A Fresh Look at the Mainframe

Productive Tools for
Application Development

ODI Needs Productive Developer Tools

We have a lot of new solutions to build. If I build them all on System z, I will have to find more System z development skills

On Demand Insurance
CIO

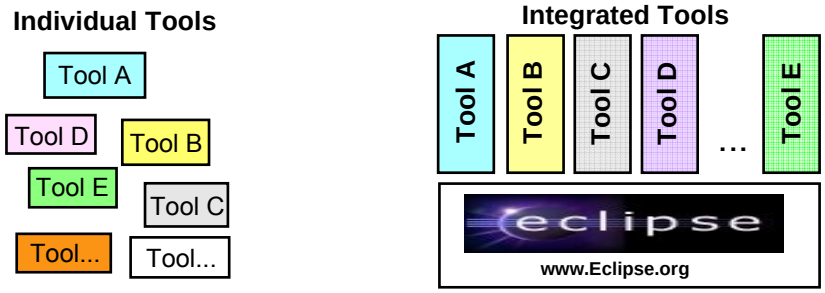
With modern tools like WebSphere Developer for zSeries, you can double the productivity of your existing team

IBM

WebSphere Developer for zSeries (WDz)

- IBM's latest tool for System z development
  - Version 6.0.1 announced Nov 2005
- Single integrated tool for developing mainframe applications using both traditional and newer technologies
  - Develop traditional COBOL/PL1/ASM/JCL/BMS/EGL applications
  - Develop Java/J2EE and Web applications, including JSF and struts
- Boosts developer productivity compared to green screen tools
  - Workstation tool based on Eclipse
- Leverage existing assets in a Service Oriented Architecture
  - Quickly expose CICS transactions as native services
  - Visually wire together CICS transactions into a business flow

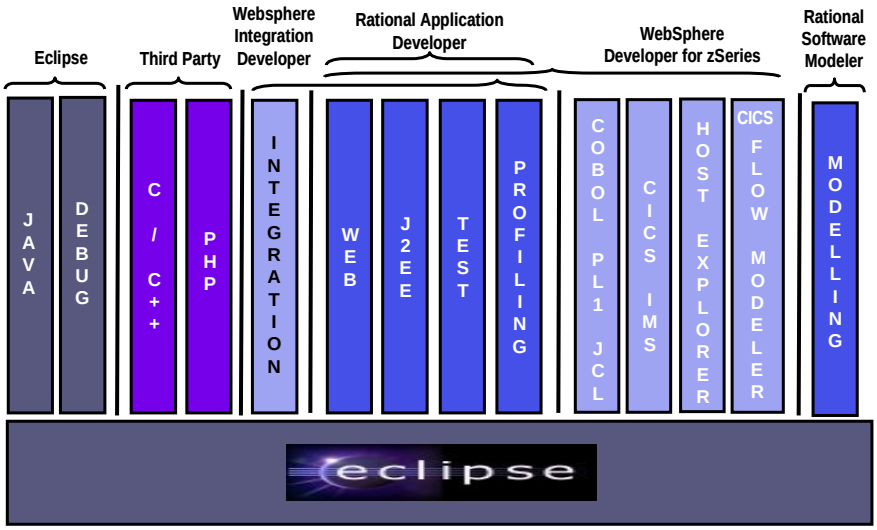
WDz is Built on Eclipse
Eclipse Provides an Integrated Toolset



- Challenges:
  - Each task requires different tool
  - Multiple repositories
  - Multiple skills required for each tool
  - Different look and feel
  - Different semantics
  - Difficult to manage
  - Duplicate functions
- Benefits:
  - Tools integrated on open source platform
  - Consistent User Interface
  - Same semantics
  - Supports multiple roles
  - Common Repository between all tools
  - Pluggable tools framework
  - Integrated Test Environment

IBM Tools for SOA Development

Function delivered as perspectives in eclipse



WDz Highlights for Traditional Development

- Interactive, workstation-based environment
  - Interactive access to z/OS
  - Work with host files as though they are local
- Boosts developer productivity compared to “green screen” tools
  - Interactive edit/compile/debug on the workstation
    - Work offline or online
  - TSO commands, job generation, submission, remote debug to z/OS
  - Language sensitive editors

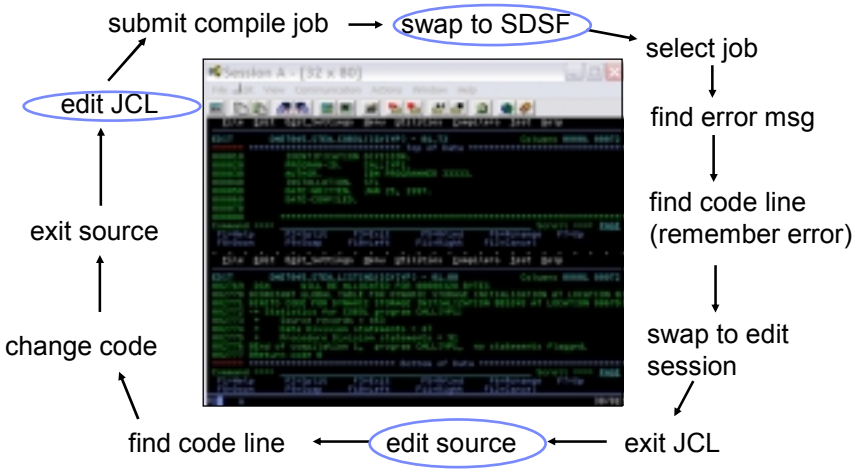
Interactive Access to z/OS

- Connect to z/OS system using Remote System Explorer
- Work with host libraries and datasets
- Job generation, submission, and monitoring
- TSO/USS command execution
- Mainframe integration:
  - Issue TSO commands or jobs from desktop
  - Full screen 3270 access
- Host Source Code Management (SCM) access through CARMA (Common Access Repository Manager)
  - Unified interface for accessing different host based SCM tools

DEMO: Introducing WebSphere Developer for zSeries

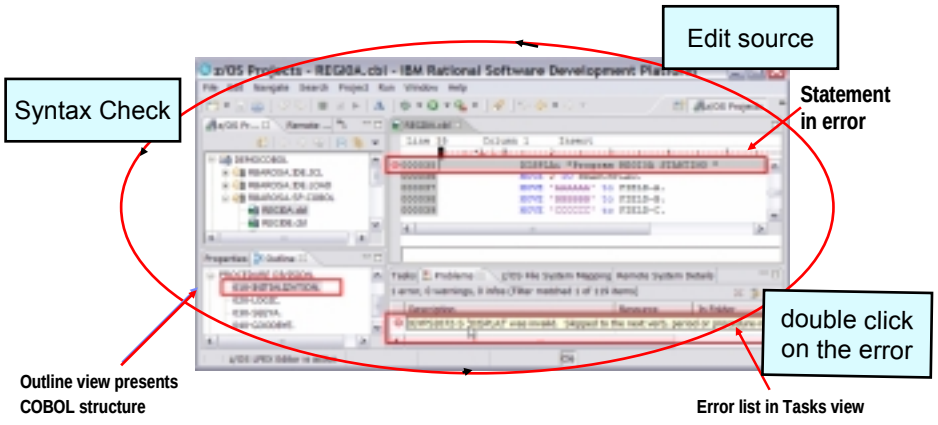
- Perspectives and views
- Remote System Explorer
- Working with host datasets
- Language-sensitive COBOL editor

ISPF: How Does Traditional "Green Screen" Development Work?



- Programmer goes through a sequence of screens in order to get the job done
  - 3.4 listings, job listings, SDSF outputs, etc.
- Programmer is constantly flipping back and forth between these ISPF screens

Simplified Development with WDz Perspectives

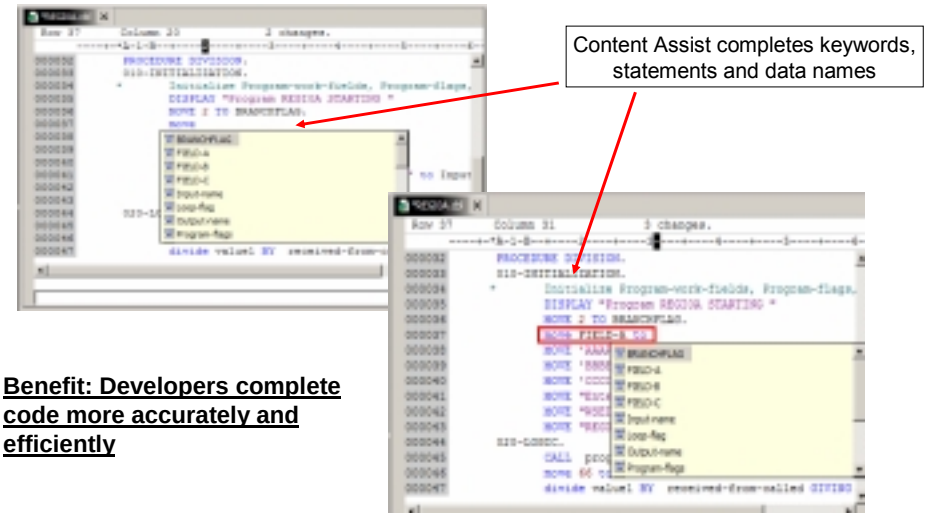


Benefit: Simplified development for COBOL and PL/1 through intuitive, interactive screens

WDz Advantages Over Traditional Host-based Tools

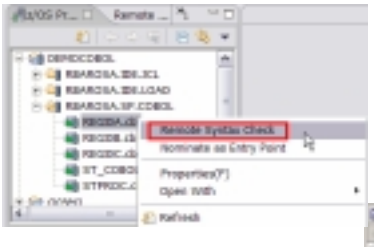
- Do multiple concurrent activities
  - With WDz, developer can multitask, all of the required perspectives are readily available
  - ISPF allows a maximum of 3 concurrent activities
    - No such limit in WDz
- Capabilities of a GUI editor
  - Color coding, Content assist, Outline view ..and many others
- Visual debugging
  - Identify problems quickly and intuitively

Productivity Features for COBOL, PL/1 Programmers: Content Assist

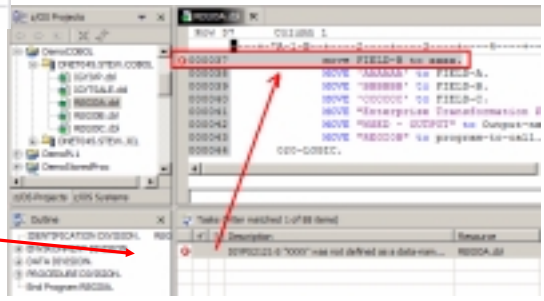


Benefit: Developers complete code more accurately and efficiently

Productivity Features for COBOL, PL/1 Programmers: Syntax Checking



Do local or remote syntax checking..

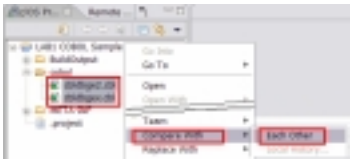


Benefit: Developers don't have to wait till compile time to check syntax

Just double-click to find the error

Productivity Features for COBOL, PL/1 Programmers: Compare Two Files

Underlying Eclipse Productivity Features are Available for z/OS Assets…

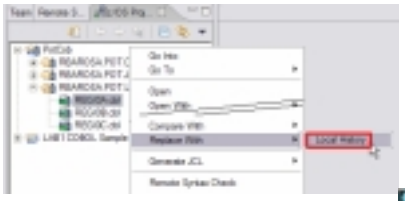


Navigate through and merge differences using icons

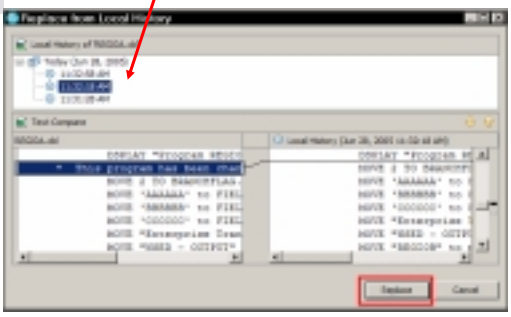
Benefit: Productivity maintaining COBOL/PL1/JCL files

Productivity Features for COBOL, PL/1 Programmers: Replace with Local History

Underlying Eclipse Productivity Features are Available for z/OS Assets…



Keep as many local versions as you want and compare with the z/OS current version..



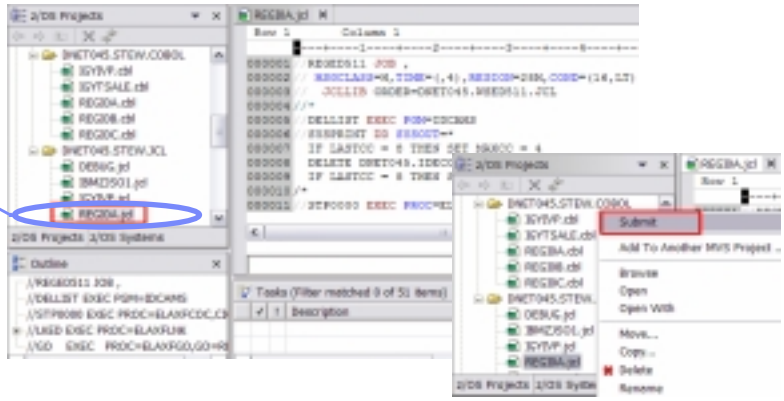
Benefit: Productivity maintaining COBOL/PL1/JCL files

DEMO: WebSphere Developer for zSeries Productivity Features

- Content assist
- Syntax checking
- Compare two files
- Local history

Productivity Features for COBOL, PL/1 Programmers: Generate JCL and Submit to z/OS

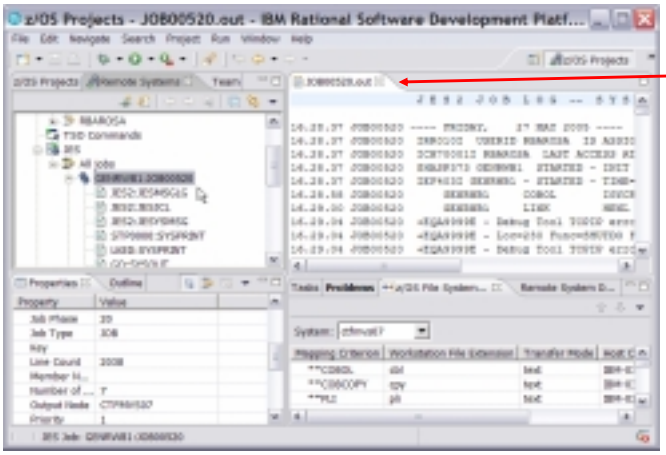
JCL generated from COBOL or PL/1 Code



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

Productivity Features for COBOL, PL/1 Programmers: Monitoring Job Output

Output can be seen like local files.. Can use Find, etc..



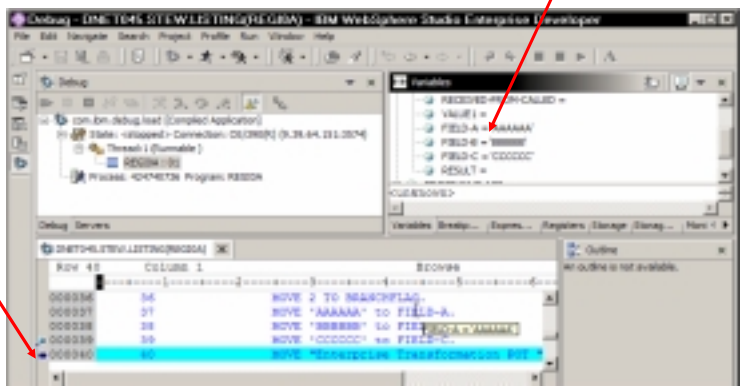
Benefit: Developers do not have to continually switch between systems to use SDSF. No TSO need.

Productivity Features for COBOL, PL/1 Programmers: Remote/Local Debug

- Same Debug Perspective used for COBOL, PL/1 & Java, etc..
- WDz is the ONLY product in the market that does END-to-END debug.. you can debug HTML --> JSP --> Java --> COBOL on z/OS... in the same IDE

Change contents, etc..

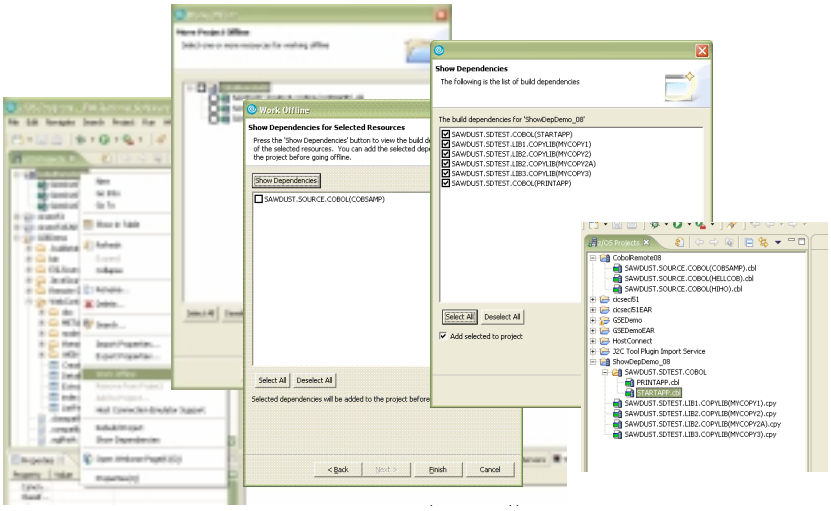
Breakpoints, watchpoints, Jump to, Run to etc..



Benefit: Consistent debugging environment for COBOL, PL/1, Java

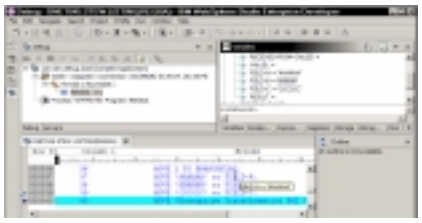
Enhanced Offline Capabilities

- Work offline or online
- Show dependencies
- Wizard to bring all dependent files that need to be in the project



DEMO: Developing a Host Application Using WDz

- Edit/Compile/Debug
- Show use of WDz productivity features
  - Write COBOL code, use Content Assist, syntax Check
  - Generate JCL for Compile-Link-Go, submit
  - Use Visual Debugger to walk through code as it executes



Productive Developer Tools for SOA Solutions

How about support for newer technologies like Web pages, Web Services and SOA?

On Demand Insurance CIO

Remember, we already used WebSphere Developer for zSeries to help build parts of that external broker solution. Let's review what it can do.

IBM

WDz Highlights for Web and SOA Solutions

- SOA and Web Services Development
  - Quickly expose CICS transactions as services
    - XML Services for the Enterprise
  - Visually wire together CICS transactions into a business flow
    - Service Flow Modeler
- Web UI Development
  - Intuitive, drag-and-drop visual construction of user interfaces based on open standards
    - JSF, SDO and Struts

Web UI Development

- Leverage the full web development capabilities of Rational Application Developer
  - Tools to develop, test and deploy standards-based web applications
- Visual, interactive Web development
  - Develop Static and Dynamic Web pages
    - HTML, Java Server Pages, J2EE
    - Templates, Stylesheets, Web Diagrams
  - Java Server Faces (JSF)
    - Drag and drop User Interface components onto web page
  - Struts
    - Implement a model-view-controller design for web applications
- Enterprise Generation Language (EGL) Java/Web development
  - Code in higher level language, then generate to language of choice (Java or COBOL)
    - Tight integration with Java Server Faces

WebSphere Developer for zSeries Dramatically Lowers the Cost of Development for System z

- Productivity Increases over ISPF:
  - COBOL and PL/I Traditional Development
    - \+ 5% in program understanding
    - +10% in edit activities
    - +30% in compile/debug activities
  - Web Services and SOA development
    - +300% in programmer productivity
- Workstation based edit, compile, debug saves mainframe cycles
  - 80% reduction in cost of compiles

Based on customer studies at Fiducia, Deutsche Bank AG, Nationwide, etc.

Discover Your Assets on the Mainframe

How do I discover what assets exist? If I make changes, what other systems or applications will be impacted?

On Demand Insurance CIO

WebSphere Studio Asset Analyzer can help you

IBM

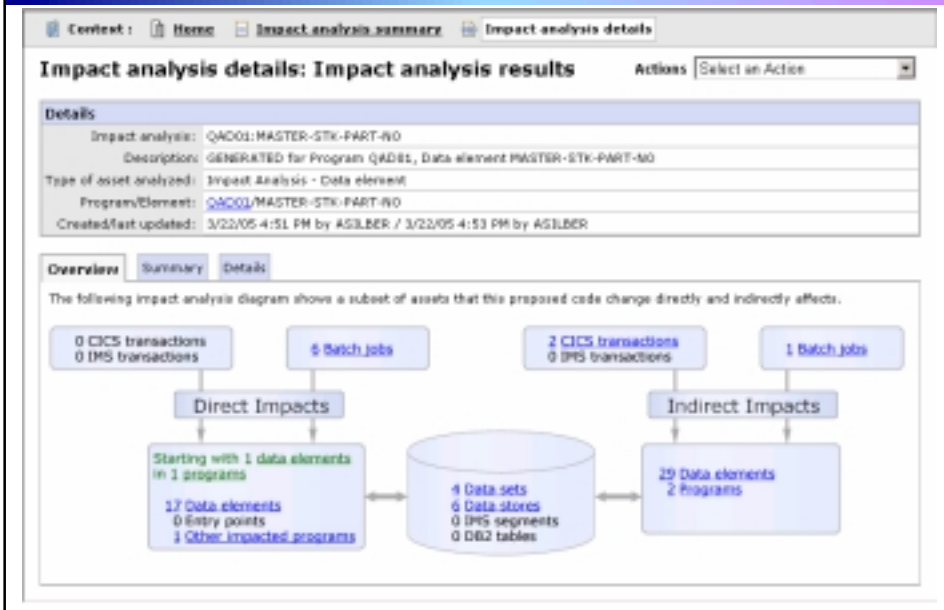
WebSphere Studio Asset Analyzer (WSAA)

- Powerful tool for Discovery and Impact Analysis
  - Search for application assets
  - Understand asset structure and relations
  - Perform data flow and impact analysis
  - Extract code for re-use
- Supports both mainframe and distributed assets
- Results are displayed in easy-to-read graphical windows
  - Web Interface available

WSAA Can Discover and Take Inventory of a Wide Variety of Mainframe Assets

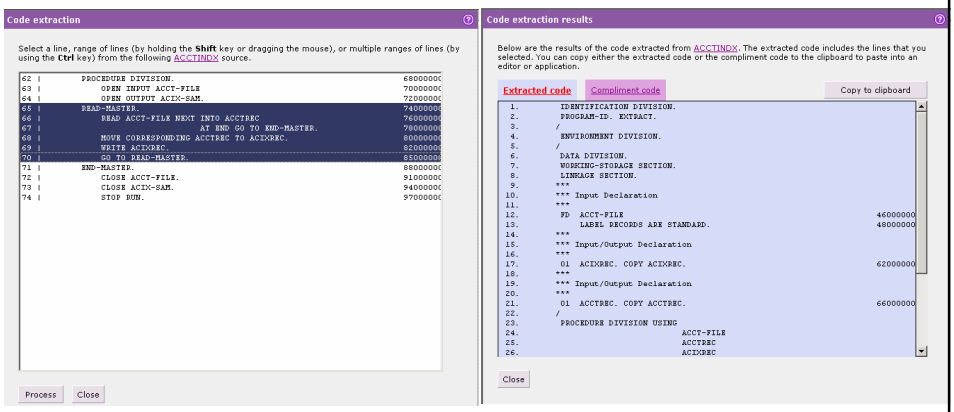
- Source code: COBOL (including copybooks), PL/I, assembler, and JCL
  - Source code can be in a partitioned data set (PDS or PDSE) or controlled by a source control management (SCM) product.
- CICS online regions and transactions
- IMS subsystems and transactions
- SQL statements (DCLCURSOR, DCLTABLE, DELETE, FETCH, INSERT, SELECT, UPDATE)
- DB2 catalogs (columns, stored procedures, systems, tables, and views)
- MQ calls and queues

What Is the Impact of a Proposed Change? WSAA Provides Graphical Results



Component Reuse through Code Extraction

- Use WSAA to identify a segment of code within a COBOL program that you want to reuse in an SOA, then select that code for extraction.
- WSAA will generate a program, and necessary data structures, and if possible a complement program to invoke the newly generated program.



Business Problem Solved

I am glad to see you have reduced our programming costs and our application backlog

On Demand Insurance
CEO

Increased productivity using WDz and WSAA made this possible. I didn't need to hire more System z development skills.

On Demand Insurance
CIO

Summary

WDz boosts developer productivity for both green screen and modern application development. WSAA brings a new level of capability to discovering and understanding your existing code.

IBM