



Let's tie it all together and play in the sandbox





Agenda

08:40 - 09:40 - Build a smarter foundation for future investments

09:40 - 09:50 - Break (10 min)

09:50 - 10:50 - Smart Reuse- Transform green screens to Web, SOA, mobile, and portal

10:50 - 11:00 - Break (10 Min)

11:00 - 12:00 - Speed the development of multiplatform applications

12:00 - 01:00 - Lunch (1 hour)

1:00 - 2:00 - Developing Web 2.0 applications using Mashup Tools

2:00 - 2:10 - Break (10 Min)

2:10 - 3:10 - Smart Work on System z: Enhance teamwork with multiplatform SCM tools

3:10 - 3:20 - Break (10 Min)

3:20 - 4:20 - Let's tie it all together and play in the sandbox

3:20 - 4:30 - Close



Existing COBOL/CICS 3270 Application

```
Session A - [24 x 80]
File Edit View Communication Artinos Window Help
Client Inquiry - calls LAB3POT (LAB3BMS)

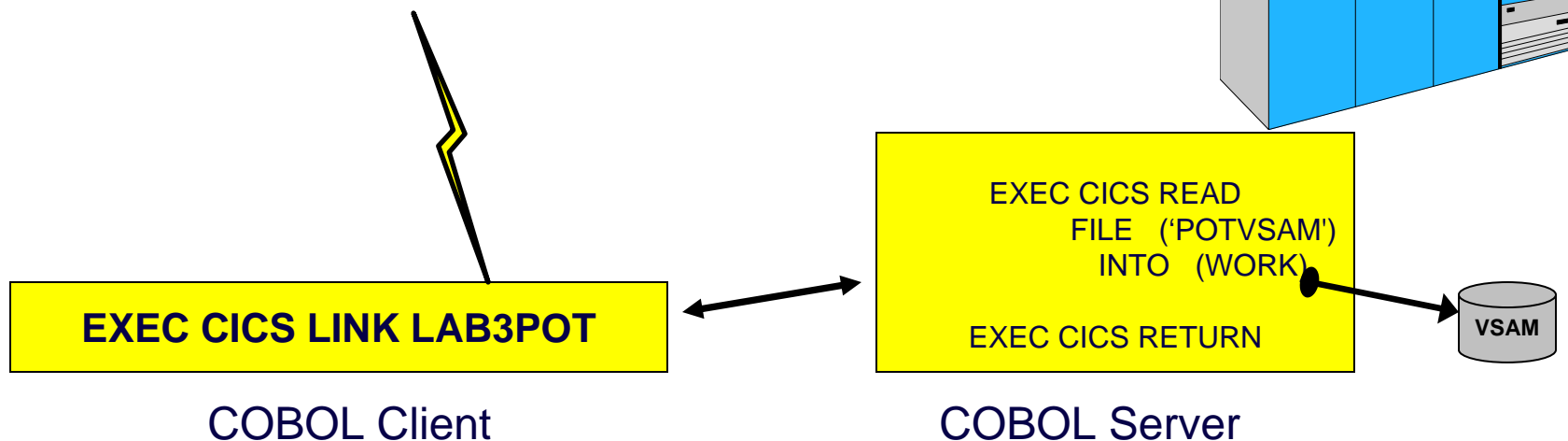
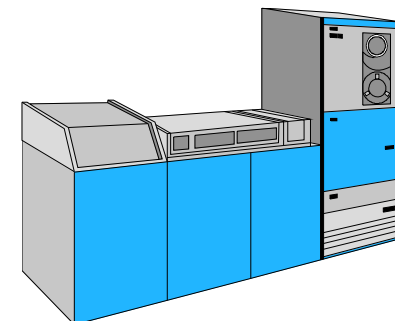
Customer number: 004
Last name: Silva
First: Lula da
Address: Palacio do Planalto
City: Brasilia
State: DF
Country: BRAZI

Type customer Number Between 1 and 10 or 99 to END

Customer retrieved sucessfully
```

z/OS Texas

ZSERVEROS

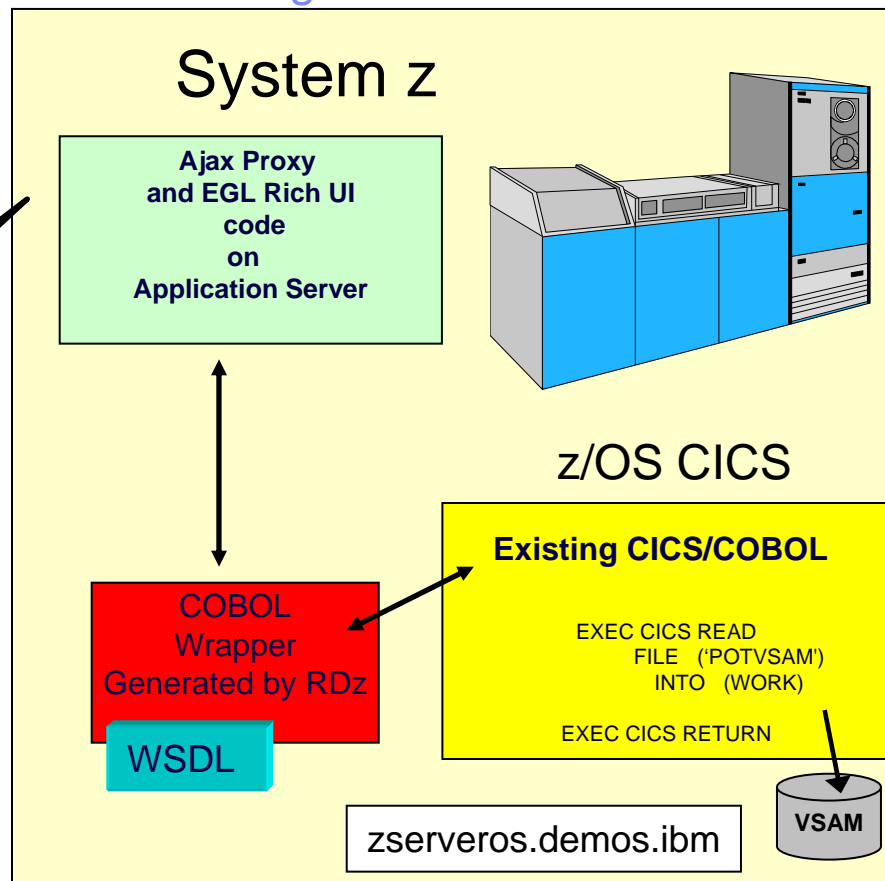


[Link to PCOM](#)
© 2010 IBM Corporation



Invoking CICS Web services from iPhone

Using COBOL/CICS/VSAM



rbarosa@us.ibm.com

<http://zserveros.demos.ibm.com:9080/iPhone/egl.html>

More at:

http://www.ibm.com/developerworks/websphere/techjournal/0909_barosa/0909_barosa.html



LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (RDz) (30 Minutes or less)

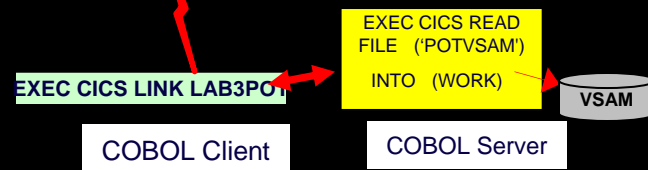
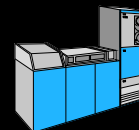
```

Session A [24 x 68]
Client Inquiry - calls LAB3POT (LAB3BMS)

Customer number: 004
Last name: Silva
First: Lúcia da
Address: Palácio do Planalto
City: Brasília
State: DF
Country: BRAZIL

Type customer Number Between 1 and 10 or 99 to END
Customer retrieved successfully
  
```

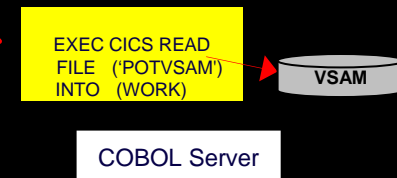
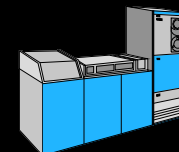
z/OS Texas



LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL) (30 Minutes or more – no optional)



z/OS Texas



You can choose both or just one



LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (RDz)

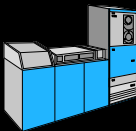
(30 Minutes or less)

```
Session A [24 x 98]
Client Inquiry - calls LAB3POT (LAB3BMS)

Customer number: 004
Last name: Silva
First: Lula da
Address: Palacio do Planalto
City: Brasilia
State: DF
Country: BRAZL

Type customer Number Between 1 and 10 or 99 to END
Customer retrieved successfully
```

z/OS Texas



EXEC CICS LINK LAB3POT

COBOL Client

EXEC CICS READ FILE ('POTVSAM') INTO (WORK)

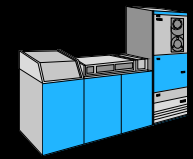
COBOL Server



LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL)



z/OS Texas



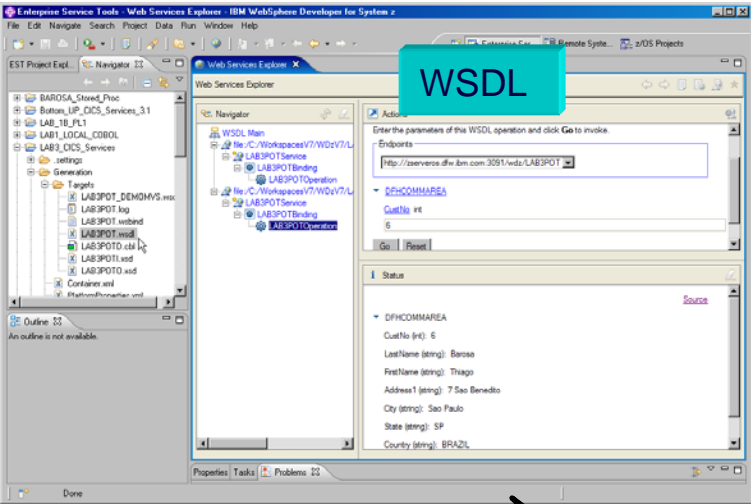
EXEC CICS READ FILE ('POTVSAM') INTO (WORK)

COBOL Server

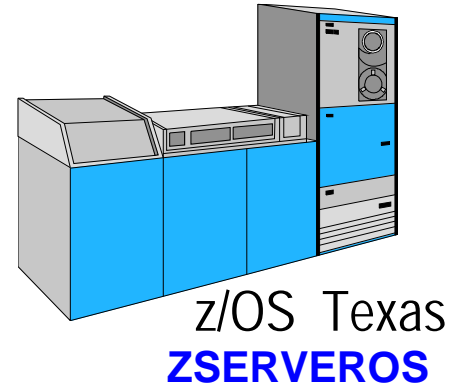




LAB #1 Create CICS Services, deploy test and debug using RDz



- Create the COBOL/CICS Web Services
- Test the Web services



z/OS Texas
ZSERVERS

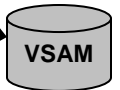
COBOL ADAPTER

Converter Driver

EXEC CICS READ
FILE ('POTVSAM')
INTO (WORK)

EXEC CICS RETURN

COBOL Server



[Link to RDz](#)

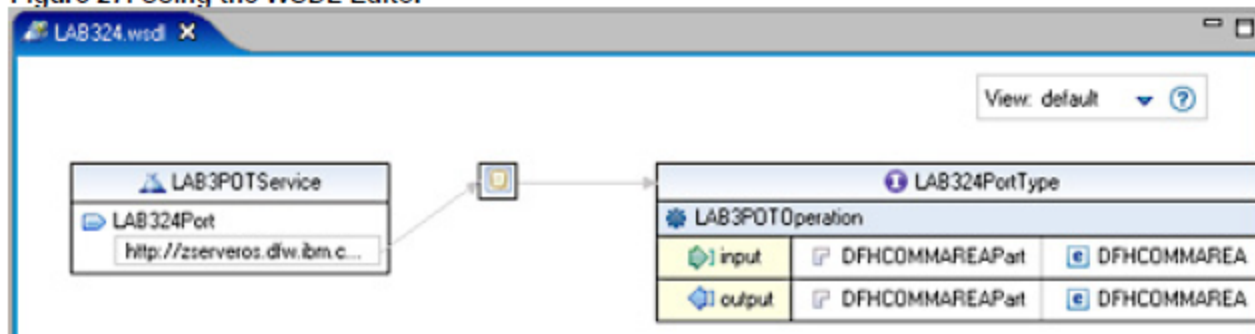
LAB1 #1 Create CICS Services, deploy test and debug using RDz

http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/emz/Wrap_existing_COBOL_program_as_Web_Services.pdf

IBM Enterprise Modernization for System z: Wrap existing COBOL programs as Web Services with IBM Rational Developer for System z

23. Verify the WSDL that was generated: double-click **LAB324D.wsdl** and use the WSDL Editor to visualize the input and output messages, as shown in Figure 27.

Figure 27. Using the WSDL Editor

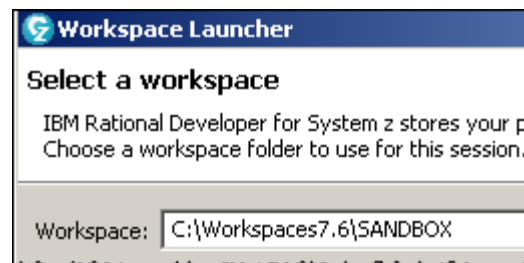


1. Start RDz

Start → Programs → IBM Software Development Platform for RDz7.6 → IBM Rational Developer for System z V7.6 → IBM Rational Developer for System z 7.6

or click on the icon in the windows quick launch:

2. Point to:

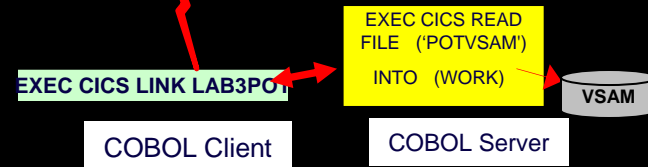
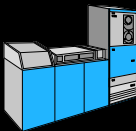




LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (RDz)



z/OS Texas

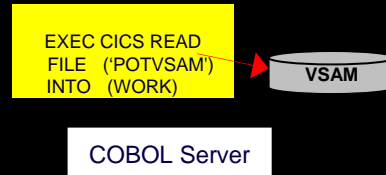
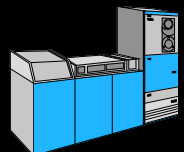


LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL)

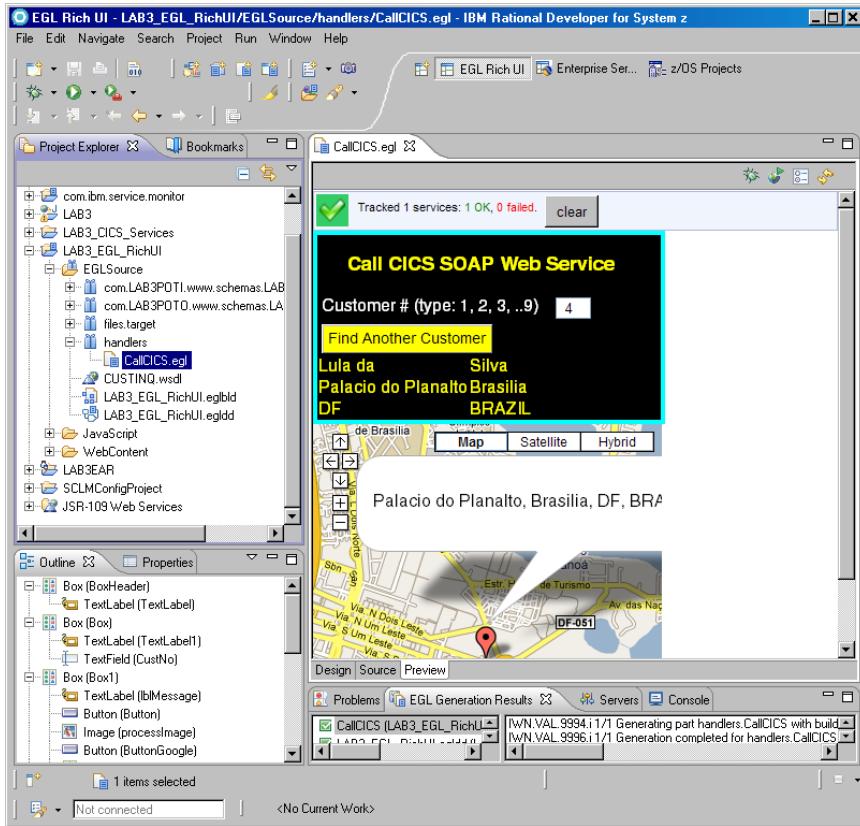
(30 Minutes or more – no optional)



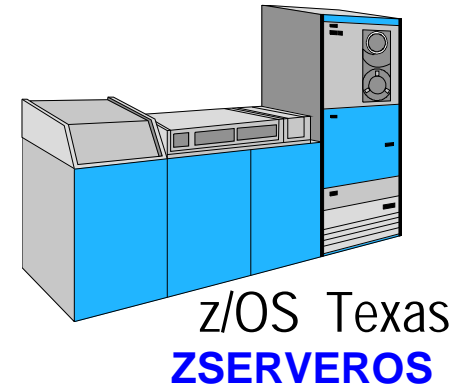
z/OS Texas



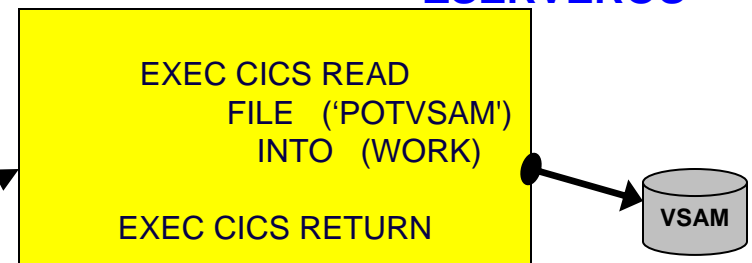
LAB #2 Using RDz with EGL create the Web 2.0 Interface



- Create the Rich UI interface with EGL
- Test/deploy the Rich UI



WSDL



COBOL Server



LAB #2 Using RDz with EGL create the Web 2.0 Interface

http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/emz/Accessing_Existing_CICS_VSAM_application_from_iPhone.pdf

IBM Enterprise Modernization for System z: Accessing existing CICS/VSAM application using an iPhone

Part 1 - Creating the service requester using Web 2.0 (User Interface).



1. Create a Rich UI EGL project and import the WSDL
2. Test the COBOL/CICS Service already deployed
3. Create the Rich UI EGL handler
4. Create the EGL Rich UI widgets
5. Create the EGL code to consume the CICS Service
6. Complete the EGL Rich UI code to invoke the CICS Service
7. Testing the Rich UI Application
8. Measuring the Service Call response time

Part 2 – Extending the Web 2.0 to invoke Google map services.

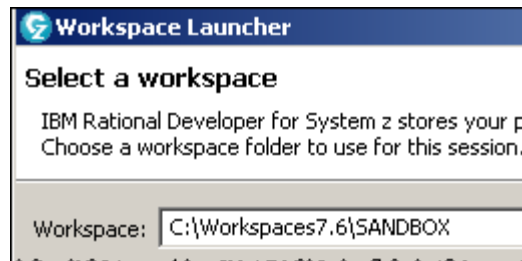
9. Modify the EGL Rich UI to add more controls and invoke the Google Map Service.
10. Testing using external Web Browser
11. Using an iPhone Browse emulator

1. Start RDz

Start → Programs → IBM Software Development Platform for RDz7.6 → IBM Rational Developer for System z V7.6 → IBM Rational Developer for System z 7.6

or click on the icon  in the windows quick launch: 

2. Point to:





Agenda

08:40 - 09:40 - Build a smarter foundation for future investments

09:40 - 09:50 - Break (10 min)

09:50 - 10:50 - Smart Reuse- Transform green screens to Web, SOA, mobile, and portal

10:50 - 11:00 - Break (10 Min)

11:00 - 12:00 - Speed the development of multiplatform applications

12:00 - 01:00 - Lunch (1 hour)

1:00 - 2:00 - Developing Web 2.0 applications using Mashup Tools

2:00 - 2:10 - Break (10 Min)

2:10 - 3:10 - Smart Work on System z: Enhance teamwork with multiplatform SCM tools

3:10 - 3:20 - Break (10 Min)

3:20 - 4:20 - Let's tie it all together and play in the sandbox

3:20 - 4:30 - Close



Next Steps: How to get started?

Resources to help kick-start your Enterprise software innovation

System z Sandboxes

Example assets and best practices providing low-risk, practical, hands-on path to leveraging IBM solutions

- ▶ Full version software trials
- ▶ 'Try online' hosted System z environments
- ▶ Hands-on-exercises

System z Starter Solutions

Solution to help you incrementally evolve core IT systems towards modern architectures and technologies

- ▶ Discover, reuse and grow
- ▶ Analyze and modernize
- ▶ Develop and manage
- ▶ Test and track

*Jump-start
your
modernization
projects!*

Education Series

Modern Application Architecture for COBOL Developers

Learn how to design and integrate composite applications across CICS and WebSphere – leveraging existing COBOL processes

EGL Distance Learning

8 days of training free electronically - lectures, labs, and electronic support



Learn more at:
www.ibm.com/rational/modernization



<http://w3-103.ibm.com/software/xl/portal/viewcontent?type=doc&srcID=R9&docID=X983555G31195K78>



QUESTIONS



Thank
YOU



Bank Scenario – Originals slides

(not Ready as Jan 26,2010)





Scenario: JKHL Bank Organization Overview and Objectives

Who we are

- JKHL Bank is a tier 2 bank, who, over the last several years has acquired 5 smaller regional banks to create a broader geographic presence.
- Our business started in 1935, we have a well recognized brand and a solid brand image. By 1999, we had 500 branches in 6 countries providing retail, corporate, commercial and private banking.
- We have achieved remarkable success in migrating banking transactions from branches to alternative distribution channels. Over 50% all our banking transactions are carried out via alternative channels including ATMs, Kiosks, internet and the call center, a ratio that is above most banks.
- We win in the marketplace with a high-touch approach to our customers. We pride ourselves with understanding our customer segmentation and providing differentiated service based on customer profitability. We are known for our innovative products

Who we want to be

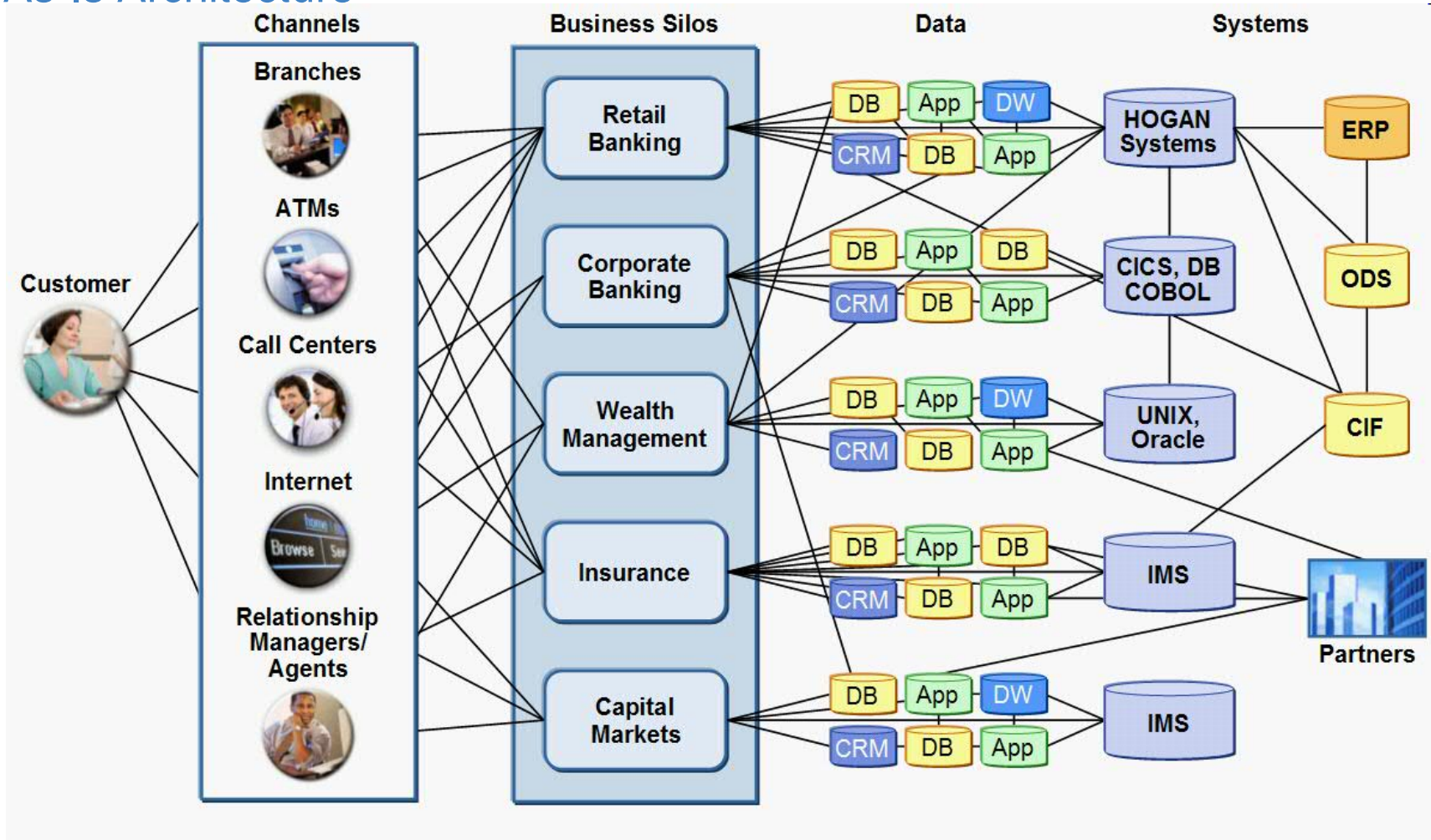
- In the short term, given recent changes in financial markets, we are reducing costs and leveraging existing assets and processes, while at the same time, **providing robust capabilities to manage dynamic issues of risks as well as regulatory compliance**
- In the long term, we want to **achieve differentiation** through improved customer experience and innovative self-service offerings
- We want to show that we really **know our customers** through the use of integrated data and analytics, and by having more meaningful customer interactions that can develop profitable customers, who go beyond loyalty to become advocates.
- We want to build on our ability to deliver **innovative products** and services as well as provide a seamless experience across the different LOBs and channels



In order to make their banking operations more profitable, JKHL Bank recruits IBM to analyze its existing business processes and provide recommendations for a business transformation.



As-Is Architecture



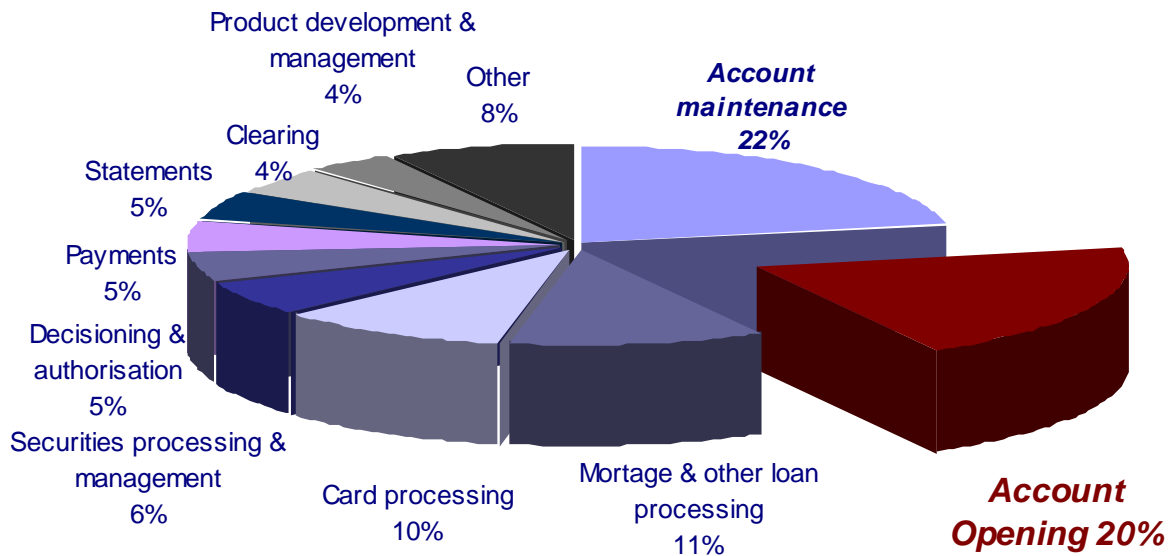
- Inconsistent View of Customer
- Information Locked in Repositories
- Applications are hard-wired
- Silos of Information
- Inconsistent Data
- Poor Channel Communication



Why JKHL Considers Account Opening to be a Key Initiative

Cost Breakdown of a Bank's Activities

Total Manufacturing & Processing Cost Base

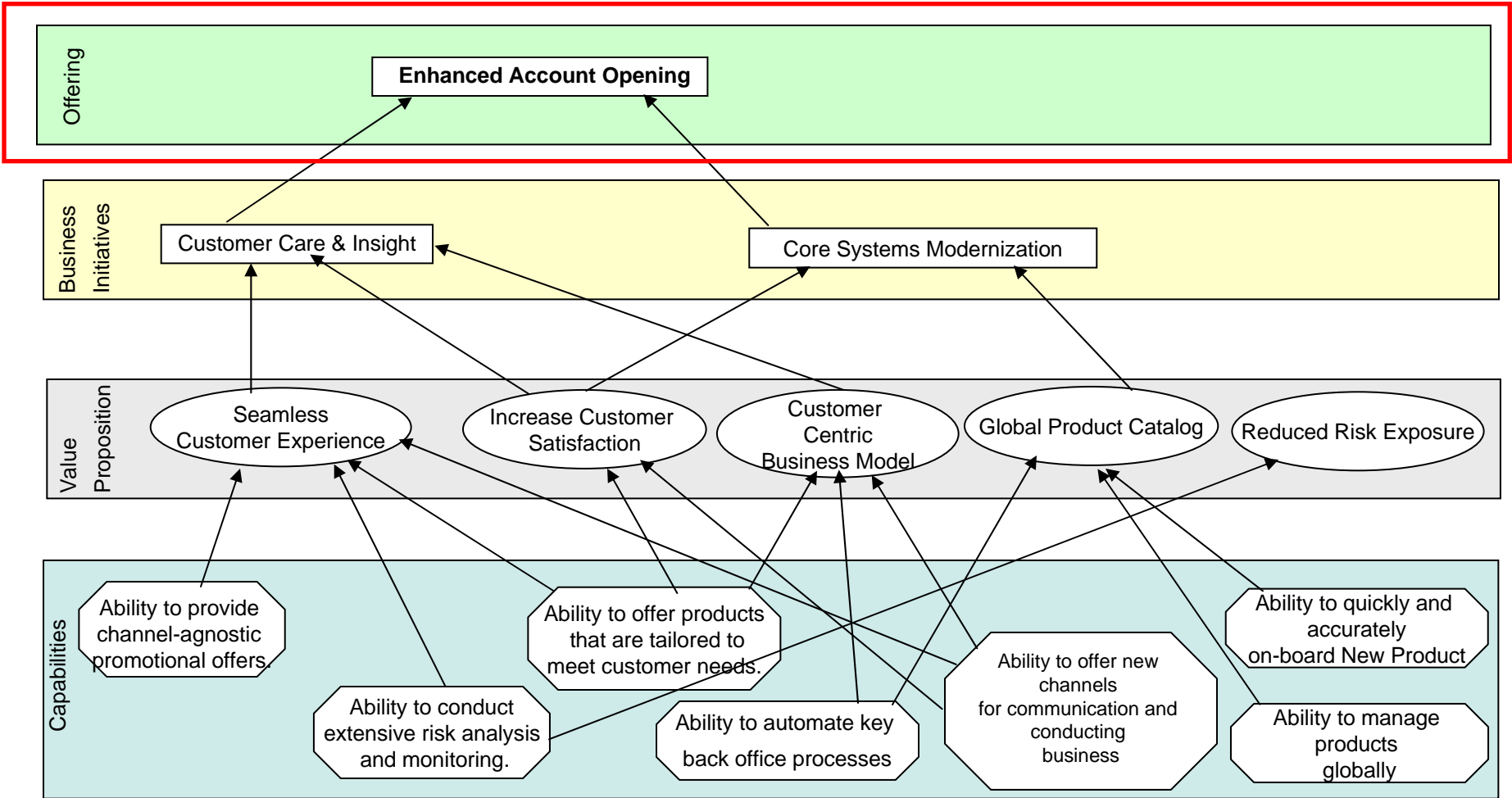


- 1 Account Opening is a major item of the cost base**
- 2 Duplication of Efforts and Systems put pressure on AO costs**
- 3 Maintaining interfaces among disparate systems is the major reason behind costs**

Source: IBM Banking Industry Thought Leadership products; IBM client engagements



Capability Model Ties Business Capabilities to Initiatives



IBM verifies that the core capabilities which JKHL Bank identified as being vital to the company, are mapped to its key business initiatives, which in turn is addressed by the company's enhanced offerings.



JKHLE - Current State Assessment / General Observations

Accounts:

- **Account information is duplicated and inconsistent within multiple systems and LOBs**
- **All pre-requisites for a new account cannot be satisfied via any channel (of choice)**
- **Promotions are not consistently offered across delivery channels**
- **Limited risk management during account open process**

Customers:

- **Lack of a single presentation of information across channels**
- **Unable to exchange information with bank via channel of choice**
- **Customers are forced to re-supply personal information for every new account they open with the bank**
- **Customers do not understand why all accounts and services are not available using a single user interface**

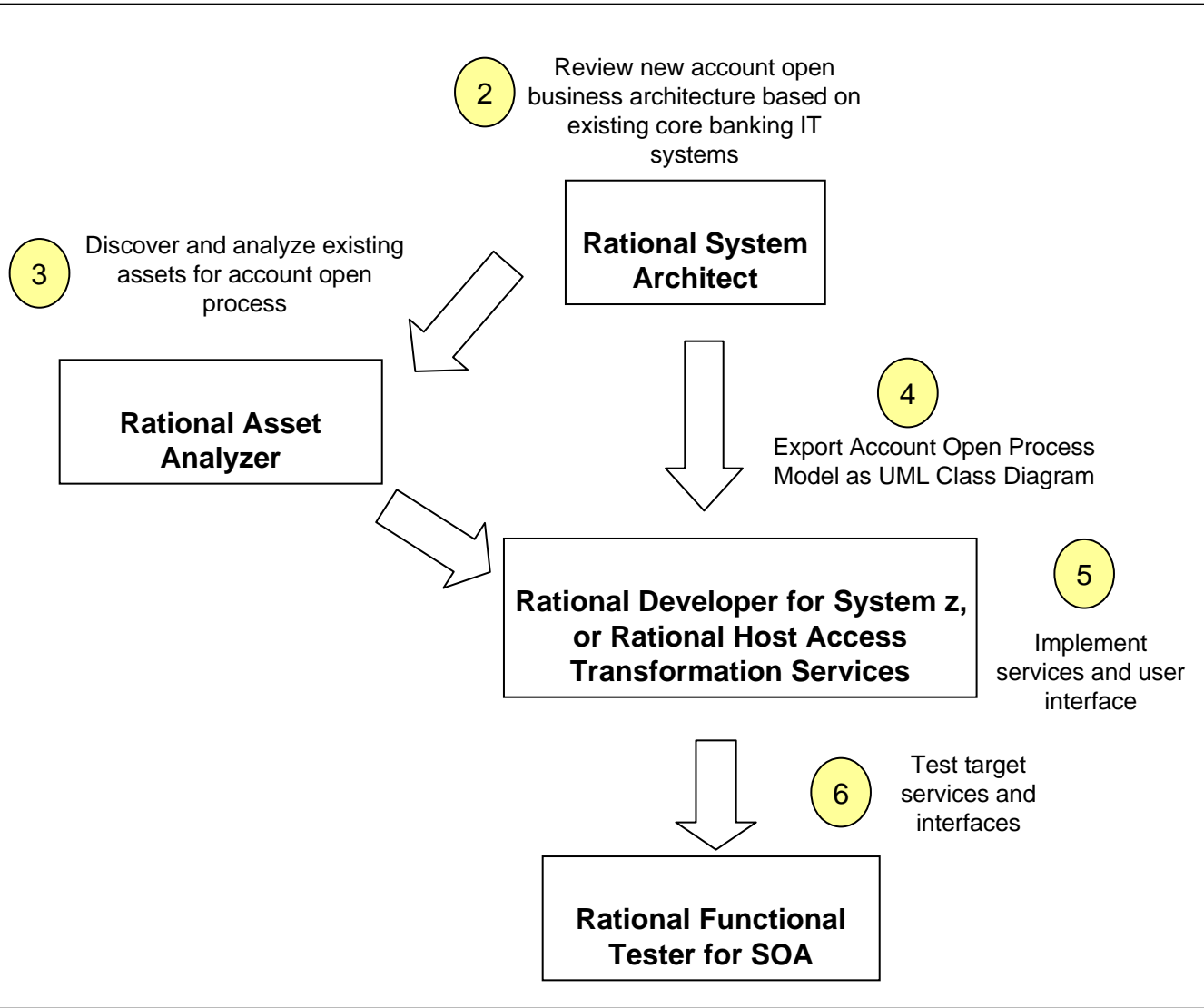
Systems:

- **IT systems designed to support various channels and lines of business as separate systems Web system, Catalog/Call Center system and Store system are on very different infrastructures**
- **Many legacy applications**
 - Less flexible than newer technologies – monolithic, home grown applications which are not easily adaptable to open standards
- **Numerous technologies supported**
 - Some supported long past end of life
 - Attempts by a few IT departments within the company to adopt SOA
 - No enterprise-wide SOA strategy
- **Some use of newer, web-based technologies**



Rational Banking Modernization

Review project requirements,
Manage work items, source code
changes, deployment of artifacts to
Mainframe



Rational Requirements Composer,
Rational Team Concert for System z,



Labs



Backup





Use scenarios to uncover customer needs

1. Brief Description
A registered customer has applied to open a new **Account**. Based on existing **Customer** information, the eligibility system will use predefined heuristics to determine whether or not the customer is eligible. In circumstances, intervention by the **Branch Manager** may be necessary.

2. Basic Flow of Events
The bank's web portal has validated the contents of a new **Account** application. The system is retrieving banking records from the bank's files. Depending on the results, third parties may also be consulted (eg: the credit bureau). The system applies its predefined heuristics to the available data based on the customer's history, **Credit History** etc. Based on this information the system will either approve or automatically reject the application. The application has been either approved or rejected.

Active Flows
This flow is related to a single area of functionality (for example specialist handling for the **Withdraw Cash** use case of an Automated Teller Machine). Conceptually related sets of flows are grouped into their own clearly defined sections.

Outline flows with **Process Diagrams**

Describe flows and capture requirements in **Rich-text Documents**
Define and reuse common terms in **Glossaries**

Describe actors, system boundaries and user goals in **Use Case Diagrams**

ClassicsCD Header with Logo

Shopping Cart
Cart is empty

Catalog
Shopping Cart
Cashier
Order Status

Search Criteria
 Composer
 Composition
 Performer

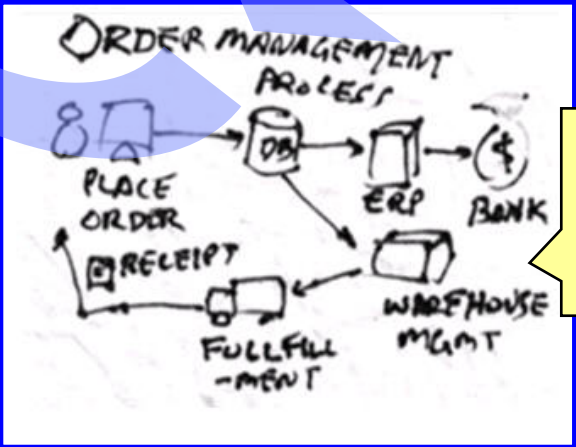
[Bach: Brandenburg Concertos 1 + 3](#)
Burlington Symphony Orchestra, Wilhelm Storrer

[Bach: Violin Concertos](#)
Burlington Chamber Orchestra, Torrence Spang

[Beethoven: Symphonie Nr. 7](#)
The Burlington Orchestra, Wilhelm Storrer

[Beethoven: Symphonie Nr. 5](#)
Burlington Symphony Orchestra, Wilhelm Storrer

Visualize scenarios with **Storyboards and Screen Flows**



Use whiteboard snapshots and other **Informal Documentation**

Team Concert Work items

Subscribe to work items you're interested in

Predefined, custom and personal queries

Understands and persists work items' relationship to SCM and build artifacts

Integrated discussion threads

Query results

The screenshot displays the Team Concert Work Items application. The main window shows a 'Defect 12' with the following details:

- Summary: UWS Add JavaDoc to CLI code
- Type: Defect
- Severity: Normal
- Found In: Unassigned
- Created: Jun 6, 2007 5:11 PM
- Created By: Jerry Jazz
- Team Area: UWS Temperature Conversion Team /
- Filed Against: Core/Command Line
- Owned By: April Blues
- Priority: Unassigned
- Planned For: -> 1.0 M1
- Estimate: 1 h
- Time Spent:
- Due: None
- Resolved: Jan 24, 2008 4:30 PM
- Resolved By: April Blues

The 'Description' field contains: "We need to add JavaDoc to our code. Note that I have sized this at one day and targeted it for M1. This means do it now while Zach gets the build going but only spend at most this much time on it, maybe less. We just need to get a start on it now."

The 'Discussion' section shows a comment: "We have decided to make the change|"

At the bottom, a table shows query results:

| I | Status | P | S | Summary | Owned By | Created By |
|----|----------|---|---|--|-------------|------------|
| 12 | Resolved | | | UWS Add JavaDoc to CLI code | April Blues | Jerry Jazz |
| 10 | Resolved | | | Expose functionality via public get method to core temperature ... | Unassigned | Jerry Jazz |

RAAi Control Flow Model

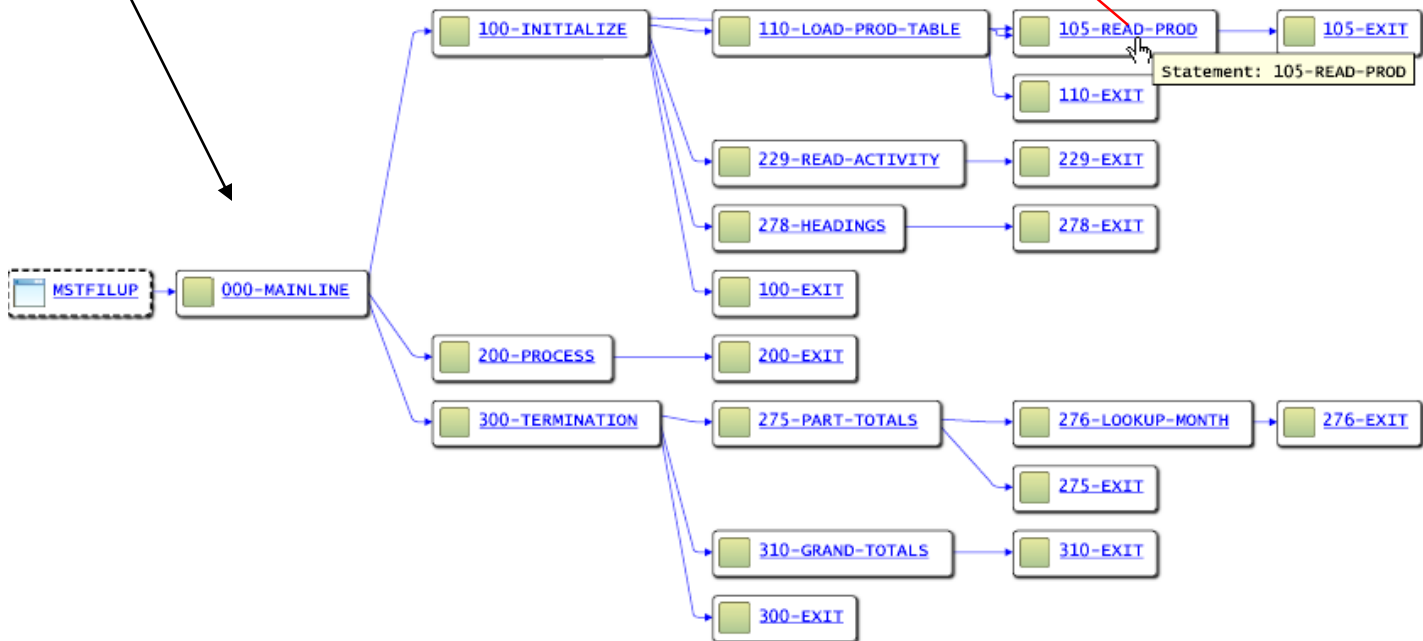
- Graphically reveals program's procedural structure

```

Line 359      Column 22      Insert
-----+*A-1-B-+-----2-+-----3-+-----4-+-----5-+-----6-+-----7-+-----8
033500  105-READ-PROD.
033600      READ PROD-LINE-FILE-IN
033700          INTO WS-PROD-LINE-REC
033800              AT END MOVE 'Y' TO SW-EOF-PROD.
033900      IF NOT EOF-PROD THEN
034000          ADD 1 TO AC-PROD-IN.
034100  105-EXIT.
034200      EXIT.
034300*
034400  110-LOAD-PROD-TABLE.
034500      MOVE PLF-PROD-LINE TO PLT-PROD-LINE (PROD-NDX).
034600      MOVE PLF-MODEL     TO PLT-MODEL (PROD-NDX).
034700      SET PROD-NDX UP BY 1.
034800      PERFORM 105-READ-PROD THRU 105-EXIT.
034900  110-EXIT.
035000      EXIT.
035100*
035200  200-PROCESS.
035300      SORT SORT-WORK-FILE
035400*          ASCENDING KEY SWF-PARTNO
  
```

- Can click a paragraph name and source code within the LPEX editor aligns

- Provides unbeatable combination of "top-down" and "bottom-up" code learning model



RAAi – Search for Variables References and Modifications

View source (default)
 Show program diagram
 Show control flow diagram
Show data element table
 RAA Details

| Name | Level | Type | Physi |
|-------------------|-------|------|-------|
| NEW-MF-RECORD | 1 | CHAR | |
| NEW-PARTS-MF-OUT | 0 | FD | |
| NEW-REC-WAITING | 88 | UNKN | |
| OLD-MF-RECORD | 1 | CHAR | |
| OLD-PARTS-MF-IN | 0 | FD | |
| OMF-ACCOUNT | 5 | CHAR | |
| OMF-DAY | 10 | CHAR | |
| OMF-LAST-ACTY | 5 | GRP | |
| OMF-MON | 10 | CHAR | |
| OMF-PARTNAME | 5 | CHAR | |
| OMF-PARTNO | 5 | CHAR | |
| OMF-PAY-AMT | 5 | NUMB | |
| OMF-PAY-QTY | | | |
| OMF-PROD-LI | | | |
| OMF-REC-AMT | | | |
| OMF-REC-QTY | | | |
| OMF-YR | 10 | CHAR | |

Show sorted X-Ref of program variables
 When found can use to answer critical and important Maintenance Questions

View source (default)
Find
 Impact Analysis
 RAA Details

References
 Modifications
References and Modifications

```

Line 470      Column 42      Insert
-----+*A-1-B-+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7--|+-----8
044300*
044400 255-PROCESS-MF.
044500 IF AF-PARTNO = OMF-PARTNO THEN
044600     PERFORM 260-UPDATE-OLD-MASTER          THRU 260-EXIT
044700     PERFORM 265-PREP-PRINT-LINE            THRU 265-EXIT
044800     PERFORM 280-RETURN-ACTIVITY-RECORD    THRU 280-EXIT
044900 ELSE
045000 IF AF-PARTNO > OMF-PARTNO THEN
045100     PERFORM 290-WRITE-NEW-MF                THRU 290-EXIT
045200     PERFORM 275-PART-TOTALS                THRU 275-EXIT
045300     PERFORM 285-READ-OLD-MASTER            THRU 285-EXIT
045400 ELSE
045500     PERFORM 262-CREATE-NEW-MF              THRU 262-EXIT.
IF AF-PARTNAME = OMF-PARTNAME
THEN
    
```

- New Connection
- jont60p.Local
- jont60p.demomvs
- jont60p.zservers
- jont60p.myWebSphe**

OMF-PARTNO: 5 Data Element Occurrences Found

DemoProj
 BatchCobol
 mstfilup.cbl (5 Matches)

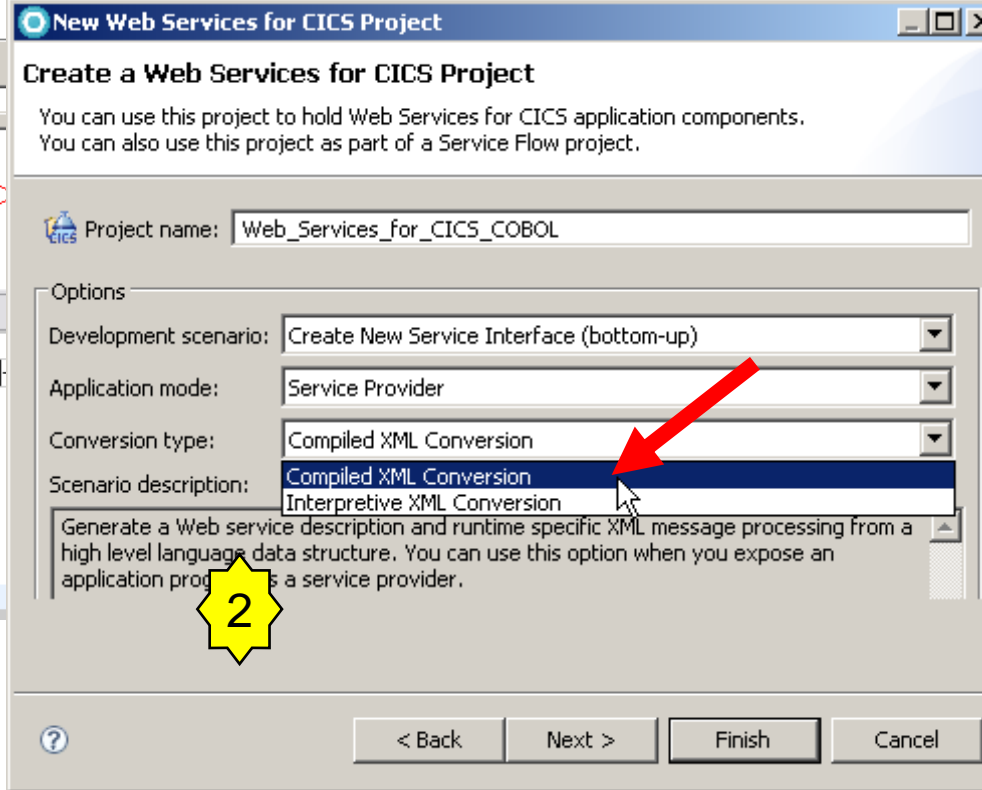
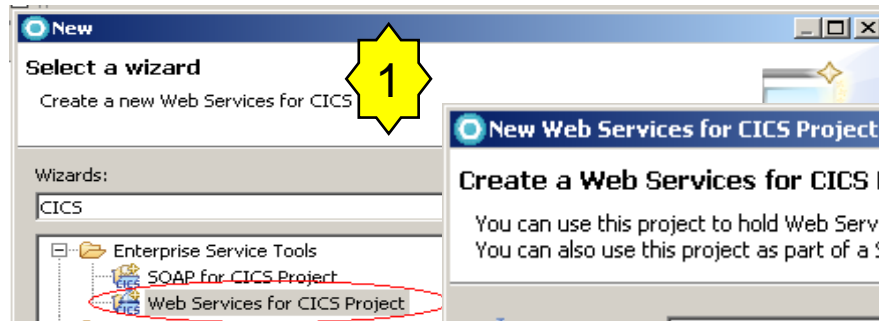
Show Next Match (Ctrl+.)



Web Services for CICS 3.x

Example Bottom-up **Compiled** XML Conversion - 1 of 2

Web service
WSDL &
Converters



Existing
COBOL or PL/I

```

COMAREA2.cpy
-----+*A-1-B-----2-----3-----+
001      01  DFHCOMMAREA.
002          02  CustNo      PIC S9(9)
003          02  LastName   PIC A(25)
004          02  FirstName  PIC A(15)
000005     02  Address1    PIC X(20)
000006     02  City        PIC A(20)
000007     02  State       PIC A(5)
000008     02  Country     PIC X(15)
000009     02  RetCode     PIC S9.
    
```

Bottom-up

CICS Web Services **Compiled** Conversion – Example – 2 of 2

RDz XML Enablement (Bottom-up)

3

4

5

→ Select language structures from a complete COBOL program or a COBOL copybook included by the program

→ Syntax check of COBOL source during import.

COMAREA2.cpy

| Line | Column | Insert |
|--------|--------|---------------------------|
| 000001 | 01 | DFHCOMMAREA. |
| 000002 | 02 | CustNo PIC S9(9) COMP-5 . |
| 000003 | 02 | LastName PIC A(25) . |
| 000004 | 02 | FirstName PIC A(15) . |
| 000005 | 02 | Address1 PIC X(20) . |
| 000006 | 02 | City PIC A(20) . |
| 000007 | 02 | State PIC A(5) . |

Web Services for CICS - Create New Service Interface (bottom-up)

Language structures

The language structures have been imported. Select the inbound, outbound or both language structures.

Inbound Language Structure

- WS-PROGRAM
- WS-LITERAL-WS
- WORK-VARIABLES
- POTVSAM-RECORD-REC
- WS-LITERAL
- DFHCOMMAREA
 - CustNo
 - LastName
 - FirstName
 - Address1
 - City
 - State
 - Country
 - RetCode

Web Services for CICS - Create New Service Interface (bottom-up)

Language structures

The language structures have been imported. Select the inbound, outbound or both language structures.

Inbound Language Structure

- WS-PROGRAM
- WS-LITERAL-WS
- WORK-VARIABLES
- POTVSAM-RECORD-REC
- WS-LITERAL
- DFHCOMMAREA
 - CustNo
 - LastName
 - FirstName
 - Address1
 - City
 - State
 - Country

EST Project Explorer

- Generation
 - Container.xml
 - PlatformProperties.xml
 - ServiceSpecification.xml
- Targets
 - LAB324.wsdl
 - LAB324D.cbl
 - LAB324I.xsd
 - LAB324O.xsd
 - LAB3POT.log
 - LAB3POT.wsbind



Recording Scripts

Verification Points

| Software |
|----------------------------|
| Trials and betas |
| Support |
| Training and certification |
| Library |
| Events |
| News |



Functional Tester Sees Data

You See...

| Property | Value |
|---------------|---|
| .bounds | Rect[84,100,787,809] |
| .class | Html.HtmlDocument |
| .cookie | IBMSurveyTest=isEnabled |
| .offsetHeight | 809 |
| .offsetWidth | 787 |
| .text | IBM Rational Software Home |
| .title | IBM Rational Software |
| .url | Href(http://www.ibm.com/software/rational/) |

Functional Tester Sees Properties

- Automated Validation
 - Functional Tester captures data and properties that can be invisible to users
 - During script execution, current results are compared to stored baselines
 - Discrepancies are flagged and reported to user in an HTML based test log

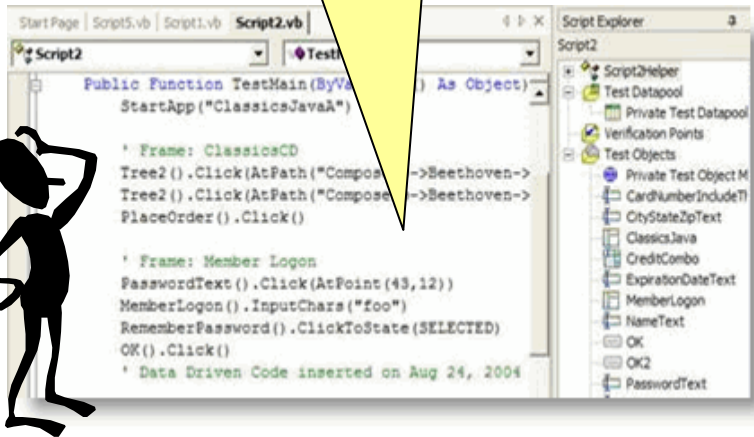


Simplified Test Representation

Scenario: My business SME must contribute to the QA activities

Today the business user is overwhelmed by technical depth

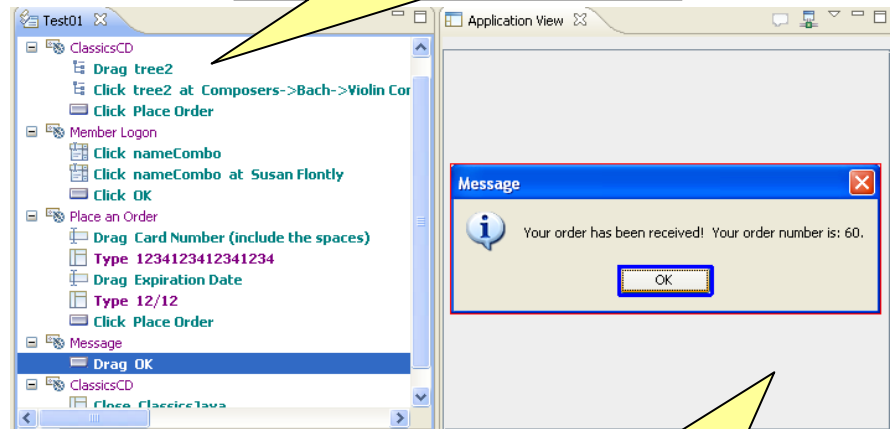
Business user is faced with a programming language ...



... and 'Objects' and other advanced options

With 8.1 we will speak the language of the business user

Natural language statements grouped by window/screen touched



Application view synchronized with group/step in the Test View