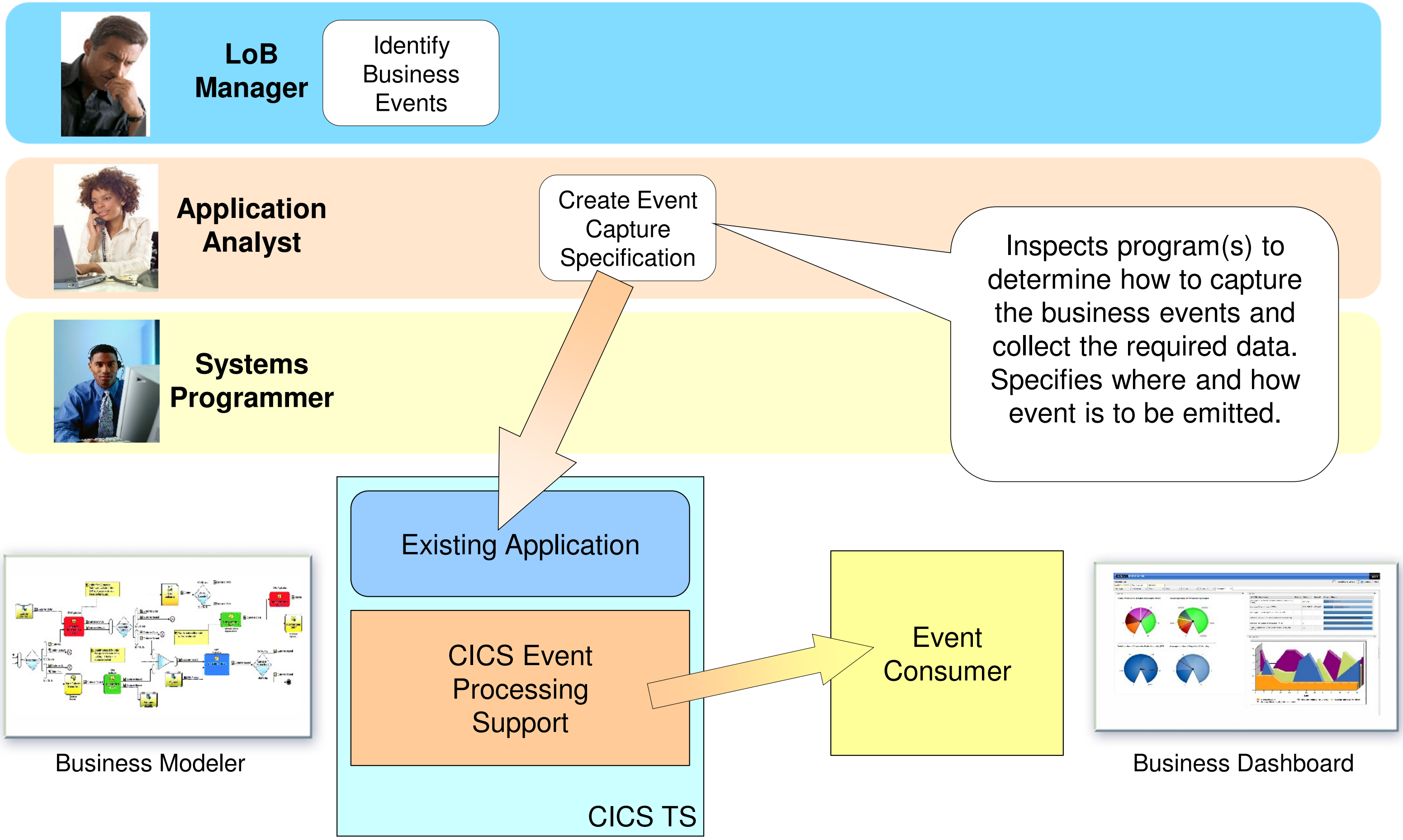
Studies existing business processing and identifies business events that occur and information to be collected.

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

LoB Defines the Business Events – Notes

- The LoB manager (or equivalent role) will define the CICS business application events in terms he is familiar with, together with the data that is to be passed to the event consumer with the event.
- For example
  - Every time an order in excess of $10,000 is received, capture the following fields:
  - Order Number
  - Order Date
  - Customer Ref
  - Customer Name
  - Customer Address

Application Analyst identifies events in application

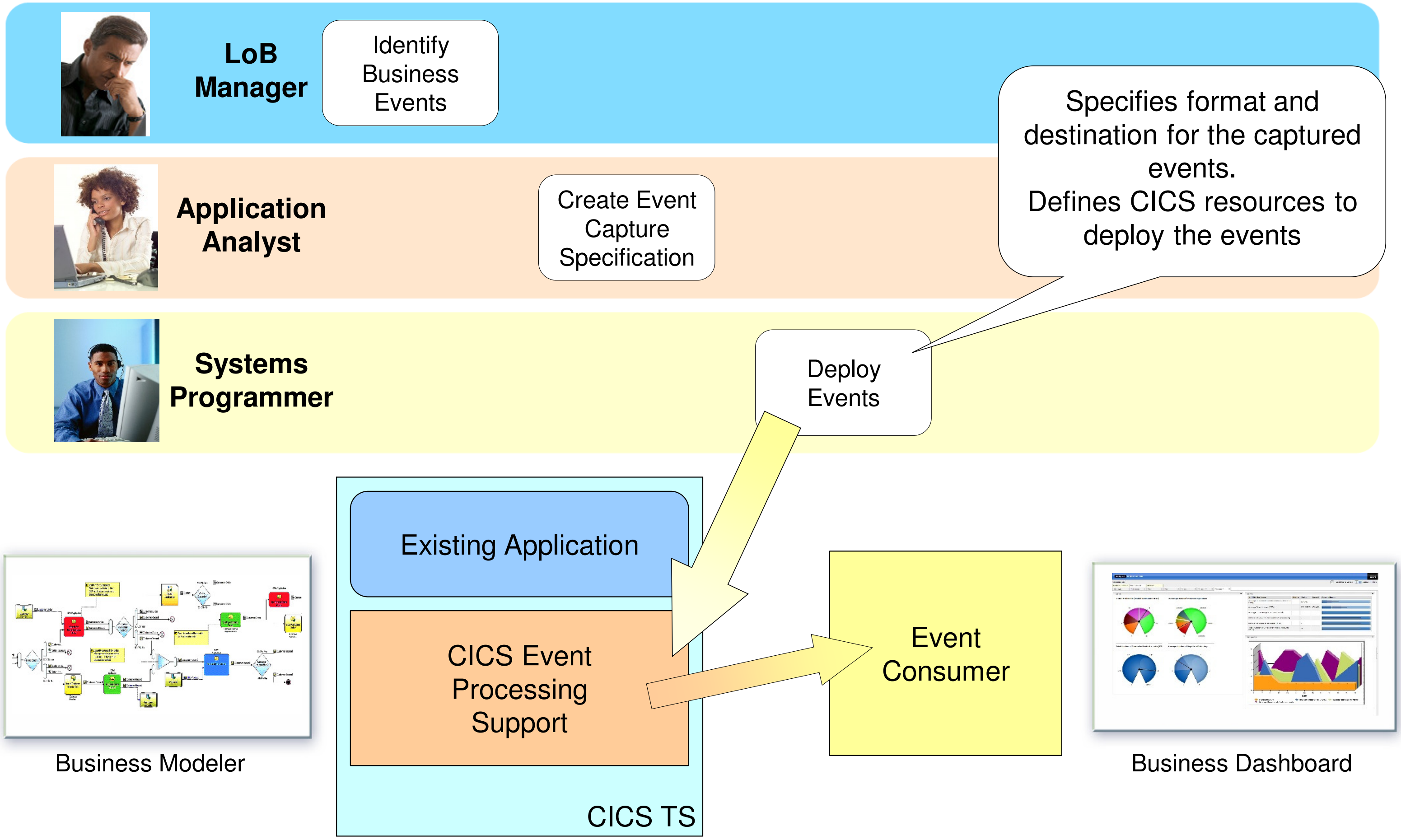


Inspects program(s) to determine how to capture the business events and collect the required data. Specifies where and how event is to be emitted.

Application Analyst identifies events – Notes

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

Systems Programmer configures the infrastructure

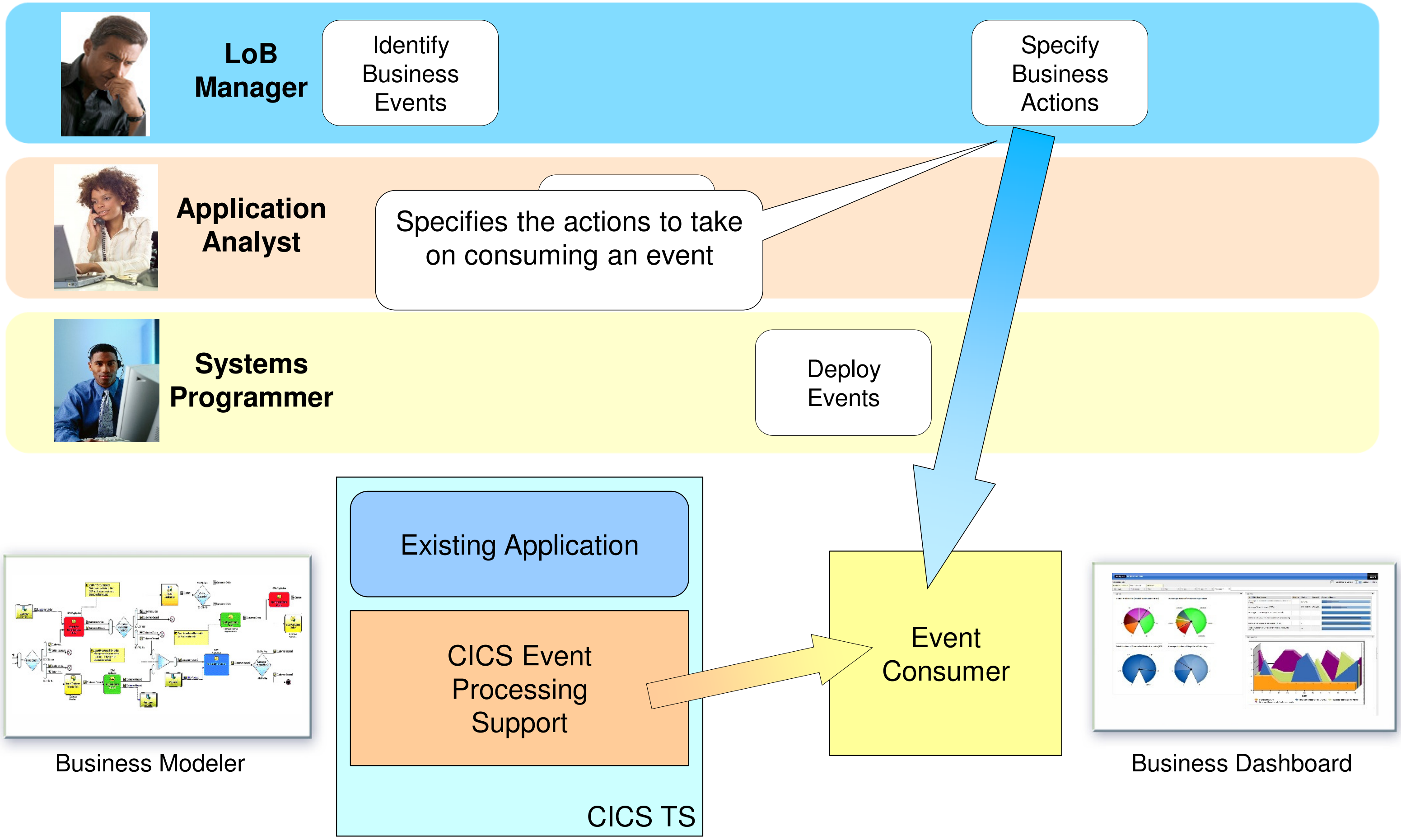


Specifies format and destination for the captured events.
Defines CICS resources to deploy the events

Systems Programmer configures the Infrastructure – Notes

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

LoB defines the business actions



LoB Manager

Identify Business Events

Specify Business Actions

Application Analyst

Specifies the actions to take on consuming an event

Systems Programmer

Deploy Events

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer

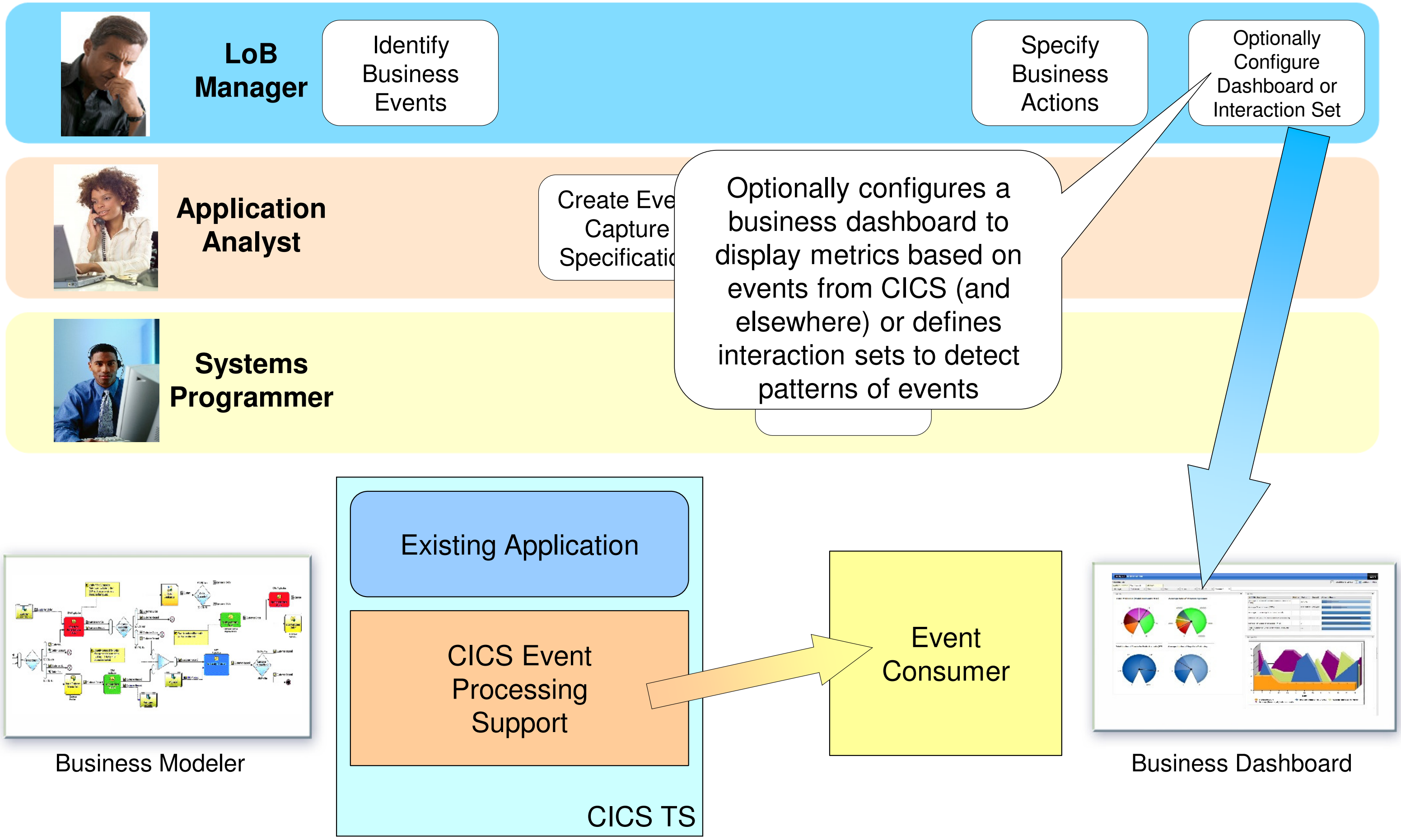
Business Modeler

Business Dashboard

LoB defines the business actions – Notes

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

LOB optionally configures a dashboard



LoB Manager

Identify Business Events

Specify Business Actions

Optionally Configure Dashboard or Interaction Set

Application Analyst

Create Eve Capture Specificati

Optionally configures a business dashboard to display metrics based on events from CICS (and elsewhere) or defines interaction sets to detect patterns of events

Systems Programmer

Existing Application

CICS Event Processing Support

CICS TS

Event Consumer

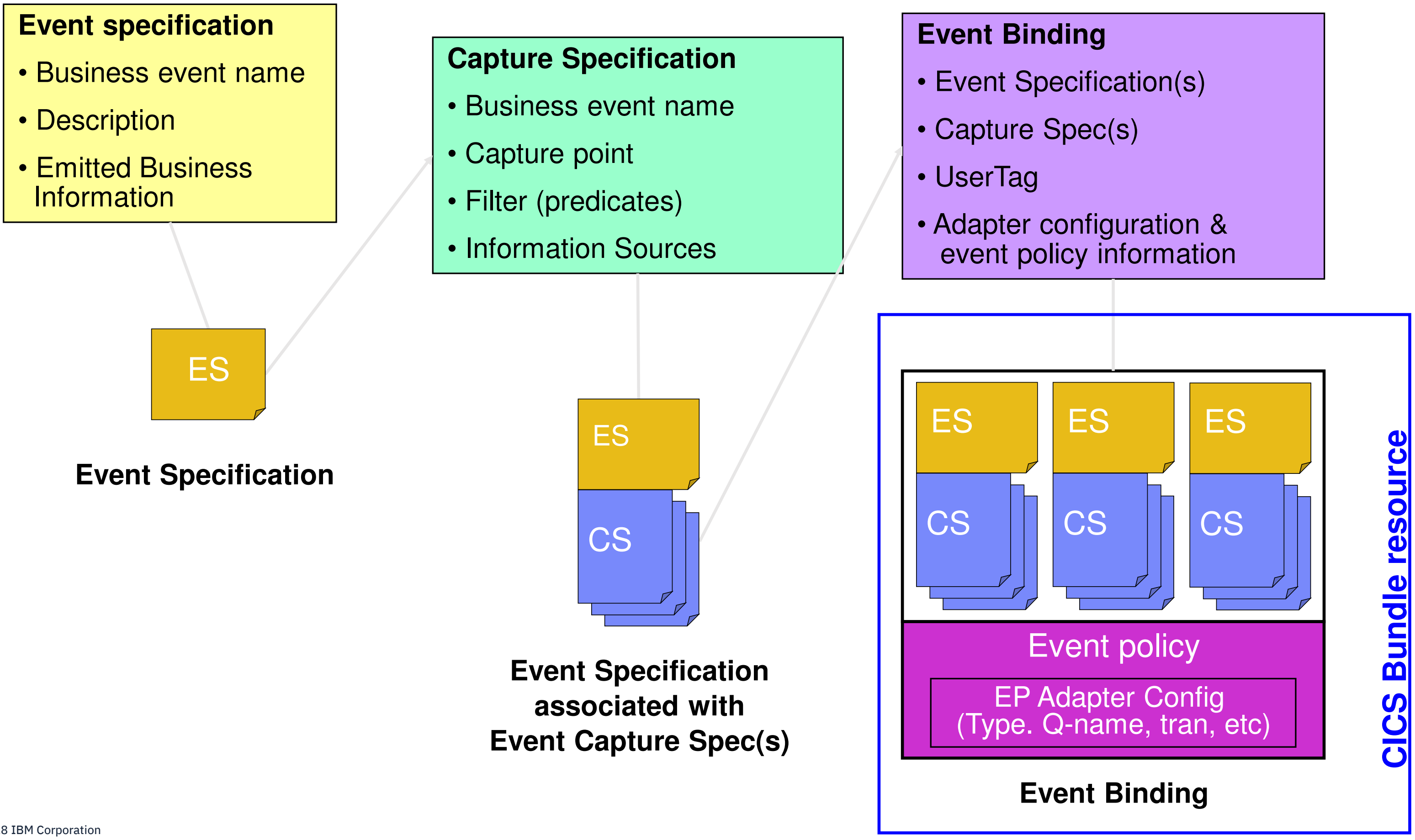
Business Modeler

Business Dashboard

LOB optionally configures a dashboard – Notes

- Optionally the LoB manager may chose to configure a business dashboard to allow an abstracted view of the events being created by the applications in the system, or might define a pattern to be detected between several events.

CICS Event Specification



Event specification
- Business event name
- Description
- Emitted Business Information

Capture Specification
- Business event name
- Capture point
- Filter (predicates)
- Information Sources

Event Binding
- Event Specification(s)
- Capture Spec(s)
- UserTag
- Adapter configuration & event policy information

Event Specification

Event Specification associated with Event Capture Spec(s)

Event Binding

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 type of CICS resource).

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.

The Event Binding

- XML specification of one or more related business events
  - Multiple business events can be grouped together where it makes sense to deploy and manage them as a single entity
  - These events can be quite different from each other, but will share
    - The same Event Processing adapter, with the same configuration
    - The same event processing policy (e.g. transactionality)

- Each event specification may have multiple capture specifications
  - More typically, each event specification will be associated with one capture specification
  - But a particular event may have several manifestations
    - e.g. “credit card usage” event may occur through several channels (Internet, in-store purchase, ATM machine)
  - Each capture specification must be able to provide the data to be emitted in the event (e.g. credit card number and customer ID) and in the same format

The Event Binding – Notes

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

Deployment of Event Bindings

- The Event Binding is deployed into CICS
  - Event Bindings are deployed via inclusion in a CICS Bundle
    - A CICS Bundle resource contains a collection of related CICS resources
      - An archive file containing resources, artifacts, etc. plus a manifest
      - Used for a number of new resources, including Event Bindings
      - Installing a BUNDLE into CICS will install the included resources
    - Export CICS Bundle to zFS (using CICS Explorer)
    - Create CICS Bundle resource definition, including the zFS location, and install in CICS

- Deploying the Event Binding into a particular CICS region will resolve the capture specifications

- CICS enables the capture specifications in the Event Binding so that they can be intercepted during runtime processing

Deployment of event bindings – Notes

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

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 tranid at a terminal)
- Getting data into or out of CICS can be a good way of finding out about business events
  - e.g. RECEIVE MAP, RECEIVE, SEND MAP. WRITE OPERATOR
- 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)**

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 (e.g. DB2 or WMQ) are not 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

- APIs
  - SIGNAL EVENT, INVOKE SERVICE, WRITE OPERATOR

- 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 and CICS TS V5.1.
- 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 applid qualifier & CICS applid, Date & time
  - Capturable (can be information source for an item of emitted business information):
    - Tranid, Current program, Userid

- Application command options and application data – Command-specific
  - e.g. For RECEIVE MAP
    - Filterable and capturable: MAP*, 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 will be captured *after* they occur, some offer the 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.
- 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.

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 pace, but are emitted if and when the unit of work in which the event took pace is committed.   If the unit of work is rolled back or abends, then no transactional events in that unit of work will be emitted.
- There are performance implications associated with defining events as transactional; most notably that these events will be kept in the system until the unit of work commits, so they can be less timely than non-transactional events.

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 MQ 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

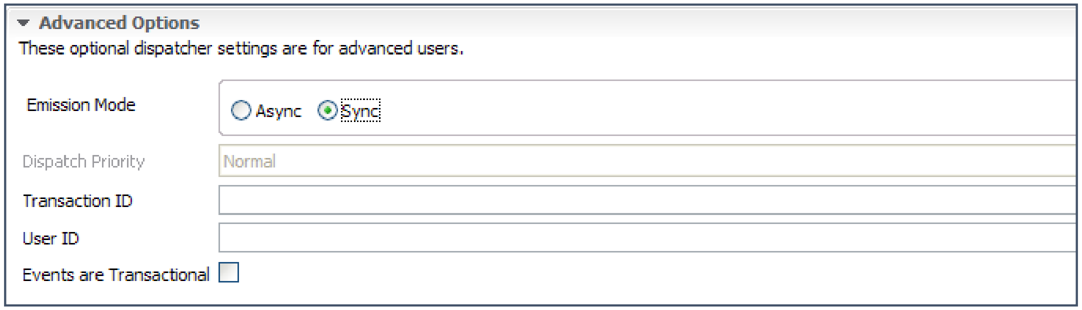
- Some performance implications of specifying tranid or userid

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).

CICS TS Event Processing: Synchronous mode

- New Synchronous emission mode option on the EP adapter
- When Synchronous emission is used event emission is assured
- Causes events to be formatted and emitted as part of the capturing UOW
- If the event fails to be emitted the capturing UOW will be backed out at syncpoint and its transaction is abended with ASP7



Advanced Options

These optional dispatcher settings are for advanced users.

Emission Mode ○ Async ● Sync

Dispatch Priority Normal

Transaction ID

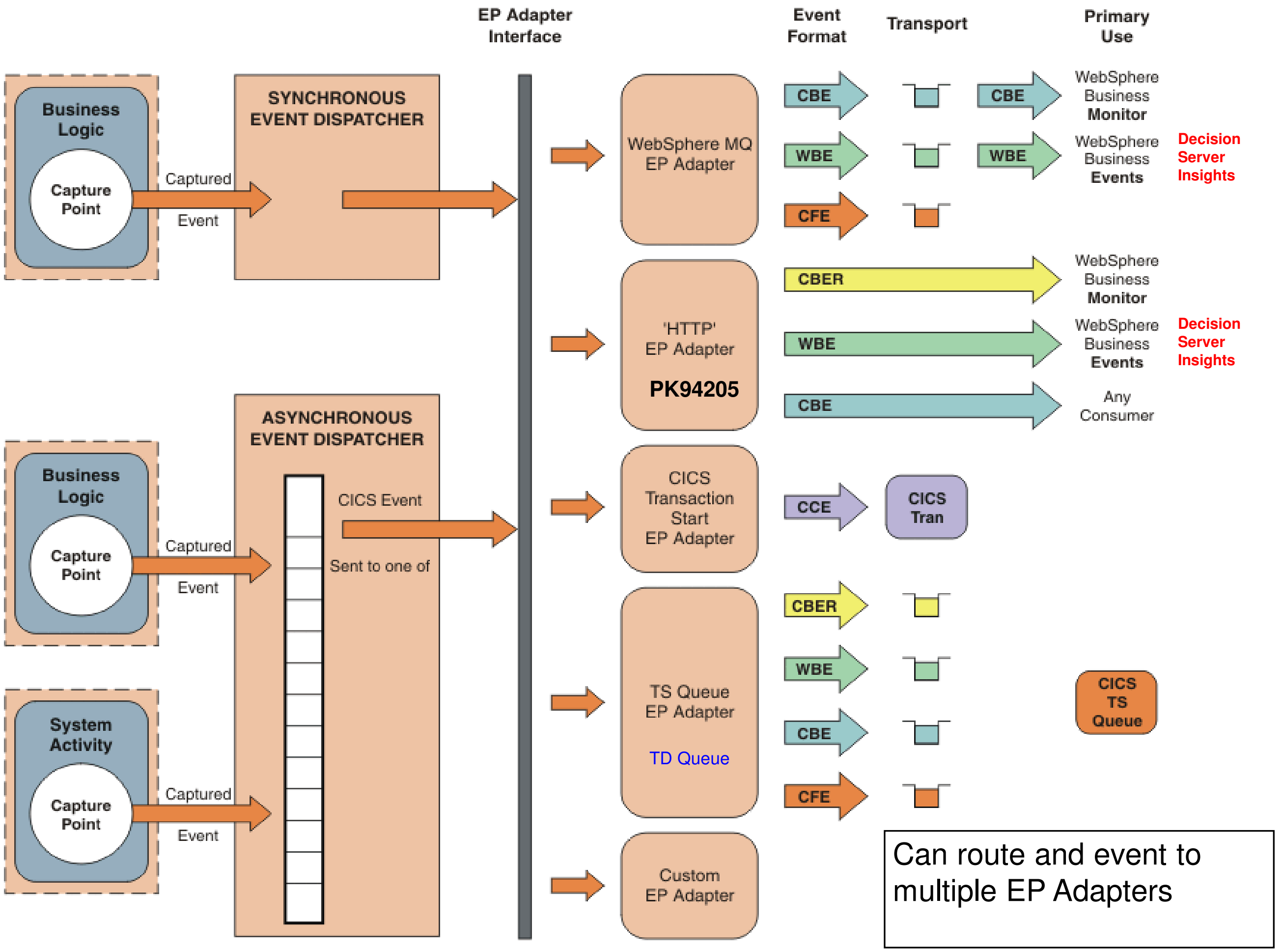
User ID

Events are Transactional □

Added in CICS TS V4.2 – Notes

- Notes page

CICS Event Processing Adapters



Can route and event to multiple EP 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.
- The EP adapters format the event and route it to the potential consumers, which include WebSphere Business Monitor, WebSphere Business Events, and a CICS transaction. More details are given on the next slide.
- This slide also illustrates the customization options for CICS Event Processing support:
  - The standard EP adapter interface allows users and IBM Business Partners to write custom EP adapters, to support formats and/or transports not provided by CICS
  - The event bindings created by the CICS Event Binding Editor conform to a schema, and the schema files are shipped with CICS, providing opportunities to create pre-built event bindings, or different tooling interfaces
  - The explicit SIGNAL EVENT API allows applications to be event-enabled, and could be used by application providers to include event opportunities within their applications at the places where events occur which might be of interest. These events could then be enabled as required, and tailored to the users needs.
- In CICS TS V5.1 you can route an event to multiple EP adapters

CICS EP Tooling

- Event Binding Editor
  - Eclipse editor feature, operating on event binding files (type .evbind)

- Event Binding Editor is a part of the CICS Explorer
  - Create event binding(s) within a **Bundle Project** in the Resource perspective
    - Creating or editing an event binding opens the Event Binding Editor
    - CICS Explorer provides support to deploy bundles containing event bindings

- Event binding can be built up in stages
  - Validated against schema each time it is saved, other validation as information added
  - A problems view keeps a running record of all errors

CICS EP Tooling – Notes

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

CICS TS Improvements (circa v4)

- More data types supported for filter and capture:
  - Sign leading/trailing Zoned Decimal
  - Hexadecimal floating point
  - Binary floating point
  - Decimal floating point (requires optional hardware facility)
  - Null terminated character
  - Null terminated hex

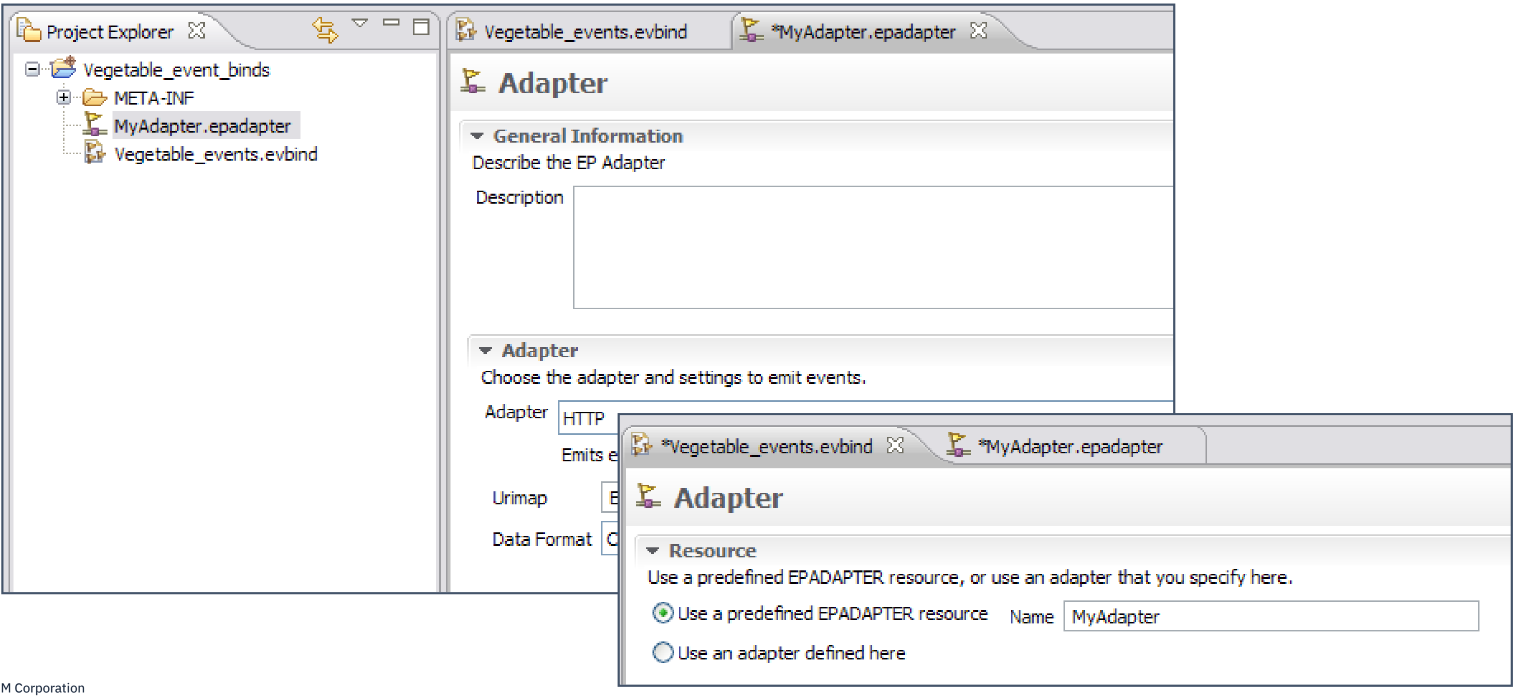
- HTTP EP adapter performance improvements if using a URIMAP with SOCKETCLOSE set

CICS TS V4.2 Improvements – Notes

- CICS TS V4.2 Improvements

CICS TS Improvements (circa v4)

- Adapter specifications can be defined and managed separately from the Event Bindings
- Event Bindings can reference these EP adapters by name



Project Explorer

- Vegetable_event_binds
  - META-INF
  - MyAdapter.epadapter
  - Vegetable_events.evbind

Vegetable_events.evbind | *MyAdapter.epadapter

Adapter

General Information

Describe the EP Adapter

Description

Adapter

Choose the adapter and settings to emit events.

Adapter: HTTP

Emits e

Urimap: E

Data Format: C

*Vegetable_events.evbind | *MyAdapter.epadapter

Adapter

Resource

Use a predefined EPADAPTER resource, or use an adapter that you specify here.

● Use a predefined EPADAPTER resource Name: MyAdapter

○ Use an adapter defined here

CICS TS V4.2 Improvements – Notes

- The slide lists CICS TS V4.2 improvements to event processing

CICS TS Improvements (circa v4)

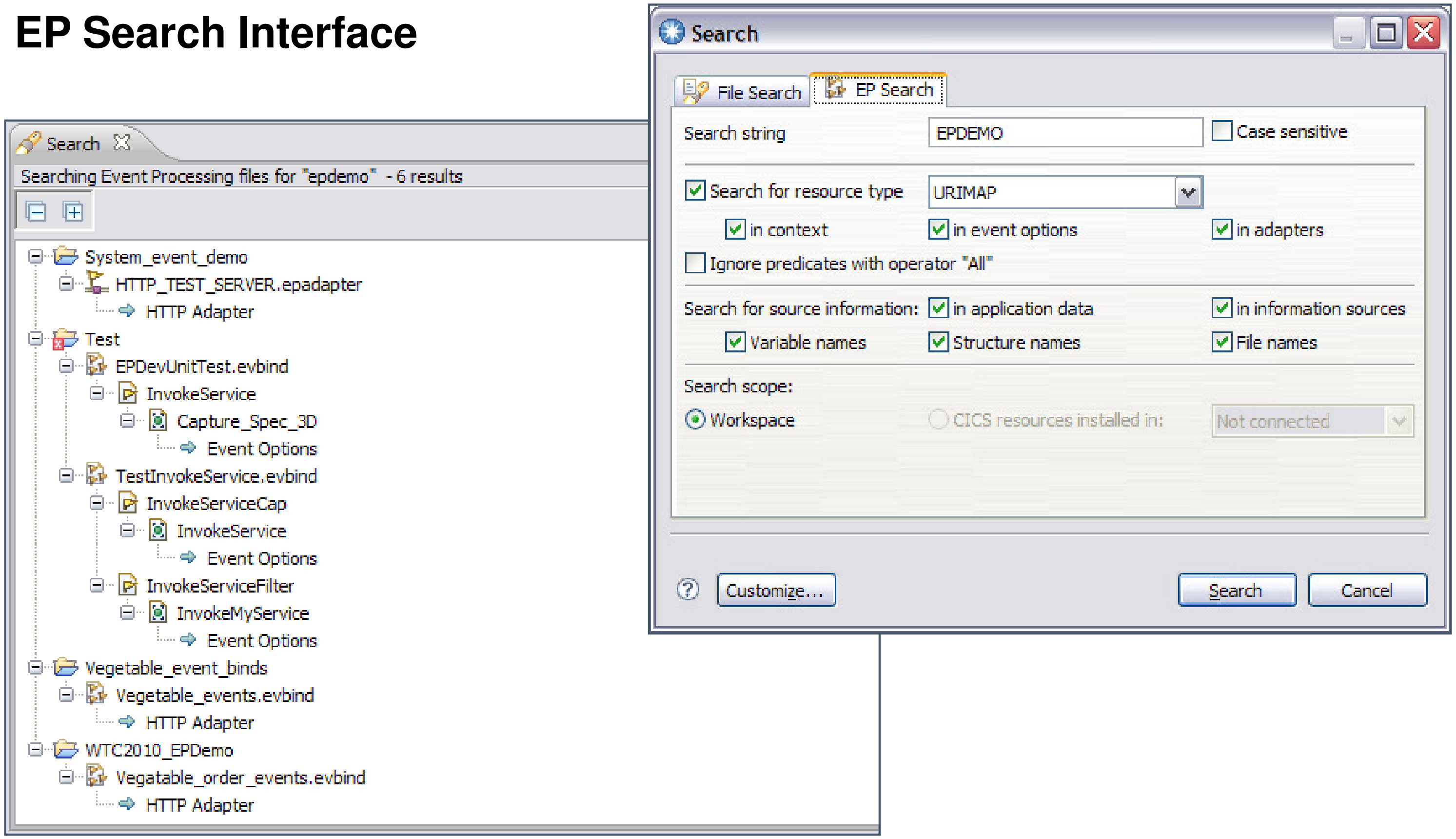
- **EP Search capability**
  - –Available in the CICS Explorer
    - Enter the name of the “thing” you have changed and EP search will tell you which event bindings may be affected
  - –‘Things’ include:
    - CICS resource names
    - Language Structure names
    - Language Structure field names
  - –Searches event bindings in the CICS Explorer workspace and those installed in CICS regions that Explorer is connected to

CICS TS V4.2 Improvements – Notes

- This slide lists improvements introduced in CICS TS V4.2

CICS TS Improvements (circa v4)

EP Search Interface



Search ☒

Searching Event Processing files for "epdemo" - 6 results

```
⊟ System_event_demo
    ⊟ HTTP_TEST_SERVER.epadapter
        HTTP Adapter
⊟ Test
    ⊟ EPDevUnitTest.evbind
        ⊟ InvokeService
            ⊟ Capture_Spec_3D
                Event Options
    ⊟ TestInvokeService.evbind
        ⊟ InvokeServiceCap
            ⊟ InvokeService
                Event Options
        ⊟ InvokeServiceFilter
            ⊟ InvokeMyService
                Event Options
⊟ Vegetable_event_binds
    ⊟ Vegetable_events.evbind
        HTTP Adapter
⊟ WTC2010_EPDemo
    ⊟ Vegatable_order_events.evbind
        HTTP Adapter
```

Search

File Search | EP Search

Search string: EPDEMO □ Case sensitive

☑ Search for resource type: URIMAP

☑ in context ☑ in event options ☑ in adapters

□ Ignore predicates with operator "All"

Search for source information: ☑ in application data ☑ in information sources

☑ Variable names ☑ Structure names ☑ File names

Search scope:

● Workspace ○ CICS resources installed in: Not connected

CICS TS V4.2 Improvements – Notes

- The slide lists improvements to event processing in CICS TS V4.2

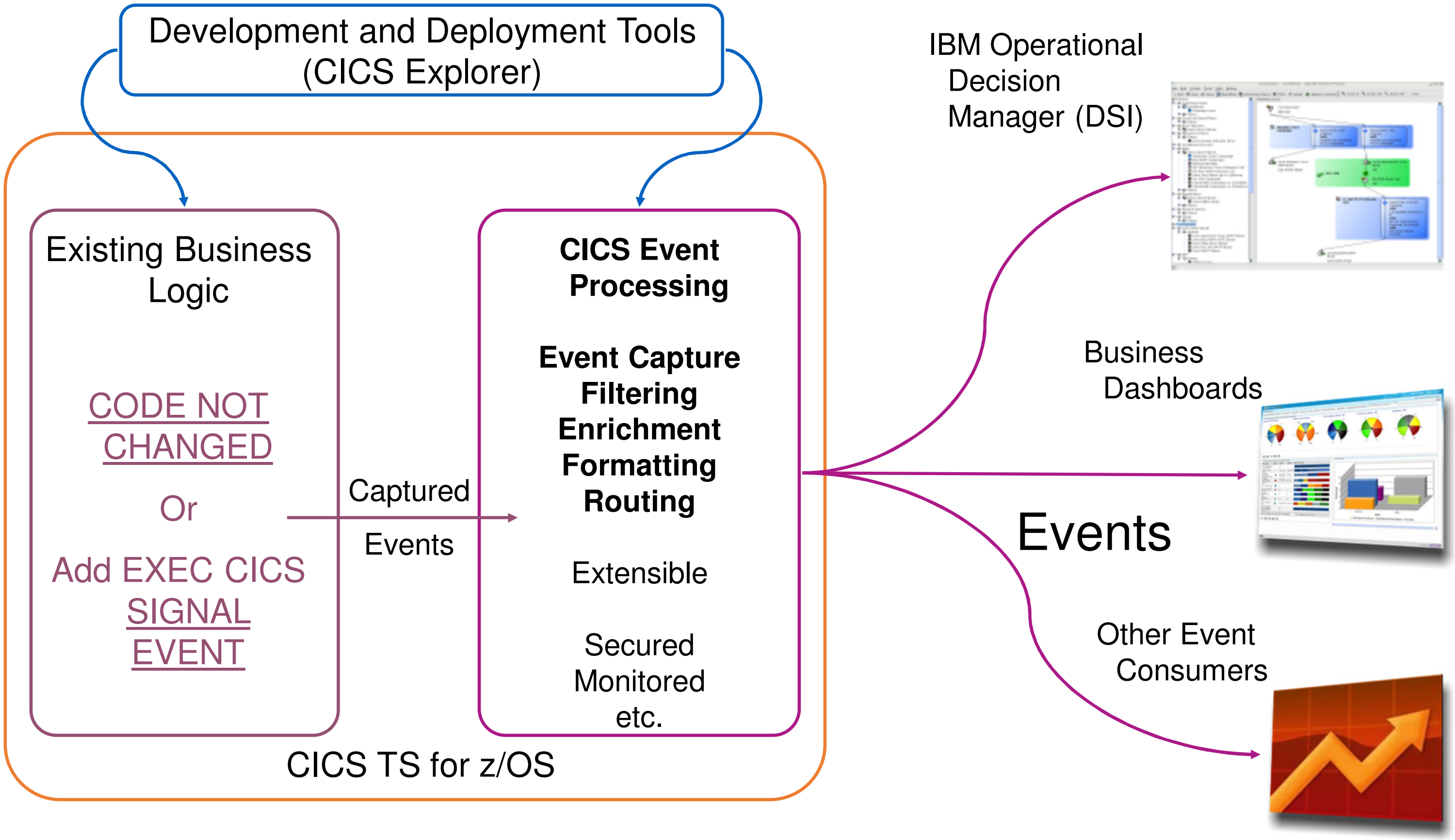
CICS TS V5 Improvements

- Application Events
  - EXEC CICS WRITE OPERATOR
- Static Data
  - Can now define static data in an event
- EP Search
  - Can now search for system message events
- Multiple EP adapters
  - Can now use a set of adapters to emit events

CICS TS V5.1 Improvements – Notes

- CICS TS V5.1 improvements

CICS Event Processing Summary



Development and Deployment Tools (CICS Explorer)

Existing Business Logic

CODE NOT CHANGED

Or

Add EXEC CICS SIGNAL EVENT

Captured Events

CICS Event Processing

Event Capture
Filtering
Enrichment
Formatting
Routing

Extensible

Secured
Monitored
etc.

CICS TS for z/OS

Events

IBM Operational Decision Manager (DSI)

Business Dashboards

Other Event Consumers

IBM **CICS**
Transaction Server for z/OS

CICS Policies

CICS policy-based management

- Control the run time behaviour of CICS based on a policy
  - Declarative mechanism to ensure CICS workload and systems continue to run effectively

- A policy defines the conditions and actions to take

  CICS user task makes excessive use of system resources
  e.g. a user task consumes too much storage

  CICS system or user task changes the state of a system resource
  e.g. a FILE resource is closed

  CICS overall system health changes
  e.g. number of active task exceeds the maximum user tasks in the CICS system (the MXT value)

- Several policies can be installed
  e.g. one for each application, and one for the CICS region

Terminology

- CICS bundle contains one or more policies and other CICS resources

- Policy is an XML definition in a CICS bundle that contains one or more policy rules
- Policy rule defines the rule type + condition + action

- Rule types are grouped into system rules and task rules - see following slides
- Condition is the criteria that has to be true before taking the action
- Action is the logic to perform
  - message, emit event, abend, or API response

Policy Task Rules

- Database request
  - the number of DB2 SQL / IMS DLI requests performed by a user task
- File request
  - the number of EXEC CICS file access requests performed by a user task
- Program request
  - the number of EXEC CICS LINK or EXEC CICS INVOKE APPLICATION requests that are performed by a user task
- Storage allocation
  - the amount of storage that is allocated by a user task
- Storage request
  - the number of GETMAIN requests performed by a user task
- Time
  - the amount of processor time that is used by a user task (CPU time policy item)
  - the amount of elapsed time that is taken by a task (Elapsed time policy item)
- Async request
  - the number of EXEC CICS RUN TRANSID requests performed by a user task

Policy Task Rules (continued)

- TS queue bytes (auxiliary, main, or shared)
  - the total amount of data that is written by a user task to temporary storage
- TS queue requests (auxiliary, main, or shared
  - the number of EXEC CICS READQ TS and EXEC CICS WRITEQ TS requests issued by a user task
- TD queue requests
  - the number of EXEC CICS READQ TD and EXEC CICS WRITEQ TD requests issued by a user task
- Start requests
  - the number of EXEC CICS START requests that are performed by a user task
- Syncpoint requests
  - the number of EXEC CICS SYNCPOINT requests that are performed by a user task
- EXEC CICS requests
  - the number of EXEC CICS requests performed by a user task

Policy Task Rules (continued)

- IBM MQ requests
  - the number of MQ requests performed by a user task

- Name Counter Server requests
  - the number of EXEC CICS GET COUNTER or GET DCOUNTER requests

- Container storage
  - define a threshold for the amount of container storage allocated to a user task

Policy System Rules

- Support for system rules to help monitor overall system health
  - the status of a DB2 connection changes from or to a specific state
    - DB2 connection status rule
  - the open status of a CICS FILE changes from or to a specific open state
    - File open status rule
  - the enable status of a CICS FILE changes from or to a specific enablement state
    - File enable status rule
  - CICS issues a DFHxxnnnn message or CICSPlex SM issues an EYUxxnnnn message
    - Message rule
  - the number of active tasks in a CICS transaction class goes above or below a percentage of a transaction class MAXACTIVE value
    - Transaction class tasks rule
  - a transaction encounters an unhandled abend
    - Transaction abend rule
  - the number of active tasks in a CICS system goes above or below a percentage of the MXT value
    - User tasks rule

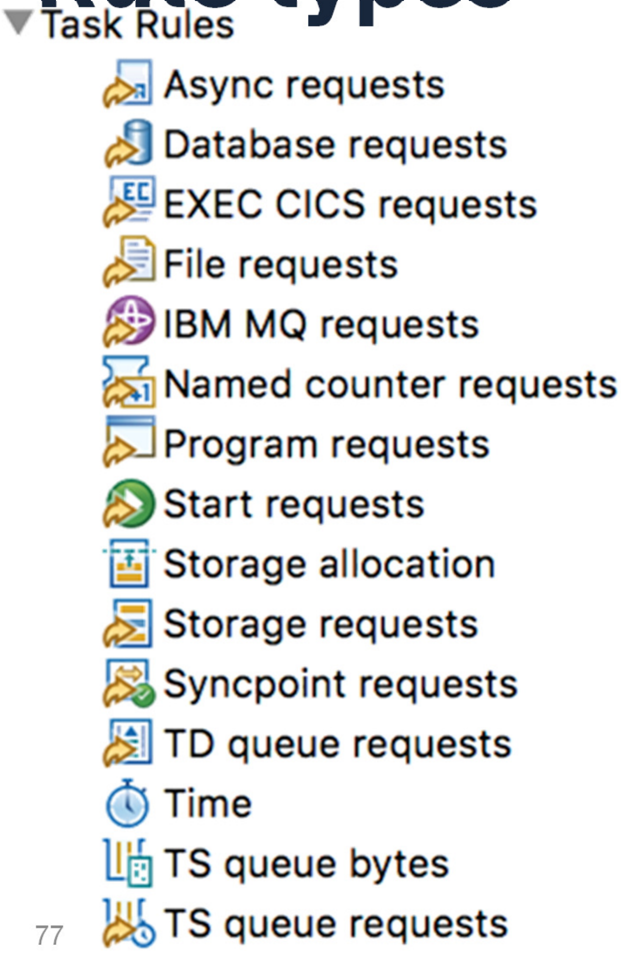
Policy system rules (continued)

- Support for system rules to help monitor overall system health
  - request would cause the current number of AIDs in the CICS system to exceed a threshold
    - AID threshold rule
  - the available status of a bundle that declares application entry points changes from or to a specific state
    - Bundle available status rule (application entry point)
  - the enable status of a bundle changes from or to a specific state, or when the enable status of a bundle changes from a specific state to another specific state
    - Bundle enable status rule
  - the status of a IPIC connection (IPCONN) changes from or to a specific state
    - IPIC connection status rule
  - the status of a MRO connection changes from or to a specific state
    - MRO connection status rule

Policy system rules (continued)

- Support for system rules to help monitor overall system health
  - the enable status of a CICS program changes from or to a specific state
    - Program enable status rule
  - monitor / react to the change in status of a connection between CICS and DBCTL
    - DBCTL connection status
  - monitor / react to the change in status of a connection between CICS and IBM MQ
    - IBM MQ connection status
  - monitor / react to the change in the enable status of a CICS PIPELINE resource
    - Pipeline enable status
  - set a maximum threshold for the total number of transaction dumps in a CICS region
    - Transaction dump threshold

Rule types



- Task Rules
  - Async requests
  - Database requests
  - EXEC CICS requests
  - File requests
  - IBM MQ requests
  - Named counter requests
  - Program requests
  - Start requests
  - Storage allocation
  - Storage requests
  - Syncpoint requests
  - TD queue requests
  - Time
  - TS queue bytes
  - TS queue requests

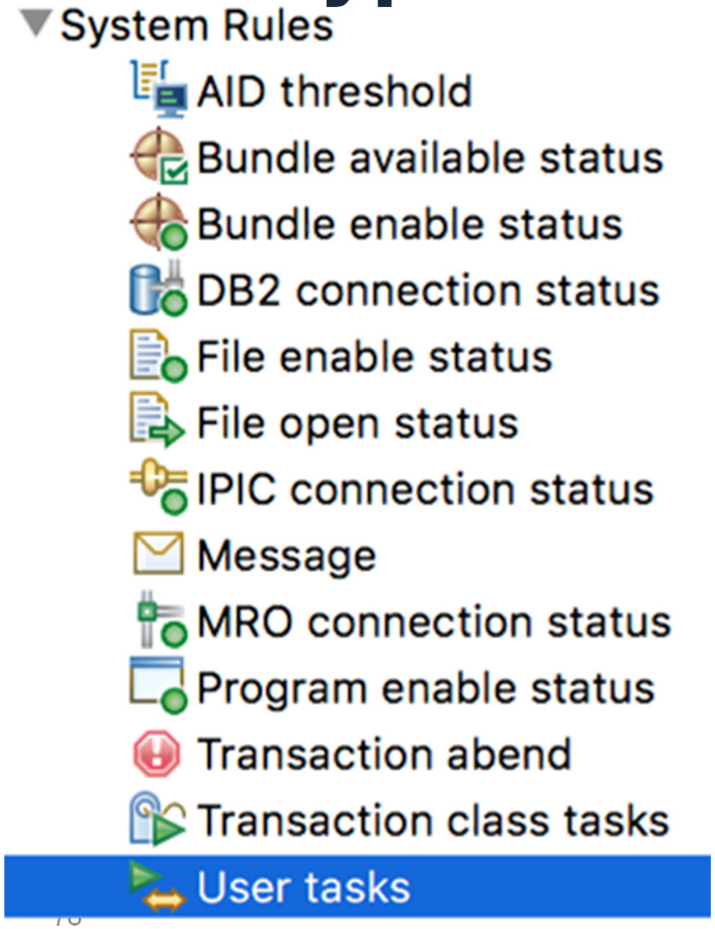
Use **task rules** to monitor the resource utilization of individual user tasks and to automatically respond when a tasks resource usage exceeds a pre-defined threshold. In this way, excessive resource usage, and looping transactions can be detected and dealt with appropriately.

Actions supported

1. Issue a message
2. Emit an event
3. Abend the task

Details in topic Policy task rules

Rule types



- System Rules
  - AID threshold
  - Bundle available status
  - Bundle enable status
  - DB2 connection status
  - File enable status
  - File open status
  - IPIC connection status
  - Message
  - MRO connection status
  - Program enable status
  - Transaction abend
  - Transaction class tasks
  - User tasks

Use **system rules** to monitor the state of critical system resources or the overall health of a CICS system and to respond automatically when any changes occur.

Actions supported

1. Issue a message
2. Emit an event
3. Change WLMHEALTH value

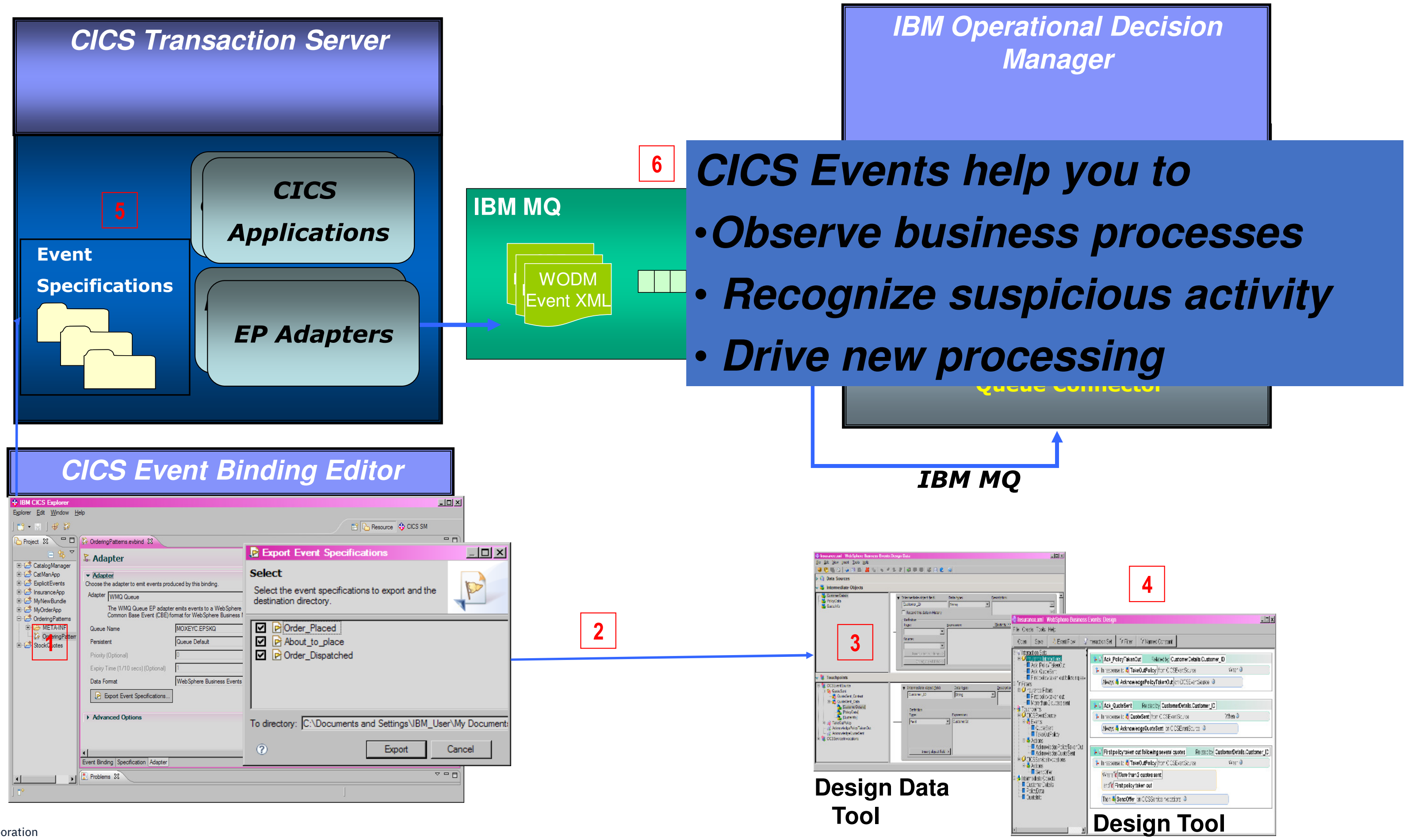
Details in topic Policy system rules

Policy Rules

- Policy rules are defined in a CICS bundle
- A condition and action pair make up a policy rule
- Multiple policy rules can be packaged in a bundle

- Policy evaluation
  - All rules that apply to the same scope and threshold are calculated
  - Rules are applied from lowest to highest threshold
  - Action are processed in the following order
    - Message actions
    - Event actions
    - Abend actions

CICS and IBM Operational Decision Manager



CICS Events help you to

- Observe business processes
- Recognize suspicious activity
- Drive new processing

IBM CICS Explorer

Adapter

Choose the adapter to emit events produced by this binding.

Adapter: WMQ Queue

The WMQ Queue EP adapter emits events to a WebSphere Common Base Event (CBE) format for WebSphere Business

Queue Name: MOXEYC.EPSKQ

Persistent: Queue Default

Priority (Optional): 0

Expiry Time (1/10 secs) (Optional): 1

Data Format: WebSphere Business Events

Export Event Specifications

Select

Select the event specifications to export and the destination directory.

☑ Order_Placed
☑ About_to_place
☑ Order_Dispatched

To directory: C:\Documents and Settings\IBM_User\My Document

CICS and WebSphere Business Events Integration – Notes

- This slide shows how CICS events are integrated with IBM Operational Decision Manager, which provides the ability to detect patterns between multiple events over time.
- The figure shows the CICS Event Binding Editor used to create event specifications which can then be installed in CICS TS V4 or CICS TS V5.1. The IBM Operational Decision Manager Design Data tooling can create event definitions from event schemas exported from the event binding editor, and these event definitions can be used in the Design tooling to indicate the patterns or interactions to be detected. The event definitions specify a Message Queue connection in order to receive the events over MQ.
- When events are captured from CICS applications, they are processed by the appropriate Event Processing Adapter. For IBM Operation Decision Manager, the EP adapter puts the event in the XML format recognised by ODM onto a WMQ queue on z/OS, which is configured to be received by the IBM Operation Decision Manger runtime. The connection could, for example, use the WAS platform messaging provided by the WAS in which ODM runs. ODM can also run on z/Os.

CICS SMART Seminar

Attend one of our free technical seminars and learn how to increase productivity of your staff and other resources, and prepare for the latest CICS technology

Every application development team faces the same mandate: Do it faster, more efficiently and use fewer resources. But whether you master that challenge depends in large part on your in-house technology, how well you use it and how prepared you are for the latest advancements in application development technology.

Join us in this complimentary technology seminar for a full day of in-depth discussions on how CICS solutions can help solve some of your biggest challenges. You will learn about the benefits of moving to the CICS Transaction Server V5.3 release, the IBM tools portfolio, and important connectivity options as you move your CICS applications to a Service Oriented Architecture.

You will also have the unique opportunity to interact with technical professionals who have extensive experience in application development tools and infrastructure, so that you can make better-informed decisions and gain control of critical processes.

Agenda

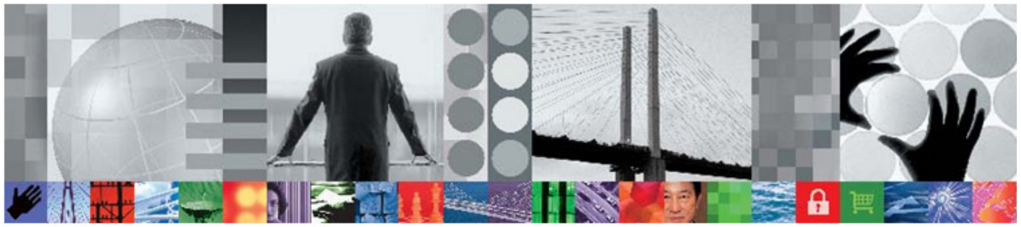
09:00	*Welcome - Opening comments*
09:15 – 11:00	*CICS Transaction Server V5 Overview*
11:00 – 12:00	*CICS Cloud Enablement*
	Working Lunch Served
12:00 – 01:30	*CICS Explorer / CICS Tools Update*
01:30 – 02:30	*New Java Improvements in CICS*
02:30 – 03:30	*CICS Event Processing / CICS Atom Feeds*
03:30 – 03:45	*Wrap-up*

Location and date:

Company
Company address
Company City, State, zip
Date

Register:

Name / Email



CICS SMART Seminar

Attend one of our free technical seminars and learn how to increase productivity of your staff and other resources, and prepare for the latest CICS technology

Every application development team faces the same mandate: Do it faster, more efficiently and use fewer resources. But whether you master that challenge depends in large part on your in-house technology, how well you use it and how prepared you are for the latest advancements in application development technology.

view
te
om Feeds

Join us in this complimentary technology seminar for a full day of in-depth discussions on how CICS solutions can help solve some of your biggest challenges. You will learn about the benefits of moving to the CICS Transaction Server V5.3 release, the IBM tools portfolio, and important connectivity options as you move your CICS applications to a Service Oriented Architecture.

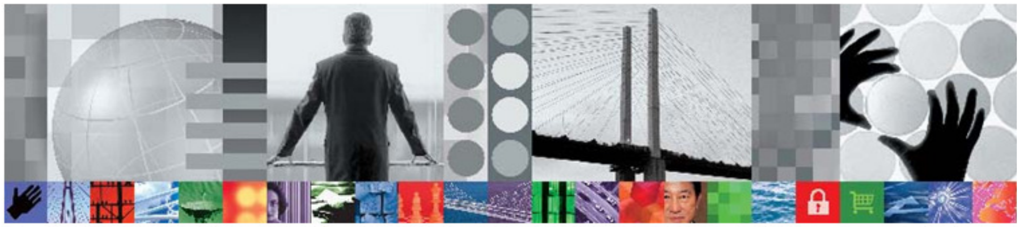
You will also have the unique opportunity to interact with technical professionals who have extensive experience in application development tools and infrastructure, so that you can make better-informed decisions and gain control of critical processes.

CICS SMART Seminar

Attend one of our free technical seminars and learn how to increase productivity of your staff and other resources, and prepare for the latest CICS technology

Agenda

09:00	*Welcome - Opening comments*
09:15 – 11:00	*CICS Transaction Server V5 Overview*
11:00 – 12:00	*CICS Cloud Enablement*
	Working Lunch Served
12:00 – 01:30	*CICS Explorer / CICS Tools Update*
01:30 – 02:30	*New Java Improvements in CICS*
02:30 – 03:30	*CICS Event Processing / CICS Atom Feeds*
03:30 – 03:45	*Wrap-up*



IBM

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 IBM Operational Decision Manger, 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 IBM Operation Decision Manger) 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.

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 and continued in CICs TS V4.2 and V5.1. This provides our strategic direction for integration with IBM event processing products.
- CICS support for events allows CICS applications to emit business events in a non-invasive way, where such flexibility is required.
- A 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 IBM Operational Decision Manger and WebSphere Business Monitor.
- There is also the ability to write custom EP adapters to support other formats and ways of emitting events.