
CICS Events and Web 2.0

Share Tech Conference

August 2010

Wayne Duquaine

Grandview Systems

Phone: 707-829-9633

E-mail: grandvu@sonic.net

Outline

- SOA and CICS
- Web 2.0 and Events
- Walkthrough how to define CICS Events
- Simple Example using COMMAREA based VSAM App
- Invoke that VSAM App using Web Services

SOA Review

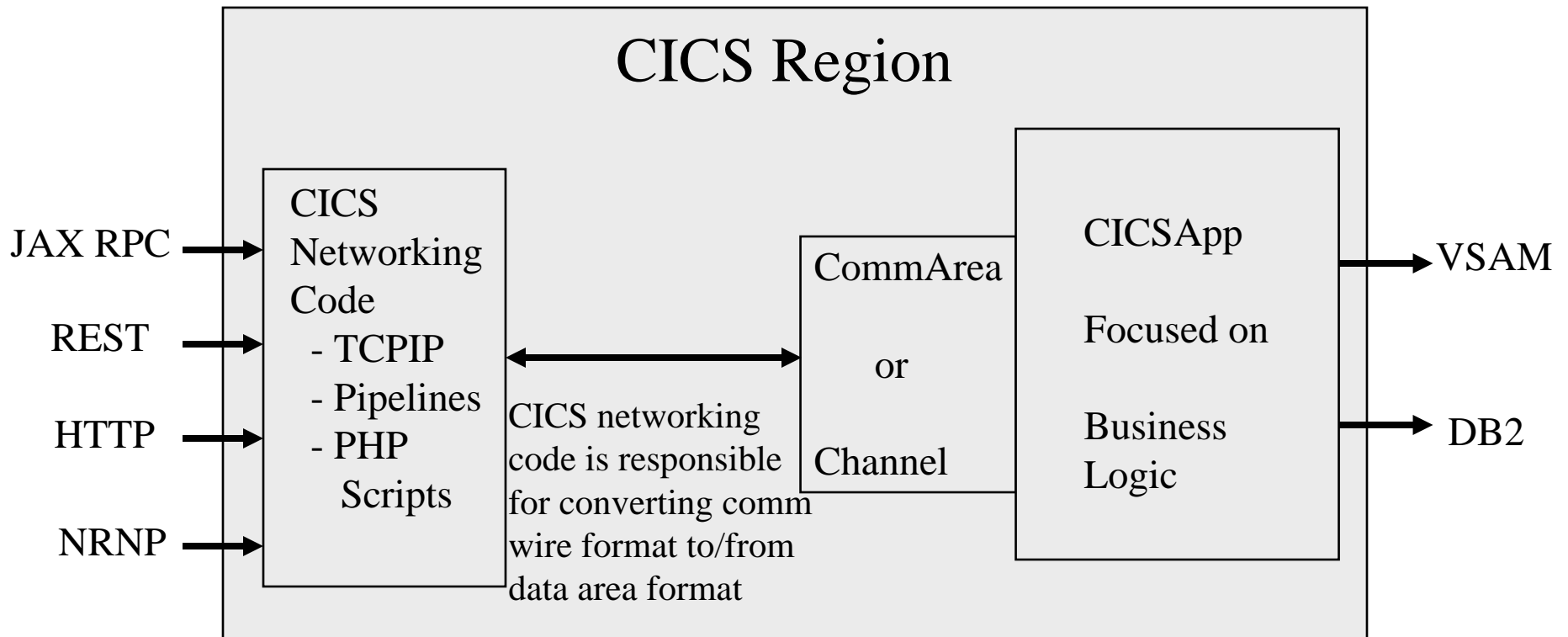
- SOA - Service Oriented Architecture
 - » Architectural style based on a service orientation
 - » Objectives: - enable business process automation
 - integrate information across apps (portals)
 - better interoperability between applications
 - » Applications call a Service, not a network API.
- Service = a repeatable business task.
(open new account, check customer credit, refresh inventory, ...)
- SOA is part of the overall Web 2.0 architecture
 - » Part of the Web Services initiative

SOA and CICS

- Under CICS, the **pragmatic** long term essence of SOA design can be boiled down to:
 - » Get EXEC CICS RECEIVE calls out of your apps
 - » Get EXEC CICS SEND calls out of your apps
 - » In other words, get all networking code out of your CICS apps
 - » Make your apps COMMAREA or CHANNEL driven apps
 - » Use Web Services as the front-ends to your apps
- Why Get Rid of Networking Code in CICS Apps (EXEC CICS SEND/RCV)
 - » Because the networking people have gone nuts with protocols:
 - JAX-RPC, JMS, REST, IIOP, RMI, RSS, ATOM, IPIC, Google Data, . . .
 - The continual maintenance tweaks/burdens will drive you crazy
 - » Allows the CICS Apps to be invoked regardless of underlying protocol
 - Let the CICS run-time worry about the network
 - Let the apps focus on the business logic (not on the network plumbing)

SOA and CICS

- Let CICS do your networking.



As each new random network protocol comes out, your apps are not impacted.

Web 2.0 Review - 1

- Goals of Web 2.0:

1. Give user control over functions and interactions of their Web experience.

2. Facilitate Machine to Machine interaction

- Service oriented calls (vs human oriented HTML)
- XML to provide self-describing data
- AJAX callbacks to make Web pages more interactive
- Ability to tweak RSS/ATOM feeds (publish/subscribe control)

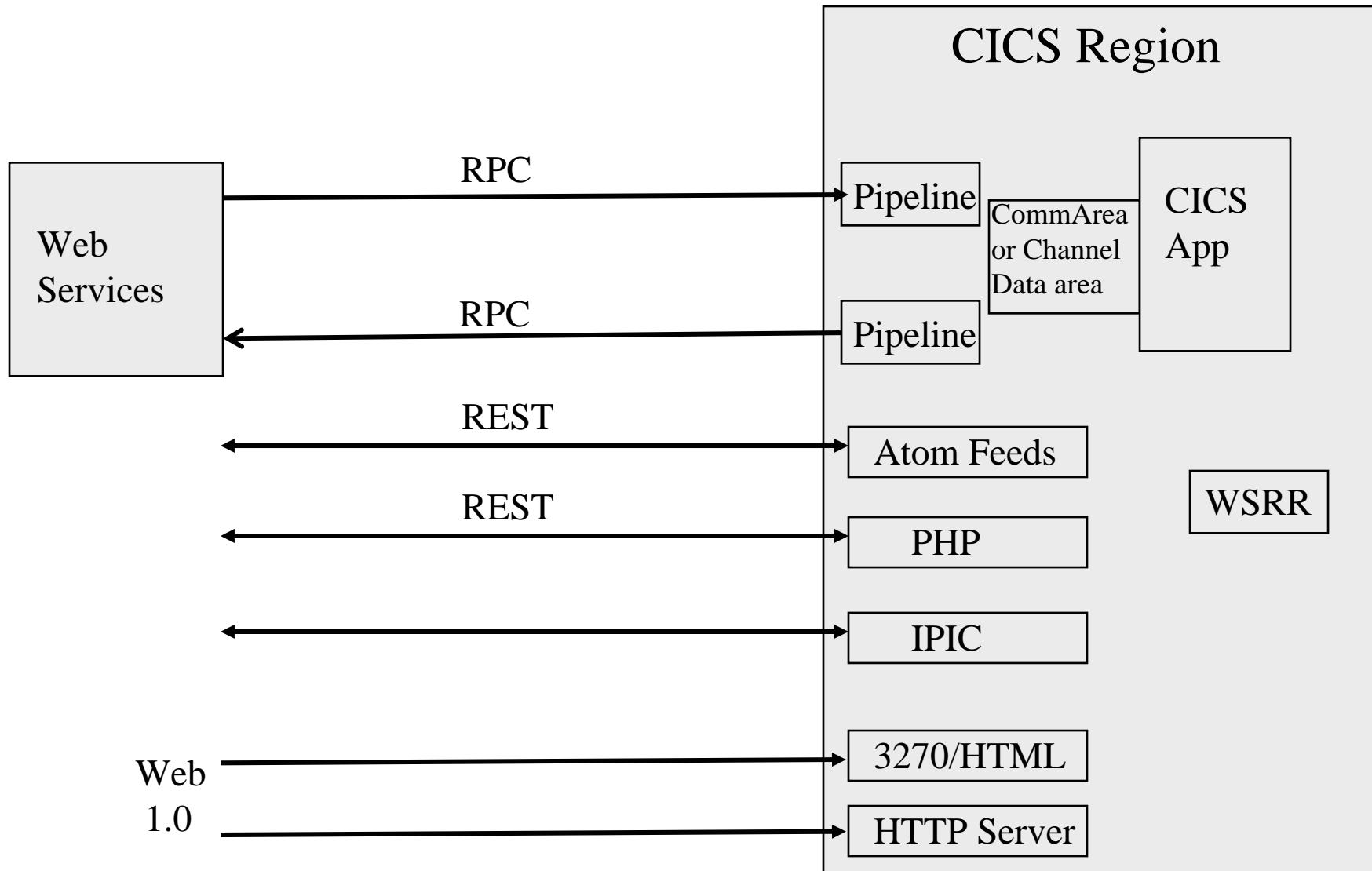
Web 2.0 Review - 2

- Primary architectural pieces of Web 2.0 (“design patterns”)
 - » SOA – Applications use Service invocation based APIs
 - » SaaS – “Software as a Service” – Service that is shared across organizational boundaries, e.g. an outside service, paid for by use.
 - » Asynchronous Updates – AJAX and related technologies that allow individual pieces of an HTML screen to be asynchronously updated while user is navigating the form.
 - » Mashups – Multiple remote data from different sources aggregated onto a single screen.
 - » RIA – “Rich Internet Application” – App that integrates multiple media formats and sources as the user interactively navigates the screen.
 - » Synchronized Web – data on local system that is periodically synchronized with remote data (e.g. data stored in the “Cloud”).
 - » Participation-Collaboration – remote users coordinate access/update of documents, designs, etc.
- At its core, Web 2.0 is an event driven architecture.

Web 2.0 and CICS

- CICS TS 4.1 has done much to improve its Web 2.0 capability
 - » Create ATOM feeds of CICS data without requiring any user application
 - » Improved Web Services support (WSRR, SCA)
 - » Easier support for AJAX
 - » Support for PHP (aka Dynamic Scripting).

Web 2.0 support on CICS

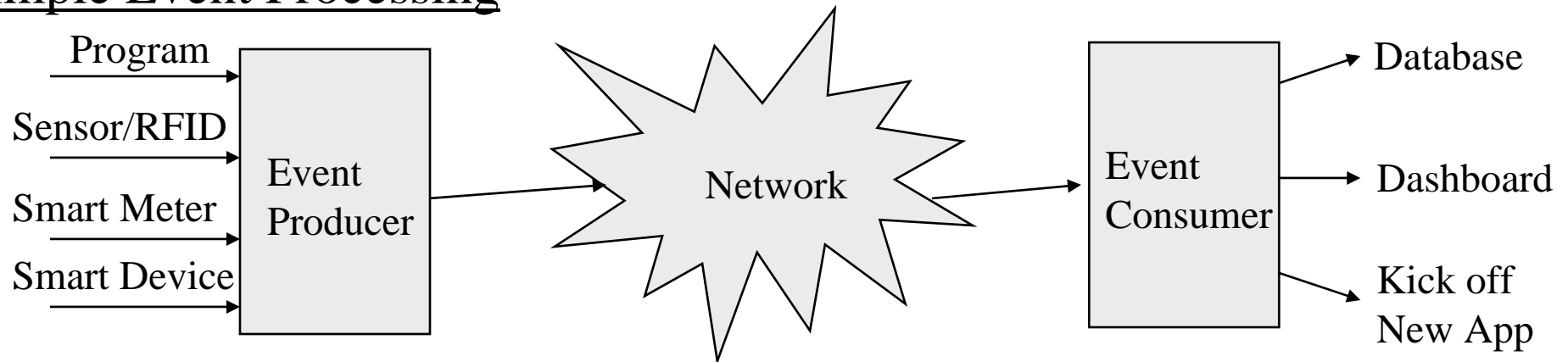


Event Basics - 1

- Business Event – something that happens that is relevant to the business
 - Order Received
 - Credit Limit Exceeded
 - Stock Trade
 - . . .
- Producer – something (program or sensor) that produces events
- Broker - optional component between producer and consumer that is used to monitor/compare/combine data from multiple events
- Consumer – consumes and reacts to events
- Events are always asynchronous
 - » Consumer app is completely de-coupled from the Producer app

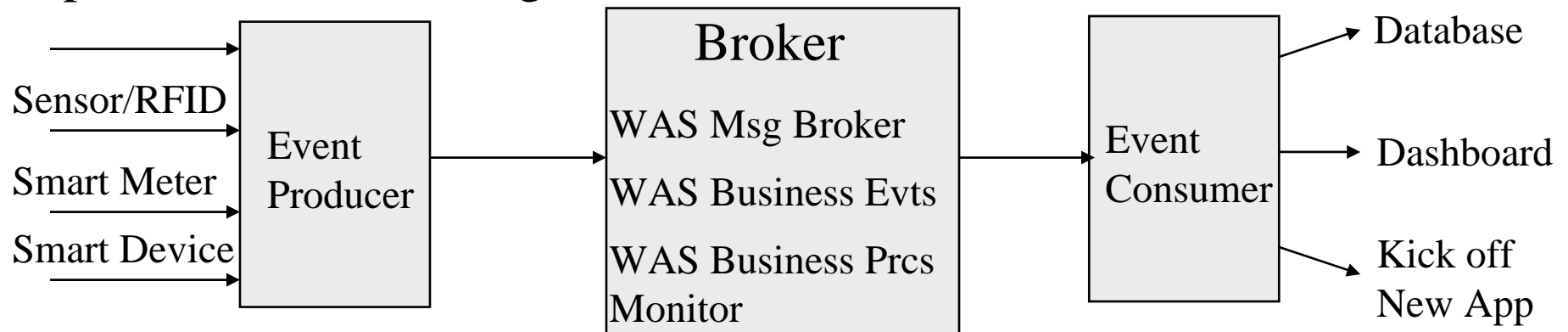
Event Basics - 2

Simple Event Processing



Typically a pub/sub type processing model

Complex Event Processing



Broker is typically a script engine or rules based engine that does additional processing on the Raw events. May use pattern-matching to monitor or combine multiple events.

Why Use Events

- Further automate existing processes
 - » Automate B2B processes for inventory re-stocking, credit limit checking, ...
 - » Reduce “middle-men” – e.g. users directly enter orders online
 - » Support JIT manufacturing, delivery, inventory control, ...
 - » Fraud detection
- Enable new applications/initiatives
 - » Real-time (sensor-based) monitoring
 - Smart Metering
 - “Intelligent Utility” Monitoring – Gas pipelines, Electrical Grid loads, ...
 - Anything “Green” is sensor based
 - » Within 10 years, half of all Internet traffic will be M2M, all of it event based.

Event Generation – What Needs to be Done

Conceptually, to capture and report events, you need to:

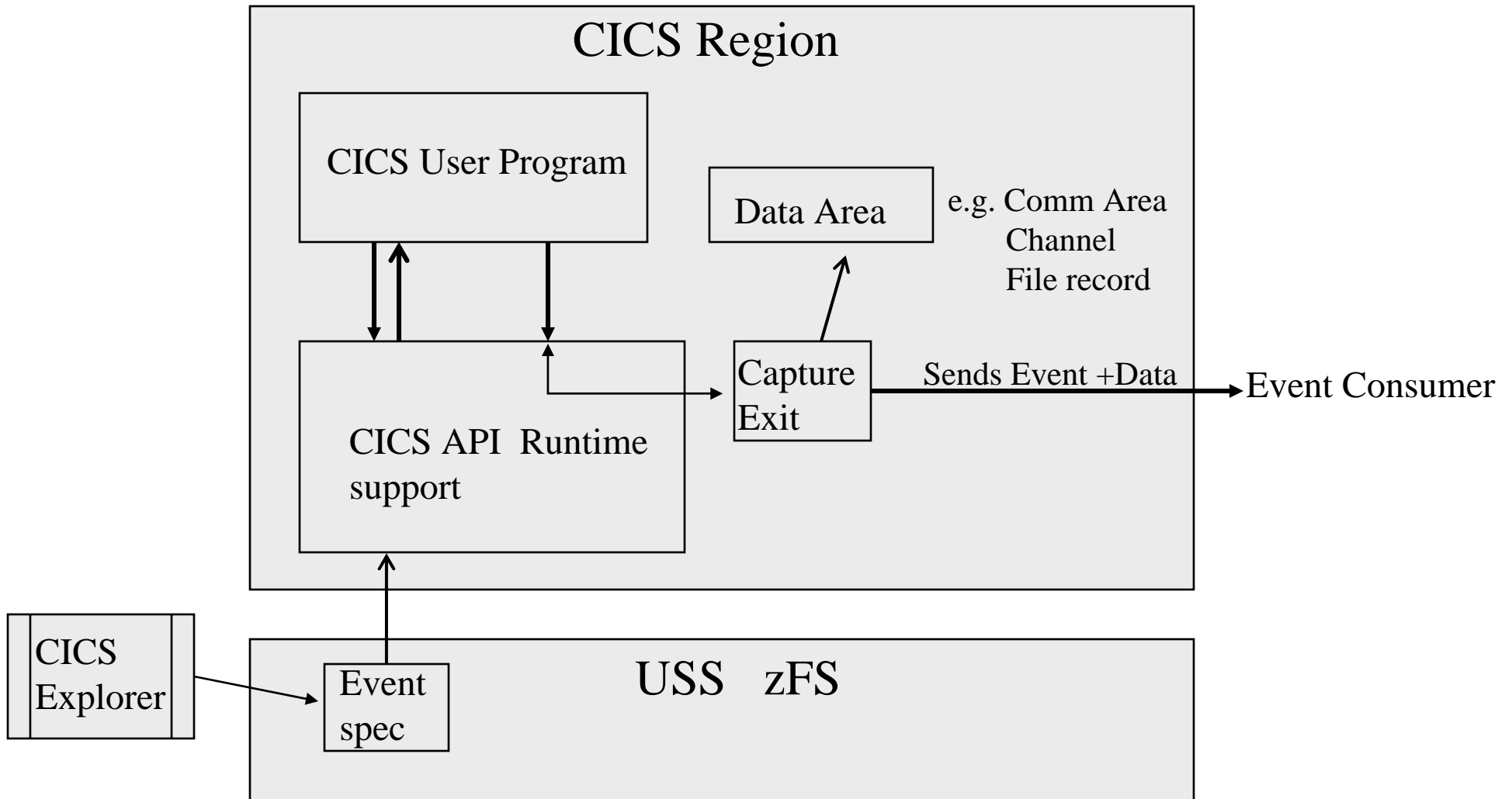
- Specify where to capture the event (where in the program)
- Filter the data (send events only for data you care about)
 - » Specify which context information (user-id, ...) data fields to look at in the CommArea of Channel data record.
 - » Specify what criteria they need to match against. Is the field above, below, or equal to a specific value (e.g. Credit Limit check).
- Give the event a name and what data fields to send on it
 - » Event name is needed so event consumer or broker knows what type of it is. Send key data fields with the event for the consumer to process
- Specify where the event is to be sent to

Event Generation on CICS

- Events can be generated two different ways:
 - » Use CICS API Event Capture Exits provided in CICS TS 4.1 run-time
 - Can add events to existing CICS apps without changing the app
 - New runtime support can capture relevant data out of a COMMAREA or CHANNEL data area
 - When and where to capture is specified via new CICS Explorer tool, that creates a special “event bundle” used by CICS runtime
 - » Use the new CICS EXEC CICS SIGNAL EVENT call
- Events generated by CICS can be sent off to:
 - » Initiate CICS Program Starts
 - » TS Queues
 - » MQ (CFE XML)
 - » WebSphere Business Events (WBE XML) or Websphere Business Process Monitor (CBE XML)
 - » Custom written Event Adapter

CICS Events Infrastructure

- Primary Moving Parts



Creating Event Definitions

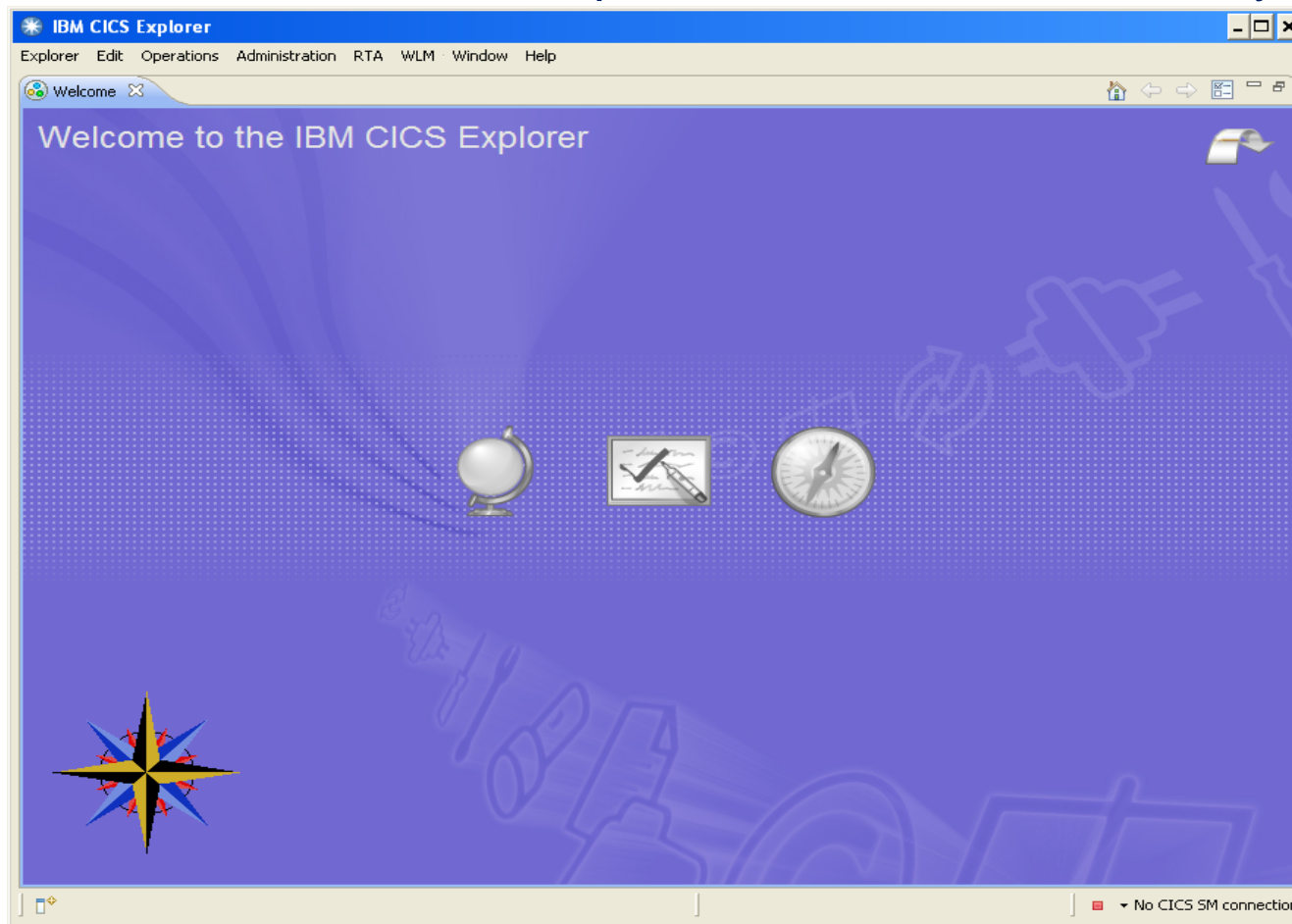
- CICS Explorer is used to create all Event Definitions
- Any event definition involves defining:
 - » Event + Data to be sent on the Event
 - » Conditions (Filtering)
 - » Where event is to be sent to (which Event Adapter EP)
- Three primary pieces need to be specified in CICS Explorer
 - » Event Specification
 - Event name + Data Fields to be sent
 - » Capture Specification
 - Capture exit to be used
 - Data to be collected at Capture exit
 - » Event Binding
 - List of related events
 - Where events are to be sent

Sample CICS Event Definition

- We will use CICS Explorer to create a simple Event Capture for an Order Entry system.
- Every time a Detail Part Number stock record is updated, we want to check if the quantity on hand has dropped down to a minimum limit. If it does, we want to fire an event, indicating that has happened, so that a “Re-Order” process can be informed to re-order new stock to replenish the inventory.
- The Parts Detail records are stored in a VSAM file.

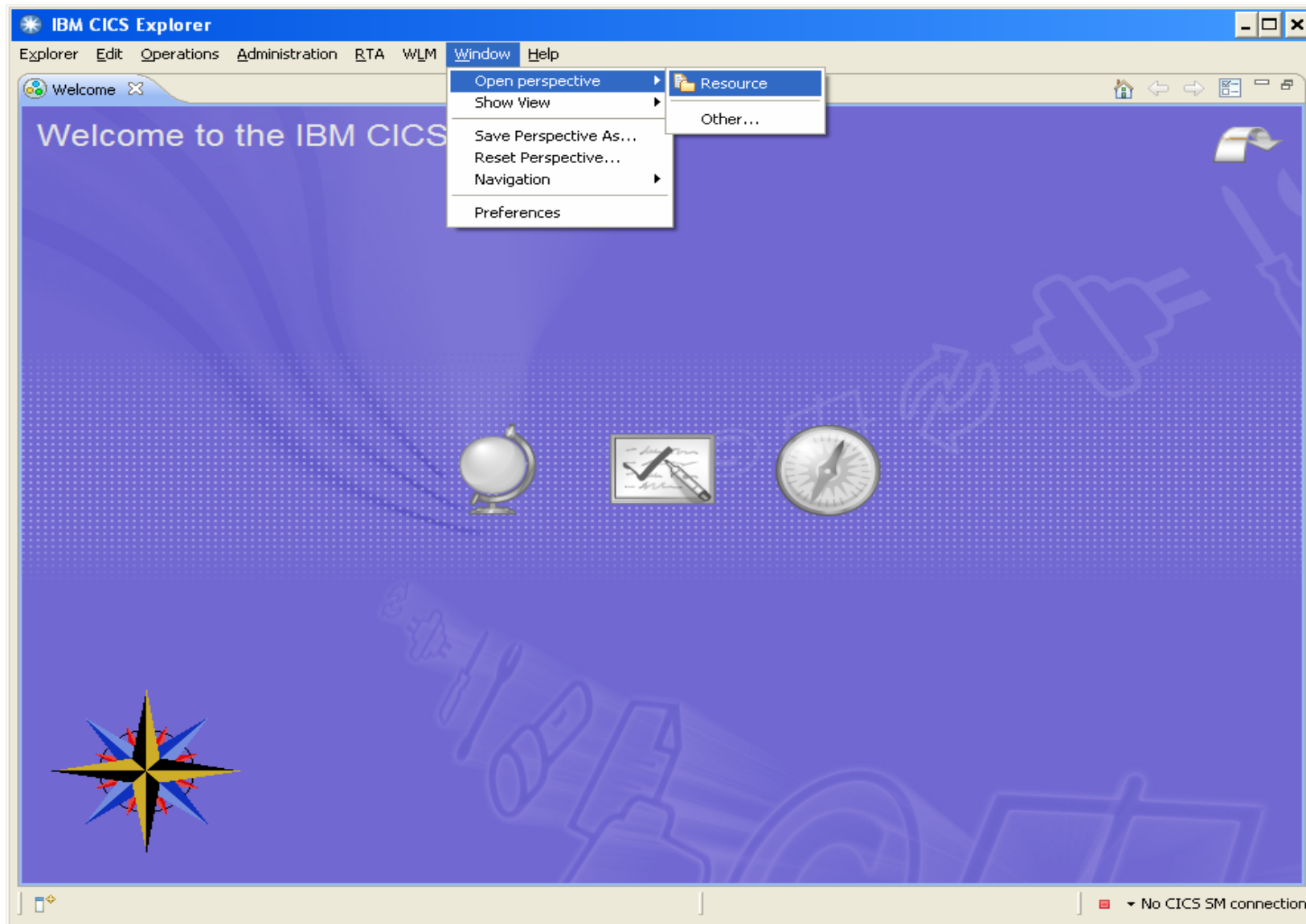
Starting CICS Explorer

- Getting CICS Explorer – visit the “CICS Explorer Downloads” page at <http://www-01.ibm.com/software/htp/cics/explorer/download/>
 - » Unzip the file into a directory on your PC
 - » Double click on the cicsexplorer.exe file in that directory to start it up.



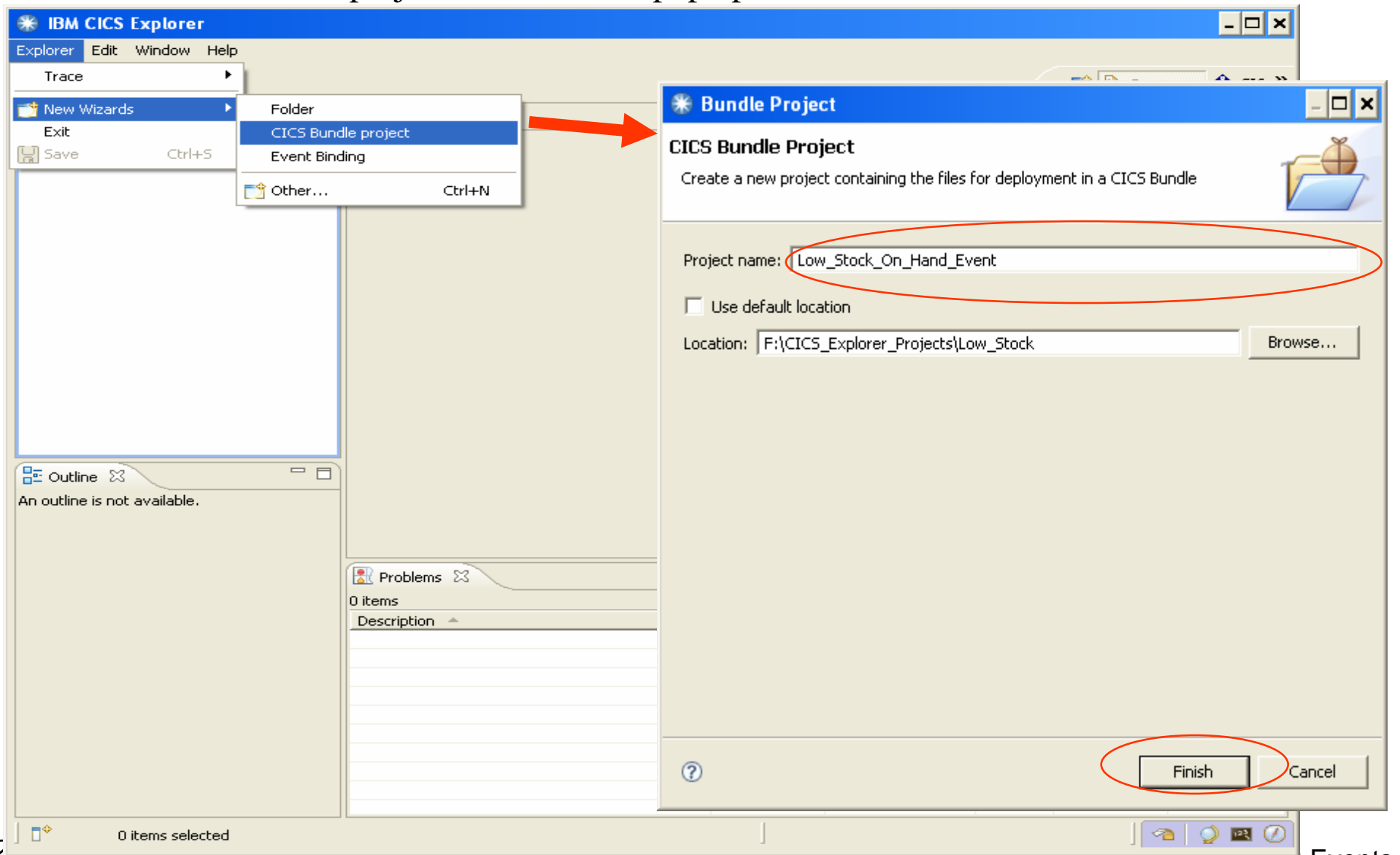
Event Definition Using CICS Explorer

Go to top menu and select: Window -> Open Perspective -> Resource



Event Definition - 2

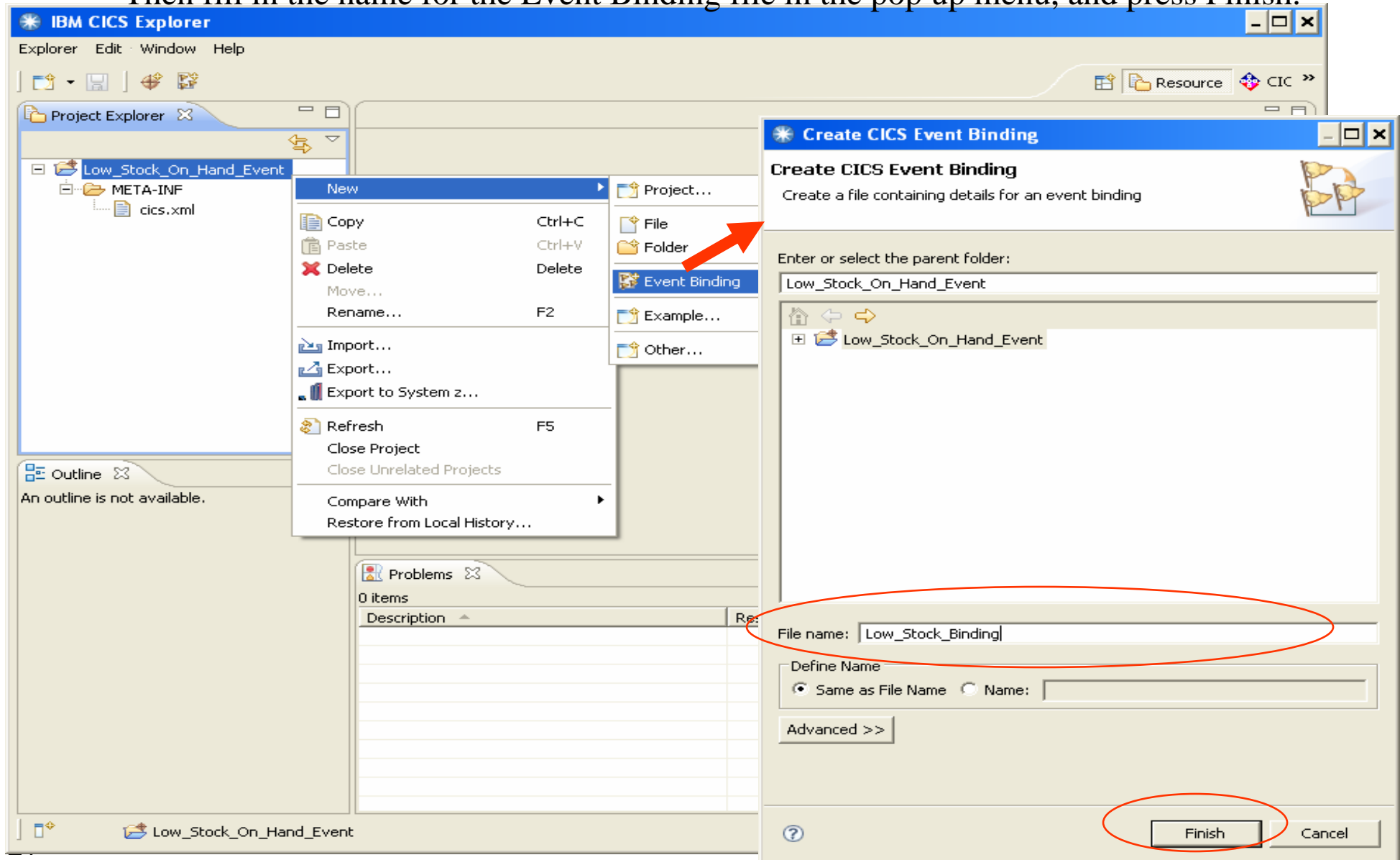
To start a new Event project, select: Explorer -> New Wizards -> CICS Bundle project. Then fill in the Event project's name in the pop up menu, and click Finish.



Event Definition - 3

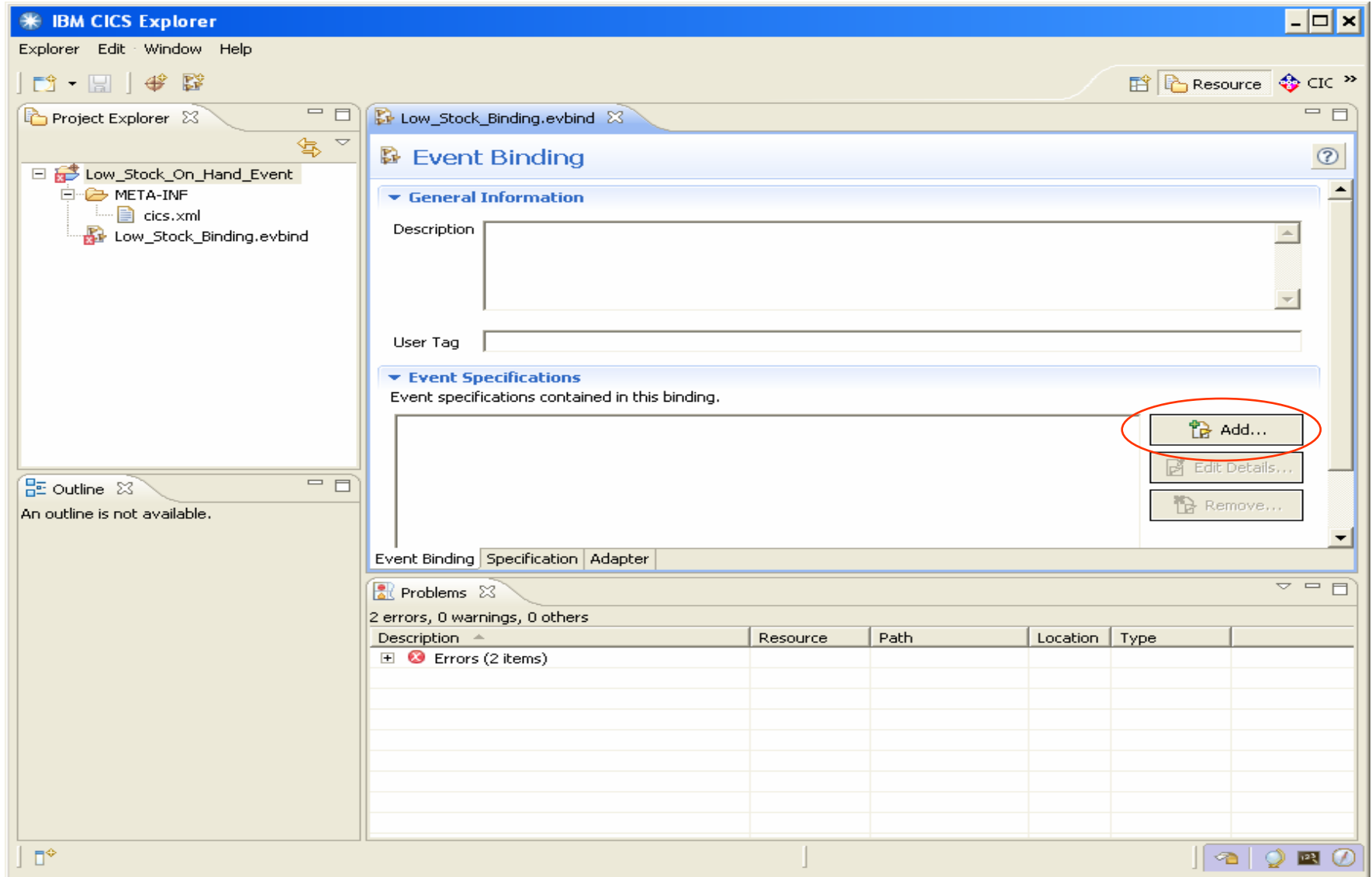
Expand the result project, then right click, and select New -> Event Binding.

Then fill in the name for the Event Binding file in the pop up menu, and press Finish.



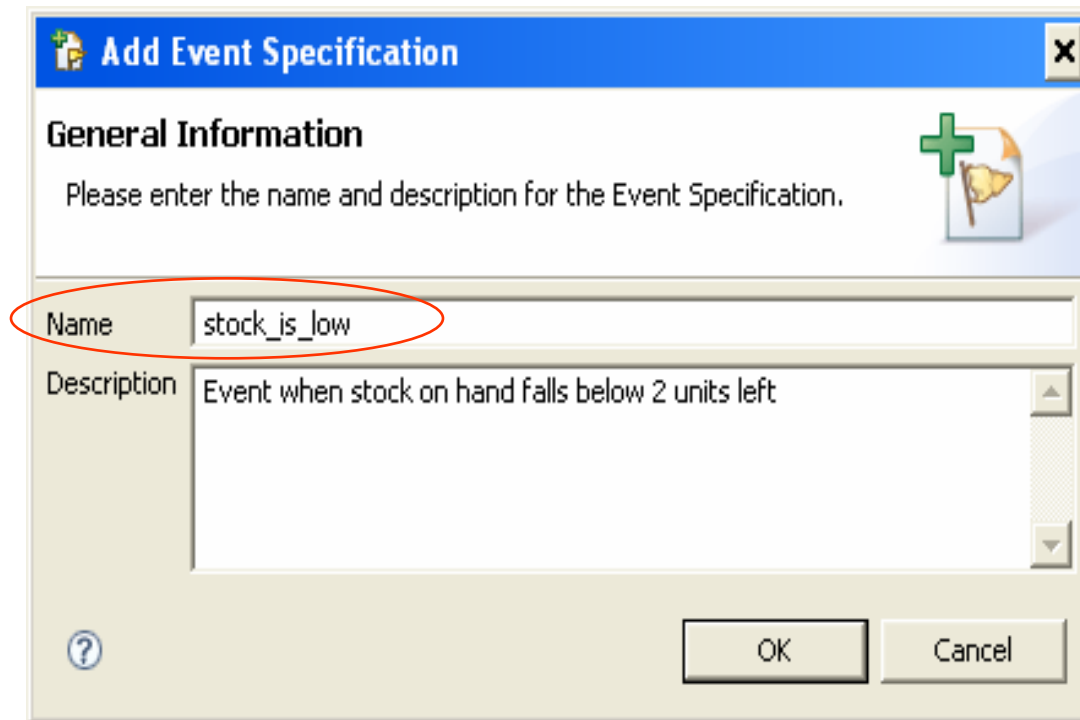
Create Event Specification

An Event Binding edit window is then presented.
Press the Add key, in order to create an event definition.



Create Event Specification - 2

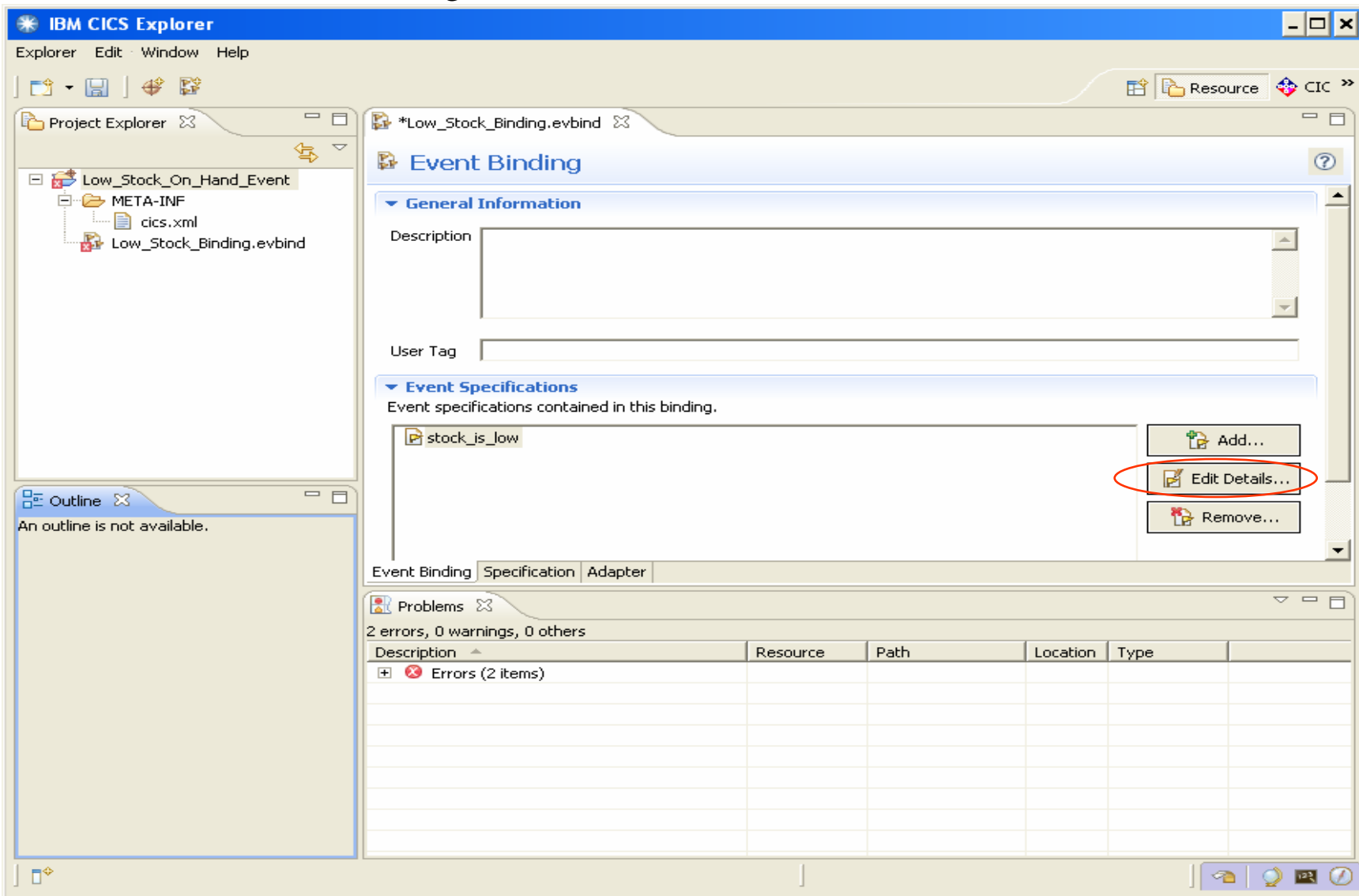
This specifies the Event Name that will be sent to the Event Consumer, So that it can identify what kind of event it is.



The screenshot shows a dialog box titled "Add Event Specification". It has a blue header bar with a close button (X) on the right. Below the header is a section titled "General Information" with a plus sign icon and a document icon. The text below this section says "Please enter the name and description for the Event Specification." There are two input fields: "Name" and "Description". The "Name" field contains the text "stock_is_low" and is circled in red. The "Description" field contains the text "Event when stock on hand falls below 2 units left". At the bottom of the dialog, there are "OK" and "Cancel" buttons, and a help icon (question mark) on the left.

Event Specification Details - 3

Once the event is specified, then press the Edit Details button, to edit the other details about the event (e.g. the associated data to be sent/emitted with the event).



Event Specification Details - 4

Press the Add key on the resulting Edit details screen and then fill in the information about each data item to be sent on the event

The screenshot shows the IBM CICS Explorer interface. The main window displays the 'Specifications' dialog for the event 'stock_is_low'. The 'General' section shows the event name and description. The 'Emitted Business Information' section is currently empty. A red circle highlights the 'Add...' button in the 'Emitted Business Information' section. A red arrow points from this button to a smaller dialog box titled 'Add Emitted Business Information'. This dialog box contains the following fields:

- Name: part_num
- Type: Numeric
- Length: 7
- Precision: 0
- Description: Part Number that is low on stock

The 'OK' button in the 'Add Emitted Business Information' dialog is also circled in red. The background shows the Project Explorer with the event specification tree and the Problems window showing 2 errors.

Event Specification Details - 5

For each piece of data to be sent on the event, press the Add key to add the item to the event specification. In this case, we will only send two items.

The image shows the Eclipse IDE interface. On the left, the 'Add Emitted Business Information' dialog is open. The 'Name' field contains 'restock_qty', the 'Type' is 'Numeric', and the 'Length' is '5'. The 'Description' field contains 'Amount to request on the re-stocking order'. The 'OK' button is highlighted with a red circle. An orange arrow points from this dialog to the 'Specifications' view on the right.

The 'Specifications' view shows the 'stock_is_low' event. Under the 'Emitted Business Information' section, a table lists the data items to be sent:

Name	Type	Length	Precision	Description
part_num	Numeric	7	0	Part Num
restock_qty	Numeric	5	0	Amount to

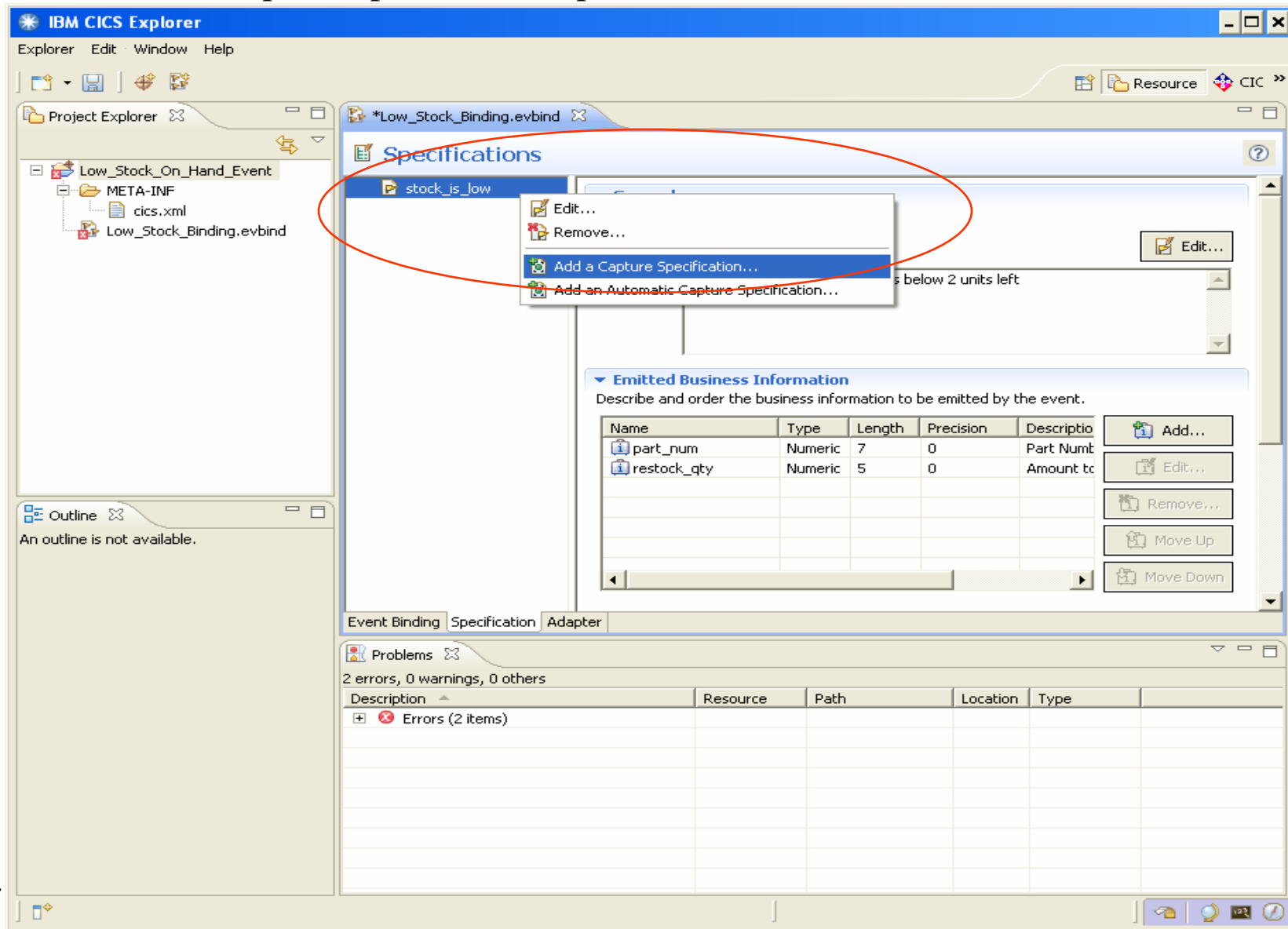
The 'restock_qty' row is circled in red. Below the table are buttons for 'Add...', 'Edit...', 'Remove...', 'Move Up', and 'Move Down'. A red arrow points from the dialog to this table.

Final Result
These fields will be sent as the Event Data to the consumer

At the bottom of the IDE, the 'Problems' view shows '2 errors, 0 warnings, 0 others'.

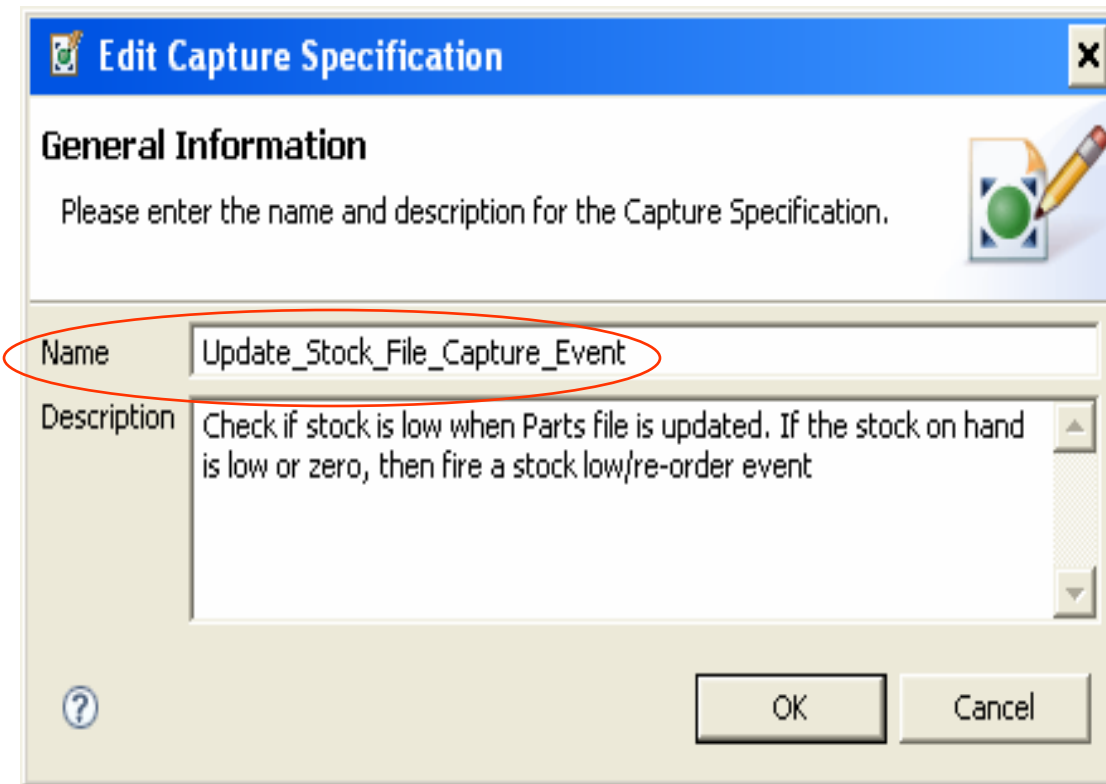
Create Capture Specification - 1

Select the event specification (stock_is_low), and right click, then select the Add a Capture Specification option.



Capture Specification Details - 2

Fill in the name of the capture spec, and its description.



Edit Capture Specification

General Information

Please enter the name and description for the Capture Specification.

Name: Update_Stock_File_Capture_Event

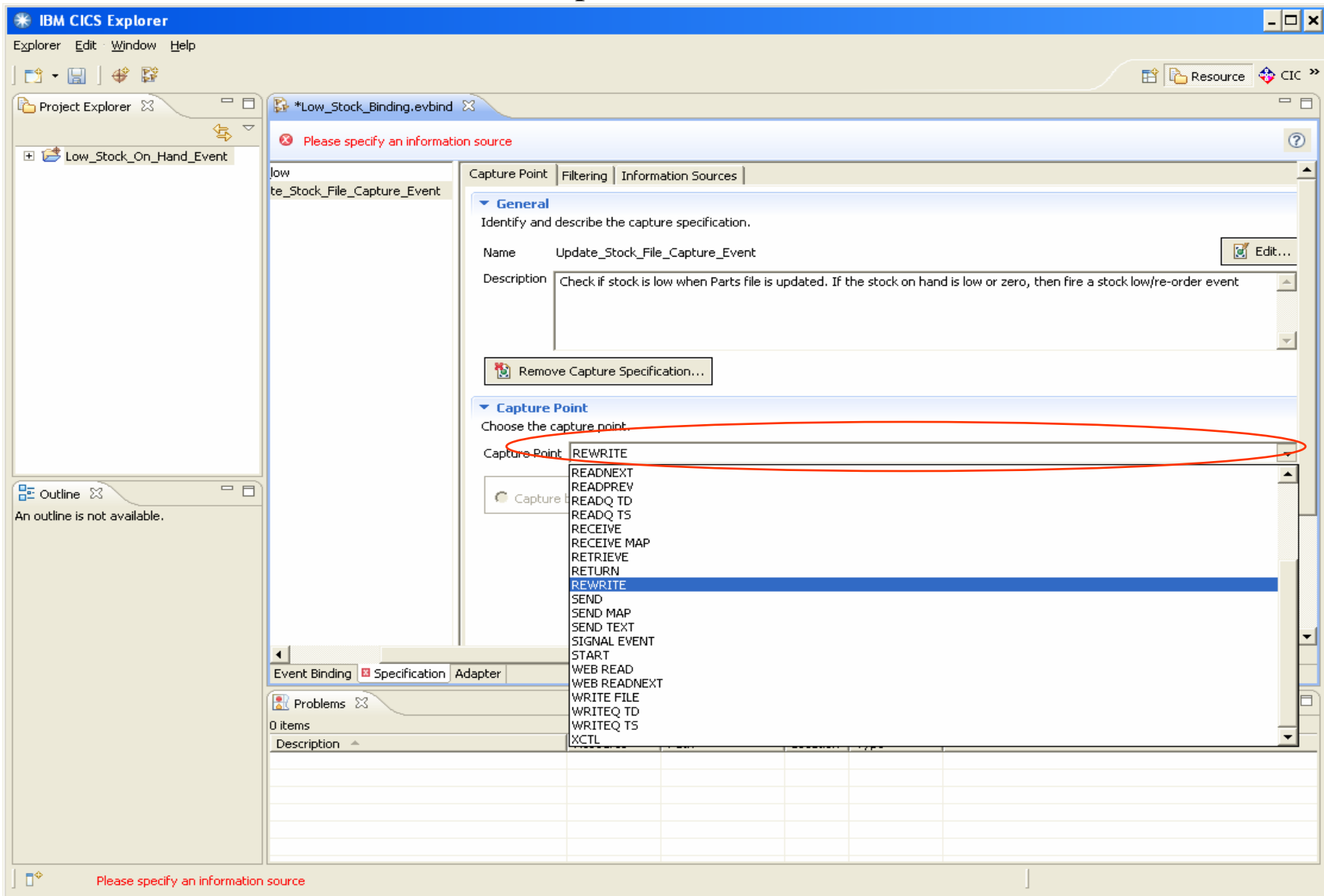
Description: Check if stock is low when Parts file is updated. If the stock on hand is low or zero, then fire a stock low/re-order event

OK Cancel

Normally you only have 1 capture specification per Event.

Capture Specification Details - 3

Specify the capture point. In this example, we want to check the current stock on hand value, after we update a PARTS record from a VSAM file



CICS Capture Points

- Are invoked by the CICS runtime, when a specific CICS API call is performed.
- EXEC CICS APIs that support capture points are:
 - » Channel: PUT CONTAINER, START tranid
 - » File: READ, READNEXT, READPREV, WRITE, REWRITE, DELETE
 - » Temp Stg: READQ TS, WRITEQ TS, DELETEQ TS
 - » Transient Data: READQ TD, WRITEQ TD, DELETEQ TD
 - » Terminal Ctl: SEND, RECEIVE, CONVERSE
 - » Web: WEB READ, WEB READNEXT, INVOKE SERVICE
 - » BMS: RECEIVE MAP, SEND MAP, SEND TEXT
 - » Program Ctl: LINK, RETURN, XCTL, START
 - » New APIs: SIGNAL EVENT
- Can also do a event capture when a program initially starts up.

Capture Specification Details - 4

Press the Filtering tab, to setup when the Capture exit should be invoked. We want the Capture exit to be invoked only when we REWRITE the file named PARTS.

The screenshot shows the IBM CICS Explorer interface with the following details:

- Project Explorer:** Shows a project named 'Low_Stock_On_Hand_Event'.
- Event Binding:** Shows a capture point named 'Stock_File_Capture_Event'.
- Filtering Tab:** The 'Filtering' tab is selected. It contains three sections:
 - Application Context:** Fields for Transaction ID, Current Program, User ID, and Response Code, all set to 'All'.
 - Application Command Options:** A field for 'Name' is set to 'FILE*', the 'Operator' is 'Equals', and the 'Value' is 'PARTS'. These three elements are circled in red.
 - Application Data:** A table with columns: Source, Container, Offset, Length, Operator, Value. It is currently empty.
- Problems:** Shows 0 items.

Capture Specification Details - 5

This specifies the “filtering” to be performed. Can filter based on “Context Data” (user-id, program name, tran-id), and/or can filter by a comparing against a data value.

The screenshot displays the IBM CICS Explorer interface for configuring a capture specification. The main window shows the 'Application Data' section with an 'Add...' button circled in red. A dialog box titled 'Application Data Predicate' is open, showing the 'Define Predicate' section with 'Operator' set to 'Equals' and 'Value' highlighted in yellow. The 'Variable location and format' section has 'Location' set to 'FROM' and 'Select from imported language structure...' circled in red. A red arrow points from the 'Add...' button in the main window to the 'Define Predicate' section in the dialog box.

Application Data Predicate

Add New Application Data

Value: Value is shorter than expected

Predicate

Define Predicate

Operator: Equals

Value: [Yellow Highlighted Field]

Variable location and format

Location: FROM

Container: [Empty]

Enter the format or choose an item from an imported language structure.

Select from imported language structure...

Type: Character

Offset: 0

Length: 1

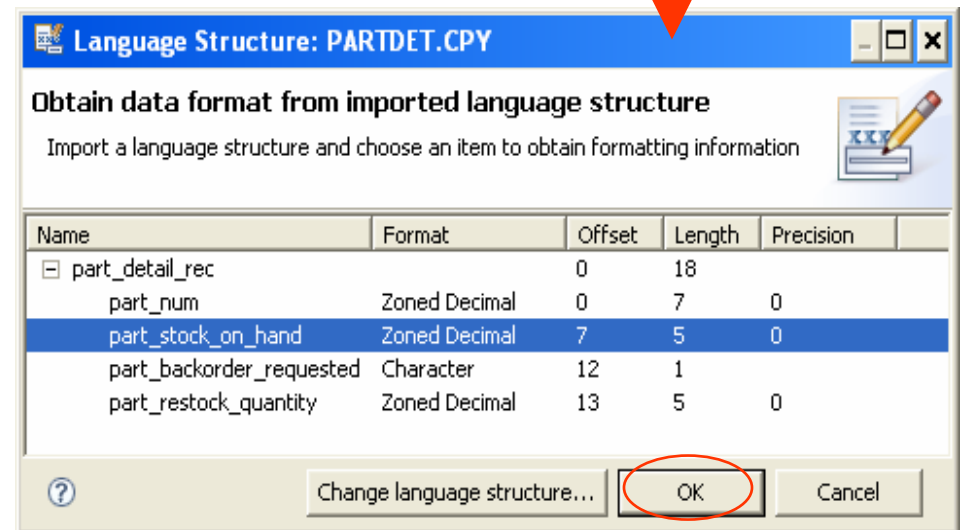
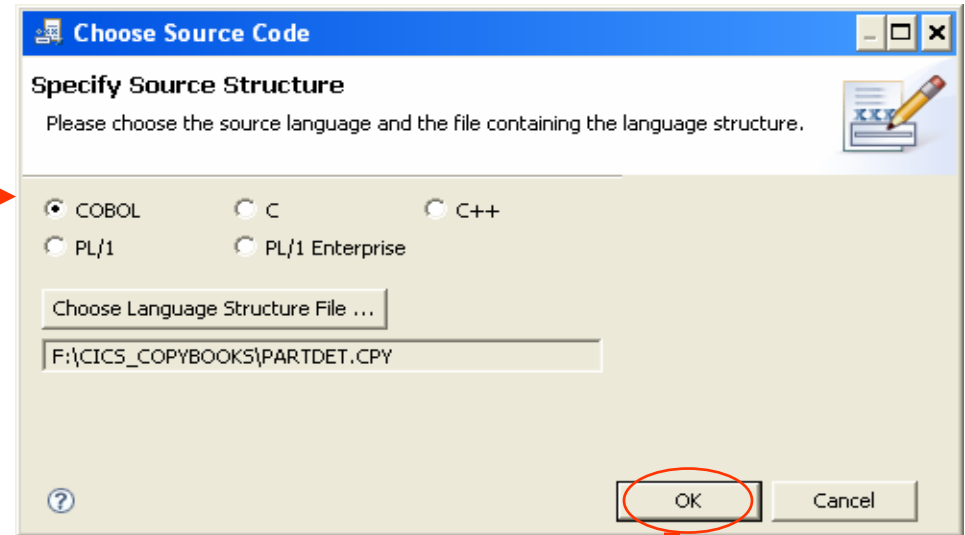
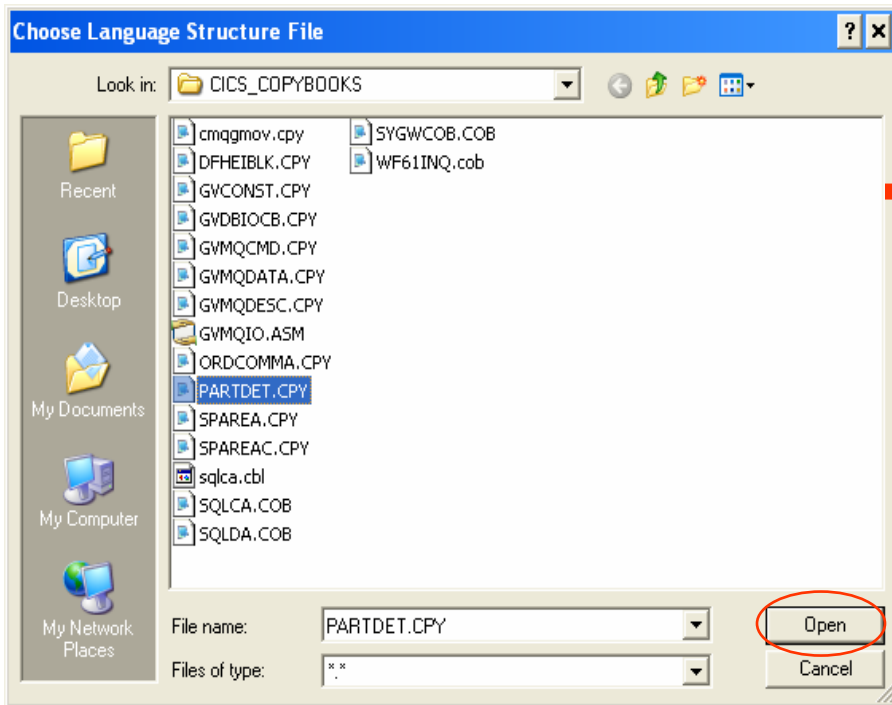
Precision: 0

Codepage: Default (LOCALCCSID)

OK Cancel

Capture Specification Details - 6

If you want to use data fields from a Record or CommArea, you need to have a copy of it on one of your PC's directories. The tool will automatically parse it and provide a nice GUI.



Actual Copy Book on disk

* PARTS DETAIL RECORD COPY BOOK *

01 PART-DETAIL-REC.

05 PART-NUM PIC 9(7).

05 PART-STOCK-ON-HAND PIC 9(5).

05 PART-BACKORDER-REQUESTED PIC X.

33 05 PART-RESTOCK-QUANTITY PIC 9(5).

Capture Specification Details - 7

Then fill in the “Predicate” information indicating what kind of data comparison you want to make and what data value it should be compared against.

Application Data Predicate

Add New Application Data

Define the predicate, choose the location where the data is found then enter the format.

Predicate

Define Predicate

Operator Not Greater Than

Value 1

Variable location and format

Location FROM

Container

Enter the format or choose an item from an imported language structure.

Select from imported language structure...

Type Zoned Decimal

Offset 7

Length 5

Precision 0

Codepage Default (LOCALCCSID)

OK Cancel

Specify an information source

Capture Point: Filtering Information Sources

Application Context

Define predicates to filter events.

Context Operator Value

Transaction ID All

Current Program All

User ID All

Response Code All Ok

Application Command Options

Define predicates for command options. Predicates marked with * should be specified to maintain CICS performance.

Name Operator Value

FILE* Equals PARTS

Application Data

Define predicates for application data. Import a language structure and pick an item to specify the data format.

Source	Container	Offset	Length	Operator	Value
FROM		7	5	Not Great...	1

Final Result

Capture Specification Details - 8

In the “Information Source” tab, we assign specific data fields (from CommArea, Channel or File Record, to the Event Data Variables that were defined earlier in the “Event Spec” portion.

The screenshot displays the IBM CICS Explorer interface. The main window shows the 'Information Sources' tab for a capture specification. The table below lists the defined information sources:

Name	Type	Format	Length	Source	Container	Offset	Capture Length	Capture Type
part_num	Numeric	7				0	0	
restock_qty	Numeric	5						

The 'part_num' row is selected, and the 'Edit...' button is visible. A dialog box titled 'Information Source for part_num' is open, showing the 'Edit Information Source' configuration. The 'Type' field is set to 'Invalid choice'. The 'Available Data' list shows the 'FROM' option selected under 'Application Data'. The 'Select from imported language structure...' button is also visible.

Capture Specification Details - 9

Name	Format	Offset	Length	Precision
part_detail_rec		0	18	
part_num	Zoned Decimal	0	7	0
part_stock_on_hand	Zoned Decimal	7	5	0
part_backorder_requested	Character	12	1	0
part_restock_quantity	Zoned Decimal	13	5	0

These fields will be extracted at run-time and moved into the “Event Data” variables that will be sent on the Event.

Name	Format	Offset	Length	Precision
part_detail_rec		0	18	
part_num	Zoned Decimal	0	7	0
part_stock_on_hand	Zoned Decimal	7	5	0
part_backorder_reque	Character	12	1	0
part_restock_quantity	Zoned Decimal	13	5	0

Capture Specification Details - 10

This screen shows the final result of the Capture Spec sequence.

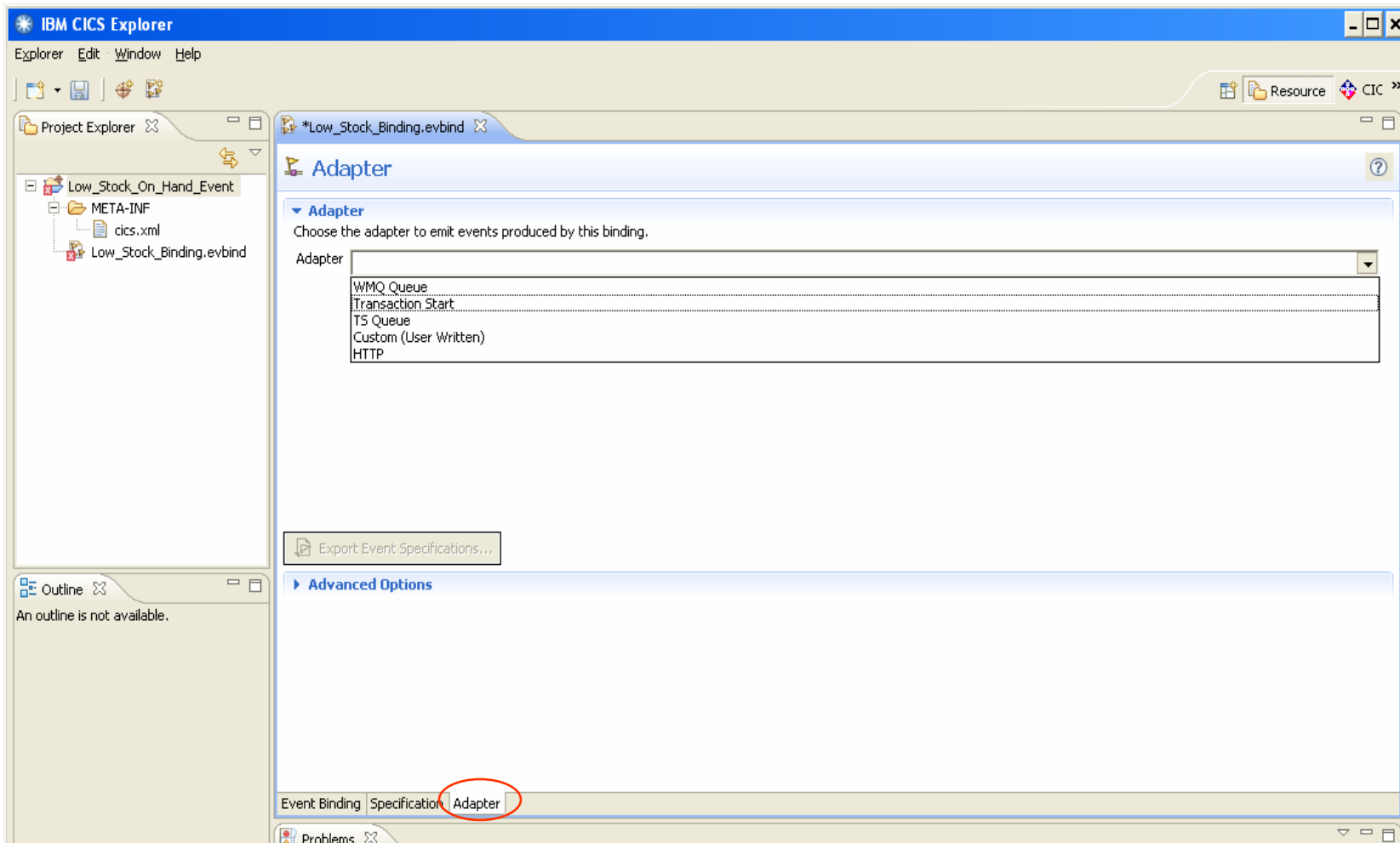
The screenshot shows the IBM CICS Explorer interface. The main window is titled 'Specifications' and is displaying the 'Information Sources' tab. The 'Information Sources' section is defined as 'Define where emitted business information is obtained by this capture specification'. Below this, there is a table with the following data:

Name	Type	Format Length	Source	Container	Offset	Capture Length	Capture Type
part_num	Numeric	7	FROM		0	7	Zoned Decimal
restock_qty	Numeric	5	FROM		13	5	Zoned Decimal

At the bottom of the 'Specifications' window, there are navigation links: '<- Back: Filtering' and 'Next: Adapter ->'. The 'Problems' window at the bottom shows '0 items'.

Select Adapter to Send Event To

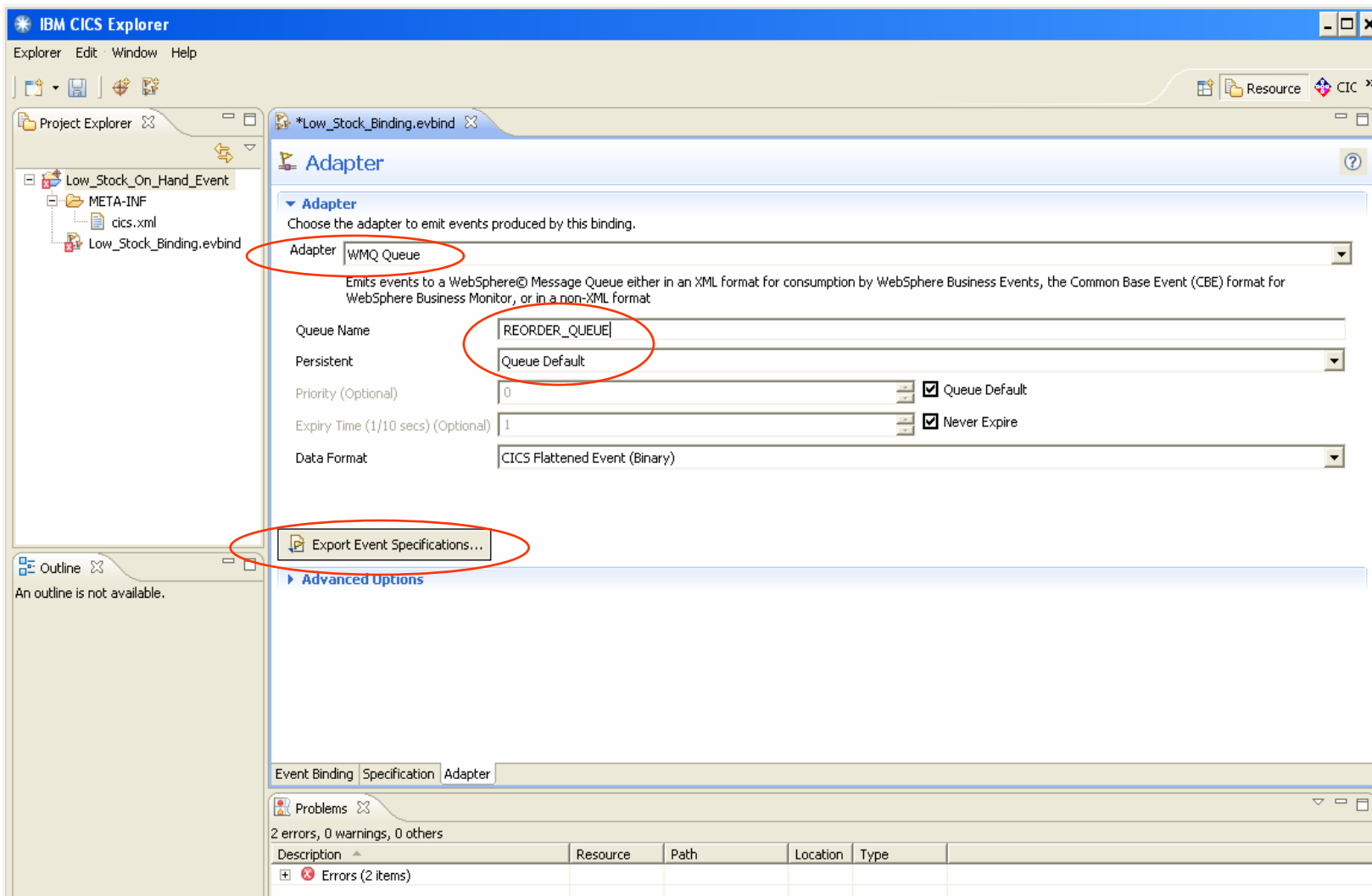
Press the Adapter tab, and then select the EP Adapter you want to use, to send the event to.



CICS Events can be sent to: MQ, WAS BE, or WAS BPM
TS Queue
CICS Start
Custom written EP Adapter

Select Adapter Details

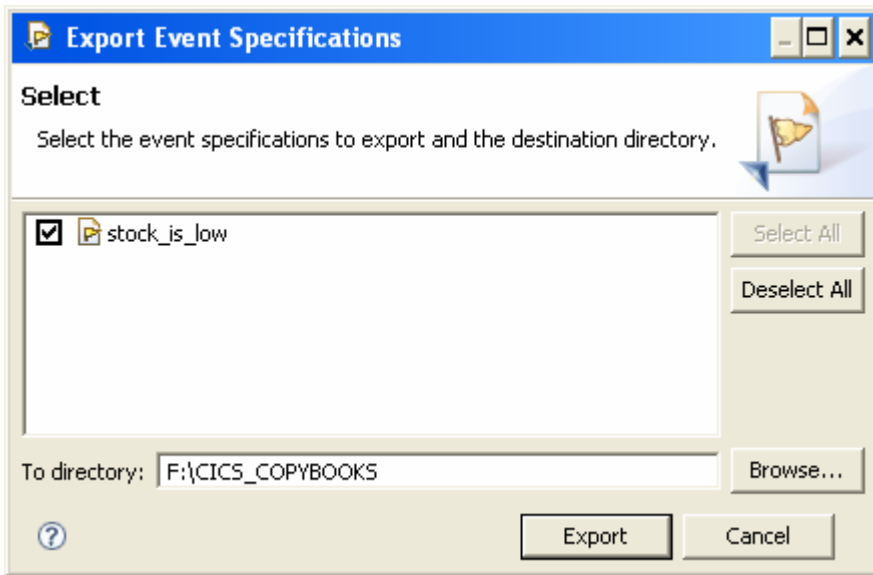
When complete, press the Export Event Specifications to export the COPYBOOK for the MQ App



Different XML formats can be generated for MQ connections, based on “Data Format” box and associated Target MQ (CFE), WAS BE (WBE), or WAS BPM (CBE)

COPYBOOK for MQ Event Consumer App

For MQ type EP Adapters, you can generate a copy book that maps the Event Data structure that will be sent to the Event Consumer.



Generated Copybook for MQ App

* Generated copybook for Event Specification

* 'stock-is-low'

01 stock_is_low.

05 ContextData.

COPY DFHEPFEO.

05 EventData.

10 part_num

PIC +9(6).

10 restock_qty

PIC +9(4).

Sample extract of .evbind file

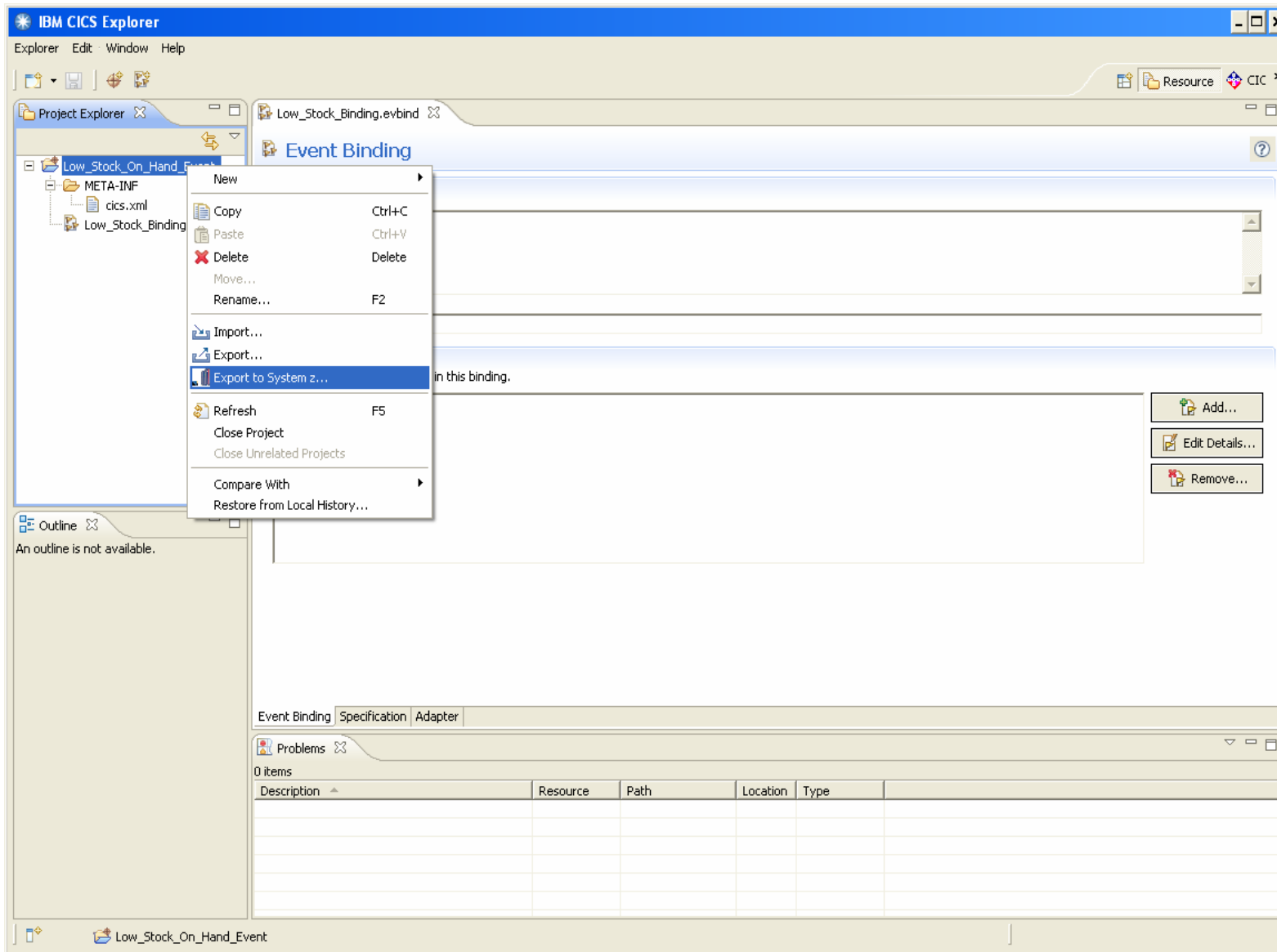
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:eventBinding CICSEPSchemaRelease="0" CICSEPSchemaVersion="1
  " xsi:schemaLocation="http://www.ibm.com/xmlns/prod/cics/eventprocessing/eventbinding
    CicsEventBinding.xsd " xmlns:ns2="http://www.ibm.com/xmlns/prod/cics/eventprocessing/eventbinding">
<eventSpecification>
  <name>stock_is_low</name>          ←Event Name
  <eventInformation>
    <eventInformationItem description="Part Number that is low on stock"      ←Event Data
      length="7" dataPrecision="0" dataType="numeric" name="part_num"/>
    <eventInformationItem description="Amount to request on the re-stocking order"
      length="5" dataPrecision="0" dataType="numeric" name="restock_qty"/>
  </eventInformation>
</eventSpecification>
<eventCaptureSpecification>
  <name>Read_Stock_File_Capture</name>
  <eventIdentifier>stock_is_low</eventIdentifier>
  <description>We check if the stock is low when the Parts file is read</description>
  <filter>
    <locationFilter filterType="CICS_API">
      <readCommand isPre="false" adVerb="" verb="READ">          ←Capture API
        <FILE filterValue="PARTS" filterOperator="EQ" keyword="FILE"/>
        <UPDATE filterOperator="OFF" keyword="UPDATE"/>
      </readCommand> </locationFilter>
    <dataFilter>
      <filterItem>
        <dataFilter filterValue="1" filterOperator="LTE" languageVariableName=""      ←Filter Value
          " dataPrecision="0" dataType="ZONED" length="5" offset="7" container="" source="INTO-SET"/>
      </filterItem>
    </dataFilter> </filter>
</eventCaptureSpecification>
</ns2:eventBinding>
```

Sample extract of .evbind file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:eventBinding CICSEPSchemaRelease="0" CICSEPSchemaVersion="1
  " xsi:schemaLocation="http://www.ibm.com/xmlns/prod/cics/eventprocessing/eventbinding
  CicsEventBinding.xsd " xmlns:ns2="http://www.ibm.com/xmlns/prod/cics/eventprocessing/eventbinding>
<eventSpecification>
  <name>stock_is_low</name>          ←Event Name
  <eventInformation>
    <eventInformationItem description="Part Number that is low on stock"      ←Event Data
      length="7" dataPrecision="0" dataType="numeric" name="part_num"/>
    <eventInformationItem description="Amount to request on the re-stocking order"
      length="5" dataPrecision="0" dataType="numeric" name="restock_qty"/>
  </eventInformation>
</eventSpecification>
<eventCaptureSpecification>
  <name>Update_Stock_File_Capture_Event</name>
  <eventIdentifier>stock_is_low</eventIdentifier>
  <description>We check if the stock is low when the Parts file is read</description>
  <filter>
    <locationFilter filterType="CICS_API">
      <adVerb="" verb="REWRITE">          ←Capture API
      <FILE filterValue="PARTS" filterOperator="EQ" keyword="FILE"/>
      <UPDATE filterOperator="OFF" keyword="UPDATE"/>
    </readCommand> </locationFilter>
    <dataFilter>
      <filterItem>
        <dataFilter filterValue="1" filterOperator="LTE" languageVariableName="" ←Filter Value
          " dataPrecision="0" dataType="ZONED" length="5" offset="7" container="" source="FROM"/>
      </filterItem>
    </dataFilter> </filter>
</eventCaptureSpecification>
```

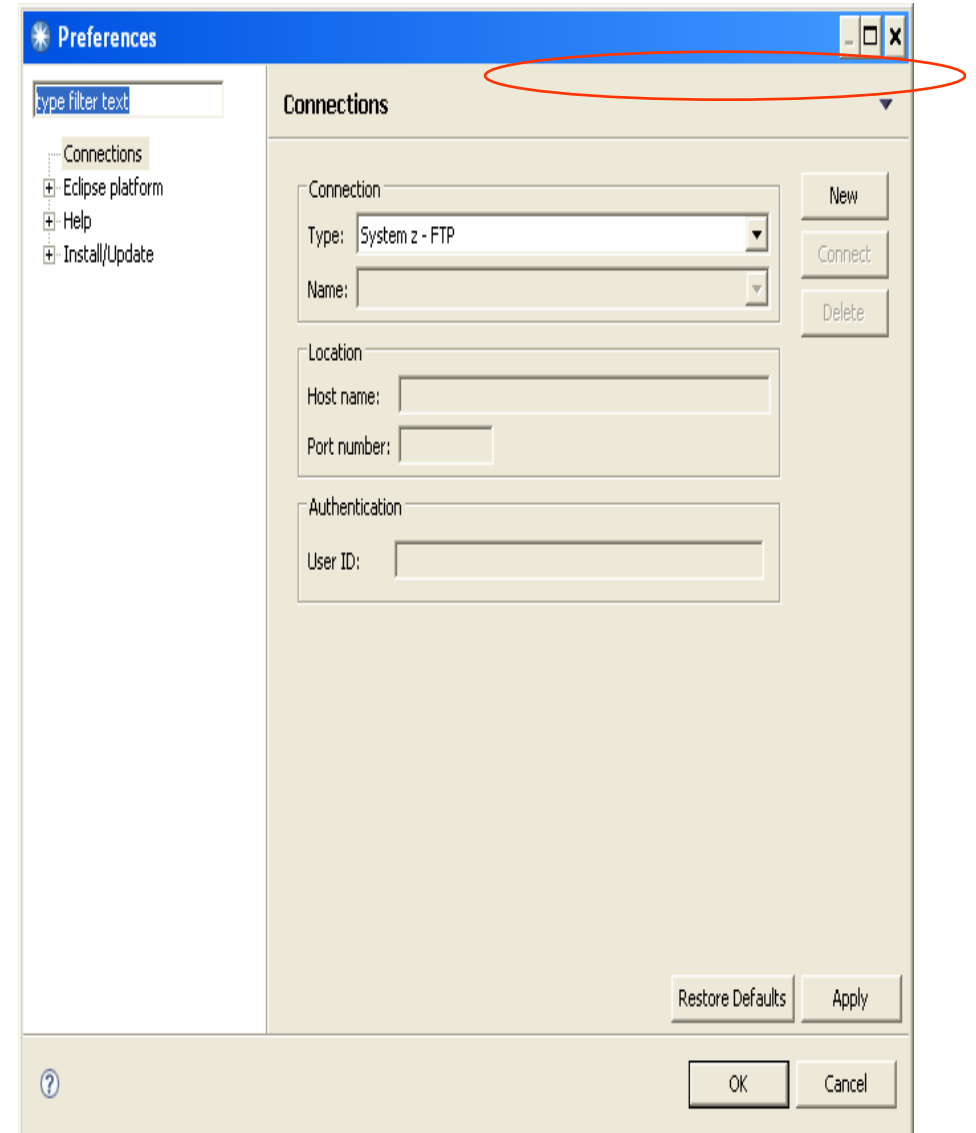
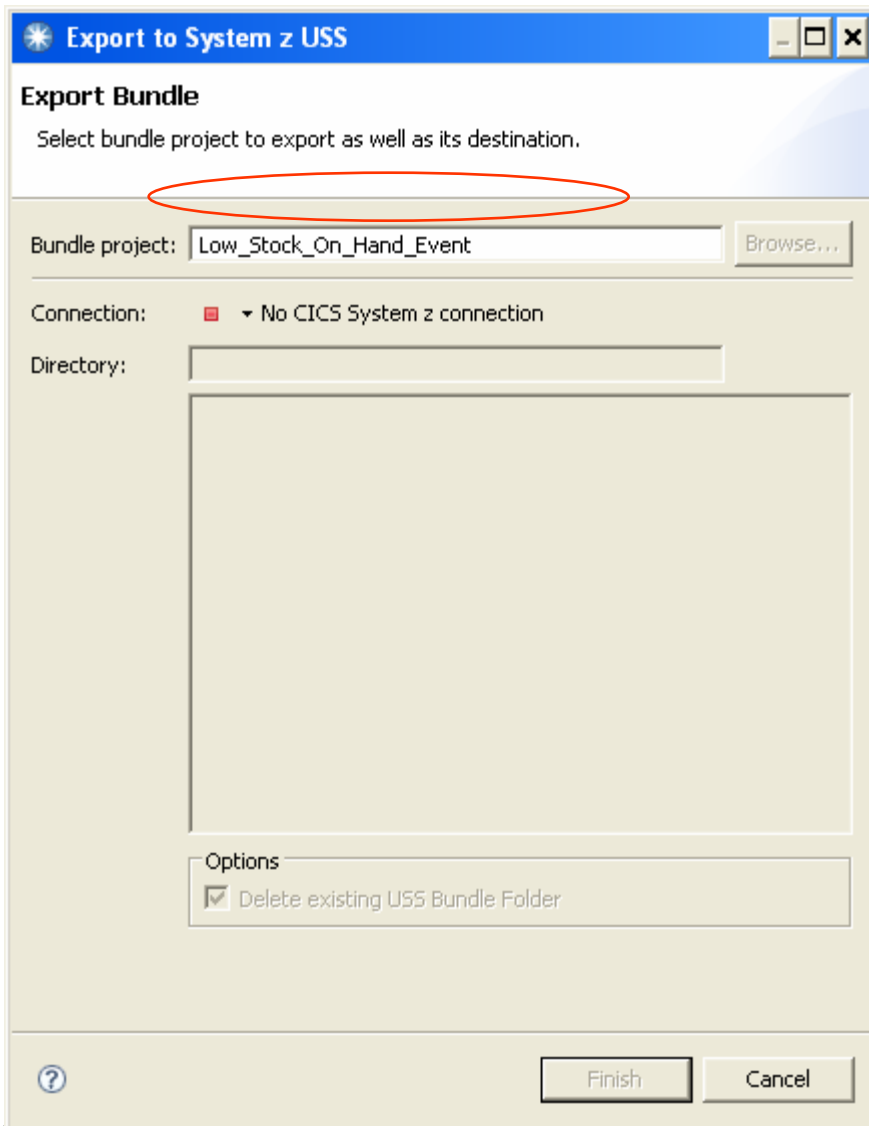
Export Event Binding file to zFS

Once you are done configuring your events, you need to save the result .evbind file onto z/OS, so that CICS can access it at run-time.



Export to z/OS – First Time Only

The very first time you try to export the event, it will request you define the FTP connection parms to your target system running CICS.



Export to z/OS – First Time Only

Second half of FTP connection parms

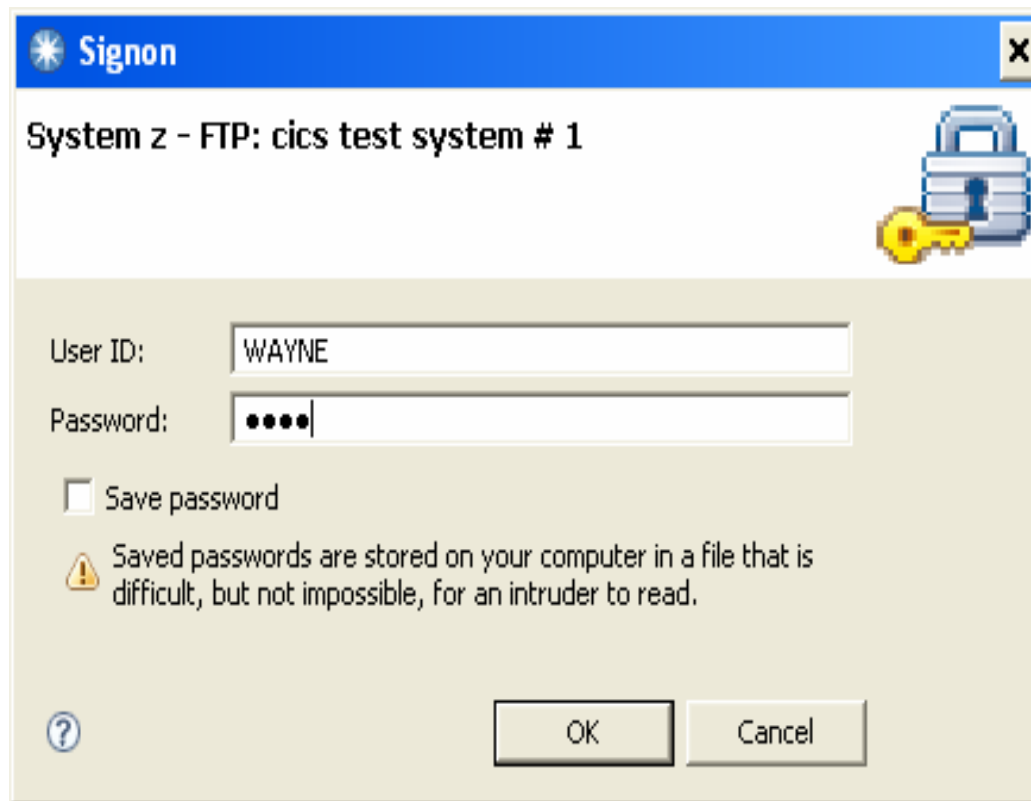
The screenshot shows the Eclipse Preferences dialog box, specifically the Connections section. The dialog has a blue title bar with the text "Preferences" and standard window controls. On the left, there is a tree view with "Connections" selected. The main area is titled "Connections" and contains three sections: "Connection", "Location", and "Authentication".

- Connection:** Type: System z - FTP, Name: cics test system # 1. Buttons: New, Connect, Delete.
- Location:** Host name: testsys1, Port number: 21.
- Authentication:** User ID: DUQUAIN.

At the bottom right, there are buttons for "Restore Defaults", "Apply" (circled in red), "OK", and "Cancel".

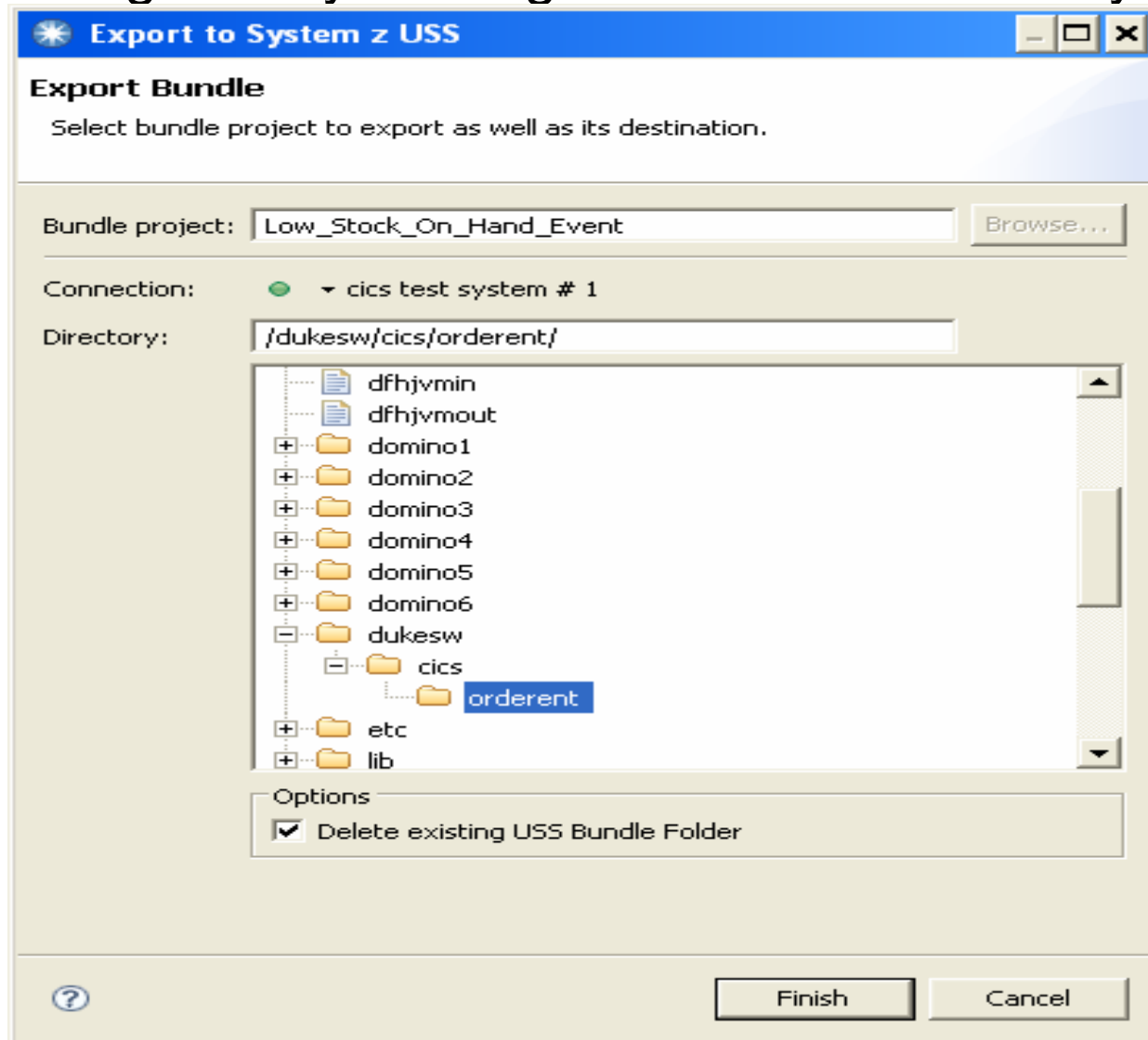
Export to z/OS – Login

Before the FTP transfer can be performed, you must provide the required User-Id and Password needed to access the zFS file system where the Event Spec file will be stored.



Export to z/OS - Upload Event Binding File

- The utility automatically goes into the USS filesystem.
- From there, navigate to your target z/OS USS directory



Upload Event Binding File - Result

```
File Edit Terminal Communication Options Script Help
[Icons]
WAYNE:/dukesw/cics/orderent: >ls -l
total 2
drwxr-x--- 3 TCPIP IPGROUP 352 Jul 30 16:58 Low_Stock_On_Hand_Event
WAYNE:/dukesw/cics/orderent: >ls -l Low_Stock_On_Hand_Event
total 10
-rw-r----- 1 TCPIP IPGROUP 3894 Jul 30 16:58 Low_Stock_Binding.evbind
drwxr-x--- 2 TCPIP IPGROUP 288 Jul 30 16:58 META-INF
WAYNE:/dukesw/cics/orderent: >ls -l Low_Stock_On_Hand_Event/META-INF
total 2
-rw-r----- 1 TCPIP IPGROUP 422 Jul 30 16:58 cics.xml
WAYNE:/dukesw/cics/orderent: >

===> █

RUNNING
ESC=␣ 1=Help 2=SubCmd 3=HlpRetrn 4=Top 5=Bottom 6=TSO
7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
█ → ⇒ 29/007
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
3270 Display 29:7 Caps Hold On Line
```


Install the Bundle under CICS

- There are two ways to install the bundle
 - » Use `CEDA DEFINE BUNDLE` and supply the full USS pathname of the Event spec `.evbind` file
 - » Use the latest CICS Explorer for TS 4.1 that contains GUI menus for the same function.
 - » Bundles are used to group together related definitions, so they can be handled all at once, rather than requiring the CICS Sys Prog to remember all the pieces in his/her head the way they have to today.
- CICS EVENTS do not require any changes to the SIT to enable the run-time logic. However the CICS system operator can control the Event system through the following commands:
 - » `SET EVENTPROCESS STARTED / DRAIN / STOPPED`
 - » `SET EVENTBINDING (name) ENABLED / DISABLED`

Test / Demo App using Web Services

- CommArea based and Channel based Apps are perfect for Web Services
- Can be invoked as JAX RPCs or as JMS or as REST (via PHP)
- The first time you define one it looks complicated
 - » But subsequent ones are just “cookie cutter” variations
 - » You really just need to tweak 8 simple parms to define a Web Service

Web Svcs Request/Reply Copy Books

* Order Entry Request COMMAREA structure

03 ORD-CUSTOMER-ID PIC 9(7).
03 ORD-DEPARTMENT PIC 9(3).
03 ORD-PO-NUM PIC 9(6).
03 ORD-BACKORDER-OK PIC X.
03 ORD-REQ-ITEMS-COUNT PIC 9(2).
03 ORD-REQ-ITEMS ORD-INPUT-DETAIL-DATA
OCCURS 16 TIMES.
05 ORD-PART-NUM PIC 9(6).
05 ORD-PART-QUANTITY PIC 9(5).
05 ORD-PART-DESCRIPTION PIC X(40).
05 ORD-PRICE PIC 9(6)V99.

* Order Entry Reply COMMAREA structure

03 ORD-REPLY-RETURN-CODE PIC 9(2).
03 ORD-REPLY-MESSAGE PIC X(50).
03 ORD-REPLY-CUSTOMER-ID PIC 9(7).
03 ORD-REPLY-DEPARTMENT PIC 9(3).
03 ORD-REPLY-PO-NUM PIC 9(6).
03 ORD-REPLY-ORDER-NUMBER PIC 9(8).
03 ORD-REPLY-ITEMS-COUNT PIC 9(2).
03 ORD-REPLY-ITEMS ORD-REPLY-DETAIL-DATA
OCCURS 16 TIMES.
05 ORD-REPLY-PART-NUM PIC 9(6).
05 ORD-REPLY-PART-QUANTITY PIC 9(5).
05 ORD-REPLY-PART-DESCRIPTION PIC X(40).
05 ORD-REPLY-PRICE PIC 9(6)V99.
03 ORD-REPLY-SHIP-DATE PIC X(8).
05 ORD-REPLY-SHIP-QUANTITY PIC 9(5).
03 ORD-REPLY-BACKORDER-DATE PIC X(8).
05 ORD-REPLY-BACKORDER-QUANTITY PIC 9(5).

Essence of what needs to be defined for Web Svcs

- URI that will be used to access the Web Service
- Name of the program that will be invoked by the Web Service
- Name of Request/Reply Data Areas (aka Copy Books)
- Where to put the output WSDL
- Where to put the output WSDLBIND file
 - » WSDLBIND file is to CICS Web Services that a assembled 3270 BMS map is to CICS 3270 BMS. It provides a binary map of the I/O buffers so that it can be quickly processed.
- If using variable size structures/arrays, turn on DATA-TRUNCATION=ENABLED

DFHLS2WS

Typical JCL to run the generation of a Web Services set of files:

```
//DFHLS2WS PROC JAVADIR='/usr/lpp/java/java6_31/J6.0',  
// PATHPREF="",  
// USSDIR='cicsts41',  
// TMPDIR='/tmp',  
// TMPFILE='LS2WS',  
// SERVICE=""  
//*  
//DOWS2LS      EXEC  DFHLS2WS  
//INPUT.SYSUT1 DD      *  
PDSLIB=//DUKESW.ZOS.SOURCE  
LANG=COBOL  
PGMNAME=ORDERENT  
PGMINT=COMMAREA  
REQMEM=ORDREQ  
RESPMEM=ORDREPLY  
DATA-TRUNCATION=ENABLED  
MAPPING-LEVEL=2.2  
URI=services/Orders  
REQUEST-NAMESPACE=http://www.grandviewsystems.orderent.ORDREQ.com  
RESPONSE-NAMESPACE=http://www.grandviewsystems.orderent.ORDREPLY.com  
WSDL=/dukesw/com/grandviewsystems/orderent/OrderEntryService.wsdl  
WSBIND=/dukesw/com/grandviewsystems/orderent/OrderEntryService.wsbind  
LOGFILE=/dukesw/com/grandviewsystems/orderent/OrderEntryService.log  
/*
```

DFHLS2WS Post Processing

- Upon completion, then use CEDA to define the associated WEBSERVICE and URI entries, e.g.

```
CEDA DEFINE WEBSERVICE
```

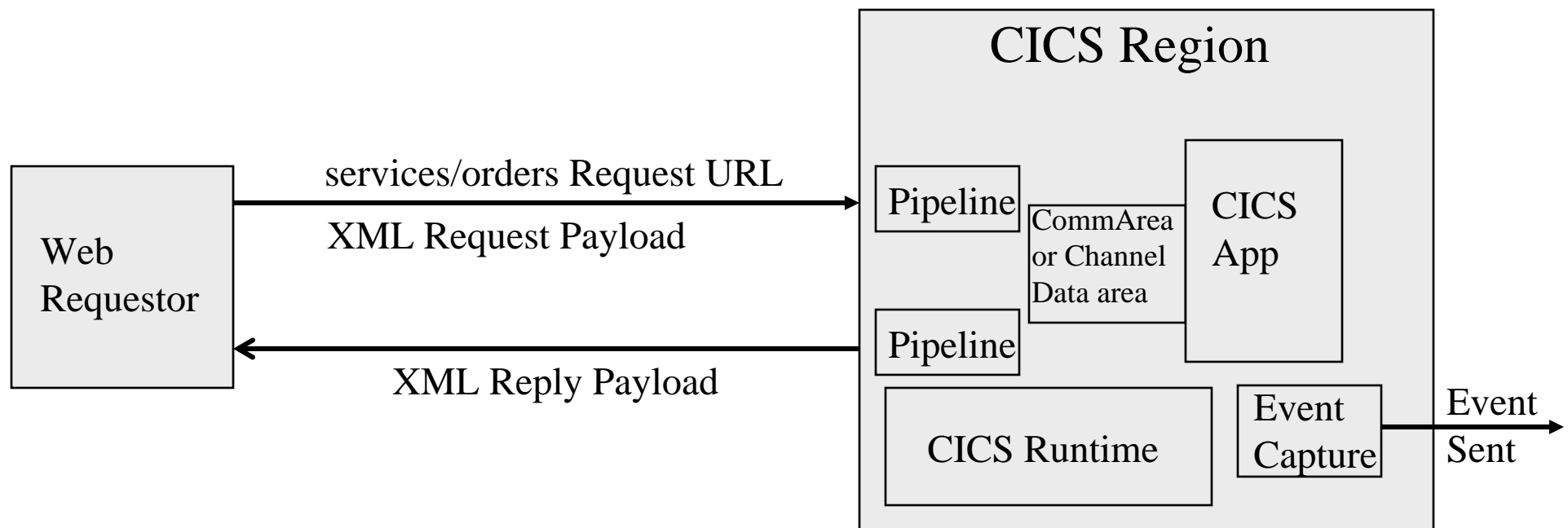
```
WSBIND(/dukesw/com/grandviewsystems/orderent/OrderEntryService.wsdl)
```

```
CEDA DEFINE URIMAP HOST(*) PATH(/services/orders)
```

- Or you can use the “PIPELINE” scanning support in CICS to automatically pick up the WSDL and URI entries, assuming that they are placed in a directory that the PIPELINE support scans.

Web Services invoke of CICS App

- Can be invoked from PC .NET software (VB, VC, C#)
- Can be invoked by PHP and Perl scripts running on Unix / Linux / PCs
- Can be invoked by Databases and App Servers: WebSphere, JBoss, Oracle, ...



Summary

- Long Term Trends
 - » Reduce/eliminate Networking code (SENDS/RECEIVES) in CICS Apps
 - » Move toward pure COMMAREA or Channel driven based Apps
- Defining CICS Events with CICS Explorer is quite easy, once you've been through it the first time.
- Defining a CICS Web Services is fairly easy ("cookie cutter" approach) once you been through it the first time.
- The Network is no longer the computer.
The Service is now the computer.
- Proliferation of network protocols is continuing unabated.
 - » So punt – abstract the network out via XML descriptors, and let CICS run-time do the low level networking calls and buffer mapping

EOJ