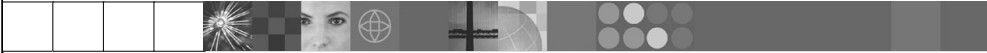




| IBM Software Group

WebSphere Development Studio Client for iSeries: The Integrated Debugger

WebSphere software

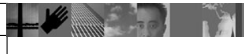


| WSC 6.0

© 2005 IBM Corporation

Table of contents

- ▣ **Overview**
 - Debugger Startup from the Remote System Explorer
 - NEW in V6.0: support for VPNs
 - Debug Perspective
 - Debugger Functions
 - Launch Configurations and Settings
 - Demo



This presentation first gives a high level overview of WDSC and where the Integrated Debugger fits in, as well as a look at the different ways to start the debugger and at its features.

IBM Software Group | WebSphere software

WebSphere Development Studio

Current 5722-WDS customers with software subscription for V5R3 and V5R2, to upgrade use feature #: 2656

Unlimited Licenses

RPG

COBOL

C/C++

**PDM
SEU
SDA,
RLU**

iSeries	iSeries	iSeries	iSeries	Web Facing	iSeries Projects	+CODE +VisualAge RPG	
Java™	Debug	Struts Web	Web Service		RSE		
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit

WebSphere Development Studio Client V6.0 www.ibm.com/software/awdtools/iseries


© 2005 IBM Corporation

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.

IBM Software Group | WebSphere software 

WebSphere Development Studio Client Advanced Edition V6.0

Workstation License
order through Passport Advantage
http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home

iSeries	iSeries	iSeries *	iSeries	Web Facing *	iSeries Projects	+CODE +VisualAge RPG	
Java	Debug	Struts Web	Web Service	RSE			
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit
	EGL * COBOL generation	EJB *	Test * Cases	Portal * Toolkit			

www.ibm.com/software/awdtools/iseries

© 2005 IBM Corporation

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.

Debuggers in WSDC

- **Integrated iSeries Debugger**
- Java iSeries Debugger
- WebSphere Application Server debug adapter
- IBM Distributed Debugger
Object Level Trace
- Compiled language debugger
- SQLJ debugger
- Stored procedure debugger
- Java Script debug adapter
- Active Script debugger

WSDC ships with 7 different debuggers, each one for a different user scenario.

Integrated iSeries Debugger – for all your host applications in RPG, Cobol, CL, C and C++

Java iSeries Debugger – for your Java development

WAS debug adapter – for EJBs, JSPs and servlets running on WAS

IBM Distributed Debugger and OLT – for Was 3.5, Java JNI calls and CODE users

Compiled language debugger – for workstation development

SQLJ debug adapter – for debugging Java embedded with SQL (not for iSeries)

Stored procedure debugger - for debugging stored procedures (not for iSeries)

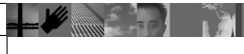
Java Script debug adapter – server side Java Script with WAS debug adapter

Active Script debugger



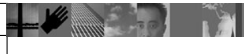
Integrated iSeries Debugger - Overview

- RPG, Cobol, CL, C, C++, and Java
- ILE and non ILE, incl. free-form RPG
- DB2 and SQL stored procedures
- Source and Listing view
- Batch and interactive
- Multi-Threaded Applications
- Client/Server Applications
- Distributed Applications



Debugger Invocation - Overview

- Remote Systems view in RSE and Project perspective
 - Pop up menu of a program or service program with source in library system or IFS
 - Pop up menu of a job
- iSeries Table view
 - Pop up menu of a program or service program
- Workbench
 - Tool bar: Debug pull down menu
 - Menu bar: Run menu



In a typical scenario, the user makes some changes to the source, runs a verify and compile and then debugs the program.

All these tasks are integrated into the RSE and can be invoked from menu items, tool buttons or pop up menus.



Table of contents

Overview

☐☐☐ **Debugger Startup from the Remote
Systems view**

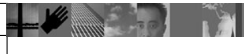
NEW in V6.0: support for VPNs

Debug Perspective

Debugger Functions

Launch Configurations and Settings

Demo



Debugger Invocation from Remote Systems view

Debug As... from Popup menu of:

Program
Service program
iSeries job

Debug (Prompt) from Popup menu of:

Program

Debug as

Batch
Interactive
Multi-threaded



You can run and debug programs from the Remote Systems view or the iSeries Table view in three ways:

- In a batch job
- In an interactive job
- In a server job

In the third case, running the program will use the same job as the Remote System Explorer communications server job. With batch and interactive jobs, you cannot monitor the status as easily, however, you do not tie up your communications server and you are notified when the program command ends. Batch jobs work as you would expect and do not require any initial setup.

Note: A multi-threaded debug session creates a new server job and this way keeps the RSE communications server job free for other tasks.

Where does the Application run?

Debug As...

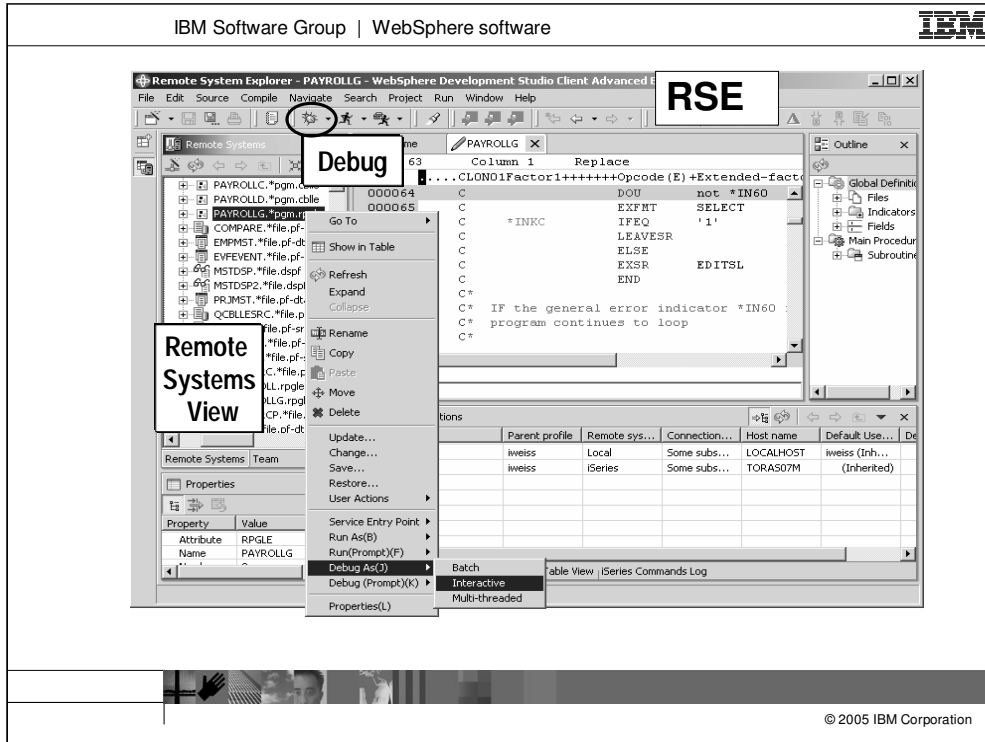
Batch	Submitted to batch
Interactive	5250 emulation, STRRSESVR
Multi-threaded	creates BCI job
Job	Specified job

Depending on the debugging mode you selected, the application will run in different jobs. Debugging in batch or interactive uses the same type of job as running the application.

Debugging Interactive requires a 5250 emulation session where the STRRSESVR command has been run.

For multi-threaded applications, instead of using the RSE server job, the debugger creates a BCI job and calls your application in that job.

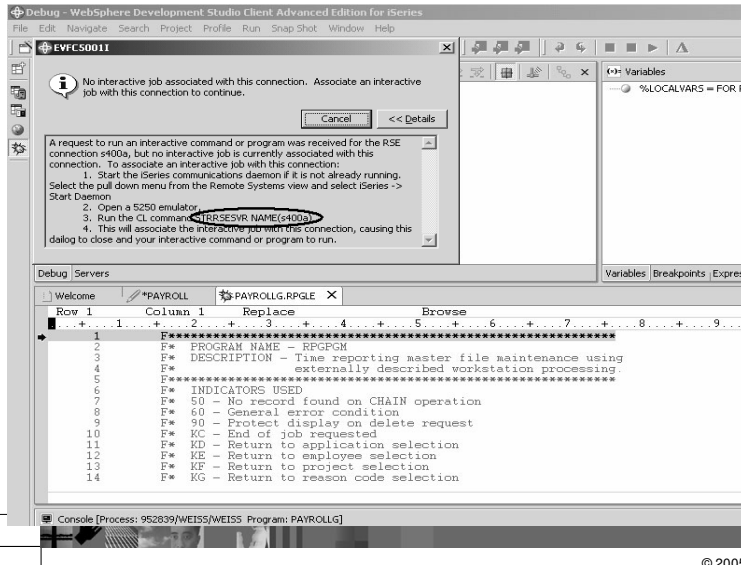
Selecting debug as job allows the user to attach to a job on the iSeries. This can be a batch, interactive, multi-threaded or even Java job.



When the debugger is invoked from the pop up menu, it will start for the selected program, step into it and terminate the debug session when the program ends.

For service programs, a dialog is displayed to collect information about the starting program.

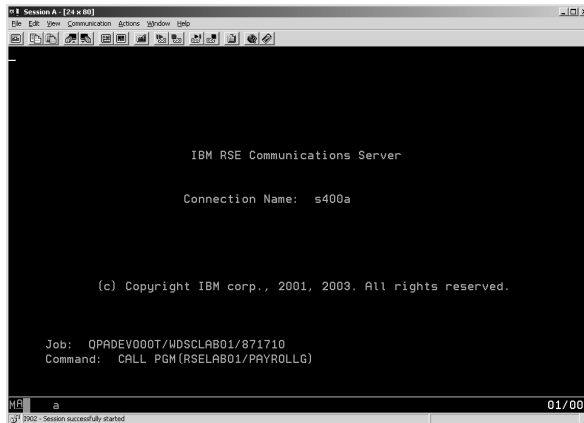
Debugging – Interactive



If you selected Debug as Interactive but there is currently no interactive RSE session, a message is displayed informing you that there is no such session and also giving you instructions how to remedy the situation.

Tip: You can copy the command from the message (STRRSESVR NAME(yourServerName)) and paste it into the 5250 command line.

Interactive RSE Session



```
IBM RSE Communications Server
Connection Name: s400a

(c) Copyright IBM corp., 2001, 2003. All rights reserved.

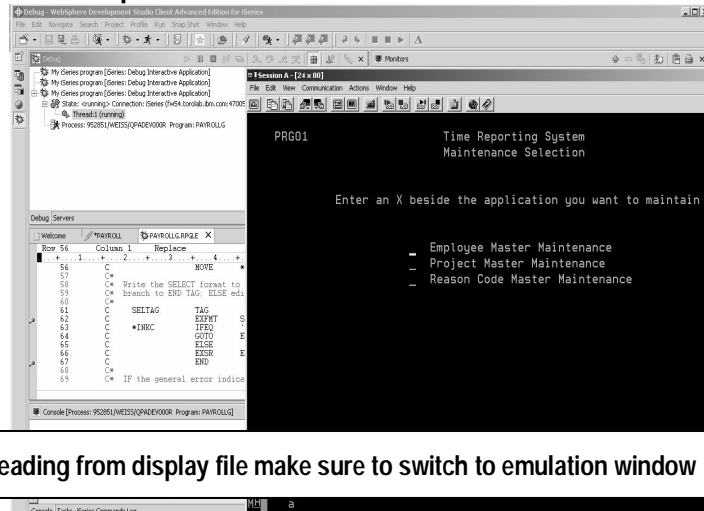
Job: QPADEV000T/WDSELAB01/871710
Command: CALL PGM (RSELAB01/PAYROLLG)
```

Message removed
when session established

Once the STRRSESVR command has been run, your 5250 session is associated with the RSE server and blocked from other use. The screen will look similar to the one above.

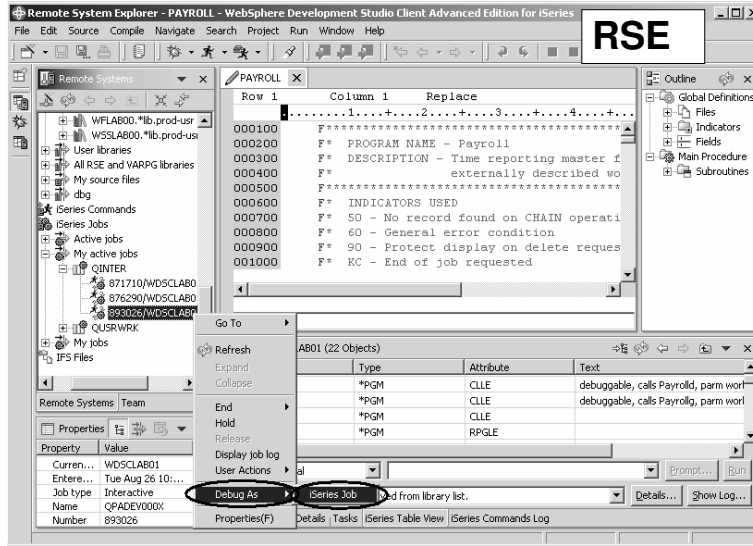
If you want control over the 5250 session back, use the pop up menu of one of the RSE subsystems (iSeries Objects, iSeries Jobs, iSeries Commands or IFS Files) and select 'Release Interactive job'.

Program output to emulation window



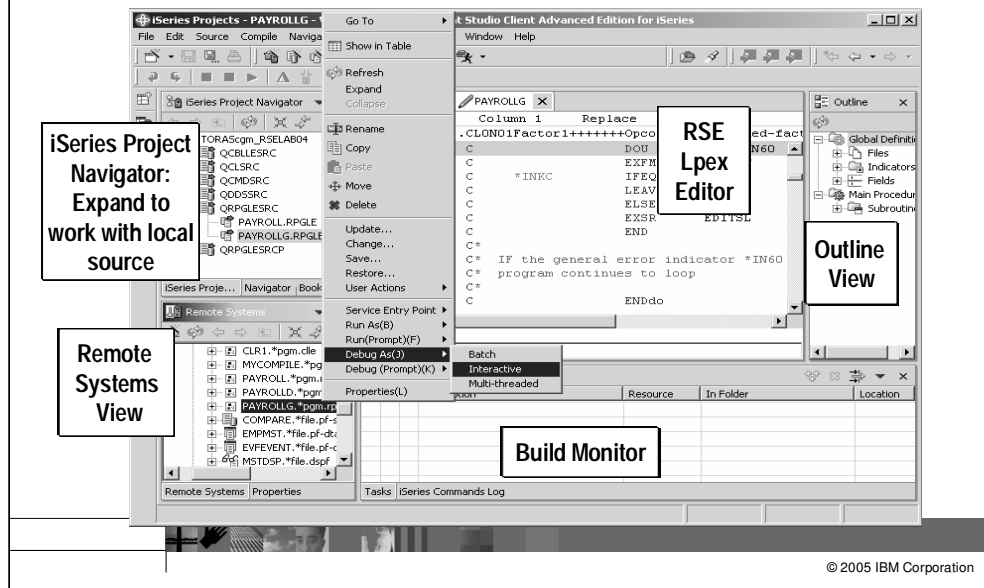
When reading from display file make sure to switch to emulation window

When you run an interactive program, the program waits for input from the 5250 emulation session.



You can launch the debugger from a pop up menu of a job.

iSeries Projects Perspective



This is the iSeries Projects perspective. The navigator view on the left is the primary view that drives the other views. It lists all the local files in the project. The LPEX editor is the same rich editor we saw for the Remote System Explorer.

When editing is complete and the project is pushed to the library and built, the build job is monitored in the job status window. When the build job is finished, you select the job and right click to see it's error list, which uses the same iSeries Error List window as the Remote System Explorer.

The debugger can be invoked from the pop up menu of the RSE view in the Project perspective.



Table of contents

Overview

Debugger Startup from the Remote Systems
view

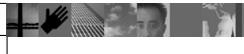
▣ **NEW in V6.0: support for VPNs**

Debug Perspective

Debugger Functions

Launch Configurations and Settings

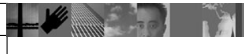
Demo





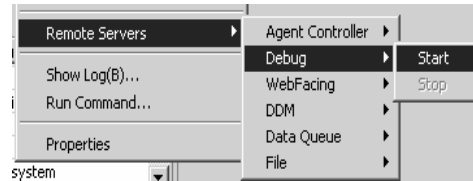
Support for VPN (Virtual Private Networks)

- Support various network settings, such as VPN, NAT and multiple client IP addresses
- New debugger router in the iSeries host
 - Hence, no debugger server call back required
 - But, need to start debug router on iSeries
- Still can start debugger in the 'usual' way
 - No need to start debug router on iSeries
 - Call back to the client is required



Start/Stop debug router on iSeries host

- From RSE, right click on one of iSeries Object, iSeries command or iSeries job subsystems. Select Remote Servers ->Debug->Start to start and Stop to stop.



- From 5250 session, issue command STRDBGSVR to start and ENDDBGSVR to stop debug router.

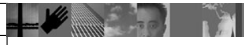




Table of contents

Overview

Debugger Startup from the Remote Systems
view

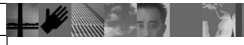
NEW in V6.0: support for VPNs

☐☐☐ **Debug Perspective**

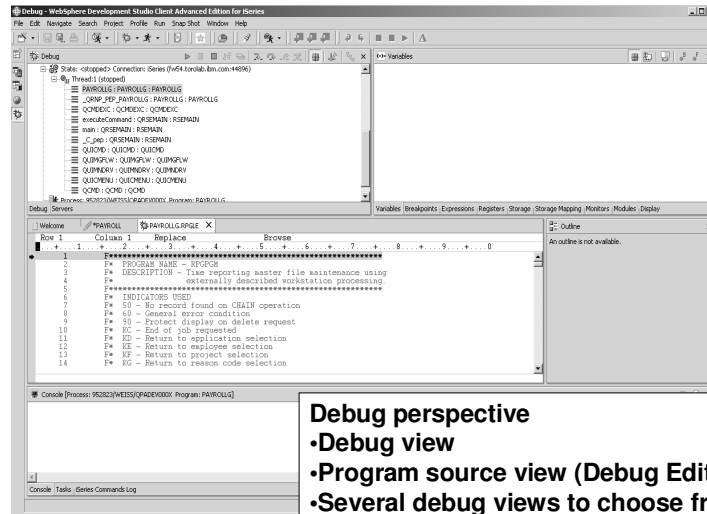
Debugger Functions

Launch Configurations and Settings

Demo



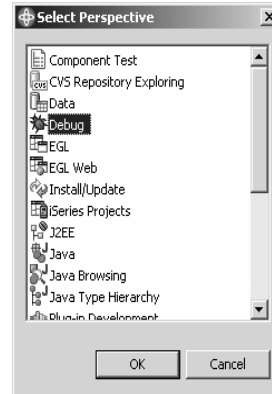
Debug perspective - Overview



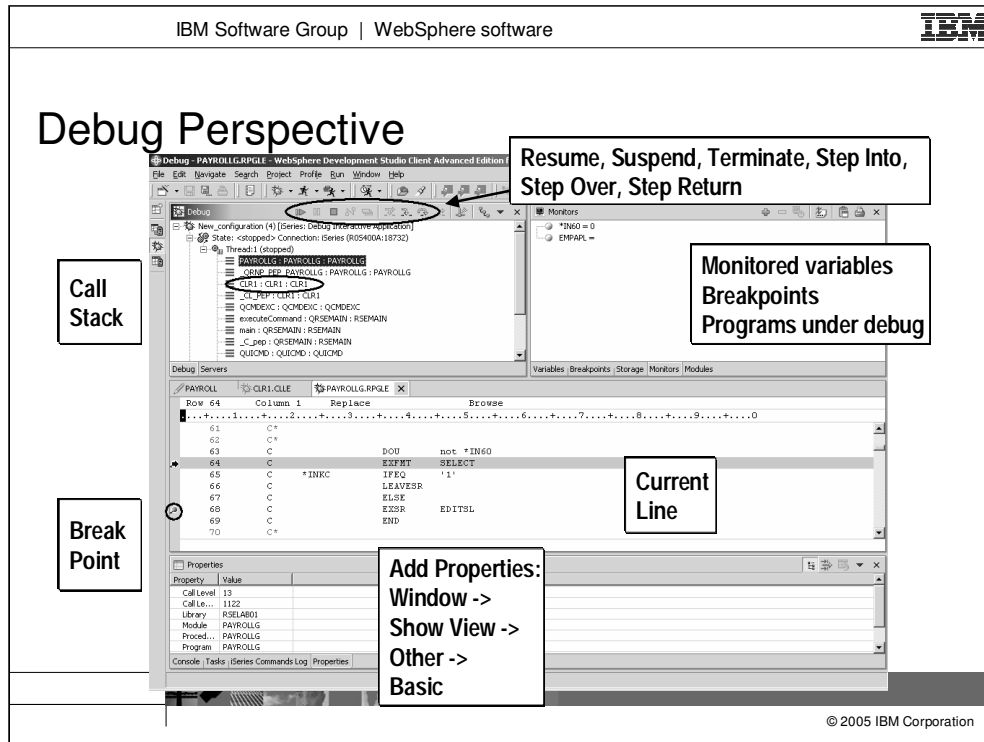
The Debug perspective contains the tools and views to debug a program. It opens when you start the debugger. Here you see the call stack and source view. There are several other views you can choose. For example, Breakpoints, Monitors, Storage.

Debug perspective

- Collection of editor and views
- Users can customize the perspective
- Views can be re-sized and re-positioned through drag and drop
- Views can be closed
- Views can be added
- Use Save Perspective As... to save customized perspective



The Debug perspective contains the tools and views to debug a program. It opens when you start the debugger. Here you see the call stack and source view. There are several other views you can choose. For example, Breakpoints, Monitors, Storage.



Here we see the common Eclipse Debug perspective, which is being used to debug an RPG program. The common debug user interface has been connected to the iSeries debug engine since Version 5.0, to offer a common and compelling debug story for OPM/ILE RPG and COBOL and CL, and ILE C and C++.

In the upper left pane is the call stack, much like option 11 in the OS/400's WRKACTJOB. It shows the calls that reflect your current program execution. When you double click an item in the stack, its source (if available) is shown in the source pane in the middle. The upper right is where all the various views are for working with data contents, breakpoints etc.. The middle is the debugger source view, with source executable (debug) lines in blue, others in green. The current line of execution is highlighted, and breakpoints appear as a dot with a check mark in the left margin.

Although not part of the common Debug Perspective, the Properties view contains valuable information about the selected object, which could be a breakpoint selected in the Breakpoints view, an entry selected in the call stack, etc. To add the Properties view, click on the menu item **Window** and **Show View** on the pull down menu, select **Other** from the submenu, expand **Basic**, select **Properties** and click OK.



Table of contents

Overview

Debugger Startup from the Remote Systems
view

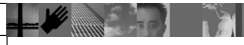
NEW in V6.0: support for VPNs

Debug Perspective

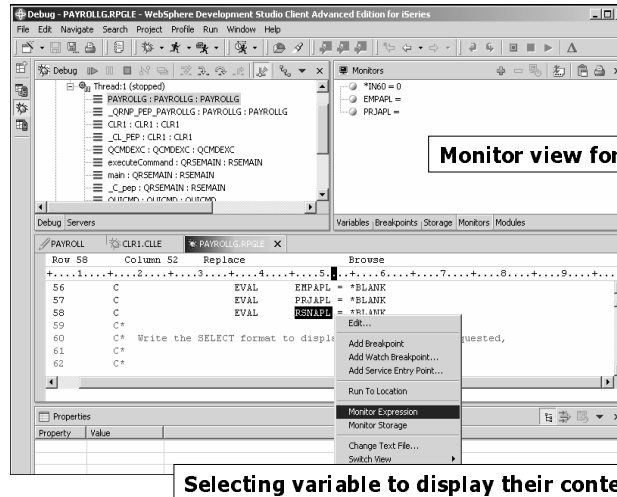
☐☐☐ **Debugger Functions**

Launch Configurations and Settings

Demo

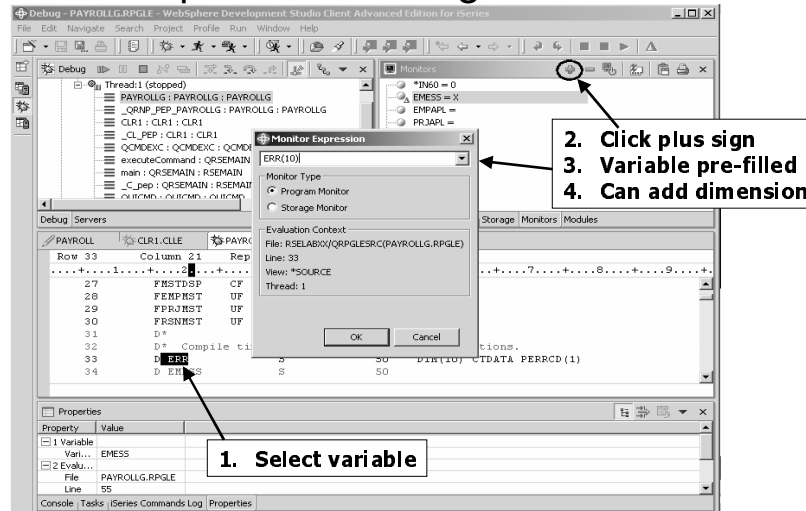


Display Variable Content



To monitor a variable, select it in the source by double clicking and use the Monitor Expression menu option.

Monitor Expression Dialog

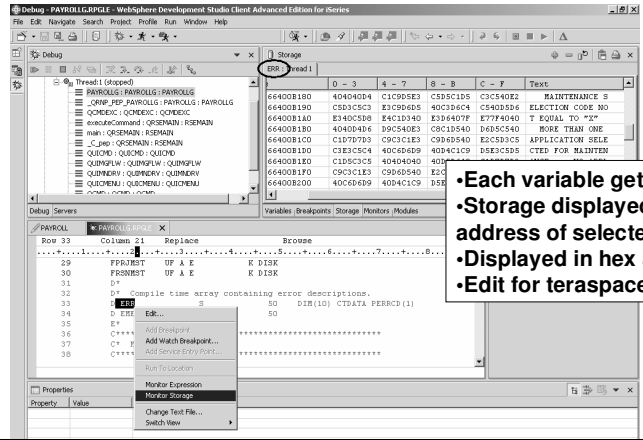


© 2005 IBM Corporation

You can also monitor a variable from the Monitor Expression dialog, which is available from the Monitors view. The dialog is pre-filled with any variable that is selected in the source. This dialog is especially useful when you want to monitor one specific array element or an element of a structure.



Display Storage Content



- Each variable gets a new tab
- Storage displayed starting with address of selected variable
- Displayed in hex and text format
- Edit for teraspace enabled pgms

Selecting variable to display the content in a storage monitor

To display the storage starting with the address of a selected variable use the Monitor Storage menu option.

Note: Teraspace enabled programs will allow you to modify the storage content.

C, C++ and Java Local Variables

The screenshot displays the IBM WebSphere Development Studio interface during a debug session. The main window shows the 'Debug - FARM.LISTING' window with the 'Variables' view open. The 'Variables' view lists local variables: 'a = 700', 'barnSize = 1800', and 't'. A context menu is open over the 'barnSize' variable, showing options: 'Select All', 'Copy Variables', 'Change representation' (with sub-options '1 Decimal' and '2 Hexadecimal'), 'Change Variable Value', and 'Set default representation'. A callout box with the text 'Change Value Change Representation' points to the 'Change representation' option. Below the 'Variables' view, the 'Debug Servers' window shows the 'FARM.LISTING' server with a 'Browse' view of the source code. The source code shows a C++ class definition for 'Farm' with a constructor and a 'show()' method. The 'Properties' window at the bottom shows the 'Property Value' table.

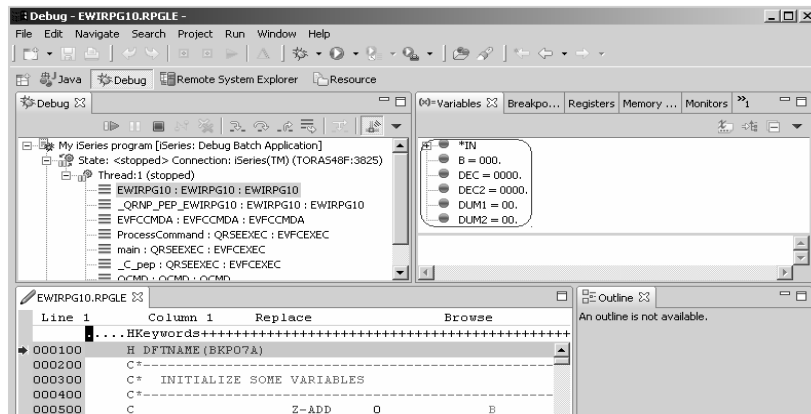
Property	Value
Property	Value

© 2005 IBM Corporation

The Variables view contains the local variables that are currently in scope.
Note: This feature is supported for C, C++ and Java.

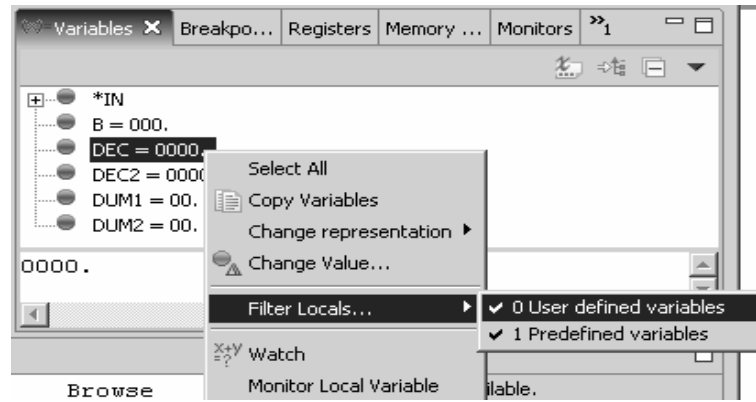
Local variables for ILE RPG and ILE COBOL

- In V5R3, local variables support is available for ILE PRG and ILE COBOL programs.
- Similarly to C and Java scenarios, Variables view is used



Filtering Local variables

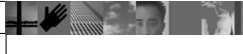
- Local variables can be filtered to show only user defined variables or pre-defined variables in ILE RPG or ILE COBOL.



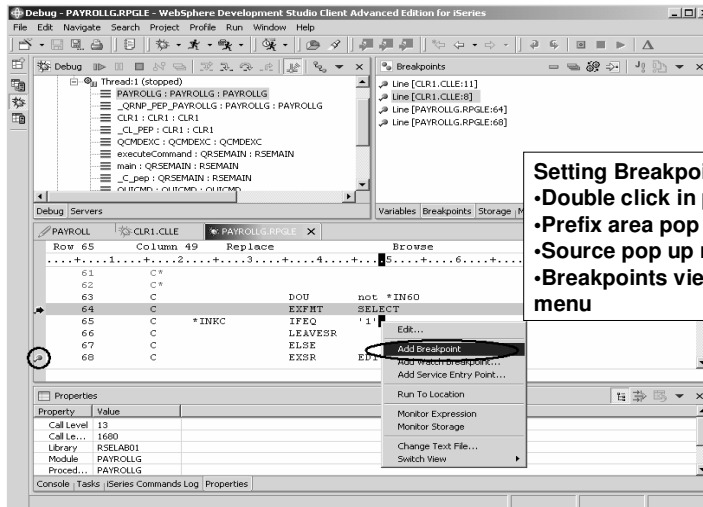


Program Execution

- **Step Into**
 - Debug the next call level
- **Step Over**
 - Run the next call level and stop at the next statement
- **Step Return (new in V5.1.2, requires V5R3)**
 - Run until you are back in the previous call level and stop at the next statement
- **Resume**
 - Run until an event is encountered
- **Run To Location**
 - Run and stop at the current cursor position
- **Suspend**
 - Halt program at point of execution
- **Terminate**
 - End the program



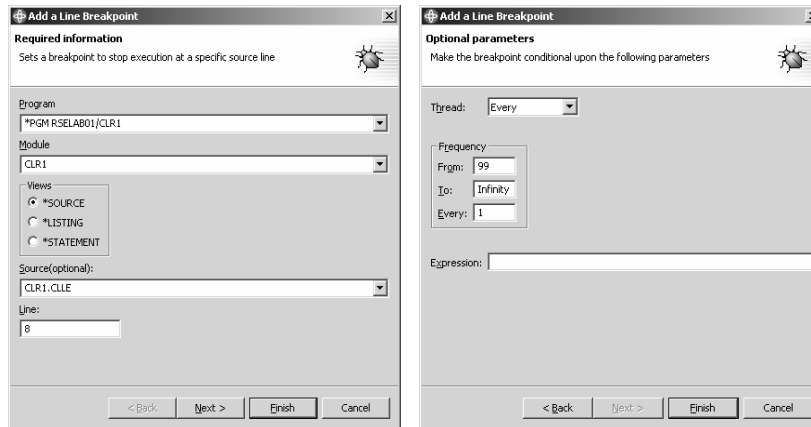
Setting Breakpoints



- Setting Breakpoints:**
- Double click in prefix area
 - Prefix area pop up menu
 - Source pop up menu
 - Breakpoints view pop up menu

You can only set breakpoints at executable lines. All executable lines are displayed in blue.

Conditional Breakpoints



The image displays two screenshots of the 'Add a Line Breakpoint' dialog box, illustrating the configuration of a conditional breakpoint.

Required information
Sets a breakpoint to stop execution at a specific source line

Program: *PGM RSELAB01/CLR1
Module: CLR1
Views:
 *SOURCE
 *LISTING
 *STATEMENT
Source(optional): CLR1.CLE
Line: 8

Optional parameters
Make the breakpoint conditional upon the following parameters

Thread: Every
Frequency:
From: 99
To: Infinity
Every: 1
Expression:

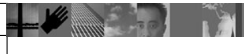
You can also set conditional breakpoints. The frequency allows you to limit the number of stops. Specifying an Expression will only stop program execution when the condition is true. The type of expression allowed depends on the programming language.



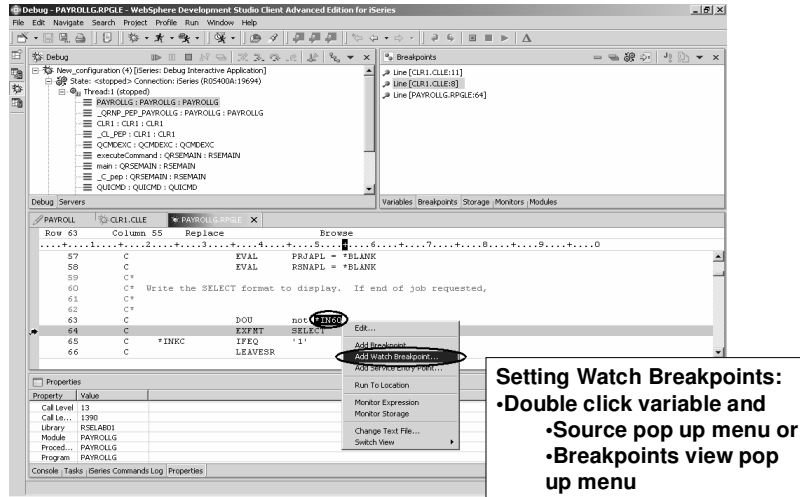
Source File breakpoint

- Use iSeries editor as the default editor for the debugger
- No prompt for program and module name when setting source breakpoint from iSeries editor
- Programs where source file breakpoints are set need to be in the program list of iSeries debugger launch configuration

NEW
In V6.0!



Setting Watch Breakpoints

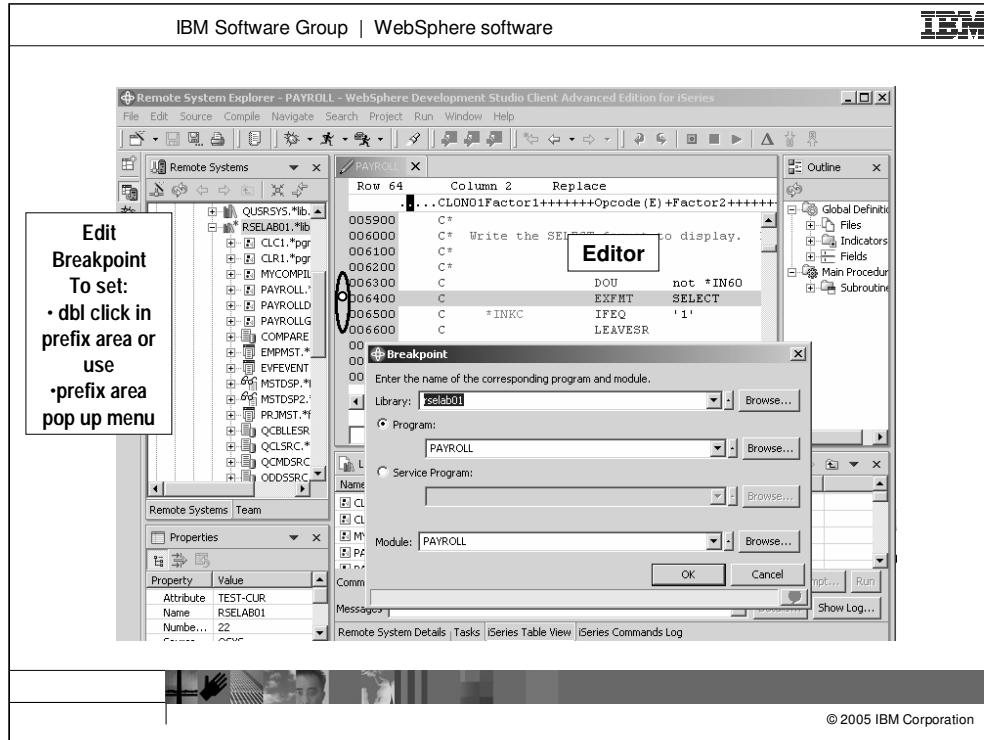


The screenshot shows the IBM WebSphere Development Studio Client interface. The main editor window displays a COBOL program with a context menu open over a variable. The 'Add Watch Breakpoint...' option is highlighted. A callout box on the right provides instructions: 'Setting Watch Breakpoints: •Double click variable and •Source pop up menu or •Breakpoints view pop up menu'.

© 2005 IBM Corporation

Watch breakpoints allow you to stop program execution when the value of the watched variable changes. The program stops at the line after the change occurred.

All breakpoints are listed in the Breakpoints view and can be manipulated from there, i.e. they can be disabled, enabled, removed and line breakpoints can be modified.



This is the dialog for edit breakpoints. Enter the library name and select Program or Service Program. The breakpoint is shown as a dot in the prefix area. If there is an active debug session, the breakpoint will be set in that session and there will be no marker in the editor. The breakpoint is then listed in the Breakpoints view in the Debug perspective.

Service Entry Points

- Service Entry Points for V5R2 and later
 - You know the program you want to debug but the program gets started in a server environment
 - You know the userid the program will be running under
 - You don't know the job name where the program will be running

You use service entry points when you wish to debug an application that makes use of the Toolbox or multiple jobs. Examples of cases where you would want to use a service entry point include:

- Applications that are invoked by a Toolbox program call. In this case, you would set a service entry point in the program that will be called by the Java application. When the program is called and the code where the service entry point is set is about to execute, the debugger can take control of the application and stop at that line. With this technique, you can put the program invoked by the Toolbox under debug when you do not know which job it will be running in.
- Programs that are spawned by other programs. In this case, you would set a service entry point in the application that will be spawned. When the program is spawned and the line where the service entry point is set is about to execute, operation will be suspended and the debugger will be able to gain control of the program and stop at that line. When a service entry point is set, it is triggered when the application not currently under debug is called.




Service Entry Points (continued)

- Great for:
 - WebFaced applicatons
 - Web applications
 - Toolbox calls
 - Any batch program you want to debug



You can debug all sorts of applications such as any batch program, a WebFaced application or a Web application.

Setting Service Entry Points

- From the Remote System Explorer (new in V5.1.2)
 - In the Remote Systems view, select the program or service program
 - In the pop-up menu, select Service Entry Point - > Set
 - Start the application
- From the Service Entry Points view (new in V5.1.2)
 - In the pop-up menu, select Set, or press 
 - Fill in the Set Service Entry Point window
 - Start the application
- From a Debug session
 - Debug the program or service program
 - Identify the line you want to stop at
 - Set the Service entry point
 - Start the application

In the Remote Systems view, you can set Service Entry Points for programs, service programs, modules or procedures.

If you set a Service Entry Point on a program, service program, or module, it will be set on all procedures in the selected object. A Service Entry Point set on a procedure, is valid for that specific procedure only.

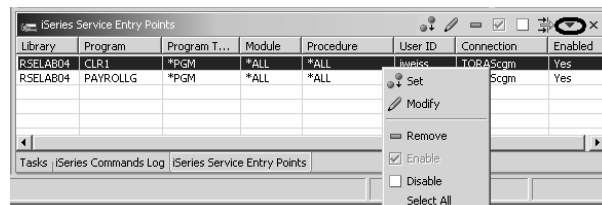
All Service Entry Points that are set from the RSE are listed in the Service Entry Points view.

Note: Whenever the program gets invoked from the specified userid, the program will be stopped at the line with the Service Entry breakpoint and the Integrated Debugger on your workstation will start a debug session for the specified program.

To set a service entry point from a Debug session, start a debug session for the program. Select **Add Service Entry Point** from the pop-up menu of the source or the prefix area for the line where you want your program to stop. This will invoke the **Add Service Entry Point** dialog box, which displays the program, module, source file, and line number of the service entry point that will be created. In this dialog box, specify the user profile for which the service entry point will be activated. By default, the user profile is set to *CURRENT (the user profile for the current debug session).

Service Entry Points view

- List of all Service Entry Points set from the RSE
- Pop-up menu to work with Service Entry Points
 - Set
 - Modify
 - Remove
 - Enable/Disable
- Filter window
- Pull down menu to Clear all or selected Service Entry points

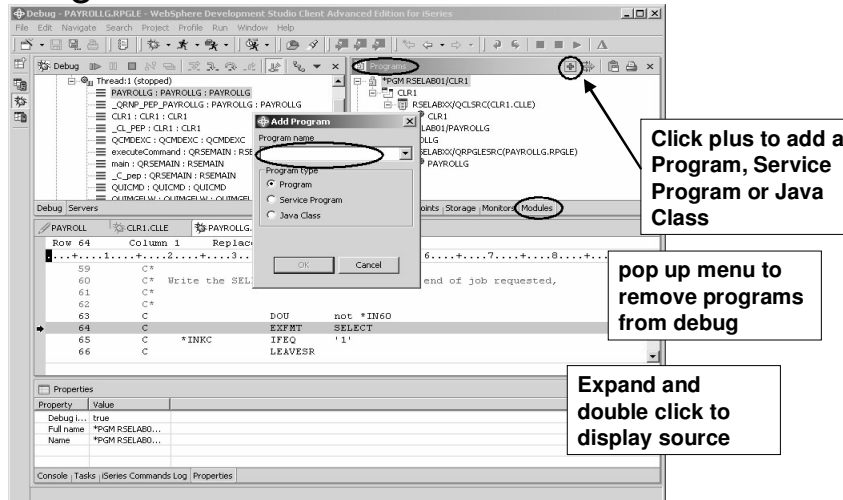


All Service Entry Points that are set from the RSE are listed in the Service Entry Points view.

A pop-up menu allows you to set new Service Entry Points, modify existing ones, remove or enable and disable them. Toolbar buttons for these actions are also available.

From the pull down menu, you can clear all Service Entry Points. This will remove them and also end the server program that gets started when you set a Service Entry Point. The server program will also end, when you exit the IDE.

Programs/Modules view



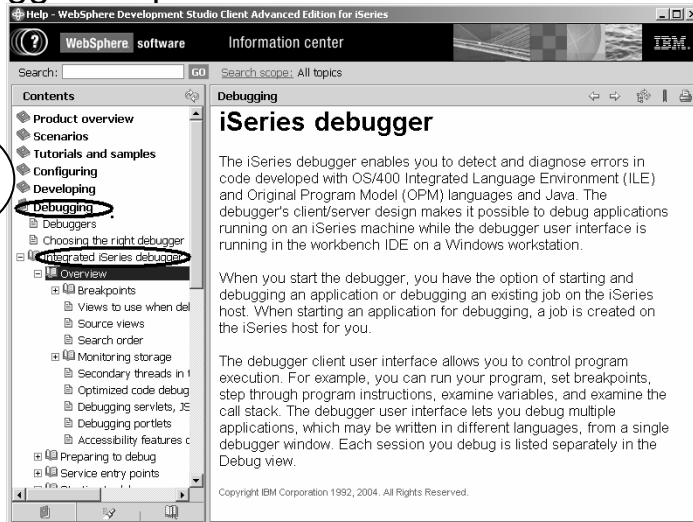
© 2005 IBM Corporation

The Programs view lists all programs, modules and procedures of the current debug session. You can use the Add Program dialog to add programs, service programs and Java classes to your debug session. Click the plus sign to bring up the dialog. The pop up menu of a program, service program or Java class allows you to remove the selected entry from the debug session. The initial program and the one you are currently stopped at cannot be removed.

Double clicking on a source or procedure entry displays its source.

iSeries debugger Help

Help ->
Help
Contents



F1 help is available from the different views.

Select **Help -> Help Contents** to look at the table of contents. The section **iSeries application development** contains the information about the Integrated Debugger.



Table of contents

Overview

Debugger Startup from the Remote Systems
view

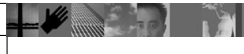
NEW in V6.0: support for VPNs

Debug Perspective

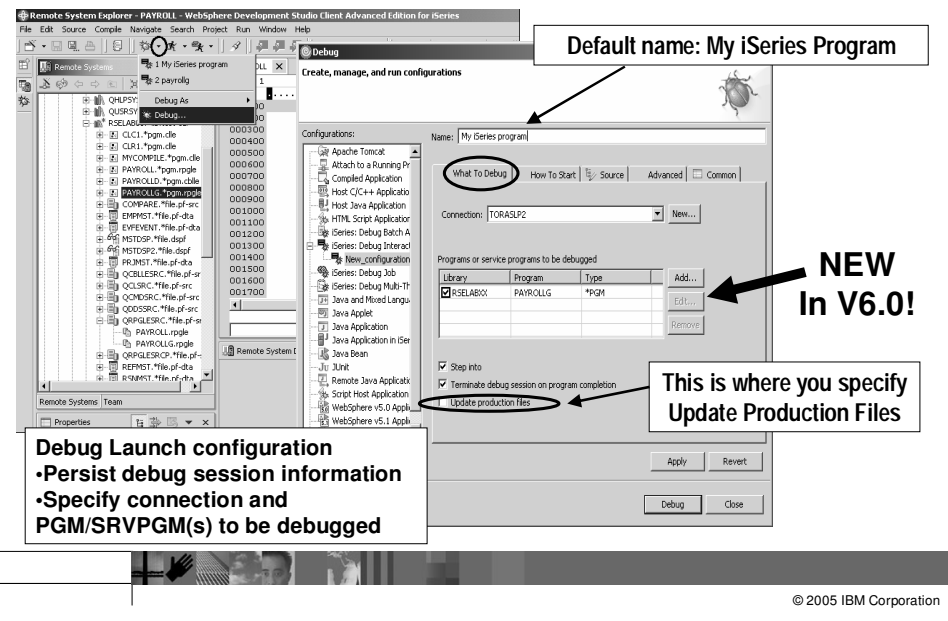
Debugger Functions

☐☐☐ **Launch Configurations and Settings**

Demo



Launch Configuration – what to debug



Default name: My iSeries program

NEW In V6.0!

This is where you specify Update Production Files

Debug Launch configuration

- Persist debug session information
- Specify connection and PGM/SRVPGM(s) to be debugged

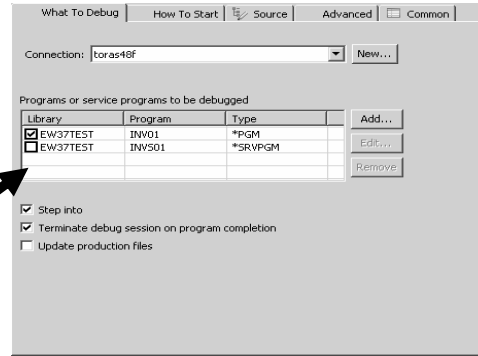
© 2005 IBM Corporation

You can start the Debugger in several ways: directly from the Remote Systems view using Debug As, or from the Launch Configuration dialog. Starting directly from the RSE without prompt does not allow you to specify parameters to be passed to the program. Prompting and using the Launch Configuration dialog allows you to modify how the program is invoked including to specify parameters.

When you invoke the debugger from the pop up menu, a default launch configuration called My iSeries program is created with the information of the selected object.

Add multiple PGM/SRVPGM via Launch Configuration

- User could specify multiple programs or service programs to be debugged in launch configuration.



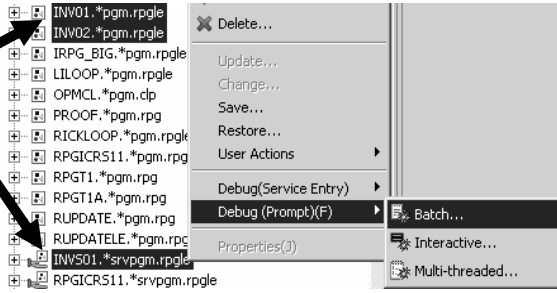
NEW
In V6.0!



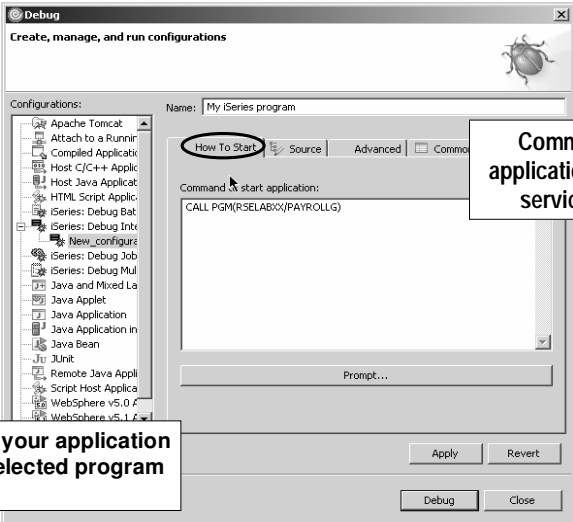
Add multiple PGM/SRVPGM via Remote Systems view

- User can select multiple programs or service programs from RSE to debug

**NEW
In V6.0!**



Launch Configuration – How to Start



**How to invoke your application
Default: Call selected program**

**Command to start application:
required for service programs**

CALL PGM(RSELAB(X)PAYROLLG)

Apply Revert

Debug Close

© 2005 IBM Corporation

In the How To Start page of the Launch Configuration you specify the command that invokes your application. By default, this page contains a call command for the selected program. You can modify this command to add parameters or you could specify a different command or program that invokes your application. You can use the Prompt button to get a prompt dialog for the specified command.

Launch Configuration – source location

- If source remains in the original library (compilation time), debugger will find it
- If source member has been moved since compile occurred, use iSeries source file source locator (next slide)
 - ✓ Moved to a different library
 - ✓ Resides on a different iSeries host
- If IFS source is moved, use Remote Folder source locator.
 - Moved to a different folder
 - Resides on a different iSeries
- Source locator can be specified from Source tab of launch configuration



In the How To Start page of the Launch Configuration you specify the command that invokes your application. By default, this page contains a call command for the selected program. You can modify this command to add parameters or you could specify a different command or program that invokes your application. You can use the Prompt button to get a prompt dialog for the specified command.



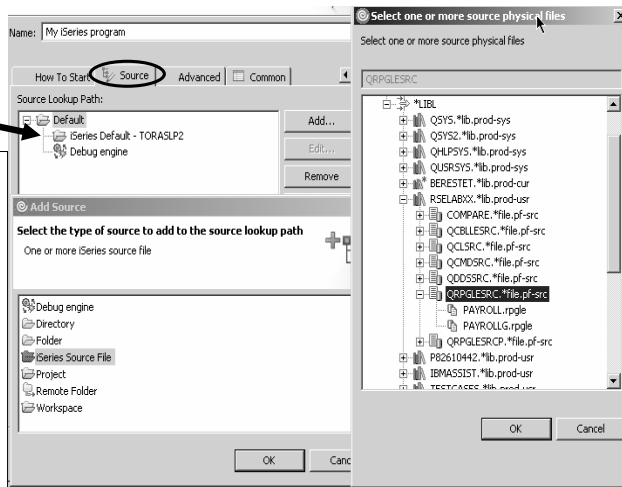
Launch Configuration – source location tab

Selected source physical file is added to the list of source lookup paths.

Click Add button in the Source tab to invoke Add Source dialog.

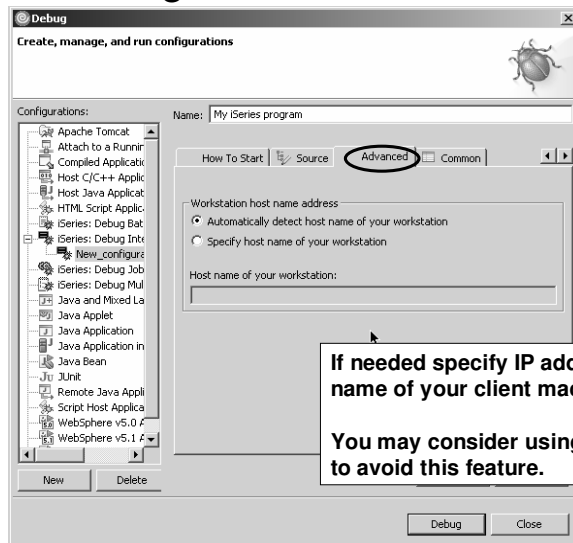
Choose iSeries Source File to invoke Select Source Physical File dialog

Choose iSeries Connection of interest and select your source physical file.



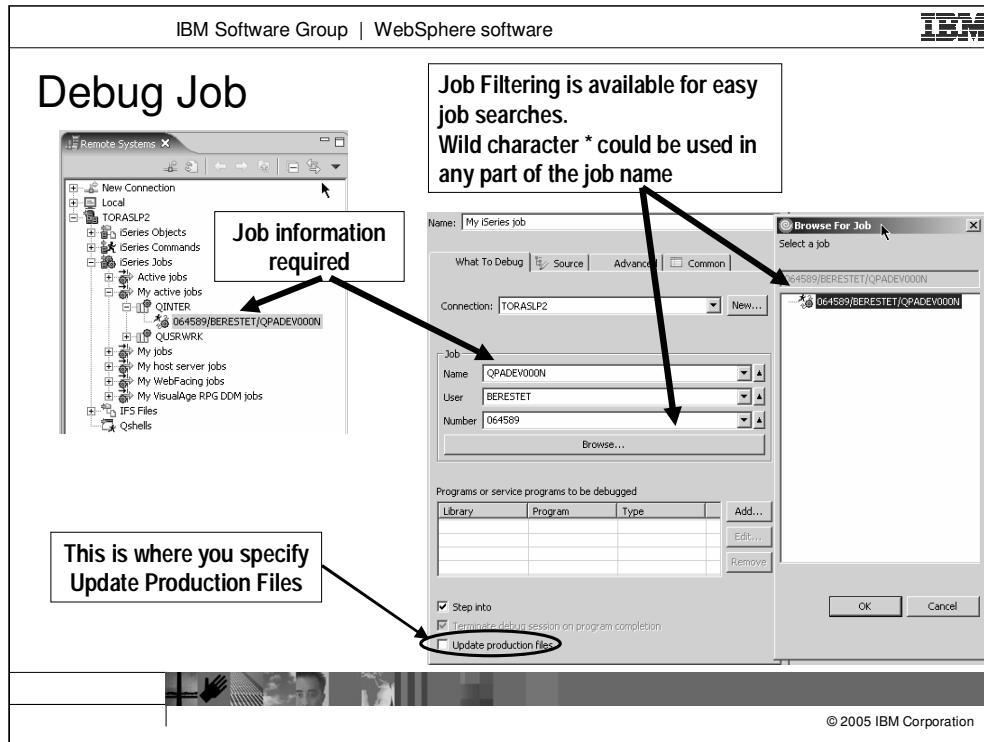
Source Location allows users add new iSeries source physical files and/or IFS folders on any configured iSeries host to search for the source members during debug.

Launch Configuration – Advanced Page



© 2005 IBM Corporation

Normally the debugger is able to resolve the address or hostname of your workstation. However, if your workstation has more than one IP address, for example when you are connecting to the iSeries via VPN,



Selecting Debug Job from the pop up menu will create a launch configuration with the name My iSeries job. If you selected a job in the RSE, this job will be put into debug mode and when prompted, you start your application in that job. Step into is selected which means that the debugger stops at the first executable statement it encounters. Terminate debug session on program completions is not available in this case since the debugger was started for a job, not a program.

If you want to debug a program in a specific job, you have to use the iSeries: Debug Job Launch Configuration and specify job and program.

Note: There is no 'How To Start' page in the job launch configuration. It is up to the user to start the application in the specified job when prompted to do so.

Library List

Initial library list from user profile

- **Change for this session from RSE pop up menu of Library List**

- Add Library List Entry

- Change Current Library

- **Change in Properties from RSE pop up menu of any subsystem**

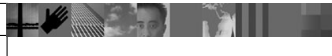
- Initial Library List node**

- Add Library List Entry

- Change Current Library

- Re-order library list entries

- **Interactive RSE session uses library list of interactive job + libraries set in Properties**

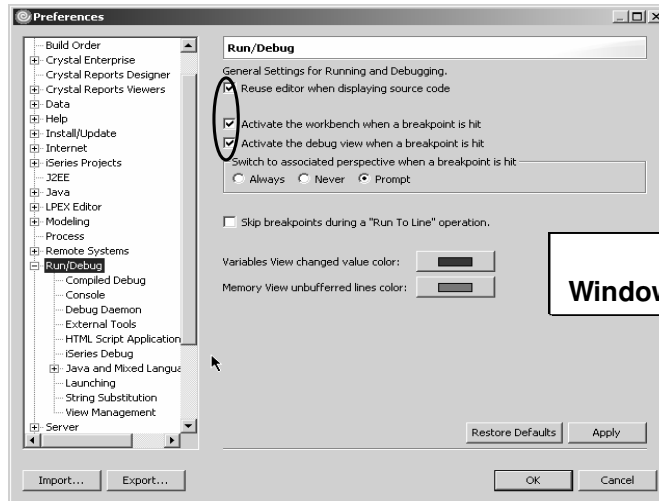


Changes to the library list affect the RSE server job as well as jobs submitted to Batch and BCI jobs created to debug multi-threaded applications.

The interactive RSE session uses the library list that is set for the 5250 session before the STRRSESVR command is run. Added to that list are the libraries that are set in the Properties. Changes to the properties settings will be used after the connection has been disconnected and connected again.

Tip: You can create multiple connections to the same iSeries host and set different properties for each one.

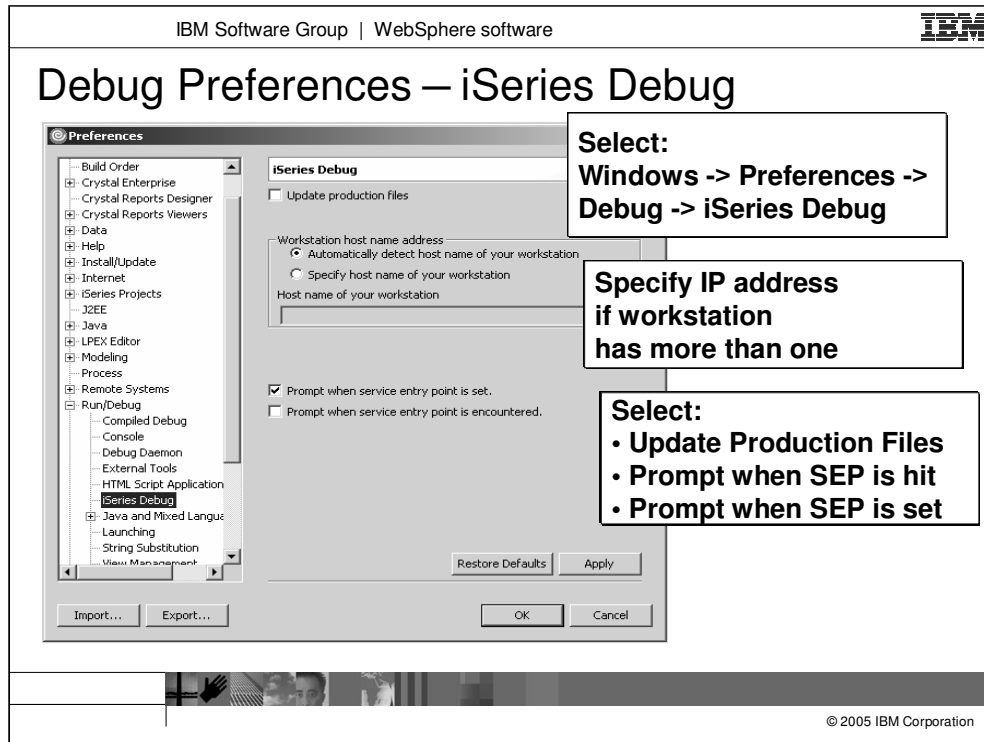
Debug Preferences



Select:
Windows -> Preferences

Check the 'Reuse editor when debugging source code' check box if you want only one tab for multiple source in the debug editor. Switching to a different source can then be done from the call stack or the Programs view. If de-selected, each source will have its own tab and selecting its tab will display the source.

Check the 'Remove terminated launches when a new launch is created' check box, to delete the messages in the call stack that belong to terminated debug sessions. That way only the currently active sessions are listed in the call stack.

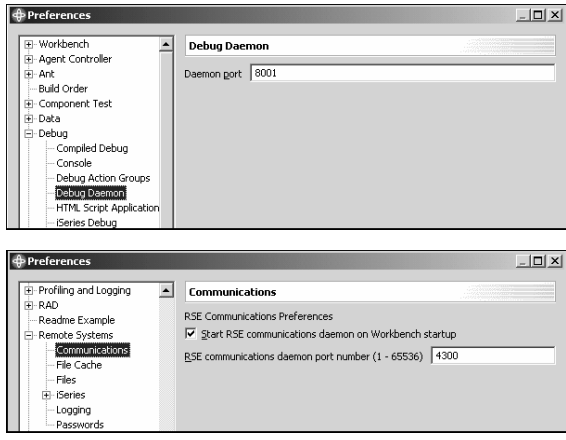


By default, Update production files is not selected. Check the check box to allow the debugger to update production files.

If your workstation has more than one IP address, the debugger may not be able to resolve the workstation's host name or IP address. In this case, select Specify host name of your workstation and enter either the host name or IP address in the entry field.

Selecting Prompt when service entry point is encountered will display a dialog to allow you the choice of starting a debug session or not.

Preferences – Port Numbers



The image shows two screenshots of the IBM WebSphere Preferences dialog. The top screenshot shows the 'Debug Daemon' section with the 'Daemon port' set to 8001. The bottom screenshot shows the 'Communications' section with the 'Start RSE communications daemon on Workbench startup' checkbox checked and the 'RSE communications daemon port number (1 - 65536)' set to 4300. To the right of each screenshot is a text box with instructions.

**Select:
Windows -> Preferences**

**Debug Daemon
port number**

**Remote Systems
Communications
RSE port number**

© 2005 IBM Corporation

The port number specified in Remote Systems -> Communications is used by the RSE communications daemon which includes the debugger communication. If you are working from behind a firewall, opening the specified port is required to give you access to the debugger. In this case, you also have to open up the debug daemon port which can be specified in **Preferences -> Debug -> Debug Daemon**.



Table of contents

Overview

Debugger Startup from the Remote Systems
view

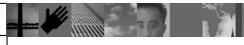
NEW in V6.0: support for VPNs

Debug Perspective

Debugger Functions

Launch Configurations and Settings

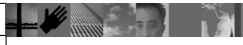
 **Demo**





Summary

- Source level debugger
- Fully integrated
- Common user interface
- Improved developer productivity
- Off-load host
- Service Entry Point to debug Web Apps!





Additional Information

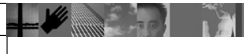
- **homepage:**
<http://ibm.com/software/awdtools/iseries>
Select Library link for Labs, Tutorials, Presentations
- **newsgroup:**
<news://news.software.ibm.com/ibm.software.websphere.code400>





Additional Information – WDSCI Mailing List

- **To post a message email: WDSCI-L@midrange.com**
- **To subscribe, unsubscribe, or change list options:
<http://lists.midrange.com/mailman/listinfo/wdsci-l>
or email: WDSCI-L-request@midrange.com**
- **Before posting, review the archives:
<http://archive.midrange.com/wdsci-l>**





Trademarks & Disclaimers

© IBM Corporation 1994-2004. All rights reserved.
 References in this document to IBM products or services do not imply that IBM intends to make them available in every country.
 The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.
 Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
 Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
 Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
 ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.
 UNIX is a registered trademark of The Open Group in the United States and other countries.
 SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
 Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

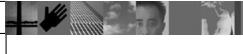
Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.





Disclaimer

- **Acknowledgement:**

This presentation is a collaborative effort of the IBM Toronto iSeries Application Development team, including work done by:

Vadim Berestetsky, Phil Coulthard, George Farr, Claus Weiss, Don Yantzi.

- **Disclaimer:**

The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

- **Reproduction:**

The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

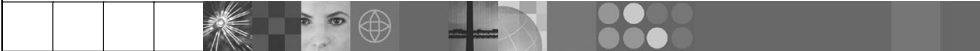




IBM Software Group

WebSphere Development Studio Client for iSeries: The Integrated Debugger

WebSphere software



| WSC 6.0

© 2005 IBM Corporation