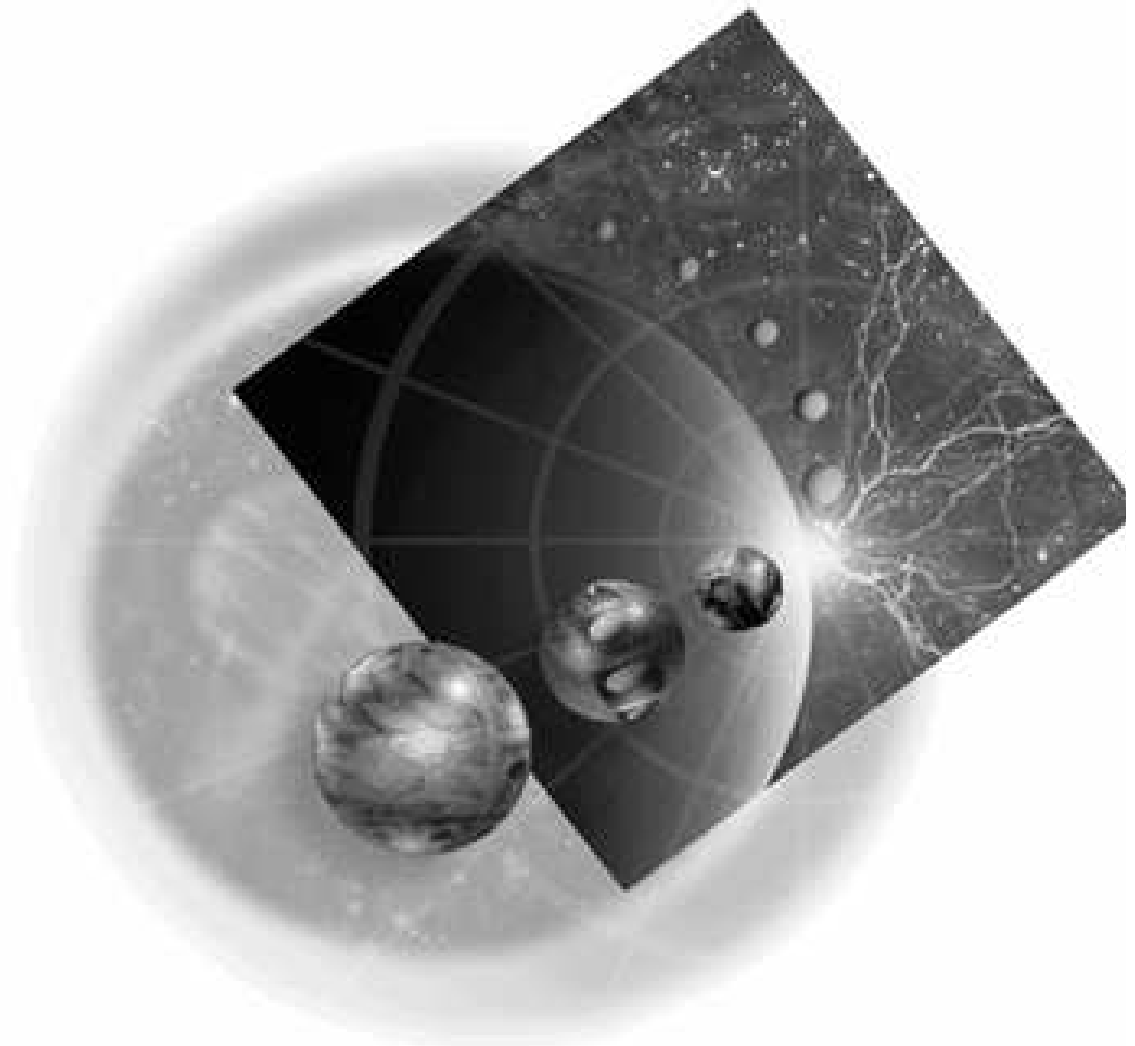


# **IBM Distributed Debugger**

Inge Weiss  
iweiss@ca.ibm.com  
IBM Toronto Lab  
Spring COMMON 2002

Session ID: 404596  
Agenda Key: 44TA



Copyright IBM Canada, Ltd, 2002



e-business

# Acknowledgment and Disclaimer

---

## Acknowledgment:

Many people contributed to this presentation. In particular thanks goes to:  
Vadim Berestetsky, IBM Corporation, Toronto Lab

## Disclaimer:

This presentation may contain examples of code and names of companies or persons. The code is given for presentation purposes and has not been tested by IBM. Therefore IBM does not guarantee the reliability, serviceability, or function of the code and the code is provided "AS IS". IBM EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY. Any names appearing in this presentation are designed to be fictitious and IBM makes no representations as to the accuracy of the names or data presented in accordance therewith.

## Reproduction:

This presentation is the property of IBM Canada, Ltd. Permission must be obtained PRIOR to making copies of this material for any reason.





e-business

# The Agenda

---

- Debugger Overview
- Debugger Startup
  - ▶ Attach Dialog
- Debugger Window
- Debugger Functions
- Other Startup Sequences
- Debugging Java Classes, Threaded Programs and Web Applications
- Demo





e-business

# IBM Distributed Debugger - Overview

---

- RPG, Cobol, CL, C, C++, and Java
- ILE and non-ILE; Step between them
- Batch and interactive
- Client/Server Applications
- Distributed Applications
- Threaded Programs
- JNI calls to C, C++, RPG, and COBOL
- Servlets, JavaServer Pages, and Enterprise Java beans

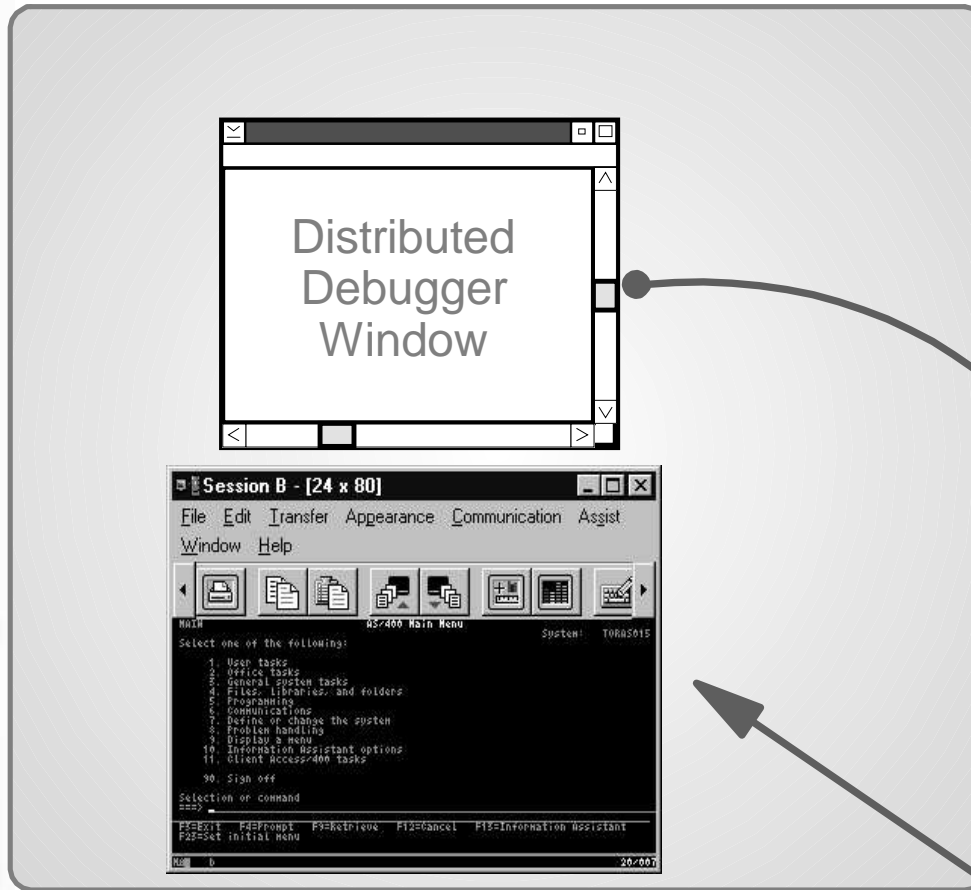


Included in WDS and WAS



e-business

# Environment Overview



Debug server job watches the application on the iSeries.  
Needs information about the application to be debugged:  
Where does it run?

- Job information
- The program to be debugged

Application to be debugged will run on the iSeries

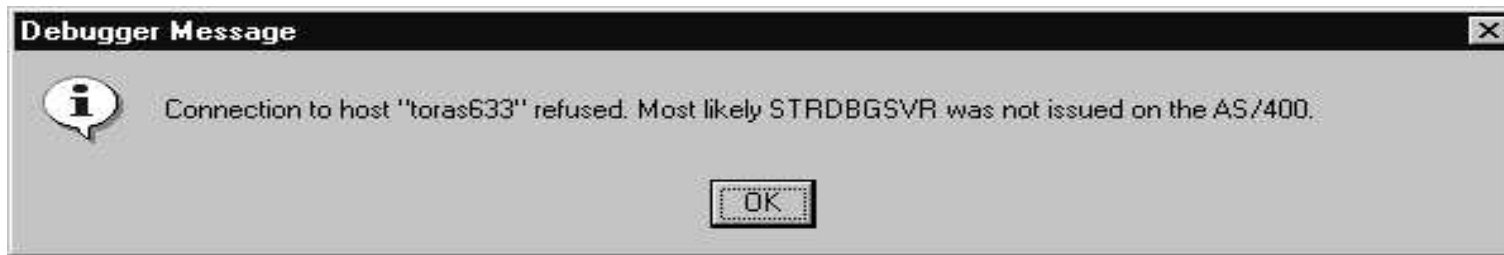




e-business

# Starting the Debug Server

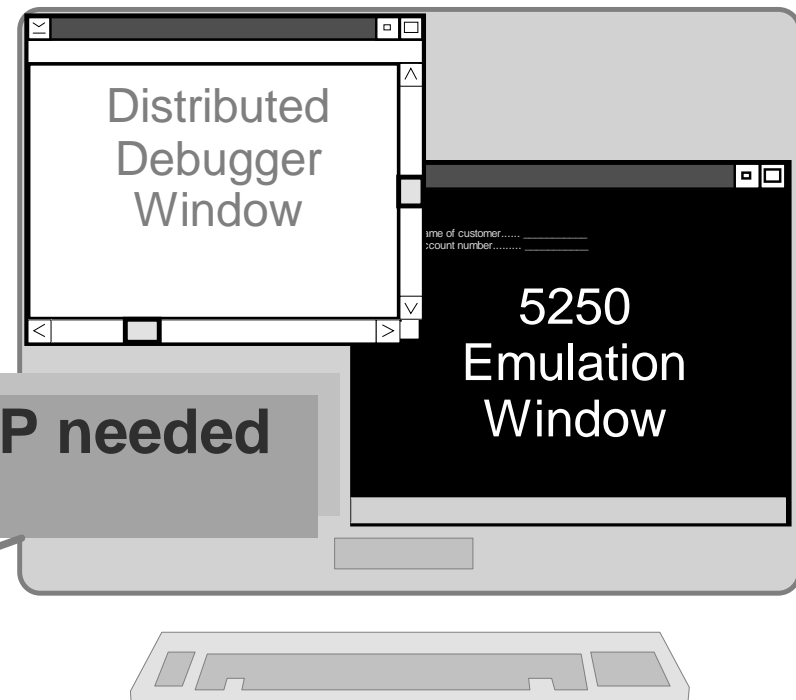
- ▶ On the iSeries start the Debug Server (STRDBGSVR)
- ▶ NOTE: If the Debug Server is not started, an error message appears when you try to sign on from your PC:



- ▶ To debug interactive programs you need a 5250 session.
- ▶ TCP/IP must be configured on workstation and iSeries.



**TCP/IP needed**



**IBM**



e-business

# Debugger Invocation

---

## ■ Start menu

- ▶ Start -> Programs -> IBM WebSphere Development Tools for iSeries -> IBM Distributed Debugger -> IBM Distributed Debugger

## ■ CODE Editor

- ▶ Actions -> Debug

## ■ CODE Project Organizer

- ▶ Context menu for \*PGM object

## ■ IBM VisualAge for Java

- ▶ Context menu for package or class: Tools -> ET/400 -> Debug

## ■ Command line

- ▶ idebug

## ■ WAS

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font with horizontal stripes through them.

IBM



e-business

# Attach Dialog

Host name and port number

**TCP/IP host name**

(e.g. TORAS999)

**TCP/IP host address**

(e.g. 9.99.99.99)

**Default port number is 3001**

**Name of the iSeries job required**

**Fully qualified program name**

**Load saved settings (breakpoints, variables)**

**Load source into debugger**

**Debug Java**







e-business

# Getting Job Information

The screenshot shows the 'Attach' application window. At the top, there are tabs for 'Compiled', 'Interpreted', and 'AS/400'. Below this, there are input fields for 'TCP/IP name of AS/400 host' with 'Host: TORAS06M' and 'Port: 3001'. A 'Job to debug' field contains '/IWEISS/' and a 'Job List..' button is next to it. Below that, a field for 'Specify the name of the program to add to debug:' contains 'EW37TEST/DBGDEMO' and a 'Browse...' button. The bottom part of the window shows a terminal window titled 'Session A - [24 x 80]' with a menu bar (File, Edit, Transfer, Appearance, Communication, Assist, Window, Help) and a toolbar (PrtScrn, Copy, Paste, Send, Recv, Display, Color, Map, Record). The terminal displays the following text:

```

Display Job
Job: QPADEV0001      User: IWEISS      Number: 033066      System: TORAS209
Select one of the following:
  1. Display job status attributes
  2. Display job definition attributes
  3. Display job run attributes, if active
  4. Display spooled files
 10. Display job log, if active or on job queue
 11. Display call stack, if active
 12. Display locks, if active
 13. Display library list, if active
 14. Display open files, if active
 15. Display file overrides, if active
 16. Display commitment control status, if active
More...
Selection
F3=Exit  F12=Cancel

```

**Type: /userid/  
Press: 'Job List' to get  
all jobs on the iSeries  
running under your  
userid**

**Format:  
Jobname/userid/Jobnumber**

Information about  
this job

Use DSPJOB to find your  
job name





e-business

# Using the 'Job List' dialog

Select the job you want to debug and click OK or double click on the job.



Information about this job





e-business

# Debug session preparation



This message indicates you are all set. Now you RUN/START your application.



1) Call your program in the 5250 emulator session

2) Click OK on the message





e-business

# Debug session view

IBM Distributed Debugger

File View Selected Debug Source Stacks Monitors Window Help

CODECOURSE/PAYROLLD:0

Stacks Breakpoints

Stacks

Thread 1

Monitors Locals

Express...

Source

Thread 1: \*PGM CODECOURSE/PAYROLLD

24	F*	- PRJMST	- project mas
25	F*	- RSNMST	- reason code
26	F*	*****	
27	→	FMSTDSP	CF E WORKST
28		FEMPMST	UF A E K DISK
29		FPRJMST	UF A E K DISK
30		FRSNMST	UF A E K DISK
31		D*	
32		D*	Compile time array containing er
33		D ERR	S 50

Source view

Session A - [24 x 80]

File Edit Transfer Appearance Communication Assist Window Help

PrtScr Copy Paste Send Recv Display Color Map Record

PRG01 Time Reporting System Maintenance Selection 0/02/14 17:36:16

Enter an X beside the application you want to maintain

- Employee Master Maintenance
- Project Master Maintenance
- Reason Code Master Maintenance

IBM

Copyright International B

End of Job

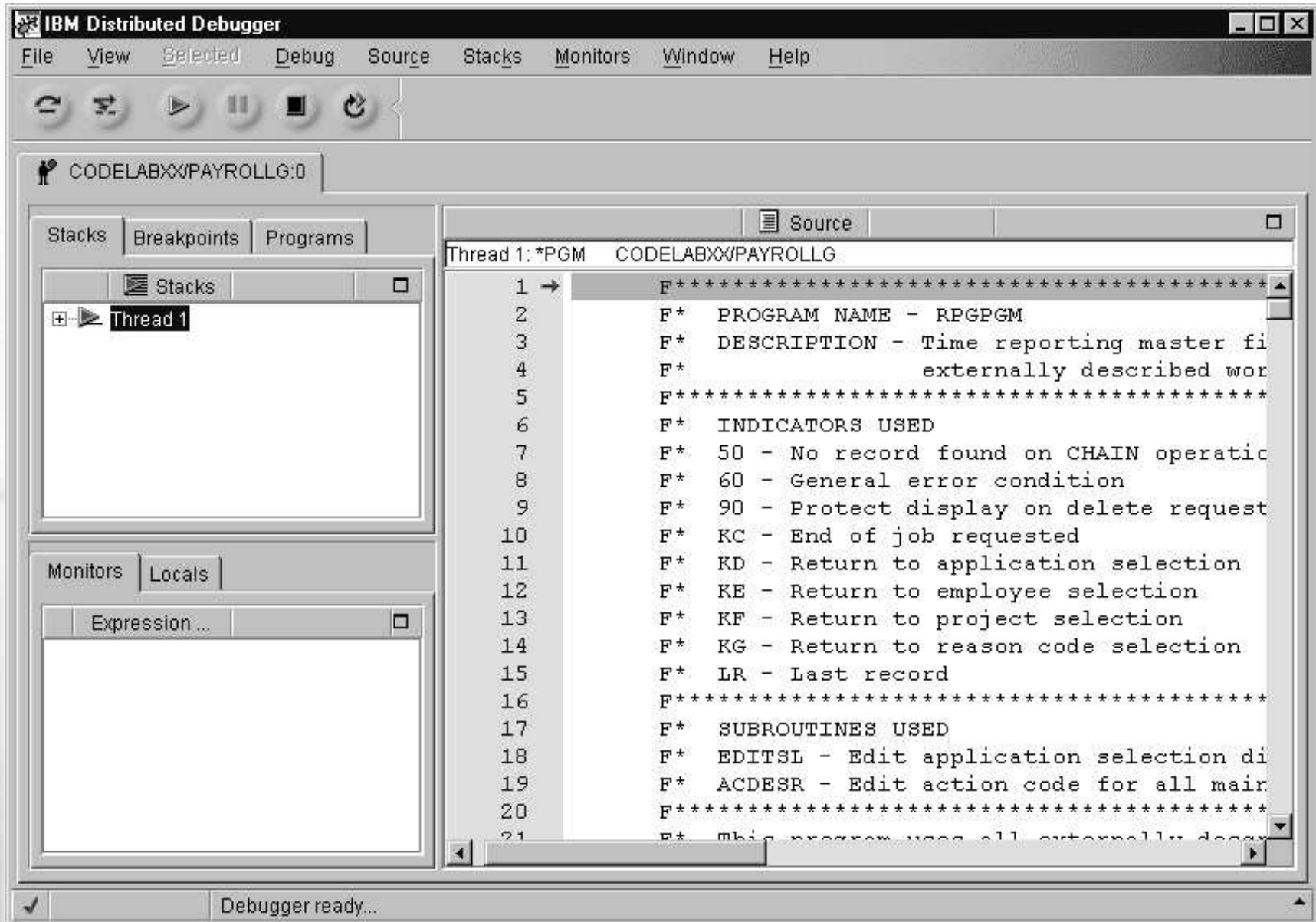
The program runs on the iSeries





e-business

# The Debugger Window





# Source Pane

e-business

## Control program execution

- Run
- Run to location
- Step over
- Step into

## Set breakpoints

## Specify variables to monitor

## Available debug views:

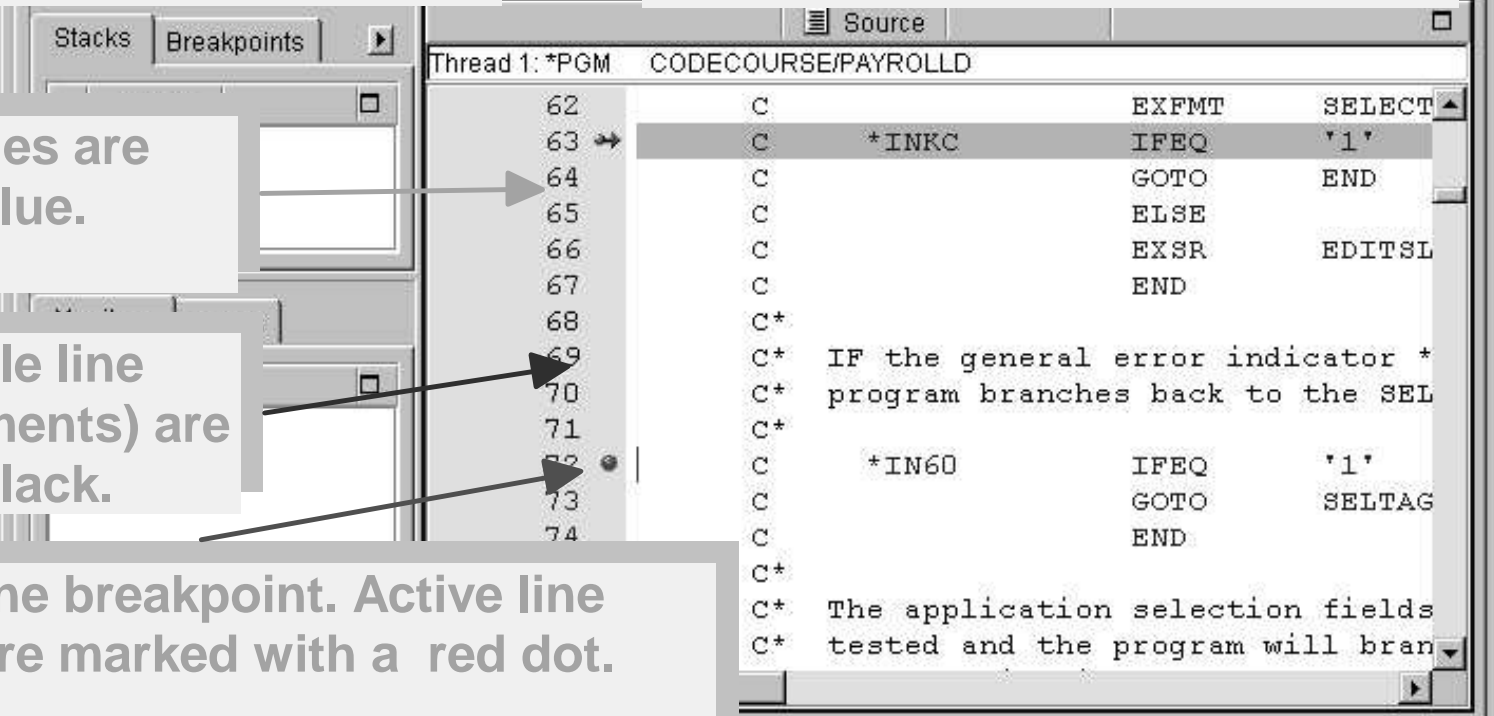
- Source view
- Listing view
- Statement view

**NOTE:** compile your source with appropriate debug options

Executable lines are displayed in blue.

Non-executable line (usually comments) are displayed in black.

Set/delete a line breakpoint. Active line breakpoints are marked with a red dot.





e-business

# Adding a variable to the monitor

## Monitor variables:

Double click on variable to select it  
(highlighted in gray)

Use mouse button 2 to bring up the  
context menu

Select Add to Program Monitor

The screenshot shows the IBM Distributed Debugger interface. The 'Monitors' panel is active, displaying a list of monitored variables: \*IN60 = '0', \*INKC = '0', and EMPAPL = 'X'. The variable EMPAPL = 'X' is highlighted in gray. A context menu is open over this variable, with the option 'Add to Program Monitor' selected. The background shows a code window with assembly-like code and a stack window.

Stacks Breakpoints Programs

Thread 1:

```
77
78
79 C* If EMPAPL (employee maintenance) equals X, th
80 C* branches to label EMPTAG.
81 C* If PRJAPL (project maintenance) equals X, the
82 C* branches to label PRJTAG.
83 C* If the prior two tests were not successful, y
84 C* reason code maintenance. The program will cc
85 C* next executable operation.
86 C*
87 C EMPAPL IFEQ 'X'
88 C GOTO EMPTAG
89 C
90 C PRJAPL IFEQ 'X'
91 C GOTO PRJTAG
92 C
93 C*
94 C*****
95 C* Reason
96 C*****
```

Monitors Locals

Expression ...

- \*IN60 = '0'
- \*INKC = '0'
- EMPAPL = 'X'

monitored variables

Context menu options:

- Add to Program Monitor
- Add to Storage Monitor
- Copy to Clipboard
- Find
- Find Next F3
- Find Procedure





e-business

# Using the Monitor Expression dialog

IBM Distributed Debugger

File View Selected Debug Source Stacks **Monitors** Window Help

Monitor Expression... Shift+F9

Collapse All

Copy View to Clipboard

EW37TEST/EWIRPG05:0

Stacks Breakpoints Programs Source

Stacks

Thread 1

Enter the expression to be evaluated:

AR1(5)

Program monitor

Storage monitor

File: \*PGM CODELAB/PAYROLLG

Line: 1

View: \*Text

Thread: 1

OK Monitor Cancel Help

Expression ...

AR1

AR1(5) = 025...

monitored variables

STRUC2 has 10 occurrences.

D\*-----

Monitor variables:

Select Monitors -->

Monitor Expression

Type the variable name and optionally  
add occurrence

Note: The dialog gets prefilled with any  
selected variable







e-business

# Monitors

## Monitored Variables:

- Values can be changed (double click)
- Representation can be changed (Context menu)

The screenshot shows the IBM Distributed Debugger interface. On the left, the 'Monitors' tab is active, displaying a list of monitored variables under the 'Expression' column. The variables are: \*IN60 = '0', EMPAPL = A7, PRJAPL = '', RSNAPL (selected), and ERR. The ERR variable is expanded to show five error codes: ERR(1) = MAINTENANCE SELE, ERR(2) = MORE THAN ONE APPL, ERR(3) = NO APPLICATION SE, ERR(4) = ACTION CODE NOT E, and ERR(5) = ADD REQUESTED PL. On the right, the 'Source' window displays the source code for 'read 1: \*PGM CODELAB01/PAYROLLWE'. The code includes several lines with comments and control statements like SELTAG, EXFMT, IFEQ, GOTO, ELSE, EXSR, and END. A status bar at the bottom indicates 'Debugger ready...'. A mouse cursor is visible on the left side of the image, pointing towards the monitored variables.





e-business

# Storage Monitor

The screenshot shows the IBM Distributed Debugger interface. A 'Monitor Expression' dialog box is open, showing the expression 'ERR(10)' and the 'Storage monitor' option selected. Below the dialog, the 'Monitors' tab is active, showing a table of memory addresses and their contents. The table has columns for address (09-0F) and content. The content is displayed in 'HEX and Character' style with 16 columns per line.

09	0A	0B	0C	0D	0E	0F	HEX and Character
40	D9	C5	D8	E4	C5	E2	DELETE REQUES
C5	C3	D6	D9	C4	40	C1	TED BUT RECORD A
D3	C5	E3	C5	C4	40	40	LREADY DELETED

Content Style: HEX and Character Columns Per Line: 16

## Storage Monitor:

- Display style can be changed (Character)
- Columns per line can be changed (1 - 16)
- Content can only be changed for teraspace enabled programs



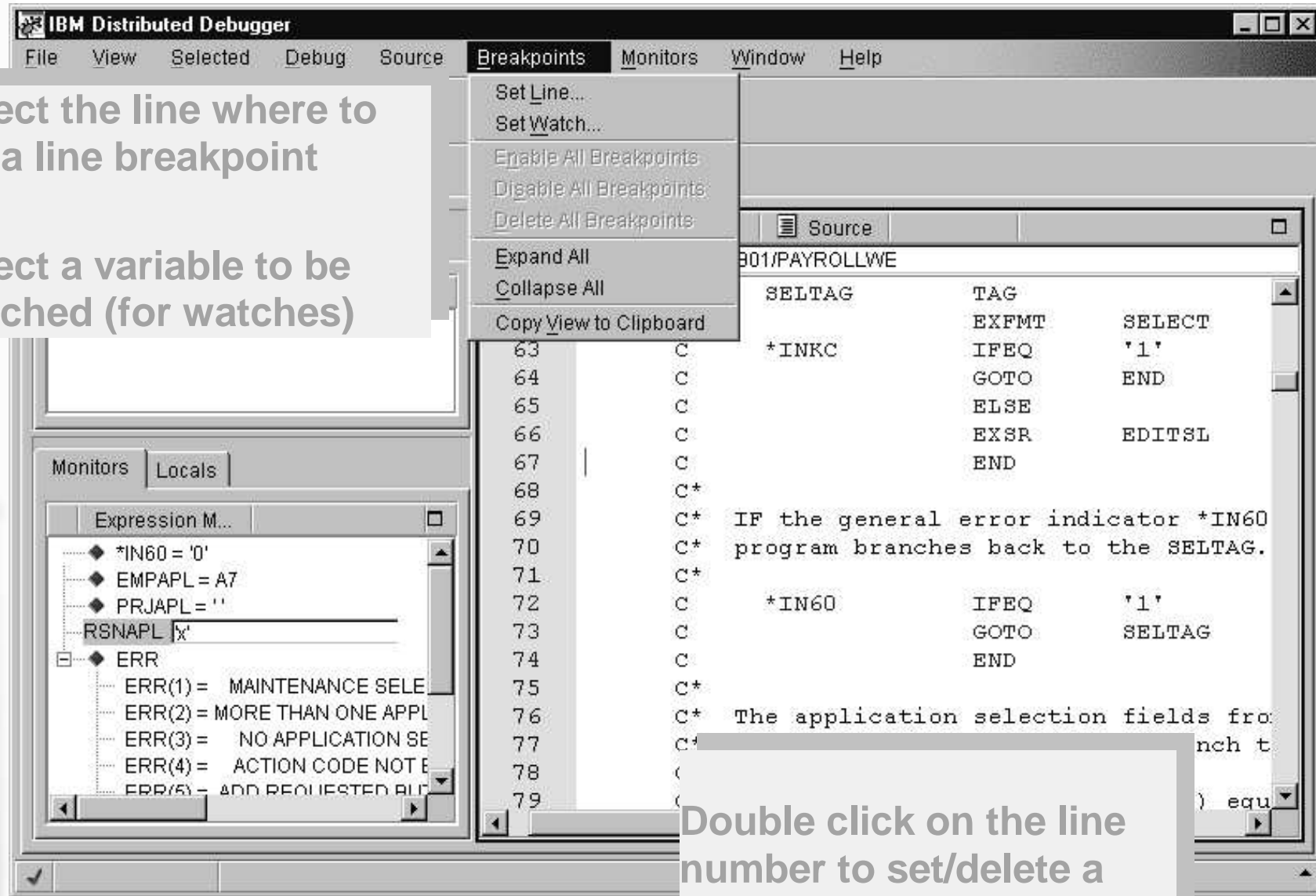
Debugger ready



e-business

# Adding line breakpoints and watches

Select the line where to set a line breakpoint or  
Select a variable to be watched (for watches)



Double click on the line number to set/delete a line breakpoint.





e-business

# Conditional Line breakpoints

**For frequency specify:**  
Value to start  
Value to end  
Number of breaks to skip

**Condition:**  
Specify expression that  
has to hold for the break  
to happen





e-business

# Watch breakpoints

**Watch Breakpoint**

Required Parameters

Expression  
\*IN60

Bytes to Monitor 0

Optional Parameters

Thread  
Every

Frequency  
From: 1  
To: Infinity  
Every: 1

OK Set Default Cancel Help

Specify bytes to be watched

0 means the length of the variable according to its definition

For frequency specify:

Value to start

Value to end

Number of breaks to skip





# Breakpoints page

e-business

The screenshot shows the IBM Distributed Debugger interface. The 'Breakpoints' pane on the left lists several breakpoints, including line breakpoints for source files like 'EW37TEST/DBGDEMO Line: 1' and 'EW37TEST/OPMCL Line: 23'. A context menu is open over the 'OPMCL Line: 23' breakpoint, showing options: 'Disable Breakpoint Ctrl+F9', 'Delete Breakpoint F9', 'Modify Breakpoint...', and 'Properties...'. The main window displays source code for 'Thread 1: \*PGM'. Line 23 is marked with a grey dot, indicating a disabled breakpoint. The code includes statements like 'CHGVAR VAR(&VAR1) VALUE(300)', 'CHGVAR VAR(&VAR1) VALUE(400)', 'GOTO LABEL1', and an 'IF COND(&VAR1 \*EQ 100) THEN(DO)' block.

## Get list of breakpoints

→ all breakpoints defined for this debug session

Disabled line breakpoints are marked with a grey dot

## Work with individual breakpoints:

- Delete
- Disable
- Modify line breakpoints
- Display Properties





e-business

# Running the program

The screenshot shows the IBM Distributed Debugger interface. The 'Debug' menu is open, highlighting the 'Run' option (F5). The 'Tool bar' contains a 'Run' button. The source code window displays the following code:

```
* MAIN SUBROUTINE
C   MAIN          BEGSR
C   BEGIN         TAG
C               MOVE      '0'
C               MOVE      *BLANK
C               MOVE      *BLANK
C               MOVE      *BLANK
C               MOVE      *BLANK
C*
C*   Write the SELECT format to displ
C*   branch to END TAG; ELSE edit SEL
C*
C   SELTAG        TAG
C               EXFMT      SELECT
C   *INKC         IFEQ      '1'
C               GOTO      END
C               ELSE
C               EXSR      EDITSL
C               END
```

To Run the program use:

- Tool bar button
- Debug menu
- F5

Program runs until:

- next Line breakpoint
- Watch is triggered
- program termination

→ you press Halt





e-business

# Run program until watch is hit

Once the watch is hit

The following message informs the user of the stop cause and position

The screenshot shows the IBM Distributed Debugger interface. A 'Debugger Message' dialog box is open, displaying the text: "Variable 'charval' has changed in program '\*PGM EW37TEST/DBGDEMO', module 'DBGDEMO'." with an 'OK' button. Below the dialog, the code window shows assembly instructions. Line 35 is highlighted, and a mouse cursor is pointing at it. A callout box points to line 35 with the text: "Execution stops at line after change occurred".

Line	Instruction	Operand 1	Operand 2	Operand 3
24	*			
25	C	Z-ADD	4	DUM1
26	C	Z-ADD	5	DUM2
27	C	CALLP	PROC2	
	*			
	C	Z-ADD	2	DUM1
	C	Z-ADD	3	DUM2
	C	eval	num = -234	
	C	eval	charval = 'num = '	
	*			
	*			
35	C	CALLP	PROC2	
36	*			
37	C*			
38	C*	Calling an internal subroutine.		
39	C*			
40	C			

Execution stops at line after change occurred







e-business

# Run to location

## Run to a specific location

- Select the line you want to run to
- Choose 'Run to location' action from the 'Debug' menu or the pop-up menu in the prefix area.
- Execution stops at selected location

The screenshot shows the IBM Distributed Debugger interface. The main window displays the source code for the program CODECOURSE/PAYROLLD. The code is as follows:

```
Thread 1: *PGM CODECOURSE/PAYROLLD
63  C      *INKC      IFEQ      '1'
64  C      GOTO      END
65  C      ELSE
66  C      EXSR      EDITSL
67  C      END
68  C*
69  C* IF the general error indicator *IN60 is
70  C* program branches back to the SELTAG.
71  C*
72  C      *TM60      IFEQ      '1'
73  C      GOTO      SELTAG
74  C      END
75  C*
```

The 'Debug' menu is open, and the 'Run to Location' option is highlighted. The 'Monitors' panel on the left shows the following expressions:

- \*IN60 = '0'
- EMPAPL = 'X'
- PRJAPL = ''
- RSNAPL = ''

The status bar at the bottom indicates 'Debugger ready...'.





e-business

# Step into vs. Step over

Step into :

→ allows debugging next level of program invocation

IBM Distributed Debugger

File View Select Debug Source Breakpoints Monitors Window Help

CODELABXXVCL1:0

Stacks Breakpoints Programs

Breakpoints

- Line Breakpoints
  - Source: \*PGM CODELABXXVCL1

Monitors Locals

Expression ...

- &NAME = Claus
- &COUNT = 00100.

Thread 1: \*PGM CODELABXXVCL1

```
1 PGM
2 dcl      &name *char 10
3 DCL     VAR(&count) TYPE(*DEC) LEN(5 0)
4 CHGVAR  VAR(&NAME) VALUE('Claus Weiss')
5 LOOP:   chgvar  var(&count) value(&count+1)
6         if (&count *lt 100) then(goto loop)
7 CHGVAR  VAR(&NAME) VALUE('Claus')
8 → CALL   PGM(CODELABxx/PAYROLLWE)
9
10        ENDPGM
```

The Debugger supports stepping into different program types  
OPM (non ILE) to ILE or vice versa  
Any language to language (CL, RPG, COBOL, C, C++)



e-business

# Listing view

Listing view allows display of:

- Copy members
- Input/Output specs from external described files
- Statement numbers

Thread 1: \*PGM CODELAB03/PAYROLLG

47	46 C*			
48	47=ISELECT			
49	48=I	A	1	1 EMPAPL
50	49=I	A	2	2 PRJAPL
51	50=I	A	3	3 RSNAPL
52	51=IEM/SEL			
53	52=I	S	1	6 OEMPNO
54	53=I	A	7	7 ACODE
55	54=IEMPMNT			
56	55=I	A	1	30 ENAME
57	56=I	A	31	31 EMCAT
58	57=I	A	32	36 EDEPT
59	58=I	A	37	66 ELOCN
60	59=I	A	67	74 EUSRI
61	60=I	S	75	77 LENHRS
62	61=IPR/SEL			
63	62=I	A	1	8 PRCDE
64	63=I	A	9	9 ACODE
65	64=IPRJMNT			
66	65=I	A	1	50 PRDSC
67	66=I	A	51	80 PRRSP
68	67=I	S	81	86 OPRSTR
69	68=I	S	87	92 OPREND

Stacks: Thread 1, PAYROLLG, \_QRNP\_PEP\_PAYROLLG, QUICMD, QUIMGFLW, QUIMNDRV, QUICMENU, QCMD

Monitors: Expression ...

- ◆ EMPAPL = ''
- ◆ PRJAPL = ''
- ◆ RSCDE = null





e-business

# Programs pane

Add and remove programs, service programs, Java classes.

- View list of modules for each program
- View list of procedures for each module

Indicate the name of the program, service program, or Java Class to add to debug.

The screenshot shows the IBM Distributed Debugger interface. The 'Programs' pane is open, displaying a tree view of the current program 'EW37TEST/DBGDEMO:0'. The tree shows a hierarchy: 'PGM EW37TEST/DBGDEMO' (expanded) contains 'DBGDEMO' (expanded), which contains 'DBGDEMO' (expanded) and 'PROC1'. The 'Add Program' dialog box is open, with 'codelabxx/payroll' entered in the 'Program name' field. The 'Program type' section has three radio buttons: 'Program' (selected), 'Service Program', and 'Java Class'. The 'OK', 'Cancel', and 'Help' buttons are visible at the bottom of the dialog. A mouse cursor is pointing at the 'DBGDEMO' module in the Programs pane. A text box at the bottom right of the screenshot contains the text: 'Programs, modules and procedures. Clicking on module or procedure name brings up its source.'





# Stacks pane

e-business

IBM Distributed Debugger

File View Selected Debug Source Breakpoints **Stacks** Monitors Help

✓ Show All Stack Information  
Expand All  
Collapse All  
Copy View to Clipboard

DBGDEMO:0

Stacks Breakpoints Programs

Stacks Control Panes

Thread 1 [0] (Runnable)

- ILECL Program:DBGDEMO Module:ILECL
- DBGDEMO Program:DBGDEMO Module:DBGDEMO**
- \_QRNP\_PEP\_DBGDEMO Program:DBGDEMO
- InvokeTargetPgm(qpl\_pcp\_cb\*) Program:QP0ZPCPN
- Qp0zNewProcess Program:QP0ZPCPN Modul
- main Program:QP0ZPCP2 Module:QP0ZPCP2
- \_CXX\_PEP Program:QP0ZPCP2 Module:QP0ZPCP2

Monitors Locals

Expression Monitors Value P...

◆ NUM = -234

Thread 1: \*PGM EW37TEST/DBGDEMO

```
42 C*-----  
43 C*  Calling an ILE CL module  
44 C*-----  
45 → C          CALLB  'ILECL'  
46 C*-----  
47 C          CALLP  PROC2  
48 C*-----  
49 C*-----  
50 C*  Calling an OPM CL program  
51 C*-----  
52 C          CALL   'OPMCL'  
53 C*-----  
54 C*-----
```

**Series Call Stack:**

- Select 'Show all Stack Information'
- Program, module, procedure, statement number
- Display associated source by clicking on a stack entry





e-business

# Debugging batch jobs

**Attach**

Compiled | Interpreted | AS/400

TCP/IP name of AS/400 host

Host: TORAS644

Job to debug

/iweiss/036538

Specify the name of the program to add to

codelabxx/testbatch

Use program profile

Step into

Debug Java

Debugger Settings...

Attach Cancel Help

**Session B - [24 x 80]**

File Edit Transfer Appearance Communication Assist Window Help

MAJOR Major Command Groups System: TORAS015

Select one of the following:

1. Select Command by Name
2. Verb Commands
3. Subject Commands
4. Object Management Commands
5. File Commands
6. Save and Restore Commands
7. Work Management Commands
8. Data Management Commands
9. Security Commands
10. Print Commands
11. Spooling Commands
12. System Control Commands
13. Program Commands

SLTCMD  
VERB  
SUBJECT  
CMDOBJMG  
CMDFILE  
CMDSAVRST  
CMDWRKMGT  
CMDDTAMGT  
CMDSEC  
CMDPRT  
CMDSPL  
CMDSYCTL  
CMDPGM  
More...

Selection or command  
==> SBJJOB CMD(CALL PGM(CODELAB01/TESTBATCH)) JOBQ(CODELAB01/LABQ)

F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant  
F16=AS/400 Main menu  
Job 001079/WEISS/QDFTJOB submitted to job queue LABQ in library CODELAB01.

MA b

→ **Submit job to the Jobq with the HOLD option set to \*YES:**

- SBJJOB CMD(CALL PGM(CODELAB01/TESTBATCH)) HOLD(\*YES)

→ **Start debugger from the Start menu or command line**

→ **Get job info of submitted job**

→ **Start debug session**

→ **Debugger will release the job and display the source when the program starts running.**

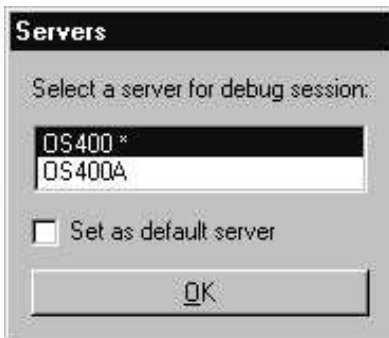
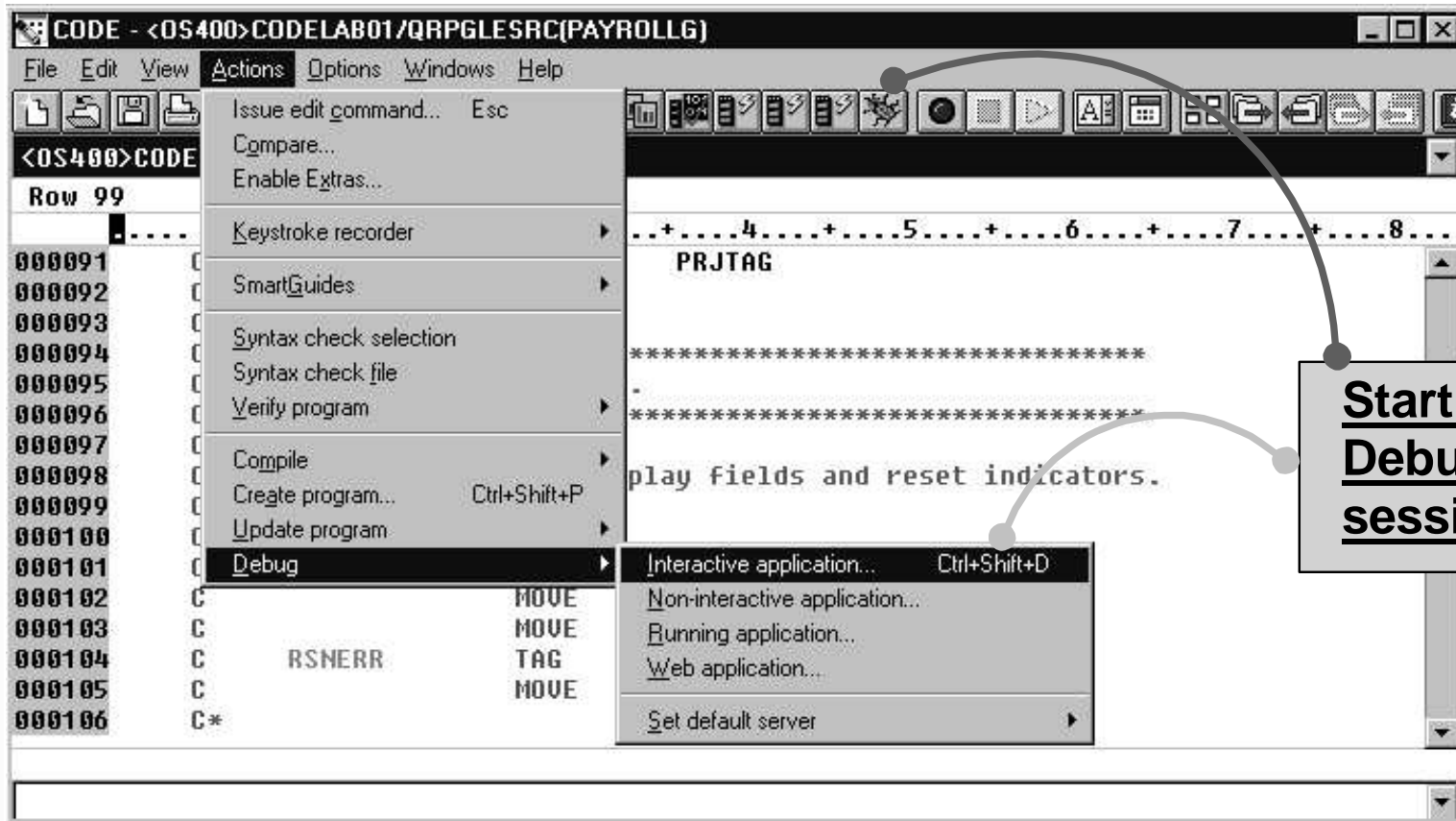
**Note: If you start from the CODE editor, you will have to release the job yourself.**





e-business

# Start from Editor (Interactive application)



## Servers dialog:

- Lists all active STRCODE servers
- Allows you to set default interactive debug server





e-business

# Start from Editor (Interactive appl.) cont.

The STRCODE session Emulation Window is used for all 5250 interaction in your debugged application.

Prompt for Call command  
Prefilled with name of  
program and library from  
editor

Can specify a program  
other than the prefilled  
one.

Specify parameters for  
invocation of program

Call Program

General Options

Program

Program: PAYROLLG

Library: CODELAB05

Parameters: \*LIST

Add

Delete

CALL PGM(CODELAB05/PAYROLLG)

OK Cancel Default Help

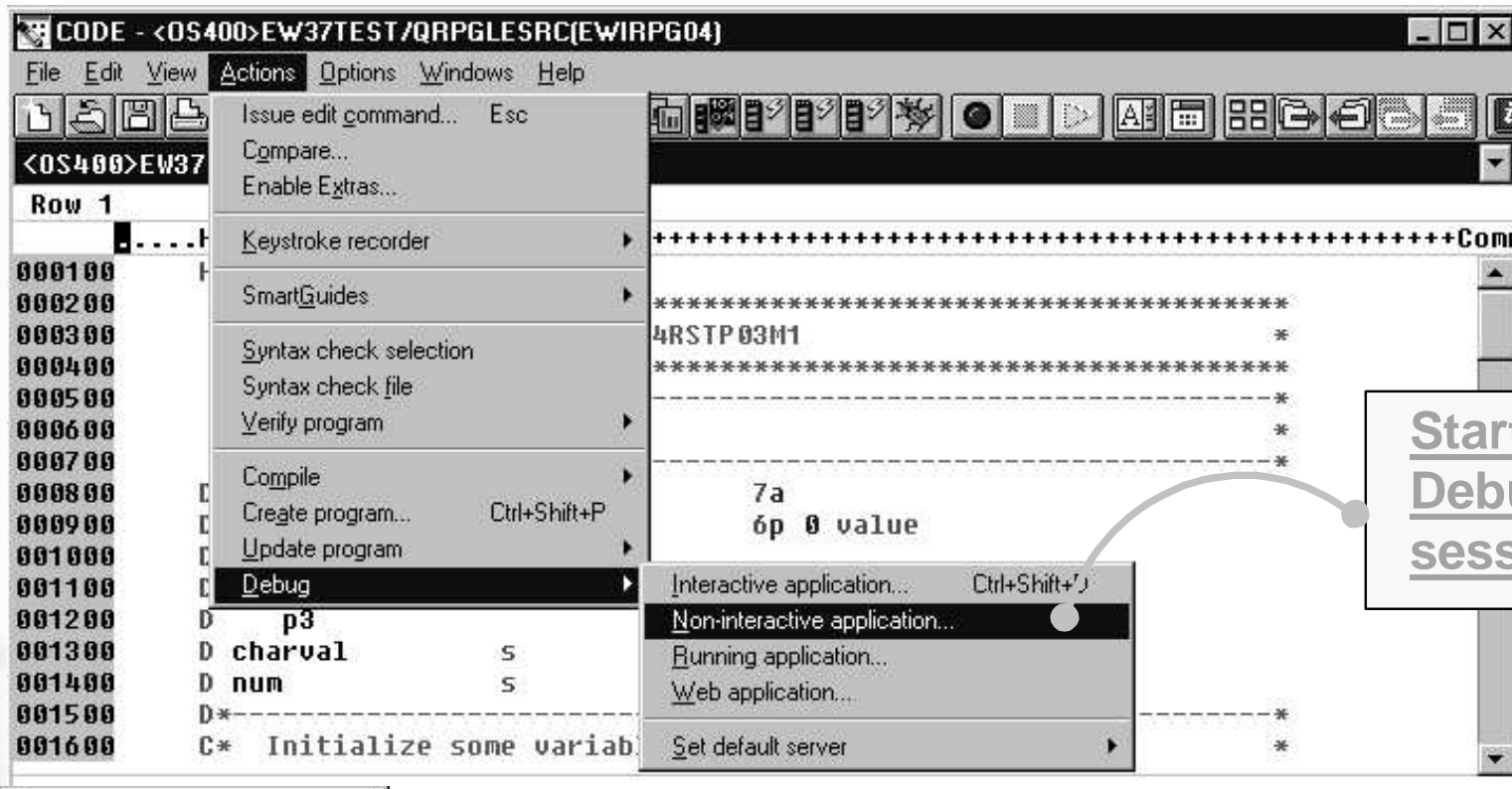




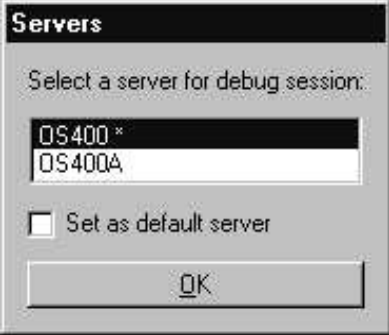


# Start from Editor (Non-interactive application)

e-business



Start  
Debug  
session



**Servers dialog:**

- ◆ Lists all active servers, STRCODE and TCP/IP auto-connect
- ◆ Allows you to set default non-interactive debug server

Starts program in a BCI job in QSYSWORK  
Library list has to be supplied





e-business

# Load Program dialog

Load program

Compiled | Interpreted | AS/400

TCP/IP name of AS/400 host

Host: TORAS644 Port: 3001

Specify the name of the program to debug:

EW37TEST/DBGDEMO Browse...

Enter any program parameters:

Use program profile  
 Step into  
 Debug Java

Debugger Settings...

Load Cancel Help

Host name and port number

**TCP/IP host name**  
(e.g. TORAS999)

**TCP/IP host address**  
(e.g. 9.99.99.99)

**Default port number is 3001**

Stop at the first executable statement

Bring up the 'Debugger Settings' window

Available from the File menu to invoke:

→ additional debug sessions

→ a new session after program termination

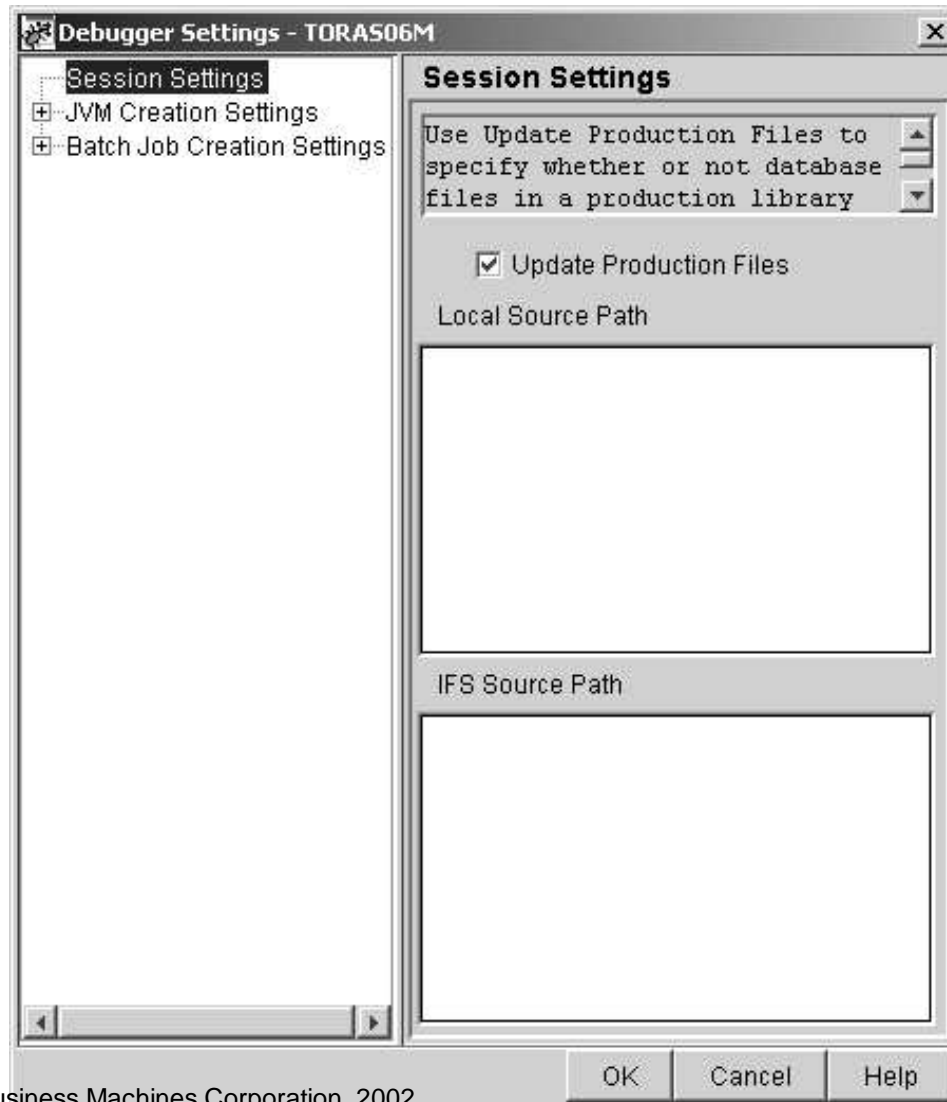




e-business

# Debugger Settings - Session Settings

## Debugger Settings are per host



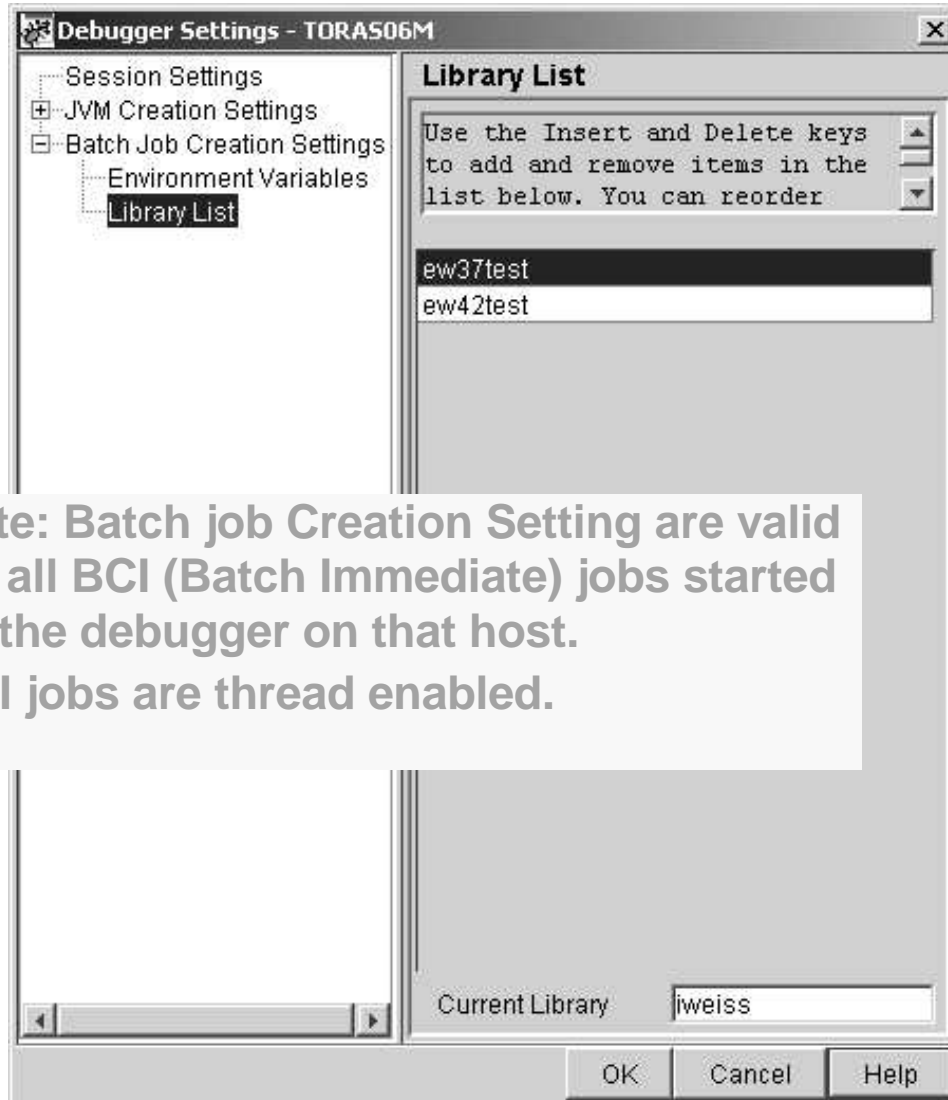
- Check 'Update Production Files' to allow access to production libraries while debugging.
- ◆
- Use 'Local Source Path' to search for locally stored source files.
- ◆
- Use 'IFS Source Path' to search for source files stored in the iSeries IFS.





e-business

# Debugger Settings - Batch Job Creation Settings



Note: Batch job Creation Setting are valid for all BCI (Batch Immediate) jobs started by the debugger on that host. BCI jobs are thread enabled.

- The Batch Job Creation Settings Library List will be used for debugging non-interactive applications (Load Program dialog).
- ◆
- Use Insert and Delete keys to add and remove entries.
- ◆
- Use drag and drop to re-order the entries in the list.
- ◆
- Single click selects the entry, double click allows editing.





e-business

# Starting in Browse mode

---

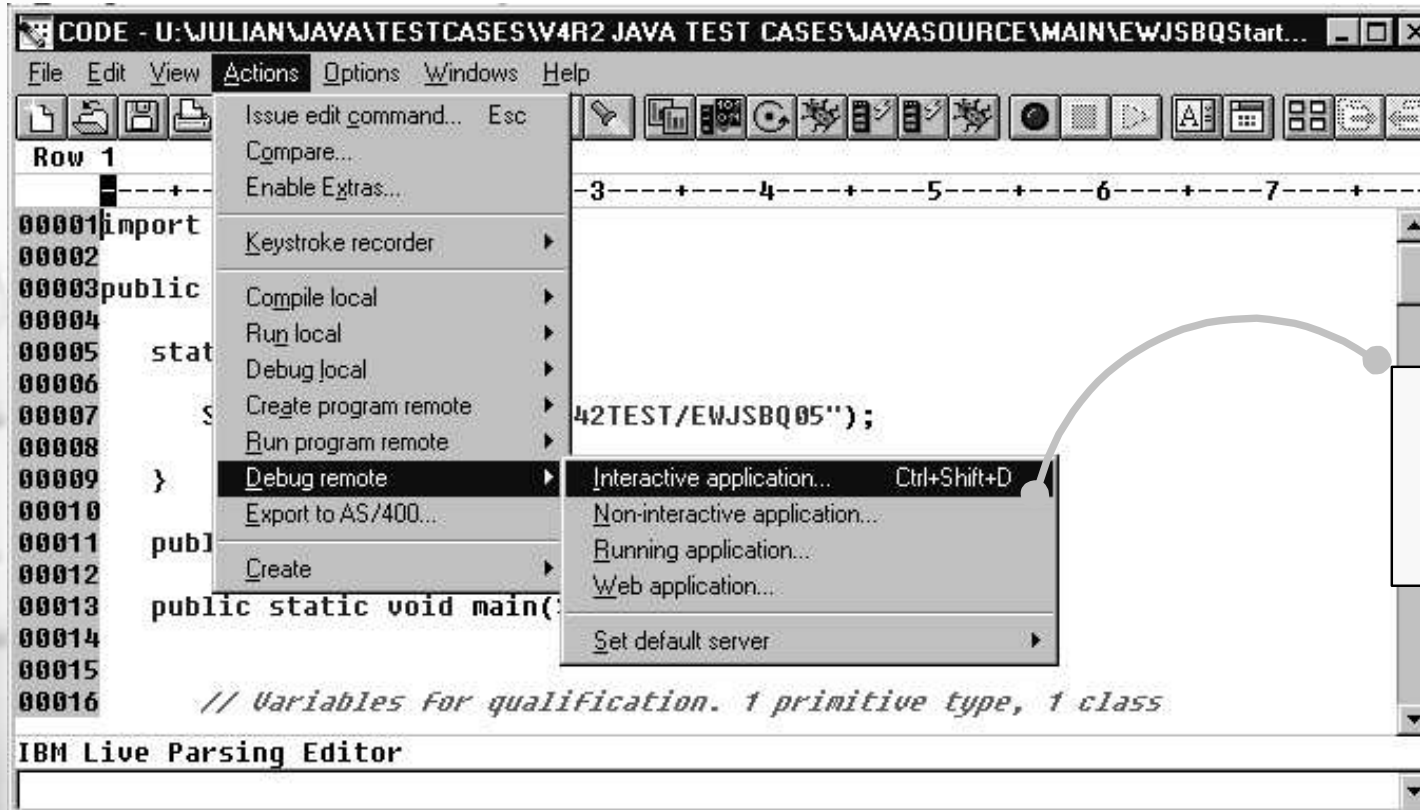
- In Attach or Load Program dialog:
  - Step into is not selected
- In Debugger Window:
  - Bring up your source from the Programs page
  - Browse the source and set line breakpoints.  
Watches are not available while browsing.
  - Run.
- Start your program on the iSeries when message tells you to.
- Dismiss message.





e-business

# Java Startup from the Editor



Start  
Debug  
session

Can use STRCODE or TCP/IP auto-connect server  
Displays Run Java Program dialog similar to Call Program dialog





e-business

# Java Startup

---

If you want the debugger to create the JVM for you, use the Load program dialog.

To attach to a running JVM, use the Attach dialog and specify its job name.

## Make sure 'Debug Java' is selected

Enter the name of your Java class and all required parameters...

Enter the name of your Java class.

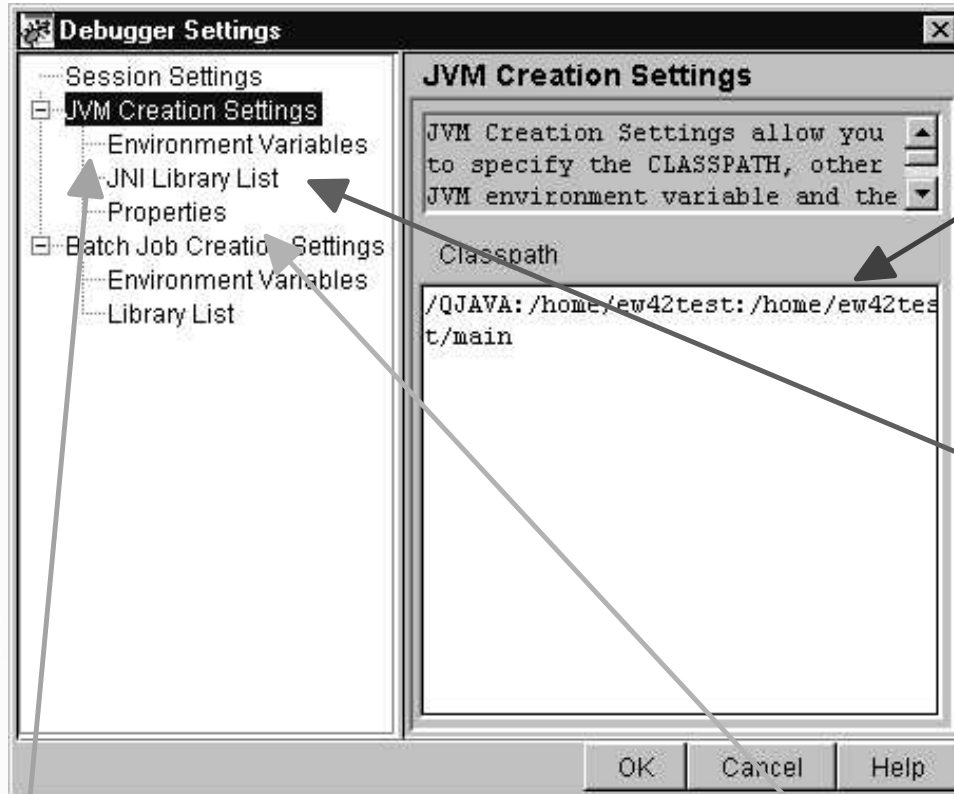
Your Java application can have I/O, which gets redirected to the AS/400 Java console...





e-business

# Debugger Settings - JVM Creation Settings



You must set the CLASSPATH here if the debugger creates the JVM for you. Directories are separated by colon.

Use the JNI Library List page to set Library List and Current Library for calls to non Java programs.

Use the Environment Variables page to setup Java Environment Variables:  
Supply Variable name and value

Use the Properties page to setup Java Properties:  
Supply Property name and Property value







e-business

# Java Source Window

Java classes can only be debugged in Source view. Use compile option -g (debug) if you want to change variables.

The Programs page lists all Java classes that are currently under debug.

The screenshot displays the IBM Distributed Debugger interface. The 'Programs' tab is active, showing a tree view of loaded classes: `*CLASS /EWJSBQThreadStartup` (containing `<init>` and `main`) and `*CLASS /jsbq/EWJSBQ02` (containing `jsbq/EWJSBQ02`). The source code window shows the following code for `Thread 1: *CLASS /EWJSBQTh`:

```
23
24 // start 3
25 → for(i=0;i<j
26     jsbqThread
27 }
28
29 // change reference of szString and value of a
30 // the reference of szString and value of a should NO
31 // by all threads
32 szString = "main 1";
33 a = 10;
34
35
36 // waiting all the threads finished
37 try{
38     for(i=0;i<jsbqThread.length;i++){
39         jsbqThread[i].join();
40     }
41 } catch(InterruptedException e){}
42
43
44
45
```

The 'Locals' tab is also visible, showing the following variables:

- `a = 0`
- `i = 1`
- `jsbqThread`
- `jsbqThread[0]`
  - `activeThreadQ = null`
  - `threadInitNumber = 4`
  - `MIN_PRIORITY = 1`
  - `NORM_PRIORITY = 5`

The status bar at the bottom indicates 'Debugger ready...'

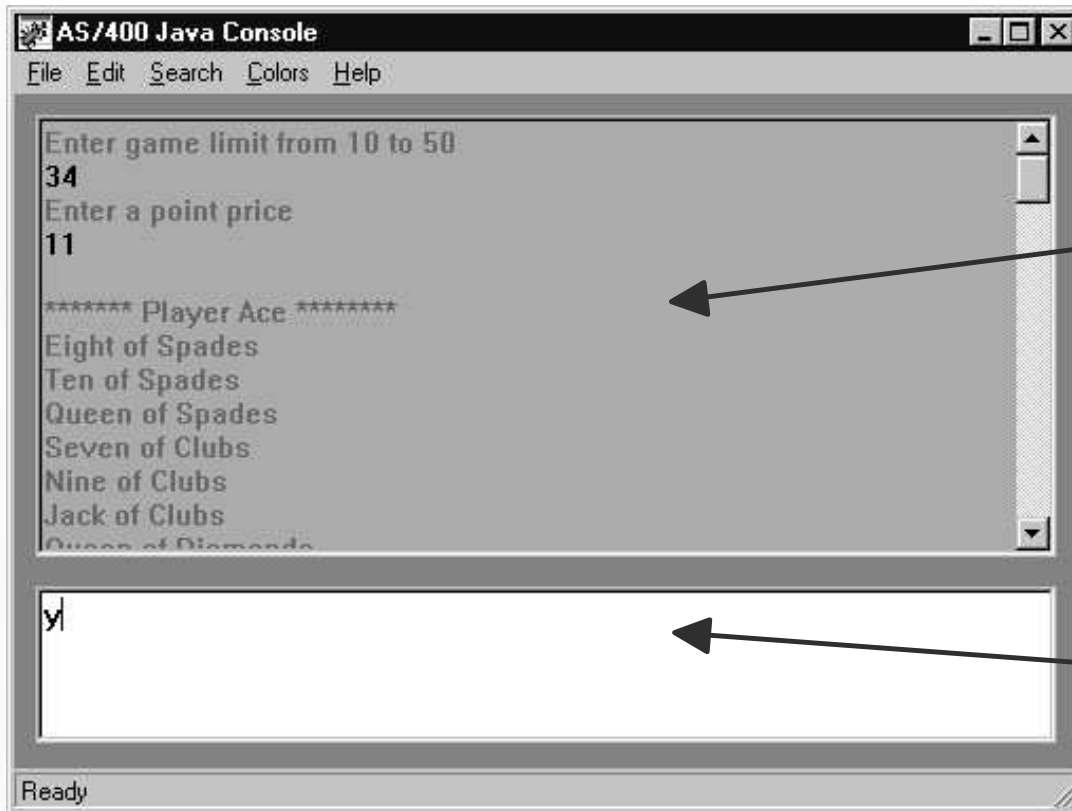




e-business

# Java Console

Standard Out, Standard Error, and Standard Input are redirected to the Java Console Window.



Session I/O history log.  
The entire log can be saved into a file.

Input area: Enter key is the trigger.





e-business

# Thread Startup

Load program

Compiled | Interpreted | AS/400

TCP/IP name of AS/400 host  
Host: TORAS644 Port: 3001

Specify the name of the program to debug:  
ew42test/ewcexp Browse...

Enter any program parameters:  
42 abc

Use program profile  
 Step into  
 Debug Java

Debugger Settings...

Your application should not have any screen I/O.

Enter the name of your application

Enter all required parameters...

Use 'Batch Job Creation Settings' to specify Environment Variables and Library List for your BCI job.





e-business

# Debugging multi-threaded applications

Initial thread is always Thread 1

The screenshot shows a Java IDE debugger interface. The main window displays the source code of a thread, with a grey triangle next to line 17, indicating it is disabled. A 'Properties for Thread 3(EWJSBQThreadStartup:0)' dialog box is open, showing the following table:

Property	Value
State	Suspended
Debug state	Frozen
Priority	0
Current Position	
System TID	0

The 'Stacks' window shows a tree view of threads. Thread 1 is the root. Thread 2 is expanded, showing its stack frames. A grey triangle is next to Thread 2 in the tree view. The 'Monitors' window shows the state of monitors, including 'jsbqThread' and 'jsbqThread[0]'. The 'Locals' window shows the current thread's local variables, including 'a = 0', 'i = 1', and 'jsbqThread'. The 'Expression' window shows the current thread's expression, 'activeThreadQ = null'. The 'Debugger ready...' status bar is visible at the bottom.

Disabled threads are:  
→ listed as Frozen in the Thread Properties  
→ marked with a grey triangle in the Stacks page





e-business

# Local Variables

The screenshot displays the IBM Distributed Debugger interface. The main window is titled "IBM Distributed Debugger" and contains several panes:

- Stacks:** Shows a call stack for "Thread 3". The stack includes frames for "Thread 3", "LE\_Create\_Thre", "startThread(void)", "run.java/lang/Thr", "run.jsbq/EWJSBQ02", and "doSomething.jsbq".
- Locals:** Shows local variables for "Thread 3". The variables are: "this = jsbq/EWJSBQ02:E670B6C", "count = 0", "temp = 0", "sb", and "serialVersionUID = 3388685".
- Source:** Shows the source code for "Thread 3: \*CLASS /jsbq/EWJSBQ02". The code includes a loop with a switch statement. The current execution point is at line 28: "count += 10;".

Arrows indicate the relationship between the stack frames and the local variables pane. A text box with the text "Entries for multiple threads in the Stacks and Locals page" is overlaid on the interface, with arrows pointing to the "Thread 3" entry in both the Stacks and Locals panes.

```
Thread 3: *CLASS /jsbq/EWJSBQ02
14   StringBuffer sb = new StringBuffer();
15
22   case 0:
23   case 5:
24   case 10:
25     count += 5;
26     break;
27   case 15:
28     count += 10;
29     break;
30   default:
31     // do nothing
32   }
33 }
34
35 // set count = 0
36 count = 0;
```





e-business

# Job-Wide and Thread specific breakpoints

Line Breakpoints that qualify to 'Every' thread are called Job-Wide breakpoints.

The screenshot shows a debugger window for a thread named 'EWJSBQThreadStartup:0'. The 'Breakpoints' tab is active, showing a list of line breakpoints. Two breakpoints are listed, both with the source '\*CLASS /jsbq/EWJSBQ02'. The breakpoint on line 28 is selected and highlighted. The 'Monitors' tab is also visible, showing the state of local variables: 'a = 0', 'i = 1', and 'jsbqThread'. The 'jsbqThread' object has 'activeThreadQ = null', 'threadInitNumber = 4', and 'MIN\_PRIORITY = 1'. The code editor shows the following code:

```
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25 case 10:  
26     count += 5;  
27     break;  
28 case 15:  
29     count += 10;  
30     break;  
31 def:  
32     /  
33 }  
34 }  
35 // s  
36 coun
```

The 'Thread 3: \*CLASS' window shows the following details for the breakpoint:

Property	Value
Class	jsbq/EWJSBQ02
Source	*CLASS /jsbq/EWJSBQ02
Method Name	doSomething
Line	28
Address	
State	Enabled
Status	Active
Thread	3
Conditional Expression	
From	1
To	Infinity
Every	1

Line breakpoints that qualify to a particular thread are called thread breakpoints.





e-business

# Debugging WebSphere Applications



- Start Administrative Console
- Enable Debug and Object Level Trace

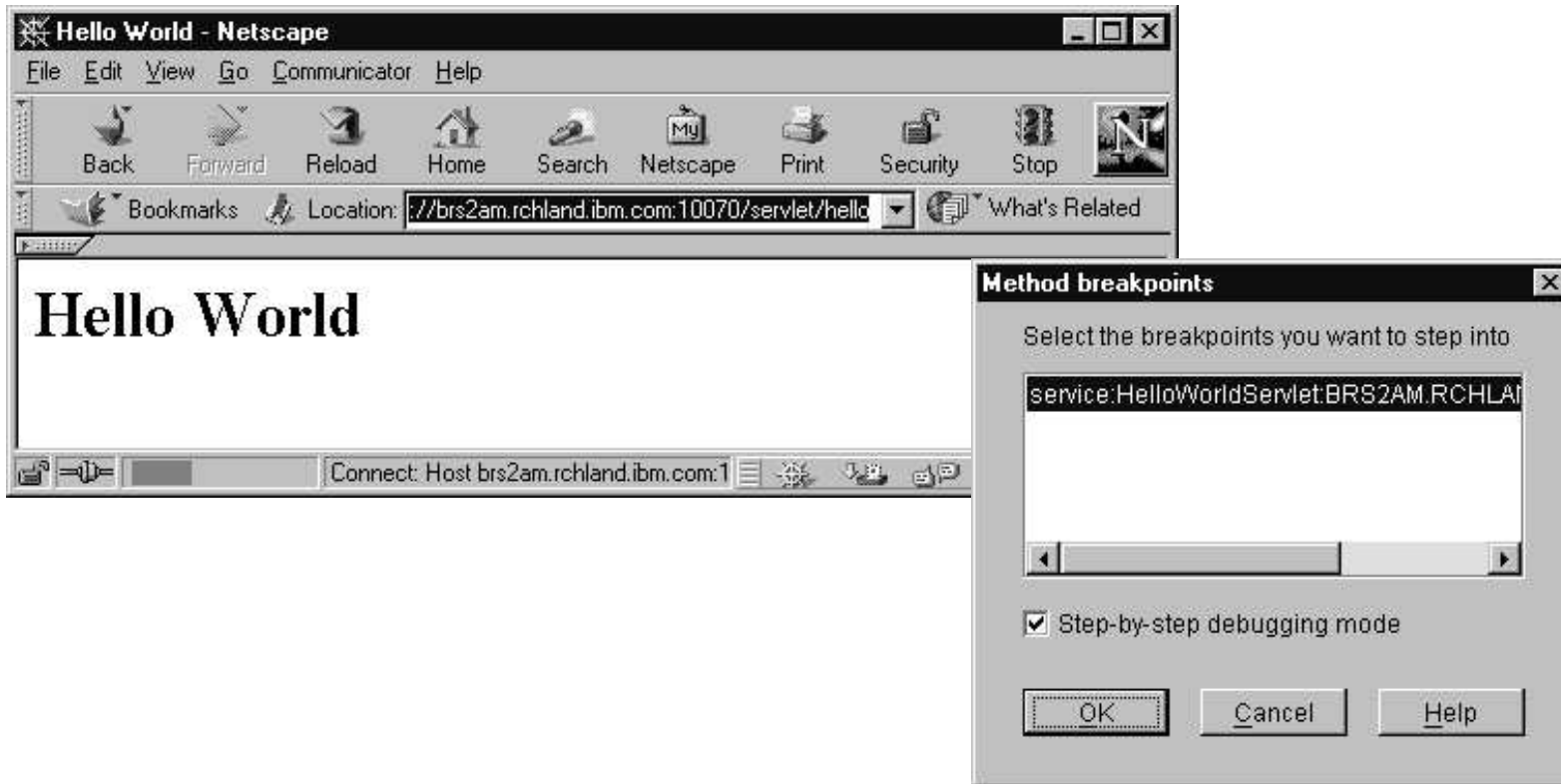




e-business

## Debugging WebSphere Applications (cont.)

- Start OLT viewer
- Start the application server



- The debugger will be started for you







e-business

# IBM Distributed Debugger

---

- Full source-level debugger
- Provides integrated Graphical User Interface, common to other Visual Age debuggers.
- Off-load application development from host.
- Improved developer productivity.





e-business

# WDT/400 Information

---

## ● Additional Information

▶ [www.ibm.com/software/ad/wdt400](http://www.ibm.com/software/ad/wdt400)

## ● Education

▶ **S6186 CODE/400 for iSeries -- Basic (2 days)**

▶ **S6205 CODE/400 for iSeries -- Advanced (1 day)**

## ● Order number

▶ **V4R5M1: 5769-WDS**

▶ **V5R1: 5722-WDS**

## ● Newsgroup

▶ **news://news.software.ibm.com/ibm. software.code400**

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font with horizontal stripes through the letters.

IBM

# **iSeries & WebSphere Resources & Deliverables**

## **Series site**

[www.iseries.ibm.com/](http://www.iseries.ibm.com/)

## **Series WebSphere Application Server**

<http://www-1.ibm.com/servers/eserver/series/software/websphere/wsappserver/>

## **PartnerWorld for Developers, iSeries & WebSphere**

<http://www.iseries.ibm.com/developer/websphere/>

## **BM eServer Solutions**

<http://www-1.ibm.com/servers/eserver/series/solutions/>

## **Series e-business Solutions**

<http://www-1.ibm.com/servers/eserver/series/ebusiness/>

## **Series B2B Solutions**

<http://www-1.ibm.com/servers/eserver/series/btob/>

## **Connect for iSeries**

<http://www-1.ibm.com/servers/eserver/series/btob/connect/v11high.html>

## **WebSphere Commerce Suite for iSeries**

<http://www-1.ibm.com/servers/eserver/series/ebusiness/wcs51.html>

## **Series and e-commerce**

<http://www-1.ibm.com/servers/eserver/series/ebusiness/ecommerce.htm>

## **Series HTTP Server**

<http://www-1.ibm.com/servers/eserver/series/software/http/index.html>

## **WebSphere Development Studio for iSeries**

<http://www-3.ibm.com/software/ad/wds400/>

## **Series and WebSphere References**

<http://www.as400.ibm.com/developer/java/solutions/jjem.html>

<http://www2.software.ibm.com/casestudies/swcsweb.nsf/platform>

## **Series Solution Finder**

<http://www.iseries.ibm.com/btobpartner/>

## **iSeries & Domino**

<http://www-1.ibm.com/servers/eserver/series/domino/>

## **Dedicated Server for Domino**

<http://www-1.ibm.com/servers/eserver/series/domino/dsd.htm>

## **Workload Estimator for iSeries, WAS, WCS & Domino, HTTP Server, Java, etc.**

<http://as400service.ibm.com/estimator/>

## **iSeries Custom Technology Center**

<http://www-1.ibm.com/servers/eserver/series/service/ctc/>

## **iSeries Technical Support**

<http://as400service.ibm.com/>

## **iSeries Technical Studio**

<http://www.as400.ibm.com/tstudio/>

## **1st Install for iSeries & WebSphere Application Server**

<http://www.iseries.ibm.com/developer/websphere/assistance.html>

## **iSeries ToolsNet (Tools & Middleware)**

<http://www.iseries.ibm.com/developer/tools/>

## **iSeries & Services Network**

<http://as400service.ibm.com/supporthome.nsf/document/19251245>

## **iSeries e-business Handbook (SG24-5694-01)**

<http://www.redbooks.ibm.com/abstracts/sg245694.html>

## **WebSphere Commerce Suite With Back-End Order Mgmt.**

<http://ibm.com/redbooks>

## **iSeries Technology Center**

<http://www.iseries.ibm.com/service/itc/ebiz.htm>

## **iSeries University**

<http://www-3.ibm.com/services/learning/community/as400/>

# **iSeries & WebSphere Resources & Deliverables**

## **WebSphere Application Server Overview**

<http://www-4.ibm.com/software/webservers/appserv/>

## **WebSphere Commerce Suite**

[www-4.ibm.com/software/webservers/commerce/](http://www-4.ibm.com/software/webservers/commerce/)

## **WebSphere Payment Manager**

[www-4.ibm.com/software/webservers/paymgr/](http://www-4.ibm.com/software/webservers/paymgr/)

## **MQSeries**

[www.ibm.com/software/ts/mqseries](http://www.ibm.com/software/ts/mqseries)

## **iSeries & WebSphere Documentation**

### **Redbooks & Red Pieces**

### **Form Numbers/Web Sites**

**Building iSeries Applications for WebSphere Advanced Edition** SG24-5691

**Building Java Applications for the iSeries with VisualAge for Java** SG24-6245

**Integrating WCS with Domino Back-End Applications**

<http://ibm.com/redbooks> (search for REDP0141)

### **Java & WebSphere Performance on iSeries**

- <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg246256.html?Open>
- **iSeries Application Development Directions white paper is now available**
- <http://www.iseries.ibm.com/developer/tools/documents/addr/index.html>
- **Connect for iSeries with WebSphere Commerce Suite Red Paper**
- <http://www.redbooks.ibm.com/redpapers/pdfs/redp0127.pdf>

### **Tools for Application Reface and Redesign**

- <http://www.as400.ibm.com/developer/comm/pidtechpapers.html?Tools>
- **Introduction to Enterprise JavaBeans for AS/400** SG24-5192-00
- **Web enabling AS/400 Applications with WebSphere Studio** SG24-5634-00
- **Building AS/400 Applications with WebSphere Standard Edition 2.0** SG24-5635-00
- **Building AS/400 C/S Apps with Java** SG24-2152-02
- **Building AS/400 Internet-based applications with Java** SG24-5337-00

## **IBM Redbooks**

<http://www.redbooks.ibm.com/>

## **iSeries Nation**

<http://www-1.ibm.com/servers/eserver/series/announce/form.html>