



Redbooks

Tools for the AD Life Cycle

Sabine Holl

sabine_holl@at.ibm.com

Alex Louwe Kooijmans

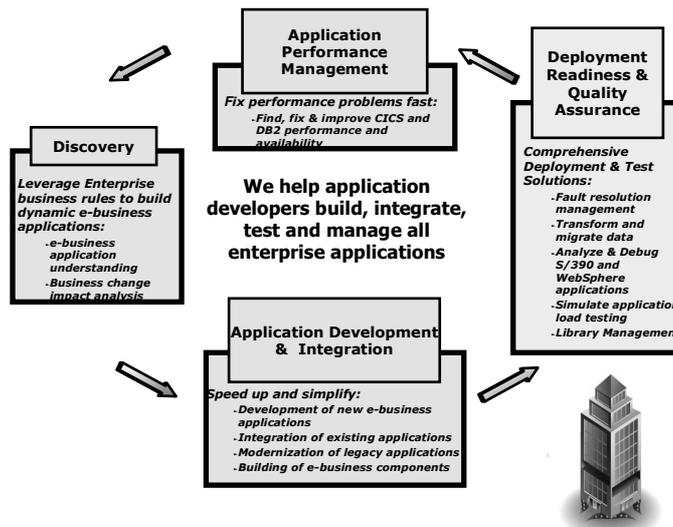
nl53347@nl.ibm.com

Kevin J. Senior

kev_senior@it.ibm.com

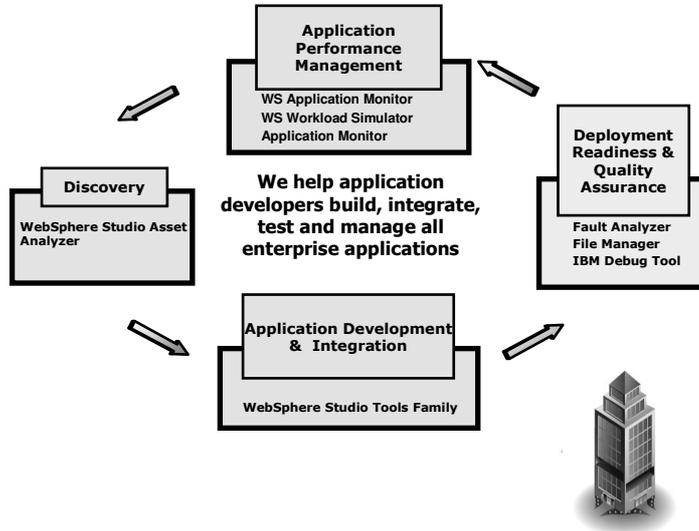


IBM Modernization Strategy





IBM's Modernization Solution

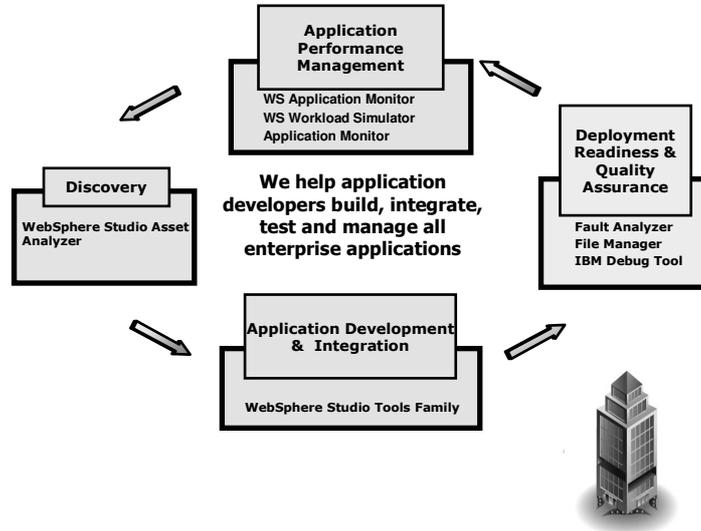


Redbooks

WebSphere Studio Asset Analyser



IBM's Modernization Solution

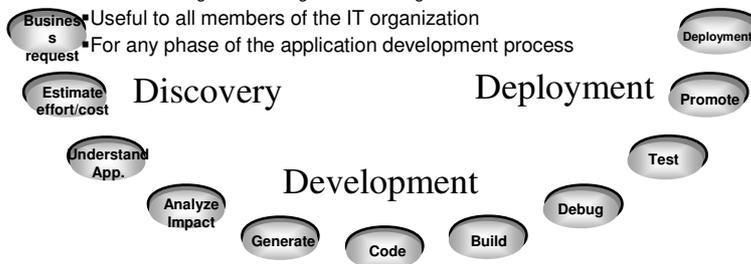


WebSphere Studio Asset Analyser

▪Aids in:

- ▶ Understanding components and their relationships
- ▶ Analyzing the impact of a proposed change
- ▶ Scoping and developing project plans
- ▶ Gathering connector information for an MVS programs
- ▶ Extracting business logic from existing code

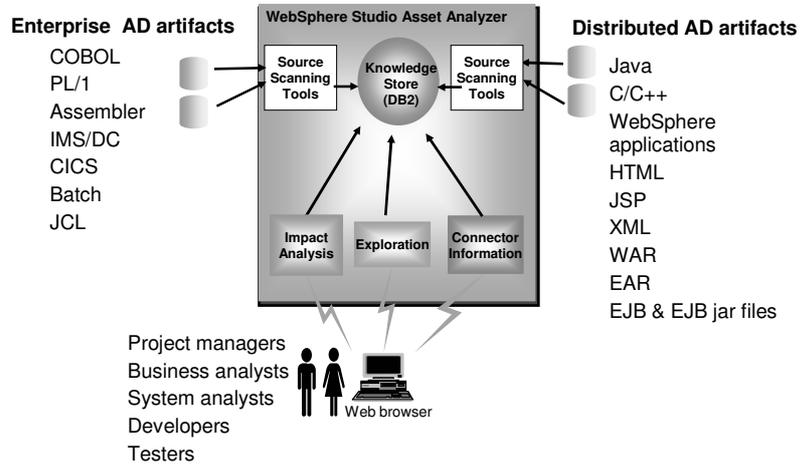
▪ Useful to all members of the IT organization
 ▪ For any phase of the application development process



Benefit: Spend less time searching for asset information

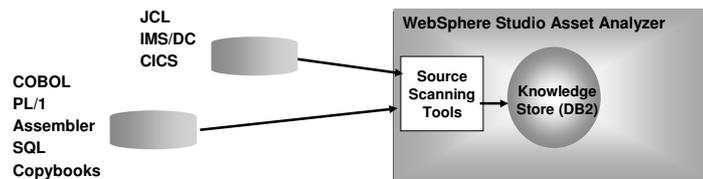


WebSphere Studio Asset Analyzer Overview



Inventory Collection

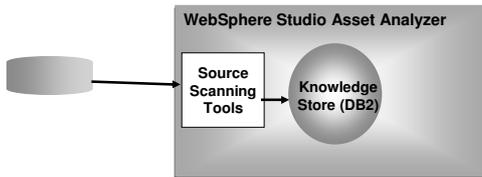
- Scans the resources that you specify, and then stores information about them in a DB2 database
- Inventory information is collected for:
 - Source code
 - CICS and IMS online regions / transactions / subsystems
- z/OS assets can be scanned from:
 - PDS / PDSE
 - SCLM (IBM Source Control Library Manager)
 - Changeman



Inventory Collection (*continued*)

- Distributed assets can be scanned from:
 - ▶ File systems
 - including the hierarchical file system (HFS) of UNIX System Services
 - ▶ An accessible WebDAV server
 - ▶ Rational ClearCase

Java source / bytecode
C / C++
JavaServer Pages
HTML
JAR and EAR files



Overview - Explore

- Used to view application components and their relationships
- View (and select components from) lists of components, or...
- Search for components by:
 - ▶ name
 - ▶ application name
 - ▶ project name
 - ▶ site
- Follow links to navigate through an application, to discover (for example):
 - ▶ what program is invoked by a batch job, or CICS transaction?
 - ▶ what subroutines are called?
 - ▶ what files are used?



Using Explore to find components

Home | **Explore** | Connect | Inventory | Database status | Help

MVS assets | Distributed assets

Explore MVS assets

Use one or more asterisks (*) to locate all assets that match the pattern of your search argument (such as *CUST*).

Search: [Advanced search](#)
 Type mixed case

Enter one or more search strings. A wildcard * character can be used.

Inventory	Total	Run time	Total	Program	Total	Data	Total
Application	17	Batch job	63	Analysis concatenation set	21	Data element	29603
Library	46	CICS group	101			Data set	426
Member	2176	CICS online region	2	BMS map definition	98	Data store	216
Project	68			BMS map set definition	56	DD name	1521
Site	1	CICS transaction	523	Entry point	222	I/O record description	473
		IMS subsystem	1	Literal	3744	SQL column reference	38
		IMS transaction	22	Program	284	SQL table reference	6
		Bup unit	399				

Or just click on any counter to view the full list.



Using Explore to understand components

Home | **Explore** | Connect | Inventory | Database status | Help

MVS assets | Distributed assets

Program details

Member: MYTRADS
 Program: MYTRADS
 Language/type: COB / Program source
 Analysis status: Completed
 Member record count: 992
 Blank lines: 1
 Comment lines: 145
 Noncomment lines: 846
 Program record count: 1038
 Scanning option: CICS
 Splitting nodes: 161
 Site: STLADS2B
 Library: PARTITIONED DATA SET DAVIN10.PDPAK.SC
 Data base updated: 10/2/02 8:39 AM by DAVIN10
 Analysis concatenation set used: DMH19

Actions

- Code extraction
- Identify analysis concatenation set
- Identify analysis options
- Queue for analysis
- View source
- View program data elements
- View e-business program information
- Show control flow diagram
- Show structure diagram

In this example we are looking at details for a program

From here, click to see more information about this program, to display diagrams, or to see a related component

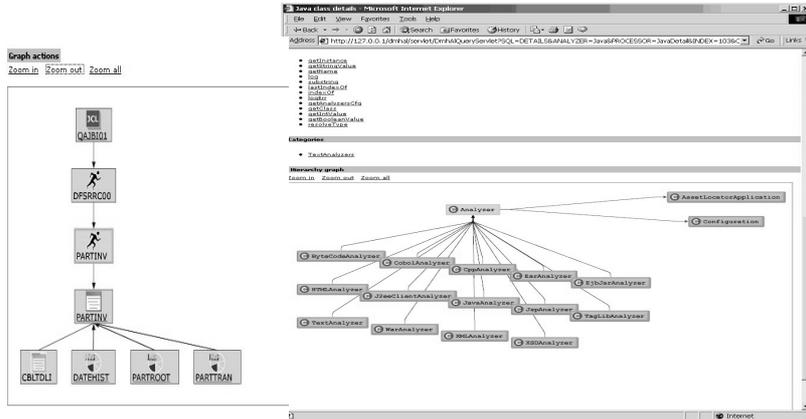
The following tables list the components related to the Program, MYTRADS .

Member (2)	Language	Type	Analysis status	Member record count	Source location
AESCCE	COB	Included source	Completed		PKGD_STUDIO_SDMHDATA(AESCCE)
CUSTFILE	COB	Included source	Completed	15	DAVIN10.PDPAK.COPYLIB(CUSTFILE)



... and their relationships

- Visualization support for zOS and distributed Assets



e-business rating

The screenshot shows the 'e-business program information' section of the IBM WebSphere Studio Asset Analyzer. It includes a search bar and a table with the following data:

Program	Language	Number of lines	Spread FCOs	Central control Spg Transfers	Data COEs	e-business transformation codes
APSELRAL	COB	1897	19	10	5	73
APSRM04	COB	2715	0	2	0	10
AS31AAS	PLI	456	0	12	2	64
CHAPASC	COB	224	0	1	8	22
CHAPASC	COB	442	0	3	12	40
CHAPASC	COB	215	6	1	10	32
COSEF	COB	701	0	11	0	5
INVESTMAK	COB	27	0	0	0	0

Benefit: Quickly understand the reusability of code



Code extraction

- Analyze, isolate, and reuse existing business logic
 - ▶ to support services oriented architecture
- Selected code is stored with generated data areas



Benefit: Spend less time harvesting from existing code



Overview - Change Impact Analysis

- Determine what components are affected based on:
 - changes to field declarations
 - changes to a section of program source code
 - changes to an entry point signature (name, parms)

Project details: Analyze-change

Project: QADD1:MASTER-STK-PART-NO:
 Description: GENERATED 20021002-23:58:41
 Project type: Impact Analysis - Field declarat
 Program/Element: QADD1:MASTER-STK-PART-NO
 Created/last updated: 10/2/02 10:58 PM by DAVIN10

The following impact analysis diagram shows a subset of

Direct impacts

STARTING WITH 1 DATA_ELEMENTS

1 PROGRAM

12 DATA_ELEMENTS

0 ENTRY_POINTS

0 OTHER IMPACTED PROGRAMS

3 DATA_SETS

11 DATA_STORES

0 IMS_SEGMENTS

2 SQL_TABLE_REFERENCES

Indirect impacts

32 DATA_ELEMENTS

4 PROGRAMS

The following table lists all the components that are affected by this proposed code change. You can access the details page for a component by selecting that component.

Inventory Results	Run time	Results	Program	Results	Data	Results	
Application	1	Batch job	Z	Analysis concatenation set	0	Data element	50
Library	1	CICS group	0	BMS map definition	n/a	Data set	2
Member	1/2	CICS online region	1	BMS map set definition	n/a	Data store	11
Project	n/a	CICS transaction	1	Entry point	1	DD name	n/a
Site	1	IMS subsystem	0	Literal	2	I/O record description	n/a
		IMS transaction	0	Program	2	SQL column reference	2
		Run unit	Z			SQL table reference	2

The result is a diagram that summarizes the impact and a list of specific components that may be impacted - both directly and indirectly.

Overview - Connector Information

- Use WSAA to
 - ▶ quickly gather input and output data structures of transactions
 - ▶ put this information in a form that you can import into a connector-building tool such as WebSphere Studio Application Developer IE or VisualAge Java
- For each transaction, WSAA generates
 - ▶ a summary report
 - ▶ a COBOL copybook containing the input/output data structures

Summary - WebSphere Studio Asset Analyzer

- Aids in:
 - ▶ Understanding components and their relationships
 - ▶ Scoping and developing project plans
 - ▶ Gathering connector information for an MVS programs
 - ▶ Extracting business logic from existing code
 - ▶ Analyzing the impact of a proposed change
- Useful to all members of the IT organization
 - ▶ Project Managers
 - ▶ Analysts
 - ▶ Developers
 - ▶ Testers
- For any phase of the application development process
 - ▶ Requirements
 - ▶ Development
 - ▶ Test
 - ▶ Deployment

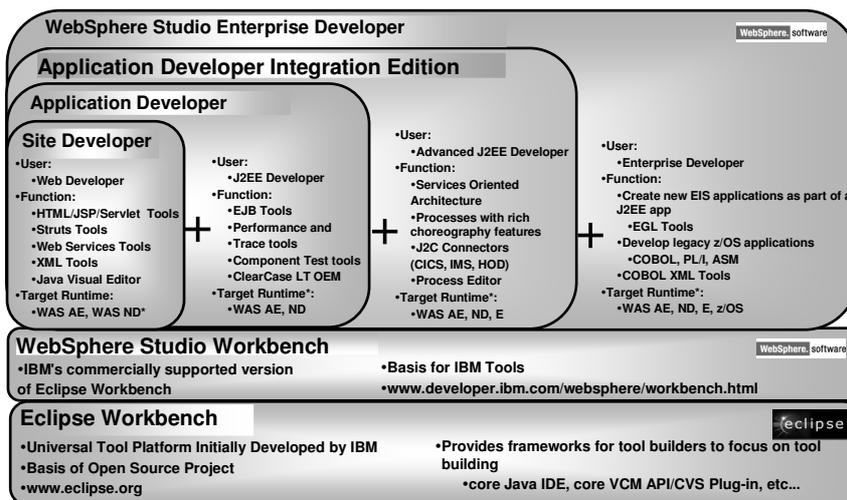


Redbooks

WebSphere Studio Application Developer



WebSphere Studio Tools Family



*Other IBM/non-IBM Runtimes could be used

Key Benefits of Integration Edition

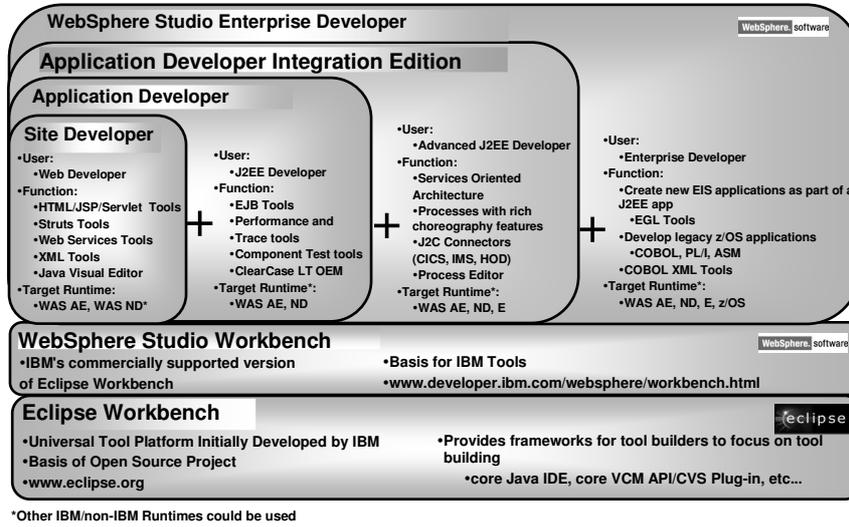
- J2C tooling allow J2C Adapters to be imported and exposed as a service
 - ▶ Integration Edition ship CICS, IMS and HOD Adapters
 - ▶ Others adapters can be imported as Resource Archives (RAR)*
 - J2EE 1.3 Specification
- Powerful tooling for data structure transformation into XML datatypes
 - ▶ e.g. COBOL COMMAREAs, 3270 Screens, C data structures, etc.
 - ▶ Makes it easy to deal with various EIS data structures from Java or other languages
 - ▶ Developers don't need to deal with low level datatype conversions (e.g. COBOL to Java)

More about J2C Resource Adapters

- J2EE Connector Architecture 1.0 defines how J2C resource adapters plug into application servers
 - ▶ Does not define how a resource adapters might plug into tooling
- IBM is defining the specification of tooling plugins
 - ▶ Target is J2EE Connector Architecture 2.0
 - ▶ We are releasing an early implementation of this specification in Integration Edition 4.1
- How does a tool acquire metadata about the data structures for an EIS system?
 - ▶ Currently, this varies significantly based on the EIS system
 - ▶ The implementation we are proposing uses WSDL to describe the data structures
 - ▶ We are working with these ISVs (tooling support): Dassault, iSphere, Siebel, SAP, Gemstone, CrossWorlds, Peregrine and Intentia



Integration Edition in WebSphere Studio Tools Family

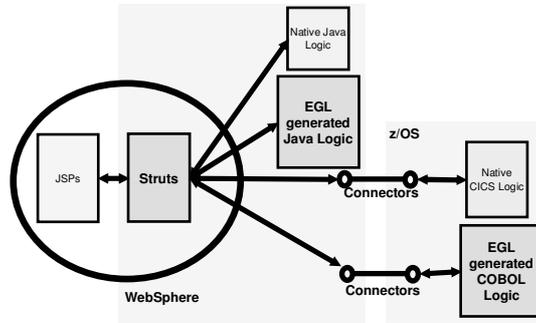


Agenda



- Struts Tools
- Business Integration Tools
- z/OS Application Development Tools
- XML Enablement Tool for z/OS Cobol applications
- Enterprise Generation Language (EGL)

Struts Tools



Struts Tools

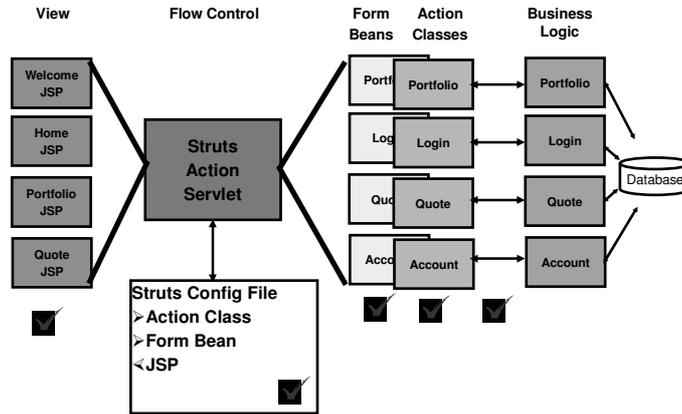
- Rapid design and construction of J2EE web apps
 - ▶Promotes well-structured web applications
 - ▶Enables development in less time with fewer errors
 - ▶Connects to business logic of choice
 - EJBs, Java, COBOL, PL/I, EGL, etc.
- Wizards and editors
 - ▶Setup J2EE web project with Struts support
 - ▶Create and update Struts components
 - ▶Visual design and assembly
- Build Support
 - ▶Validates changes against existing resources



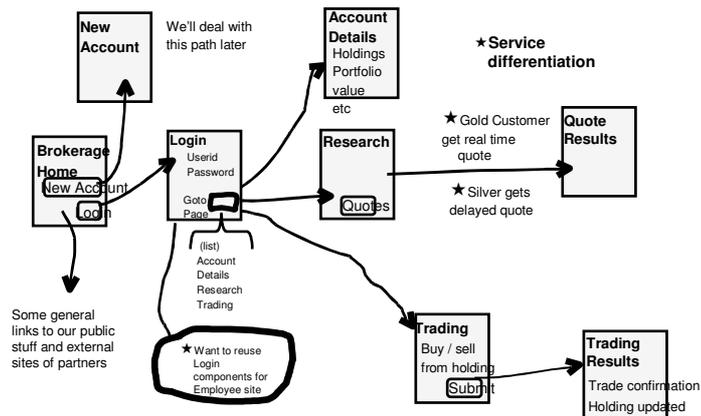
Create Struts Components

☑ = Struts Tool support

Struts: 100% 50%
Struts Tools: 50% 25%



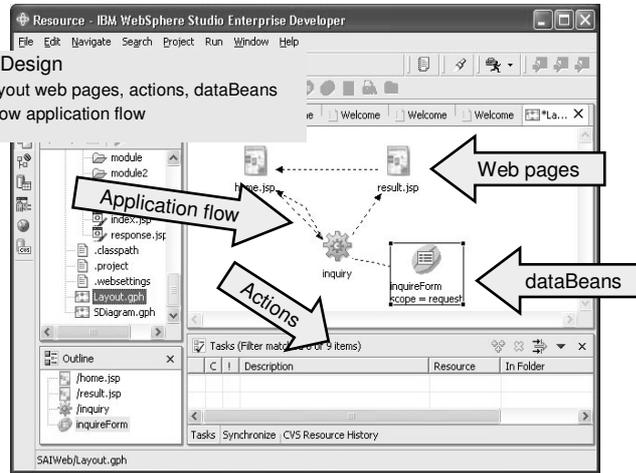
Visual design using yellow sticky notes



Web Diagram Editor - Design

Visual Design

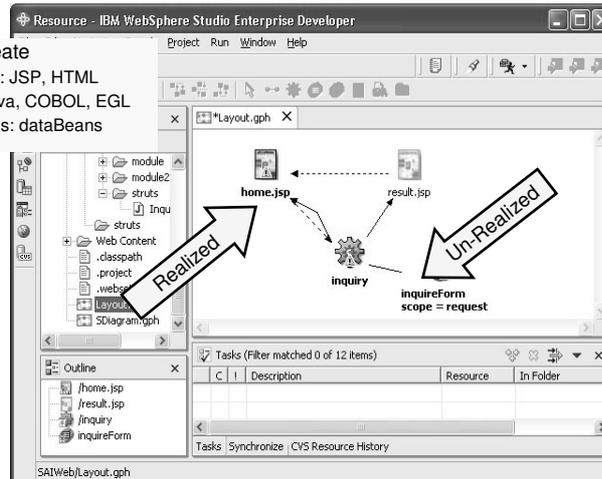
- ▶ Layout web pages, actions, dataBeans
- ▶ Show application flow



Web Diagram Editor - Definition

Wizards to create

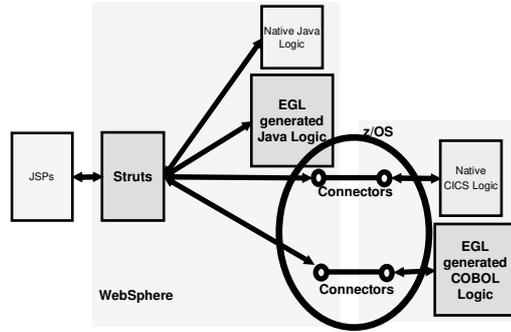
- ▶ Web pages: JSP, HTML
- ▶ Actions: Java, COBOL, EGL
- ▶ ActionForms: dataBeans



Benefit: Faster construction of web applications



Business Integration Tools



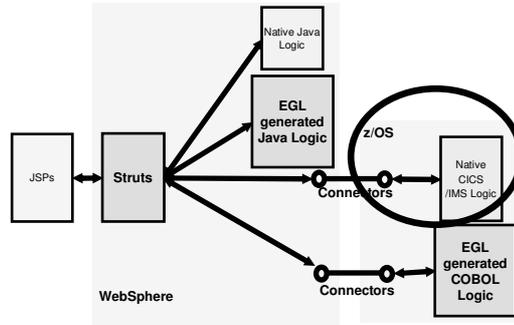
New and Improved JCA Adapters

New and Improved JCA Adapters

Adapter		Details
SAP		<ul style="list-style-type: none"> •New •Available from IBM
IMS		<ul style="list-style-type: none"> •Improved •Support 2PC when communicating via TCP/IP with IMS Connect •Support for MFS •Developer license included in Integration Edition
Siebel		<ul style="list-style-type: none"> •New •Available with Siebel CRM offering
Tuxedo		<ul style="list-style-type: none"> •New •Built by Prolifics (www.prolifics.com)
CrossWorlds		<ul style="list-style-type: none"> •New •JMS to invoke WICS (WebSphere InterChange Server formerly Crossworlds) Collaboration (runtime engine ICS) •Access to WBI Adapters via JCA in the near future •WBI Adapters Cover all major EIS vendors and a wide range of ISV Application Vendors
J.D. Edwards		<ul style="list-style-type: none"> •New •Will become available starting in mid-2003 •Available with J.D. Edwards back-end systems software

IBM JCA included in Integration Edition: CICS ECI, CICS EPI, IMS, 3270 HOD.

z/OS Application Development Tools

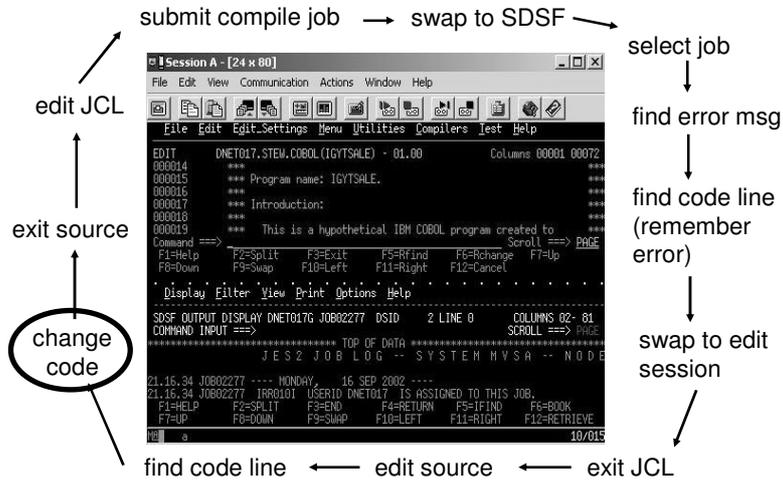


z/OS Application Development

- Interactive, workstation-based environment
 - ▶ Faster development with less errors
 - ▶ Host offload of development CPU cycles
- Edit/compile/debug on the workstation
 - ▶ Remote or Local
 - ▶ Language sensitive editors for COBOL, PL/I, ASM, JCL
- Interactive access to the mainframe
 - ▶ Job generation, submission, and monitoring
 - ▶ TSO command execution
- SCLM support



ISPF based development

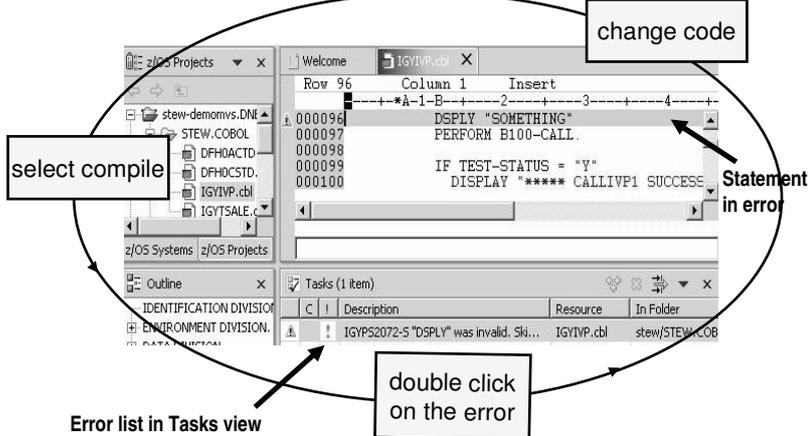


```

!Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Edit.Settings Menu Utilities Compilers Test Help
EDIT DNET017.STEW.COBL(IGYTSALE) - 01.00 Columns 00081 00072
000014 ***
000015 *** Program name: IGYTSALE. ***
000016 ***
000017 *** Introduction: ***
000018 ***
000019 *** This is a hypothetical IBM COBOL program created to ***
Command ==> Scroll ==> PAGE
F1=Help F2=Split F3=Exit F4=Find F5=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
Display Filter View Print Options Help
SDSF OUTPUT DISPLAY DNET017G JOB02277 DSID 2 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> SCROLL ==> PAGE
***** TOP OF DATA *****
JES2 JOB LOG -- SYSTEM MVS -- NODE
21.16.34 JOB02277 ---- MONDAY, 16 SEP 2002 ----
21.16.34 JOB02277 TRR0101 USERID DNET017 IS ASSIGNED TO THIS JOB.
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
10/015
    
```



WebSphere Studio based development



```

Welcome IGVIVP.cbl X
Row 96 Column 1 Insert
+---+---+---+---+
| *A-1-B | 2 | 3 | 4 |
+---+---+---+---+
000096 D$PLY "SOMETHING"
000097 PERFORM B100-CALL.
000098
000099 IF TEST-STATUS = "Y"
000100 DISPLAY "***** CALLIVP1 SUCCESS*****"
    
```

Benefit: Simplified Development for COBOL and PL/I on z/OS



Interactive Access to z/OS

The screenshot displays the IBM WebSphere Studio Enterprise Developer interface. On the left, a 'z/OS Projects' tree shows a project named 'denomvms.zosau' containing various source files like 'DFHOCSTD.cbl', 'IGYVNF.cbl', and 'XML.cbl'. Below this is an 'Outline view' showing the structure of a selected file. The main window is an 'ISPF like editor' displaying COBOL code with columns for Row, Column, and Insert. At the bottom, a 'z/OS Job Monitor' table shows job details, and a 'TSO Commands view' is also visible.

Job ID	Job Name	Owner	Hold Status	Exec Node
denomvms.f1	*	DNET017	*	*
denomvms.f2	XML*	DNET017	*	*
stplex@b.f1	*	WILBERT	*	*



SCLM Support

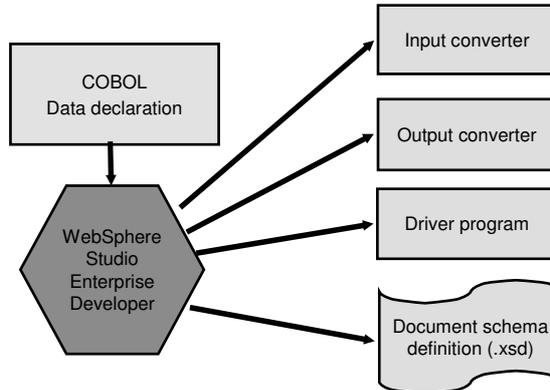
- Access to SCLM on z/OS
 - ▶ Connect to SCLM repository
 - ▶ View a list of projects
 - ▶ List project members
 - ▶ Execute SCLM actions
- Check-in/check-out support
 - ▶ TEMP WORK AREA

The screenshot shows the SCLM Repository interface. A 'Show View' dialog is open, displaying a tree view of the repository structure. A context menu is open over a file, listing various SCLM actions such as 'SCLM Promote Function', 'SCLM Build Function', 'SCLM Delete Local File From Workspace', and 'Save File back to Repository'.



XML Enablement Enhancements for z/OS

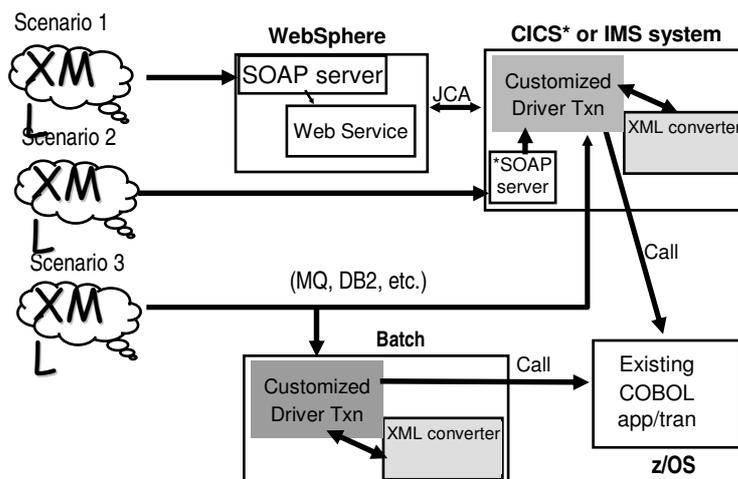
- ▶ Creates COBOL programs to handle XML messages
- ▶ Original COBOL program unchanged



Benefit: Speeds development of XML-enabled COBOL applications



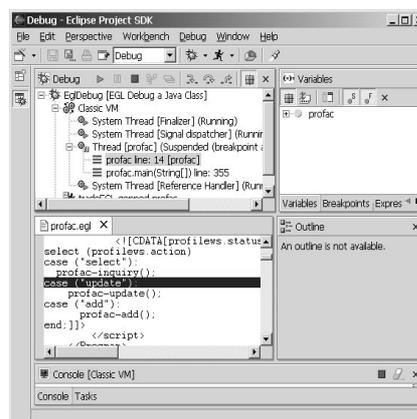
z/OS XML Implementation Scenarios



Additional Features

Testing and Debugging

- End-to-end Debugging
 - ▶ Java and JSP debugger
 - ▶ COBOL, PL/I debugger
 - ▶ EGL debugger
- Verifying application Flow
 - ▶ breakpoints
 - ▶ changing variable values
- WebSphere Test Environment
 - ▶ integrated in Workbench
 - ▶ choice of versions



Benefit: End-to-end test and debug from the Workbench

Cheat Sheets

- Help the novice Integration Edition user to learn key programming tasks faster.
- Provide active guidance when creating a specific type of Service.
- Actual wizard that you need for each stage of development is launched.
- Once a step is completed, the cheat sheet adds a check mark to the list of steps and you progress to the next step.



Summary

- Comprehensive end-to-end development environment
 - Single development environment
 - provides integration of process, tools, infrastructure and assets
- Supports more runtimes, developers, and tooling
 - WebSphere Application Server, CICS, IMS, z/OS batch
 - Web, Java, and Enterprise Developers
 - Java, J2EE, Web, XML, COBOL, EGL, and Web services
- Higher-quality applications in a fraction of the time
 - Web Diagram editor
 - Language sensitive editors
 - Integrated WebSphere Server test environment
 - Integrated deployment automation tools
- How will Integration Edition reduce development time?
 - Supports J2C Resource Adapters (Avoid low level coding to access a J2C enabled backend)
 - Business Process editor allows service flows to be created visually (instead of writing code)
 - Isolates developers from other low-level coding:
 - COBOL/Java conversions
 - SOAP invocations
 - Defining WSDL operations/messages



Redbooks

WebSphere Studio Application Monitor V2.1



WebSphere Studio Application Monitor



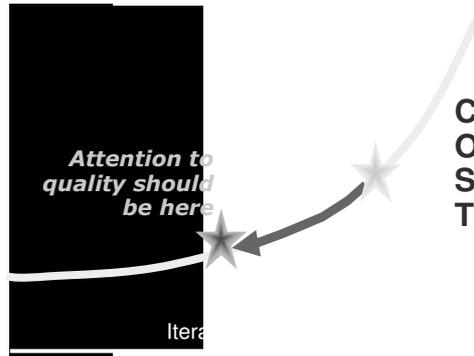
- Strategy
- Overview
- Architecture
- Product Demonstration



Strategy:

To prevent, detect, diagnose and remove defects throughout the software development and deployment lifecycle

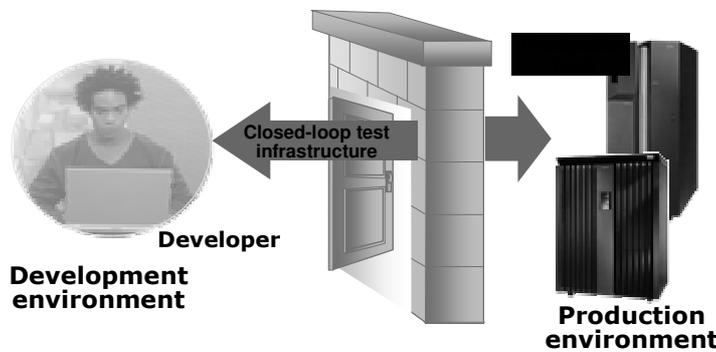
- Improve application quality
- Provide early warnings of activities susceptible to failure
- Analyze across disciplines to understand root causes
- Establish a test foundation for downstream test activities



Strategy:

Provide development and deployment tools integration
Reduce business downtime by automating system diagnosis, repair, & rebuild

- Find and fix errors post-deployment quickly
- Speed up application rebuild and redeploy
- Bridge development teams and operation teams





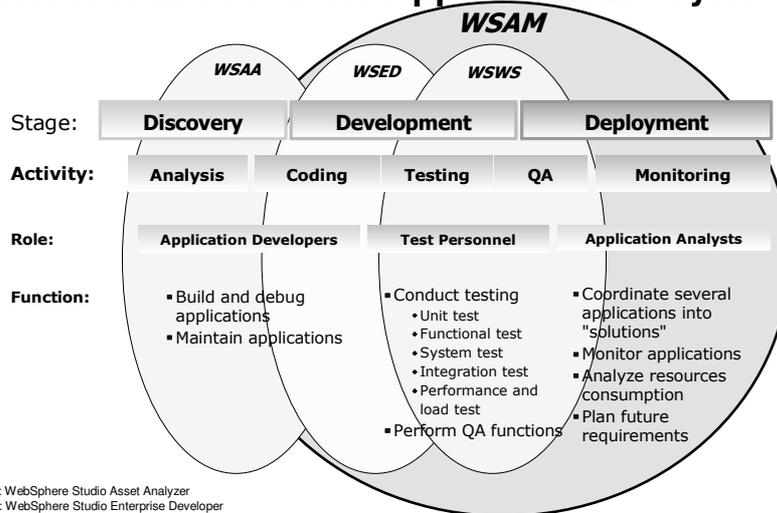
The AD Lifecycle: Discover - Develop - Deploy



Stage:	Discovery	Development	Deployment
Activity:	Analysis	Coding	Testing QA Monitoring
Role:	Application Developers	Test Personnel	Application Analysts
Function:	<ul style="list-style-type: none"> ▪ Build and debug applications ▪ Maintain applications 	<ul style="list-style-type: none"> ▪ Conduct testing <ul style="list-style-type: none"> • Unit test • Functional test • System test • Integration test • Performance and load test ▪ Perform QA functions 	<ul style="list-style-type: none"> ▪ Coordinate several applications into "solutions" ▪ Monitor applications ▪ Analyze resources consumption ▪ Plan future requirements



How WSAM Fits in the Application Lifecycle

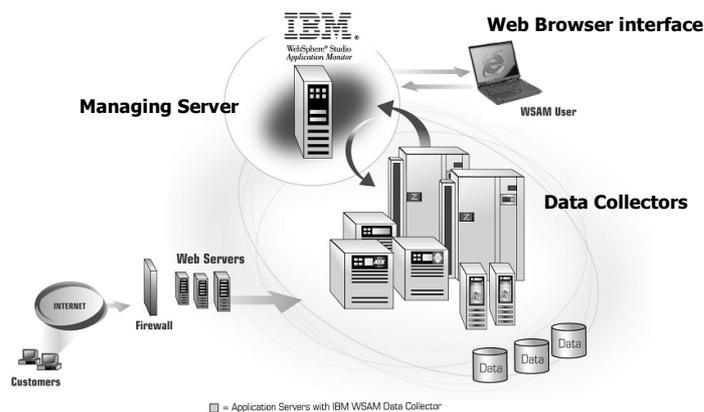


WSAA: WebSphere Studio Asset Analyzer
 WSED: WebSphere Studio Enterprise Developer
 WSWS: WebSphere Studio Workload Simulator
 WSAM: WebSphere Studio Application Monitor

What Does WSAM Provide?

- Real-time **problem determination** for J2EE applications
- **Performance management** through a rich set of historical analysis functions
- Non-intrusive, **production-safe** architecture
- **Application independence** brings total manageability without the need to modify or access application code
- A **lifecycle** tool for Development, Test and Deployment

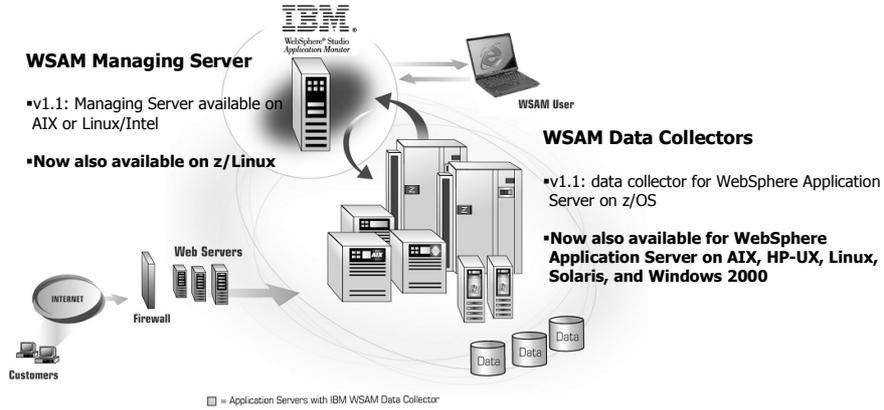
Conceptual Overview of WSAM



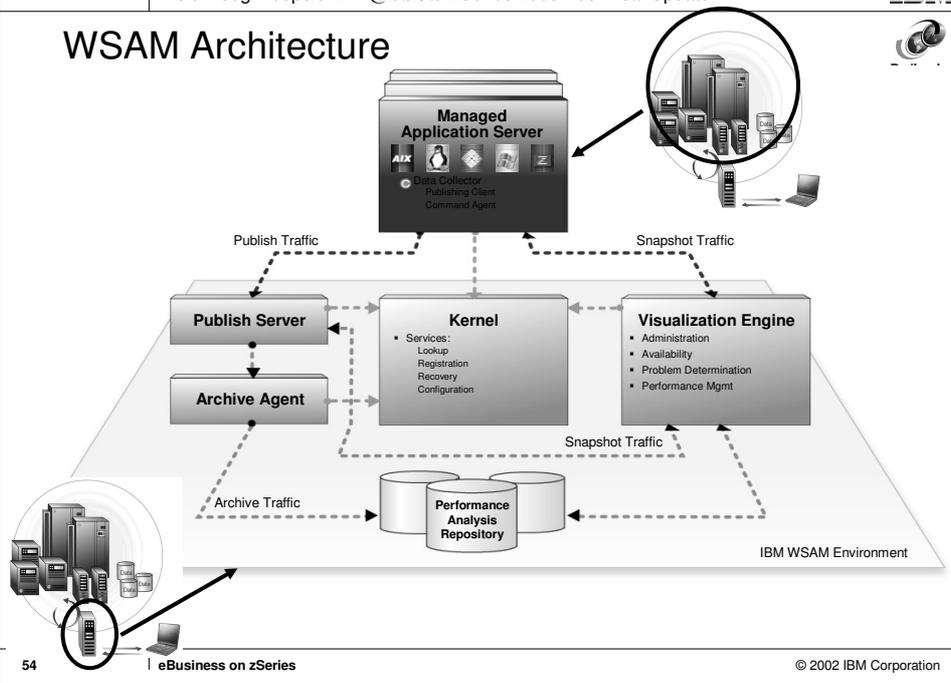


WSAM - Environments Supported

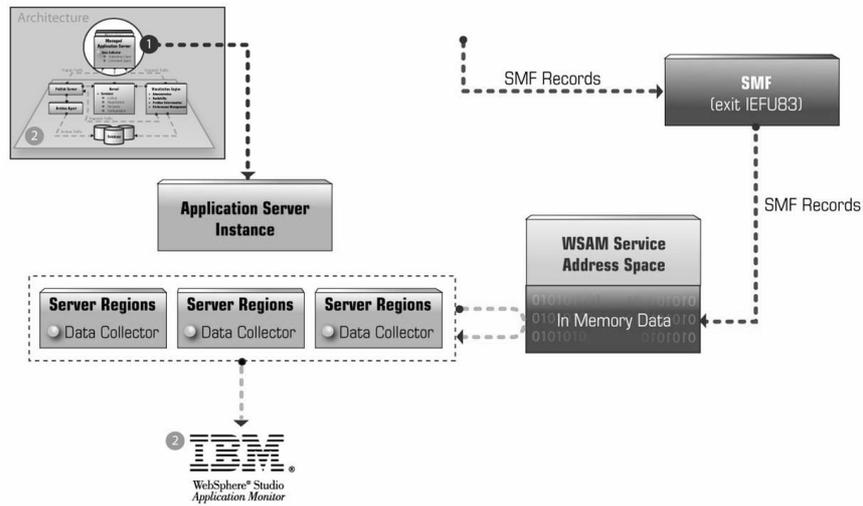
- **Distributed Data Collectors** - A comprehensive solution for customers with mixed WebSphere platforms (z/OS and Distributed)
- **Managing Server on z/Linux**
- **Support for WebSphere Application Server v5**



WSAM Architecture

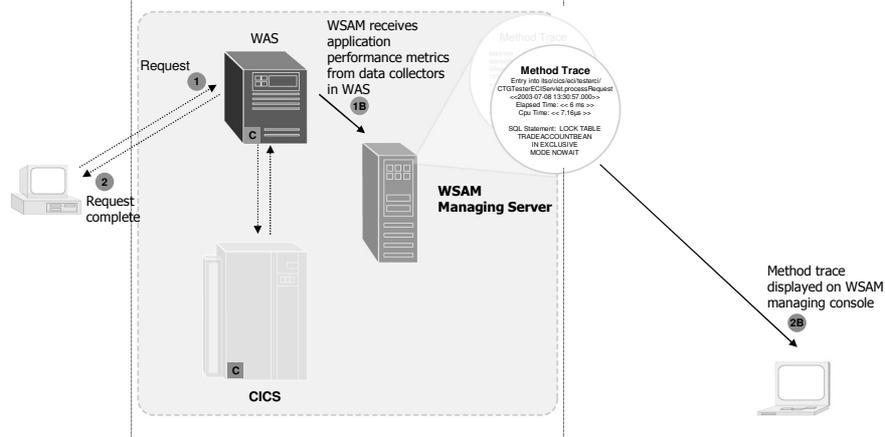


WSAM Architecture: z/OS



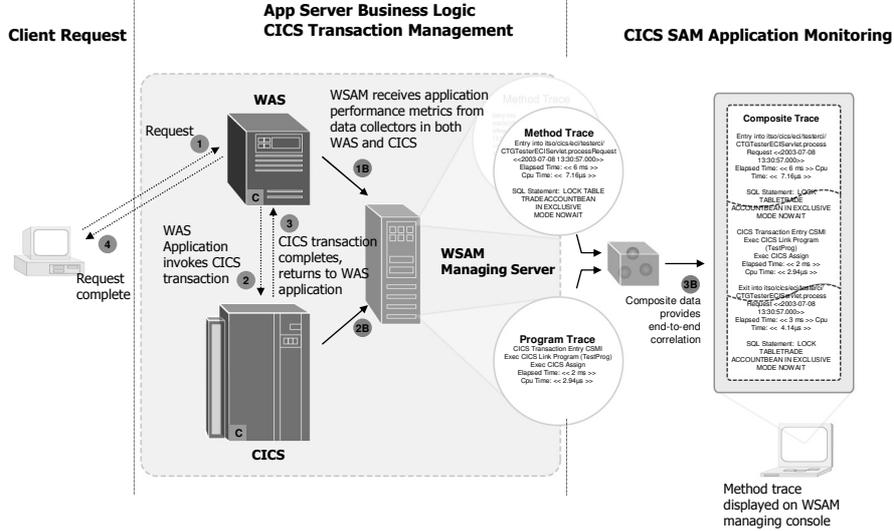
J2EE Transaction

Client Request





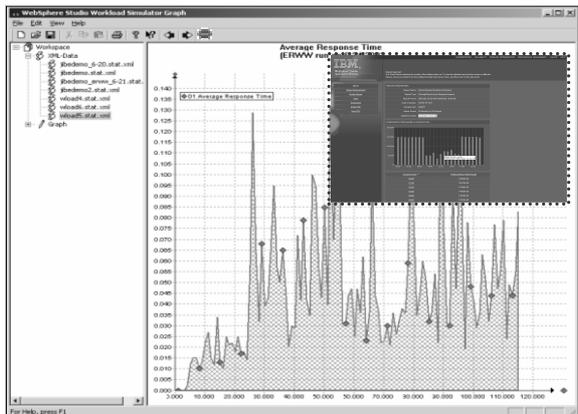
J2EE-CICS Composite Transaction



C = Application Servers with WSAM Data Collector
57 eBusiness on zSeries



WSAM-WSWS Integration



- **WSAM v2.1 is integrated with WS Workload Simulator v1.1.1**
- Data is exchanged between WSAM and WSWS during the load-test, and reports created at test completion

58 eBusiness on zSeries

Other Features



- **System Resources**
Displays summary information for the system resources on the selected application server
JVM CPU Usage, JVM Memory Usage, EJB Coverage, EJB Activity, Transactions Initiated and JSP/Servlet Activity and Coverage
- **Trap and Alert Management**
A trap can be set with a threshold on an event or behavior
When the system meets the criteria of the trap, the action (alert) occurs
Integration via SNMP to Tivoli, Openview, BMC, etc.
New trap types, e.g. "application hung" traps in WSAM 2.1
Stack trace and method trace information now included in trap data
Multiple UI enhancements to facilitate creation and administration of traps
- **Software Consistency Check**
Detects software mismatches in "cloned" runtime environments

Other Features



- **Account and Server Group Management**
A "server group" consists of a user-defined collection of servers
Accounts may be associated with specific groups
Access to data and operations of the group can be restricted
- **Dynamic Monitoring**
Monitoring Scope and granularity of information returned may be changed without restarting either the applications or the application servers
No need to pinpoint specific classes or methods in advance (i.e., no need to designate what needs to be monitored)
- **Memory Analysis**
In-depth analysis of JVM memory utilization, GC frequency and heap contents
Memory leak detection and analysis

Other Features



- **Problem Determination**

- Monitoring support for additional J2EE API's within nested request events, including JMS, JCA, JNDI, JavaMail, and improved JDBC request capture
 - New L2 monitoring level captures these events
- Full JVM active thread display with complete thread dump options
 - Includes non-J2EE threads
- Recent history is captured
 - Recent requests per JVM
- Enhanced method trace with pagination and new method flow views, and abstracted event statistics views that make it easy to quickly identify and drill-down to specific problematic methods
- Reduced L3 monitoring overhead on distributed platforms

Other Features



- **Other**

- Ability to set sampling rate by app server and by monitoring level (formerly single global setting)
- Full Web/HTTP Server information view - important HTTP server statistics are now available for concurrent web server monitoring
- Multiple UI enhancements - unified overview page, new server, overview metrics, customizable landing page, new graphical displays WLM monitoring (z/OS only)



WSAM and Tivoli Products

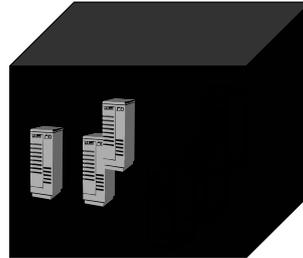


System and Network Health,
Application Server Health
End-to-end Monitoring
Tivoli Monitoring for Web Infrastructure

End-User View



Tivoli Monitoring for Transaction Performance



J2EE Application Health/Problem Determination
WebSphere Studio Application Monitor



How a Customer Used WSAM

- A European financial services firm deployed a new version of a third-party back-office "Black Box" application, and experienced performance degradation as a result
- Application slowdowns caused a 60% drop in overall response time, resulting in lower employee productivity
- WebSphere Resource Analyzer was able to identify high CPU utilization and other system-level information.
- IBM WSAM was deployed to isolate the specific cause of the problems at an application level.

How WSAM Helped



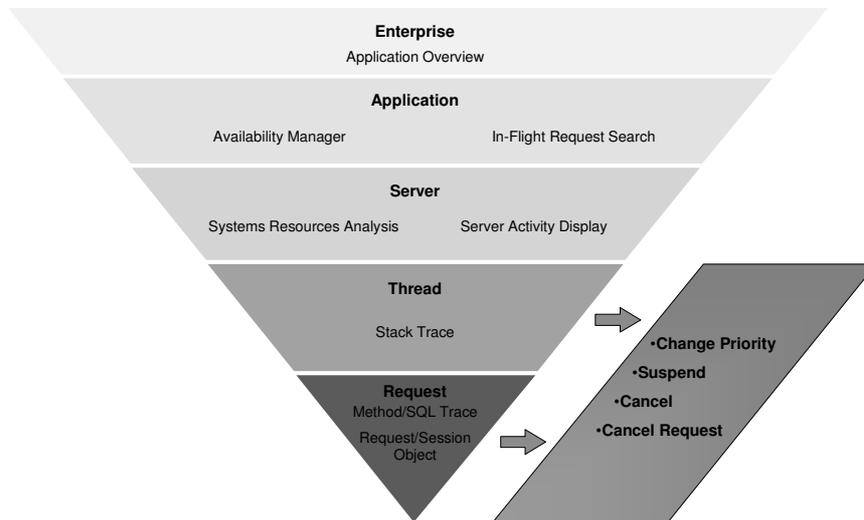
- Rapid implementation in one day allowed for immediate analysis and data capture against the application
- Ability to see all applications within the JVM without byte code modification to quickly find problematic application code
no access to development resources required (valuable feature, as this was a third-party app)
- WSAM provided the necessary "Method trace" information, which showed exactly which methods were consuming excessive CPU cycles
- Data Center management presented the WSAM findings report to the app vendor, which resulted in the vendor rewriting the problematic portions of the application to resolve the issues
End result was a dramatic reduction in CPU consumption, and end-user response times decreasing by over 50%

Product Demonstration



- Problem Determination
- Performance Analysis
- Memory Analysis

Drill-Down Problem Determination



Scenario: Trades Are Not Executing



- Large online brokerage requires sub-five (<5) second trading response time to meet SLA with internal business units.
- Customers begin calling in to complain that trades are not executing.
- Call center manager notices a significant increase in trading-related calls.
- Data center is notified that trades are not executing properly.



Problem Determination Drill Down with WSAM



In-Flight Request Search

Server Name	Client Request	Thread ID	Total Resident Time (ms)
demo-win-s01:trade01.2182	/WebSphereSamples/TradeSample/servlet/PingJDBCRead	212429224	275796
1 demo-win-s01:trade02.3044	/WebSphereSamples/TradeSample/servlet/PingJDBCRead	212428088	275796
demo-win-s01:trade01.2182	/WebSphereSamples/TradeSa	168228272	6407

Server Activity Display

Thread ID	Client Request	Idle Time (ms)	Last Known Class Name	Last Known Class Name
212428088	/WebSphereSamples/TradeSample/servlet/PingJDBCRead	125400	trade_client/TradeConfig	rndSymbol

Request Detail

Detail	
Thread ID:	212428088
Client Request:	/WebSphere Samples/TradeSample/servlet/Ping...
Accum. CPU:	17 (µs)
Idle Time:	125400 (ms)
Thread Type:	HttpServlet
Resident Time:	275796 (ms)
Last SQL:	LOCK TABLE TRADEHOLDINGBEAN IN EXCLUSIVE MODE

Method Trace

Method	
oracle.jdbc.driver/OracleStatement.isRegularResultSet	2002-11-11 13:30:57.000>> Elapsed Time: <<275796 ms >> CPU Time: <<17 µs >>
SQL Statement:	LOCKTABLE TRADEACCOUNTBEAN IN EXCLUSIVE MODE

Cancel Request

Cancel

Change Priority [v]

Change Thread Status [v]

5 **Cancel Request**



Overall View – Application Groups



ADMINISTRATION AVAILABILITY PROBLEM DETERMINATION PERFORMANCE MANAGEMENT LOGOUT HELP

WebSphere® Studio Application Monitor

APPLICATION OVERVIEW
View availability for all applications running in assigned server groups, and a baseline response time indicator relating current and historical response time. Configure the baseline by going to the Administration tab and clicking Group Management. To view additional information for the applications in a server group, select from the drop-down menu.

Quotes: 2 of 2 Servers Available 100 PERCENT

Volume Throughput: [Bar Chart]

Trading: 2 of 2 Servers Available 100 PERCENT

Volume Throughput: [Bar Chart]

Select Detail View: [Dropdown]

Total Volume (last hour): 379

Total Volume (last hour): 295

Click "In-Flight Request Search"

Volume Throughput

Response Time Indicator

% Server Availability

→ This page shows a bird's-eye view of the applications that are available in the server groups

→ Drill-down options are available from this page

→ For example, we want to see all of the active requests on the Trading Group



Application View – All Currently Running Requests



IN-FLIGHT REQUEST SEARCH
In the Search Request box, type the name of the request for which you are searching. If you leave this box empty, all active requests will display.

SEARCH CRITERIA

Group: Trading
Server: All Servers
Search Request: OK

SEARCH RESULTS

Timestamp	Server Name	Client Request	Start Date/Time	Thread ID	Resident Time (ms)
Oct 4, 2002 6:51:37 AM	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/PingJDBCRead?ttl=3000&lock=true	Oct 4, 2002 6:52:39 AM	6545501132	78442
	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	653860168	38927
	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	653536272	38927
	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	653933288	38927
	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:51:15 AM	647304776	22927
	demo-win-s02.Trade02.2428	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:51:15 AM	653859584	22927

→ "In-Flight Request Search" can be narrowed down with user-defined criteria
→ This display is sorted by resident time, longest first
→ Drill down further by clicking the AAD button ("Application Activity Display")



Server View – Thread Data in an App Server



APPLICATION ACTIVITY DISPLAY
The Application Activity Display provides thread data for an application server. Click the thread's ID to review more request detail.

SERVER SELECTION

Group: Trading Server: demo-win-s02.Trade02.2428

SERVER INFO

Snapshot Date: Oct 4, 2002 Application Server Name: Trade02
Snapshot Time: 6:57:25 AM Application Server IP Address: 192.168.3.45
Platform CPU % Utilization: 0.78% Total Thread Count: 1

THREADS

Thread ID	Priority	Client Request	Client Request Start	Resident Time (ms)	Accumulated CPU (ms)	Idle Time (ms)	Thread Status	Last Known
6545501132	5	/WebSphereSamples/TradeSample/servlet/PingJDBCRead?ttl=3000&lock=true	Oct 4, 2002 6:52:39 AM	78442	0	78440	Waiting Condition	com.ibm/je/sj/cmip
653536272	5	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	32959	0	32959	Waiting Condition	com.ibm/je/sj/cmip
653933288	5	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	31959	0	31959	Waiting Condition	com.ibm/je/sj/cmip
651328872	5	/WebSphereSamples/TradeSample/servlet/GetQuote?ttl=30	Oct 4, 2002 6:50:59 AM	31959	0	31959	Waiting Condition	com.ibm/je/sj/cmip

→ "Application Activity Display" shows thread data for an application server at a specific point in time
→ The first thread above shows a high idle time
→ Drill-down further by clicking on the thread ID



Thread View – Detail on a Selected Thread

The screenshot displays the 'REQUEST DETAIL' page in WebSphere Studio Application Monitor. The interface includes a top navigation bar with options like ADMINISTRATION, AVAILABILITY, PROBLEM DETERMINATION, PERFORMANCE MANAGEMENT, LOGOUT, and HELP. On the left, a 'MENU' panel contains buttons for 'Select New Thread', 'Request Detail', 'Stack Trace', 'Method Trace', and 'Request/Session Object'. The main area shows 'REQUEST PROPERTIES' and 'REQUEST DETAIL' for a specific thread.

REQUEST PROPERTIES:

Snapshot Date	Oct 4, 2002	Application Server Name	Trade02
Snapshot Time	6:52:39 AM	Application Server IP Address	192.168.3.45
Platform CPU % Utilization	0.00%	Total Thread Count	1

REQUEST DETAIL:

Thread ID	85450132	Accumulated CPU	0 ms
Client Request	/WebSphereSamples/TradeSamples/ServletPingJDBCRead?#=-500&isdir=true	Idle Time	78440 ms
Client Request Start Date	Oct 4, 2002	Thread Type	HttpServlet
Client Request Start Time	6:52:39 AM	Last Known Class Name	com.ibm.ws.com.proxy.StatementProxy
Resident Time	78442 ms	Last Known Method	getStatement
Priority	5	Thread Status	Waiting Condition
Change Priority	No Change	Change Thread Status	No Change

Last SQL: LOCK TABLE TRADEHOLDING\$EW IN EXCLUSIVE MODE NOWAIT

Last SQL Call: (indicated by an arrow pointing to the Last SQL field)

Annotations:

- An arrow points from the 'Stack Trace' button in the menu to the 'REQUEST DETAIL' section with the text: "Click 'Stack Trace'".
- A callout box on the right contains the text: "Details for the chosen thread are displayed" and "Execute a Stack Trace or Method Trace on the Left Nav to determine next course of action".



Stack Trace

The screenshot displays the 'STACK TRACE' page in WebSphere Studio Application Monitor. The interface includes a top navigation bar with options like ADMINISTRATION, AVAILABILITY, PROBLEM DETERMINATION, PERFORMANCE MANAGEMENT, LOGOUT, and HELP. On the left, a 'MENU' panel contains buttons for 'Select New Thread', 'Request Detail', 'Stack Trace', 'Method Trace', and 'Request/Session Object'. The main area shows 'STACK TRACE PROPERTIES' and a list of stack frames.

STACK TRACE PROPERTIES:

Snapshot Date	Oct 4, 2002	Application Server Name	Trade02
Snapshot Time	7:30:49 AM	Application Server IP Address	192.168.3.45
Platform CPU % Utilization	0.00%	Total Thread Count	13

STACK TRACE:

Depth	Class	Method
0	java/lang/Thread	sleep
1	web_pmtv/PingJDBCRead	doGet
2	javaservlethttp/HttpServlet	service
3	javaservlethttp/HttpServlet	service
4	com.ibm.servletengine.webapp/StrictServletInstance	doService
5	com.ibm.servletengine.webapp/StrictLifecycleServlet	_service
6	com.ibm.servletengine.webapp/IdleServletState	service
7	com.ibm.servletengine.webapp/StrictLifecycleServlet	doService
8	com.ibm.servletengine.webapp/ServletInstance	doService
9	com.ibm.servletengine.webapp/ValidServletReference	doService
10	com.ibm.servletengine.webapp/ServletInstanceReference	doService
11	com.ibm.servletengine.webapp/WebAppRequestDispatcher	handleWebAppDispatch
12	com.ibm.servletengine.webapp/WebAppRequestDispatcher	dispatch

Annotations:

- An arrow points from the 'Method Trace' button in the menu to the 'STACK TRACE' section with the text: "Click 'Method Trace'".
- A callout box on the right contains the text: "Stack Trace shows all the methods that have not completed execution".
- An arrow points to the 'Method' column of the stack trace table with the text: "Last Method Invoked".



Method Trace

METHOD TRACE PROPERTIES

Snapshot Date	Oct 4, 2002
Snapshot Time	6:52:39 AM
Platform CPU % Utilization	0.00%
Current Total Elapsed Time	78442 ms
Current Total CPU Time	0 ms

METHOD TRACE

Entry	Date/Time
web_prrmv/PingJDBCRead.doGet	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.mdSymbol	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.mdInt	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.random	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.random	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.mdInt	Oct 5, 2002 6:51:29 PM
trade_client/TradeConfig.mdSymbol	Oct 5, 2002 6:51:29 PM
com.ibm.ejbs/cm/proxy/StatementProxy	Oct 5, 2002 6:51:29 PM
SQL Statement: LOCK TABLE TRADEHOLDINGBEAN IN EXCLUSIVE MODE NOWAIT	
com.ibm.ejbs/cm/proxy/StatementProxy.executeCommon	Oct 5, 2002 6:51:29 PM
SQL Statement: LOCK TABLE TradeQuoteBean IN EXCLUSIVE MODE NOWAIT	



Cancel Request – Take Action on a request

REQUEST PROPERTIES

Snapshot Date	Oct 4, 2002	Application Server Name	Trade02
Snapshot Time	6:52:39 AM	Application Server IP Address	192.168.3.45
Platform CPU % Utilization	0.00%	Total Thread Count	1

REQUEST DETAIL

Thread ID	054560132	Accumulated CPU	0 ms
Client Request	WebSphereSamples/TradeSampleServlet/PingJDBCRead?B=3500&icon=true	Idle Time	78440 ms
Client Request Start Date	Oct 4, 2002	Thread Type	HttpServlet
Client Request Start Time	6:52:39 AM	Last Known Class Name	com.ibm.ejbs/cm/proxy/StatementProxy
Resident Time	78442 ms	Last Known Method	getStatement
Priority	5	Thread Status	Waiting Condition
Change Priority	No Change	Change Thread Status	No Change
Last SQL	LOCK TABLE TRADEHOLDINGBEAN IN EXCLUSIVE MODE NOWAIT		



Normal Processing Resumes




Performance Analysis Drill Down



Extract Data Set

Trend & Total

Decompose Data

Quote	15%
Buy	50%
Login	20%
Sell	15%

Detail

Request ID	Request Name	R.T. (seconds)	Server Name
83335	Sell	23858	app_win_s01
83336	Sell	83762	app_lnx_s09

Trace

Method Name	Elapsed Time	Elapsed CPU	Time Date Stamp
trade_client/AppServlet.doPost	23858	2378	2002-06-24 17:38:37.011
trade_client/AppServlet.performTask	83762	2838	2002-06-24 17:38:37.011



Trend Report

TREND REPORT
The Trend Report displays the results of the defined data set. To view the detailed report broken down by different criteria, choose an option from the Additional Detail drop-down menu, and then click on the data point.

REPORT PROPERTIES

- Report Name: Demo Request Analysis All Servers
- Report Type: ThroughPUT per hour Request Analysis
- Report Period: 09/13/02 12:00 AM to 09/27/02 12:00 AM
- Data Grouping: HOUR OF DAY
- Sample Size: 243971
- Server Scope: All Servers on All Groups
- Additional Detail: Application Name [v]

THROUGHPUT PER HOUR vs. HOUR OF DAY

Mouse over for detail; click to drill-down

→ A "Trend Report" can be created for a specific WAS data set
→ Click on a particular bar to drill-down "Decomposition Report"

HOUR OF DAY	THROUGHPUT PER HOUR
00:00	148580.00
01:00	150700.00
02:00	147900.00
03:00	146540.00
04:00	148380.00
05:00	147560.00



Decompose

DECOMPOSITION REPORT
The Decomposition Report provides a breakdown of the Trend Report by the criteria selected. Click on a data point to view more detail.

REPORT PROPERTIES

- Report Name: Request Analysis All Servers - 11/11
- Report decomposed by: Application Name on HOUR OF DAY: 19:00
- Report Type: ThroughPUT per hour Request Analysis
- Report Period: 11/11/02 12:00 AM to 11/12/02 12:00 AM
- Data Grouping: Hour of Day
- Server Scope: All Servers on All Groups

Select data point of most interest

→ "Decomposition Report" shows a further breakdown of "Trend Report", according to the criteria selected ("Application Name", in this case)
→ Clicking on a data point provides more details

APPLICATION NAME	THROUGHPUT PER HOUR	ACTUAL COUNT	PROJECTED COUNT
Trade	1191.00	1191	1191
Login	1042.00	1042	1042
Quote	524.00	524	524
Account	5.00	5	5



Detail

REPORT PROPERTIES

Report Name	RequestAnalysis All Servers - 11/11
Report detail on	HOUR OF DAY : 18:00
Report decomposed by	Application Name on HOUR OF DAY : 18:00
Report Type	ThroughPut per hour RequestAnalysis
Report Period	11/11/02 12:00 AM to 11/11/2002 12:00 AM
Server Scope	All Servers on All Groups

REQUEST NAME	REQUEST TYPE	RESPONSE TIME (ms)	CPU TIME (ms)	SERVER	TIME-STAMP
Trade	Servlet	42045	15.625	demo-win-s02.Quote01	2002-11-11 17:59:19.0
Trade	Servlet	42045	0.000	demo-win-s02.Quote01	2002-11-11 17:59:18.0
Trade	Servlet	41967	0.000	demo-win-s02.Quote01	2002-11-11 17:59:18.0
Trade	Servlet	38108	0.000	demo-win-s02.Quote01	2002-11-11 17:59:29.0
Trade	Servlet	39014	0.000	demo-win-s02.Quote01	2002-11-11 17:59:30.0
Trade	Servlet	39014	0.000	demo-win-s02.Quote01	2002-11-11 17:59:30.0

→ The "Request Report Detail" opens
 → Drilling-down is available through the "Trace Report"



Trace Report

REPORT PROPERTIES

Report Name	RequestAnalysis All Servers - 11/11
Request Trace on	Trade
Report detail on	HOUR OF DAY : 18:00
Report decomposed by	Application Name on HOUR OF DAY : 18:00
Report Period	11/11/02 12:00 AM to 11/11/2002 12:00 AM
Server Scope	All Servers on All Groups

Entry into web_pmmwGetQuote.doGet <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

SQL Statement: SELECT symbol,price,details FROM TradeQuoteBean WHERE symbol

Entry into trade_clientTradeConfig.mdSymbol <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

Entry into trade_clientTradeConfig.mdInt <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

Entry into trade_clientTradeConfig.random <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

Exit from trade_clientTradeConfig.random <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

Exit from trade_clientTradeConfig.mdInt <<2002-11-11 17:59:19.367>> Elapsed Time: <<0 ms >> Cpu Time: <<0.000 ms >>

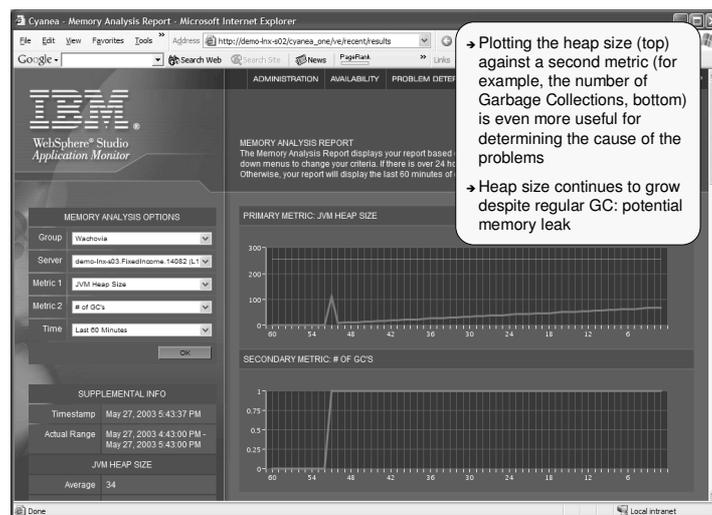
→ The "Trace Report" provides details on any one record selected

Memory Analysis (available with v1.1.2)

- Memory leaks in Java are hard to pinpoint
 - Memory handling is done by the JVM
 - Pro: This frees up the developers to focus on building the applications
 - Con: But it makes memory issues hard to trace back to a specific component

- WSAM has a "Memory Analysis" feature to track down memory leaks

Heap Size Can Be Plotted Against A Second Metric





Heap Analysis

→ "Heap Analysis" shows the detailed heap content

→ Quickly identify classes that contribute the most to heap utilization

Class name	Total size (tbt)	Percent of total size	# of instances	Percent of total #
com.trading.Bond	25	0%	2188	0%
com.trading.Bond\$factory	0	0%	1	0%
com.trading.Bond\$Servlet	0	0%	1	0%
com.trading.Node	477	0%	61180	27%
com.trading.Trie	0	0%	1	0%
object[]	61474	93%	66763	29%
primitive[]	2663	4%	26777	11%



Memory Leak Analysis

→ "Memory Leak Analysis" shows a snapshot of the heap at two different times

→ Comparing the results shows significant growth in two classes: those are the likeliest sources of leaks

Class name	Original # of instances	Original total size (tbt)	Δ # of instances	Δ Total size (tbt)
com.trading.Bond\$Servlet	1	0	0	0
com.trading.Trie	1	0	0	0
com.trading.Bond\$factory	1	0	0	0
com.trading.Bond	1859	21	1255	14
com.trading.Node	51860	405	34375	14



Summary: A Robust Platform for Application Management

WSAM: an integral part of a **24/7 enterprise** IT infrastructure

- Advanced **Problem Determination** for real-time problem resolution
- Detailed historical analysis with **Performance Management** functionality
- True **Application Independence** allows views of applications without requiring access or modification to application code

Rapid Implementation enables fast ROI



Redbooks

WebSphere Studio Workload Simulator for z/OS and OS/390



Agenda

1. Do We Need Load-Testing?
2. WebSphere Studio Workload Simulator for z/OS
3. Scenario
4. Q&A



Agenda

1. Do We Need Load-Testing?
2. WebSphere Studio Workload Simulator for z/OS
3. Scenario
4. Q&A



Do We Need Load-Testing?



When Load-Testing Was Not Done...



1996 chess match between Kasparov and IBM's computer

"Thousands of users per second were turned away from IBM's website..." [USA Today 2/14/96]

1999 Christmas shopping at KBKids.com

- KBKids.com launched its website in July 1999 for the Christmas shopping season
- Seriously underestimated response of children and parents (Pokemon toys): sharp increase in traffic caught company offguard
- Entire website taken offline to add more servers during critical shopping season
- Loss of customer good will

[Source: abcnews.com, 1999]

Right after the 9-11 event

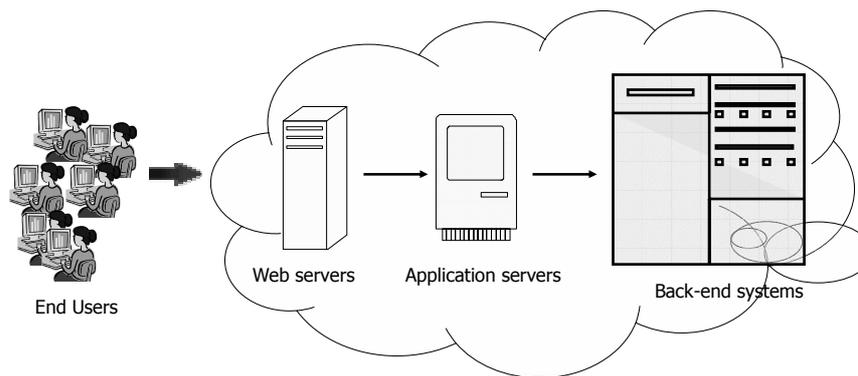
- USAToday.com website - 18% availability
- FBI.gov website - usually 2-3 secs response time; went to 40 secs response time
- ABCnews.com website - 5% availability

[Source: NetworkWorldFusion, 2001]

Agenda

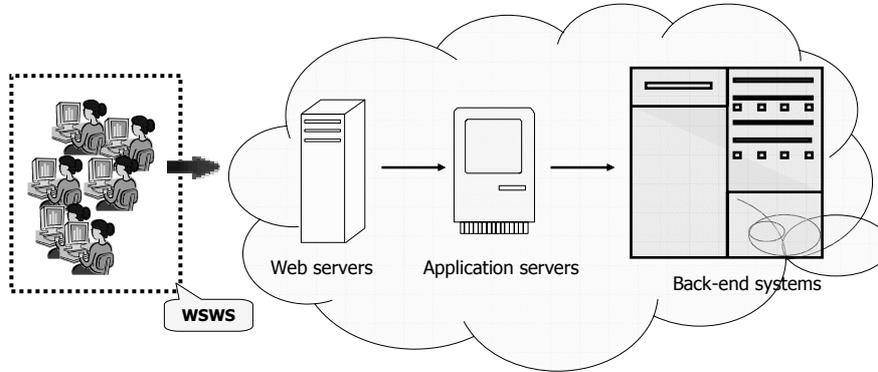
1. Do We Need Load-Testing?
2. WebSphere Studio Workload Simulator for z/OS
3. Scenario
4. Q&A

A Typical Set-up





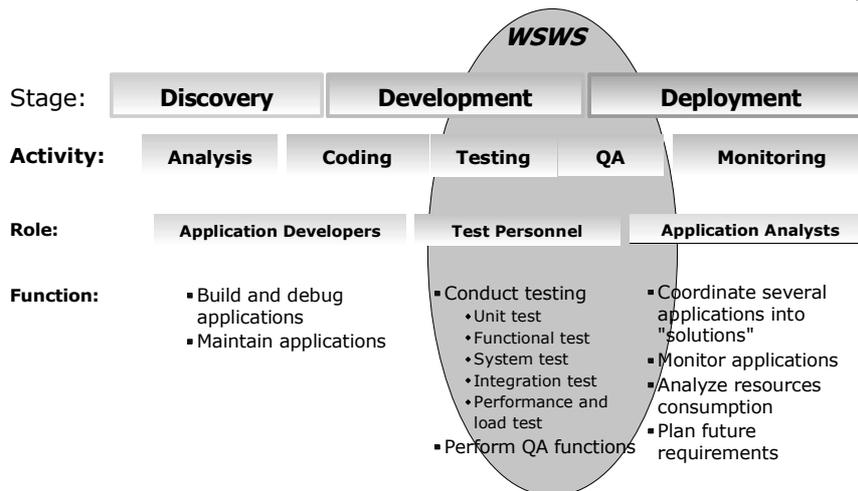
What WSWS Does



- WSWS simulates a large number of web browsers (or "virtual users"), and is used to test web applications and web servers
- WSWS does not pre-req the WebSphere Application Server



Where WSWS Fits





WSWS - Architecture



Windows

Windows-based "Controller"

- Test Case Capture
- Test Case Scripting
- Test Execution and Monitoring
- Test Analysis



zSeries

z/Series-based "Engine"

Load generation

All test functions are accessible through the Windows-based controller



WSWS Can Test Any Web-Serving Environment



Windows

Windows-based Controller

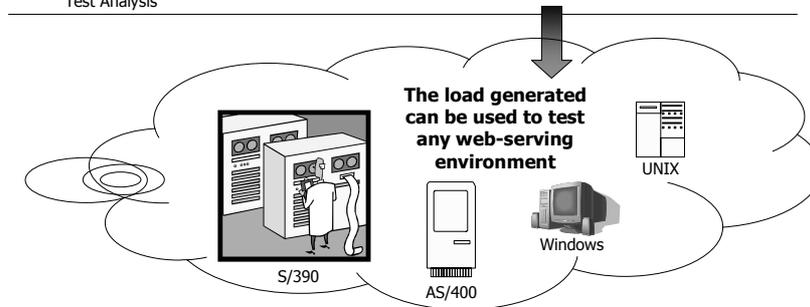
- Test Case Capture
- Test Case Scripting
- Test Execution and Monitoring
- Test Analysis



zSeries

z/Series-based Engine

Load generation



WSWS – Value Proposition



WSWS is scalable

- WSWS is the only 390-based solution on the market currently
- The load is generated from a zSeries server
- It can be used to test any web-serving environment
- A "server farm" is not required

WSWS is priced on machine capacity, not "per-virtual user"

- Other load-testing tools are priced by per-virtual-user
- They can be very costly if the customer needs to run large simulations
- WSWS is OTC/IPLA

WSWS has a user-friendly interface

- All test functions are accessible from a Windows interface

WSWS Usage Scenarios



What can WSWS be used for?

•Verify web-serving performance

- Stress/Load a web server
 - Simulate large numbers of users executing a distribution of functions
- Benchmark web applications
 - Execute industry web benchmarks (e.g., Trade 2) to estimate response time under production conditions
- Verify load-distribution functions

•Quality Assurance on web applications

- Simulate large numbers of users on web applications
- Perform regression test prior to web deployment

•Functional test of server-side functions

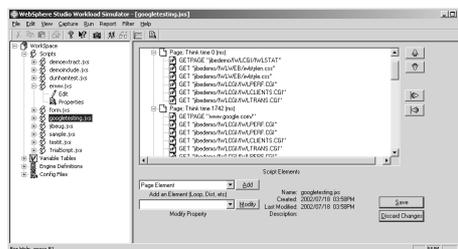
- Drive web-server functions and validate the functionality

1-2-3 Testing...

- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

1. Create Test Scripts

- Capture/automatic script generation
 - User's actions can be captured to automatically build a test script
- Test scripts are managed through a Windows GUI
- Variable content is supported
 - Ex: multiple users with different log-on IDs and passwords
- Programming functions (e.g., distribute, looping)
 - Add flexibility to test scripts
 - Response validation
 - String manipulation
- Full Cookies support
- Socks support



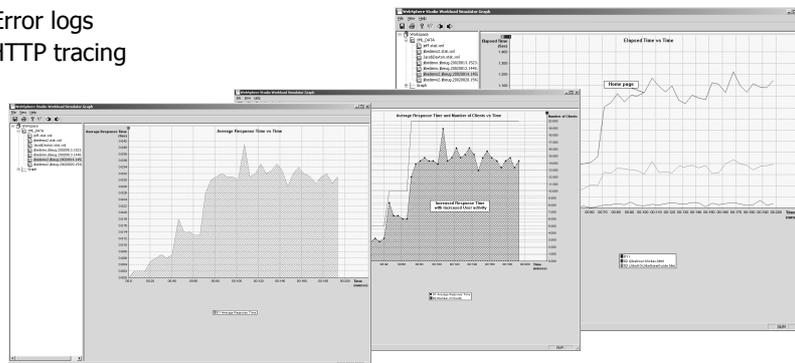
2. Execute and Monitor Test

- Various run-time parameters can be set:
 - Number of clients, delay controls
 - Number of times the script is to repeat, or a time limit for the test
 - Various log/trace options
 - Socks support
- The test engines can be monitored in real-time through a Windows GUI or a browser interface
 - Tests can be started/stopped remotely through the browser
- Support for multiple engines
- SSL support



3. Analyze Results

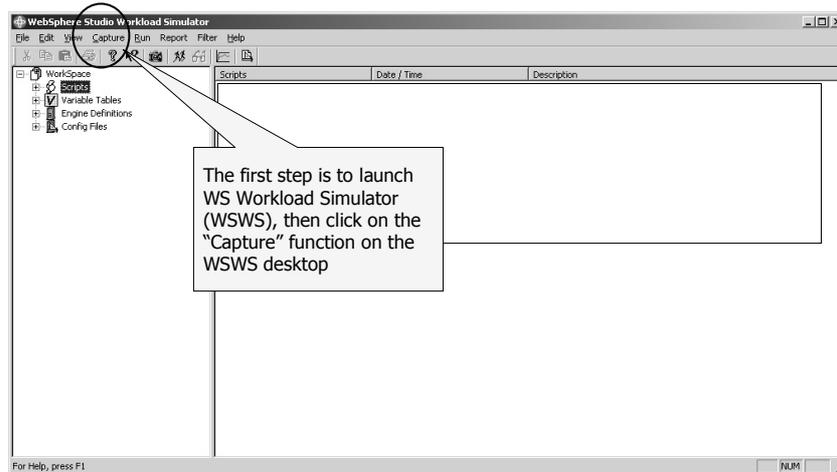
- Graphing of performance measurements:
 - Response time
 - Data read or written transfer (throughput)
 - Throughput of Page elements, Pages, and Transactions
 - CPU or Memory Utilization
 - The above items can be plotted against time or the number of virtual users
- Error logs
- HTTP tracing



Agenda

1. Do We Need Load-Testing?
2. WebSphere Studio Workload Simulator for z/OS
3. Scenario: an airline needs to load-test its flight-inquiry system
4. Q&A

- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results





- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

The capture session starts:

- a web browser opens up
- the web session's data stream is recorded



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

Carrier	Flight Number	Departing City	Departing Date & Time	Arriving City	Arriving Date & Time	Aircraft Type
AA	828	RDU Durham/Raleigh	11/06/2002 09:24 AM	ORD Chicago	11/06/2002 09:30 AM	McDonnell Douglas Super MD-80 (900)
AA	781	ORD Chicago	11/06/2002 10:53 AM	SFO San Francisco	11/06/2002 01:27 PM	Boeing 737-600 Passenger (736)

The user searches for flights, then requests details on specific flights



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

Scripts	Date / Time	Description
AA_CheckReservation.js	2002/09/09 04:31PM	
AA_ViewDetail.js	2002/09/09 04:34PM	
demonstrat.js	2002/06/21 12:39PM	Popup window to view details
demoninclude.js	2002/06/21 12:39PM	This will get a HttpResponse and then pass it to demonstrate.js
dunhamtest.js	2002/07/18 04:52PM	This will extract all of the dollar values from the response
erwin.js	2002/06/21 08:32AM	Example of an Order Entry test
Form.js	2002/06/21 08:26AM	Form Example
googletesting.js	2002/07/18 03:58PM	
ibout.js	2002/07/19 02:55PM	
sample.js	2002/07/18 02:54PM	
testit.js	2002/06/20 10:57AM	Just Testit ... ibedemo server simple test (with a loop)
TriasScript.js	2002/07/31 04:10PM	Test script with SVL

When the "Capture" session ends, a test script is automatically generated



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

The test script shows all elements of the web session, and can be visually edited

```

AA_CheckReservation.js
  HTML percent = 10;
  #! get_percent = 0;
  Page: Think time 0 (ms)
  GETPAGE "www.aa.com/apps/common/ps/HM...oad.html"
  Page: Think time 481 (ms)
  GETPAGE "www.aa.com/"
  GET "www.aa.com/content/images/hav/a1.gif"
  GET "www.aa.com/content/images/hav/a2.gif"
  GET "www.aa.com/content/images/hav/b1.gif"
  GET "www.aa.com/content/images/hav/b2.gif"
  GET "www.aa.com/content/images/hav/c1.gif"
  GET "www.aa.com/content/images/hav/e1.gif"
  GET "www.aa.com/content/images/hav/e2.gif"
  GET "www.aa.com/content/images/hav/f1.gif"
  
```

Page Element: Add
 Add an Element (Loop, Dist, etc)
 Modify Property
 Name: AA_CheckReservation.js
 Created: 2002/09/09 04:31PM
 Last Modified: 2002/09/09 04:31PM
 Save
 Discard Changes



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

Variable elements in the test script are revealed through a "filter"

WebSphere Studio Workload Simulator - [AA_CheckReservation.js]

Length Validation

Page Element	URL	Response Time
Basic Authentication	www.aa.com/apps/common/js/HM_ScriptDOM.js"	time 310 (ms)
Post Data & Query Strings	GE "www.aa.com/apps/common/js/HM_Arrays.html"	
Cookies	www.aa.com/content/images/hav/more.gif"	
Display Query String	www.aa.com/content/images/hav/more.gif"	
Display Advanced Tags	www.aa.com/content/images/hav/more.gif"	
Page: Thank	Page: Thank time 27739 (ms)	
GET PAGE	GET PAGE "www.aa.com/apps/common/js/calendar.html"	
Page: Thank	Page: Thank time 1453 (ms)	
POST PAGE	POST PAGE "www.aa.com/apps/reservations/FlightSearchResults.html"	
Page: Thank	Page: Thank time 260 (ms)	
GET PAGE	GET PAGE "www.aa.com/apps/common/js/airportcode.html"	
Page: Thank	Page: Thank time 140 (ms)	
GET PAGE	GET PAGE "www.aa.com/apps/common/js/HM_preload.html"	

Name: AA_CheckReservation.js
Created: 2002/09/09 04:31PM
Last Modified: 2002/09/10 07:58AM
Descriptor: AA Home Page and Check



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

Variable content can be changed by editing a html table

Variables

Name	Value
Query String	
_DARGS	%2Fapps%2Fmodules%2FmakeReservation.html.1
Post-Data	
currentCalForm	dep
currentCodeForm	
tripType	round trip
_D_W%3AtripType	+
origin	AA.From
_D_W%3Aorigin	+
departureMonth	AA.DepartMonth
_D_W%3AdepartureMonth	+
departureDay	AA.DepartDay
_D_W%3AdepartureDay	+
departureTime	1
_D_W%3AdepartureTime	+
departureYear	
destination	AA.To
_D_W%3Adestination	+
returnMonth	AA.ReturnMonth
_D_W%3AreturnMonth	+
returnDay	AA.ReturnDay



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

html table containing variable content

From	DepartMonth	DepartDay	To	ReturnMonth	ReturnDay
RDU	10	30	SFO	12	12
SFO	11	14	LGA	11	24
RDU	10	4	JFK	12	21
JFK	12	1	SFO	12	3
RDU	11	1	DFW	11	1
JFK	11	12	ORD	11	28
ORD	10	26	DFW	11	9
JFK	10	1	FRA	11	9
JFK	12	10	CDG	12	18
JFK	11	19	LHR	11	21
RDU	10	6	ORD	12	25
ORD	12	20	SFO	12	22
JFK	11	14	DFW	12	15
JFK	11	11	FRA	12	7
JFK	11	3	CDG	11	23
JFK	10	19	LHR	11	15
RDU	10	24	DFW	11	14
RDU	11	24	LGA	11	29
RDU	10	16	ORD	12	23
RDU	10	10	JFK	10	14



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results

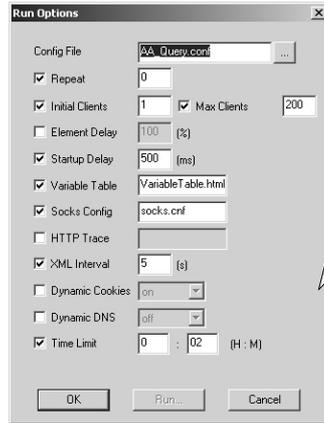
"Weight Distribution" can be incorporated into the test script, to simulate real-life situations

In this example, half of the simulated users will query flights and check for details, the other half will only query flights



- 1 - Create test scripts
- 2 - **Execute and monitor test**
- 3 - Analyze results

Prior to executing the test script, runtime options can be set, e.g.:



- Repeat**: number of times the test is to be repeated
- Initial/Max Clients**: number of virtual users to be simulated
- Start-up delay**: to simulate the delay between users coming on-line (not all users go on-line at exactly the same moment)
- Variable Table**: specifies where the variable content should be drawn from

The runtime parameters are saved in a configuration file that can be re-used



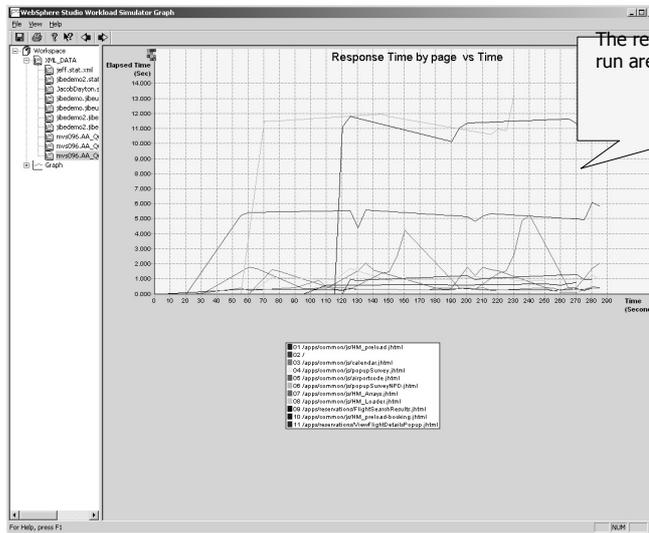
- 1 - Create test scripts
- 2 - **Execute and monitor test**
- 3 - Analyze results

Monitoring can be done through

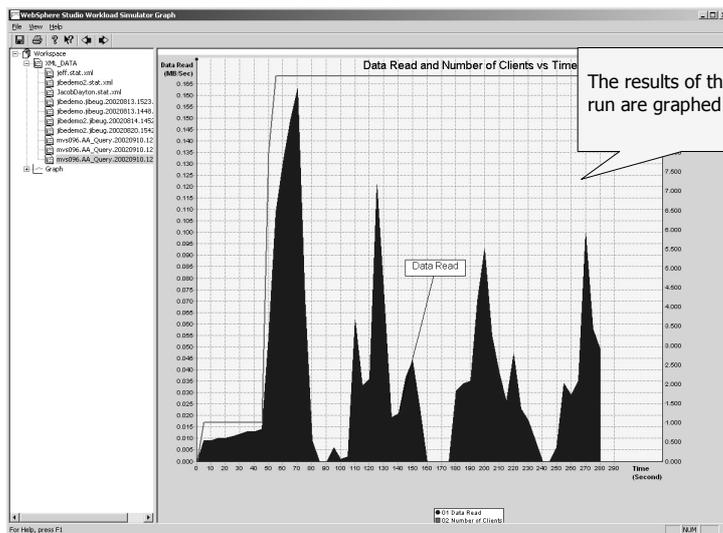
- the Windows interface
- a web browser (useful in extended runs)



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results



- 1 - Create test scripts
- 2 - Execute and monitor test
- 3 - Analyze results





Agenda

- 1. Do We Need Load-Testing?**
- 2. WebSphere Studio Workload Simulator for z/OS**
- 3. Scenario**
- 4. Q&A**