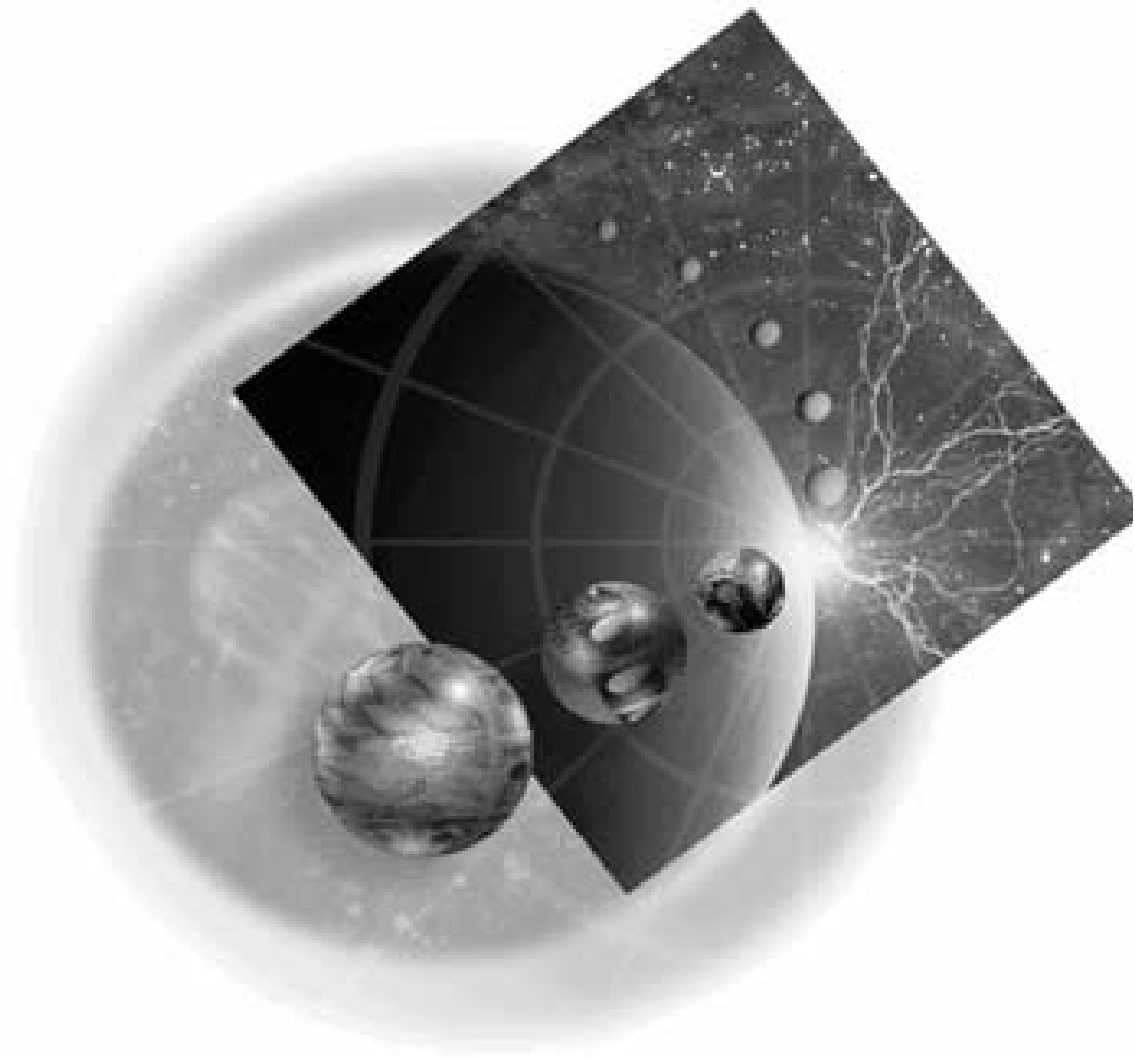


IBM Distributed Debugger

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

Session ID: 404596
Agenda Key: 53TB



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.





The Agenda

- ✿ Debugger Overview
- ✿ Debugger Startup from the CODE Editor
- ✿ Debugger Window
- ✿ Debugger Functions
- ✿ Other Startup Sequences
- ✿ Debugging Threaded Programs, Java Classes and Web Applications
- ✿ Demo





e-business

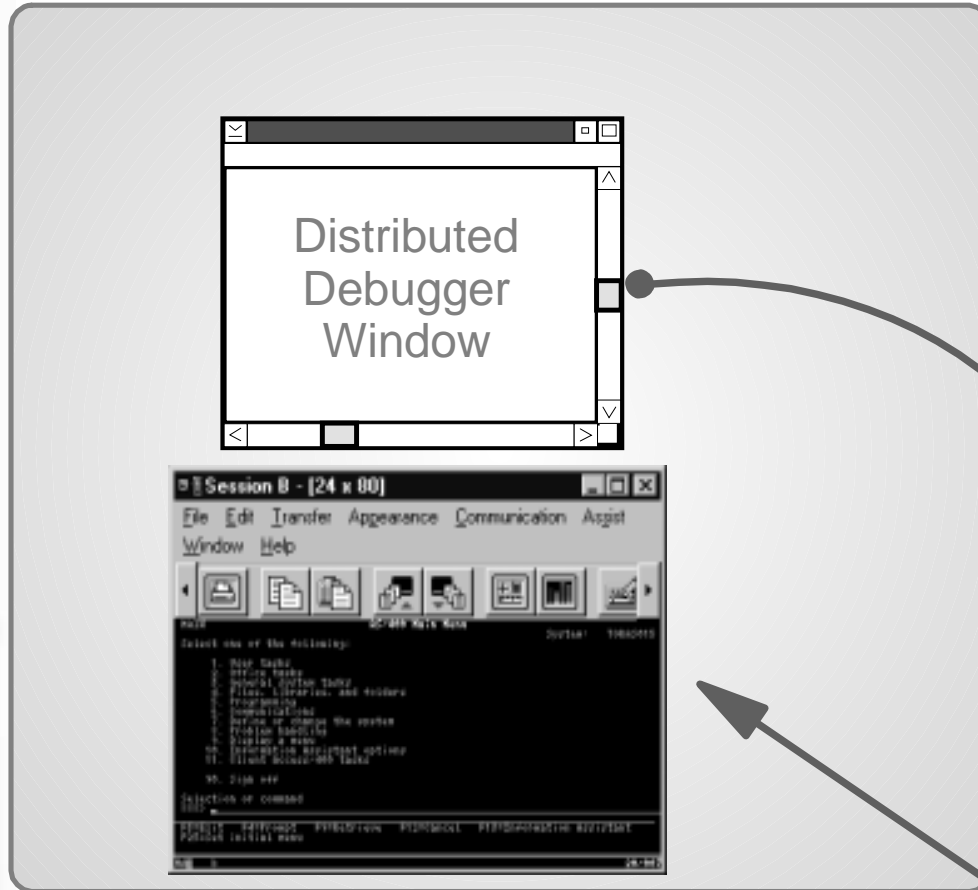
IBM Distributed Debugger - Overview

- RPG, Cobol, CL, C, C++, and Java
- I LE and non-I LE; 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



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





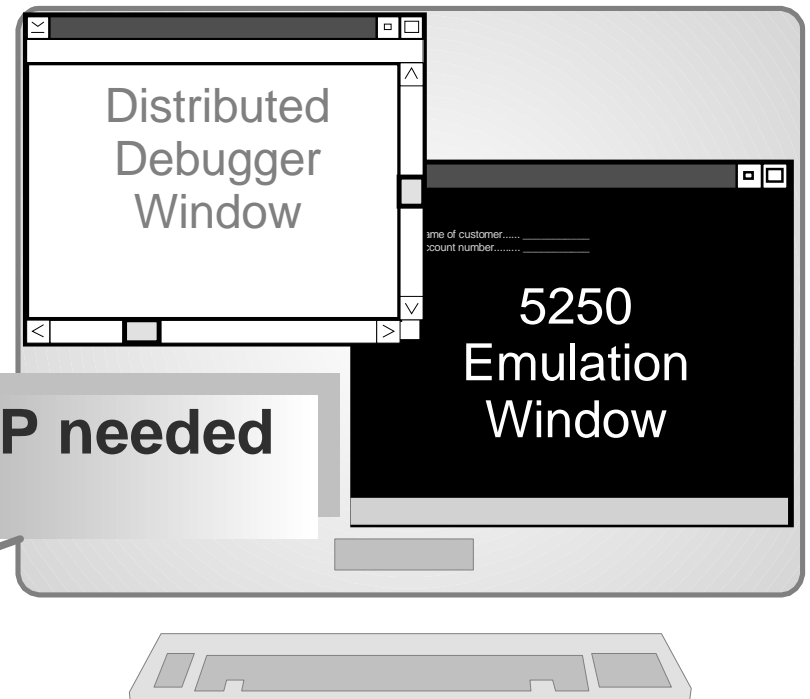
e-business

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.





Debugger Invocation

- **CODE Editor**

- ▶ Actions -> Debug

- **Remote Systems Explorer**

- ▶ Context menu of a program or job

- **Start menu**

- ▶ Start -> Programs -> I BM WebSphere Development Studio Client for iSeries -> I BM Distributed Debugger -> I BM Distributed Debugger

- **Command line**

- ▶ idebug

- **WAS**





Starting from the CODE Editor

- Actions -> Debug ->
 - Interactive application...
 - Non-interactive application...
 - Running application...
 - Web application...

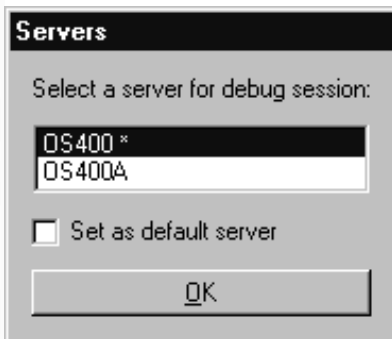
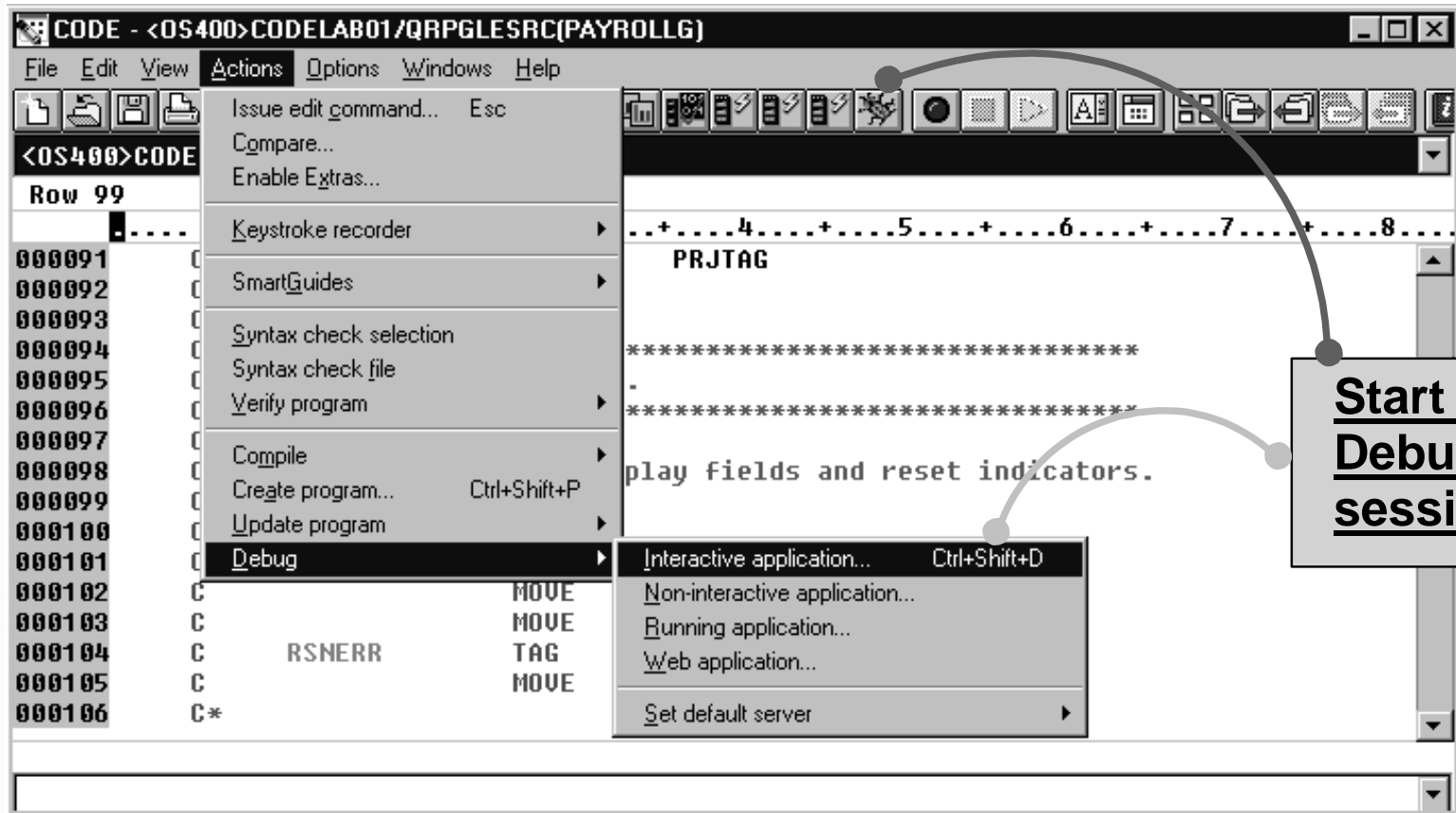
Can also use the debug tool button or
Ctrl+Shift+D for interactive application





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)

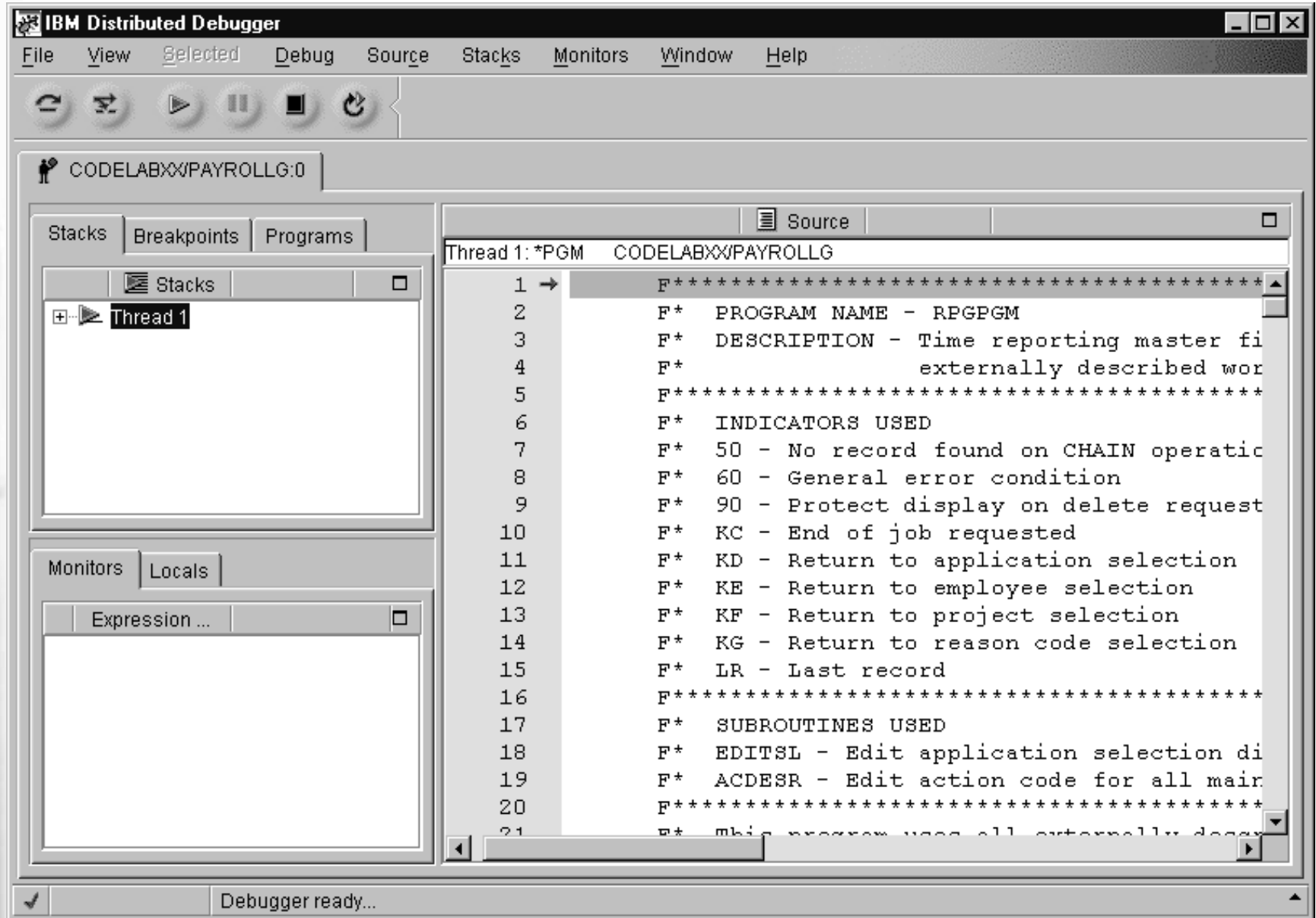
OK Cancel Default Help





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.

Stacks Breakpoints

Stacks

Monitors Locals

Thread 1: *PGM CODECOURSE/PAYROLLD

62	C	EXFMT	SELECT
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 *	
70	C*	program branches back to the SEL	
71	C*		
72	C	*IN60	IFEQ '1'
73	C	GOTO	SELTAG
74	C	END	
	C*		
	C*	The application selection fields	
	C*	tested and the program will bran	



e-business

Listing view

Listing view allows display of:

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

The screenshot shows a debugger window with the following components:

- Stacks:** Thread 1, PAYROLLG, _QRNP_PEP_PAYROLLG, QUICMD, QUIMGFLW, QUIMNDRV, QUICMENU, QCMD.
- Monitors:** Expression ...
 - ◆ EMPAPL = ''
 - ◆ PRJAPL = ''
 - ◆ RSCDE = null
- Source:** Thread 1: *PGM CODELAB03/PAYROLLG

Line	Code	Label	Offset	Code	Label
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=IEMFSEL				
53	52=I	S	1	6	OEMPNO
54	53=I	A	7	7	ACODE
55	54=IEMPMT				
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=IPRJSEL				
63	62=I	A	1	8	PRCDE
64	63=I	A	9	9	ACODE
65	64=IPRJMT				
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





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 expressions: `*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 the following code:

```
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 PRJTAG  
92 C  
93 C*  
94 C*****  
95 C* Reason F3  
96 C*****
```

monitored variables





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

Monitor Expression

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

Monitors Locals

Expression ...

- AR1
- AR1(5) = 025.

monitored variables

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 'Thread 1'. The variables are: *IN60 = '0', EMPAPL = A7, PRJAPL = '', RSNAPL (with a cursor), 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 BLT. On the right, the 'Source' window displays the source code for program CODELAB01/PAYROLLWE. The code includes comments and statements such as SELTAG, EXFMT, *INKC, IFEQ, GOTO, ELSE, EXSR, and END. A status bar at the bottom indicates 'Debugger ready...'. Arrows from the text on the left point to the 'RSNAPL' variable and the 'ERR' variable in the Monitors window.





e-business

Storage Monitor

The screenshot shows the IBM Distributed Debugger interface. A 'Monitor Expression' dialog box is open, allowing the user to configure a storage monitor. The dialog has a text field containing 'ERR(10)' and radio buttons for 'Program monitor' and 'Storage monitor'. The 'Storage monitor' option is selected. Below the radio buttons, there is a table with the following data:

File:	*PGM	CODELAB/PAYROLLG
Line:		101
View:		*Text
Thread:		1

Buttons for 'OK', 'Monitor', 'Cancel', and 'Help' are at the bottom of the dialog. In the background, the debugger's 'Monitors' pane shows 'ERR(10)' selected. Below it, the 'Value Panes' section displays a memory dump table:

Value Panes							
Go to Address 0xCFC3BA584200B302							
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

At the bottom of the Value Panes section, there is a 'Content Style' dropdown set to 'HEX and Character' and a 'Columns Per Line' spinner set to '16'. The status bar at the bottom of the debugger window reads 'Debugger ready...'

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)

The screenshot shows the IBM Distributed Debugger interface. The 'Breakpoints' menu is open, displaying options: Set Line..., Set Watch..., Enable All Breakpoints, Disable All Breakpoints, Delete All Breakpoints, Expand All, Collapse All, and Copy View to Clipboard. The source code window shows the following code:

```
63 C *INKC TAG EXFMT SELECT
64 C IFEQ '1'
65 C GOTO END
66 C ELSE
67 C EXSR EDITSL
68 C*
69 C* IF the general error indicator *IN60
70 C* program branches back to the SELTAG.
71 C*
72 C *IN60 IFEQ '1'
73 C GOTO SELTAG
74 C END
75 C*
76 C* The application selection fields fro
77 C* nch t
78 C
79 C ) equ
```

The 'Locals' window shows the following variables:

- *IN60 = '0'
- EMPAPL = A7
- PRJAPL = ''
- RSNAPL [x]
- ERR
 - ERR(1) = MAINTENANCE SELE
 - ERR(2) = MORE THAN ONE APPL
 - ERR(3) = NO APPLICATION SE
 - ERR(4) = ACTION CODE NOT E
 - ERR(5) = ADD REQUESTED ALL

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 pane

- Lists all breakpoints set for this debug session.
- Work with individual breakpoints
 - Delete
 - Disable
 - Modify line breakpoints
 - Display Properties

Disabled line breakpoints are marked with a grey dot.





e-business

Running the program

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

```
IBM CODECOURSE/PAYROLLD
* 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:

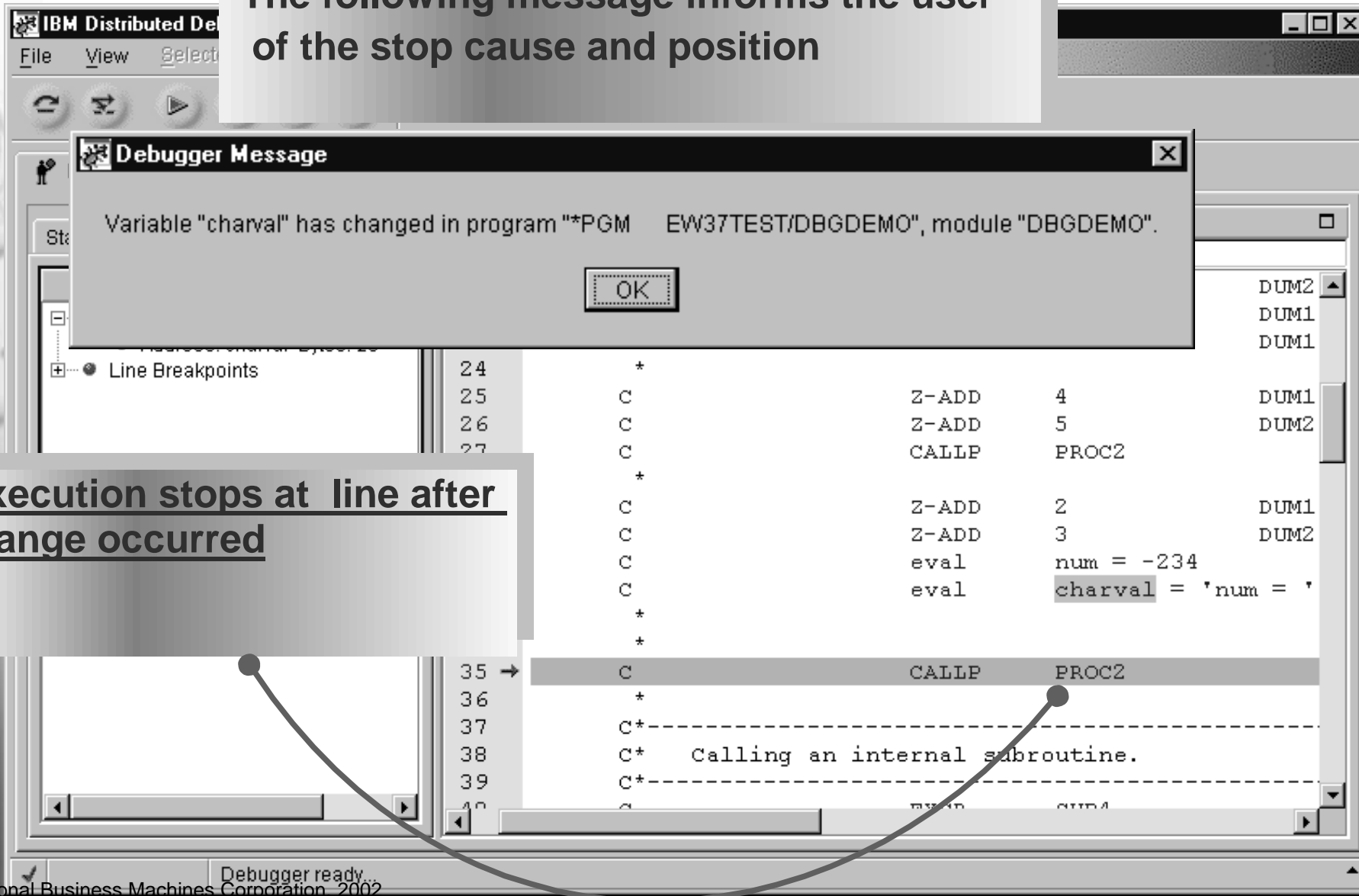
- next Line breakpoint
- Watch is triggered
- program termination
- you press Halt



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, a code window shows the following assembly code:

```
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          EVAL      SUB4
```

Line 35 is highlighted, and a curved arrow points from the 'Debugger Message' dialog to this line. The status bar at the bottom indicates "Debugger ready..."

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 a program named CODECOURSE/PAYROLLD. The code is as follows:

```
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. A mouse cursor is positioned over the 'Run to Location' option. The 'Breakpoints' panel on the left shows a watch breakpoint set for 'EMPAPL = X' at address EMPAPL. The 'Monitors' panel shows several expressions being monitored, including '*IN60 = 0', 'EMPAPL = X', 'PRJAPL = ''', and '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++)



Programs pane

- Lists all programs, modules and procedures of the current debug session.
- Add and remove programs, service programs, Java classes.
 - View list of modules for each program
 - View list of procedures for each module

Clicking on module or procedure name brings up its source.





e-business

Add Program

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' tree on the left lists the following structure:

- *PGM EW37TEST/DBGDEMO
 - DBGDEMO
 - DBGDEMO
 - PROC1

The 'Add Program' dialog box is open, with the following details:

- Program name: `codelabx/payrollg`
- Program type: Program, Service Program, Java Class

The background source code window shows the following code:

```
1 → H
2 *****
3 *           Debug demo program
4 *****
5 *-----*
```





Stacks pane

e-business

IBM Distributed Debugger

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

DBGDEMO:0

Stacks Breakpoints Programs

Stacks Control Panes

Thread 1 [0] (Runnable)

- ILECL Program:DBGDEMO Module:ILECL
- DBGDEMO Program:DBGDEMO Module:DBG**
- _QRNP_PEP_DBGDEMO Program:DBGDEMO
- InvokeTargetPgm(qp0z_pcp_cb*) Program:QP0ZPCPN
- Qp0zNewProcess Program:QP0ZPCPN Module:QP0ZPCPN
- main Program:QP0ZPCP2 Module:QP0ZPCP2
- _CXX_PEP Program:QP0ZPCP2 Module:QP0ZPCP2

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*	-----

Monitors Locals

Expression Monitors Value

NUM = -234

Series Call Stack:

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





e-business

Startup from the Remote Systems Explorer

The screenshot shows the 'Remote Systems Explorer - Development Studio Client' window. The left pane displays a tree view of 'Remote Systems' with folders for S400A and S400B. Under S400B, there is an 'iSeries Objects' folder containing sub-folders like 'Your libraries...', 'Your objects...', and 'Your members...'. A 'Library list' folder is expanded, showing objects like QSYS, QHLP SYS, QUSR SYS, IWEISS, and CODELAB01. The 'CODELAB01' folder is expanded to show program objects: CLR1.*pgm.cle, PAYROLLG.*pgm.rpg (selected), STRCODECPP.*pgm, EMPMST.*file.pf-dta, MSTDSP.*file.dspf, and SELECT. A context menu is open over the 'PAYROLLG.*pgm.rpg' object, with the 'Debug' option highlighted. The 'Properties' pane at the bottom shows details for the selected object: Name: PAYROLLG, Number of childr...: 0, Source: CODELAB01.

Property	Value
Name	PAYROLLG
Number of childr...	0
Source	CODELAB01

- ▶ Debug action available from the context menu of program objects and jobs.
- ▶ Launches the debugger's Attach dialog.
- ▶ Prefills dialog with hostname and name of selected program or job.

To start working with your iSeries objects, create a new system connecti

ing Tool
WebFacing Tool, you can take your classic applications to the
ay files and creates a browser based version of the user interfa

iSeries Comm...
ies Job Log Commands





e-business

Attach Dialog - Entry Fields

Attach

Compiled | Interpreted | AS/400

TCP/IP name of AS/400 host

Host: toras3rm Port: 3001

Job to debug

Job List...

Specify the name of the program to add to debug:

CODELAB/PAYROLLG Browse...

Use program profile

Step into

Debug Java

Debugger Settings...

Attach 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

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

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

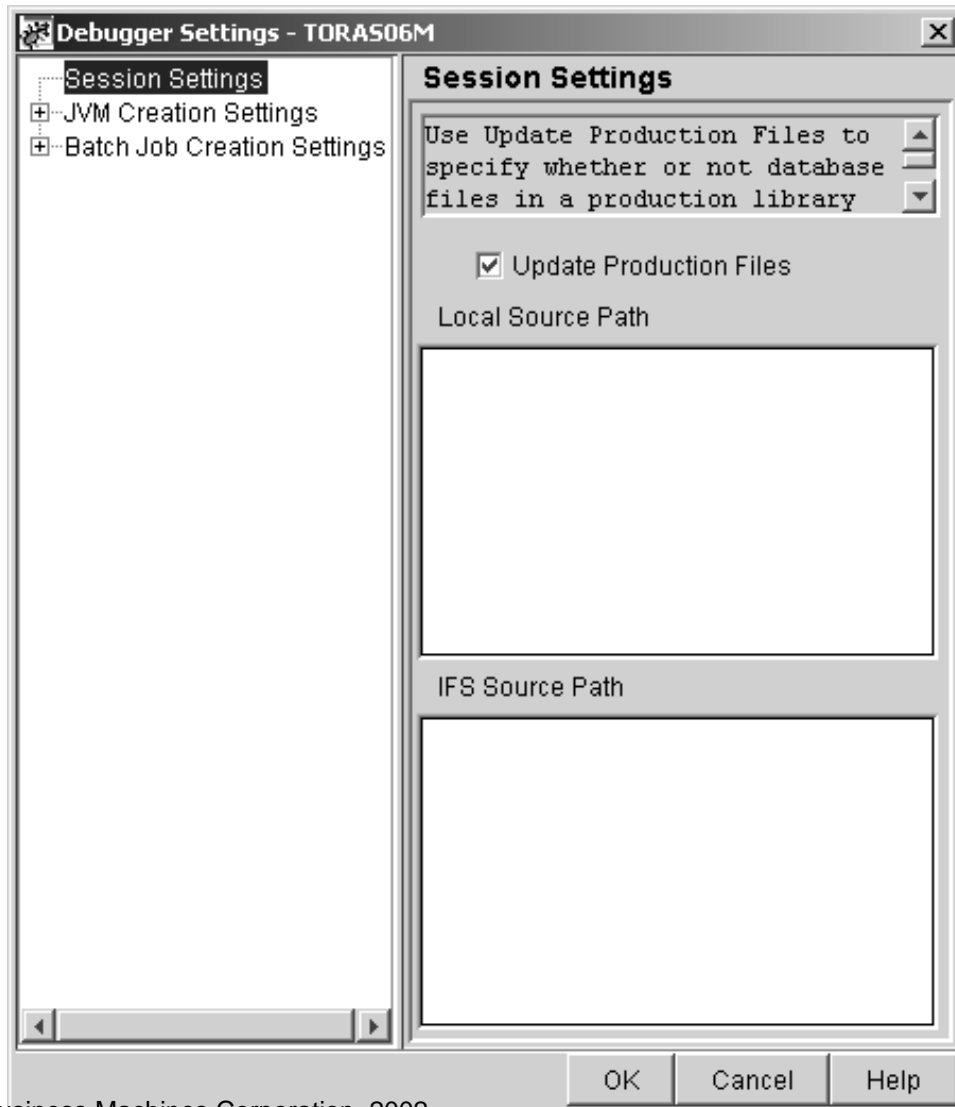




e-business

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

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

Debugging batch jobs

The screenshot shows the 'Attach' window on the left and the 'Session B' terminal window on the right. The 'Attach' window has fields for 'Host' (TORAS644), 'Job to debug' (/iweiss/036538), and 'Specify the name of the program to add to' (codelabxx/testbatch). It also has checkboxes for 'Use program profile', 'Step into', and 'Debug Java'. The 'Session B' window shows a menu of 'Major Command Groups' with '1. Select Command by Name' selected. Below the menu, the command `SBMJOB CMD(CALL PGM(CODELAB01/TESTBATCH)) JOBQ(CODELAB01/LABQ)` is entered. A text box on the right contains instructions for submitting the job and starting the debugger.

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

- `SBMJOB CMD(CALL PGM(CODELAB01/TESTBATCH)) HOLD(*YES)`
- Start debugger from the Start menu or command line or the CODE editor as Running application.
- Get job info of submitted job
- Start debug session
- Debugger will release the job and display the source when the program starts running.

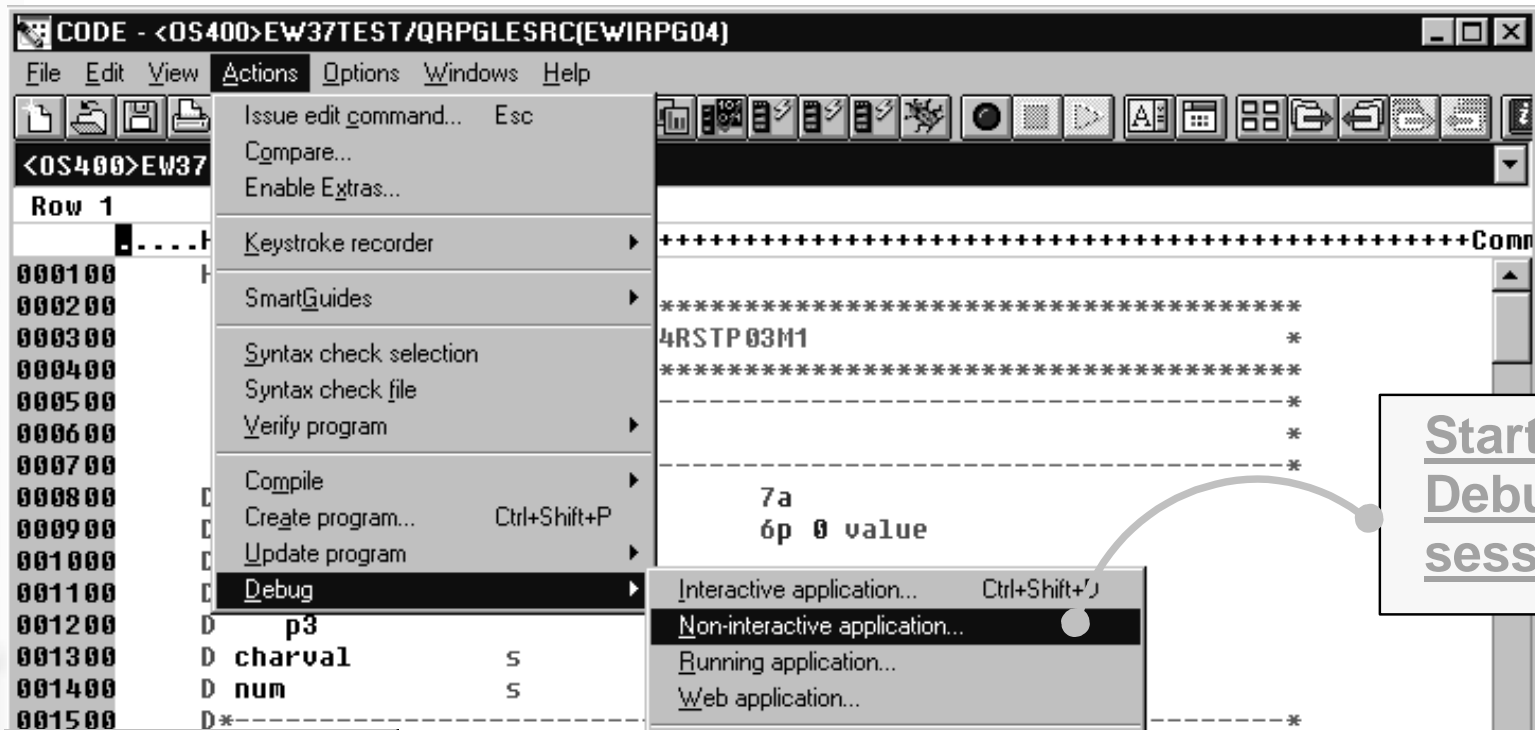
Note: If you start from the CODE editor, you will have to release the job yourself.



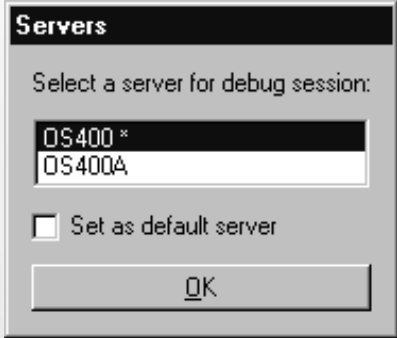


Start from Editor (Non-interactive application)

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

Brings up Call Program dialog (same as for interactive appl.)

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:

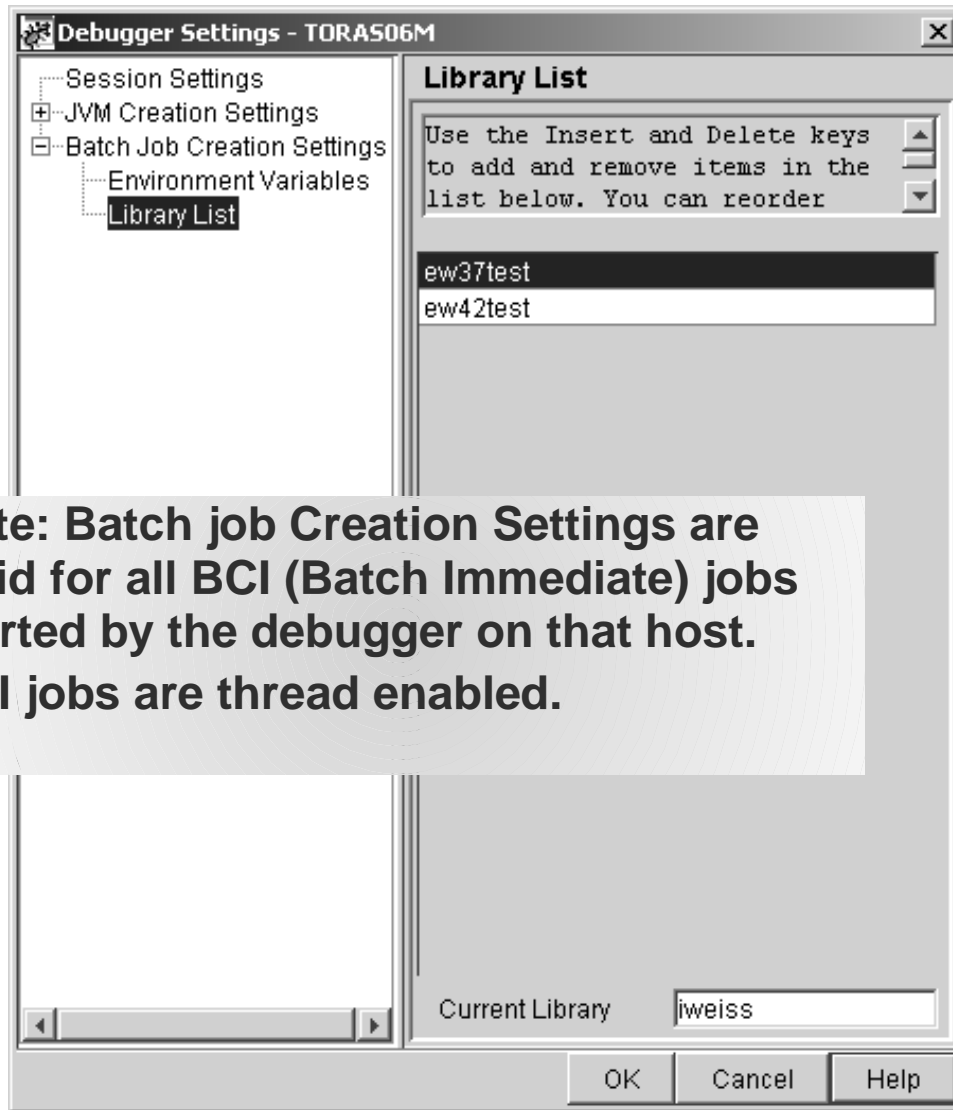
- additional debug sessions
- a new session after program termination





e-business

Debugger Settings - Batch Job Creation Settings



Note: Batch job Creation Settings 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 pane
 - 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

Starting from the CODE Editor

- Actions -> Debug ->
 - Interactive application... - starts the debugger for the program specified in the Call Program dialog and runs it in the selected STRCODE server job.
 - Non-interactive application... - starts the debugger for the program specified in the Call Program dialog and runs it in a BCI job in QSYSWRK.
 - Running application... - starts the debugger and displays the Attach dialog.
 - Web application... - starts Object Level Trace

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



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...

Load Cancel Help

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 'Stacks' window on the left lists threads, with Thread 1 at the top and Thread 2 below it. Thread 2 is marked with a grey triangle, indicating it is disabled. The 'Thread Properties' dialog box is open for Thread 3, showing its state as 'Suspended' and 'Frozen'. The 'Locals' window shows the current thread's local variables. The 'Monitors' window shows the current thread's active monitors. The 'Debugger ready...' status bar is visible at the bottom.

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

Thread 3: *CLASS /js
14 Str
15
16 // while loop with broken condition
17 → while(count <= 20){
18
19 temp++;
20
21 switch(count){
22 case 0:
23 case 5:
24 case 10:
25 count +=
26 break;
27 case 15:
28 count +=
29 break;
30 default:
31 // do not
32 }
33 }
34
35 // set count
36 count = 0;

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 shows the IBM Distributed Debugger interface. At the top, the title bar reads "IBM Distributed Debugger" with standard window controls. Below the title bar is a menu bar with "File", "View", "Selected", "Debug", "Source", "Stacks", "Locals", "Window", and "Help". A toolbar with various icons is located below the menu bar.

The main workspace is divided into several panes:

- Stacks Pane:** Located in the upper left, it shows a call stack for "Thread 3". The stack includes frames for "doSomething.jsbq", "run.jsbq/EWJSBQ02", "run.java/lang/Thread.startThread(void)", and "LE_Create_Thread".
- Source Pane:** Located in the upper right, it displays the source code for "Thread 3: *CLASS /jsbq/EWJSBQ02". The code includes a loop with a switch statement. Line 28 is highlighted, showing "count += 10;".
- Locals Pane:** Located in the lower left, it shows the local variables for "Thread 3". A "*LOCALVARS" section is expanded, showing "this = jsbq/EWJSBQ02:E670B6C", "count = 0", "temp = 0", and "sb" (with "serialVersionUID = 3388685").

A semi-transparent text box with the text "Entries for multiple threads in the Stacks and Locals page" is overlaid on the Stacks and Locals panes. Arrows point from this text box to the "Thread 3" entry in the Stacks pane and the "Thread 3" entry in the Locals pane.

```
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 interface with the following components:

- Breakpoints Window:** Shows a tree view of breakpoints. Under 'Line Breakpoints', there are two entries with source '*CLASS /jsbq/EWJSBQ02'. The second entry is selected.
- Properties Window:** Shows details for the selected 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
- Code Editor:** Shows a switch statement with a breakpoint at line 28:

```
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 default:  
32     /  
33 }  
34 }  
35 // s  
36 coun
```
- Monitors/Locals Window:** Shows the current state of variables:

```
Expression ...  
◆ a = 0  
◆ i = 1  
◆ jsbqThread  
  └─ jsbqThread[0]  
      activeThreadQ = null  
      threadInitNumber = 4  
      MIN_PRIORITY = 1  
      NORM_PRIORITY = 5
```

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



Debugger ready...



e-business

Java Startup

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

Classpath must be specified in Debugger settings.

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.

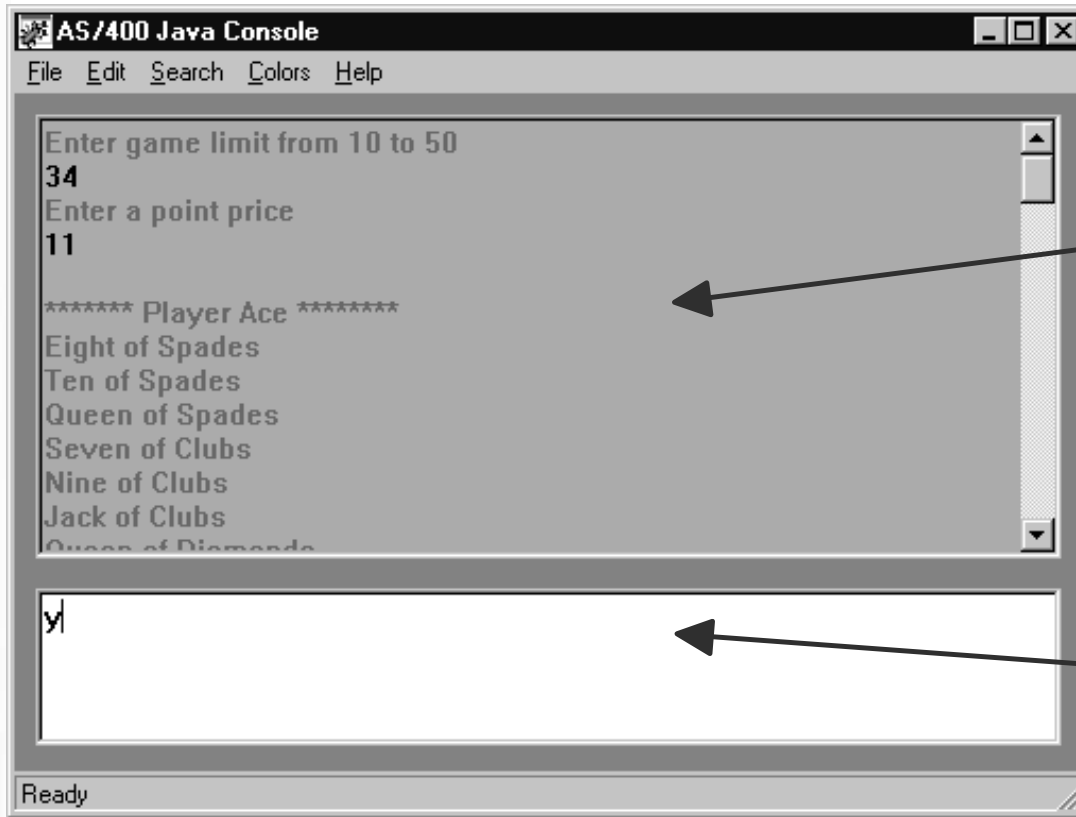
The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font with horizontal stripes.



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.





Debugging WebSphere Applications

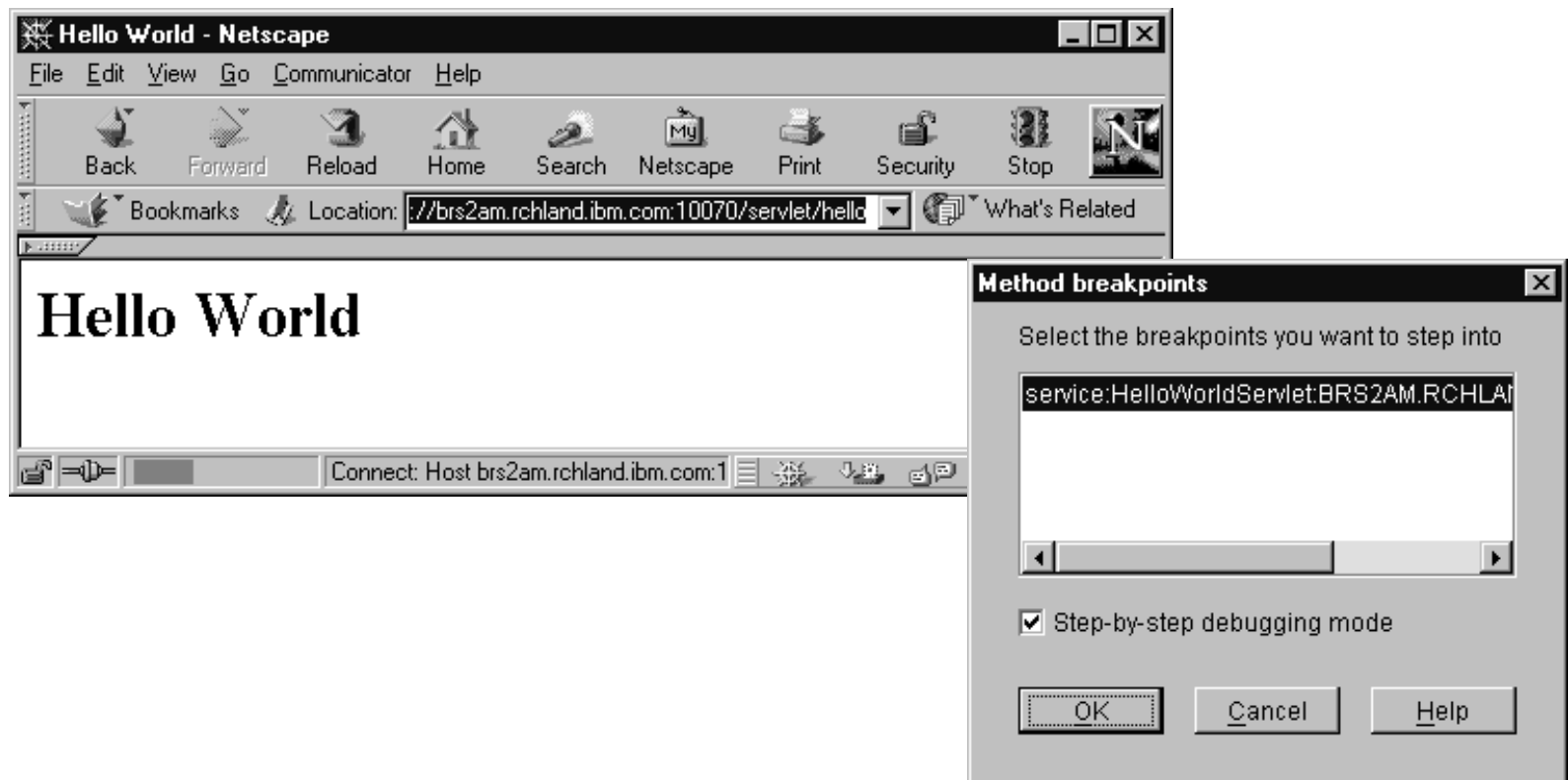


- Start Administrative Console
- Enable Debug and Object Level Trace



Debugging WebSphere Applications (cont.)

- Start OLT viewer
- Start the application server



- The debugger will be started for you



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.





WDT/400 Information

- **Additional Information**

- ▶ www.ibm.com/software/ad/iseries

- **Education**

- ▶ S6186 CODE/400 for iSeries -- Basic (2 days)
 - ▶ S6205 CODE/400 for iSeries -- Advanced (1 day)

 - ▶ S6286 iSeries Application Development using WDS*c* for iSeries -- Basic (2 days)

- **Newsgroup**

- ▶ news://news.software.ibm.com/ibm_software.code400



iSeries & WebSphere Resources & Deliverables

WebSphere Development Studio for iSeries
<http://www-3.ibm.com/software/ad/wds400/>

iSeries site
www.iseries.ibm.com/

iSeries 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/>

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

iSeries e-business Solutions
<http://www-1.ibm.com/servers/eserver/series/ebusiness/>

iSeries 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>

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

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

iSeries and WebSphere References
<http://www.as400.ibm.com/developer/java/solutions/jjem.html>
<http://www2.software.ibm.com/casestudies/swcsweb.nsf/platform>

iSeries 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/

WebSphere Payment Manager

www-4.ibm.com/software/webservers/paymgr/

MQSeries

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/addir/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>