



IBM Software Group

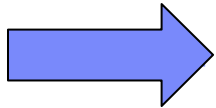
# Enterprise COBOL and Rational Developer for System z – How they play in an SOA world

Michelle A. Cordes  
Rational System z Ecosystem Team  
[mcordes@us.ibm.com](mailto:mcordes@us.ibm.com)

**Rational** software



# Agenda

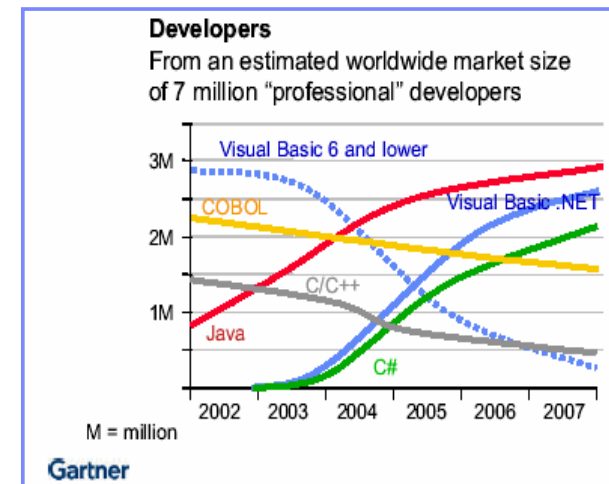


- Introduction
  - Enterprise COBOL
    - ▶ XML Support
    - ▶ CICS V3.x Support
    - ▶ Unicode Support
    - ▶ Object Oriented COBOL
  - Rational Developer for System z
    - ▶ Mainframe development features
    - ▶ XML and Web Services support
    - ▶ CICS V3.x and Service Flow Feature support
  - More Information
  - Questions?

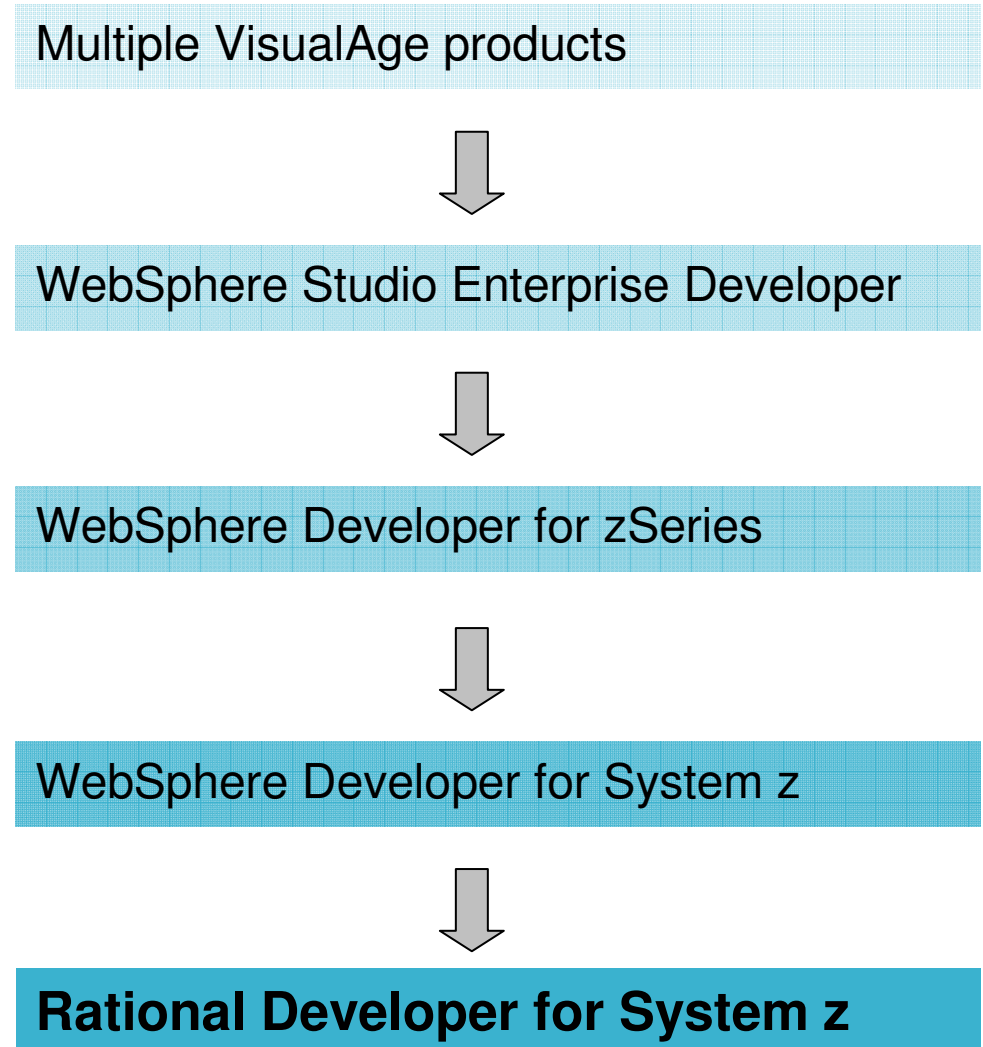
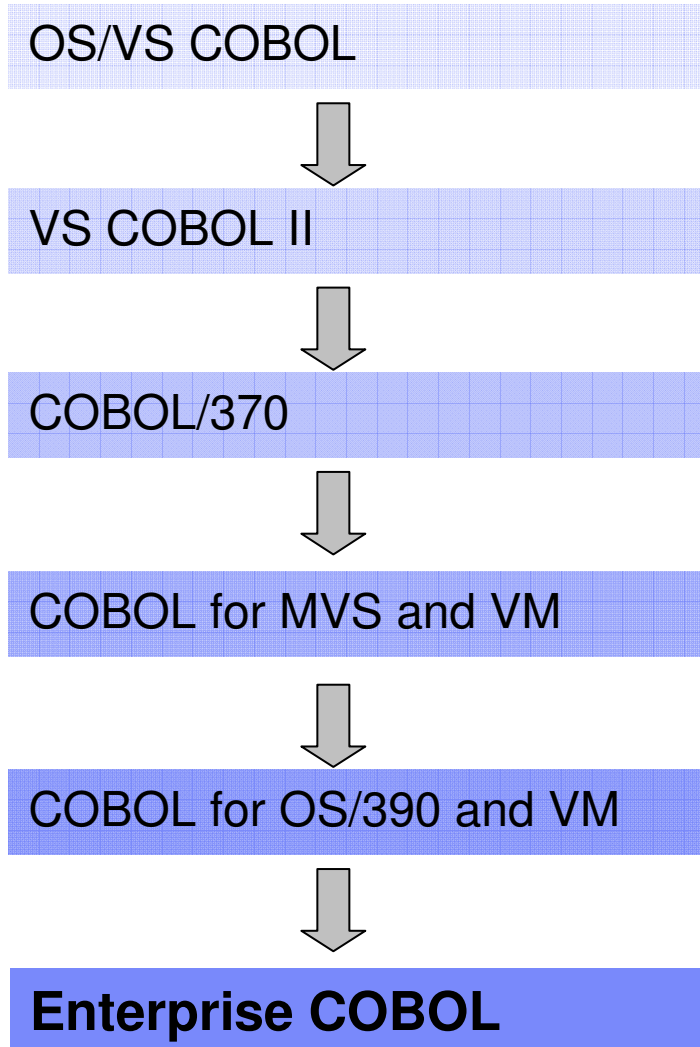


## Legacy applications


- Significant business intelligence exists in core systems
  - ▶ "200 Billion lines of COBOL code in existence" *eWeek*
  - ▶ "5 Billion lines of COBOL code added yearly" *Bill Ulrich, TSG Inc.*
  - ▶ "2 Million COBOL developers" *Gartner*
  - ▶ "Majority of customer data still on mainframes" *Computerworld*
  - ▶ "Replacement costs \$20 Trillion" *eWeek*
  
- Rewriting - is it an option.....
  - ▶ How long will it take? (lose strategic benefit)
  - ▶ Who will do it? (who has the business knowledge?)
  - ▶ How much will it cost?
  - ▶ Risk?



# Product History



# Agenda

- Introduction
- ▶  Enterprise COBOL
  - ▶ XML Support
  - ▶ CICS V3.x Support
  - ▶ Unicode Support
  - ▶ Object Oriented COBOL
- Rational Developer for System z
  - ▶ Mainframe development features
  - ▶ XML and Web Services Support
  - ▶ CICS V3.x and Service Flow Feature support
- More Information
- Questions?



# Introduction to XML

- What is XML?
  - ▶ A markup language, for describing data (rather than its presentation)
  - ▶ Each piece of data is identified via the markup language
  - ▶ Unlimited number of tags can be defined
- Why XML?
  - ▶ It is becoming the interconnection layer of e-business
  - ▶ The industry direction for application integration and platform independent data interchange
    - e.g., for Web Services
  - ▶ Allows sender and receiver to evolve independently of each other (flexible interface)
    - as opposed to Electronic Data Interchange (EDI) for example

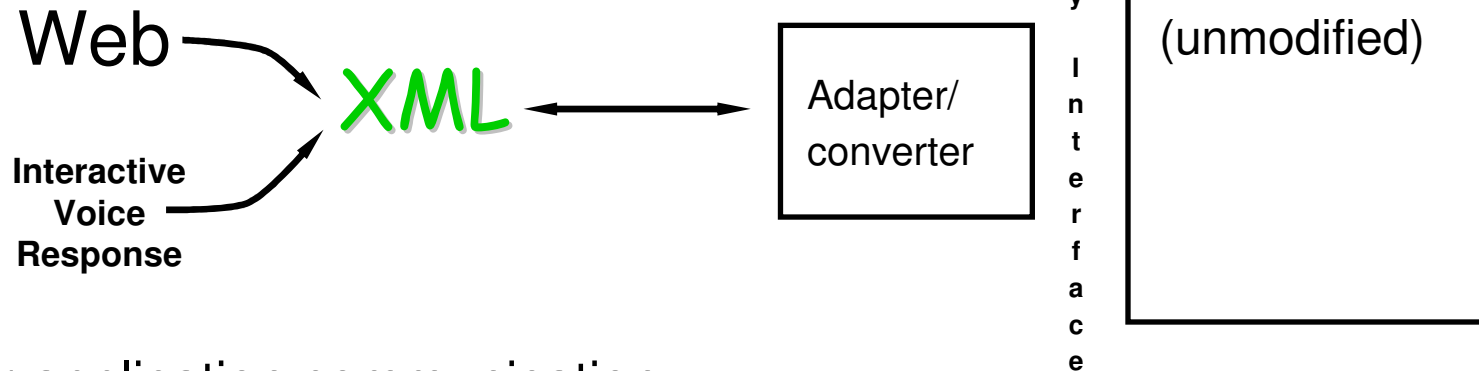
## Sample XML document:

```
<?xml version="1.0" encoding="ibm-1140"?>
<TRADE type="short sale">
  <SYMBOL>IBM</SYMBOL>
  <PRICE>$78.75</PRICE>
  <SHARES>200</SHARES>
  <COMMISSION>$29.95</COMMISSION>
</TRADE>
```



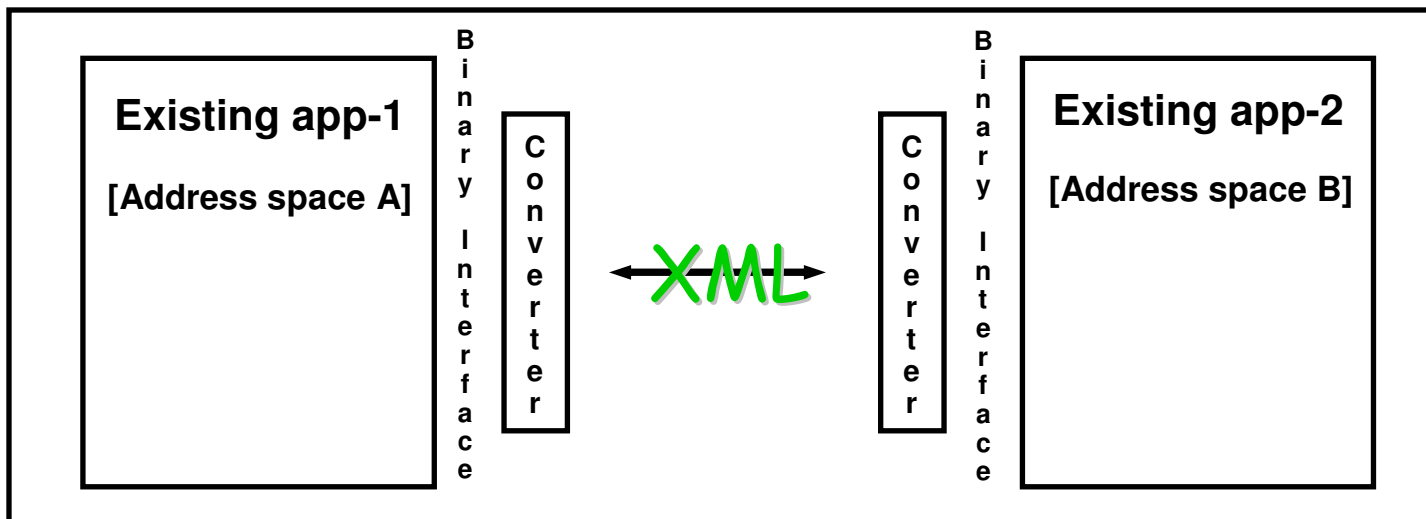
## Customer scenarios/problems

- Refacing existing applications/transactions



- Inter-application communication

Single Enterprise system/complex



## Why process XML in ...

### In applications?

- Coherent development context and methodology
- Centralizes business logic within the application
  - ▶ Versus some business function here, some there, ...
- Independent of middleware choices, characteristics
- Allows business logic to be conveniently applied during and after message acquisition/generation
- Incremental step from existing application design
- Can process XML messages as such
  - ▶ Versus forcing conversion to traditional data structures

### In COBOL?

- Keep development control in one place/style
- Guarantee correct language semantics
  - ▶ sign configuration
  - ▶ layout/padding
  - ▶ picture constraints
- Exploits your existing assets/skills/literacy





# Enterprise COBOL XML Parser Support

- Much faster than general purpose parsers
  - ▶ Designed for high-speed transaction processing
- Runs in all COBOL run-time environments:
  - ▶ CICS, IMS, batch, TSO, USS, ...
- Works with any transport mechanism for XML documents
  - ▶ Use MQSeries, CICS transient queue or COMMAREA, IMS message processing queue, WebSphere, etc.
- XML Parser is part of the run-time library
  - ▶ Can be used from Enterprise COBOL or Enterprise PL/I
- Supports both inbound and outbound XML documents
- Parses XML documents that are in memory, in a COBOL alphanumeric or national data item
- Parses XML documents into individual pieces
  - ▶ Passes each piece to user-written processing procedure
- During parsing you can populate COBOL data structures with the data from XML messages
  - ▶ Advantage: non-COBOL programs can communicate data to/from COBOL without having to know the COBOL data structure formats!



# Hello XML World

Program

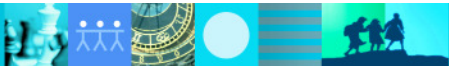
```

Identification division.
Program-id. HelloXML.
Data division.
Working-storage section.
  1 M.
    2 pic x(21) value '<?xml version="1.0"?>'.
    2 pic x(40) value '<msg type="succinct">Hello, World!</msg>'.
Procedure division.
  Display 'XML Event          XML Text'
  XML Parse M
  Processing procedure P
  End-XML
  Goback.
P.
  If XML-Code = 0
    Display XML-Event XML-Text
  End-if.
End program HelloXML.

```

Output

XML Event	XML Text
START-OF-DOCUMENT	<?xml version="1.0"?><msg type="succinct">Hello, World!</msg>
VERSION-INFORMATION	1.0
START-OF-ELEMENT	msg
ATTRIBUTE-NAME	type
ATTRIBUTE-CHARACTERS	succinct
CONTENT-CHARACTERS	Hello, World!
END-OF-ELEMENT	msg
END-OF-DOCUMENT	



# XML Generate

## XML GENERATE statement

- Generates XML message from COBOL group data items

```

1 Employee1.
  2 Name pic X(5) Value 'Tom'.
  2 Idn  pic 9(9) comp Value 123456789.
  2 Addr.
    3 Street pic X(20) Value '555 Bailey
Ave'.
    3 City   pic X(20) Value 'San Jose'.
    3 State  pic X(20) Value 'California'.
  2 More.
    3 Age    pic +99.99 Value '45.9'.
    3 Firm   pic BBXXX9B Value 'IBM4'.
    3 Salary COMP-2      Value
+.00012327E+06.
1 XMLDOCUMENT pic X(500).

```

Procedure division.

```
XML GENERATE XMLDOCUMENT FROM EMPLOYEE1
```

- Output from sample XML GENERATE statement

```

<Employee1>
  <Name>Tom</Name>
  <Idn>123456789</Idn>
  <Addr>
    <Street>555 Bailey Ave</Street>
    <City>San Jose</City>
    <State>California</State>
  </Addr>
  <More>
    <Age>45.9</Age>
    <Firm>IBM4</Firm>

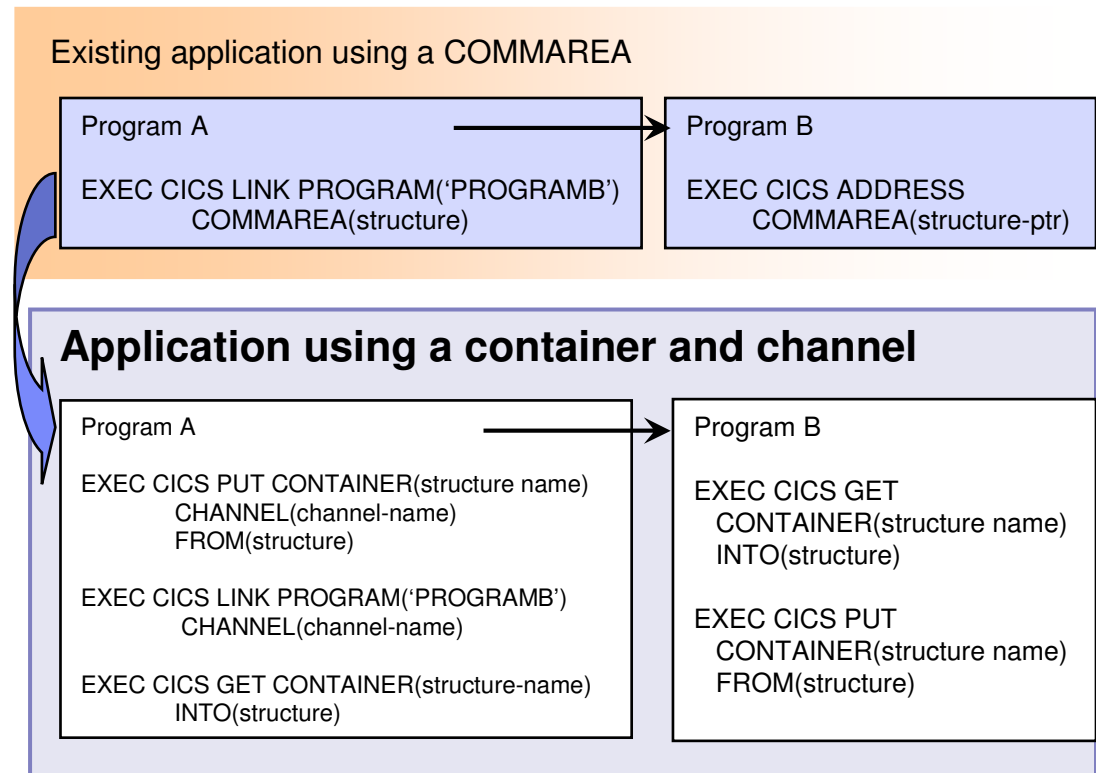
    <Salary>1.2327000000000000E+02</Salary>
  </More>
</Employee1>

```



## Optimized data exchange between CICS programs with Containers and Channels

- Offers a more flexible and intuitive alternative to the COMMAREA
  - ▶ By using separate containers for logically different data it will simplify language structures and minimize the impact of changes to the interface
    - For example; input, output, error
    - Avoids “overloading”
  - ▶ Dynamic creation and discovery by applications
- Enables large amounts of data to be passed between CICS applications
  - ▶ Not subject to 32KB restriction
- Optimized and managed by CICS
- Requires minimal application changes required to use



## What is Unicode and why should you care that Enterprise COBOL supports it?

- Industry standard for coded character set
  - ▶ defined by Unicode Consortium and ISO
- Covers all commonly used characters in the world in one code page (vs. one "language" per ASCII, EBCDIC, or EUC code page)
- Characters: text, digits, special characters, symbols, control characters, ...
- Multiple Unicode encoding formats: UTF-8, UTF-16, UTF-32
- "Stateless" encoding: meaning of an encoding unit is self defining
- Support global e-business environment:
  - ▶ Applications for multi-cultural/multi-geographic businesses
  - ▶ Networks of heterogeneous systems
- Enables a common implementation for global application versus separate code page for each geographic area or system platform
- Supported by all key operating system and middleware platforms
- Required by: XML, HTML, Java, ...
- Interoperates with DB2 Unicode, Java, COBOL XML



# Unicode Support Overview

- Unicode literal and value clause
- Unicode data type
- New compiler options
- Implicit conversions for EBCDIC data assigned to or compared with Unicode data
- Explicit conversions via intrinsic functions
- XML Processing Example →→→→

```
01 XMLdocument pic N(10000) usage national.  
  
XML PARSE XMLdocument  
    Processing procedure XMLproc  
End-XML.  
...  
XMLproc.  
    Evaluate XML-Event  
        When 'START-OF-ELEMENT'  
            If XML-NText = N'Ελλάδα'  
                Display 'Processing <Greece>  
element'  
                ...  
            End-if  
        ...  
    End-evaluate.
```

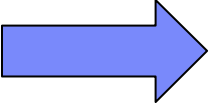


# IBM Object Oriented COBOL

- Enable fine-grained interoperation of COBOL and Java within an address space, both:
  - ▶ COBOL invocation of Java
  - ▶ Java invocation of COBOL
- Complement existing COBOL:Java interoperation
  - ▶ mediated by middleware, based on connectors
  - ▶ COBOL and Java running in different address spaces or machines
  - ▶ only for COBOL transactions
- Fine-grained interoperation (interlanguage communication) provides:
  - ▶ better performance
  - ▶ use of non-transactional COBOL
- Improve integration of COBOL with WebSphere Application Server
  - ▶ COBOL client invocation of enterprise beans
  - ▶ Future support for COBOL execution within WebSphere server regions



# Agenda

- Introduction
- Enterprise COBOL
  - ▶ XML Support
  - ▶ CICS V3.x Support
  - ▶ Unicode Support
  - ▶ Object Oriented COBOL
-  Rational Developer for System z
  - ▶ Mainframe development features
  - ▶ XML and Web Services Support
  - ▶ CICS V3.x and Service Flow Feature support
- More Information
- Questions?





# IBM Rational Developer for System z

## JES and PD Tools

- Read/Write/Update VSAM datasets via integration with IBM File Manager
- Access IBM Fault analyzer reports for analyzing ABENDS and associating back to source code
- Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output
- Debug zOS applications from with workstation as they execute live in the remote runtime

## Integration with EGL using RBDe

- Quick and easy development of modern enterprise applications for procedural programmers
- Simplify and speed up creation of Web applications and services without having to learn Java or J2EE

## Traditional Development

### Development Environment

- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/1, C, C++, JCL, etc.
- 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
- Create, build, and catalog DB2 stored procedures on zOS
- Compile and test programs locally to ensure correctness

### Screen design

- Visually create, modify, build, and deploy BMS maps sets or MFS/IMS maps remotely or on the local workstation

### Code Generation

- Generate CRUD DB2 program code from UML, which can also be easily integrated into web service applications

## IBM Rational Developer for System z

RBDe

### Host Tooling Integration

[JES, FA, FM, Debug Tool]

### zOS Application Development

[COBOL, PL/1, C/C++, JCL, Screens, Stored Procedures, etc]

### Enterprise Service Tools

[Web Services For CICS/IMS]

### Host / Distributed SCM Integration

### IBM Rational Application Developer

## zOS Web Service and Flow Creation

- Implements SOA and Web Services
- SOA access to CICS V3.2 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
- Service Flow Modeler to build/deploy service flows out of your existing Commarea, Channel, MQ, and Terminal CICS applications.

## SCM Support

- Access to host SCMs such as SCLM
- Framework for writing/deploying custom SCM integration code
- Support for storing zOS resources in distributed SCMs such as ClearCase

## Web and JEE Development

- Create Web Pages / JSF / Struts
- JEE/Java Development
- JCA Connectors
- Distributed debugger
- Web Services and Test environment



# RDz IDE - Eclipse-based development

- Common development environment for COBOL, PL/I, C/C++, and Java
- Simplified development with more information at your fingertips

**Syntax Check**

**Edit Source**

**Open and edit multiple source and JCL members simultaneously**

**Submit jobs, access job output, or open source members with a single click**

**Statement in error indicated in source**

**Outline view presents COBOL structure**

**Double-Click on the Error**

**Error list in Problems view**

The screenshot shows the RDz IDE interface with the following elements:

- Project Explorer:** Shows a project named 'DEMOCOBOL' with members 'RBAROSA.IDE.JCL', 'RBAROSA.IDE.LOAD', 'RBAROSA.SP.COBL', 'REGIOA.cbl', and 'REGIOB.CBL'. 'REGIOA.cbl' is selected.
- Properties View:** Shows the structure of the selected member, including 'PROCEDURE DIVISION.', '010-INITIALIZATION.', '020-LOGIC.', '030-SEEYA.', and '040-GOODBYE.'. '010-INITIALIZATION.' is highlighted.
- Editor:** Displays the source code for 'REGIOA.cbl'. Line 35 contains the statement 'DISPLAI "Program REGIOA STARTING "' which is highlighted in red. The error message in the Problems view indicates that 'DISPLAI' is invalid.
- Problems View:** Shows a list of errors, with one error highlighted: 'IGYPS2072-S "DISPLAI" was invalid. Skipped to the next verb, period or procedure-name d...'. A yellow box highlights the instruction 'Double-Click on the Error'.

# Interactive Access to z/OS

**Files on workstation**

**JES Listings**

**MVS Files**

**MVS datasets**

**MVS datasets mapping**

Mapping Criterion	Workstation Extension	Transfer Mode	Host Code Page	Local Code Page
**COBOL	cbl	text	IBM-037 (inhe...	Cp1252 (inheri...
**COBCOPY	cpy	text	IBM-037 (inhe...	Cp1252 (inheri...
**PLI	pli	text	IBM-037 (inhe...	Cp1252 (inheri...
**ASSEMBLE	asm	text	IBM-037 (inhe...	Cp1252 (inheri...
**OBJ	obj	binary	IBM-037 (inhe...	Cp1252 (inheri...
**LOAD	exe	binary	IBM-037 (inhe...	Cp1252 (inheri...
**CLIST	cmd	text	IBM-037 (inhe...	Cp1252 (inheri...
**JCL	jd	text	IBM-037 (inhe...	Cp1252 (inheri...
**SIGYCLST	cmd	text	IBM-037 (inhe...	Cp1252 (inheri...
**CNTL	jd	text	IBM-037 (inhe...	Cp1252 (inheri...
**FILES	<undefined>	text	IBM-037 (inhe...	Cp1252 (inheri...
COB**	*BL	text (inherited)	IBM-037 (inhe...	Cp1252 (inheri...
JCL**	*	text (inherited)	IBM-037 (inhe...	Cp1252 (inheri...
BMS**	BM	text (inherited)	IBM-037 (inhe...	Cp1252 (inheri...
**LISTING	lst	text	IBM-037 (inhe...	Cp1252 (inheri...
**OUTLIST	out	text	IBM-037 (inhe...	Cp1252 (inheri...
**INCLUDE	inc	text	IBM-037 (inhe...	Cp1252 (inheri...

**member mapping**

**Monitoring job output**

Name	Job ID	Job Name	Job Owner	Return C...	Return Info	System r...	User retu...	Return St...
DNET045C:JOB03276	JOB03276	DNET045C	DNET045	U0004	NORMAL		004	COMPLET...
DNET045X:JOB03410	JOB03410	DNET045X	DNET045	U0000	JCLERROR			



# IBM Debug Tool, File Manager, and Fault Analyzer Integration

Template Associated: SKOONCE.FM1.TEMPLATE(CRA390)    HEX On

Name	Employee Number	Age	Salary	Month
Grant Smith	771235	7	5000	6
Andrew Apple	664553	7	8500	30
Graham Prescott	558328	4	8000	7
15 records excluded				
Bill Somers	441883	6	8000	5
24 records not selected				
2 records suppressed				
Ted Dexter	332752	6	0250	14

Single Mode  
Record 4 of 10, Top Line is 1 of 2

Field	Data
Name	Bill Somers
Employee Num...	441883
Age	6
Salary	8000
Month	5

IBM Distributed Debugger

ATCDEMO.42262872@demomvs.demopkg.ibm.com

Stacks Breakpoints Modules

Breakpoints Control Panes

- Storage Change Breakpoints
- Address: record-day Bytes: 2
- Line Breakpoints
- Source: DNET603.DEMOS.DT.PDPAK.SIDEFIL(ATCDEMO) Line: 183

Monitors Locals Storage Registers Mapping Mapping2

Thread 1: DNET603.DEMOS.DT.PDPAK.SIDEFIL(ATCDEMO)

```

174 IF CURRENT-DATE = RECORD-DATE
175     ADD 10 TO RECORD-DAY
176 ELSE
177     ADD 5 TO RECORD-DAY
178 END-IF.
179
180 *****
181 * COMPARE THE SYSTEM DATE IS LESS THAN THE RECORD DATE
182 *****
183 IF CURRENT-DATE LESS THAN RECORD-DATE
184     CALL 'ATCDM2'
185     ADD 10 TO RECORD-DAY
186 ELSE
187     ADD 10 TO RECORD-YEAR
188 END-IF.
189
190 *****
191 * COMPARE THE ELEMENTS OF THE SYSTEM DATE WITH THE
192 * ELEMENTS OF THE RECORD DATE
193 *****
194 IF CURRENT-DAY = RECORD-DAY
195     CALL 'ATCDM2'
196 ELSE
197     END-IF.
198
199 IF CURRENT-MONTH = RECORD-MONTH
200     CALL 'ATCDM4'
201 ELSE
202     CALL 'ATCDM5'
203

```

Thread 1

- INPUT-RECORD = EQA2302E Not allocated
- CURRENT-DATE
  - CURRENT-YEAR = 04
  - CURRENT-MONTH = 03
  - CURRENT-DAY = 28
- CURRENT-TIME
- WS-INPUT-RECORD
- DATE-SW = 'N'
- QSADMIN-STATUS = ''
- RF-STATUS = ''
- SW-SWITCHES-AREA
- WS-COUNTERS

Debugger ready. Daemon listening on port 8001...

z/OS Projects - F00031.far - IBM WebSphere Developer for System z

z/OS Pr... FA Artf... F00031.far

History Files

- PSTRAND.FA.HISTORY
- PSTRAND.FAE.HISTORY
- PSTRAND.FAT.HISTORY
- PSTRAND.FAT2.HISTORY
- TSOE1
- History Files
- PSTRAND.FA.HISTORY

Browse Dump

**Fault Summary**

Module COBBLOW, program COBBLOW, source line # 19  
: Abend SOC9 (Fixed-Point-Divide Exception)

**Synopsis**

IBM FAULT ANALYZER SYNOPSIS

A system abend 0C9 occurred in module COBBLOW program COBBLOW at offset X'3B2'.

A program-interruption code 0009 (Fixed-Point-Divide Exception) is associated.

Main Report | System Wide Information | Misc Information

Remote Error List | z/OS File System Mapping | Remote System Details | Default

Column Configuration

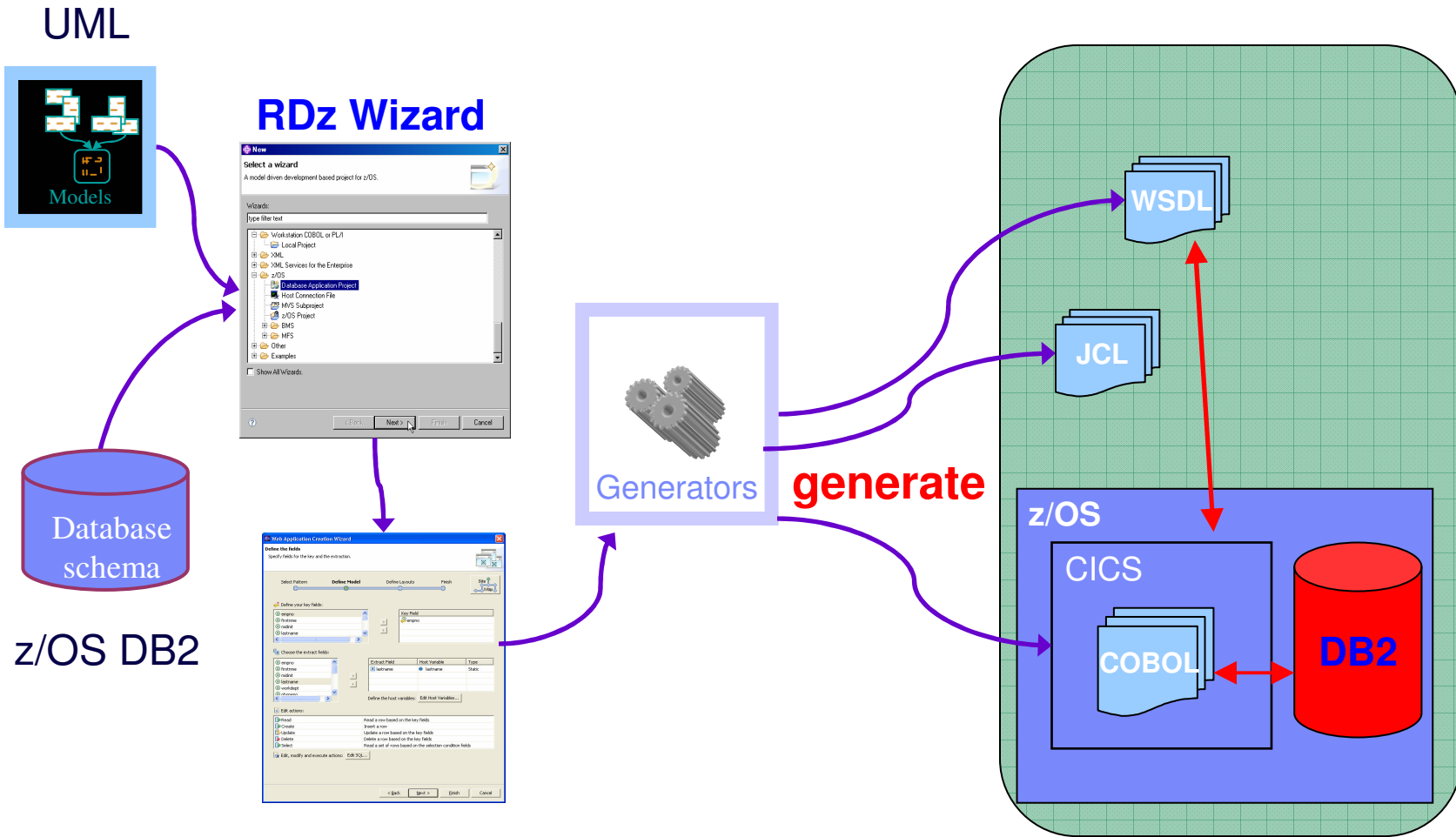
System Name: TSOE1

Fault History File or View: PSTRAND.FA.HISTORY

Fault_ID	Date	Job/Tran	User_ID	Sys/Job	Abend	I_Abend
F00011	March 12, 2004	WPSBMS	CBP8MS	C2C2	SEC3	n/a
F00009	March 12, 2004	WPSBMS	CBP8MS	C2C2	SEC3	n/a
F00012	March 12, 2004	WPSBMS	CBP8MS	C2C2	SEC3	n/a
IMS00104	October 28, 2003	IMC80010	BILLU	MVS2	SOC8	n/a
F00006	August 1, 2003	PSTRAND4	PSTRAND	STLABF1	SOC9	n/a
F00001	June 7, 2003	BILLUUR	PSTRAND	MVS2	SOC9	n/a
F00031	January 11, 2007	COBBLOW	PSTRAND	FAE1	SOC9	SOC9



# Database Application Generator wizard Architecture Overview



# New on version 7.1 - Template-driven program creation

**1 Define Templates**

**2 Define Features**

**3 Create a Program with pre-defined comments and Code**

**Benefit: Speed application creation and ensure conformance to standards**

```

000001 *****
000002 * DESCRIPTION : BLA BLA BLA
000003 * Receive xxxx, read yyyy, mode data to zzzz
000004 * ENVIRONMENT : CICS, COBOL II
000005 * CICS TRANSACTION NAME : uuuu
000006 * SUBROUTINE: zzzz
000007 * DB2 TABLES/VSAM FILES: HQ.name1.name2
000008 * date: Sep 26, 2007
000009 *****
000010 *** MODIFICATIONS
000011 *** WHO DATE CHANGE
000012 *****
000013 IDENTIFICATION DIVISION.
000014 PROGRAM-ID. PROGX.
000015 AUTHOR. Barosa.
000016 ENVIRONMENT DIVISION.
000017 INPUT-OUTPUT SECTION.
000018 DATA DIVISION.
000019 FILE SECTION.
000020
000021 WORKING-STORAGE SECTION.
000022 * This line will be included when having "Use SQL statements"
000023 EXEC SQL INCLUDE SQLCA END-EXEC.
000024
000025 LOCAL-STORAGE SECTION.
    
```

## New on Version 7.1 - Snippet Insertion

- Define small bits of code and save code into a snippet view
- Optionally define variables in the code block
- Insert the code later directly into the editor

### Benefits:

- Speed application creation and ensure conformance to shop standards
- Define re-usable logic in an organized manner that can be easily used in programs
- Define organization-wide code templates to share with others

**Customize Palette**

**REGI\_Dynamic\_Call**

Name: REGI\_Dynamic\_Call

Description: This is a snippet to do a dynamic call passing one parameter

Hide

Name	Description	Default Value
Program-name	This is the called pr...	REGIOB
passed-param	This is the passed fi...	received-from-called

Template Pattern:

```
MOVE "${Program-name}" to program-to-call
CALL program-to-call USING ${passed-param}
```

000221 END-IF .  
 000222 FILL-CUST-DATA-EXIT.  
 000223 EXIT.  
 000224 UPDATE-CUST-DATA.

000221 END-IF .  
 000222 FILL-CUST-DATA-EXIT.  
 000223 MOVE "REGIOB" to program-to-call  
 000224 CALL program-to-call USING received-from-called  
 000225 EXIT.

## New in 7.1

- *Core z/OS development*
  - ▶ Enhanced COBOL outline view including filtering capabilities
  - ▶ Content assist and syntax check error display for new CICS TS V3.2 syntax
  - ▶ SSL support for job submissions and monitoring
  - ▶ Simplified configuration of host servers
  - ▶ Additional ISPF commands and keyboard access available through editors
  - ▶ Support for DB2 V9 applications and stored procedures
  - ▶ Extensible template-driven COBOL program generation framework to create COBOL program skeleton including comments, divisions, and code based on intended usage:
    - Basic Mapping Support (BMS) Screens
    - CICS Applications
    - SQL error handling
    - Customer created extension
  - ▶ Next Generation Development Tooling for COBOL (NGDT) Technology Preview speeds application development using enhanced COBOL editors, pattern-based code generation, and UML to source skeleton generation





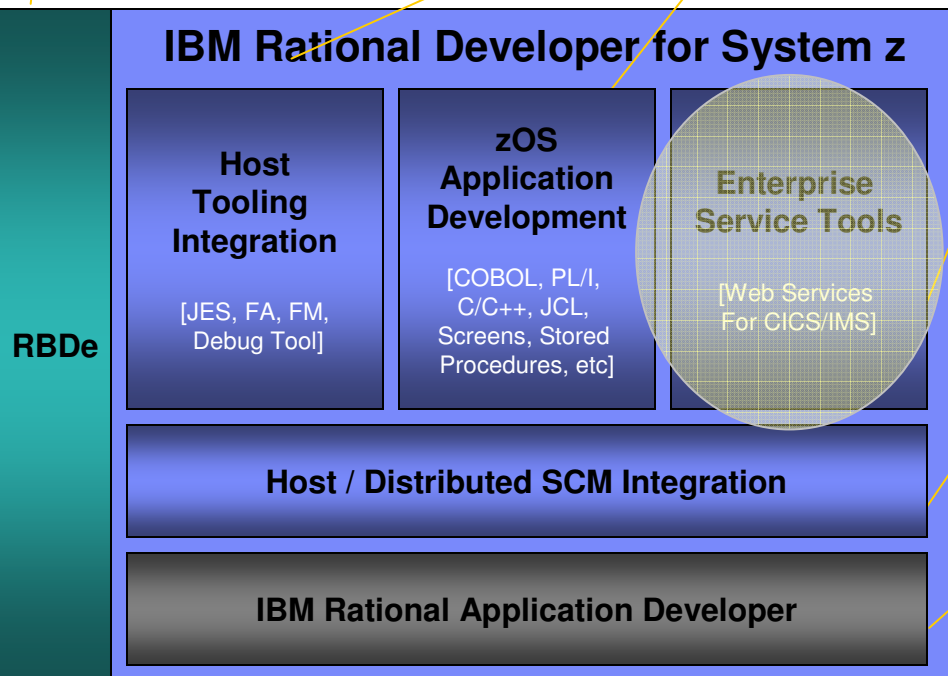
# IBM Rational Developer for System z

## JES and PD Tools

- Read/Write/Update VSAM datasets via integration with IBM File Manager
- Access IBM Fault analyzer reports for analyzing ABENDS and associating back to source code
- Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output
- Debug zOS applications from with workstation as they execute live in the remote runtime

## Integration with EGL using RBDe

- Quick and easy development of modern enterprise applications for procedural programmers
- Simplify and speed up creation of Web applications and services without having to learn Java or J2EE



## Traditional Development

### Development Environment

- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/1 , C, C++, JCL, etc.
- 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
- Create, build, and catalog DB2 stored procedures on zOS
- Compile and test programs locally to ensure correctness

### Screen design

- Visually create, modify, build, and deploy BMS maps sets or MFS/IMS maps remotely or on the local workstation

### Code Generation

- Generate CRUD DB2 program code from UML, which can also be easily integrated into web service applications

## zOS Web Service and Flow Creation

- Implements SOA and Web Services
- SOA access to CICS V3.2 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
- Service Flow Modeler to build/deploy service flows out of your existing Commarea, Channel, MQ, and Terminal CICS applications.

## SCM Support

- Access to host SCMs such as SCLM
- Framework for writing/deploying custom SCM integration code
- Support for storing zOS resources in distributed SCMs such as ClearCase

## Web and JEE Development

- Create Web Pages / JSF / Struts
- JEE/Java Development
- JCA Connectors
- Distributed debugger
- Web Services and Test environment

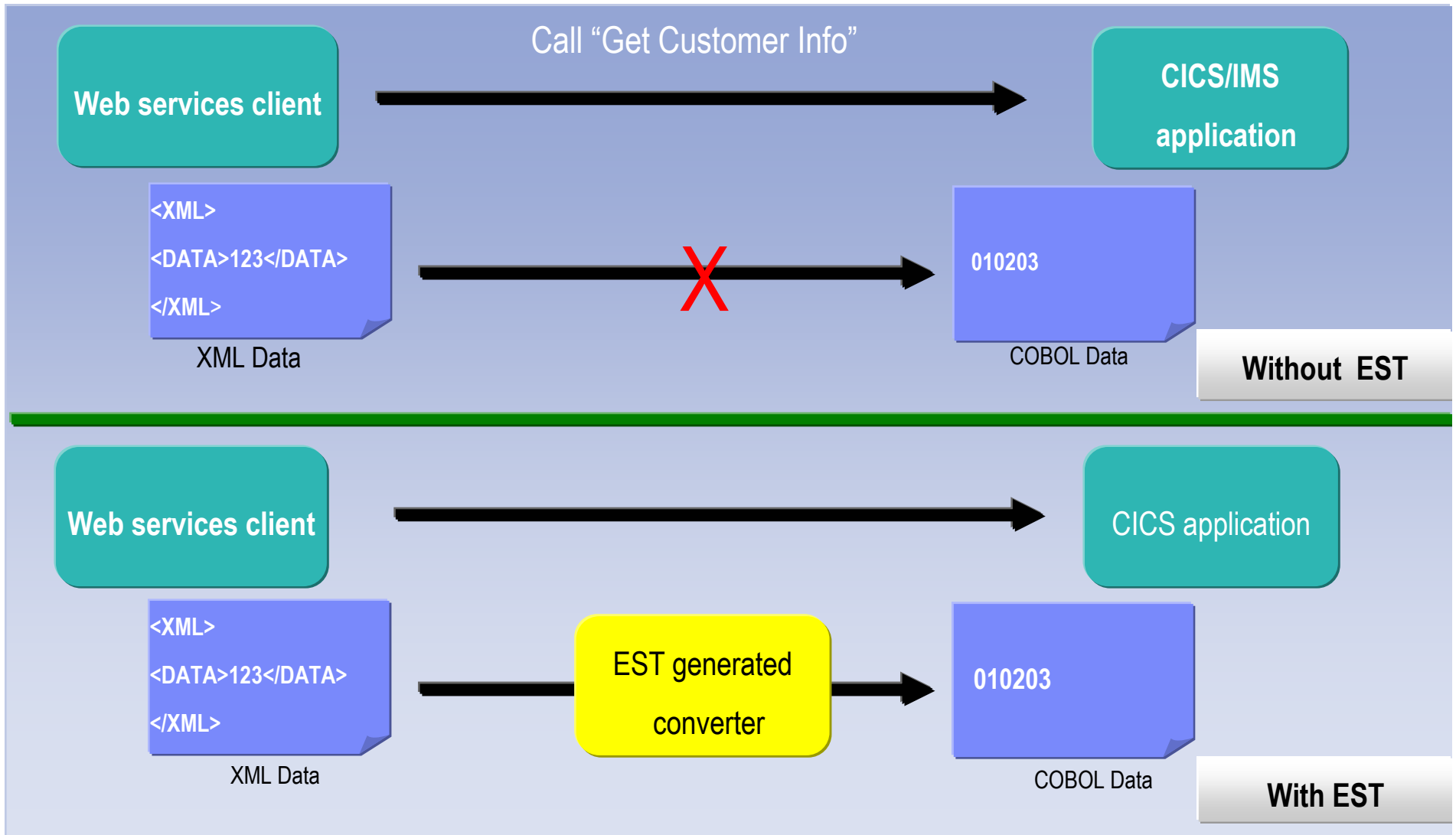


## Why Web Services?

- Web services provide standardized access to assets for different software applications residing on disparate platforms
- Web service definitions provide abstract interfaces which allow for loose coupling between business components – implementation can vary without affecting consumers
- You can reuse applications exposed as Web services in a variety of service-oriented architecture frameworks, such as a process choreographer or an enterprise service bus.

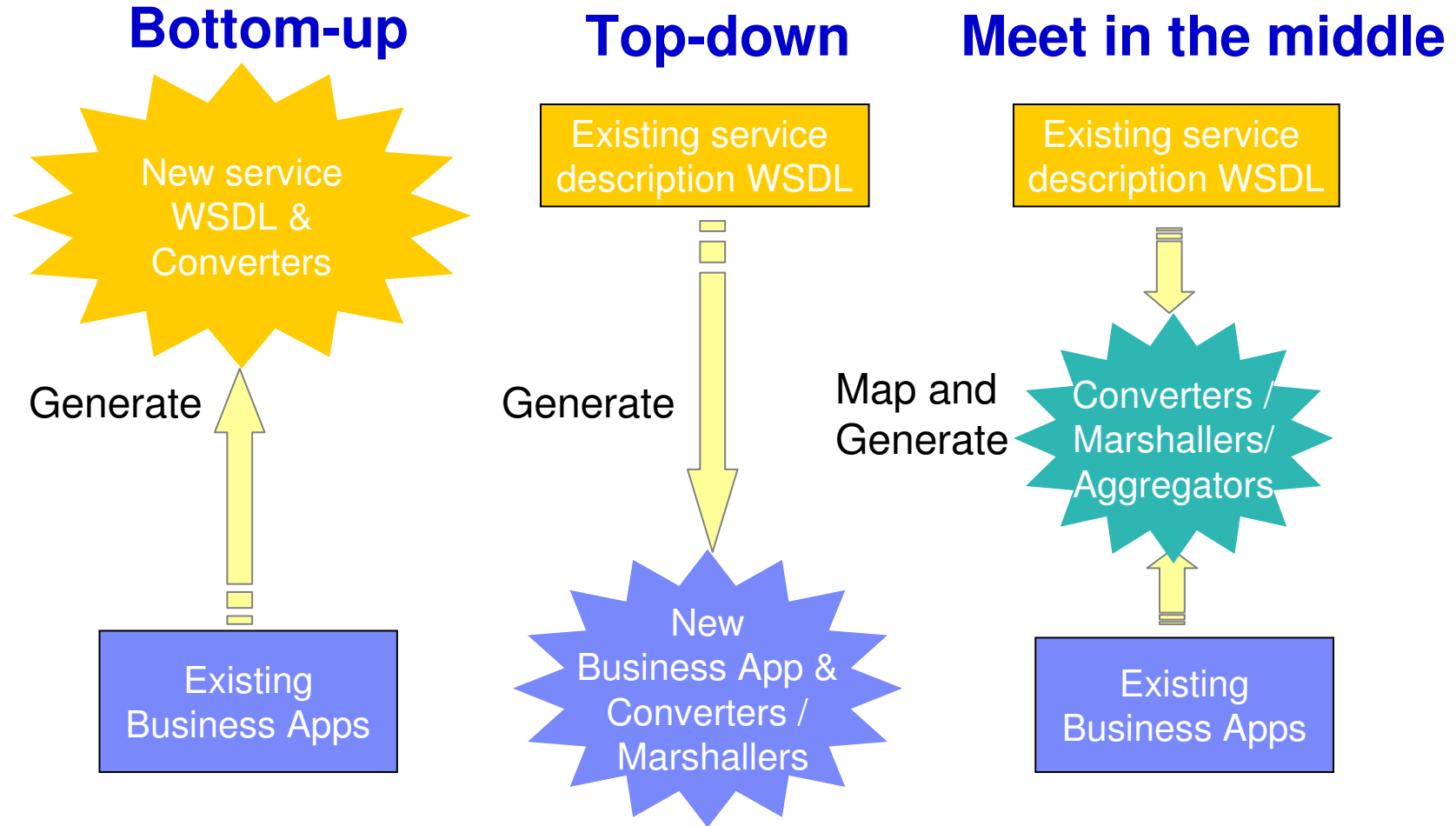


# We need interfaces to talk “XML” ....



# Enterprise Service Tools (EST)

## Web Service Enablement Styles



# Enterprise Service Tools (EST)

## Example Bottom-up

**Web service WSDL & Converters**

**Existing COBOL or PL/I**

**1**

**2**

**3**

**Generate**

**Bottom-up**

**Enterprise Service Tools Wizard Launchpad**

Specify options to start a web service wizard

Host runtime: Web Services for CICS

Development scenario: Create New Service Interface (bottom-up)

Application mode: Web Service Provider

Conversion type: Compiled XML Conversion

Scenario description:  
Generate a Web service description and runtime specific XML message processing from a high level language data structure. You can use this option when you expose an application program as a service provider.

Save generation properties

Start Cancel

Targets

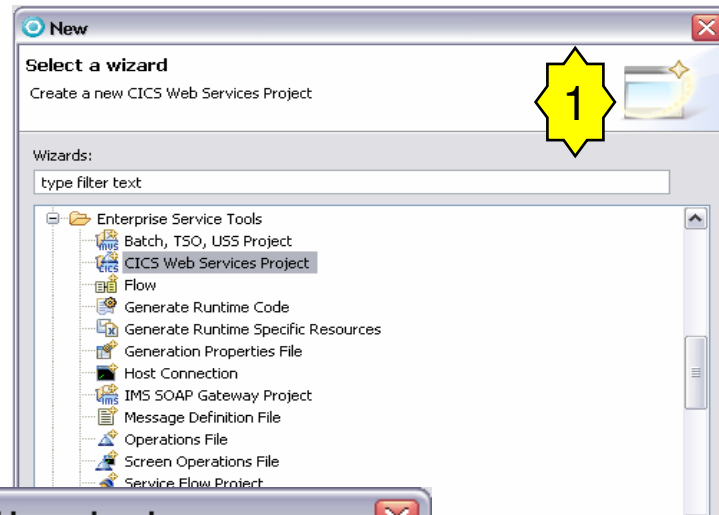
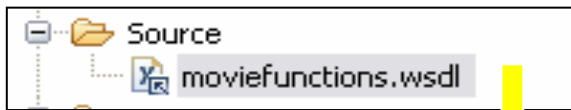
- LAB324.wsdl
- LAB324D.cbl
- LAB324I.xsd
- LAB324O.xsd
- LAB3POT.log
- LAB3POT.wsbind

# Enterprise Service Tools (EST)

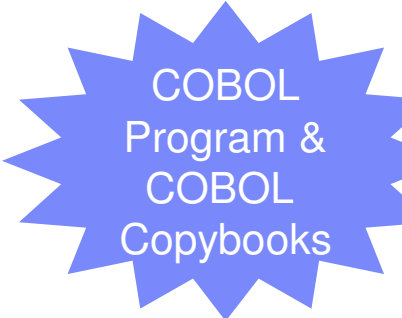
## Example Top-down

### Top-down

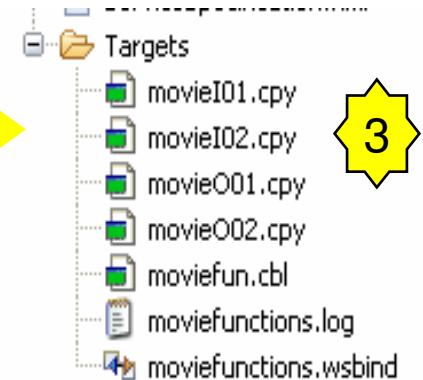
Existing service description WSDL



### Generate



Generate



# Enterprise Service Tools (EST)

## Example Meet in the middle

### Meet in the middle

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

**Enterprise Service Tools Wizard Launchpad**

Specify options to start a web service wizard

Host runtime: Web Services for CICS

Development scenario: Map an Existing Service Interface (meet-in-middle)

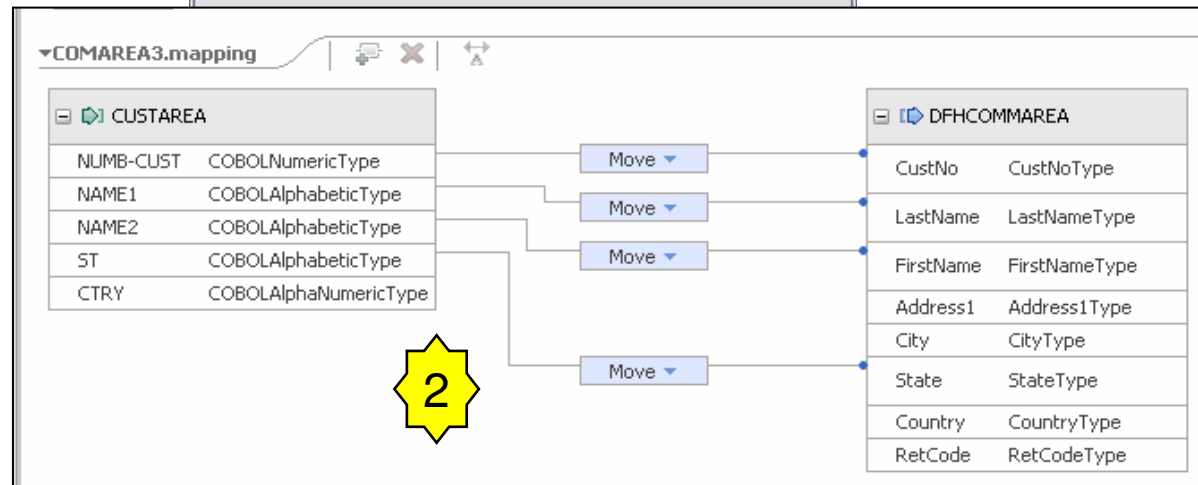
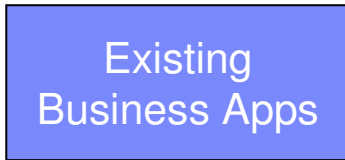
Application mode: Web Service Provider

Conversion type: Compiled XML Conversion

Scenario description:  
 Define mappings between high level language data structures and WSDL, XML XSD or DTD files. You can use this option to generate runtime specific XML message processing based on the mappings.



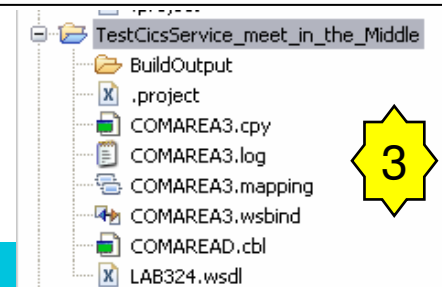
Map and Generate



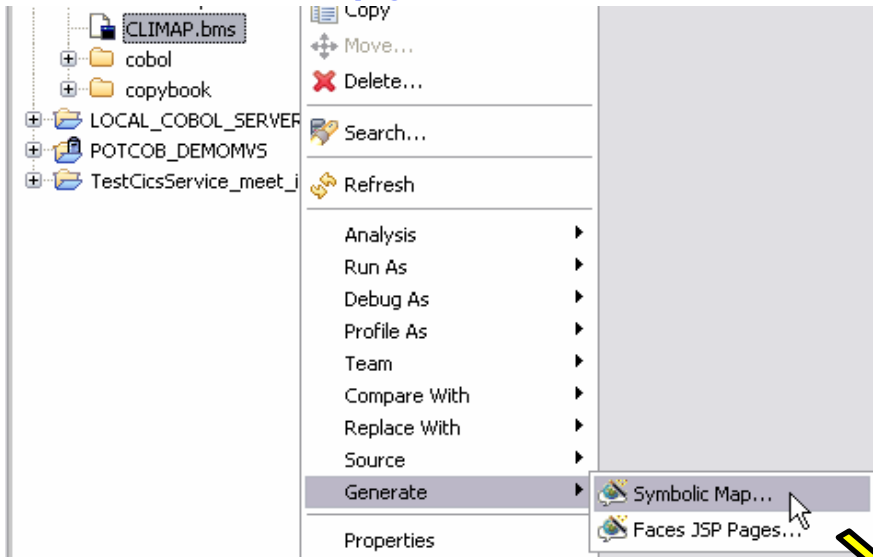
```

01 CUSTAREA.
02 NUMB-CUST PIC S9(5).
02 NAME1 PIC A(25).
02 NAME2 PIC A(15).
02 ST PIC A(5).
02 CTRY PIC X(15).
    
```

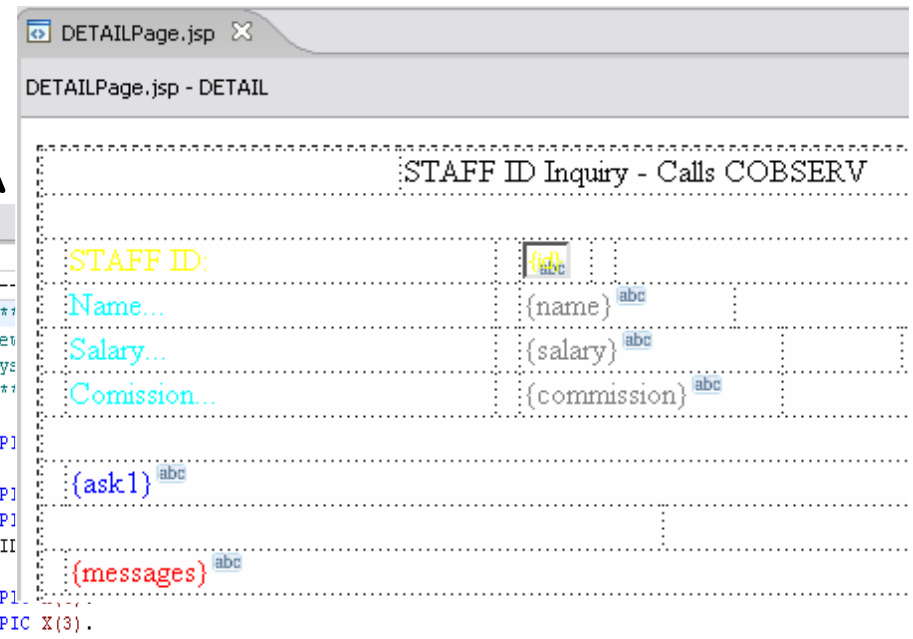
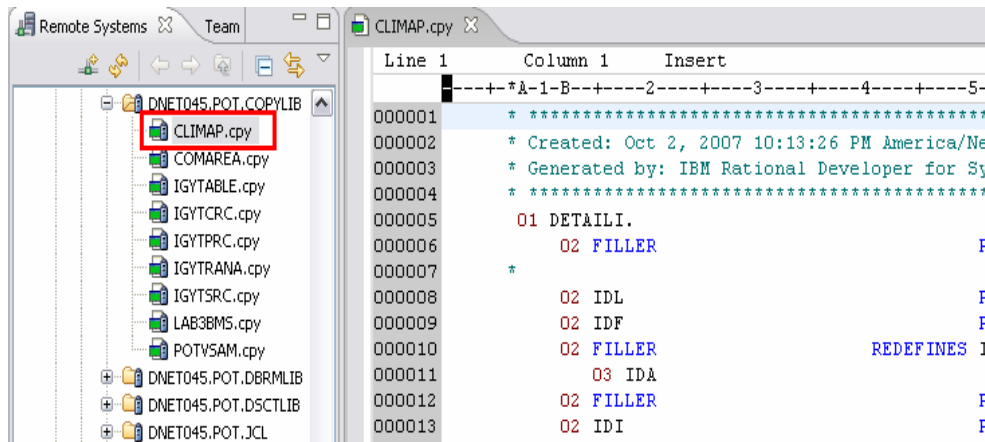
Generate



# Generate Copybook or JSP for CICS



- Generate COPYBOOK to be used in the program (symbolic map)
- Generate Faces JSP





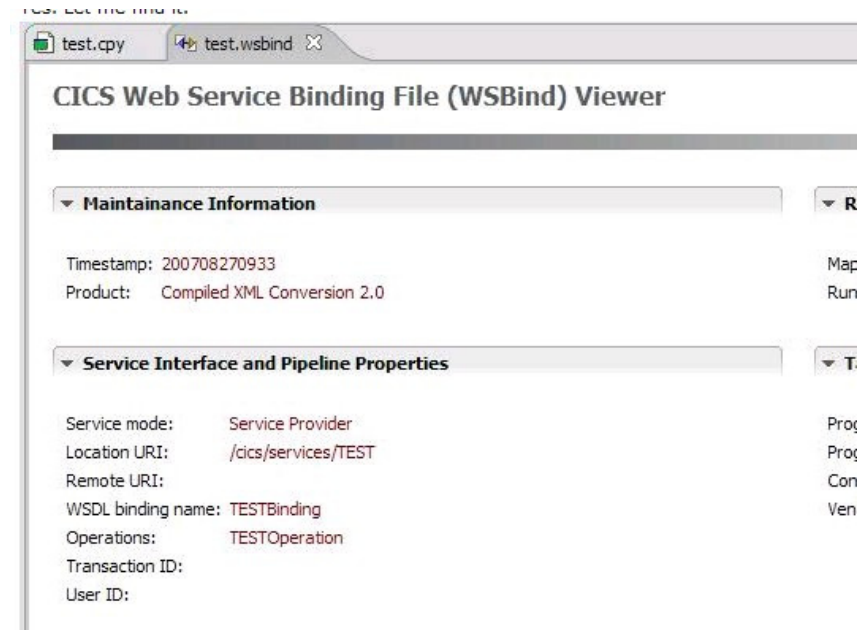
# Example: Testing using RDz

The screenshot displays the IBM Rational software interface. On the left, a project tree shows a folder named 'EOT\_3\_LOCAL\_Create\_services' containing several files, including 'CUSTINQ.wsdl'. A context menu is open over this file, with the 'Web Services(H)' option selected, and its sub-menu is visible, showing 'Test with Web Services Explorer' as the active option. A large blue arrow points from this menu item towards the right side of the interface. On the right, a window titled 'CUSTINQBinding' is open, showing a test interface. A text input field labeled 'CustNo int' contains the value '6', which is highlighted with a red box. Below this, a 'Status' window displays the test results for the 'DFHCOMMAREA' object, listing the following data: 'CustNo (int): 6', 'LastName (string): Barosa', 'FirstName (string): Thiago', 'Address1 (string): 7 Sao Benedito', 'City (string): Sao Paulo', and 'State (string): SP'. This status window is also highlighted with a red box.

## New on 7.1 - Service Support

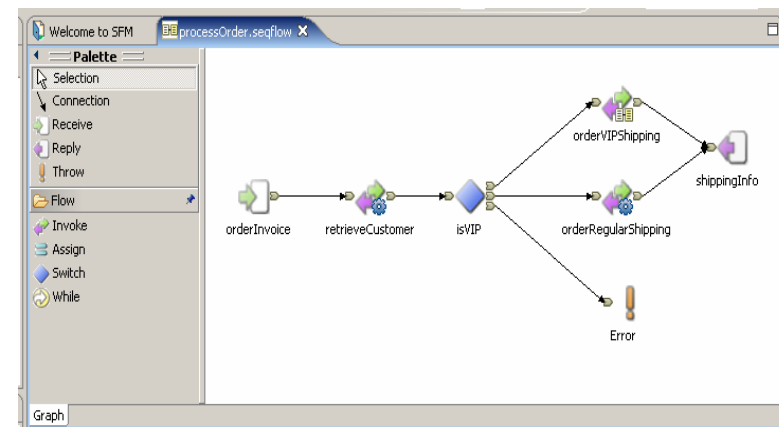
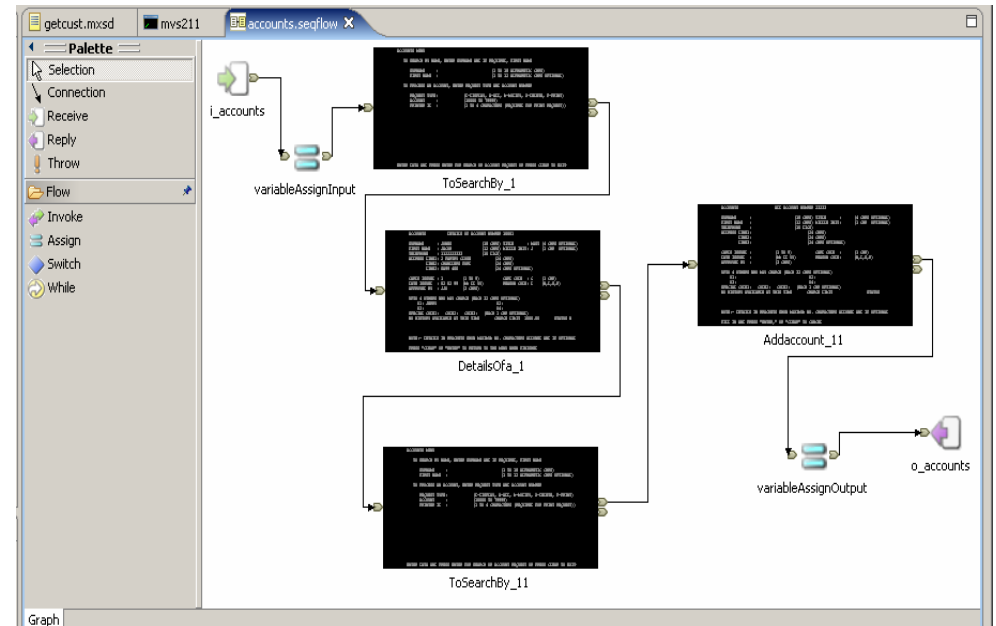
### Single Runtime Services

- Support for new **CICS TS V3.2** constructs
  - ▶ WSDL 2.0
  - ▶ SOAP 1.2
- Support for **IMS V10** Web Service callouts
- Support for PLI web services
- Usability improvements for web-service creation wizards/projects/scenarios
- CICS **WSBind** file viewer
- Web Service Namespace toleration in web service creation and test environment

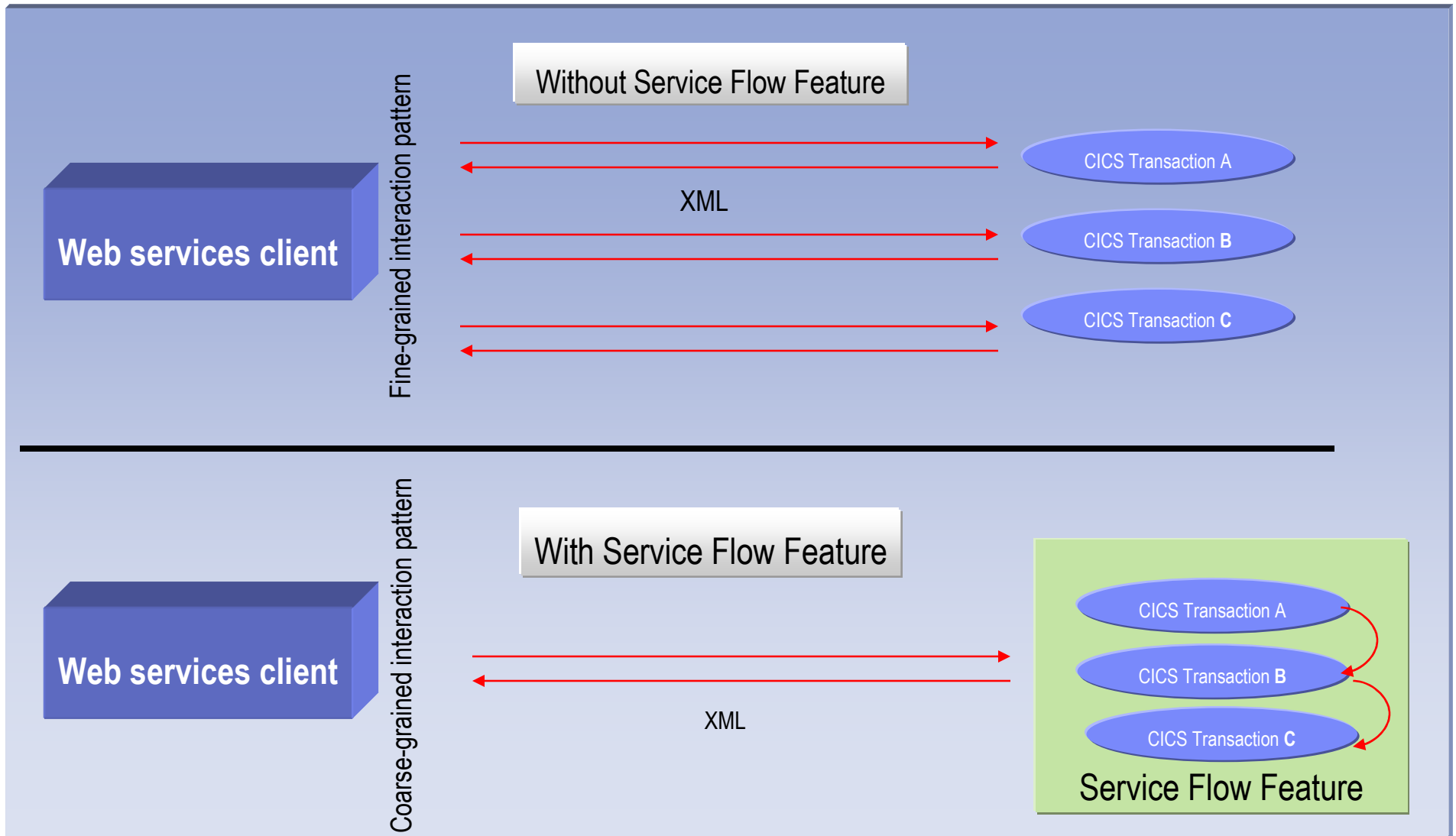


# What is Service flow support?

- Service Flow Feature is a CICS feature. **Service flow support** is a RDz tool to build service flows out of your existing **COMMAREA/container, WSDL 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.x or WebSphere Application Server
  - ▶ Optionally deploy business process as a web service
- Development concepts consistent with other SOA development tasks
- Support for Service Flow Runtime V3.2
- Channel/Container Support



# Service Flow in CICS V3.x

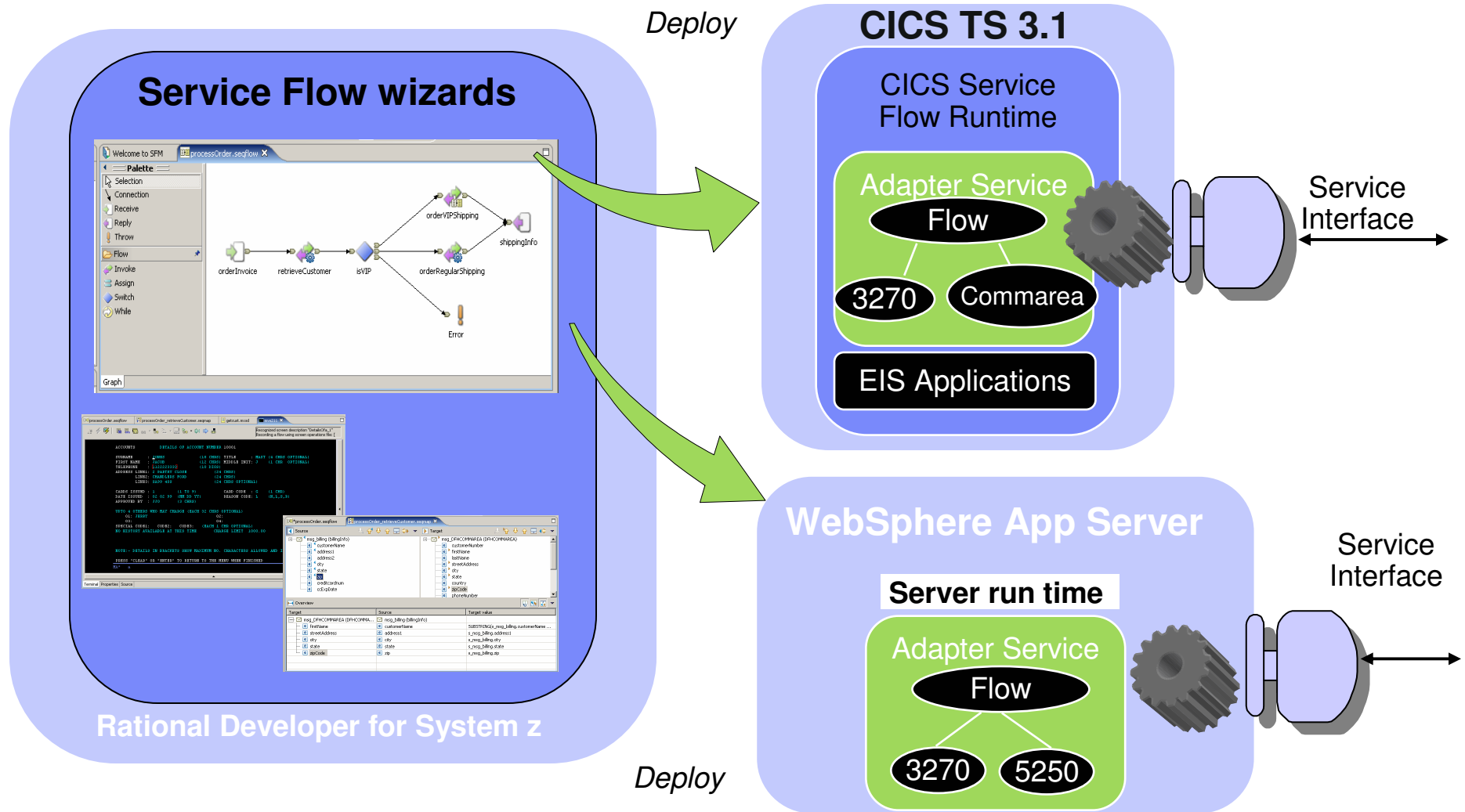


## Why Service Flow Feature?

- **Increase Productivity**
  - ▶ By building libraries of annotated components representing current assets
  - ▶ By rapidly assembling new applications out of existing components using graphical tools
  - ▶ By exploiting existing developer skills and literacy
- **Transform the Enterprise**
  - ▶ By unlocking critical IT assets and re-purposing them to participate in a service oriented architecture
  - ▶ By opening access to existing fine-grained applications as coarse-grained business functions, while maintaining QOS
  - ▶ By providing a layer of abstraction between service consumer and application implementation / user interface
  - ▶ By fostering SOA skills in traditional developers



# Supported Runtimes



## RDz Application Deployment Manager for CICS

- Helps developer to execute CICS tasks
- Miscellaneous server development aids
  - ▶ New copy for programs and mapsets
  - ▶ Perform validate check for existing resource names (avoid resource name clash that is a problem today)
  - ▶ List CICS regions
  - ▶ DFHRPL list
- Web service development aids
  - ▶ Perform Pipeline scan to autoinstall URIMAP and WEBSERVICE definitions
  - ▶ Provide pipeline and WSBind pickup directory to populate SFM selection list
  - ▶ Provide WSDL file directory to populate SFM selection list
  - ▶ Provide End Point URI to populate SFM selection list



## New in 7.1

- Support for CICS TS V3.2 Web Service runtime introduced, including support for Web Services Description Language (WSDL) 2.0 and SOAP 1.2 standards
- Tools for IMS V10 outbound Web service callout code generation introduced
- PL/I language support in Web Services wizards introduced for generating Web Services from existing PL/I programs for CICS, IMS, and batch processing
- Improved meet-in-the-middle development scenario tooling wizards to link-up Web Services specifications with existing CICS, IMS, and batch applications
- Improved bottom-up development scenario tooling allowing for more flexible Web Service creation based on existing programs
- New CICS WSBIND file viewer for browsing existing/generated CICS Web Services configurations
- Web Service testing and generation tools now tolerate custom XML namespaces defined as part of the service input or output specification





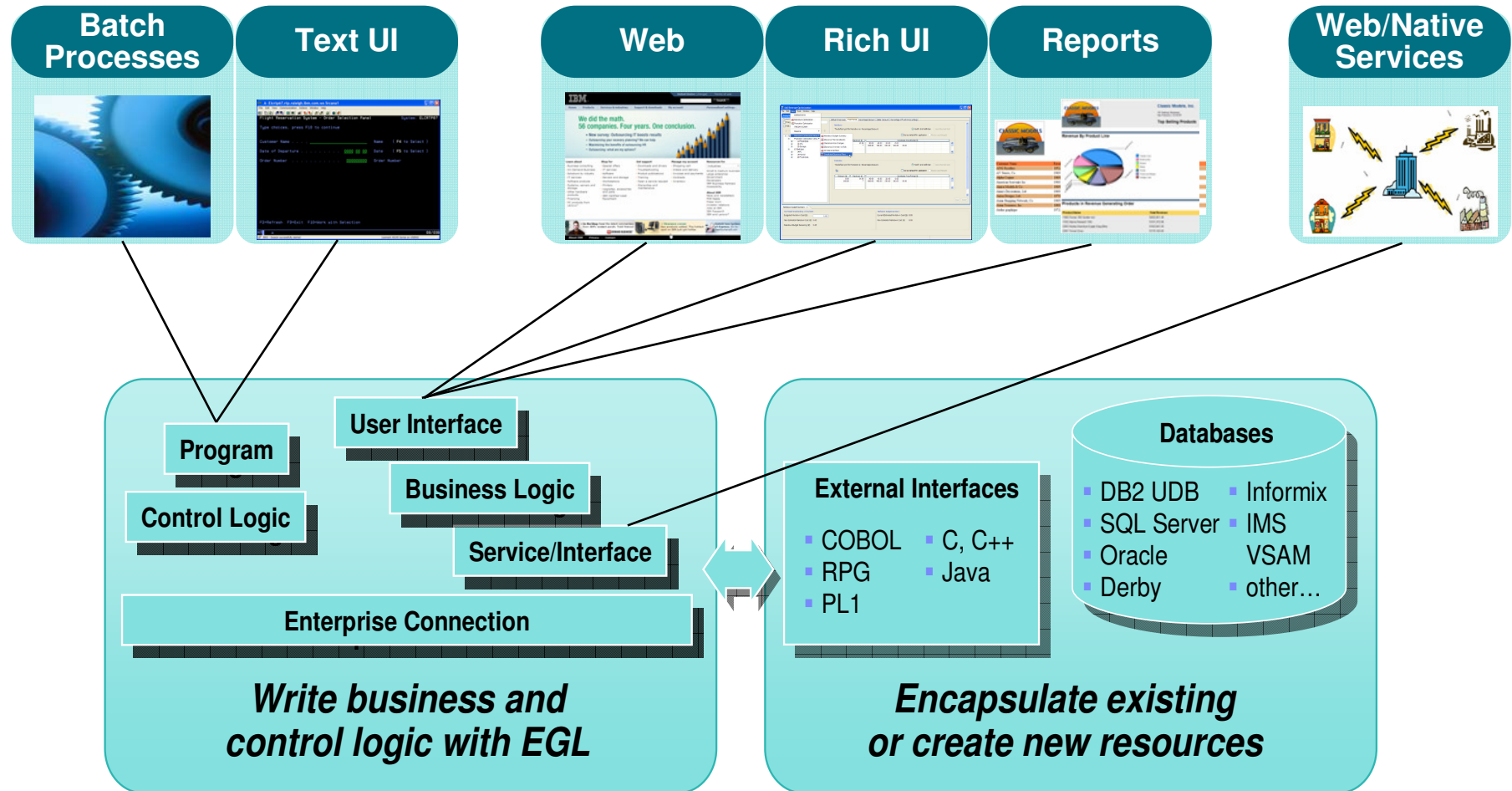
## Benefits of RDz XML Enablement

- Enterprise modernization:
  - ▶ Easy to “reface” existing COBOL applications to support XML messages
  - ▶ Adapts the rigid, binary interface of traditional programs.
- Programmer productivity:
  - ▶ Generates complete programs that easily convert between XML and COBOL datatypes
  - ▶ Generates a sample program that illustrates use of converter programs with existing COBOL
  - ▶ Exploits customers' existing assets/skills/literacy
- Performance
  - ▶ XML processing uses the native z/OS high-speed parser
- Supports multiple runtime scenarios
  - ▶ Including web services



# Rational Business Developer Extension provides Application Flexibility

End-to-end development for a broad variety of applications

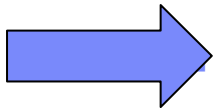


Note-Plugs in to Rational Developer for System z



# Agenda

- Introduction
- Enterprise COBOL
  - ▶ XML Support
  - ▶ CICS V3.x Support
  - ▶ Unicode Support
  - ▶ Object Oriented COBOL
- Rational Developer for System z
  - ▶ Mainframe development features
  - ▶ XML and Web Services Support
  - ▶ CICS V3.x and Service Flow Feature support
- ▶ More Information
- Questions



## More information

- Webcast/teleconference replay
  - ▶ Using SCMs and WDz to optimize mainframe application development
    - [www-306.ibm.com/software/sw-events/teleconference/M644523H65158H87.html](http://www-306.ibm.com/software/sw-events/teleconference/M644523H65158H87.html)
- Websites
  - ▶ RDz – [www.ibm.com/software/awdtools/rdz](http://www.ibm.com/software/awdtools/rdz)
  - ▶ COBOL - [www-306.ibm.com/software/awdtools/cobol/zos/](http://www-306.ibm.com/software/awdtools/cobol/zos/)
  - ▶ Enterprise Modernization – [www.ibm.com/rational/modernization](http://www.ibm.com/rational/modernization)
- Developerworks
  - ▶ UML to COBOL
    - article
      - [www.ibm.com/developerworks/websphere/techjournal/0708\\_col\\_barosa/0708\\_col\\_barosa.html](http://www.ibm.com/developerworks/websphere/techjournal/0708_col_barosa/0708_col_barosa.html)
    - Tutorial
      - [www.ibm.com/developerworks/edu/wes-dw-wes-uml2cobol1.html](http://www.ibm.com/developerworks/edu/wes-dw-wes-uml2cobol1.html)
  - ▶ Model Driven Development in the mainframe environment article
    - [www.ibm.com/developerworks/websphere/library/techarticles/0708\\_england/0708\\_england.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0708_england/0708_england.html)
  - ▶ Writing java programs in a mainframe environment tutorial
    - [//www.ibm.com/developerworks/websphere/library/techarticles/0703\\_england/0703\\_england.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0703_england/0703_england.html)



# More information...IBM Education Assistant

<http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp>

**Contents**

- IBM Education Assistant
- WebSphere Application Server Version 5
- WebSphere Application Server Version 6
- WebSphere Application Server Version 6.1 Feature Pack
- Rational Application Developer Version 6
- WebSphere Extended Deployment Version 6
- WebSphere Process Server, WebSphere Integration Designer
- WebSphere Enterprise Service Bus, WebSphere Integration Designer
- WebSphere Adapters Version 6
- WebSphere Business Modeler Version 6
- WebSphere Business Monitor Version 6
- WebSphere Commerce
- WebSphere Partner Gateway Version 6
- WebSphere Developer for System z
  - V7.0
    - Enterprise application development and transformation
    - Web services development and transformation
    - User assistance
    - Technical preview
  - WebSphere Everyplace Deployment Version 6
  - WebSphere Host Access Transformation Services
  - WebSphere Message Broker Version 6
  - WebSphere MQ Version 6
  - WebSphere Studio Asset Analyzer
  - TXSeries for Multiplatforms Version 6
  - IBM Guided Activity Assistant
  - IBM Support Assistant

**WebSphere Developer for System z**

IBM Education Assistant

IBM WebSphere® Developer for System z  
Version: V7.0 Platform: z/OS  
Enterprise application development and transformation

Provide feedback on this material

**Enterprise application development**

- Creating a z/OS® connection
- Editing with the z/OS LPEX editor
- Customizing the z/OS LPEX editor
- Creating actions and menus with menu manager
- Creating actions and menus with menu manager - interactive actions
- Introducing the BMS features
- Introducing the MFS features

**Web Services Development and Transformation**

- Generating and deploying a single Web service (Bottom-up Scenario) (Part 1 of 2)
- Generating and deploying a single Web service (Bottom-up Scenario) (Part 2 of 2)
- Developing, deploying, and testing a Web service to automate CICS terminal application interactions (Part 1 of 2)
- Developing, deploying, and testing a Web service to automate CICS terminal application interactions (Part 2 of 2)
- Aggregating program and Web service invocations into a service flow (Part 1 of 2)
- Aggregating program and Web service invocations into a service flow (Part 2 of 2)
- Creating a database application using the z/OS® database application generator

Includes audio (items without audio might have audio added in the future)

## Summary

- The combination of Enterprise COBOL and Rational Developer for System z provides developers with the ability to
  - ▶ Quickly create and maintain traditional COBOL applications and dynamic Web applications and Web services using rapid application development tools
  - ▶ Reuse and transform existing applications to reduce costs, shorten the development cycle and move into an SOA development environment
  - ▶ Collaborate across the process of development, testing and deployment of multitiered, composite or mixed-workload applications
  - ▶ Leverage existing skills to write Web or COBOL applications and facilitate skill and knowledge transfer





# And now, time for your Questions!

© Copyright IBM Corporation 2007. All rights reserved. The information contained in these materials is provided for informational purposes only, and 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, these materials. Nothing contained in these materials 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 these materials 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 these materials 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. IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.





# Thank You for Joining Us today!

Go to [www.ibm.com/software/systemz](http://www.ibm.com/software/systemz) to:

- ▶ Replay this teleconference
- ▶ Replay previously broadcast teleconferences
- ▶ Register for upcoming events

