



IBM zSeries Software

WebSphere Developer for zSeries (WDz) or Integrating zSeries in an SOA

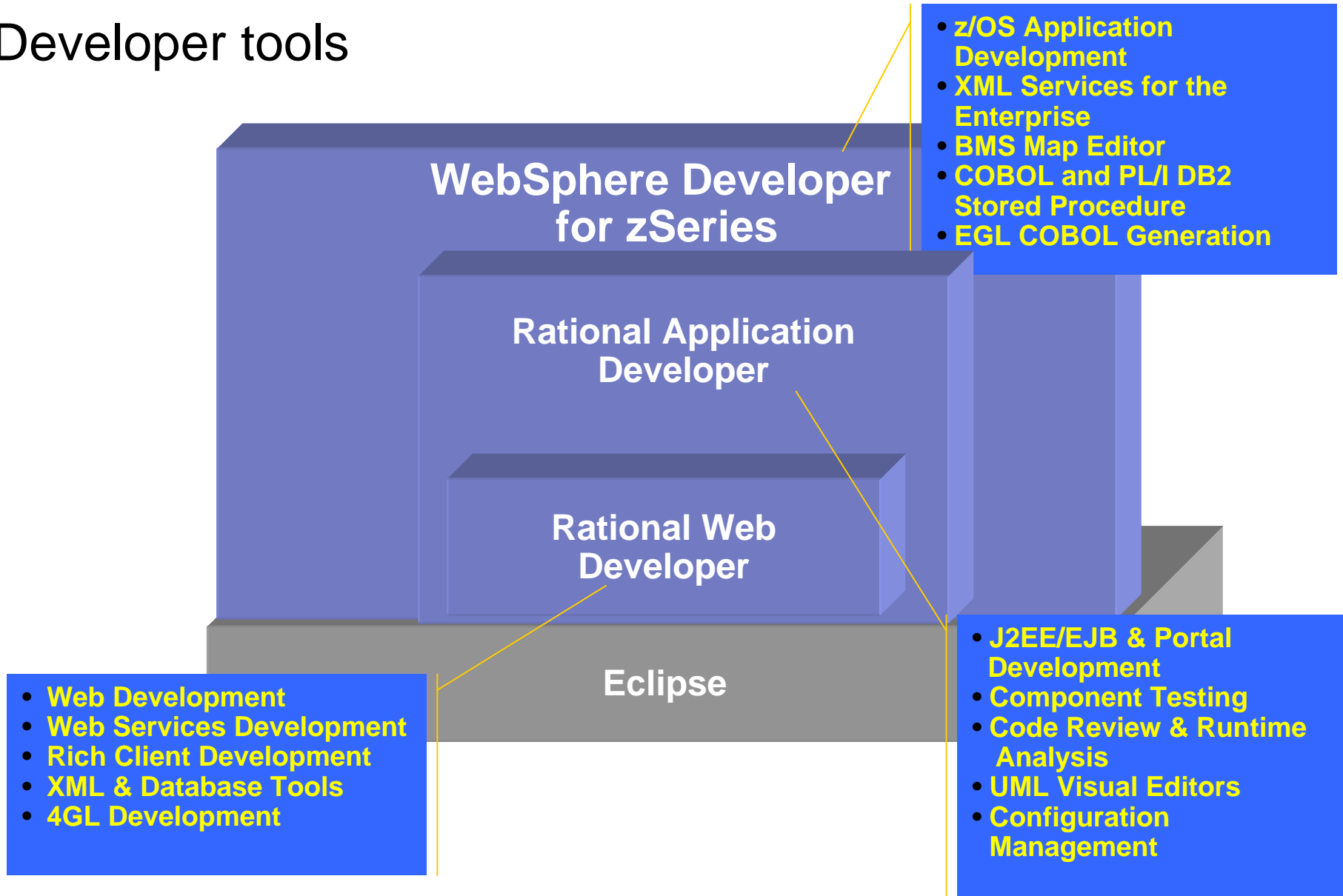


ON DEMAND BUSINESS™

Agenda

- Where do we fit
- **Introduction SOA and WDz**
- **Introduction to tool concepts**
- **Detail information and demonstration of ZOS development**

Developer tools



Decades of existing assets

Rewriting all existing applications and moving them to new platforms is not a viable option

★ New code **cost 5X** than reusing existing code
Software Productivity Research (SPR)

★ **200 Billion lines of COBOL** code in existence
eWeek

★ **5 Billion lines of COBOL code added yearly**
Bill Ulrich, TSG Inc.

★ Between **850K and 1.3 Million COBOL developers**
with **12,000 per year attrition**

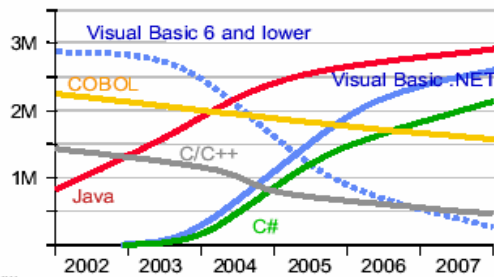
IDC

★ Majority of **customer data still on mainframes**, even though a lot of it is front-ended through the Web and e-Commerce applications

Don Greb, Mellon Financial Corp from Computerworld

Developers

From an estimated worldwide market size of 7 million "professional" developers



M = million

Gartner

What is WZ 6.0.1?

- Eclipse-based integrated development environment for developing enterprise-level, multi-tier applications
- Extends Rational Application Developer (RAD) 6.0.1.1 (or RSA)
 - Inherits capabilities of the base
 - Extends access to zOS systems from the workstation
- EGL to create COBOL/CICS/JFace Multi-tier apps
- Supports a project structure for building zOS applications
 - COBOL, PLI, HLASM
 - TSO/Batch, CICS, IMS, DB2
 - DB2 Stored Procedures – COBOL, PLI, Java, SQL
- Enables COBOL applications for SOA via XML Services for Enterprise
- Implements SOA and Web Services using CICS
 - AND more...

IBM WebSphere Developer for zSeries



XML Services for the Enterprise

- SOA access to CICS V3.1 and IMS V9 COBOL applications
- Bottom-up/Top-down or meet-in-the-middle COBOL to XML mapping support
- Integrated COBOL XML converters, XML schemas, and WSDL generation

DB2 Stored Procedure for COBOL and PL/I

- Create DB2 stored procedures on z/OS in either COBOL or PL/I
- Build and catalog support for the DB2 stored procedure
- Debug z/OS based stored procedures from workstation

EGL COBOL Generation

- Deploy EGL applications to zSeries CICS or batch environments
- Connectivity to CICS through JCA
- JSF UI components integrated with CICS services

z/OS Application Development

- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/1, JCL, etc.
- Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output
- Perform dataset management actions like allocating datasets and migrating datasets
- Perform typical edit, compile, and debug tasks on remote z/OS resources from the workstation

CICS BMS Map Support

- Visually create and modify BMS Map sets
- Generates JCL
- Work with local or remote maps

Service Flow Modeler

- Implements SOA and Web Services
- Service Flow Modeler is a tool to build service flows out of your existing Commarea and Terminal based CICS applications.

WebSphere Developer for zSeries V 6.0.1

All features available in Rational Application Developer PLUS..

1. z/OS Application Development Tools

Interactive, workstation-based development for mainframe COBOL, PL/I, ASM applications

2. XML Services for the Enterprise

SOA access to CICS V2.2, V3.1 and IMS V9 COBOL applications, COBOL to XML mapping support, COBOL XML converters and WSDL generation

3. CICS BMS Map Support

Visually create and modify BMS Map sets
Work with local or remote maps
Generates JCL

4. z/OS-based DB2 Stored Procedure Builder

Create, Build, Test and Debug DB2 stored procedures on z/OS in either COBOL or PL/I

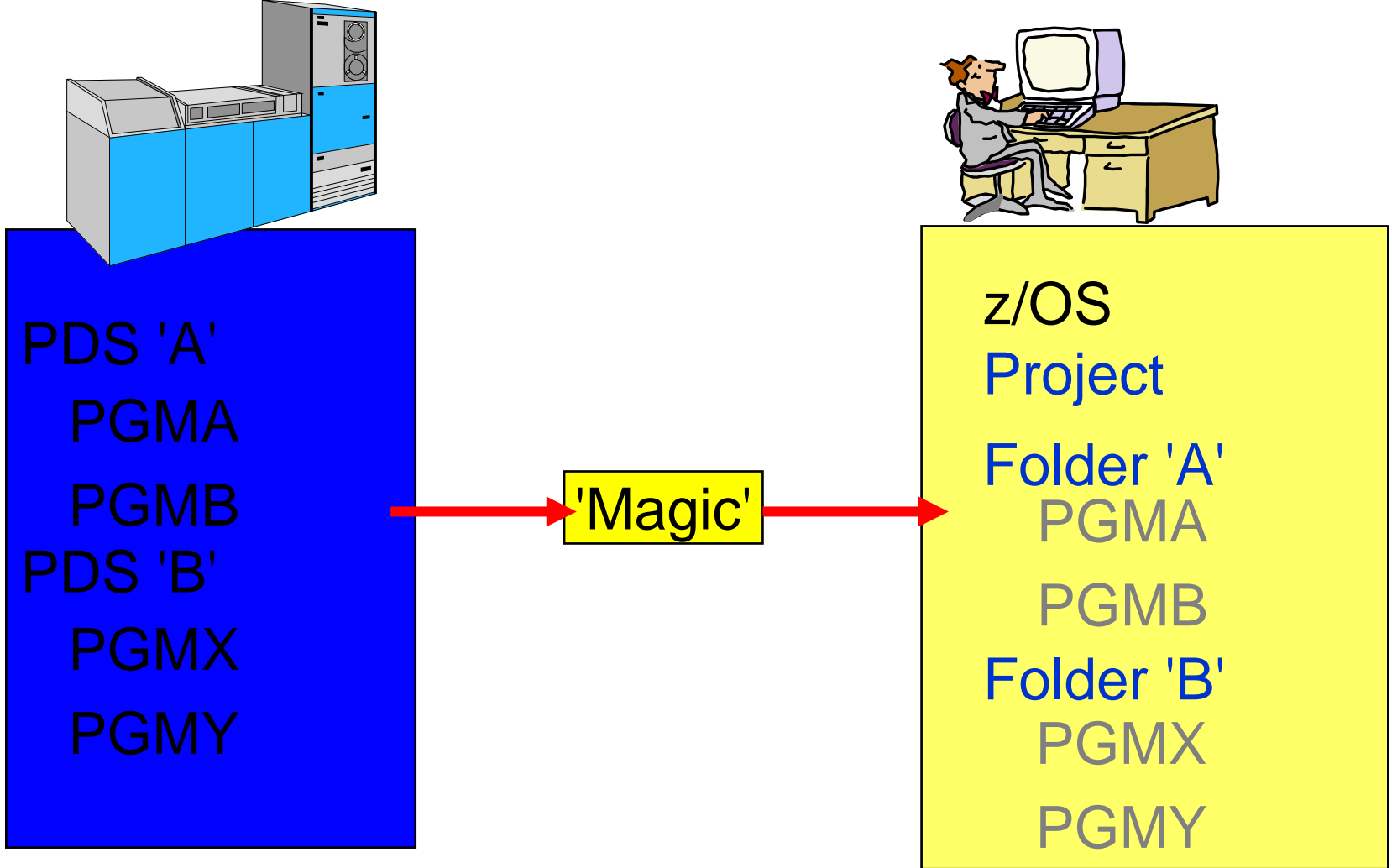
5. Service Flow Modeler

Implements SOA and Web Services for CICS 3.1

6. Enterprise Generation Language (EGL) feature generating COBOL

Simple, high level programming specifications
Create full-function COBOL and J2EE Java applications

Host -> Workstation Overview



Files on the host look as though they are workstation files

Eclipse, WebSphere based Development

The screenshot displays the IBM Rational Software Development Platform interface for z/OS Projects. The main editor shows a COBOL program with the following code:

```
Line 35      Column 1      Insert  
-----*A-1-B-----2-----3-----4-----5-----  
000035      DISPLAI "Program REGIOA STARTING "  
000036      MOVE 2 TO BRANCHFLAG.  
000037      MOVE 'AAAAAA' to FIELD-A.  
000038      MOVE 'BBBBBB' to FIELD-B.  
000039      MOVE 'CCCCCC' to FIELD-C.
```

Annotations and workflow:

- Syntax Check:** A red arrow points from the 'Syntax Check' box to the 'Properties' view, which shows the COBOL structure with '010-INITIALIZATION.' highlighted.
- Edit source:** A red arrow points from the 'Edit source' box to the main editor window.
- Statement in error:** A red arrow points from the 'Statement in error' box to the error message in the 'Tasks' view: 'IGYPS2072-S "DISPLAI" was invalid. Skipped to the next verb, period or procedure-n'.
- double click on the error:** A red arrow points from the 'double click on the error' box to the error message in the 'Tasks' view.
- Error list in Tasks view:** A red arrow points from the 'Error list in Tasks view' box to the 'Tasks' view, which shows '1 error, 0 warnings, 0 infos (Filter matched 1 of 119 items)'.

Benefit: Simplified development for COBOL and PL/I on a common development environment

Content Assist for COBOL and PL/1

Row 37 Column 20 2 changes.

```
000032 PROCEDURE DIVISION.  
000033 010-INITIALIZATION.  
000034 * Initialize Program-work-fields, Program-flags,  
000035 DISPLAY "Program REGIOA STARTING "  
000036 MOVE 2 TO BRANCHFLAG.  
000037 move
```

010 BRANCHFLAG
010 FIELD-A
010 FIELD-B
010 FIELD-C
010 Input-name
010 Loop-flag
010 Output-name
010 Program-flags

Row 37 Column 31 3 changes.

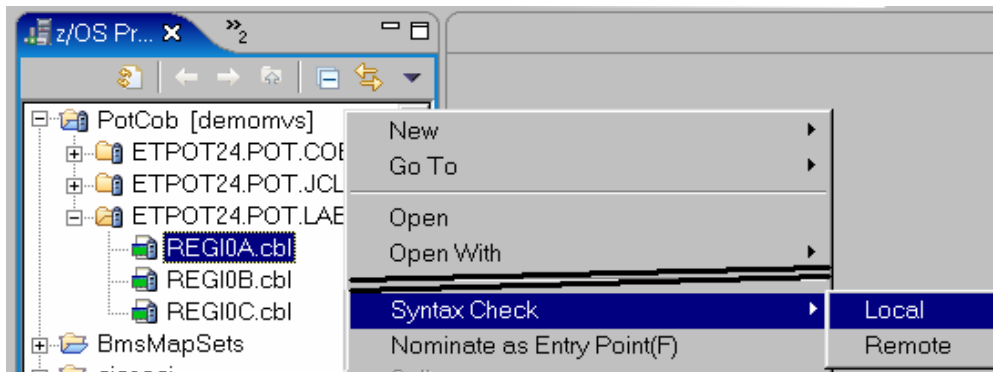
```
000032 PROCEDURE DIVISION.  
000033 010-INITIALIZATION.  
000034 * Initialize Program-work-fields, Program-flags,  
000035 DISPLAY "Program REGIOA STARTING "  
000036 MOVE 2 TO BRANCHFLAG.  
000037 move FIELD-A to  
000038 MOVE 'AAAA'  
000039 MOVE 'BBBB'  
000040 MOVE 'CCCC'  
000041 MOVE "Ente"  
000042 MOVE "WSEI"  
000043 MOVE "REGI"  
000044 020-LOGIC.  
000045 CALL prog  
000046 move 66 to  
000047 divide value1 BY received-from-called GIVING
```

010 BRANCHFLAG
010 FIELD-A
010 FIELD-B
010 FIELD-C
010 Input-name
010 Loop-flag
010 Output-name
010 Program-flags

Find all statements and data names

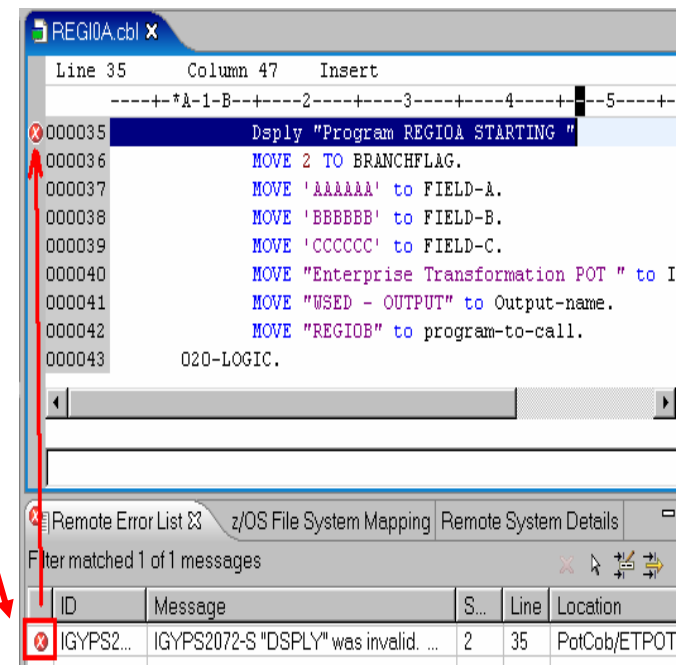
Benefit: Developers complete code more accurately and efficiently.

Use local/remote compiler to do syntax checking



Local Syntax checking..

Just double-click to find the error



Benefit: Uses local CPU

JCL Generation and Submission to z/OS execution

JCL generated from COBOL Code

The screenshot displays the z/OS Projects IDE interface. On the left, a file explorer shows a project structure with folders 'DNET045.STEW.COBOL' and 'DNET045.STEW.JCL'. The file 'REGIOA.jcl' is highlighted with a blue circle and a red box. A blue arrow points from the text 'JCL generated from COBOL Code' to this file. The main editor window shows the JCL code for 'REGIOA.jcl' with the following content:

```

Row 1      Column 1
000001 //REGED511 JOB ,
000002 //  MSGCLASS=H, TIME=( , 4) , REGION=28M, COND=(16, LT)
000003 //  JCLLIB ORDER=DNET045.WSED511.JCL
000004 // *
000005 //DELLIST EXEC PGM=IDCAMS
000006 //SYSPRINT DD SYSOUT=*
000007 IF LASTCC = 8 THEN SET MAXCC = 4
000008 DELETE DNET045.IDECC
000009 IF LASTCC = 8 THEN S
000010 /*
000011 //STP0000 EXEC PROC=EL
  
```

Below the editor is an 'Outline' window showing a summary of the JCL code:

```

//REGED511 JOB ,
//DELLIST EXEC PGM=IDCAMS
//STP0000 EXEC PROC=ELAXFCOC,CI
+//LKED EXEC PROC=ELAXFLNK
//GO EXEC PROC=ELAXFGO,GO=RI
  
```

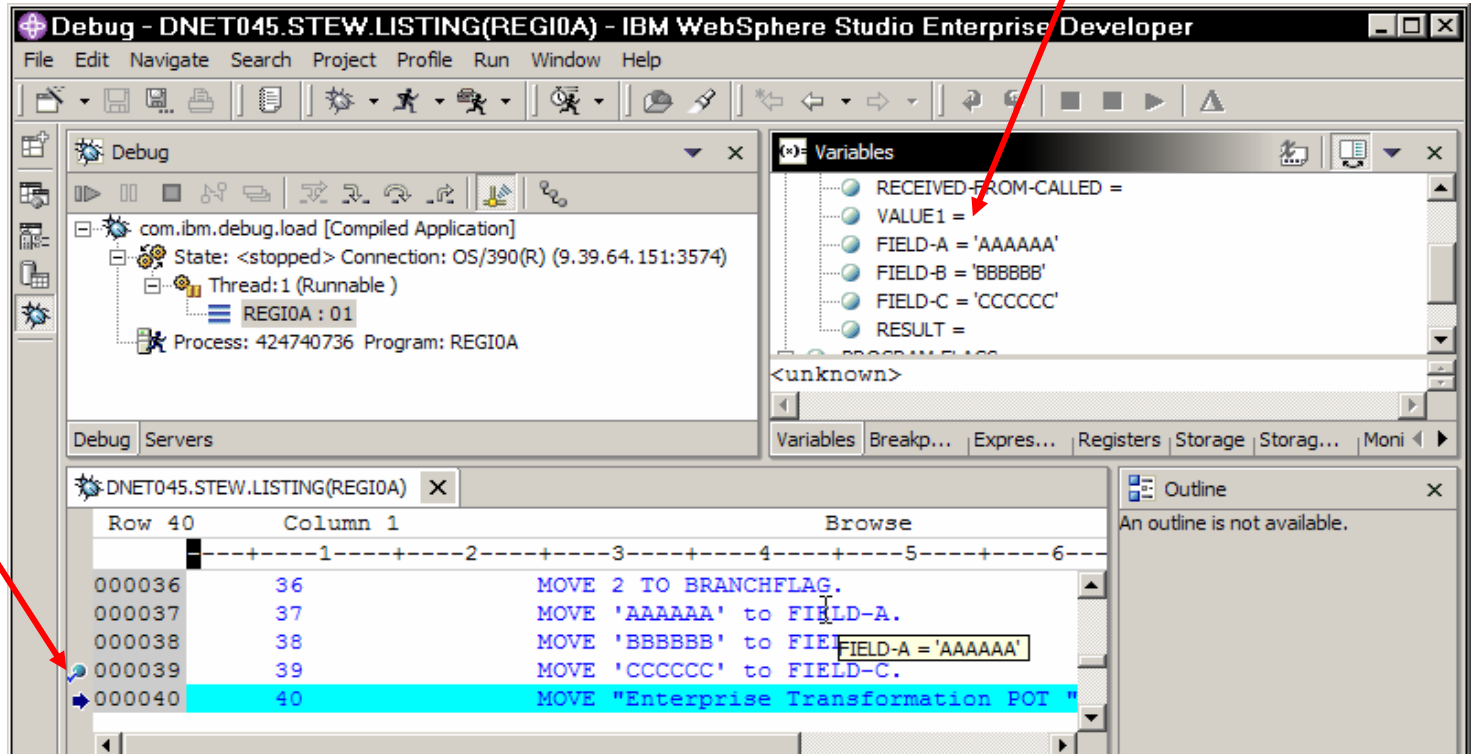
On the right, a context menu is open over the 'REGIOA.jcl' file, with the 'Submit' option highlighted in a red box. Other menu options include 'Add To Another MVS Project ..', 'Browse', 'Open', 'Open With', 'Move...', 'Copy...', 'Delete', and 'Rename'.

Benefit: Developers focused on business logic and not on writing JCL, JCL smart editor, Outline...

Remote/Local debug

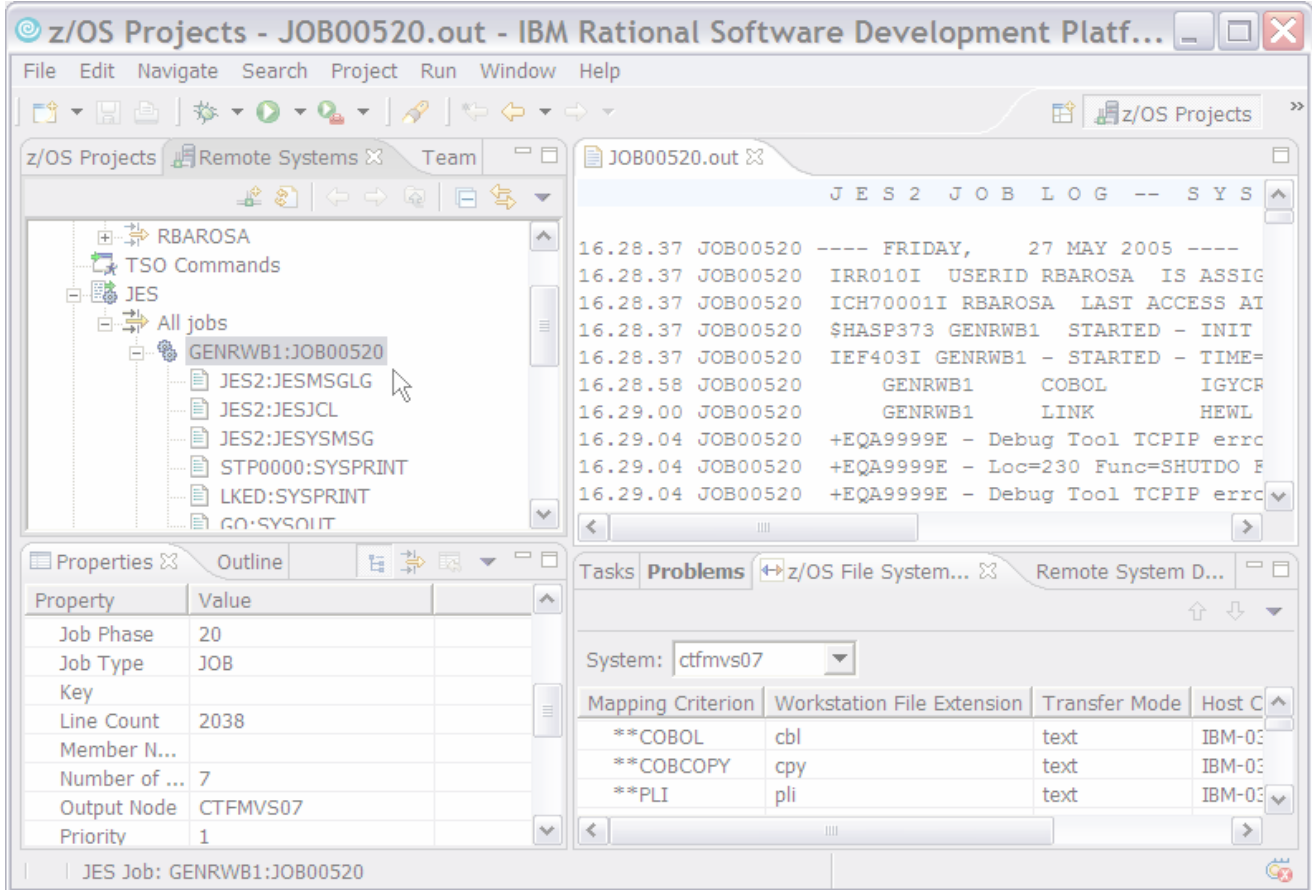
Change contents, etc..

Breakpoints, watchpoints, Jump to, Run to etc..



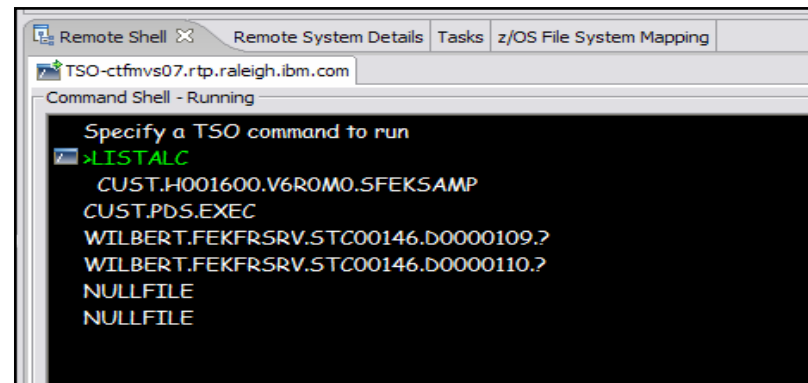
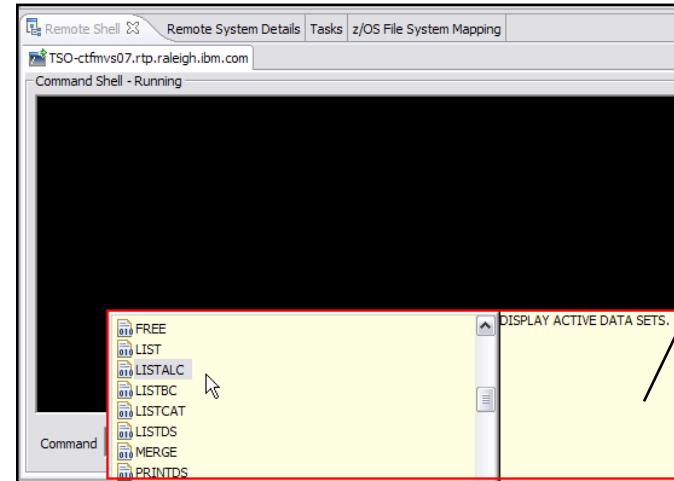
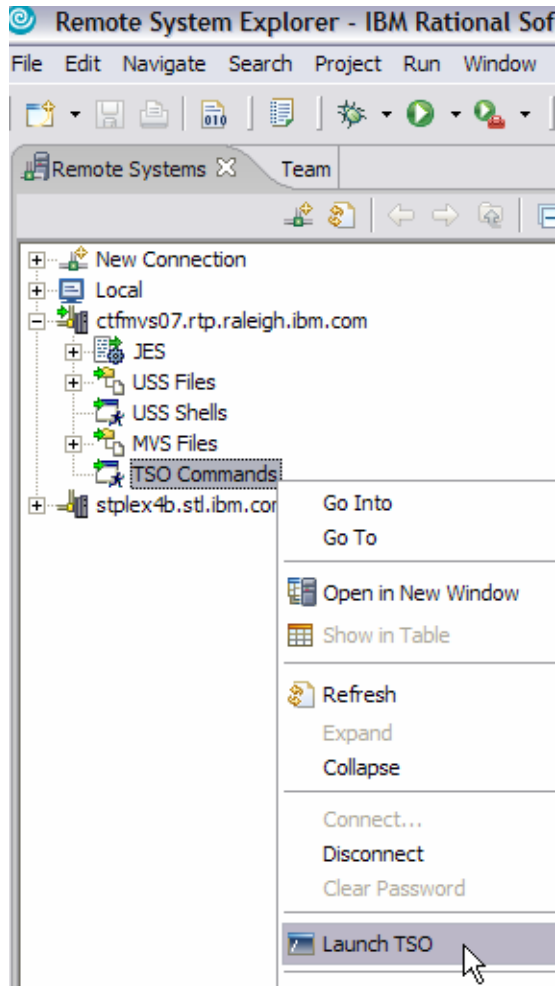
Benefit: Same Debug Perspective used for COBOL, PL1 & Java, etc..

Monitoring Job Output

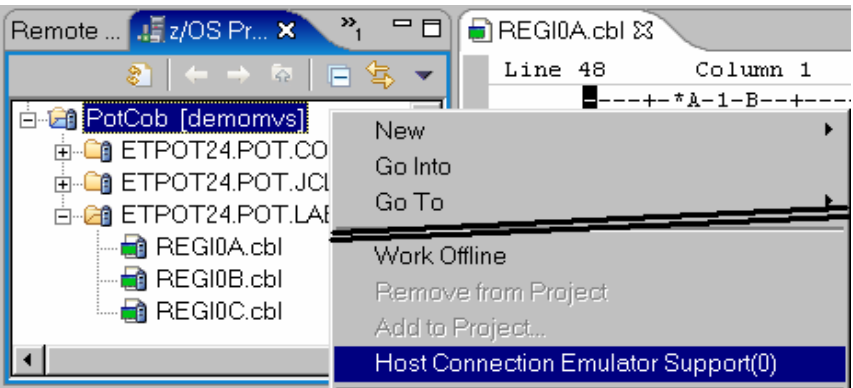


Benefit: Developers do not have to continually switch between systems to use SDSF. No TSO and SDSF needs.

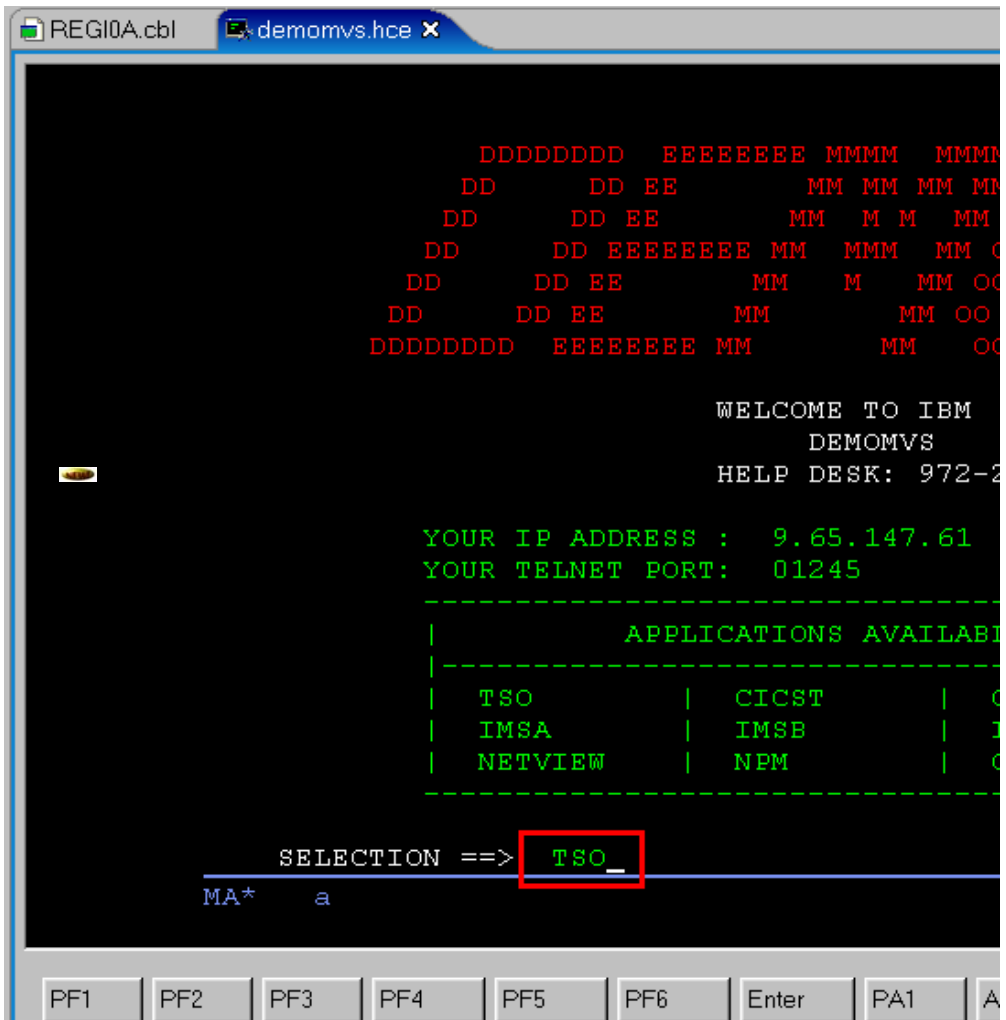
TSO Commands (as well USS)



Invoke 3270 screens from WdZ (for example a TSO session)



Can use macros
to record operating
sequence



Benefit: Eliminates need of terminal emulation, complement developer needs

Benefits of z/OS Application Development

- **Utilizes Workbench features/tools to support COBOL, PL/I, Assembler development for the z/OS platform**
 - ▶ Simplifies development process
 - ▶ Provides consistent development environment
 - ▶ Better interface, no need for TSO
- **Provides development support for traditional runtimes**
 - ▶ CICS, IMS, DB2, batch
- **Help developers with no mainframe skills to work with mainframe assets**

Same tool (Eclipse based) for Java and COBOL or PL1

WebSphere Developer for zSeries V 6.0.1

All features available in Rational Application Developer PLUS..

1. z/OS Application Development Tools

Interactive, workstation-based development for mainframe COBOL, PL/I, ASM applications

2. XML Services for the Enterprise (XSE)

SOA access to CICS V2.2, V3.1 and IMS V9 COBOL applications, COBOL to XML mapping support, COBOL XML converters and WSDL generation

3. CICS BMS Map Support

Visually create and modify BMS Map sets

Work with local or remote maps

Generates JCL

4. z/OS-based DB2 Stored Procedure Builder

Create, Build, Test and Debug DB2 stored procedures on z/OS in either COBOL or PL/I

5. Service Flow Modeler

Implements SOA and Web Services for CICS 3.1

6. Enterprise Generation Language (EGL) feature generating COBOL

Simple, high level programming specifications

Create full-function COBOL and J2EE Java applications

XML Services for the Enterprise (XSE)

- **Web Services Enablement wizard (bottom-up)**
 - Generate Web Service interface from existing COBOL application
 - Bottom-up approach since COBOL at the bottom (base) of the creation process
- **Web Services Enablement wizard (top-down)**
 - Generate COBOL Program and copybooks from existing WSDL
- **Web Services Enablement wizard (meet-in-the-middle)**
 - Map existing WSDL or XML to existing COBOL app.
 - Meet-in-the-middle since Web Services/XML definition “meets” or maps to the existing COBOL interface

WebSphere Developer for zSeries V 6.0.1

All features available in Rational Application Developer PLUS..

1. z/OS Application Development Tools

Interactive, workstation-based development for mainframe COBOL, PL/I, ASM applications

2. XML Services for the Enterprise

SOA access to CICS V2.2, V3.1 and IMS V9 COBOL applications, COBOL to XML mapping support, COBOL XML converters and WSDL generation

3. CICS BMS Map Support

Visually create and modify BMS Map sets

Work with local or remote maps

Generates JCL

4. z/OS-based DB2 Stored Procedure Builder

Create, Build, Test and Debug DB2 stored procedures on z/OS in either COBOL or PL/I

5. Service Flow Modeler

Implements SOA and Web Services for CICS 3.1

6. Enterprise Generation Language (EGL) feature generating COBOL

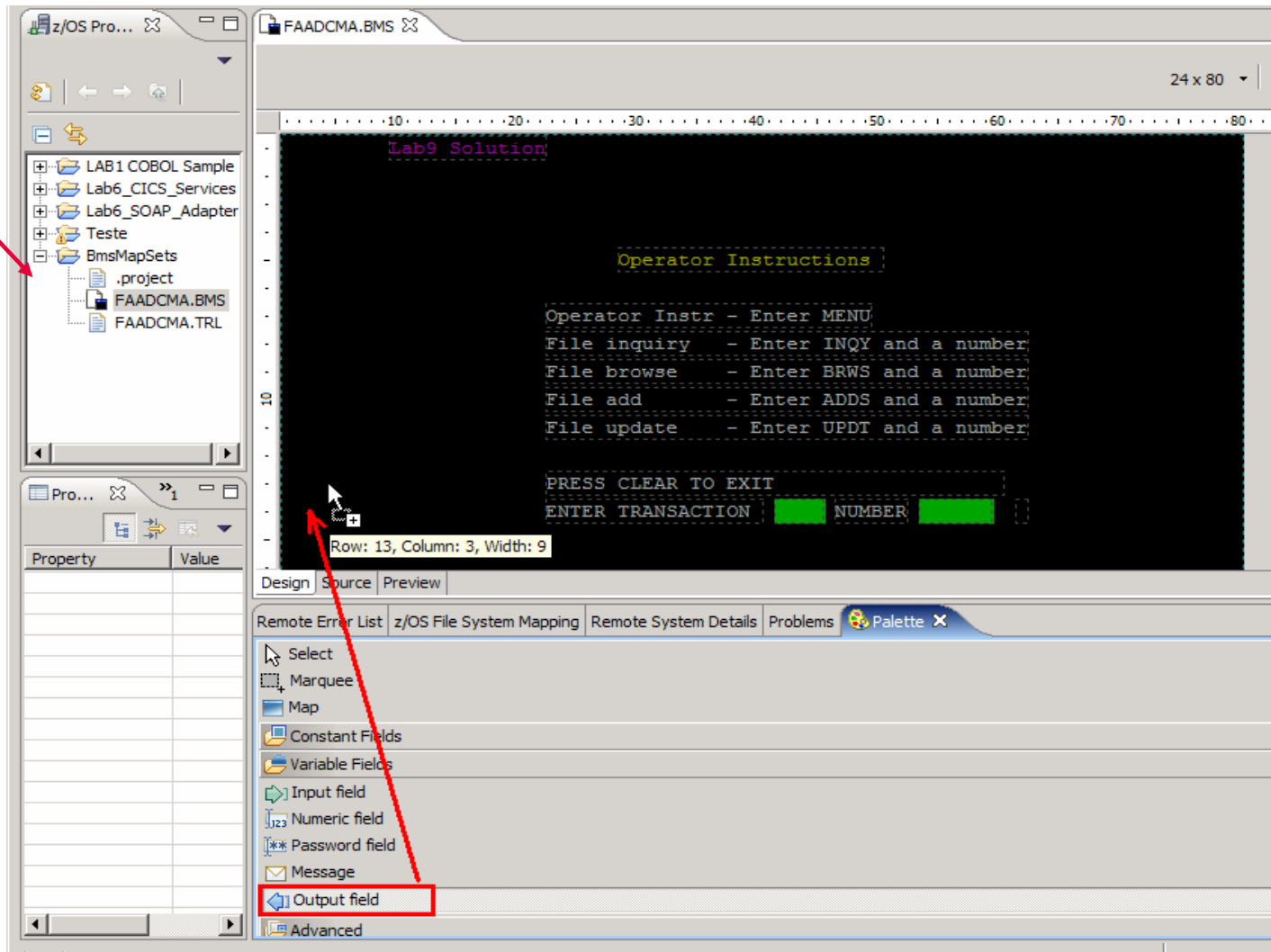
Simple, high level programming specifications

Create full-function COBOL and J2EE Java applications

CICS BMS Map Support

Create BMS Project

- Wizard for creating new BMS map set files
- Drag & Drop BMS editor
- Design, Source and Preview views
- Create new or import/edit existing BMS maps
- Works with local and remote scenarios



WebSphere Developer for zSeries V 6.0.1

All features available in Rational Application Developer PLUS..

1. z/OS Application Development Tools

Interactive, workstation-based development for mainframe COBOL, PL/I, ASM applications

2. XML Services for the Enterprise

SOA access to CICS V2.2, V3.1 and IMS V9 COBOL applications, COBOL to XML mapping support, COBOL XML converters and WSDL generation

3. CICS BMS Map Support

Visually create and modify BMS Map sets

Work with local or remote maps

Generates JCL

4. z/OS-based DB2 Stored Procedure Builder

Create, Build, Test and Debug DB2 stored procedures on z/OS in either COBOL or PL/I

5. Service Flow Modeler

Implements SOA and Web Services for CICS 3.1

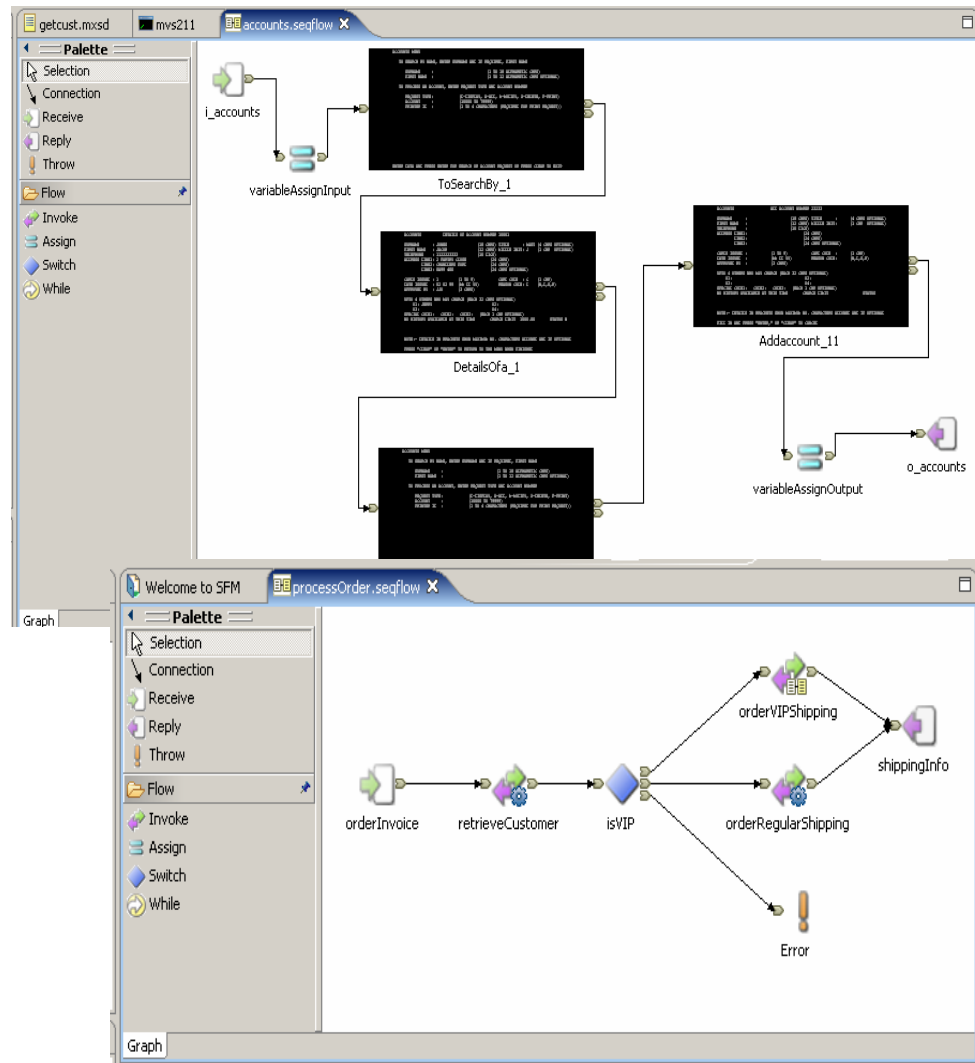
6. Enterprise Generation Language (EGL) feature generating COBOL

Simple, high level programming specifications

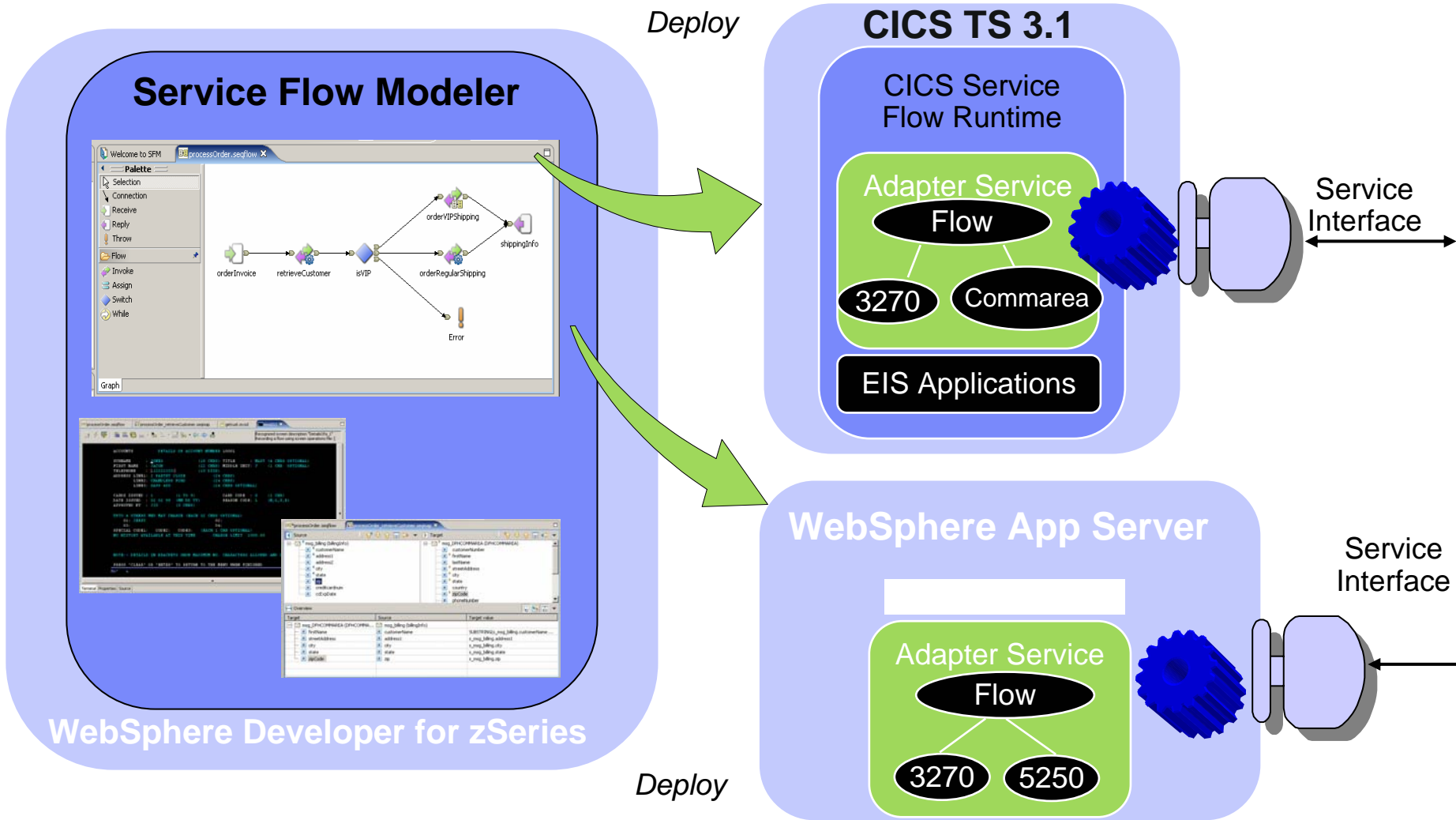
Create full-function COBOL and J2EE Java applications

What is Service Flow Modeler?

- Service Flow Modeler is a tool to build service flows out of your existing **Commarea** and **Terminal** based CICS applications.
- It allows you to:
 - Model business processes
 - Implement business processes by aggregating multiple transaction invocations, terminal interactions, and sub-flows
 - Deploy these aggregations to runtimes in CICS Transaction Server V3.1 or WebSphere Application Server
 - Optionally deploy business process as a web service
- Development concepts consistent with other SOA development tasks



Supported Runtimes



WebSphere Developer for zSeries V 6.0.1

All features available in Rational Application Developer PLUS..

1. z/OS Application Development Tools

Interactive, workstation-based development for mainframe COBOL, PL/I, ASM applications

2. XML Services for the Enterprise

SOA access to CICS V2.2, V3.1 and IMS V9 COBOL applications, COBOL to XML mapping support, COBOL XML converters and WSDL generation

3. CICS BMS Map Support

Visually create and modify BMS Map sets

Work with local or remote maps

Generates JCL

4. z/OS-based DB2 Stored Procedure Builder

Create, Build, Test and Debug DB2 stored procedures on z/OS in either COBOL or PL/I

5. Service Flow Modeler

Implements SOA and Web Services for CICS 3.1

6. Enterprise Generation Language (EGL) feature generating COBOL

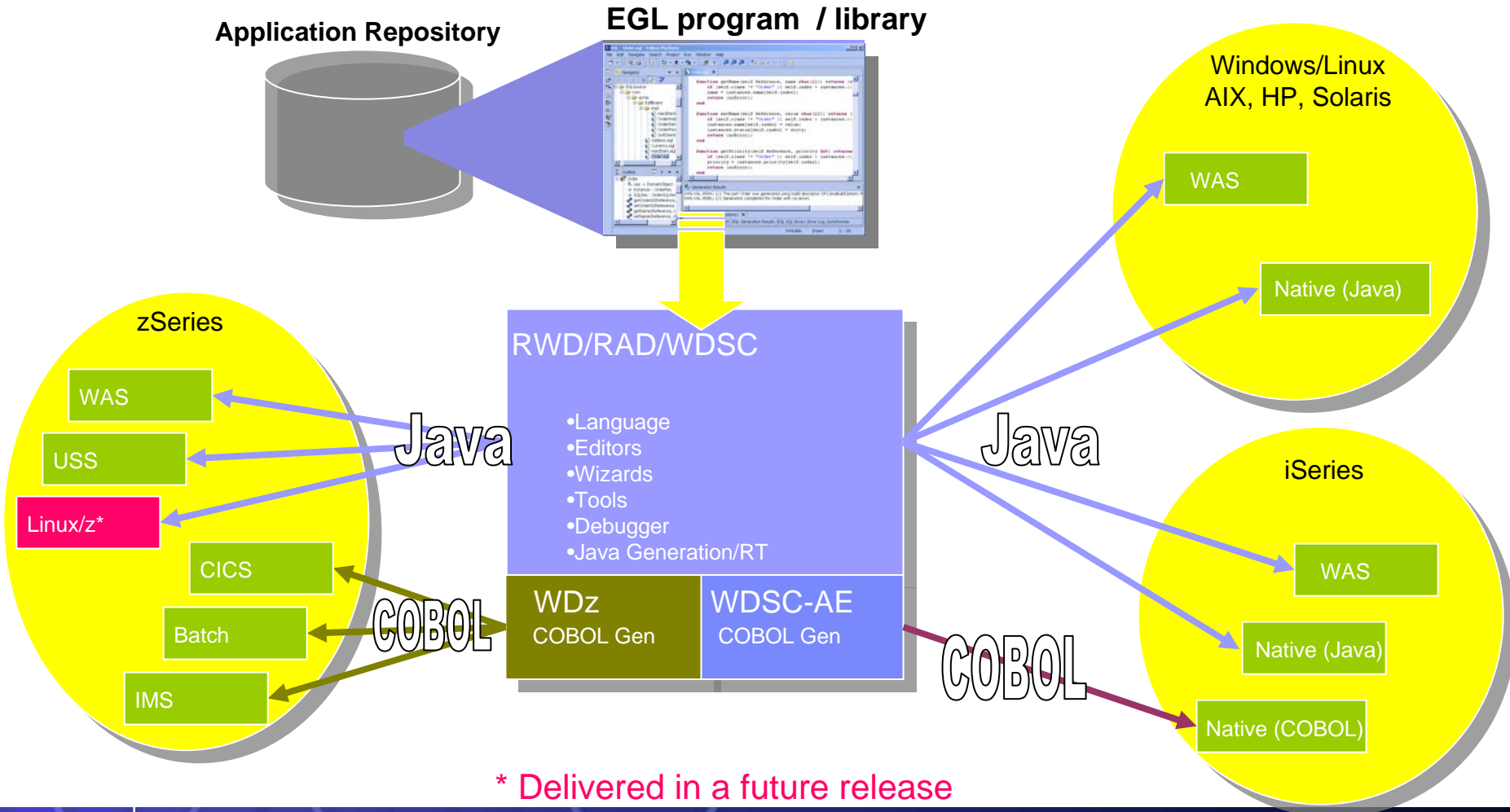
Simple, high level programming specifications

Create full-function COBOL and J2EE Java applications

What is EGL ?

- **EGL** = **E**nterprise **G**eneration **L**anguage
 - High level programming specifications
 - Hides complexities of implementation technology
 - For the non-Java programmer
 - For the non-CICS programmer
- Special Parts + Scripting Language
- Interactive Development and Debugging
 - Environment independent language
 - Built-in debugger
 - Can be used for RAD development

Environments supported



Scenario: Support for Mixed Workload -

WDz:

- Brings the power of J2EE, rapid Application Development, and robust team support to diverse enterprise IT organizations
- Consists of:
 - ▶ An intuitive, visual construction based on open standards (Java Server Faces)
 - ▶ Broad SOA support through Web Services and JCA with specialized zSeries capabilities
 - ▶ An easy to learn, language neutral environment for rapid application development
 - ▶ Comprehensive state-of-the-art facilities for developing, debugging and deploying Java, COBOL, EGL, and PL/I applications and services
 - ▶ WebSphere Host Access Transformation Services (HATS)

Benefits:

- Increase developer productivity
- Leverage existing processing
- Integrate with lifecycle
- Extend skill sets across the organization
- Enterprise Generation Language
 - ✓ **Limits need for Java or traditional expertise**
 - ✓ Generate Java for WAS
 - ✓ Generate COBOL for CICS



Reuse is key

- Reuse is a key part of the SOA value proposition
- IBM's enterprise transformation tools make reuse more achievable



IBM WEBSHERE PRESENTS

**YOU VS REWRITE, REVISE
RE-EVERYTHING**

INFLEXIBILITY: MEET SOA

PLUS: TANGIBLE BUSINESS BENEFITS ★ BEST PRACTICES FOR BEST RESULTS
OVER 10 YEARS OF WORLD-CLASS INTEGRATION EXPERTISE

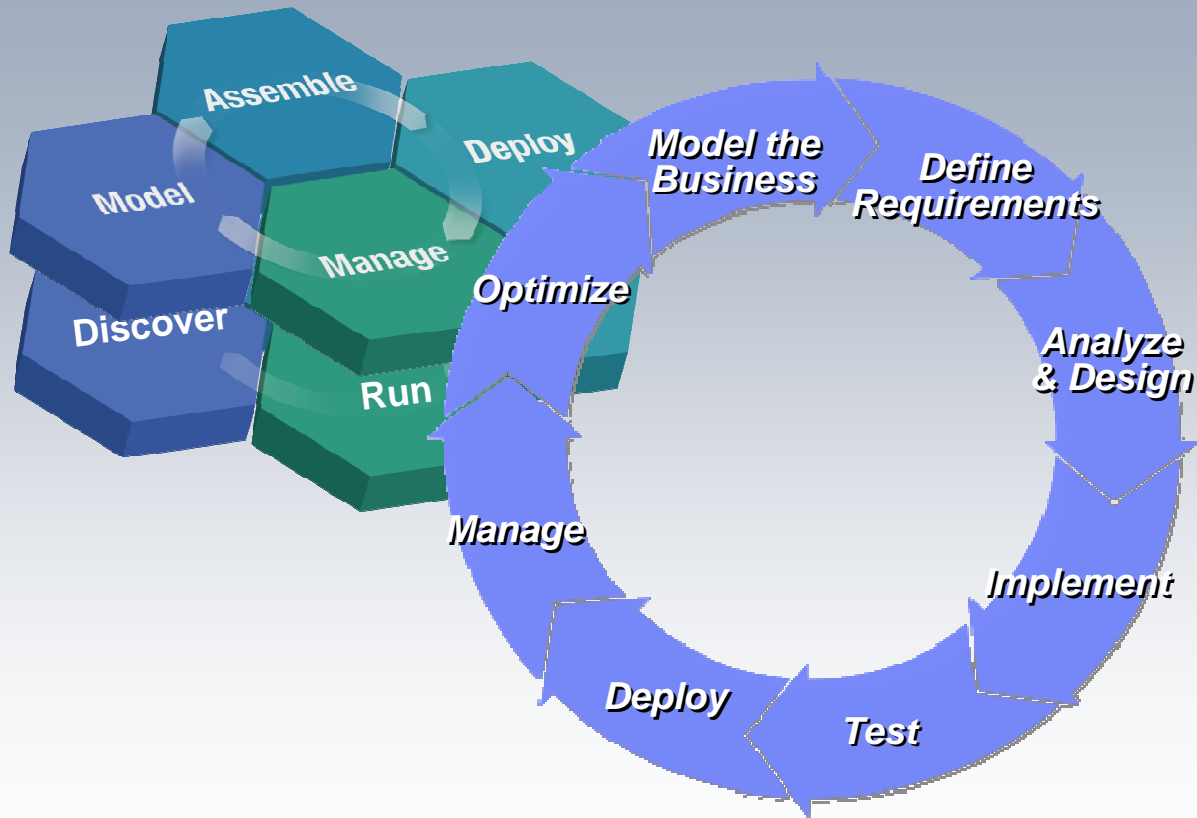
FEATURING:

A FASTER, EASIER WAY TO IMPLEMENT TRUE SOA	SUPPORT FOR OVER 80 O.S. CONFIGURATIONS	UNPARALLELED INDUSTRY KNOWLEDGE & PROCESS SKILL
--	---	---

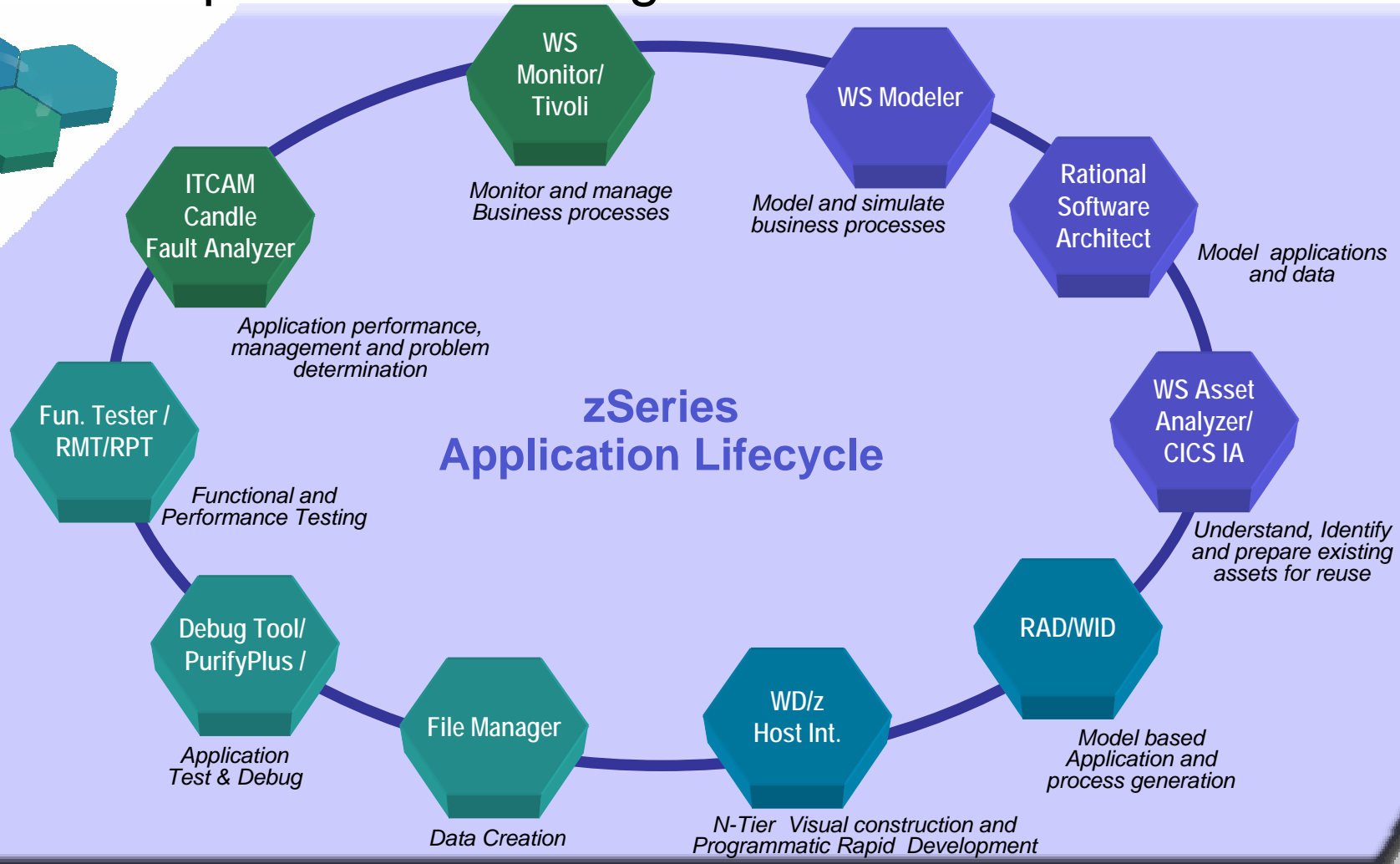
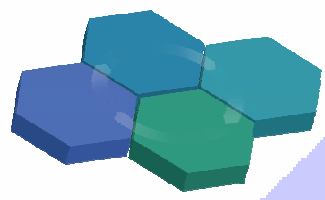
IBM MIDDLEWARE. POWERFUL. PROVEN.
FIGHT BACK AT WWW.IBM.COM/MIDDLEWARE. THIS IS A RIP-AND-REPLACE-FREE EVENT.

IBM, the IBM logo and WebSphere are registered trademarks or trademarks of IBM International Business Machines Corporation in the United States and/or other countries. ©2009 IBM Corporation. All rights reserved.

Enterprise Platform – Life Cycle



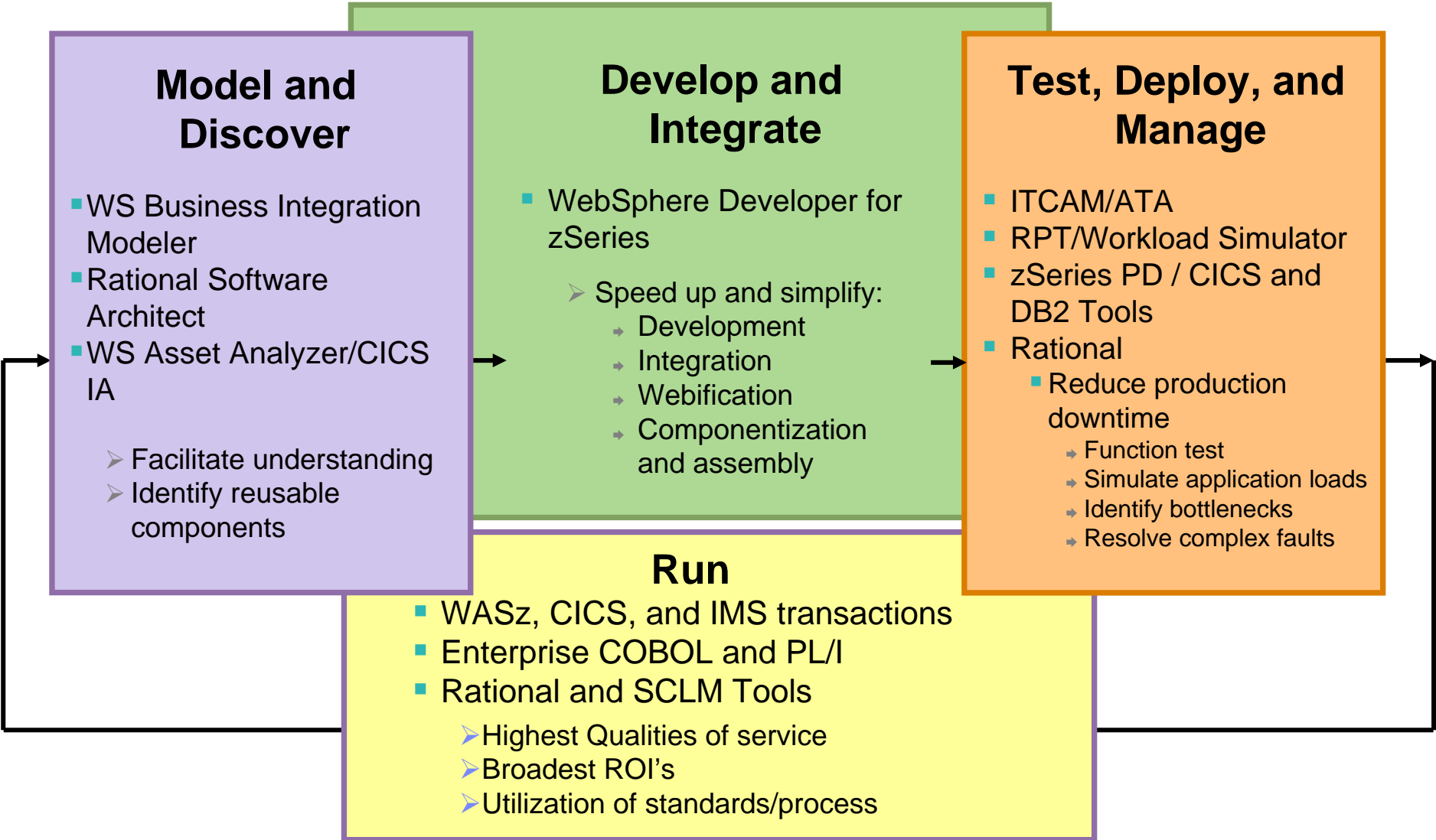
IBM zSeries process and integrated workflow



Common Process and Guidance – Reqpro, ClearQuest, Rational Method Composer

Software Configuration Management – ClearCase, ClearQuest, SCLM

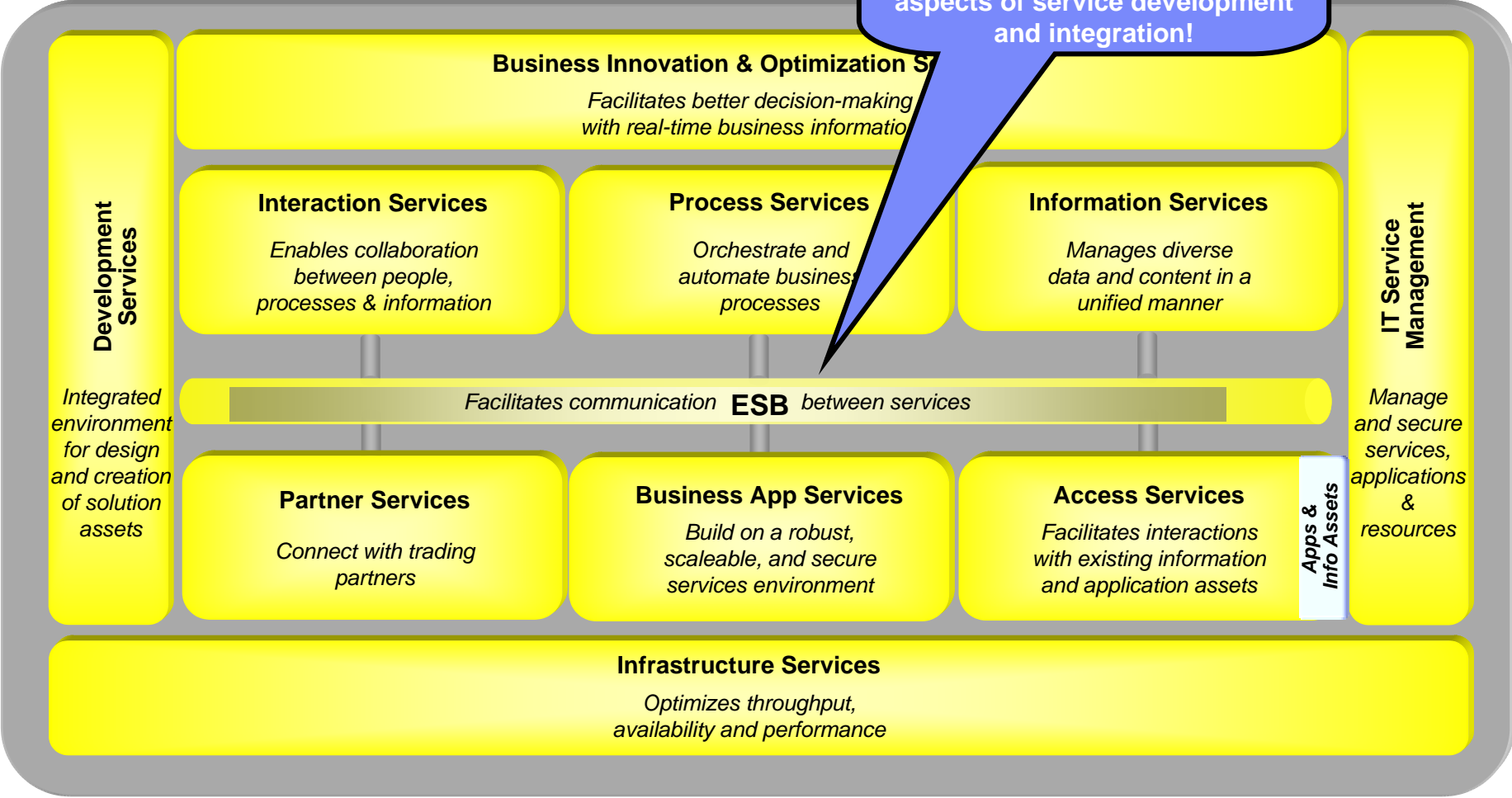
IBM SDP for zSeries and Composites



SOA Reference Architecture

Supporting your SOA Lifecycle

Note that all the blocks in this diagram are yellow. The SOA Reference Architecture applies directly to zSeries for all aspects of service development and integration!



 Leverage zSeries middleware for maximum business flexibility.

Agenda

- **Where do we fit**
- Introduction Modern SOA, CICS, IMS, WAS and WDz
- **Introduction to tool concepts**
- **Detail information and demonstration of ZOS development**

Enabling a robust, flexible SOA runtime environment

While maximizing the value of existing assets **Fully SOA capable!**

WebSphere Application Server V6

March 2005

- Extend existing Java assets with support for Web Services standards and standards-based messaging
- Help ensure 24x7 availability of business-critical applications with clustering and high availability
- Build and deploy Web Services quickly and easily with rapid development and deployment features

CICS Transaction Server V3.1

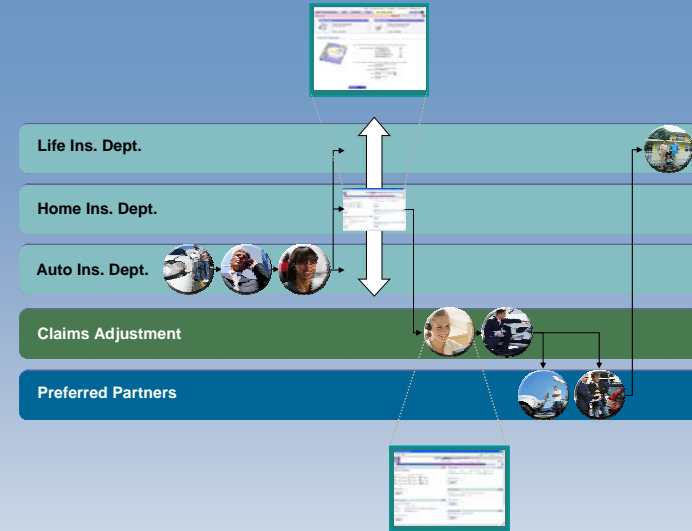
March 2005

- Exploit provider/requestor Web service support for CICS assets, based on full Web service standards
- Extend the value of CICS transactions in a mixed language environment
- Build Web services from CICS transactions with no change to existing applications.

IMS Transaction and Database V9

October 2004

- Exploit Web service support for IMS assets, based on full Web service standards
- Extend the value of IMS transactions in a mixed language environment
- Build Web services from IMS transactions with no change to existing applications



#1 in market share for Application Server software

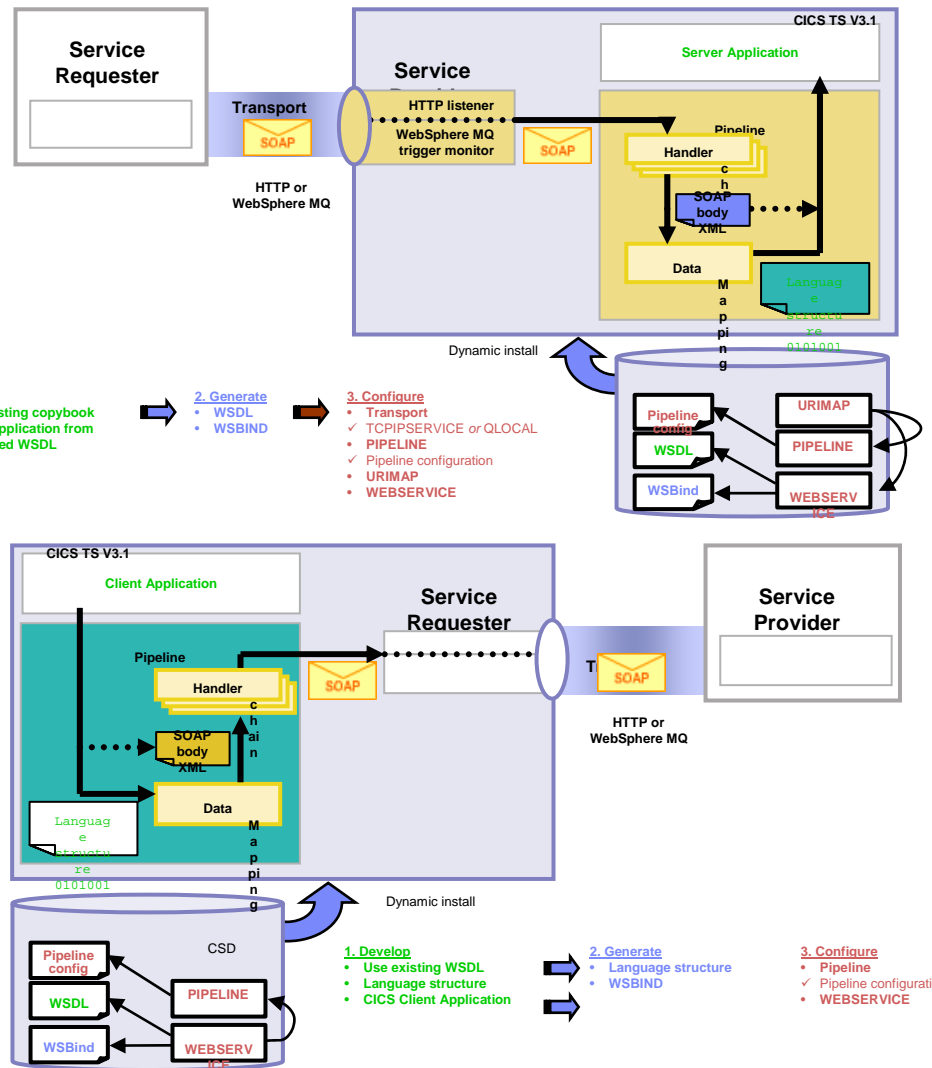


IBM WebSphere Application Server comes out on top

35+ years of maturity and innovation in transaction and data systems

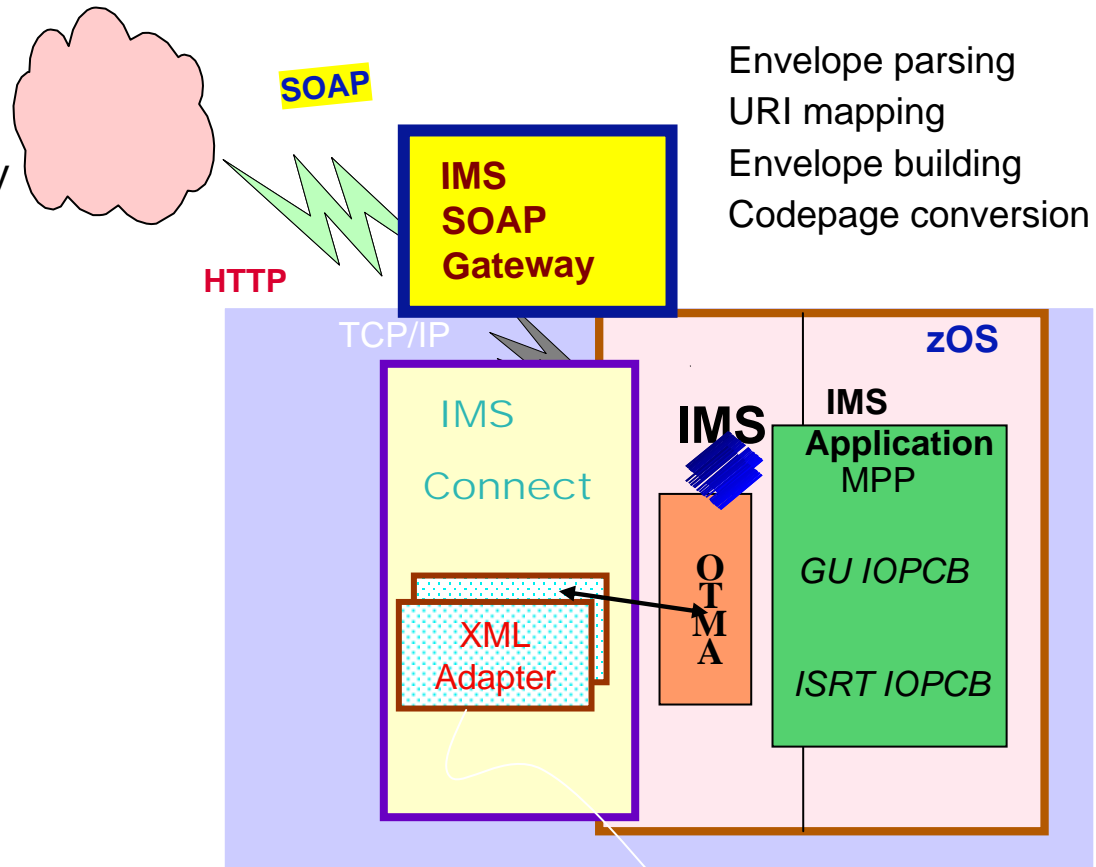
CICS Web Services

- Web services capabilities extend CICS applications directly to a Service Oriented Architecture
 - A CICS application can now be a Web service provider and requester
- Evolution of SOAP for CICS feature
 - Simplification of pipeline and system management
 - Fully integrated into CICS
 - RDO, problem determination, monitoring & statistics
 - New tooling support for easier application development
 - Guidance provided to assist migration from the SOAP for CICS Feature
- Rich set of Web services standards supported
 - SOAP 1.1 and 1.2 to send and receive Web services messages
 - WS-I Basic Profile 1.0a for interoperability with between providers and requesters using SOAP
 - WS-Coordination extensible coordination framework, and specific coordination of transactions
 - WS-AtomicTransaction for transaction coordination
 - WS-Security for authentication and encryption of all or part of a message
 - SOAP Message Security, Username Token Profile 1.0, X.509 Certificate Token
- Both HTTP and WebSphere MQ network layers supported
 - For flexible deployment options dependant on application and IT requirements
 - CICS applications acting as providers or requesters are agnostic to the transport mechanism used



IMS And Web Services

- SOAP for IMS at www.ibm.com/ims
 - Technology preview recently announced in February
- Maximize re-use of customer enterprise assets via standard interfaces
- Support collaboration among IMS and IBM and non-IBM components, both within and beyond enterprise boundaries

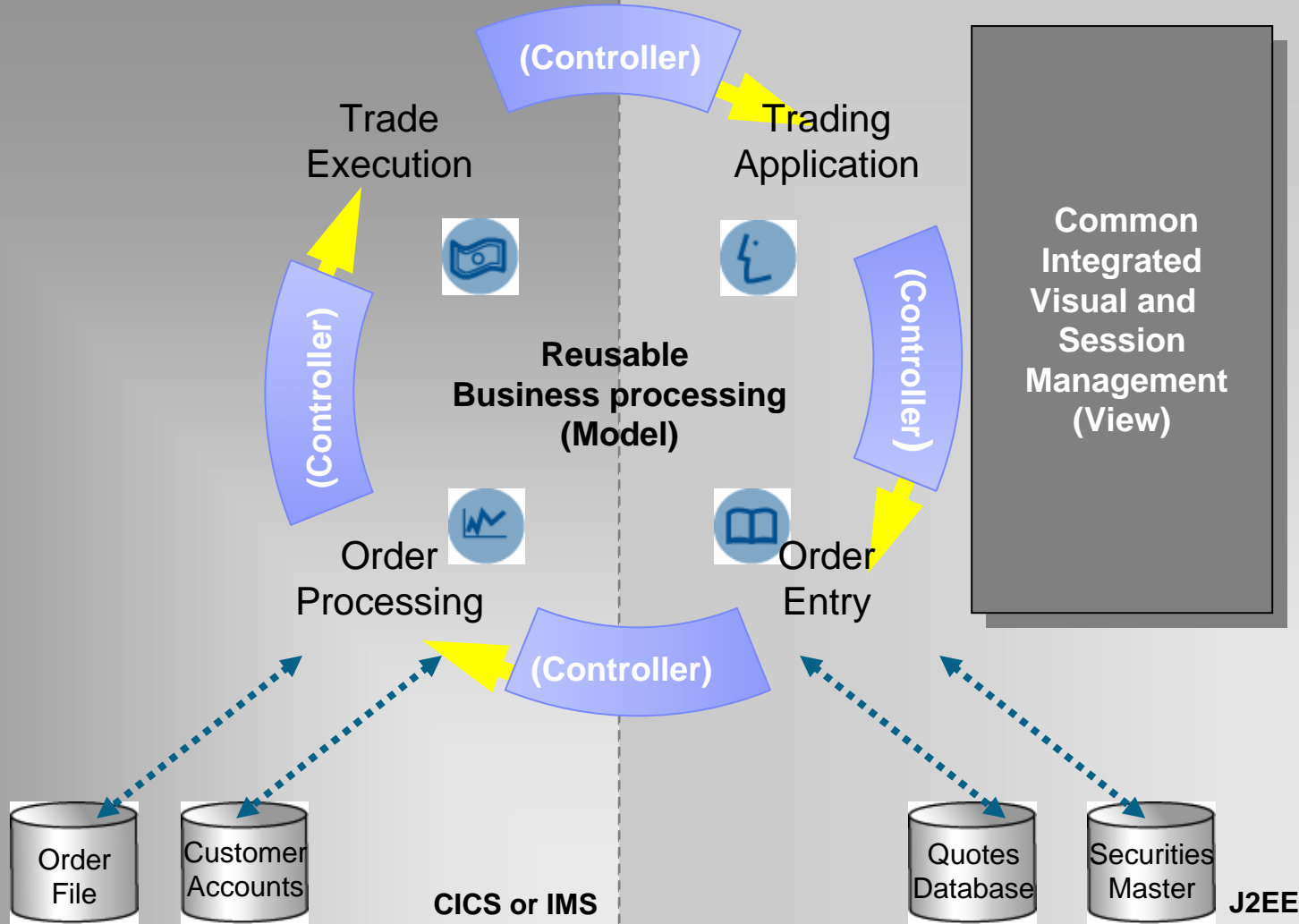


Envelope parsing
 URI mapping
 Envelope building
 Codepage conversion

WSED-generated XML adapter for COBOL

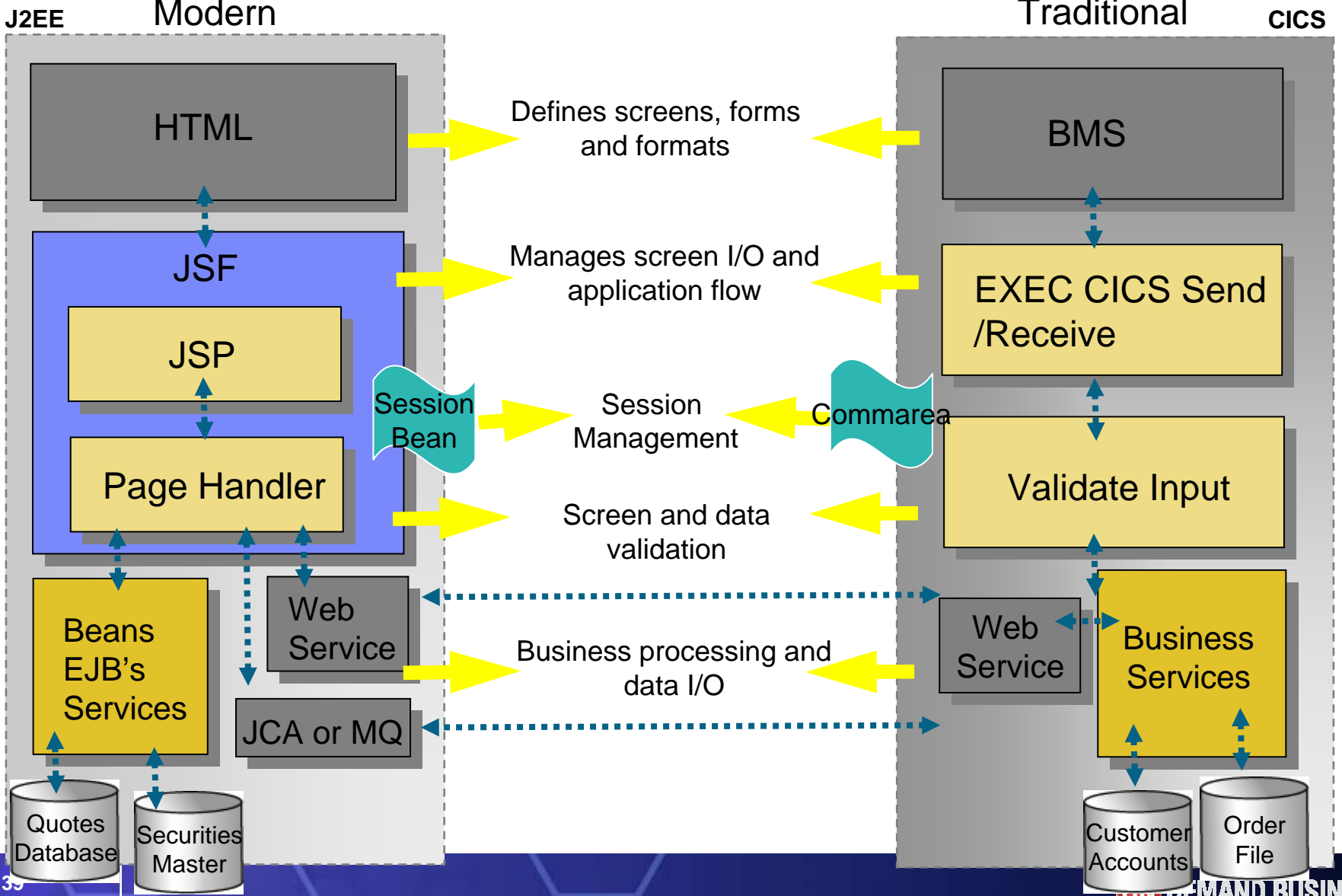
Mixed Workload Application Components

Spans multiple system and middleware boundaries



It's not that different

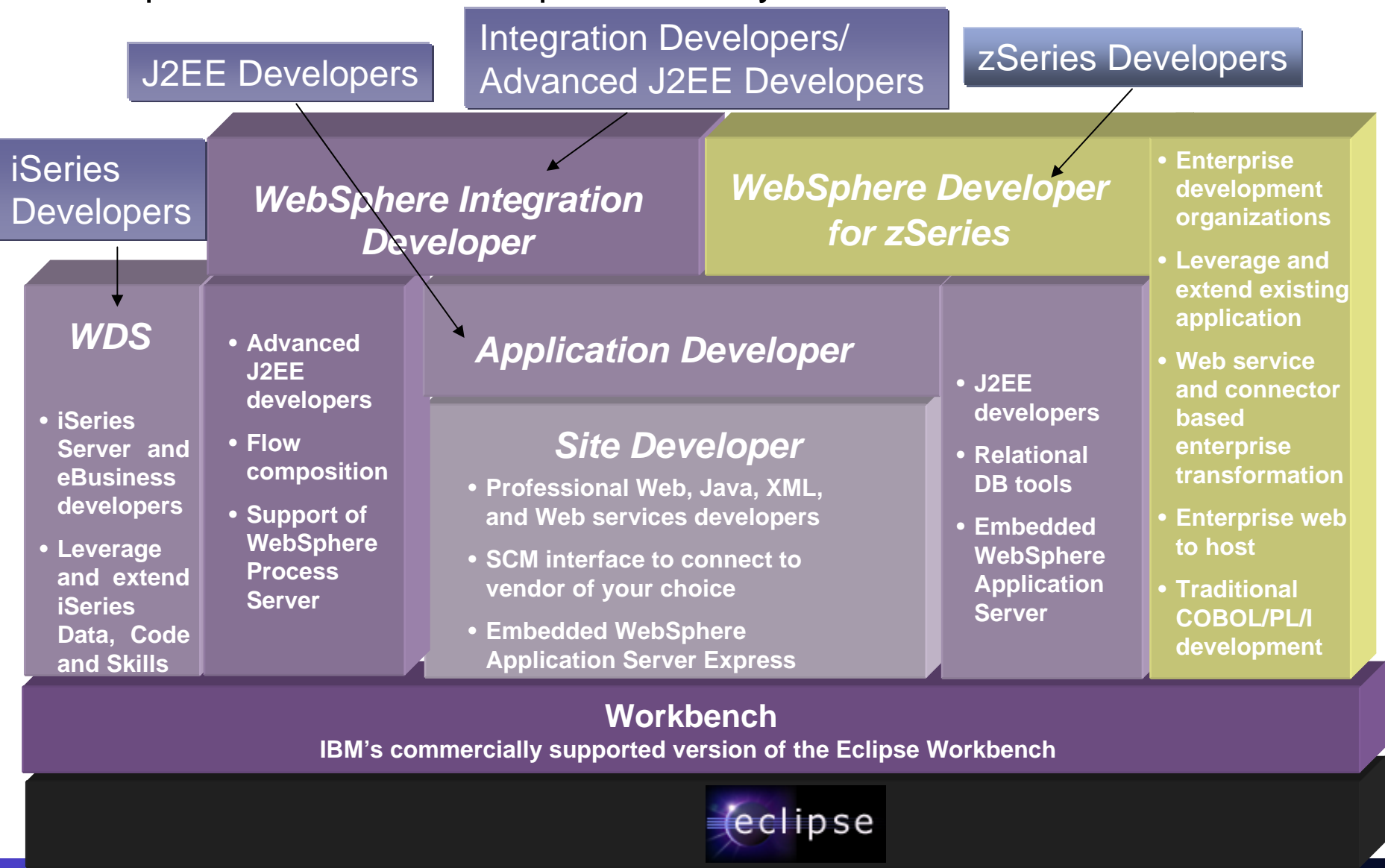
Spans multiple system and middleware boundaries



Agenda

- **Where do we fit**
- **Introduction Modern SOA, CICS, IMS, WAS and WDz**
- Introduction to tool concepts
- **Detail information and demonstration of ZOS development**

WebSphere/Rational Development Family



WebSphere Developer for z/OS

What is WebSphere Developer for z/OS?

Brings the power of modern application architectures and rapid application development and robust team support, to diverse enterprise IT organizations

- Intuitive, visual construction based on open standards (JSF and Struts)
- Broad SOA support through Web services and JCA linking visual environments and user sessions to CICS QOS
- Easy to learn, COBOL like language for rapid UI and Business dev.
- Facilities to develop, debug and deploy Java, COBOL, & PL/I applications and services



V6 New Functionality

CICS V3 exploitation - Subsystem support latest – CICS, WAS, DB2

- Connectivity enhancements
 - WSDL automation from existing processing
 - Support for new CICS WS run timemarshallers
 - XML based COBOL adapter enhancements
 - JCA connectors supporting latest CTG
- Modern Architectural enhancements
 - Service Flow Modeler support (Preview)
 - Leverages support for channels
- Traditional support for:
 - EGL support for VG based Web Transactions
 - BMS Editor
- Integration with other IBM application lifecycle products
- Eclipse V3 exploitation

V6.0.1

- GA Service Flow Modeler
- 3rd party and open SCM support
- Preview: CICS Patterns “List Detail, CRUD, etc.)
- ...and more

Benefits

Single tool for all application transformation

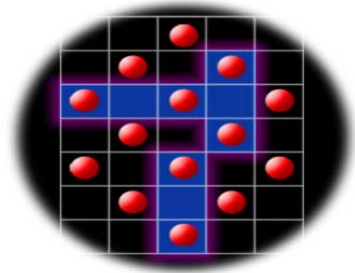
- Increase developer productivity
- Leverage existing processing by enabling legacy assets to be used in SOA’s
- Integrate with lifecycle
- Extend skill sets across the organization
 - Enterprise Generation Language limits need for Java or traditional expertise

zOS Application Development tools

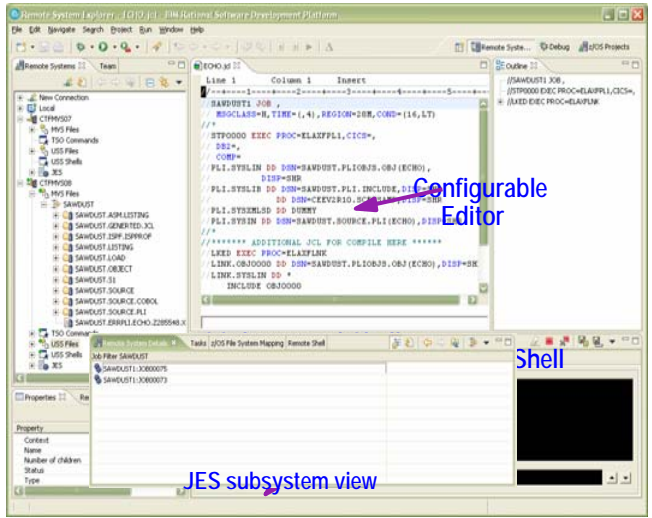
- Interactive, workstation-based environment
 - Faster development with less errors
 - Work offline or online
 - Local/workstation projects

- Edit/compile/debug on the workstation
 - Remote or Local
 - Language sensitive editors for COBOL, PL/I, ASM, JCL
 - BMS Map development

- Interactive access to zOS
 - Job generation, submission, and monitoring
 - TSO/USS command execution



Traditional applications and COBOL/PL/I Services



Disconnect -vs- Connected

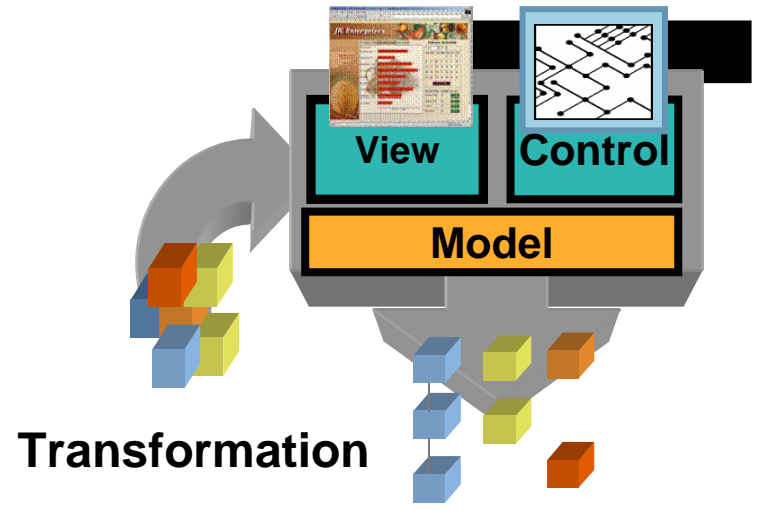
MVS PDS members

Data set characteristics

Configurable Editor

Shell

JES subsystem view

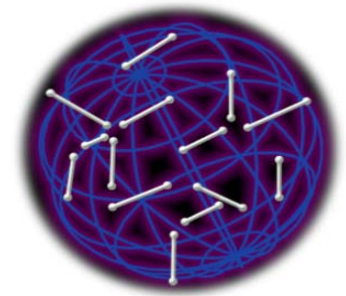


Web Development tools

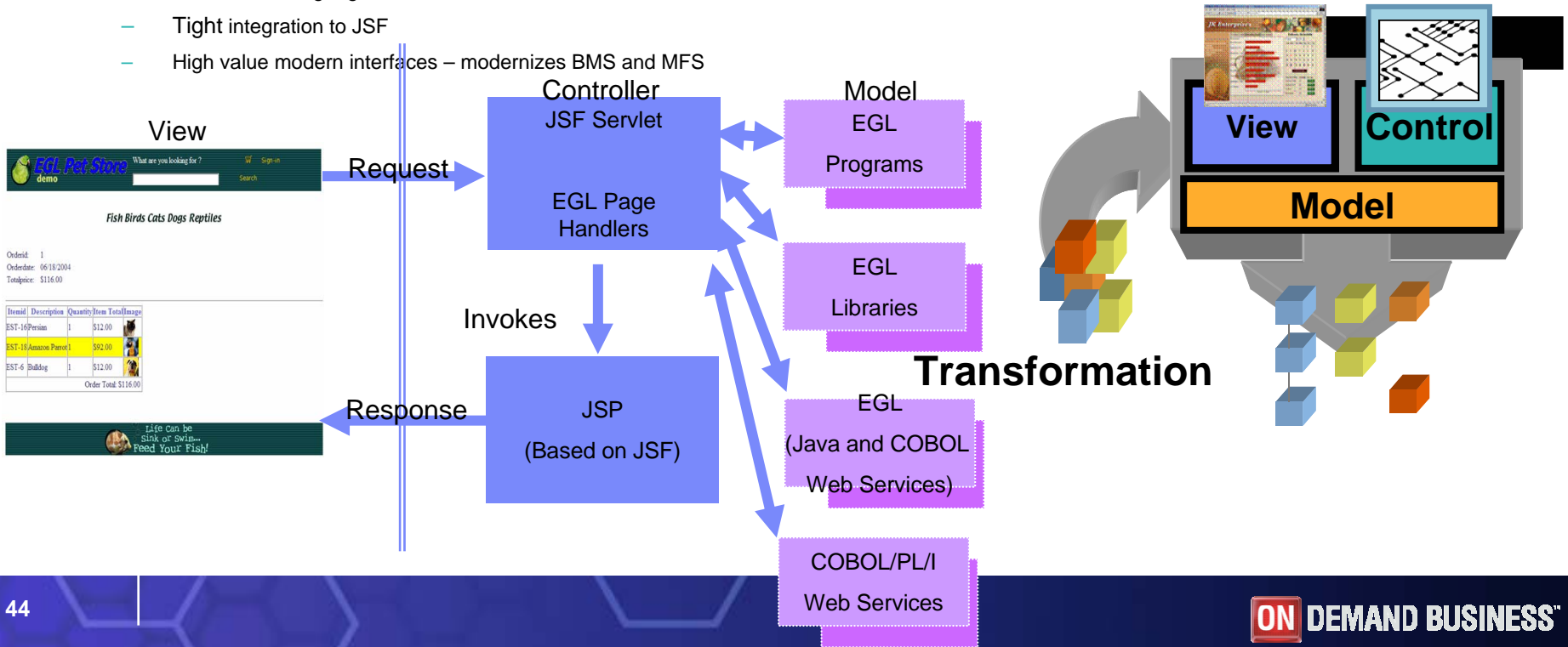
- Interactive, Web development
 - Static and Dynamic Web development
 - XML

- Java Development
 - Java and J2EE development
 - Java Server Faces
 - Struts

- EGL 4GL Java/Web development
 - Generate to language of Choice
 - Tight integration to JSF
 - High value modern interfaces – modernizes BMS and MFS

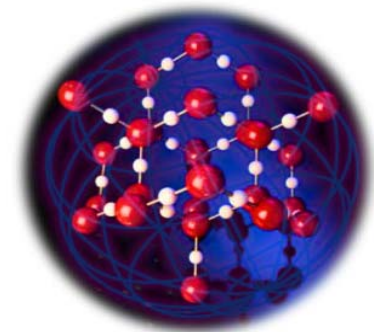


Web applications and services

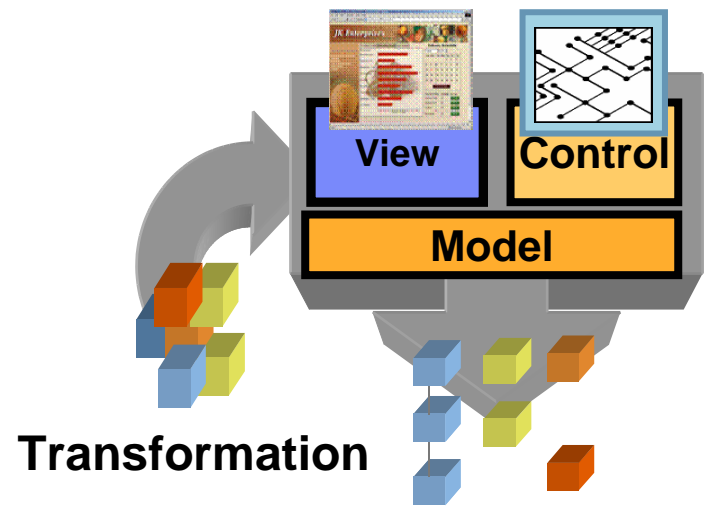
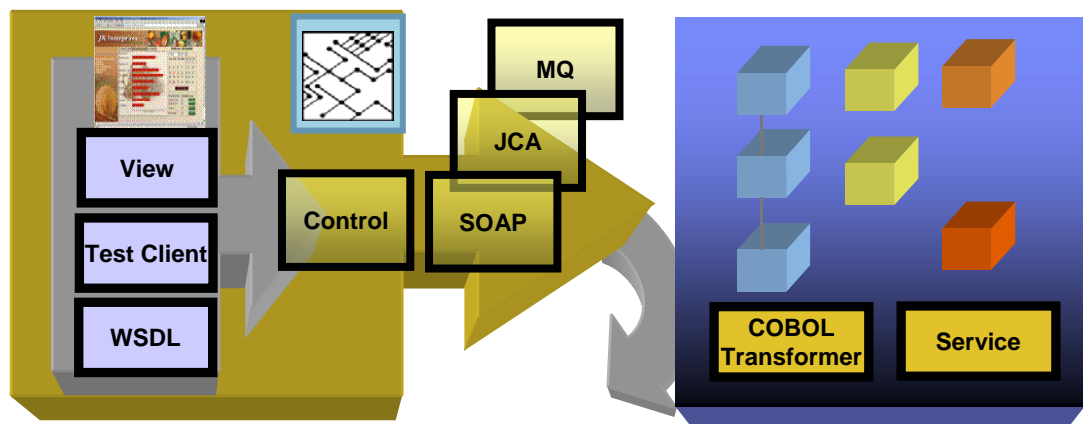


z/OS Composite Development tools

- Transition of Traditional environments to Web and Mixed Workload or Composite applications
- SOA / SOAP / XML / Enablement
- JCA Support
- Service Flow Modeler
- HATS
- Enterprise Generation Language (EGL) / JSF
 - COBOL/CICS generation
 - Java generation



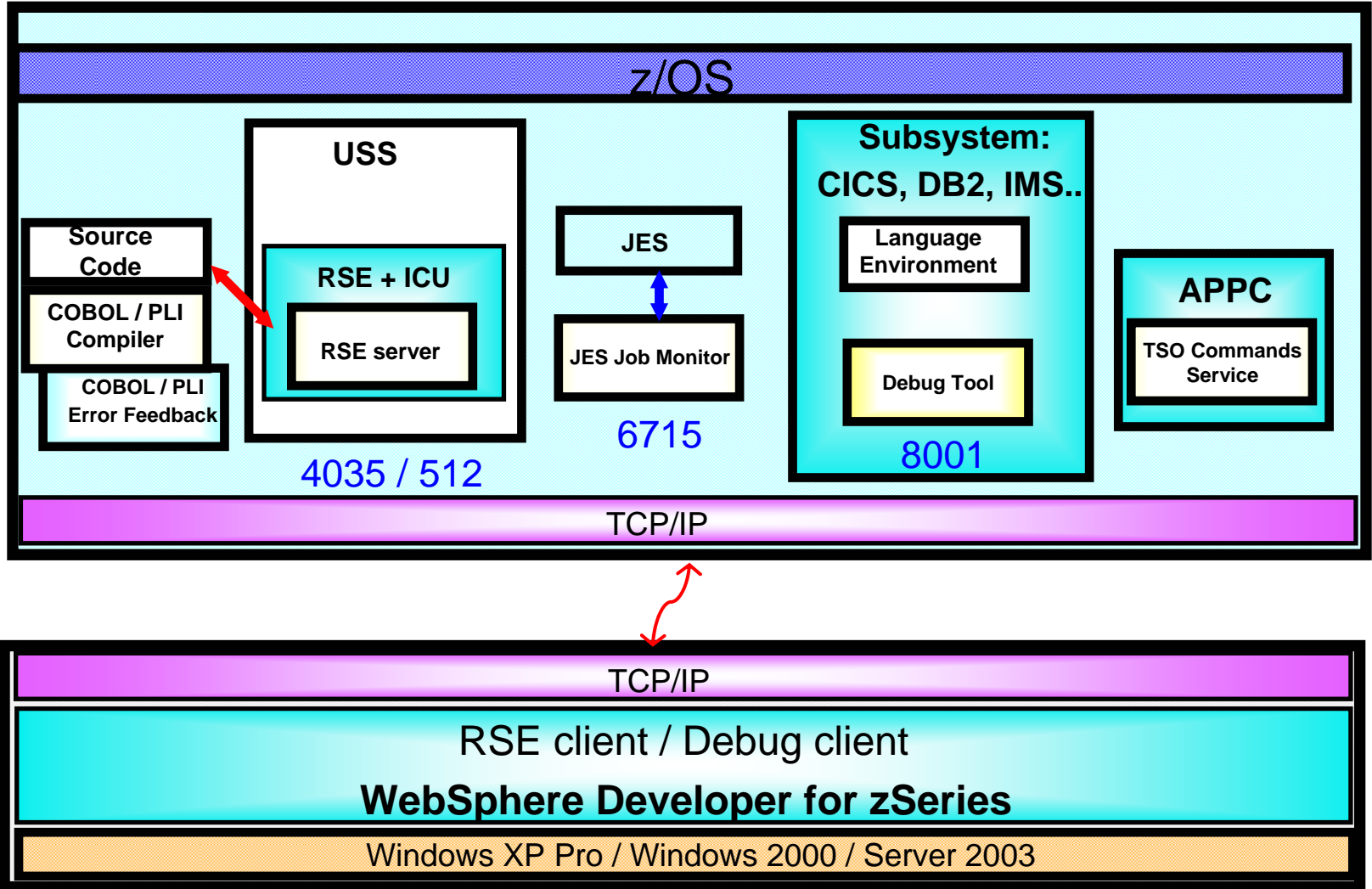
On Demand



Agenda

- **Where do we fit**
- **Introduction Modern SOA, CICS, IMS, WAS and WDz**
- **Introduction to tool concepts**
- **Detail information and demonstration of ZOS development**

Host / Client Interaction



Perspectives for zOS Application Development

- What's a Perspective?
 - An arrangement of views (windows) and editors targeted for a particular task
- zOS Systems Perspective – Remote Systems Explorer (RSE)
 - View of datasets & members
 - TSO Command Processor, Job Monitor, Edit Window
- zOS Projects Perspective
 - IDEs organize work in projects
 - Project properties - compile options, link options, etc apply to artifacts in project
 - Remote projects – application artifacts exist on zOS
 - Local projects – application artifacts existing on the workstation
- EGL Perspectives
 - Works in conjunction with JSF
 - Generates COBOL/CICS or Java

z/OS System Perspective

The screenshot displays the IBM Rational Software Development Platform interface for a z/OS system. The main window shows a JCL file named ECHO.jcl with the following content:

```

Line 1      Column 1      Insert
//-----1-----2-----3-----4-----5-----+
//SAWDUST1 JOB ,
//MSGCLASS=H, TIME=(,4), REGION=28M, COND=(16,LT)
// *
//STP0000 EXEC PROC=ELAXFPL1,CICS=,
//DB2=,
//COMP=
//PLI.SYSLIN DD DSN=SAWDUST.PLI OBJ (ECHO) ,
//DISP=SHR
//PLI.SYSLIB DD DSN=SAWDUST.PLI INCLUDE,DISP=SHR
//DD DSN=CEEV2R10.SCEESAMP,DISP=SHR
//PLI.SYSXMSL DD DUMMY
//PLI.SYSIN DD DSN=SAWDUST.SOURCE.PLI (ECHO) ,DISP=SHR
// *
//***** ADDITIONAL JCL FOR COMPILE HERE *****
//LKED EXEC PROC=ELAXFLNK
//LINK.OBJ0000 DD DSN=SAWDUST.PLI OBJ (ECHO) ,DISP=SH
//LINK.SYSLIN DD *
//INCLUDE OBJ0000
  
```

Annotations on the screenshot include:

- Disconnected -vs- Connected:** Two red circles in the Remote Systems tree on the left indicate connection status for different nodes.
- MVS PDS members:** A purple arrow points to the expanded SAWDUST directory in the Remote Systems tree, which lists various members like SAWDUST.ASM.LISTING, SAWDUST.GENERATED.JCL, etc.
- Data set characteristics:** A purple arrow points to the Properties window at the bottom left, which shows details for the selected Remote Scratchpad.
- Configurable Editor:** A purple arrow points to the main code editor window displaying the JCL file.
- USS Command Shell:** A purple arrow points to the Remote System Details window, which shows a table of job filters for the SAWDUST subsystem.
- TSO Command Shell:** A purple arrow points to the Remote Shell window at the bottom right, which shows a command prompt.
- JES subsystem view:** A purple arrow points to the Job Filter table in the Remote System Details window.

Property	Value
Context	
Name	TSO-CTF...
Number of children	0
Status	Running
Type	Command Shell

Job Filter SAWDUST	
SAWDUST1:JOB00075	
SAWDUST1:JOB00073	

z/OS Project Perspective – remote project

The screenshot displays the IBM Rational Software Development Platform interface for a z/OS project named 'PrintApp.cbl'. The interface is divided into several panes:

- Remote System Explorer (Left):** Shows a tree view of project files. Annotations include:
 - Remote -vs- Local:** A red circle highlights the 'Remote07_cobol' folder, and a red arrow points to the 'ExampleLocalCobol_1' folder.
 - MVS Projects - able to work offline / online:** A purple arrow points to the 'PrintApp.cbl' file.
- Code Editor (Center):** Displays COBOL source code for 'PrintApp.cbl'. Annotations include:
 - Language sensitive Editor:** A blue arrow points to the 'Data Division' section.
 - System Navigator:** A blue arrow points to the 'MVS Files' folder in the Remote System Explorer.
- Remote Systems (Right):** Shows a tree view of remote systems and their connections.
- Properties (Bottom Left):** Shows the source outline for the selected file. A purple arrow points to the 'Procedure Division using Recvd-Parms.' section.
- z/OS File System Mapping (Bottom Center):** A table showing root connections to local and remote systems.

Name	Parent profile	Remote system type	Connection status	Host name
Local	sawdust	Local	Some subsystems connected	LOCALHOST
CTFMVS07	sawdust	z/OS	Some subsystems connected	CTFMVS07.RTP.RALE
CTFMVS08	sawdust	z/OS	No subsystems connected	CTFMVS08.RTP.RALE

Integrated Editor

- Language Sensitive editing (COBOL, PLI, JCL, etc)
- Code Assist for COBOL, PL/I, HLASM, JCL source
 - language construct completion
 - variable completion
- Open Copybook/Include/Macro
 - Name is resolved via standard search order.
- Both Local and Remote Syntax Check or Compiles / integration with task list
 - Similar to Java, click on task list entry, opens editor on source file
- User extensible via Java
- Used by the debug Perspective to set breakpoints, etc
- Outline view of source for ease of navigation

Remote System Explorer perspective

Select Perspective

- Data
- Debug
- Host Access Transformation Services
- J2EE (default)
- Java
- Java Browsing
- Java Type Hierarchy
- Remote System Explorer**
- Resource
- Service Flow Modeler
- Web
- z/OS Projects

Remote System Explorer - IBM Rational Software Development Platform

File Edit Navigate Search Project Run Window Help

Remote Systems Team z/OS File System Mapping

System: testmvs

Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page
**COBOL	cbl	text	IBM-037 (Inh...	CP1252 (Inhe...
**COBCOPY	cpy	text	IBM-037 (Inh...	CP1252 (Inhe...
**PLI	pli	text	IBM-037 (Inh...	CP1252 (Inhe...
**ASSEMBLE	asm	text	IBM-037 (Inh...	CP1252 (Inhe...
**OBJ	obj	binary	IBM-037 (Inh...	CP1252 (Inhe...
**LOAD	exe	binary	IBM-037 (Inh...	CP1252 (Inhe...
**CLIST	cmd	text	IBM-037 (Inh...	CP1252 (Inhe...
**JCL	jd	text	IBM-037 (Inh...	CP1252 (Inhe...
**SIGYCLST	cmd	text	IBM-037 (Inh...	CP1252 (Inhe...
**CNTL	jd	text	IBM-037 (Inh...	CP1252 (Inhe...
**JCL	jd	text	IBM-037 (Inh...	CP1252 (Inhe...

Remote System Details Tasks

Connection testmvs

Name	User ID	Port	Connected
JES	DNET017	0	Yes
USS Files	(Inherited)	0	No
USS Shells	(Inherited)	0	No
MVS Files	(Inherited)	0	No
TSO Commands	(Inherited)	0	No

Properties

Property	Value
Connecti...	Some subsystems connected
Default ...	(Inherited)
Description	Dallas TESTMVS system
Host name	TESTMVS.DEMOPKG.IBM.COM

Flexible Access to Remote Artifacts

The screenshot displays the IBM Rational Software Development Platform interface. The main window is titled "Remote System Explorer - IGYTSALE.cbl - IBM Rational Software Development Platform".

Remote System Explorer: The left pane shows a tree view of "Remote Systems". Under "Local", there are "Local Files" and "Local Shells". A callout "Files on workstation" points to the local file system. Under "Remote Systems", there is a system named "ctfmvs07.rtp.raleigh.ibm.com". Under this system, there are "USS Files" and "MVS Files". A callout "Files on remote z/OS" points to the remote system. Under "MVS Files", there are folders like "COBOL stuff", "CICS 3.1", and "Wilbert's STEW Stuff". A callout "COBOL and JCL" points to files like "COBSP.cbl", "COBTEST.cbl", and "IGYIVP.jcl".

IGYTSALE.cbl Editor: The main editor shows the source code of the program. The code includes comments and program identification information:

```

Line 1      Column 1      Insert
-----*A-1-B-----2-----3-----4-----
000001      Cbl_noadv,lib,map,nonumber,quote,sequence
000002      ISO100 Title "IGYTSALE * Main Program".
000003      Identification Division.
000004
000005      ISO120 Program-id.      IGYTSALE.
000006
000007      Author.      A. Programmer.
000008      Installation. IBM - Santa Teresa Laborat
000009      Date-written. April 1991.
000010      Date-compiled.
  
```

z/OS File System Mapping: The bottom right pane shows the "Remote System Details" and "z/OS File System Mapping" tab. The system is set to "ctfmvs07.rtp.raleigh.ibm.com". The mapping table is as follows:

Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page
**LOAD	exe	binary	IBM-037 (Inherited)
**CLIST	cmd	text	IBM-037 (Inherited)
**JCL	jcl	text	IBM-037 (Inherited)
**SIGYCLST	cmd	text	IBM-037 (Inherited)
**CNTL	jcl	text	IBM-037 (Inherited)
**LISTING	lst	text	IBM-037 (Inherited)
**OUTLIST	out	text	IBM-037 (Inherited)
**OBJS	obj	binary	IBM-037 (Inherited)
**INCLUDE	inc	text	IBM-037 (Inherited)
**MACRO	mac	text	IBM-037 (Inherited)
**COPYLIB	cpy	text	IBM-037 (Inherited)
**XML	xml	text	IBM-037 (Inherited)
**BMS	bms	text	IBM-037 (Inherited)
**JCLLIB	jcl	text	IBM-037 (Inherited)
**FILES			IBM-037 (Inherited)
COB**	cbl	text	IBM-037 (Inherited)
JCL**	jcl	text	IBM-037 (Inherited)
**JOB	jcl	text	IBM-037 (Inherited)
PLI**	pli	text	IBM-037 (Inherited)

A callout "Resource (e.g., member) mapping" points to the mapping table.

Properties: The bottom left pane shows the "Properties" of the selected file, with the following values:

Property	Value
Attribute	
BLKSIZE	32720
DSNTYPE	DATA_LIBRARY
DSORG	PO
EXTENTS	1
LRECL	80
PRIMARY	301

COBOL and PL/I Content Assist

```

*ACTDDRV.cbl X
Row 100 Column 12 1 change.
-----+*A-1-E-----2-----3-----4-----
000085 * ** New Business Program XML Interfac
000086 * *****
000087 * XML Stream Byte Length
000088 * XML Stream
000089 * 1 DFHCOMMAREA.
000090 1 a-xml-interface.
000091 2 a-xml-int-len pic 9(9) binary.
000092 2 a-xml-int-txt pic x(32768).
000093 * Procedure Division using DFHCOMMAREA
000094 Procedure Division using a-xml-interf.
000095 Mainline Section.
000096 * +-----+
000097 * | Enable Exception Handler |
000098 * +-----+
000099 perform a-register-exception-hand
000100
000101
000102
000103
000104
000105

```

- ABC DIVIDE - NOT ON SIZE ERROR - END-DIVIDE
- ABC DIVIDE - ON SIZE ERROR - END-DIVIDE
- ABC DIVIDE - ON SIZE ERROR - NOT ON SIZE ERROR - END
- ABC EJECT.
- ABC ENTRY
- ABC EVALUATE - WHEN - END-EVALUATE
- ABC EVALUATE - WHEN - WHEN OTHER - END-EVALUATE

```

* procedure DIVISION using DFHCOMMAREA.
Procedure Division using a-xml-interface.
Mainline Section.
* +-----+
* | Enable Exception Handler |
* +-----+
perform a-register-exception-handler
MOVE

```

- 010 a-converter-return-code
- 010 a-error-code
- 010 a-error-description
- 010 a-error-message-number
- 010 a-exception-occurred
- 010 a-failure-data
- 010 a-failure-message-number
- 010 a-failure-response

Benefit: Developers complete code more accurately and efficiently.

JCL Generation and Submission

The screenshot shows the IBM Rational Software Development Platform interface for a z/OS project named 'IGYIVPCL.jcl'. The project browser on the left shows a tree structure with 'DNET017.STEW.JCL(IGYIVPCL).jcl' highlighted in red. A blue arrow points to this file with the text 'JCL generated'. The main editor displays the following JCL code:

```

Line 1      Column 1      Insert
000001 //WKIVP JOB ,
000002 //  MSGCLASS=H,MSGLEVEL=(1,1),TIME=(,4),REGION=28M,COND=(16,LT)
000003 //*
000004 //STP0000 EXEC PROC=ELAXFCOC,CICS=,
000005 //  DB2=,
000006 //  COMP=
000007 //COBOL.SYSPRINT DD DSN=DNET017.STEW.LISTING(IGYIVP),
000008 //          DISP=SHR
000009 //COBOL.SYSLIN DD DSN=DNET017.STEW.OBJS(IGYIVP),
000010 //          DISP=SHR
000011 //COBOL.SYSLIB DD DSN=DNET017.STEW.COPYLIB,DISP=SHR
000012 //COBOL.SYSXMLSD DD DUMMY
  
```

The context menu on the right includes options like Cut, Copy, Paste, Select, Selected, Deselect, Filter view, Show all, Source View, Add to Snippets..., Save, Add Breakpoint, and Submit. The 'Submit' button is highlighted with a mouse cursor.

Benefit: Developers focused on business logic and not on writing JCL

Monitoring Job Output / Issuing Commands

JES2 JOB LOG -- SYSTEM MVSA -- NO

```

14.07.02 JOB07143 ---- THURSDAY, 16 JUN 2005 ----
14.07.02 JOB07143 IRR010I USERID DNET017 IS ASSIGNED TO THIS JOB.
14.07.02 JOB07143 ICH70001I DNET017 LAST ACCESS AT 13:53:30 ON THURSDAY, JU
14.07.02 JOB07143 $HASP373 WKIVP STARTED - INIT 5 - CLASS A - SYS MVSA
14.07.03 JOB07143 IEF403I WKIVP - STARTED - TIME=14.07.03
14.07.05 JOB07143 - --TIMINGS (MINS.
14.07.05 JOB07143 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CL
14.07.05 JOB07143 -WKIVP STP0000 COBOL 00 792 .00 .00
14.07.05 JOB07143 IEF404I WKIVP - ENDED - TIME=14.07.05
14.07.05 JOB07143 -WKIVP ENDED. NAME- TOTAL CPU TIME
14.07.05 JOB07143 $HASP395 WKIVP ENDED
----- JES2 JOB STATISTICS -----
16 JUN 2005 JOB EXECUTION DATE
15 CARDS READ
211 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
13 SYSOUT SPOOL KBYTES
0.05 MINUTES EXECUTION TIME
1 //WKIVP JOB ,
// MSGCLASS=H,MSGLEVEL=(1,1),TIME=(,4),REGION=28M,COND=(16,LT)

```

Remote Shell Remote System Details Tasks z/OS File System Mapping

TSO-ctfmvs07.rtp.raleigh.ibm.com

Command Shell - Running

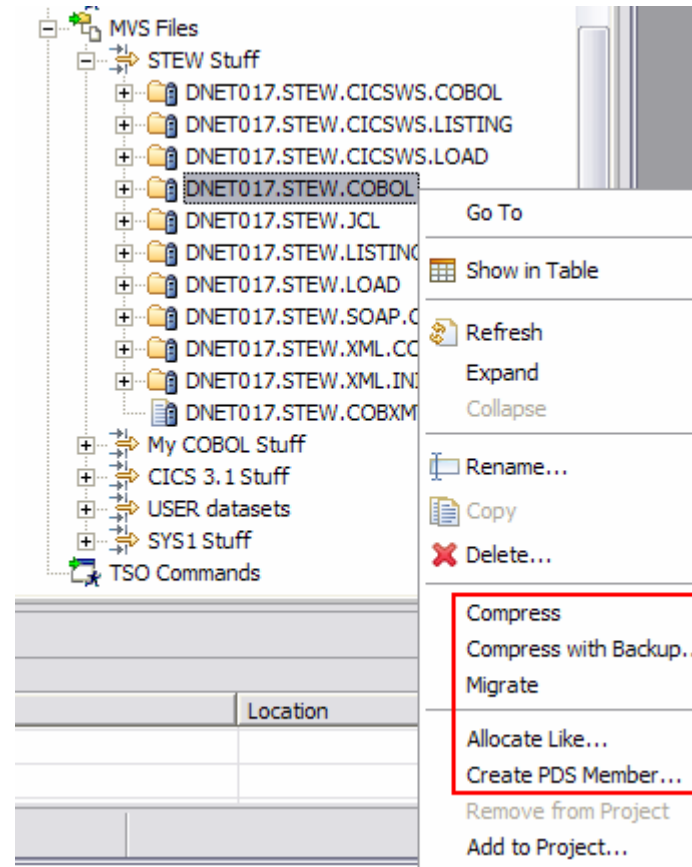
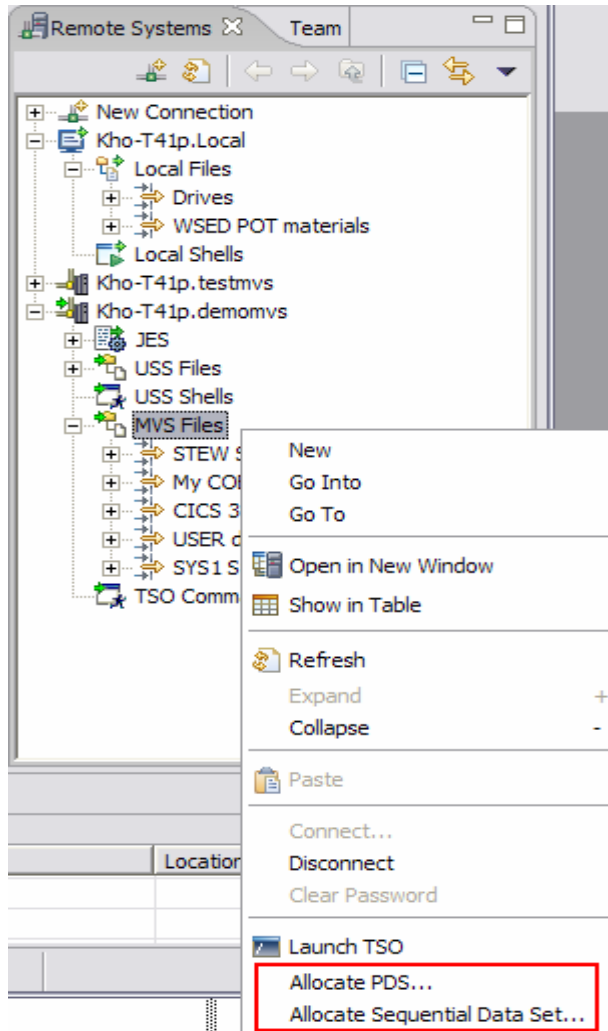
```

Specify a TSO command to run
>LISTALC
CUST.H001600.V6ROM0.SFEKSAMP
CUST.PDS.EXEC
WILBERT.FEKFRSRV.STC00146.D0000109.?
WILBERT.FEKFRSRV.STC00146.D0000110.?
NULLFILE
NULLFILE

```

Benefit: Developers do not have to continually switch between systems

z/OS Dataset Management



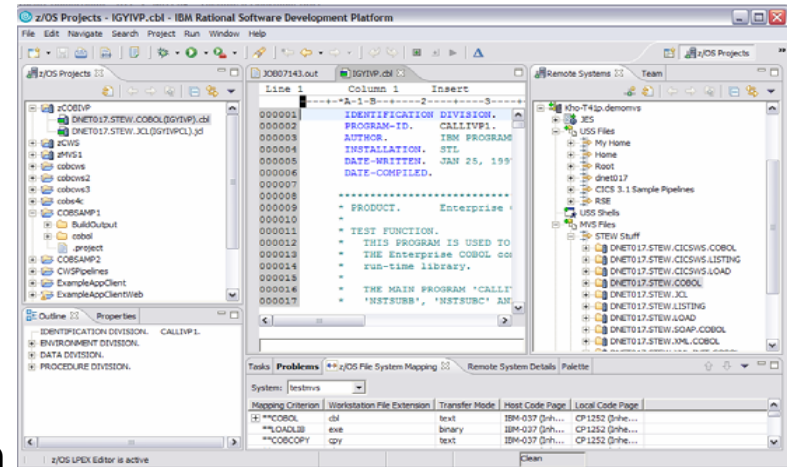
COBOL and PL/I Remote Edit/Compile/Debug

- Comprehensive state-of-the-art facilities for editing, build and debugging existing or new COBOL or PL/I programs
- Remote edit/compile/debug: keep z/OS artifacts on the mainframe and avoid costly downloads/uploads

- ✓ Powerful smart editing
- ✓ Graphical navigation
- ✓ Syntax check
- ✓ Control of remote compile
- ✓ Compile feedback
- ✓ Graphical debugger on workstation

- Program executes on mainframe (CICS, IMS, Batch, Stored Procedures)

- Exploits IBM Software Development Platform
 - ✓ Task manager, Projects/Perspectives , etc
- Live host connectivity (TSO Commands, Job queue mgmt, etc)



Using Enterprise COBOL to service-enable z/OS

CICS/IMS/Batch/DB2 COBOL

What's new ...

- XML Language based generation from COBOL data structure
 - XMLGenerate Verb
- WebSphere EJB support
- DB2 V8 preprocessor
- CICS preprocessor

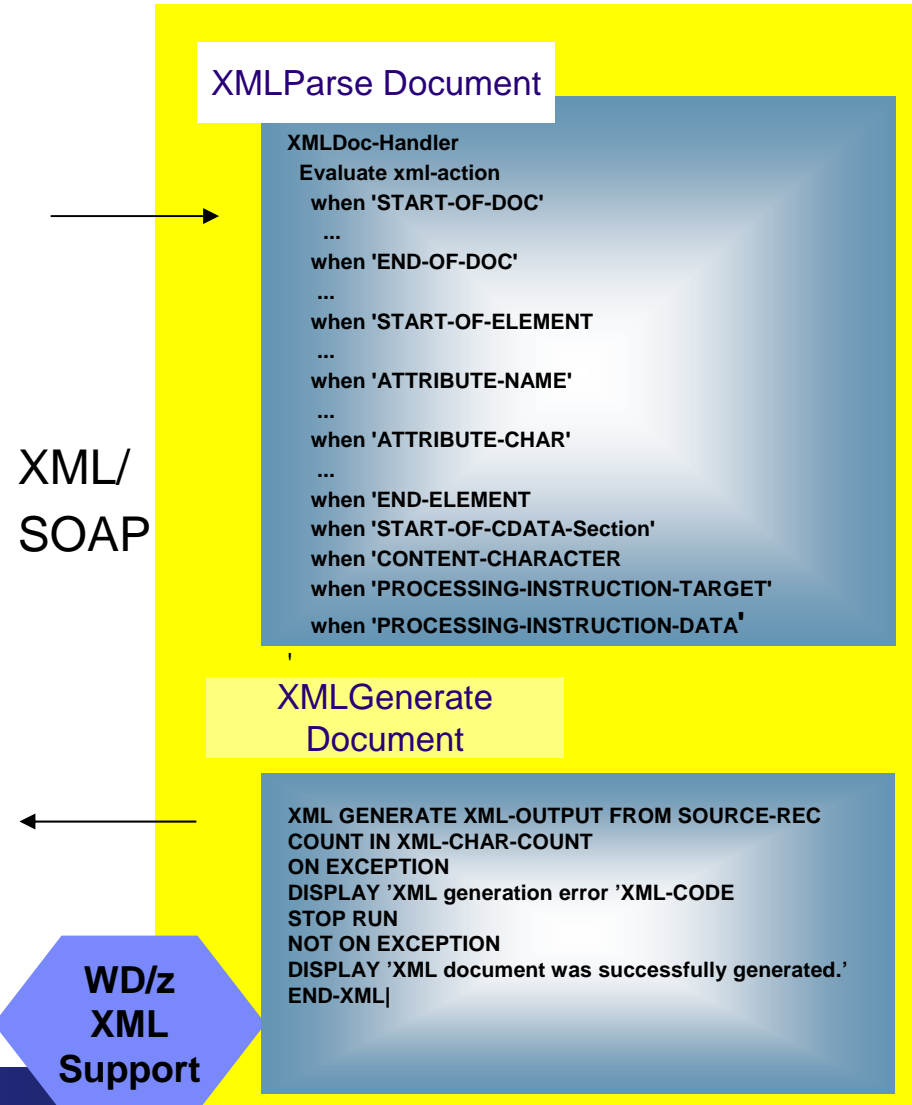
High speed XML Sax based parsing

- XMLParse
- XMLGenerate
- Related verbs

Object Oriented Support for Java COBOL Interoperability

Unicode support

Similar XML parsing support available in Enterprise PL/I



WDz SOA or ESB Lite tools

- **Enable Web Services and XML access to existing CICS and IMS transactions**

- **XML and Web Services for the Enterprise (XSE)**
- Quickly maps existing COBOL interfaces to XML and Web Services.
- No code changes for the COBOL application
- Supports IMS, CICS BMS (terminal-based) & CICS commarea applications

- **Model and deploy complex CICS processes to support SOA**

- **Service Flow Modeler (SFM)**
- Aggregate CICS transactions into high-level business processes through visual (drag n drop) modeling
- Highly optimized CICS COBOL runtime to increase overall throughput
- Supports COBOL commarea-based applications and terminal-based applications

- **Recent Announcement of Note: CICS Service Flow Feature**

- *0 Cost feature*
- *Design Web Services / XML flows in WDz*
- *Consists of:*
 - *CICS Service Flow Runtime*
 - *Limited WDz licensed for:*
 - **Service Flow Modeler**
 - **XMLE and Web ServicesXMLE**
- *Positioned for all CICS V3 customers needing to simply or programmatically integrate via Web Services*

XML Services for the Enterprise (XSE)

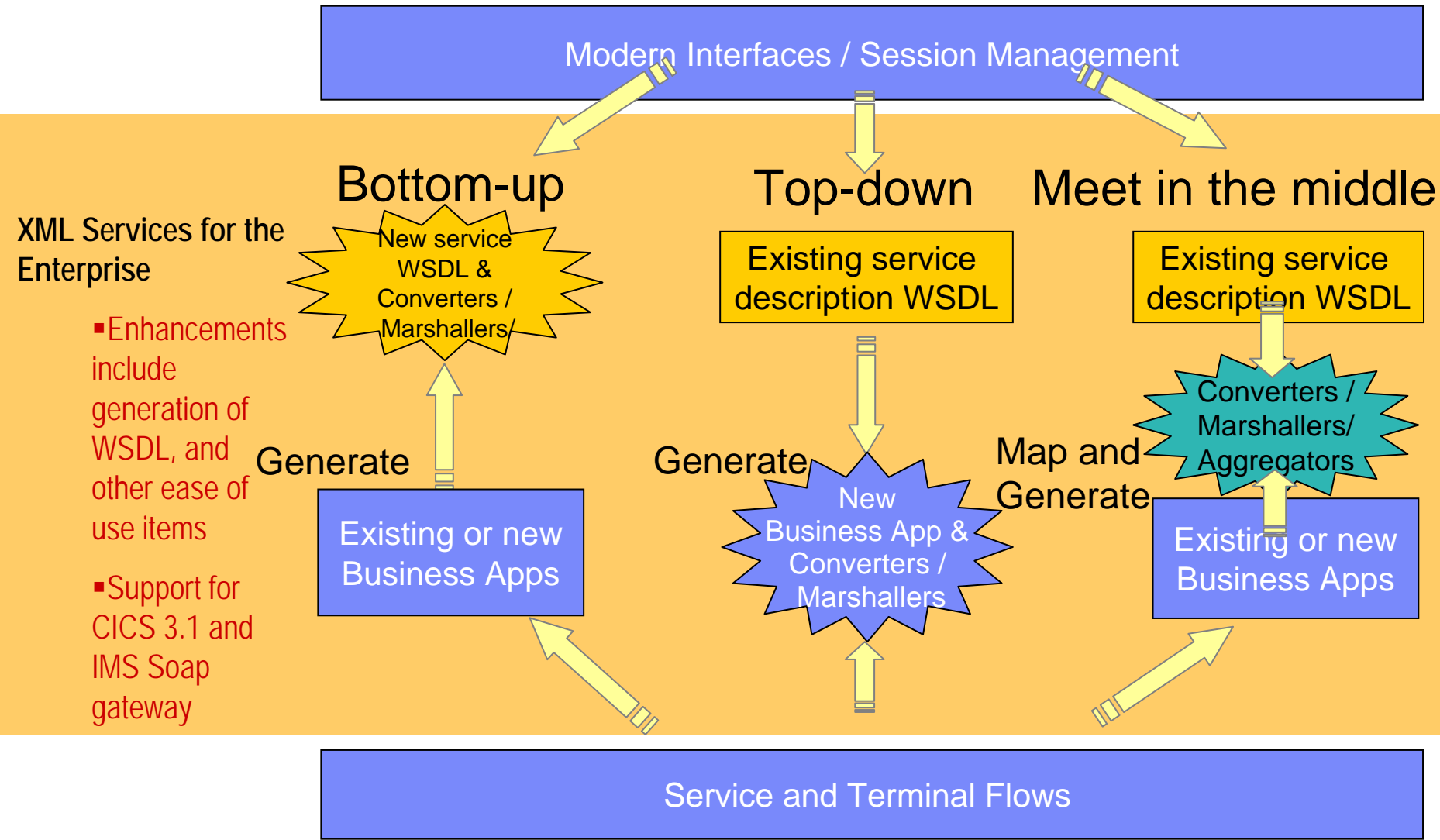
- Provide tools to adapt COBOL based applications
 - Process and produce XML messages
 - Web Services
 - Participate in a larger system that uses XML
- Web Services Enablement wizard
 - Generate Web Service interface from existing COBOL application
 - Bottom-up approach since COBOL at the bottom (base) of the creation process
- XML to COBOL Mapping tool
 - Map existing Web Service interface or XML to existing COBOL app.
 - Meet-in-the-middle since Web Services/XML definition “meets” or maps to the existing COBOL interface
- Batch processor
 - Runs unattended or in batch mode using the bottom-up approach

XML and Web Services Enablement

Enables COBOL-based applications to consume and produce XML messages

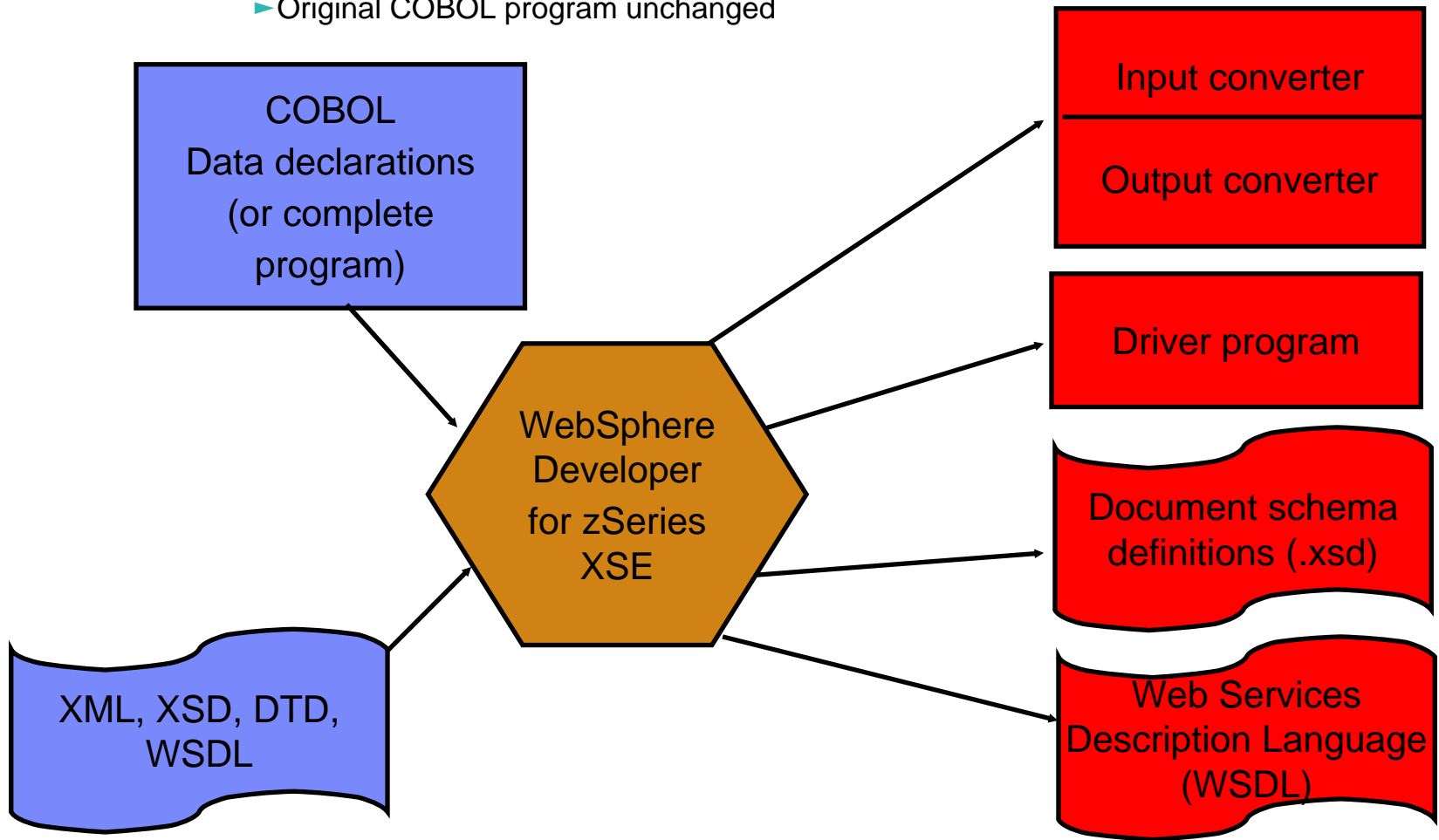
- Leverages XML parsing capabilities of IBM Enterprise COBOL V3.1
- Creates COBOL converter programs
 - ▶ Inbound to convert XML messages into native COBOL data
 - ▶ Outbound to convert native COBOL data into XML messages
- Creates COBOL driver program
 - ▶ Illustrate the invocation of converters
 - ▶ Illustrate the interaction with existing application
- Creates WSDL that describes COBOL based service
- Enables communication with XML based systems
- Batch interface to Web Services Enablement Tool for COBOL
 - Create COBOL adapters and WSDL via command line

Web Service Enablement Styles (XSE)



Mapping COBOL Data

- Enables COBOL-based applications to consume and produce XML messages
 - ▶ Original COBOL program unchanged

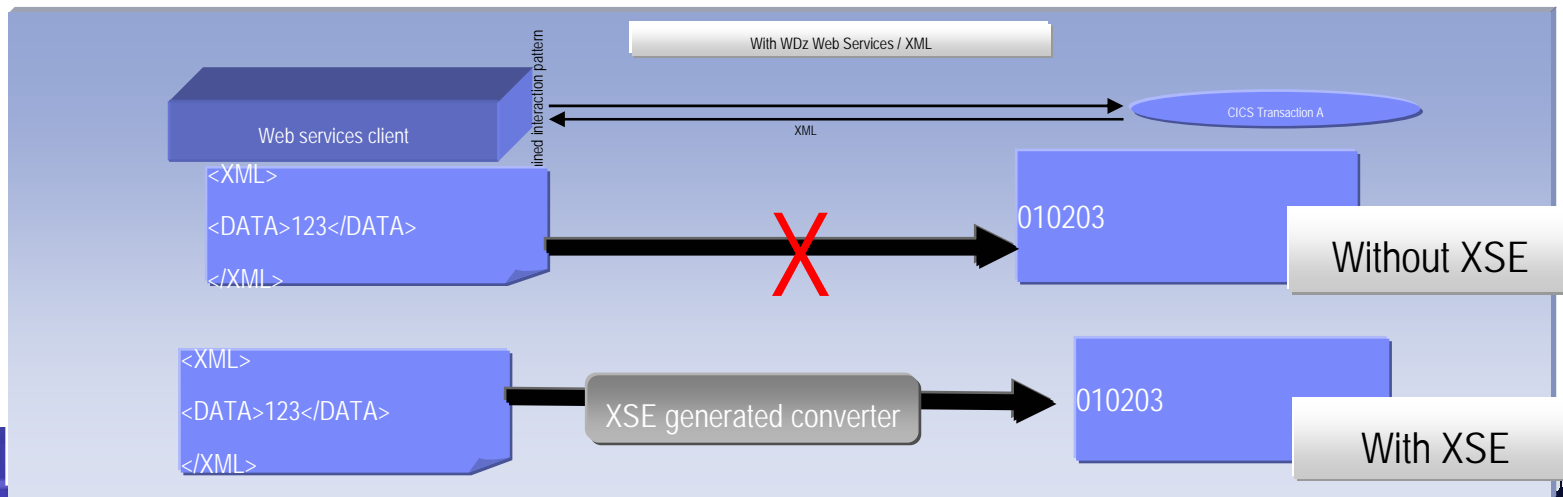
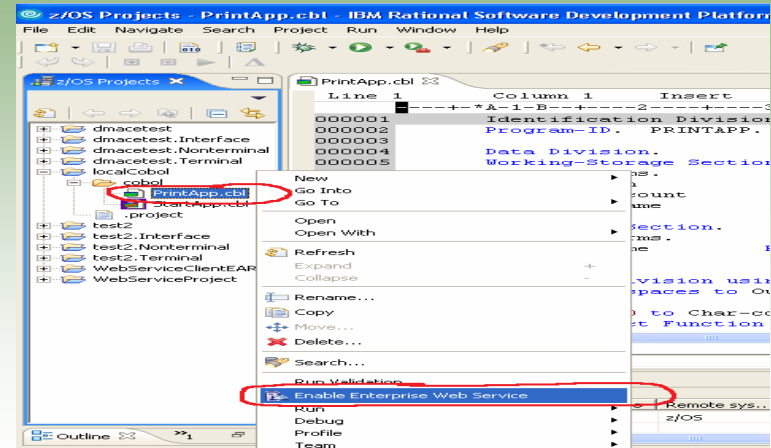


What is XML Services for the Enterprise (XSE)

XSE in WebSphere Developer for zSeries

Most rapid building of Web services from existing CICS applications

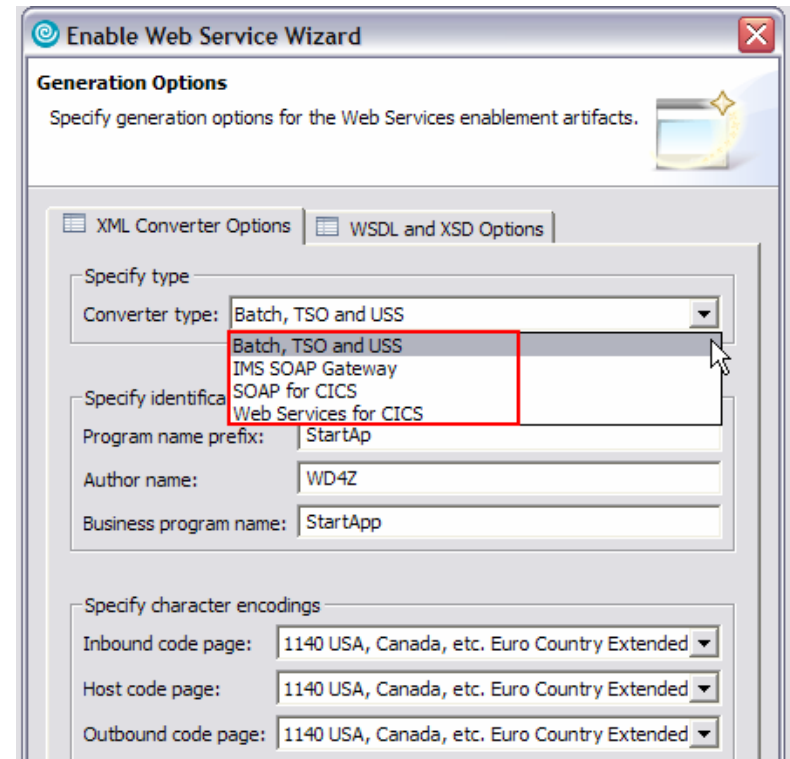
- Single CICS and IMS transactions enabled for Web Services
- Supports IMS Message Queue, CICS Commarea and new Channels/Container based applications
- Rapid generation of WSDL, CICS WSBind, and Adapter generation eliminating complex hand coding of XML to/from language conversions
- Includes complete Web Services Test and Java generation environment



Converter Types Supported

- Batch, TSO and USS
- IMS SOAP Gateway
- SOAP for CICS
- Web Services for CICS

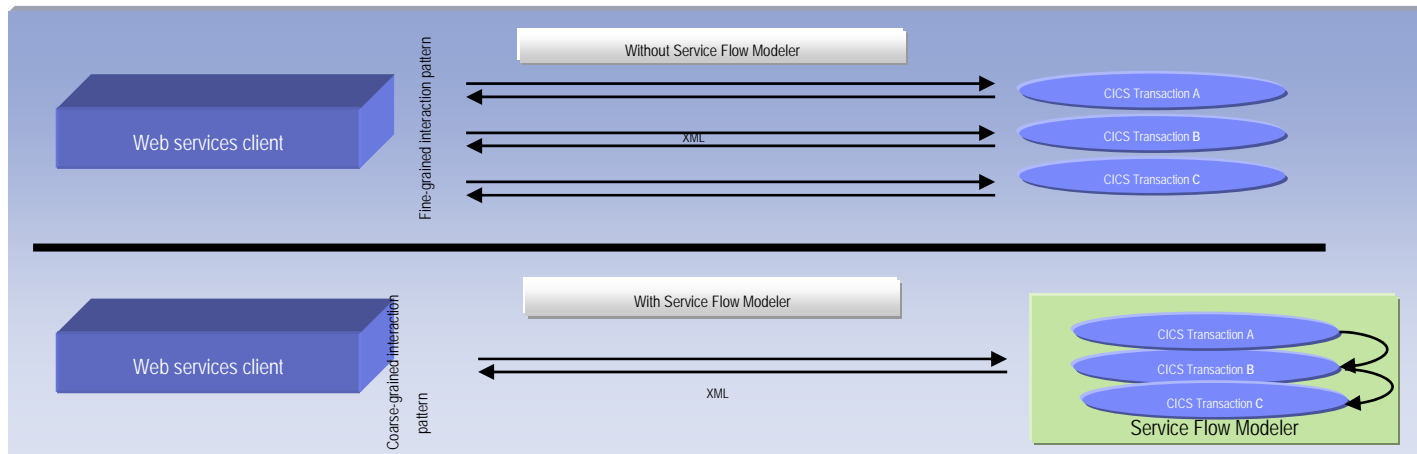
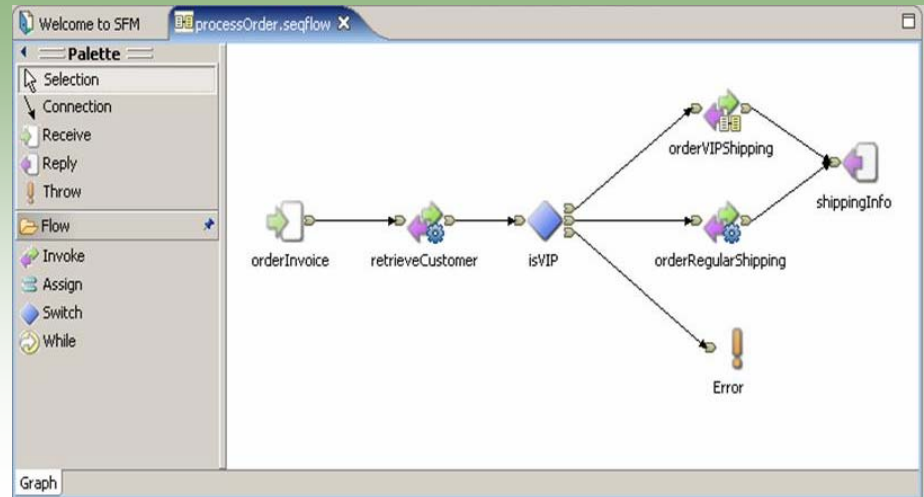
← *New!*



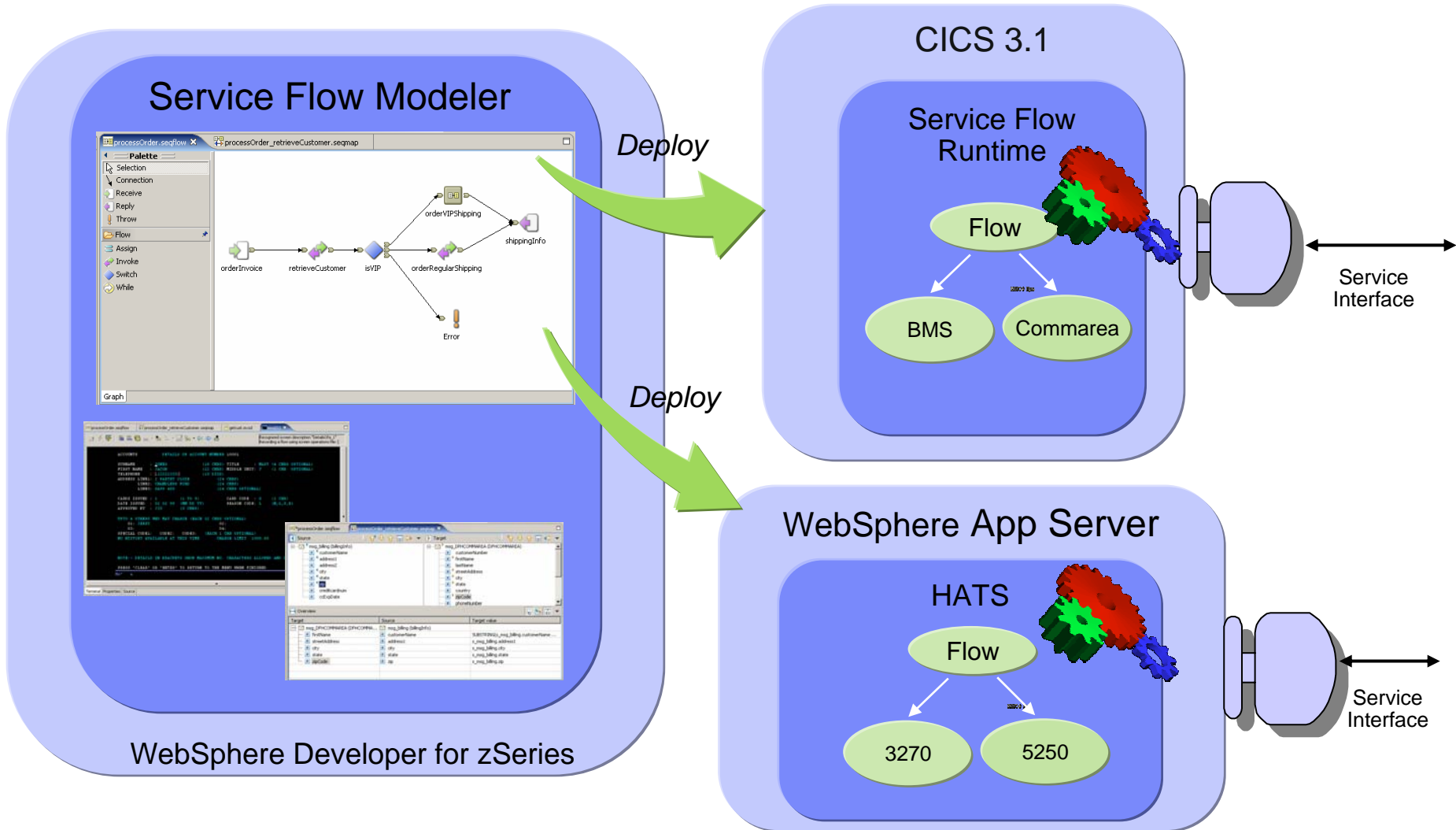
What is Service Flow Modeler?

New Feature! Service Flow Modeler in WebSphere Developer for zSeries

- **Builds Web services from existing CICS applications**
 - Aggregates multiple CICS transactions into high-level business processes through visual modeling
 - Supports CICS BMS (terminal-based) applications & CICS commarea applications
 - Highly optimized CICS runtime supporting Web services and XML interfaces



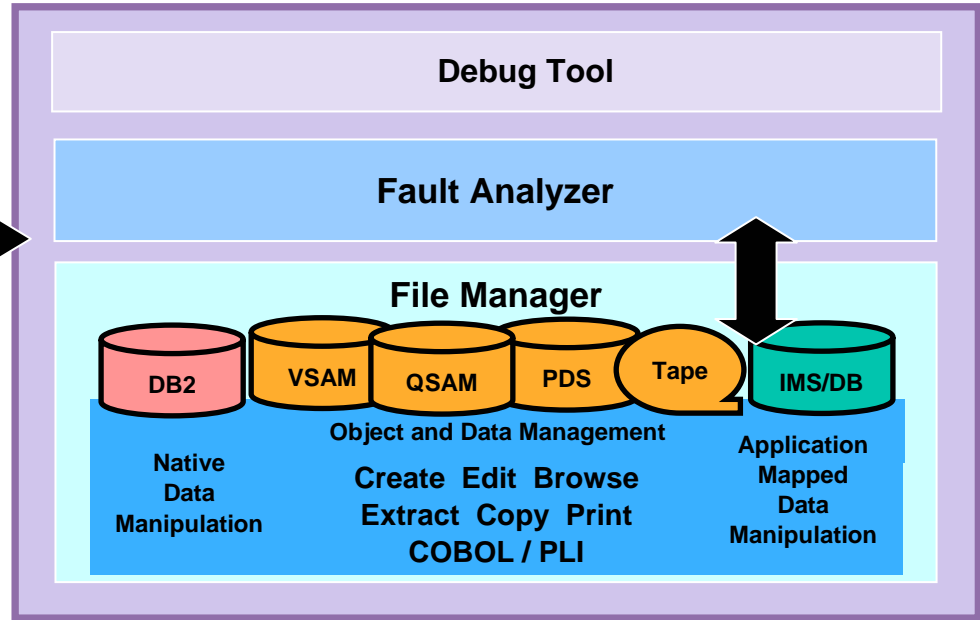
WDz's Service Flow Modeler Deployment Options



Test and Problem Determination

Integration speeds time to market
 Pervasive data access speeds time to market

WS Enterprise Developer



Benefits:

- Simplify development of zSeries test cases
 - ▶ Data creation for DB2, IMS/DB, VSAM, and QSAM
 - ▶ Extract and load
- Reduced deployment complexity
 - ▶ Production data validation and creation
- Common environment
 - ▶ Reuse of skills across e-bus and traditional applications

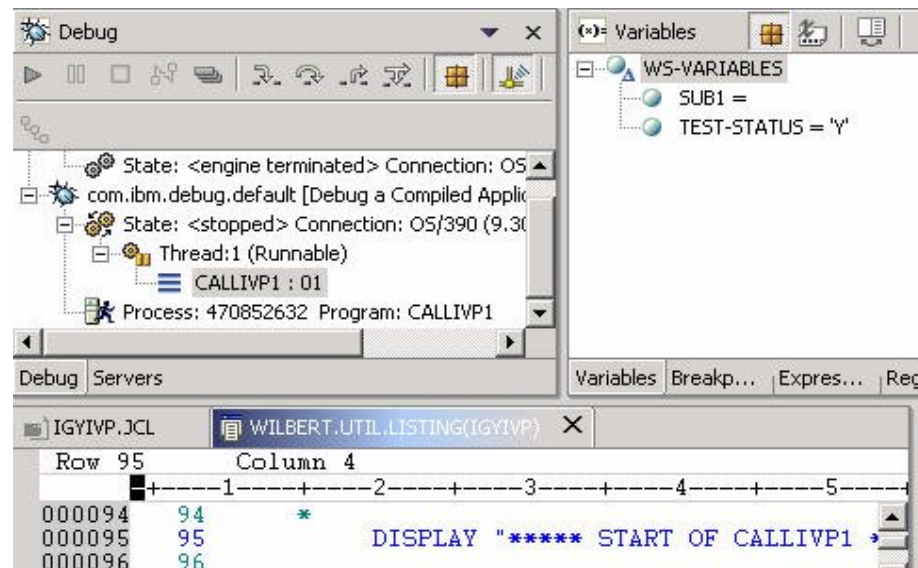
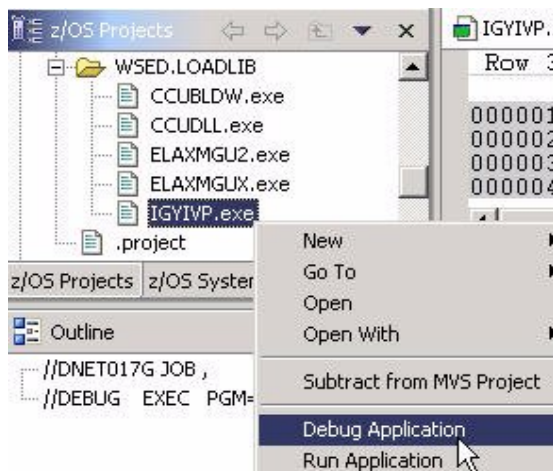
Systems Support Manager

Systems Programmer

Data Center Operator

WebSphere Developer based Debugging

Same Debug Perspective



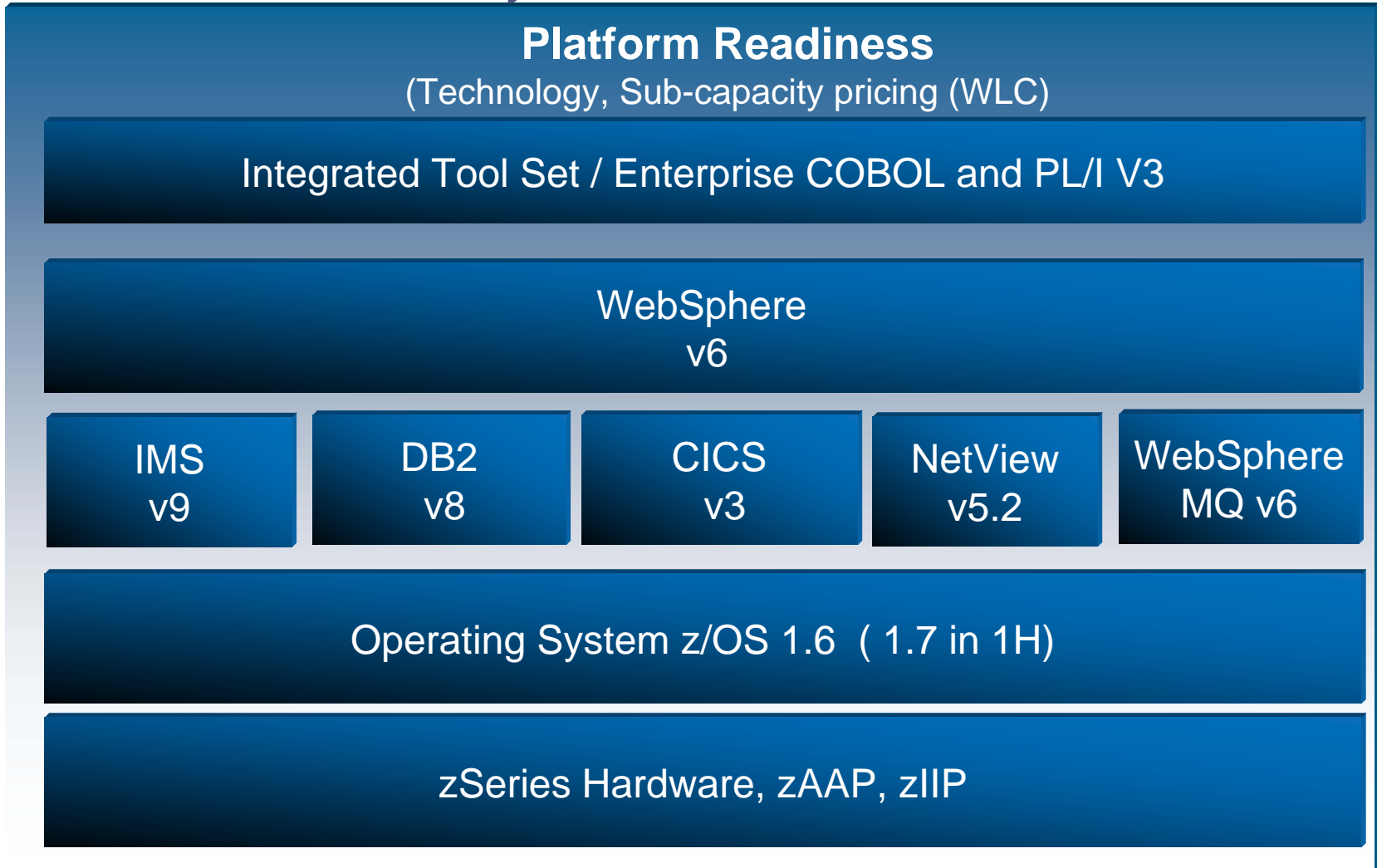
Benefit: Consistent debugging environment for COBOL, PL/I, Java

Gartner: Best Practices for Mainframe SOA

- **Act tactical, think strategic**
- **Evaluate tools that provide good microflow orchestration**
- **Create services that utilize function from across existing application boundaries.**
- **Build a reuse culture and technology infrastructure.**
- **Work with operations to create management/performance-monitoring support.**
- **Use code understanding/inventory/restructuring tools to improve service granularity.**
- **Define the role of the mainframe in future application architecture.**

IBM zSeries Software Solutions

Platform Readiness is Key



End Game: A Single Point of Access for People and Projects: Moving to The Developer Dashboard

Simplify organizational management in mixed workload and distributed environments

