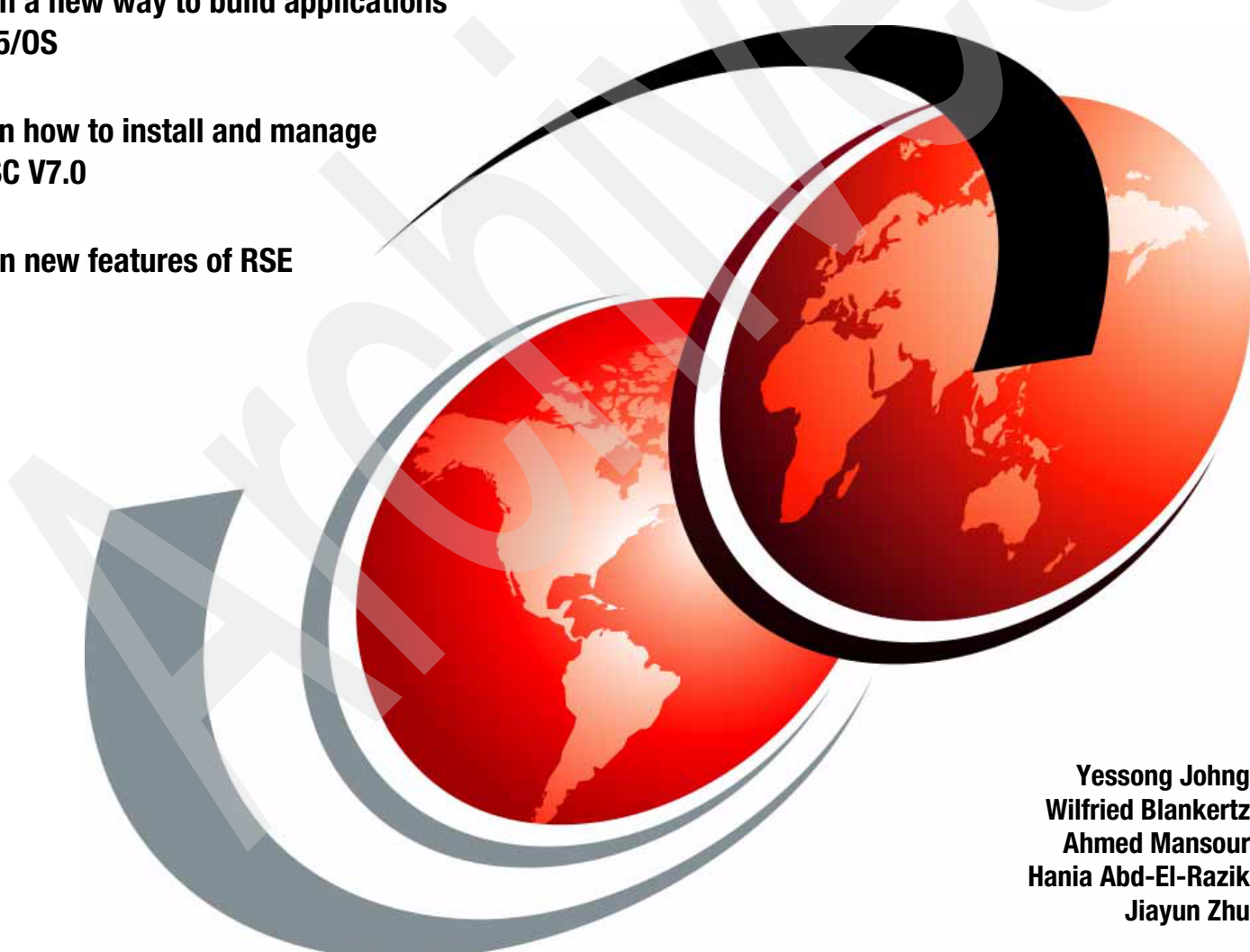


Mastering WDSC V7.0 and WDSC V7.0 Advanced Edition for i5/OS V5R4

Learn a new way to build applications
for i5/OS

Learn how to install and manage
WDSC V7.0

Learn new features of RSE



Yessong Johng
Wilfried Blankertz
Ahmed Mansour
Hania Abd-El-Razik
Jiayun Zhu



International Technical Support Organization

**Mastering WDSC V7.0 and WDSC V7.0 Advanced
Edition for i5/OS V5R4**

June 2008

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

Archived

First Edition (June 2008)

This edition applies to Version 7.0 of WebSphere Development Studio Client for System i and WebSphere Development Studio Client Advanced Edition for System i.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this book	xii
Become a published author	xiii
Comments welcome	xiii
Part 1. Introduction to WDSC V7.0	1
Chapter 1. A new set of application creation tools for System i	3
1.1 IBM WebSphere Development Studio for System i	4
1.1.1 ILE RPG	5
1.1.2 ILE C/C++	5
1.1.3 ILE COBOL	5
1.1.4 Application Development ToolSet	5
1.2 WebSphere Development Studio Client for iSeries	6
1.2.1 Summary of enhancements from V5.1.2 to V7.0	7
1.2.2 IBM WebFacing Tool	10
1.2.3 Web tools: System i extensions	11
1.2.4 Java development tools	11
1.2.5 CoOperative Development Environment	12
1.2.6 VisualAge RPG	13
Chapter 2. Eclipse, RSDP, and RAD	15
2.1 Building blocks of WDSC	16
2.1.1 Eclipse: Open source integrated development environment	17
2.1.2 Rational Software Delivery Platform	18
2.1.3 Rational Application Developer	19
2.2 Components of the Eclipse development environment	19
2.2.1 Workbench	20
2.2.2 Projects	20
2.2.3 Perspectives	21
2.2.4 Views	23
2.2.5 Editors	25
2.2.6 Crash recovery	28
2.3 Development Studio Client features	29
2.3.1 Tools	29
2.3.2 Common features	30
2.4 Online help	37
Chapter 3. WebFacing, HATS, and WDHT	43
3.1 Why Web-enable an application	44
3.1.1 What is blocking Web enablement of green screens	44
3.1.2 Web-enabling tools in WDSC	44
3.2 Web enablement using WebFacing	46
3.2.1 Introduction to WebFacing	46
3.2.2 WebFacing features and benefits	48
3.3 Web enablement using HATS	56

3.3.1	Introduction to HATS	56
3.3.2	HATS features and benefits	57
3.4	WebFacing and HATS interoperability	62
3.5	WDHT	62
3.6	V7.0 enhancements and changes summary	65
Chapter 4. Bringing Your i5/OS applications to the Web		67
4.1	Tools for Web development	68
4.2	Web development with JSF	69
4.2.1	JSF-based architecture - the new way	70
4.3	Enabling SOA through Web Services	71
4.3.1	Using a Web Service	71
Part 2. Installing and managing WDSC		73
Chapter 5. Installing WDSC		75
5.1	Overview of the installation process	76
5.1.1	WDSC installation wizard	76
5.1.2	Introduction to Installation Manager	77
5.2	Planning for installation	77
5.2.1	WDSC product prerequisites	77
5.2.2	Installation planning checklist	81
5.2.3	Migration from previous WDSC versions	86
5.3	Installation of WDSC	87
5.3.1	Starting the installation process with your media of choice	87
5.3.2	Step-by-step guide using Installation Manager	89
5.3.3	Various installation options	98
5.4	Starting WDSC for the first time	101
5.5	Welcome	102
Chapter 6. Managing your WDSC instance		107
6.1	Updating and installing fixes for WDSC	108
6.2	Extending WDSC with optional software	115
6.2.1	List of optional software	115
6.3	Modifying the installation	118
6.3.1	Adding and removing components	118
6.4	Placing WDSC on an HTTP server	122
6.4.1	IBM Packaging Utility	123
6.4.2	Installation using IBM Packaging Utility	123
6.5	Uninstalling WDSC	127
Part 3. The Remote System Explorer and System i Projects		137
Chapter 7. Working with i5/OS-based development resources using the Remote System Explorer		139
7.1	Introduction to Remote System Explorer	141
7.1.1	A guided tour of RSE	142
7.2	Connecting to your i5/OS server	156
7.2.1	Profiles	156
7.2.2	Connections	158
7.2.3	TCP/IP ports required for WDSC	164
7.3	RSE subsystems are not i5/OS subsystems	165
7.3.1	System i Objects subsystem	166
7.3.2	System i Commands subsystem	167

7.3.3	Starting and stopping the remote servers	171
7.3.4	System i jobs subsystem.	172
7.3.5	Integrated File System Files subsystem	177
7.3.6	Qshells subsystem	181
7.4	Managing your System i objects	183
7.4.1	Types of filters in the System i Objects subsystem.	183
7.4.2	Working with System i Objects filters	190
7.4.3	Creating i5/OS objects in the System i Objects subsystem	191
7.4.4	Searching operations	198
7.4.5	Working with save files	200
7.5	Working with your library list	204
7.5.1	Adding a library list entry.	207
7.5.2	Changing the current library	208
7.5.3	Removing a library list entry	208
7.5.4	Moving libraries within your library list.	209
7.5.5	Specifying initial libraries and commands.	210
7.6	System i Table views	211
7.6.1	Populating System i Table views	212
7.6.2	Working with items in a table view	213
7.7	User actions	218
7.7.1	Creating user actions	218
7.7.2	Invoking user actions	220
7.7.3	Named types.	220
7.8	Additional parameters with actions issued from a table view	223
7.9	Drag and drop functionality	229
7.10	Compiling source members in Remote System Explorer	229
7.10.1	Verifying source members in RSE	229
7.10.2	Compiling RPGLE source members.	232
7.10.3	Working with compile commands	236
7.11	Running programs from the Remote System Explorer	239
7.11.1	Running a program: An example	239
7.11.2	Running modes	241
7.11.3	Ad hoc commands	244
7.12	Debugging programs in the RSE	245
7.13	Tips for using the Remote System Explorer	246
7.13.1	Use the right mouse button.	246
7.13.2	Standard workstation shortcut keys	246
7.13.3	Additional shortcut keys to remember: Ctrl+F6, Ctrl+F7, and Ctrl+F8	246
7.13.4	Context-sensitive help through F1	246
7.13.5	Missing a view? Reset Perspective!	247
7.13.6	Maximize the view by double-clicking the title bar	247
Chapter 8.	Remote Systems LPEX Editor: The modern SEU and much more	249
8.1	Editing source members in the RSE	250
8.1.1	Launching the Remote Systems LPEX Editor.	250
8.1.2	Working with the Remote Systems LPEX Editor.	252
8.1.3	Remote Systems LPEX Editor functions.	256
8.1.4	Outline views.	273
8.1.5	Wizards in the Remote Systems LPEX Editor	278
8.1.6	Customizing the automatic formatter	286
8.1.7	New in V7: Language template locations	287
8.1.8	New in V7: ILE RPG SQL parser	288
8.1.9	Other editing enhancements in V7	290

Chapter 9. Screen Designer Technical Preview - the modern Screen Design Aid (SDA) in WDSC AE 7.0.	293
9.1 Screen Designer overview	295
9.2 Launching the Screen Designer	295
9.3 Understanding the Screen Designer perspective	297
9.4 Using the Screen Designer editor pages	298
9.4.1 Design area for WYSIWYG DDS development.	299
9.4.2 Adding new items to your display file using the palette.	301
9.4.3 Adding a database reference field to your record format	305
9.4.4 Screen and record selection	306
9.4.5 Design Area Toolbar	311
9.4.6 Changing the magnification of the design area and the grid brightness	312
9.4.7 Seeing and editing your DDS statements in the Source page	312
9.5 See everything at a glance and navigate quickly with the Outline view	312
9.6 Working with detailed attributes in the Properties view	314
9.6.1 Changing field attributes	314
9.6.2 Changing file attributes	315
9.6.3 Working with reference fields	315
9.6.4 Using a properties template	316
9.7 Setting your preferences	316
Chapter 10. Application Diagram in WDSC AE - visualize program structures	319
10.1 The Application Diagram	321
10.1.1 Source Call Diagram to analyze ILE RPG or ILE COBOL sources.	321
10.1.2 Visualize ILE program dependencies with Program Structure Diagrams	322
10.1.3 The Application Diagram Viewer	323
10.2 Ways to create an Application Diagram	324
10.2.1 Create an Application Diagram while editing a source member	324
10.2.2 Using the Remote Systems view	325
10.2.3 Creating an Application Diagram from a System i Project	328
10.3 Working with an Application Diagram	328
10.3.1 Understand the Application Diagram Viewer legend.	328
10.3.2 Zoom in or out.	330
10.3.3 Application Diagram at a glance in the Outline view.	333
10.3.4 Get details from the Properties view.	335
10.3.5 Find the main entry point of your program	336
10.3.6 Change the automatic layout of the diagram	337
10.3.7 Modifying the diagram layout manually.	340
10.3.8 Switch the view	345
10.3.9 Using the Application Diagram Viewer Palette	346
10.3.10 Edit or browse the source	346
10.3.11 Adding notes and descriptive text to the diagram	346
10.3.12 Storing an application diagram for future use	349
10.3.13 Printing an Application Diagram	352
10.3.14 Setting your preferences.	353
Chapter 11. System i Projects	357
11.1 The System i Projects perspective	359
11.1.1 Working in disconnected mode.	359
11.2 Opening the System i Projects perspective	360
11.3 Creating a System i project.	361
11.3.1 Creating the project container.	362
11.3.2 Creating a local source physical file	366
11.3.3 Creating a local source member.	368

11.4	Importing and viewing remote objects	370
11.4.1	Importing remote host objects	370
11.4.2	Viewing remote objects	372
11.5	Working with local source members	373
11.5.1	General management	373
11.5.2	Editing	373
11.5.3	Source verify (offline)	373
11.5.4	Putting your connection into offline mode	375
11.6	Remote actions	377
11.6.1	Push operations	378
11.6.2	Compile actions	380
11.6.3	Build operations	386
11.7	Working disconnected actions	389
11.7.1	Actions applicable	389
11.7.2	Actions not applicable	389
11.8	System i projects enhancements in V7	390
11.8.1	Label decorations	390
11.8.2	Sequence numbers	391
11.8.3	Save files support for System i projects (WDSC AE 7.0)	396
11.8.4	Push method selection during compile	401
11.9	Team support and version control with System i Projects	401
11.9.1	Rational ClearCase	401
11.9.2	Using Rational ClearCase for i5/OS application development	402
	Chapter 12. Integrated System i Debugger	407
12.1	Available debuggers for i5/OS-based applications	408
12.2	Prerequisites for debugging System i applications	409
12.2.1	Compile options	409
12.2.2	Authority required to debug i5/OS applications	409
12.2.3	Debug considerations when writing programs	410
12.2.4	Starting up the debug router	410
12.2.5	Verify the preference for updating production files	410
12.2.6	Setting source line breakpoints prior to debugging	410
12.3	How to start a debug session	412
12.3.1	Service entry point for ILE applications	413
12.3.2	Start debugging an application using a service entry point	414
12.3.3	How to use the power of a debugging session	418
12.3.4	Attach to a running application by debugging a System i job	440
12.3.5	Launching a program object for debugging	445
12.3.6	Setting debug preferences	448
	Related publications	453
	IBM Redbooks	453
	Other publications	454
	Online resources	454
	How to get IBM Redbooks	456
	Help from IBM	456
	Index	457

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™	e-business on demand®	Redbooks®
AIX®	i5/OS®	Redbooks (logo)  ®
AS/400®	IBM®	RPG/400®
CICS®	Integrated Language Environment®	System i™
ClearCase®	Language Environment®	System i5™
COBOL/400®	Lotus®	VisualAge®
DB2®	OS/400®	WebSphere®
Domino®	Passport Advantage®	zSeries®
DRDA®	Rational®	

The following terms are trademarks of other companies:

Adobe Flash, Adobe, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

EJB, Enterprise JavaBeans, J2EE, Java, JavaBeans, JavaScript, JavaServer, JDBC, JRE, JSP, JVM, Sun, Sun Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Expression, Internet Explorer, Microsoft, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

WebSphere® Development Studio Client (WDCS) is a collection of many features and functions. This IBM® Redbooks® publication discusses the new features of WDCS 7.0, focusing on Remote System Explorer (RSE) and new ways of installing and managing the products. Specific topics that are covered in detail in this book include:

- ▶ Installing WDCS V7.0 using a new installation management tool of Installation Manager
- ▶ Managing a WDCS V7.0 instance
- ▶ Working with i5/OS®-based development resources using the Remote System Explorer
- ▶ Remote Systems LPEX Editor - The modern SEU and much more
- ▶ Screen Designer Technical Preview - The modern Screen Design Aid (SDA) in WDCS AE 7.0
- ▶ Application Diagram in WDCS AE - Visualizing program structures
- ▶ System i™ Projects
- ▶ Integrated System i Debugger

For features of the previous version, WDCS 5.1.2, refer to *WebSphere Development Studio Client for System i Version 5.1.2*, SG24-6961-01.

WebSphere Development Studio Client for System i (WDCS) is a collection of many features and functions. A list of these includes:

- ▶ **Integrated Development Environment:** Seamlessly integrate all your tools, teams, assets, and other workbench-based projects.
- ▶ **IBM WebFacing Tool:** Convert your existing 5250 interfaces to Web interfaces with minimal changes to the server application.
- ▶ **Remote System Explorer and System i Development Tools:** Develop and maintain System i applications while working with remote systems.
- ▶ **System i Java™ Development Tools:** Develop Java applications in an integrated development environment that supports the complete cycle of Java development.
- ▶ **System i Web Development Tools:** Create new e-business applications that use Web-based front ends to communicate with System i business logic.
- ▶ **Struts environment support:** Build Web applications based on the Model-View-Controller paradigm.
- ▶ **Web Services Development Tools:** Create a Web service to call one or more System i programs or service program procedures.
- ▶ **Integrated System i Debugger:** Debug code that is running on a System i machine.
- ▶ **Database Development Tools:** Develop and test the database elements of your application views, wizards, editors, and more.
- ▶ **Server Development Tools:** Increase productivity and save time by using this integrated test environment.
- ▶ **XML Development Tools:** Build DTDs, XLM schemes, XML and XSL files in a comprehensive visual XML development environment.
- ▶ **Performance Analyzer:** Recognize, isolate and fix performance problems early in the development cycle.
- ▶ **Team Development Support:** Get open support for pluggable source code control repositories.

- ▶ IBM WebFacing Tool Advanced Edition: Take advantage of enhancements to the standard edition, such as system screen support, portal support, single sign on and more.
- ▶ J2EE™ Development Environment: Create quality applications using J2EE.
- ▶ Portal Development Environment: Build and test portlets just as you would any other type of programming artifact.
- ▶ Enterprise JavaBeans™ Support (including Profiling): Simplify your middleware application development with automatic support for services.
- ▶ Log and Trace Analyzer: A common format correlates entries from multiple sources, facilitating problem determination and debugging of distributed applications.
- ▶ Unified Modelling Language (UML) Visualization: Visualize and edit Java code using standard UML diagrams.
- ▶ System i J2EE Connector Architecture (JCA) Connector: Eliminates the need for different access models for different back-end EIS systems.
- ▶ Clear Case LT: Reliable entry-level version control for small and medium project workgroups.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the IBM Toronto lab.

Yessong Johng is an IBM Certified IT Specialist at the IBM International Technical Support Organization, Rochester Center. He started his IT career at IBM as a S/38 Systems Engineer in 1982 and has been with S/38, AS/400®, and System i now for 25 years. He writes extensively and develops and teaches IBM classes worldwide in the area of IT Optimization, including Linux®, AIX® 5L™, and Windows® implementations on the System i platform. His other coverage areas include TCP/IP, Data and Networking Security, and WebSphere.

Wilfried Blankertz is a Senior IT Specialist for System i Technical Sales in the IBM IOT North located in Frankfurt, Germany. From 1995 to 1998, he was assigned to the ITSO, Rochester Center, where he co-authored more than 20 IBM Redbooks and taught IBM classes worldwide in all areas of OS/400® Groupware solutions and Systems Management. He also co-authored the first and second editions of this book. Before joining the ITSO, he worked as a Systems Engineer in IBM Germany, supporting customers with the AS/400 system and all its predecessor systems for 30 years. He is also a Certified Lotus® Professional for Administration and for Domino® R5 Application Development. He owns several certifications for WebSphere application development for System i. You can reach Wilfried at WilBlank@de.ibm.com.

Ahmed Mansour is a software engineer at the Cairo Technology Development Center, IBM Lab in Egypt. He holds a Bachelor's degree in Computer Science. He has been with IBM since 2006. Ahmed is a member of the System i/Rational® NLS team, focusing on providing Bidirectional languages and globalization support for IBM products and platforms, and also provides support for customers running DB2® Text Extender on the System i platform.

Hania Abd-El-Razik is a software engineer at the IBM Toronto Laboratory in Canada. She started her career at IBM in 2005, when she joined the WebFacing team. Since 2006 she has been the team's Test Lead, responsible for assuring the WebFacing Tool's quality. Her other areas of expertise include Web design and development, human-computer interaction, and graphic design. Hania holds a degree in Computer Science with Honours from York University, Canada, and can be reached at haniaa@ca.ibm.com.

Jiayun Zhu is a staff software engineer at the IBM Toronto Lab. She has worked on the IBM WebFacing Tool since 2000 and has been involved in the design and implementation of the WebFacing runtime infrastructure and various features including cluster support, CSS-P, and WebFacing/HATS interoperability. Now serving as Technical Lead for the WebFacing Tool, she is taking on a critical role in architecture design and major function definitions. Jiayun has a Master's degree in Computer Science from the University of Calgary, Canada.

Thanks to the following people for their contributions to this project:

Alison Butterill
Xuan Chen
Li Ding
Brian Farn
George Farr
Nazmin Haji
Mike Hockings
Howard Lau
Kou-Mei Lui
Maha Masri
Eric D. Peters
Eric Simpson
Claus Weiss
Don Yantzi
IBM Toronto

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Archived

Introduction to WDSC V7.0

This part provides an overview of WDSC V7.0. As we stated in the preface of the book, this IBM Redbooks publication discusses two main topics in detail in separate parts:

- ▶ Installing and managing of WDSC 7.0
- ▶ RSE and System i projects

Before we move on to those two main topics, this part discusses the following:

- ▶ A new set of application creation tools for System i provided in WDSC V7.0
- ▶ Overview of Eclipse, RSDP, and RAD
- ▶ WebFacing, HATS, and WDHT
- ▶ Overview of Web application development tools of WDSC V7.0

Archived



A new set of application creation tools for System i

This chapter describes the WebSphere Development Studio Client for iSeries V7.0 and WebSphere Development Studio Client for System i Advanced Edition V7.0 (or briefly WDSC V7.0 and WDSC AE V7.0).

1.1 IBM WebSphere Development Studio for System i

Starting with OS/400 Version 5 Release 1, most previous program products that were sold separately for application development were bundled together into a single product called IBM WebSphere Development Studio. WebSphere Development Studio for System i gives customers and solution providers rapid and cost-effective ways to increase the number of Web-enabled, e-business applications for the System i server. It marks a major shift in the packaging of IBM application development tools for the System i server. Rather than continuing to sell individual tools and compilers, this package consolidates all the key System i development tools, both host and workstation, traditional and new, into one attractively-priced System i offering. This product is a consolidated set of host and workstation tools optimized for a complete application development solution built on the System i server.

WebSphere Studio provides a flexible, portal-like integration of multi-language, multi-platform, and multi-device application development tools that maximize developer productivity, increase return on investment (ROI), and improve overall time to value.

During the year 2001, WebSphere Studio and VisualAge® for Java were merged together with a new application development workbench based on the Eclipse project. The new product was called WebSphere Studio Application Developer. Currently, there are several development tools based on the Eclipse platform.

Similarly, with OS/400 V4R5, WebSphere Development Tools for System i were restructured and renamed to WebSphere Development Studio Client for iSeries, which consolidates the key application development tools into one client-based package; it simply referred to the client-side tools.

Figure 1-1 illustrates how WebSphere Development Studio is packaged.

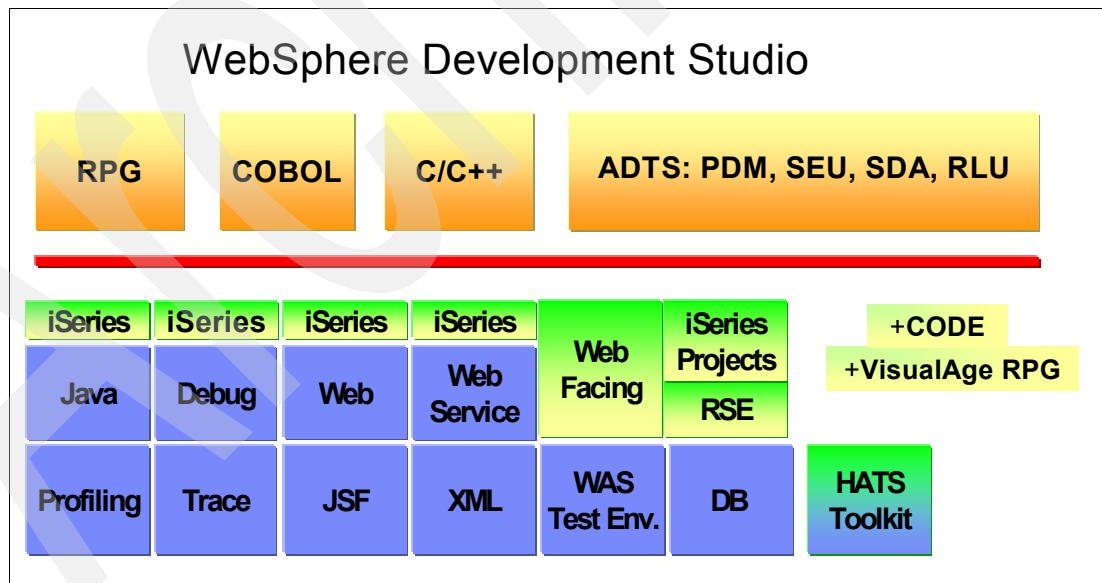


Figure 1-1 WebSphere Development Studio packaging

1.1.1 ILE RPG

ILE RPG is the compiler for RPG IV, the latest generation of RPG, enriched with the new functions you have wanted. With ILE, you have the power of a suite of compilers for multilanguage interoperability and faster call performance. Your investment in RPG applications is protected with full upward compatibility. A conversion aid helps you easily convert your RPG III applications to the RPG IV language definition.

To learn about the V5R3 enhancements made for RPG IV, go to:

http://www.ibm.com/software/ad/wds400/about/ile_rpg.html

You can also refer to the following IBM Redbooks about RPG programming on System i:

- ▶ *Moving to Integrated Language Environment for RPG IV*, GG24-4358
- ▶ *AS/400 Programming with VisualAge for RPG*, SG24-2222
- ▶ *Who Knew You Could Do That with RPG IV? A Sorcerer's Guide to System Access and More*, SG24-5402

1.1.2 ILE C/C++

The ILE C/C++ Compiler supports program development on System i servers in both the C and C++ programming languages. The C compiler is a high-performance, high-function compiler for C development on the System i platform. It gives you the combination of function-rich C and the power of ILE. ILE offers fast calls and a uniform run-time model, which are definite benefits for call-intensive C.

With the C++ Compiler, you can develop object-oriented, mission-critical applications for the client, System i server, or both in a client/server solution. It is a single, powerful object-oriented C++ programming development environment that integrates cooperative processing with graphical user interface development.

To learn about the V5R3 enhancements made for ILE C/C++, go to:

http://www.ibm.com/software/ad/wds400/about/ile_ccpp.html

1.1.3 ILE COBOL

With the ANSI-85 functions of ILE COBOL, you can easily port code from other platforms to the System i server. Your investment in applications is protected. You can be confident that IBM will continue to support COBOL on System i so you can still use applications written in COBOL.

To learn more about the V5R3 enhancements made for COBOL, see the WebSphere Development Studio for System i Web site at:

http://www.ibm.com/software/ad/wds400/about/ile_cobol.html

1.1.4 Application Development ToolSet

Application Development ToolSet (ADTS) for System i is a well-integrated set of host-based development tools you can use with confidence as you build your applications. It combines established tools, such as Programming Development Manager (PDM), Screen Design Aid (SDA), and Source Entry Utility (SEU), with new utilities. There are two optional features in the package that are especially well suited to help you to plan, manage, and develop applications in a multi-developer organization.

Here is a list of the ADTS components:

- ▶ Source Entry Utility (SEU)
- ▶ Screen Design Aid (SDA)
- ▶ Report Layout Utility (RLU)
- ▶ Data File Utility (DFU)
- ▶ Character Generator Utility (CGU)
- ▶ Advanced Printer Function (APF)
- ▶ Programming Development Manager (PDM)
- ▶ Host Access Programs for CODE and VisualAge RPG
- ▶ Interactive Source Debugger (ISDB)
- ▶ File Compare and Merge Utility (FCMU)

To learn more about ADTS, go to:

<http://www.ibm.com/software/ad/wds400/about/adts.html>

1.2 WebSphere Development Studio Client for iSeries

Figure 1-2 shows the contents of WebSphere Development Studio Client for System i.

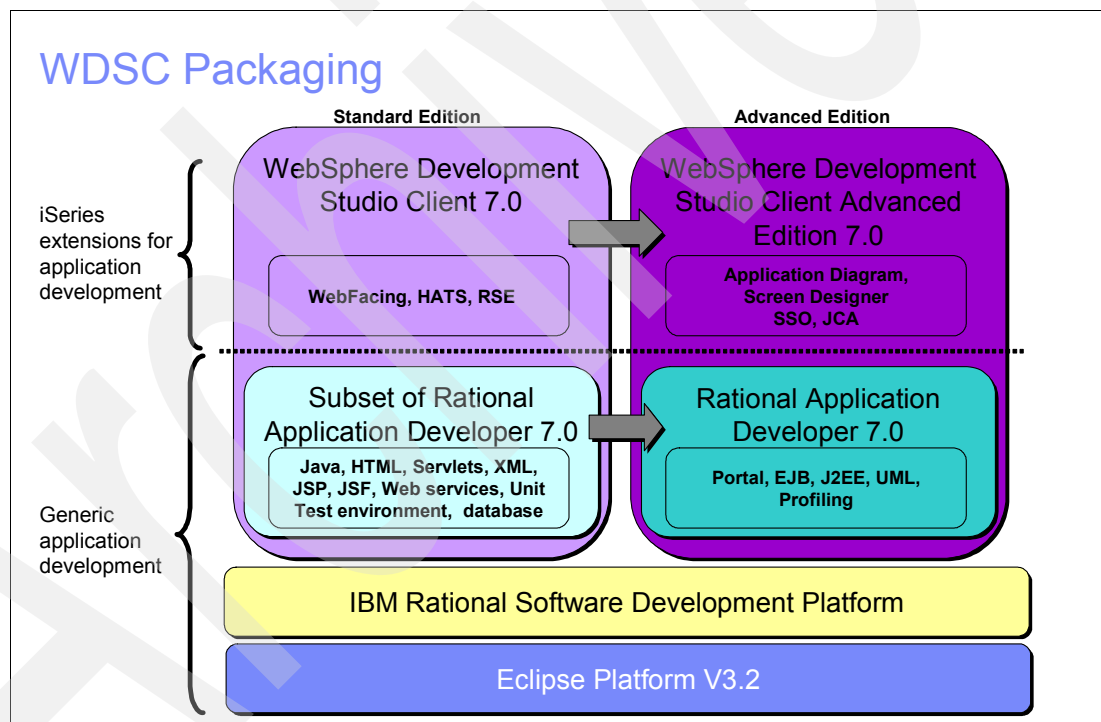


Figure 1-2 WebSphere Development Studio Client for System i packaging

The standard features of WDSK, which are depicted in the left column of Figure 1-2, come integrated with WebSphere Development Studio. You can purchase a separate product of WDSK AE if you need any of the features depicted in the right column of Figure 1-2, under the WDSK AE heading.

All System i-specific tools in WebSphere Studio products are built as plug-ins. They include:

- ▶ Remote System Explorer and System i development tools (see Part 2, "Installing and managing WDSK" on page 73)
- ▶ IBM WebFacing Tool

- ▶ System i Web and Java development tools
- ▶ Integrated System i debugger (see Chapter 12, “Integrated System i Debugger” on page 407)

You can build your own plug-ins for System i and integrate them with Development Studio Client.

1.2.1 Summary of enhancements from V5.1.2 to V7.0

Since V5.1.2, we have multiple releases with enhancements. This section summarizes the enhancements of each release.

WDSC V5.1.2 enhancements

- ▶ WebSphere Studio Site Developer Advanced V5.1.2:
 - Eclipse-based IDE for creating Web, Web services, and Java applications.
 - Enterprise Generation Language (EGL) for generating Java applications from a 4GL procedural language.
- ▶ A powerful IDE for System i development.
- ▶ Wizards to help Java, Web, and Web services developers access System i data and applications.
- ▶ Cooperative Development Environment (CODE): formerly called CODE/400, the classic edit/compile/debug environment for current CODE users.
- ▶ IBM integrated System i debugger (Eclipse-based) for all languages.
- ▶ Both System i refacing tools, IBM WebFacing Tool and the Host Access Transformation Services (HATS) toolkit.

WDSC AEV5.1.2 enhancements

WebSphere Development Studio Client Advanced Edition for System i Version 5.1.2 provides all the capabilities of WebSphere Development Studio Client for System i Version 5.1.2 plus the following enhancements:

- ▶ WebSphere Studio Application Developer 5.1.2 to develop sophisticated, scalable, and fully J2EE-compliant applications (including Enterprise JavaBeans).
- ▶ IBM WebFacing Tool advanced enhancements to deliver portal support and to leverage and extend Web-enabled applications with Web applications.
- ▶ Portal development tools.
- ▶ System i Web and Web Facing Tools to support Enterprise Identify Mapping (EIM)-based single sign-on capability.
- ▶ System i Java Tools Java Connector Architecture (JCA) support.

You can quickly develop and deploy traditional and e-business applications on your System i server with IBM WebSphere Development Studio Client for System i. This powerful suite of tools represents the next generation of WebSphere development tools for System i. It is the client component of the IBM WebSphere Development Studio product, which contains all of the host Integrated Language Environment® (ILE) compilers for RPG, C, C++, COBOL, and the Application Development Toolset. This client component of the overall package is designed to help you accomplish three primary programming goals:

- ▶ Develop and maintain System i business logic
You have several workstation components that help you to develop and maintain the ILE programs: CODE, Remote Systems Explorer, System i exploration, and command execution.
- ▶ Create Web front ends to System i business logic
Development Studio Client gives application developers, with traditional System i programming skills, the ability to quickly develop e-business applications without having to learn Java and other Web application skills. It provides a set of wizards that takes you through the steps required to create Web input and output pages.
- ▶ Create graphical user interface (GUI) front ends to System i business logic
With VisualAge RPG, you can develop and maintain client/server applications in a visual development environment on the workstation. You leverage your existing RPG skills to create GUI applications that you can deploy as native Windows applications or as Java applets that can run on any Java-capable Web browser. These applications can access System i data and other System i objects.

WDSC V6.0 enhancements

WDSC V6.0 is an IDE for creating traditional i5/OS applications and e-business on demand® solutions. Key enhancements include:

- ▶ WebSphere Development Studio Client V6.0 is based on RWD.
- ▶ WebFacing Tool delivers additional Web Settings to make it easier to customize your Web-enabled interface to 5250 applications to look and feel like a native Web application. These applications run in batch mode with no Online Transaction Processing (OLTP) requirements.
- ▶ HATS Toolkit V6.0 delivers a new Java Server Faces (JSF) support for Integration Objects and enhanced sub-file support for its dynamic run-time transformation of the 5250 data stream. These applications require OLTP capacity.
- ▶ Remote System Explorer (RSE), the replacement for PDM and SEU, delivers quality, function, and usability enhancements for developing native System i applications.
- ▶ System i Web Tools improves subfile support and Web diagram support.
- ▶ System i Java Tools adds visual components and beans to the Visual Editor for Java palette and delivers ease of use enhancements.
- ▶ System i integrated debugger enhances support for end-to-end debugging of System i native and e-business applications across multiple tiers and platforms.
- ▶ WDSC AE for System i V6.0 includes all the capabilities in RAD.

WDSC AE V6.0 enhancements

- ▶ RAD delivers everything you need to create well-designed, scalable code ready for production.
- ▶ WebFacing Tool delivers advanced capabilities for customization, access to System i system screens in batch mode, the ability to generate native portlets, and single signon support using Enterprise Identity Mapping (EIM).
- ▶ Support for Cascading Style Sheets-Positioning (CCS-P) will be available in a fixpack update.
- ▶ RSE now supports AIX and Linux server development on System i.
- ▶ In System i Java Tools, the generated Program Call Java bean now supports J2EE Connector Architecture (JCA).

- ▶ The Enterprise Generation Language (EGL) component in WDSC AE is replacement product for VisualAge Generator for AS/400.

WDSC V6.0.1 enhancements

- ▶ It is based on IBM Rational Web Developer for WebSphere Software (RWD) refresh pack V6.0.1 (part of the IBM Rational Software Development Platform) and it includes other Eclipse-based development tools that allow you to adapt and extend your development environments with IBM and partner tools.
- ▶ IBM WebFacing Tool delivers extensions to allow Web-enabled applications created with the WebFacing Tool to interoperate with 5250 applications in a browser.
- ▶ Host Access Transformation Services (HATS) Toolkit V6.0.4 delivers dynamic run-time transformation of the 5250 datastream with no requirement for Online Transaction Processing (OLTP) capacity working through the WebFacing server starting with i5/OS V5R4.
- ▶ Deployment of HATS applications and applications using the WebFacing Tool extensions requires IBM WebFacing Deployment Tool for WebSphere Development Studio with HATS technology.
- ▶ A new WDSC Lite technology preview provides a lightweight edit/compile/debug environment for developing native i5/OS applications.
- ▶ It provides V5R4 support for RPG, COBOL, and CL.
- ▶ It has a new extension to the Web Service wizard to create Web services from ILE RPG/COBOL programs in one step, making it easier to create SOA applications.
- ▶ The WebFacing Tool and Web Tools generate Common Base Event (CBE) logging data that can be used by the Log and Trace Analyzer, provided in WebSphere Development Studio Client Advanced Edition for System i.

WDSC AE V6.0.1 enhancements

- ▶ It is based on Rational Application Developer for WebSphere Software refresh pack V6.0.1.
- ▶ It includes all of the WDSC enhancements mentioned in “WDSC V6.0.1 enhancements” on page 9.
- ▶ The Log and Trace Analyzer consolidates i5/OS, Web, and WebFacing messages for easier problem determination of composite (mixed workload) applications.
- ▶ Support for repositioning of DDS fields unconstrained by the boundaries of the original 5250 area.

WDSC V7.0 enhancements

WDSC V7.0 inherits and extends the robust, easy-to-use Rational Software Delivery Platform (RSDP) and a subset of Rational Application Developer for WebSphere Software V7.0 to deliver an integrated development environment (IDE) with tools for developing Web, Web services, client/server, and i5/OS server applications using programming languages such as RPG, COBOL, CL, and Java.

WDSC AE V7.0 contains all of the development tools included in WebSphere Development Studio Client, plus it inherits and builds on additional premium Web, Enterprise Java Bean (EJB™), and J2EE development capabilities from IBM Rational Application Developer V7.0. In addition, it provides specific advanced System i tools including Navigation support, ClearCase® integration, and an Eclipse-based Screen Designer technology preview.

The focus for WDCS V7.0 is to deliver the power of the RSDP to the System i market and to provide enhancements for System i application developers. WDCS AE V7.0 includes additional capabilities to meet more advanced development needs.

WDCS V7.0 enhancements

- ▶ Based on Rational technology.
- ▶ Provides selective installation features for flexibility.
- ▶ Includes Remote System Explorer (RSE) that delivers function and usability enhancements for developing native i5/OS applications.
- ▶ Delivers the Integrated System i Debugger that now supports debugging a suspended program.
- ▶ Includes the WebFacing Tool and the HATS Toolkit that provide the ability to link WebFacing Web applications with HATS applications.
- ▶ Uses WebFacing Deployment Tool for WebSphere Development Studio with HATS technology V7.0 run-time license for deploying WebFacing, HATS, and linked WebFacing/HATS applications to production.
- ▶ Includes HATS that lets you transform access to your existing host applications as a rich client application.
- ▶ Includes the i5/OS Java Tool that has now updated the Program Call Bean wizard to generate JavaBeans that can be used with the JavaServer™ Faces tools in WebSphere Development Studio Client.

WDCS AE V7.0 enhancements

- ▶ Includes the full capabilities of Rational Application Developer V7.0 and WebSphere Development Studio Client V7.0.
- ▶ Provides a new tool called the Application Diagram component for visualizing ILE RPG and ILE COBOL applications.
- ▶ Provides a technology preview in RSE: Screen Designer to graphically design and modify DDS display files.
- ▶ Delivers enhanced System i project support to make it easier to use Rational ClearCase as your software configuration and change management repository for native i5/OS application development.

For more information, refer to the WebSphere Development Studio Client for System i Web site at:

<http://ibm.com/software/ad/wdt400/>

1.2.2 IBM WebFacing Tool

The IBM WebFacing Tool is introduced to help you add a Web browser interface to existing 5250 applications. The IBM WebFacing Tool creates the Web interface by converting, at development time, the display file source into JSP™ and Java databeans. These Java components are deployed to the WebSphere Application Server. In combination with the WebFacing run-time, they provide a Web interface to the existing 5250 application.

No or little change is required to the application's business logic. The same logic can support both a 5250 and Web interface. There is no dual maintenance. The conversion process is easy and cost-effective. There are no separate tool costs or run-time charges associated with the IBM WebFacing Tool.

For more information about the IBM WebFacing Tool, see the IBM WebFacing Tool Web page at:

<http://www.ibm.com/software/awdtools/wdt400/about/webfacing.html>

1.2.3 Web tools: System i extensions

The System i specific Web tools based on Struts, for example, Web Interaction wizard and Web Components, are deprecated. JSF is the new strategic direction.

1.2.4 Java development tools

Java development tools and System i Java development tools give you the ability to develop Java applications and write, compile, test, debug, and edit programs written in the Java programming language for Java application development. Java development tools include an integrated environment that supports the complete cycle of Java program development.

The System i Java development tools include:

▶ **Program Call wizard**

This extension generates a JavaBeans that contains the IBM Toolbox for Java code needed to call your System i programs from Java code. You can also use the output to create Web Services that call System i programs.

▶ **File Export/Import wizards**

This extension provides the ability to import and export files to and from the integrated file system on a System i server into a Java project.

▶ **The System i Java Transform and Run view**

This extension provides all the essential tools to remotely develop System i Java applications. You can remotely transform, run, and debug Java applications.

▶ **Pre-supplied JavaBeans**

The extension continues to provide useful JavaBeans previously available in the IBM VisualAge for Java Enterprise Toolkit for AS/400 (ET/400) package. These beans include:

- *Data File Utility beans*: To extend the support of code to access one or more System i database files and to manipulate records within the files.
- *Swing JFormatted beans*: To let you convert System i fields and attributes and to provide edit code, edit word, formatting, and verification capabilities.
- *Object List beans*: To let you access System i object names (for example, libraries, files within a library, or user IDs on the system) and to allow you to set listing properties for selecting the desired type of object list.
- *Database Field List beans*: To retrieve a list of fields in a specified System i database file and record.

▶ **Integration of the IBM Toolbox for Java into the WebSphere Workbench**

This extension includes all help and examples associated with the Toolbox.

▶ **Java Tools for System i examples**

This extension provides a series of examples and samples for various tasks related to Java development for the System i using Java Tools.

1.2.5 CoOperative Development Environment

Planning statements: This is a planning statement for CoOperative Development Environment (CODE), as of March 2007:

“The CoOperative Development Environment (CODE) tools (CODE Editor, CODE Designer, CODE Project Organizer, CODE Program Generator and the IBM Distributed Debugger) are deprecated in WDSC V7.0 and WDSC AE V7.0.

This means there will be no future enhancements made to these tools, they will not be updated to work with new releases of i5/OS, and they will not be ported to run on future versions of the Windows operating system. Customers using these tools should migrate to use the equivalent function in the Remote System Explorer and System i Integrated Debugger.

The CODE tools will continue to be supported until Development Studio Client V7.0 reaches end of service.

All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.”

CODE is a client/server development environment for writing host or server code. It allows you to develop or maintain applications in ILE RPG, RPG/400®, ILE COBOL, COBOL/400®, ILE C, CL, ILE CL, data description specification (DDS), and Java. It is an easy, cost-effective, and productive way to work with System i host applications on your Windows 98, ME, NT, 2000, or XP workstations.

You can use Client Access, Personal Communications, or TCP/IP to connect to the System i server. However, CODE does not require a continuous connection to the System i. Many of the CODE features can function in a disconnected mode.

CODE includes a set of flexible workstation tools for developing and maintaining System i application programs. The workstation tools include the editor, screen, report and database designer, syntax checker, program verifier, project organizer, and a debugger front end. This combination provides the productivity of the PC environment and integrity of the System i.

CODE includes the CODE Designer, a *what you see is what you get* (WYSIWYG) utility for designing display files and printer files. This new, powerful, and easy-to-use interface can make designing your DDS screens, reports, and physical files both fast and fun.

Another useful feature of CODE is the CODE Project Organizer. With this utility, you can access and manipulate System i objects with greater ease than PDM. You can also create a project and set up filters to gain quick GUI access to your frequently used OS/400 objects, members, application development manager projects, groups, and parts. Use the pop-up menus on these items to perform actions such as edit, compile, and debug. Use the CODE Actions window (notebook) to create and manage user-defined actions.

With Development Studio Client for System i, instead of using CODE Organizer, we recommend that you use the Remote Systems Explorer in WebSphere Studio Workbench.

1.2.6 VisualAge RPG

Planning statements: This is a planning statement for Rich User Interface Development, as of March 2007:

“Currently you can leverage your existing RPG IV skills to create rich client applications by using VisualAge RPG 6.0, which is included in WDSC 7.0 and WDSC AE 7.0.

IBM is exploring enhanced functions to enable you to use your RPG skills in the creation of rich user interfaces more effectively, targeting a wide variety of devices. IBM plans to define a path that allows VisualAge RPG users to take full advantage of these exciting new capabilities with minimal effort.

All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.”

VisualAge RPG is the premier IBM development tool, compiler, and run-time for creating GUI applications that run on Windows or any GUI-capable client and provide seamless access to System i data and applications. The VARPG compiler is based on RPG IV.

VisualAge RPG features point-and-click simplicity for RPG programmers at the workstation. Use VARPG to capitalize on your existing RPG skills and start developing powerful e-business applications.

When built, your applications run on a workstation and can access System i host data and objects. VARPG integrated components allow application developers to preserve their current skills and easily develop OS/400 applications with GUIs. With VARPG, you can build an application from the top down. You start by focusing on the look and feel of the interface. Then, you tie all the parts together with workstation RPG logic that you write in the VARPG language. You can reuse RPG logic and display files (DSPFs) from an existing application.

You can now generate Java applications and Java applets from the VARPG source, making RPG a powerful e-business language. VARPG offers:

- ▶ Visual Development Environment
- ▶ RPG compiler on the workstation
- ▶ Programmable language-sensitive editor
- ▶ Display file import facility
- ▶ Transparent PC-to-OS/400 connectivity
- ▶ Source-level debugger

Archived



Eclipse, RSDP, and RAD

This chapter briefly introduces you to the new GUI framework called *WebSphere Studio Workbench*. WebSphere Development Studio Client for iSeries (WDSC) is built on top of this framework.

After reading this chapter, you should understand the concept of the WebSphere Studio Workbench and be able to navigate through the new GUI. For more detailed information, refer to the online help.

2.1 Building blocks of WDSC

WebSphere Development Studio for System i is a comprehensive suite of application development tools for both e-business and System i server development.

WDSC Packaging

WDSC V7.0 delivers the power of the Rational Software Delivery Platform to the System i market and provides enhancements for System i application developers. WDSC AE V7.0 includes additional capabilities to meet more advanced development needs.

WDSC V7.0

WDSC V7.0 inherits and extends the Rational Software Delivery Platform (RSDP) and a subset of Rational Application Developer for WebSphere Software V7.0 to deliver an integrated development environment (IDE) with tools for developing Web, Web services, client/server, and i5/OS server applications using programming languages such as RPG, COBOL, CL, and Java; see Figure 2-1.

WDSC V7.0 includes the following:

- ▶ Based on Rational technology.
- ▶ Provides selective installation features for flexibility.
- ▶ Includes Remote System Explorer that delivers function and usability enhancements for developing native i5/OS applications.
- ▶ Delivers the Integrated System i Debugger that now supports debugging a suspended program.
- ▶ Includes the IBM WebFacing Tool and the HATS (Host Access Transformation Services) Toolkit that provide the ability to link WebFacing Web applications with HATS applications.
- ▶ Uses IBM WebFacing Deployment Tool for WebSphere Development Studio with HATS technology V7.0 run-time license for deploying WebFacing, HATS, and linked WebFacing/HATS applications to production.
- ▶ Includes HATS that lets you transform access to your existing host applications as a rich client application.
- ▶ Includes the i5/OS Java Tool that has now updated the Program Call Bean wizard to generate JavaBeans that can be used with the JavaServer Faces tools in WebSphere Development Studio Client.

WDSC AE V7.0

WDSC AE V7.0 contains all of the development tools included in WDSC, plus it inherits and builds on additional premium Web, Enterprise Java Bean (EJB), and J2EE development capabilities from IBM Rational Application Developer V7.0. In addition, it provides specific advanced System i tools including Navigation support, ClearCase integration, and an Eclipse-based Screen designer technology preview.

WDSC AE V7.0 includes the following:

- ▶ Includes the full capabilities of Rational Application Developer V7.0 and WDSC V7.0.
- ▶ Provides a new tool called the IBM Application Diagram Component for visualizing ILE RPG and ILE COBOL applications.
- ▶ Provides a technology preview in Remote System Explorer: Screen Designer to graphically design and modify DDS display files.

- ▶ Delivers enhanced System i project support to make it easier to use Rational ClearCase as your software configuration and change management repository for native i5/OS application development.

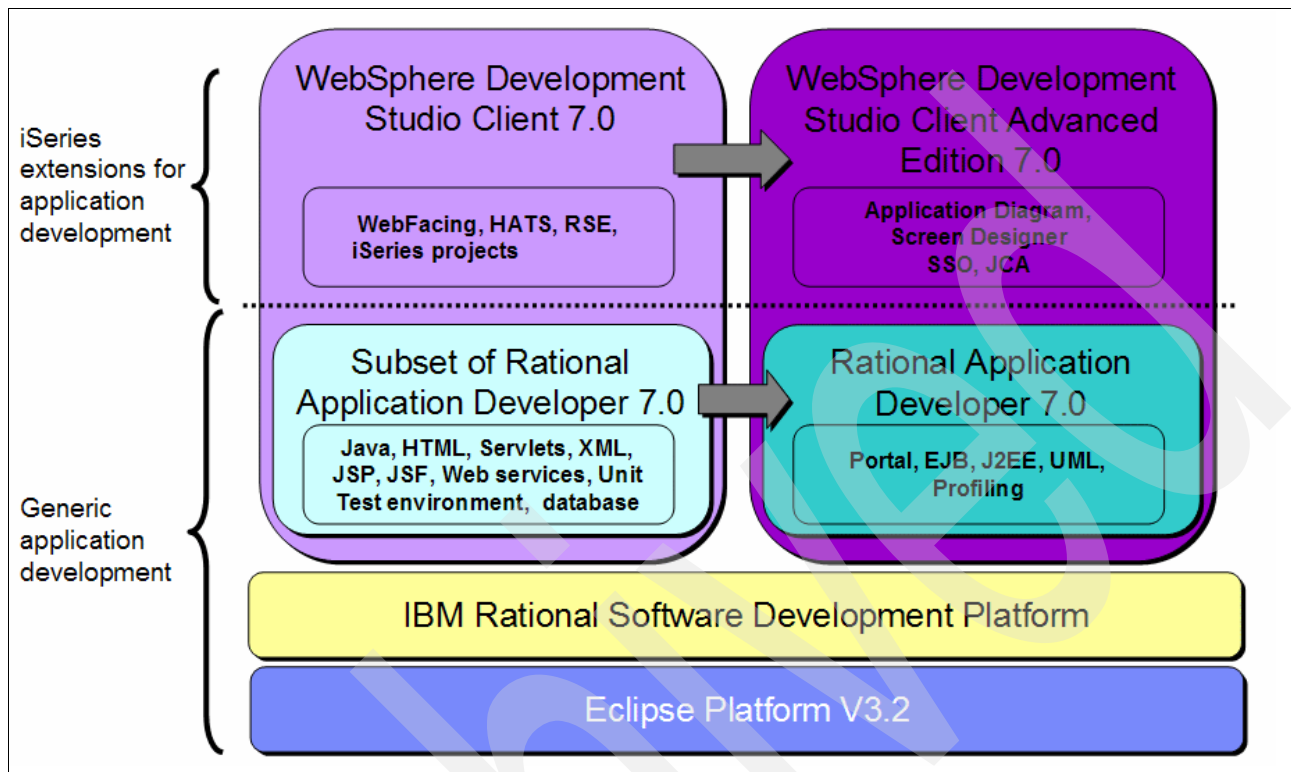


Figure 2-1 WDCS Building blocks and packages

2.1.1 Eclipse: Open source integrated development environment

The last ten years in the IT industry were turbulent. The birth of Java, the adoption of the new application model, and the new set of standards all contributed to this fact. The wide spread of the Java language brought to life the new paradigm “Write Once, Run Anywhere”. As Java matured and the Internet became an integral part of our lives, the new ideas and proposals came from many IT companies. These ideas tried to leverage the advantages of the Internet and the portability of Java applications. With time, this collective effort evolved into the specification called Java 2 Platform, Enterprise Edition (J2EE). In fact, this specification embraced many other specifications, such as Java Server Pages (JSP) and Enterprise JavaBeans (EJB).

All these new developments required a new set of tools. While the tools were built by many vendors, IBM was one of the leaders among tool builders with its VisualAge for Java and WebSphere Studio products. However, adoption of the J2EE application model changed the way applications were built and packaged. A new application model was a challenge for many tool builders. It was hard to catch up with the frequent changes in the industry.

IBM developed a new approach to solve this problem. That was to build a tool framework that was flexible enough to quickly adjust to the new technologies. The result of this effort is known as *Eclipse*. To make Eclipse a universal framework, IBM donated it to the open source community and created the Eclipse consortium. More than 150 companies are part of this consortium.

Eclipse is a Java-based integrated development environment (IDE), which is fully extendable by users, customers, and vendors. The code base is written in Java and runs on Windows and Linux. To extend it, programmers write Java code in the form of plug-ins. You can download Eclipse, both the binary form and the source code itself, from the Eclipse Web site:

<http://www.eclipse.org>

The IBM framework, which is built on Eclipse technology, is called WebSphere Studio Workbench. You can learn more about the WebSphere Studio Workbench in Chapter 1 of *WebSphere Studio Application Developer Programming Guide*, SG24-6585.

A new approach to the application development process

New technologies have changed the way we develop an application. Wide acceptance of the J2EE specification imposes new requirements on the development tools. For example, J2EE can define such things as platform roles, where the process of building, assembling, and deploying an application is split into multiple roles or responsibilities.

Consider the role in J2EE called *Application Component Provider*. The responsibilities for this role include building an application or application component, such as Hypertext Markup Language (HTML) pages, JavaBeans, or servlets. IBM has used this role-based approach in the design of WebSphere Studio Workbench.

This chapter looks at the implementation of this design in WebSphere Development Studio Client for iSeries.

2.1.2 Rational Software Delivery Platform

Organizations are continually reinventing business services without a structured approach to reuse. They are challenged with connecting and managing disparate IT resources across functions as well as physical locations, and are investing redundantly across IT and systems development projects.

IBM Rational Software Delivery Platform, desktop products V7.0 offer new capabilities empowering global teams to better implement and manage the delivery of software and systems architectures with improved life cycle quality.

They enable distributed software and systems development teams to:

- ▶ Accelerate SOA initiatives by accelerating service delivery within a business-driven architecture
- ▶ Ensure global architectural integrity and software quality, including support for packaged applications
- ▶ Simplify modular systems development with improved hardware and software modeling and standards support
- ▶ Improve consumability from the latest Eclipse community standards and new IBM collaborative engineering innovation

IBM Rational software development products simplify communication and collaboration across geographically distributed teams. This release enables organizations to ensure architectural integrity and quality across geographically distributed teams and a wide variety of packaged applications and systems for successful end results. Delivery teams can now leverage improved ease-of-use features and functionality that enable them to integrate and trace requirements across the entire delivery life cycle, synchronize, deploy and run test assets remotely, and ensure quality across software and systems delivery cycles.

V7.0 also offers model-driven development capabilities that include reverse transformations to help team members visualize differences between as-designed and as-built architectures. To compress delivery cycles, IT managers can decompose the models into separately controllable units for parallel development. In addition, the model-driven approach now supports SOA to encourage broad corporate-wide reuse of business services, tighten integration of composite application testing, and accelerate creation of enterprise-level Web services.

Governance infrastructure provides extensive process guidance and best practices for the accelerated creation and reliability of reusable services. Systems development and delivery is a highly technical process of analysis, design, implementation and maintenance. These new products help you govern the entire process, ensuring that all code development, testing, fixes, and maintenance for complex systems are in line with broader business objectives and meet compliance regulations.

These integrated sets of tools expand the proven capabilities of Eclipse 3.2 and extend the Eclipse framework across the enterprise. Additional capabilities can be developed or added with plug-ins from the open source community or with software and technical support from IBM and its ecosystem of IBM Business Partners.

2.1.3 Rational Application Developer

Rational Application Developer is built on Eclipse 3.2, an extensible platform for constructing and integrating software development tools. Rational Application Developer is part of the Rational Software Development Platform family of products. Each product in the Rational desktop family uses the same integrated development environment (IDE). The differences among these products reflect the optional features that are available and installed in each configuration.

You can use RAD to quickly design, develop, analyze, test, profile, and deploy high-quality Web, service-oriented architecture (SOA), Java, J2EE, and portal applications.

For more information on RAD, select **Help** → **Help Contents** → **Rational Developer product documentation**.

2.2 Components of the Eclipse development environment

In general, during any development process, you work with such resources as source files, connections, images, and so on. A development tool gives you the way to look, access, and manipulate these resources in multiple ways. For example, a Java developer's view of the source files is different from the view of a person responsible for building or assembling an application from the source files. By recognizing this fact, you expect a tool builder to present the project resources to you in a most convenient and productive way.

WebSphere Studio Workbench fulfills these expectations by providing the following GUI components:

- ▶ Workbench
- ▶ Projects
- ▶ Perspectives
- ▶ Views
- ▶ Editors
- ▶ Crash recovery

2.2.1 Workbench

When you start the WDSC, you are prompted to select your workspace location (Figure 2-2). You may accept the default location, or you can browse the new location.

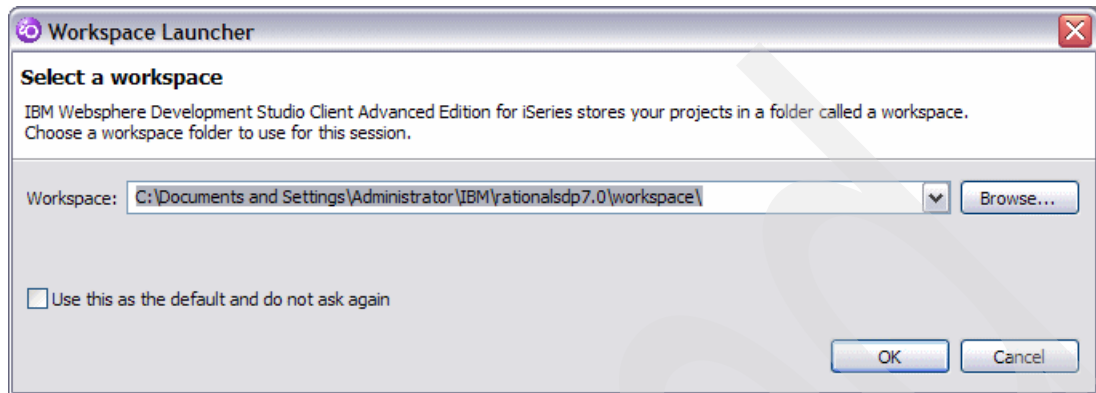


Figure 2-2 Selecting the workspace location

If you select the “Use this workspace as the default and do not show this dialog box again” option and later you need to change your default workspace location:

1. Select **Window** → **Preferences** from the workbench menu.
2. Select **General** → **Startup and Shutdown** in the preferences dialog.
3. Select the option **Prompt for workspace on startup**, see Figure 2-3.

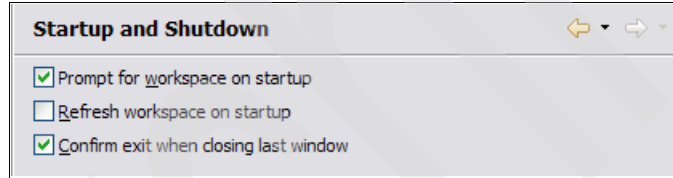


Figure 2-3 Prompt for workspace

Tip: You can create several work spaces. By doing so, your work is organized properly and you can achieve better performance.

After you start the Workbench, you see a single window. The Workbench window displays one or more perspectives. A perspective contains views, such as the Navigator, and editors. More than one Workbench window can be open at any given time.

2.2.2 Projects

Projects are base components for building, version management, sharing, and organizing resources. A project can contain session and persistent properties, settings for environmental variables, and references to other projects.

Types of projects include:

- ▶ **Simple project:** Simple projects are generic projects that contain files and folders.
- ▶ **System i project:** System i projects provide you with a workspace on your local workstation that will contain local code for a project. You can work while disconnected with

the System i Project Navigator, and occasionally perform synchronization to build your project and push your work to the server.

- ▶ **Java project:** Java projects contain Java code.
- ▶ **Web project:** Web projects contain the resources needed for Web application, including servlets, JSP files, Java files, static documents (for example HTML pages or images), and any associated metadata.
- ▶ **WebFacing project:** WebFacing projects enables you to quickly convert your DDS display file source members so that the user interface of your System i programs can run in a browser.
- ▶ **J2EE project:** J2EE projects contain Connector projects, Application client projects, Utility projects and Enterprise application projects.
 - The Connector is a J2EE standard extension mechanism for containers to provide connectivity to enterprise information systems (EISs).
 - The Enterprise application projects contain references to the resources needed for enterprise applications and can contain a combination of Web modules, Java Archive (JAR) files, EJB modules, and application client modules.
 - The Application client projects contain the resources needed for application client modules.
- ▶ **EJB project:** EJB projects contain the resources for EJB applications.
- ▶ **Eclipse Modeling Framework (EMF) project:** Eclipse Modeling Framework is a Java framework and code generation facility for building tools and other applications based on a structured model.
- ▶ **Server project:** Server projects describe the test environment for applications.
- ▶ **Plug-in Development project:** Plug-in Development projects help you to develop platform plug-ins while working inside the platform workbench.
- ▶ **Enterprise Generation Language (EGL) project:** Enterprise Generation Language is a development environment and programming language that enables you to write full-function applications quickly, thereby freeing you to focus on the business problem your code is addressing, rather than on software technologies.

Note: EGL is a separate product now, which needs to be purchased separately and installed.

2.2.3 Perspectives

A *perspective* defines the layout of views in a workbench window and the available actions that can be performed on the resources. Multiple perspectives can exist in a workbench window. Each perspective targets a specific role or a set of responsibilities in the development process. Figure 2-4 shows a sample workbench window with multiple perspectives opened. The Remote Systems perspective is opened by default.

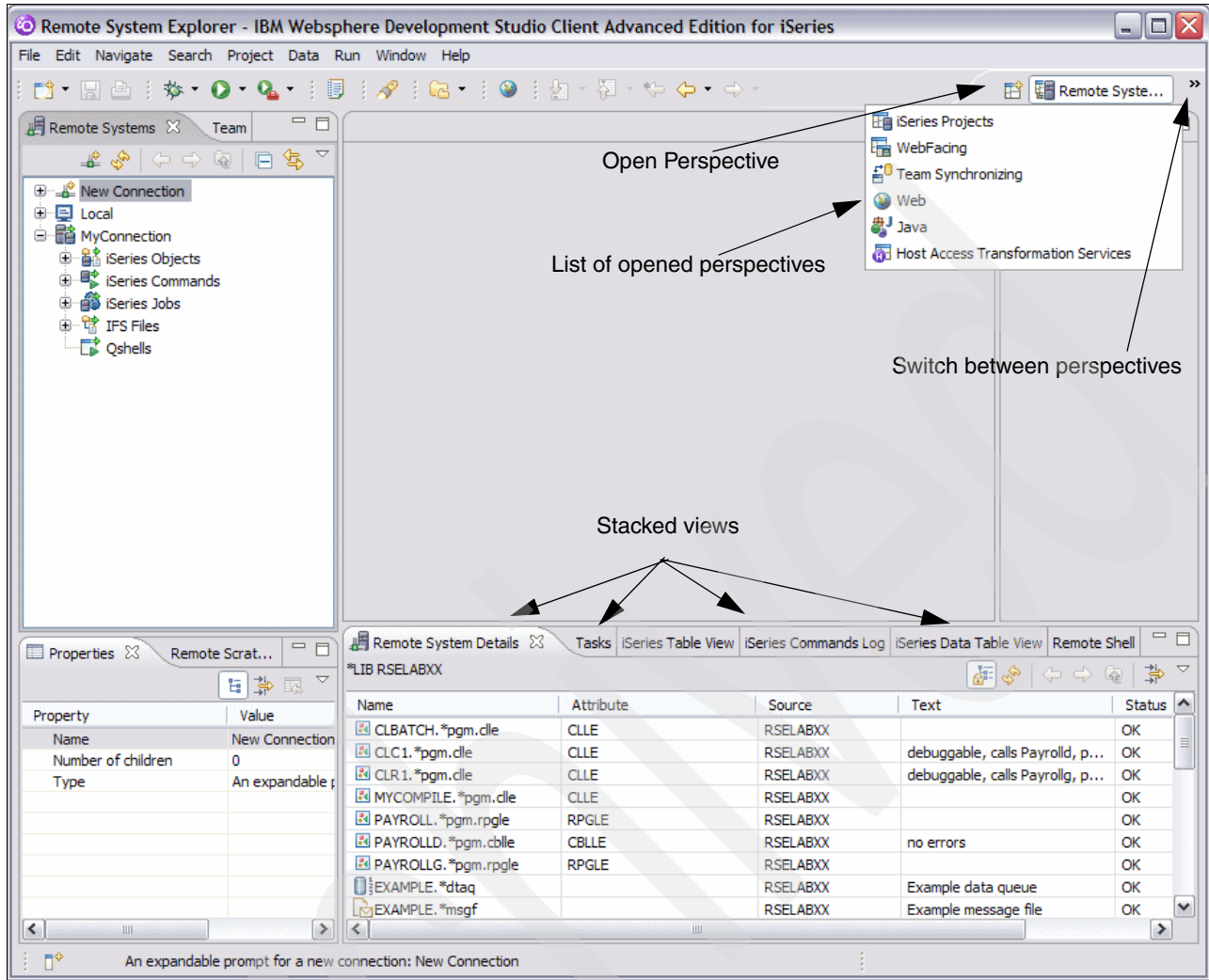


Figure 2-4 A workbench window

To open more perspectives, select **Window** → **Open perspective** and select a suitable perspective from the menu, as shown in Figure 2-5.

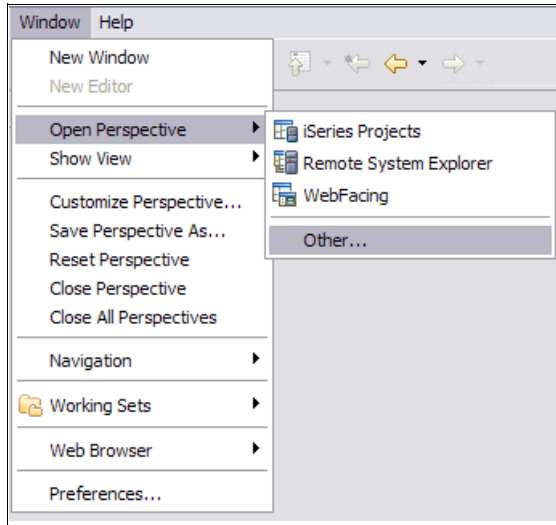



Figure 2-5 Open perspective

Alternatively, you can click the Open Perspective () button (see Figure 2-5) and select a perspective.

Let us look at the basic navigation techniques in WDSC. First, you can switch between opened perspectives by clicking the >> symbol to see a complete list of opened perspectives; see Figure 2-5. Alternatively, you can use the Ctrl+F8 shortcut key to toggle between opened perspectives. If you need to open another perspective, click **Open Perspective** and select a perspective from the menu. Each perspective button has a distinct image associated with it. Right-clicking any perspective button displays a menu from where you can select the Close Perspective option.

2.2.4 Views

Views provide a way to show the resources to a user. They also support multiple editors. Each view shows a subset of the resources available in the workbench, but depends on the purpose of a view. For example, the Gallery view in Web Perspective provides the way to see all multimedia resources that come with the tool. The Navigator view presents the existing projects in a tree-like manner. By its functionality, the Navigator view is similar to Windows Explorer.

Besides a distinct way of presenting the resources, you can activate a view-specific menu by right-clicking a resource or anywhere in the view. WebSphere Studio Workbench presents only the menu options that are available for the selected resource. If you experience a situation where the instructions say, “Select from the pop-up menu...”, and you don’t see that choice, verify that you are working with the correct view.

In our example, the Remote Systems perspective has five views:

- ▶ **Remote Systems view** (Figure 2-6): The Remote Systems view contains information about your current connections, filters, and much more.

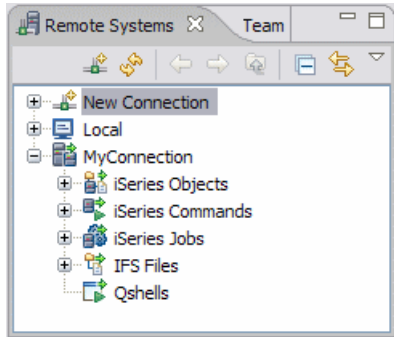


Figure 2-6 The Remote Systems view

- ▶ **Properties view** (Figure 2-7): The Properties view displays property names and basic properties of a selected resource.

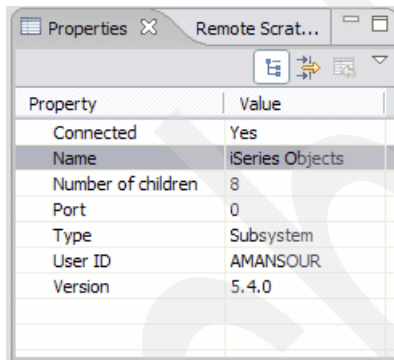


Figure 2-7 Properties view

- ▶ **Tasks** (Figure 2-8): The Tasks view contains line items for system-generated problems, warnings, and errors. You can add your own entries to the table, for example to build a list of to-do items, or tasks.

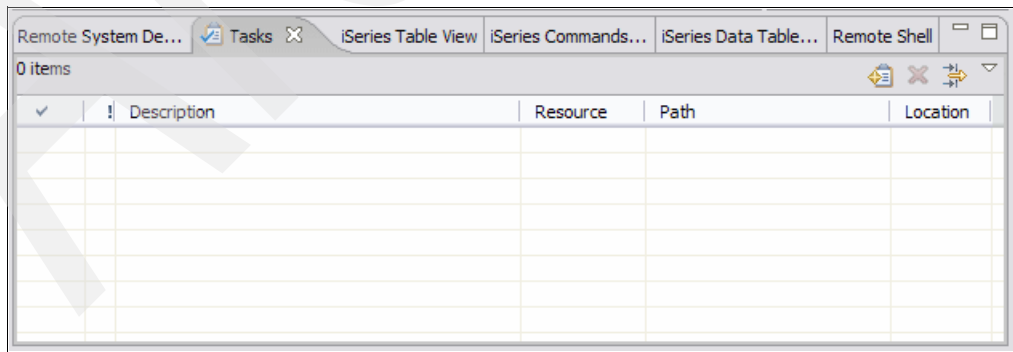


Figure 2-8 Tasks view

- ▶ **Outline view** (Figure 2-9): The Outline view displays an outline of a structured file that is currently open in the editor area. It lists structural elements. The contents of the outline view are editor-specific. In our example, an ILERPG program is opened in the Editor view.

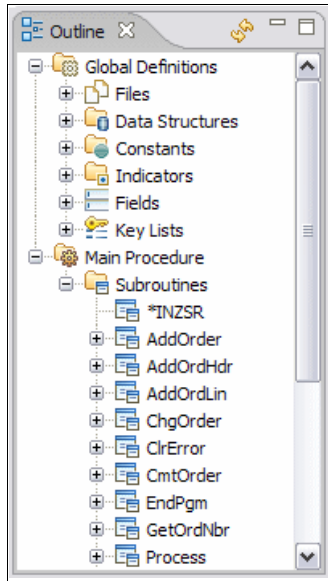


Figure 2-9 Outline view

There are several tabs next to the Remote Systems Details and Tasks views. These tabs provide an easy way to switch between opened views labeled as “Stacked views” in Figure 2-4.

If you need to open a view, which is not shown in your Workspace, select **Window** → **Show view**. Click a suitable view in the selection list (Figure 2-10) or browse for more views by clicking **Other**.

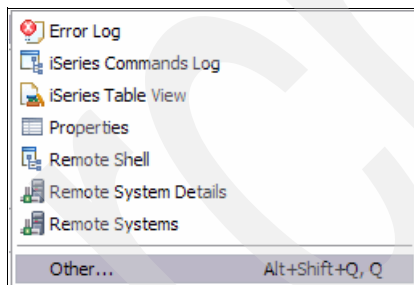


Figure 2-10 Opening a new view

2.2.5 Editors

WebSphere Studio Workbench allows you to use internal or external editors. These components are implemented as plug-ins (internal editors) or as stand-alone applications (external editors) that can be started from within the workbench.

Depending on the type of resource, there may be a specific editor to work with that resource. Different members of the WebSphere Studio Workbench family of tools have different sets of available editors, such as an XML, Page Designer, or Java editor. Each editor simplifies the work with a particular type of resource. By default, the largest portion of the workbench window is reserved for the editors (Figure 2-11).

Tabs in the editor area indicate the names of resources that are currently open for editing. An asterisk (*) indicates that an editor has unsaved changes.

The gray border at the left margin of the editor area is the *Editor information area*. It may contain icons that flag errors, warnings, or problems detected by the system. Icons also appear if you created bookmarks, added breakpoints for debugging, or recorded notes in the Tasks view.

If you hover a mouse cursor over the editor tab, the temporary source location is displayed.

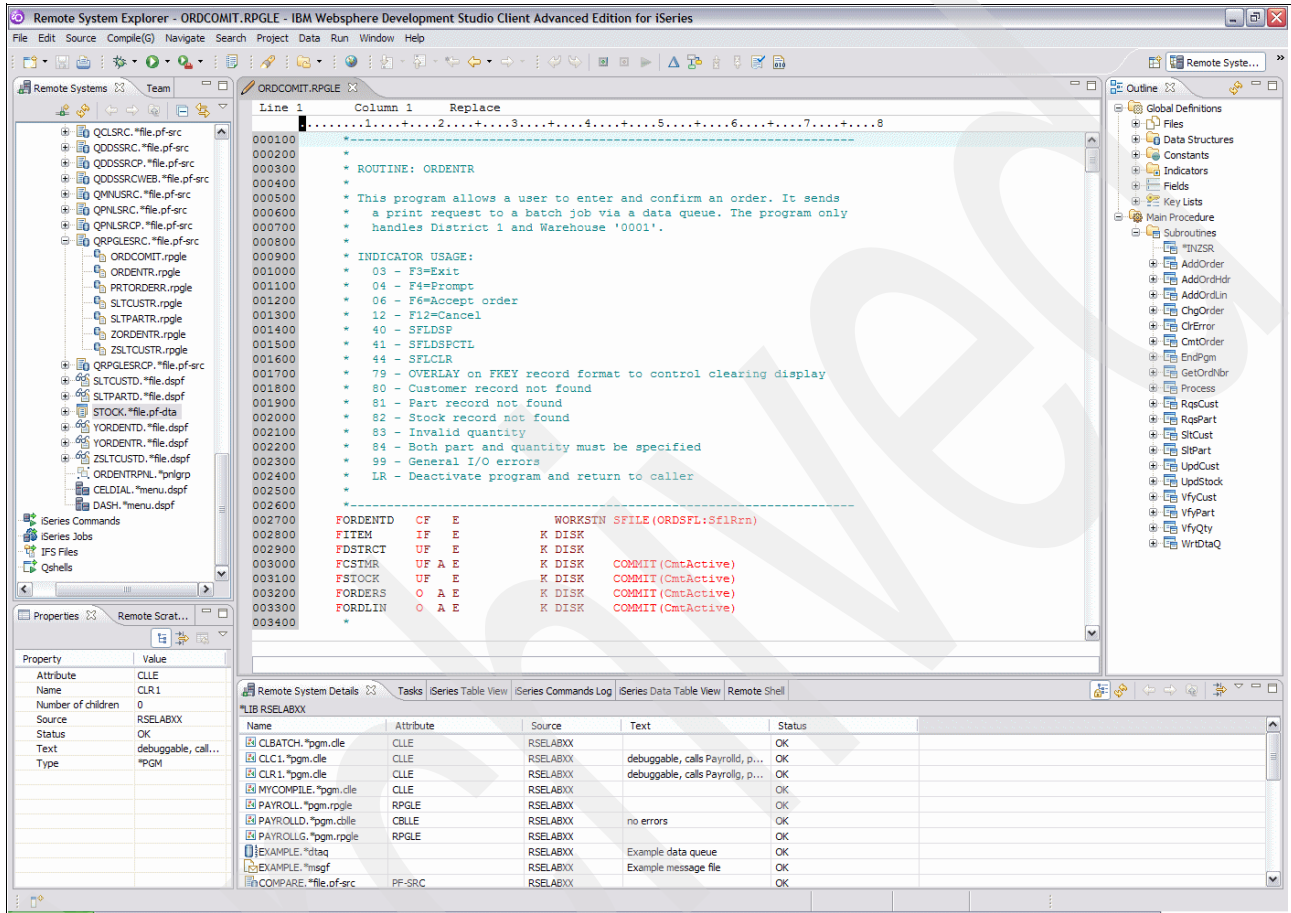


Figure 2-11 Editor area

You can customize which editor is used with what type of resource. You go to the workbench properties select **Window** → **Preferences**. On the Preferences window (Figure 2-12), expand the **General** → **Editors** tree and click **File Associations**.

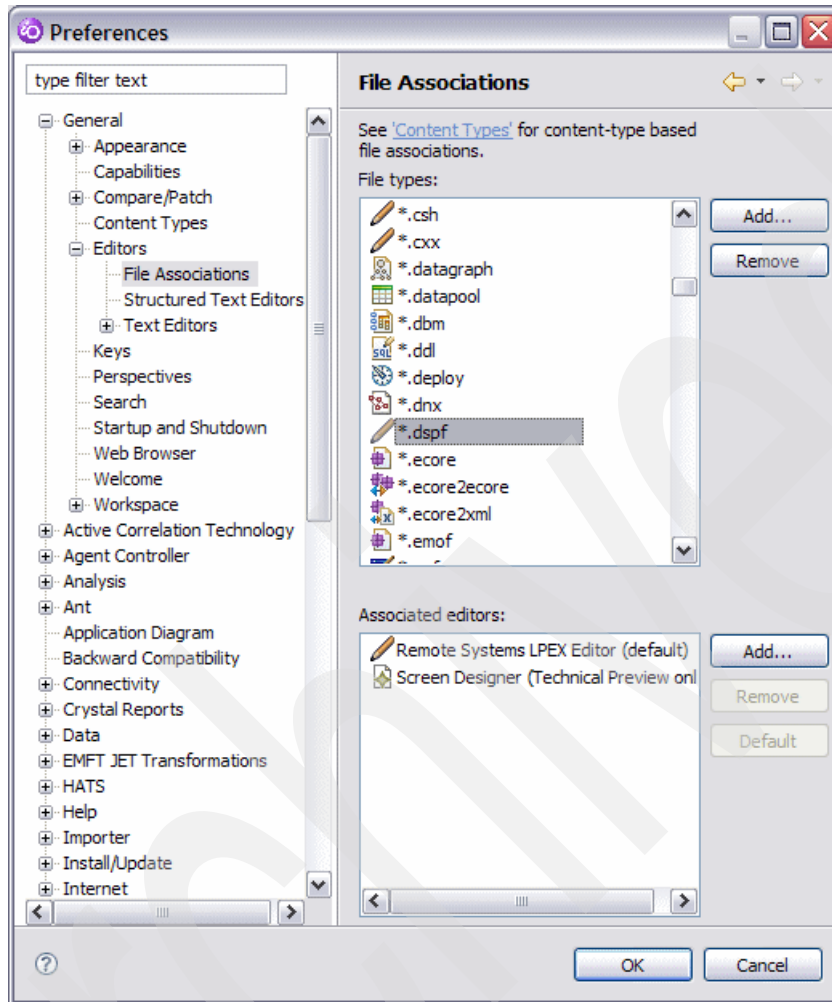


Figure 2-12 File associations

The workbench allows you to open multiple files in multiple editors. Unlike views, editors cannot be dragged outside the workbench to create new windows. However, you can tile editor sessions within the editor area to view source files side by side (Figure 2-13).

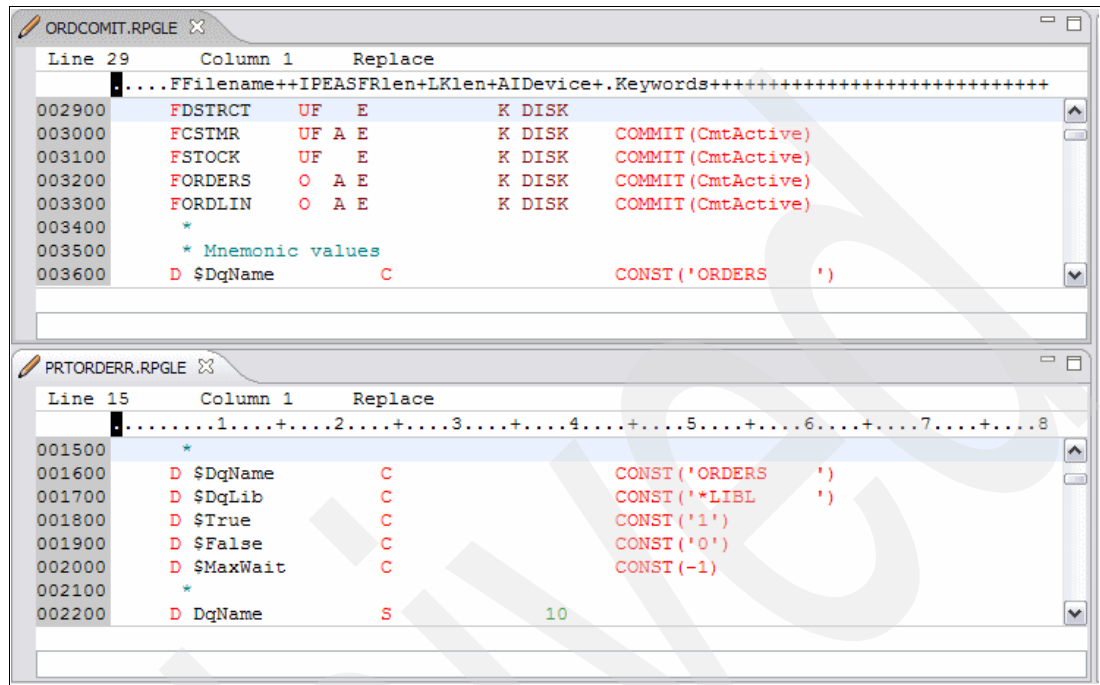


Figure 2-13 Editor view with tiled source code

To tile the editor view, open two or more files to editor view, select one of editor tabs. Holding down the left mouse button, drag the editor over the left, right, top, or bottom border of the editor area and release the mouse button.

For more information about the editors, see 8.1, “Editing source members in the RSE” on page 250.

2.2.6 Crash recovery

The workbench periodically saves a snapshot to reduce the risk of losing data due to crashes:

- ▶ Saved data is never lost since it is written to disk immediately (on save).
- ▶ Unsaved data in open editors may be lost, depending on the editor implementation.
- ▶ Bookmarks and tasks may be lost.
- ▶ If a crash occurs during Concurrent Versions System (CVS) synchronization, the workbench may be out of sync. You can check by performing the synchronize operation again.
- ▶ Previously-created projects are never lost.

To set the save interval, select **Window** → **Preferences**. On the Preferences window (Figure 2-14), click **General** → **Workspace**. Type a new value in the Workspace save interval field and click **OK**.

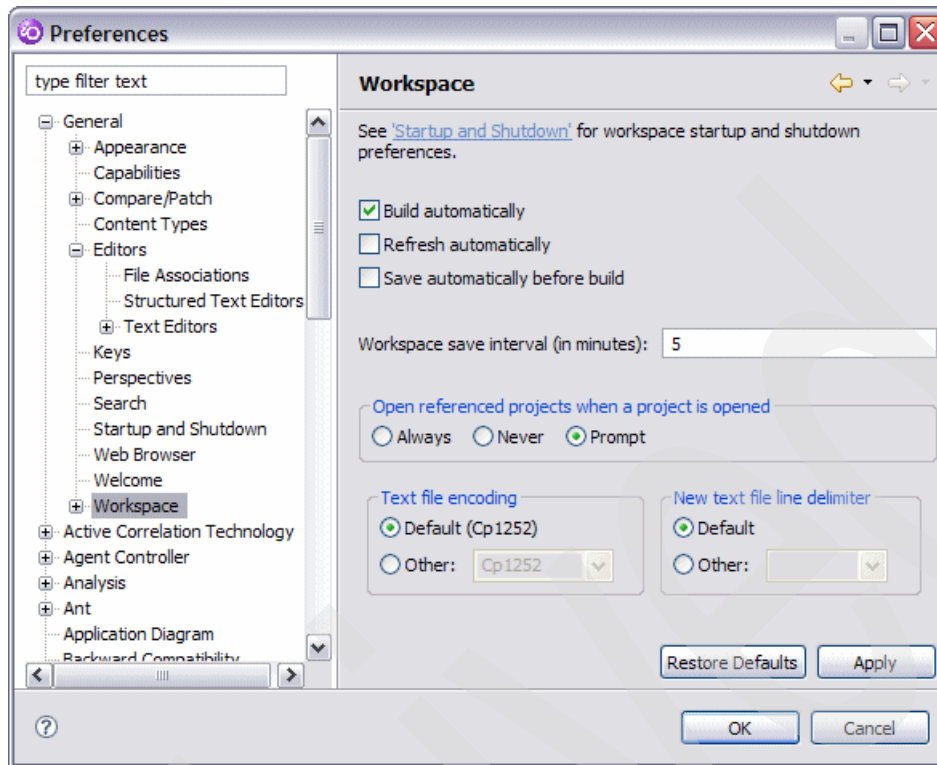


Figure 2-14 Workspace save interval

2.3 Development Studio Client features

Earlier we mentioned that Eclipse Software Development Kit (SDK) only includes the base for developing a full-reach IDE. WebSphere Workbench adds several more plug-ins, or extensions, to the Eclipse platform. WDSC AE, which is based on Rational Application Developer, enriches the workbench with an array of advanced plug-ins. This makes Development Studio Client a premier platform for developing applications for the System i server.

2.3.1 Tools

The main tools of WebSphere Development Studio Client Advanced Edition for iSeries are:

- ▶ The workbench-based integrated development environment
- ▶ Remote System Explorer and System i development tools
- ▶ System i Web development tools
- ▶ System i Java development tools
- ▶ Struts environment support
- ▶ Web Services development tools
- ▶ XML development tools
- ▶ Server development tools
- ▶ Database development tools
- ▶ The IBM WebFacing Tool
- ▶ VisualAge RPG
- ▶ CODE
- ▶ Integrated System i debugger

2.3.2 Common features

We found several features that are useful during the development of applications. They include:

- ▶ Importing and exporting files by dragging and dropping
- ▶ Copying, renaming, and moving
- ▶ Searching
- ▶ Bookmarks
- ▶ Rearranging views and editors
- ▶ Fast views
- ▶ Perspectives
- ▶ Builders
- ▶ Local history
- ▶ Team environment
- ▶ Compare editor

Importing and exporting files by dragging and dropping

You can use your operating system's file system explorer to import or export a copy of a folder or file from the workbench to the file system.

Note: You must drag the resource or resources to the exact location in the Navigator view hierarchy where you want the resources to reside. You cannot simply drag and drop them onto a blank area in the Navigator view.

Copying, renaming, and moving

You can copy, rename, and move workbench resources using menu options in the main Perspective view (Figure 2-15).

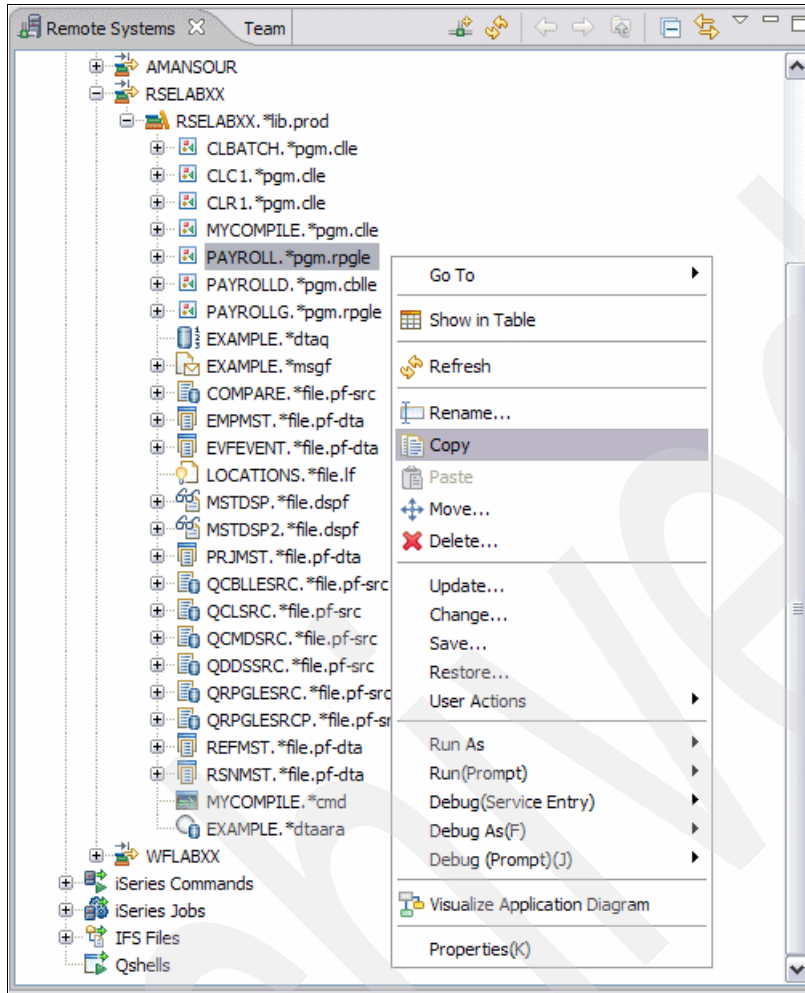


Figure 2-15 Copying, moving, and renaming workbench resources

Searching

You can search for text strings and files from the workbench. Click the **Search** (🔍) icon on the toolbar to open the Search window (Figure 2-16).

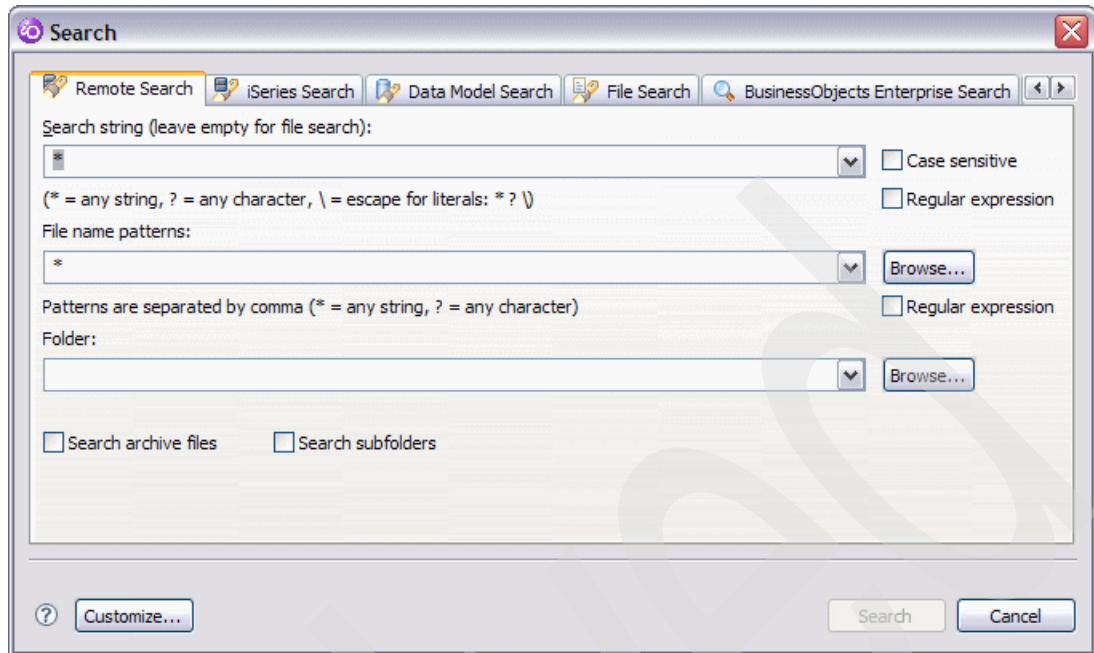


Figure 2-16 Search window

Bookmarks

Bookmarks are a simple way to navigate to resources that you frequently use. To create a bookmark, follow these steps:

1. Right-click the **Editor information area** (Figure 2-13 on page 28).
2. Select **Add Bookmark**. A bookmark marker is added to the Editor view.
3. Open the Bookmark view. Select **Window** → **Show view** → **Bookmarks**. Then you see the Bookmarks window as shown in Figure 2-17.
4. Double-click the bookmark item and the source code opens.

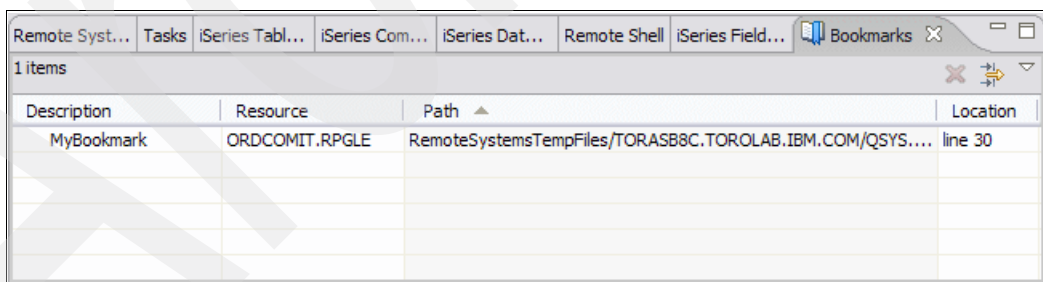


Figure 2-17 Bookmark view

Rearranging views and editors

You can also rearrange the location of the views in the workbench window. You can drag and drop any view in a workbench window. Simply watch the cursor shape as you move the view in the perspective:

- ▶ When the cursor changes its shape to a black arrow (➡), the direction of the arrow shows the location to where a view will be dropped.

Note: The cursor changes to a black arrow only when you move it over the border of the workbench window.

- ▶ When the cursor looks like a window (☐), you can drop the view, and it becomes a floating view (like a separate window).
- ▶ When the cursor looks like stacked folders (📁), you can drop the view and it is stacked with the views underneath the cursor.
- ▶ When the cursor looks like a circle with a line through it (⊘), you *cannot* drop the view at the current position.

After you are satisfied with this arrangement, you can save the perspective by selecting from the workbench toolbar menu **Window** → **Save Perspective As**.

If you are not satisfied with the layout of the workbench, you can always reset it by selecting **Window** → **Reset Perspective**.

Fast views are hidden views, which can be quickly made visible. A good reason for using fast views is to maximize your editor view:

1. Click the title bar of a view (in our example, Remote Systems view). Drag it to the shortcut bar at the left bottom of the window (see Figure 2-18).
2. Open the source file, right-click the title bar in the Editor view and select **Maximize**. Now you have a large area for your editor and can quickly switch views (Figure 2-18).

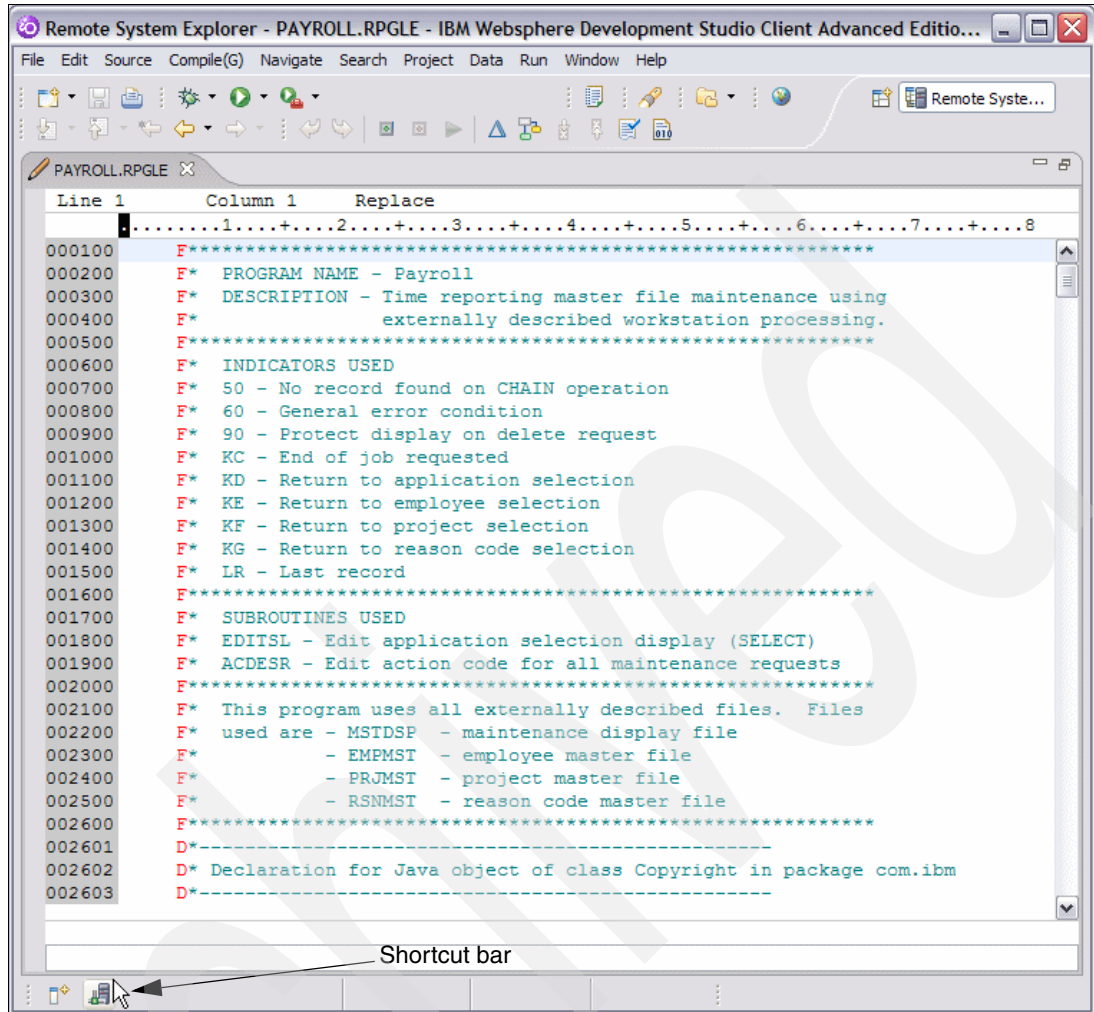



Figure 2-18 Maximize Editor

3. Click the **Remote Systems view** () button on the left bottom side of the window. Your Remote Systems view (Figure 2-19) opens.

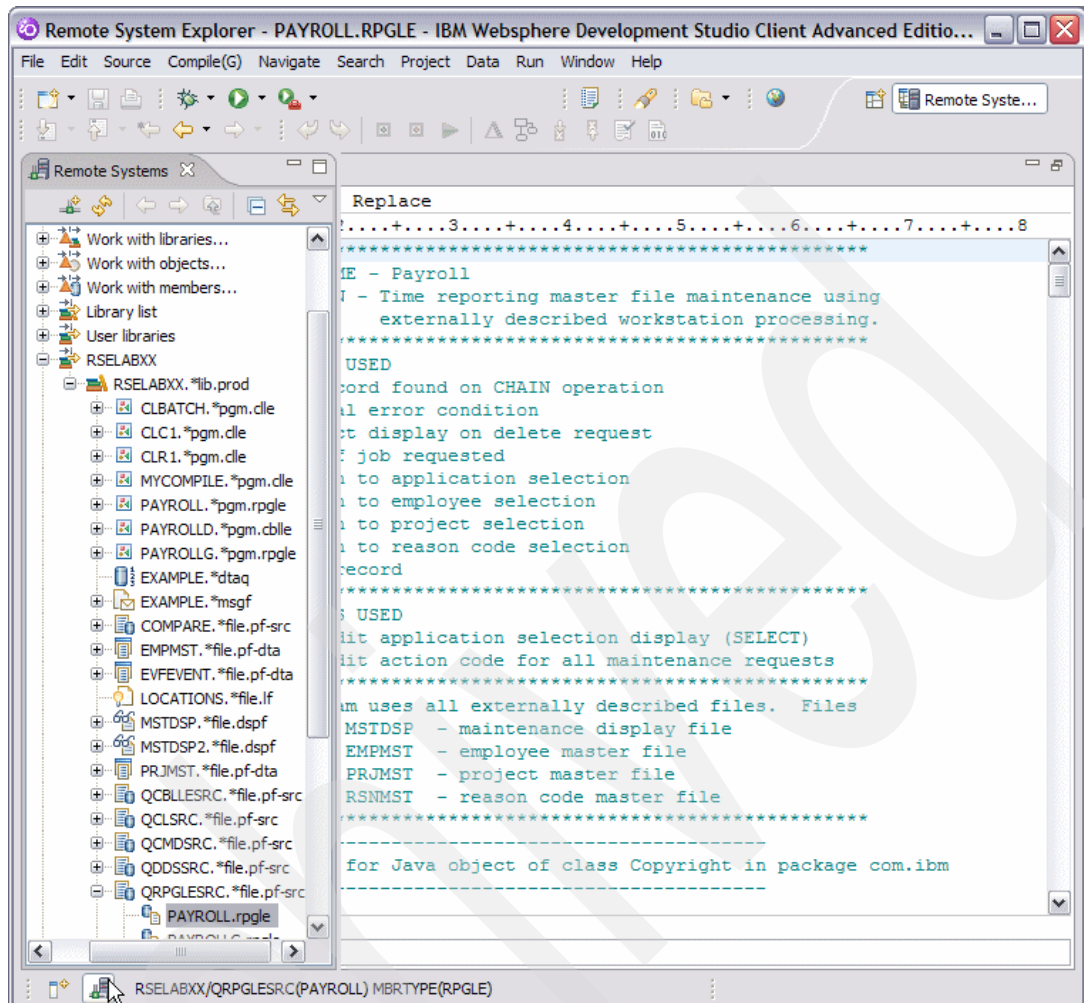


Figure 2-19 Remote Systems fast view

4. Click the same button again, and the Remote Systems view closes.

Builders

Builders represent a utility that takes the existing resource and creates a new resource or updates the existing resource. For example, Java builder produces a compiled version of a Java file. Depending on the selection, Java builder can compile a single file, a package, or an entire project. A nice feature of this builder (as with VisualAge for Java) is the incremental build, in which a Java file is compiled on every modification. If there is any error during the build, it is displayed in the Task view and the line in error is marked with a small red icon.

Local history

A local history of a file is maintained by the tool. The default settings allow you to save the last 50 versions of a file for at least seven days. If you make a mistake and want to go back to an older version of a file, you don't need to have a team environment installed. You can use your local history feature.

To change the default setting for the local history feature, select **Window** → **Preferences**. On the Preferences window, expand the **General** → **Workspace** tree and select **Local History**.

To use this feature, right-click any file in the Navigator view and select **Replace With** → **Local History**. The workbench shows you the list of available versions and the compare utility.

Team environment

Team environment is based on the support of a repository. The workbench comes with built-in CVS support. You can download and install the CVS server from:

<http://www.cvshome.org/>

To learn how to use Team environment, refer to WebSphere Development Studio Client Advanced Edition for iSeries Help: from the workbench menu, select **Help** → **Help Contents** → **Rational Developer product documentation** → **Introduction to Rational Application product documentation** → **Get Started with the workbench** → **Concepts** → **Team programming with CVS**.

CVS controls access to a resource by multiple users.

Compare editor

The Compare editor provides a visual way to compare two files. They can be part of any project, local history, or repository. Figure 2-20 shows the Compare editor for two Java files in two different projects.

To compare two files from the project tree, follow these steps:

1. Select the first file.
2. Press the Ctrl key and click the second file.
3. Right-click any of the highlighted files and select **Compare With** → **Each Other**.

The Compare editor is quite sophisticated. As you cycle through the changes, it highlights the lines that are different and the part of the line (a method name, variable, and so on) that is different. The editor's control buttons are located above the file view area.

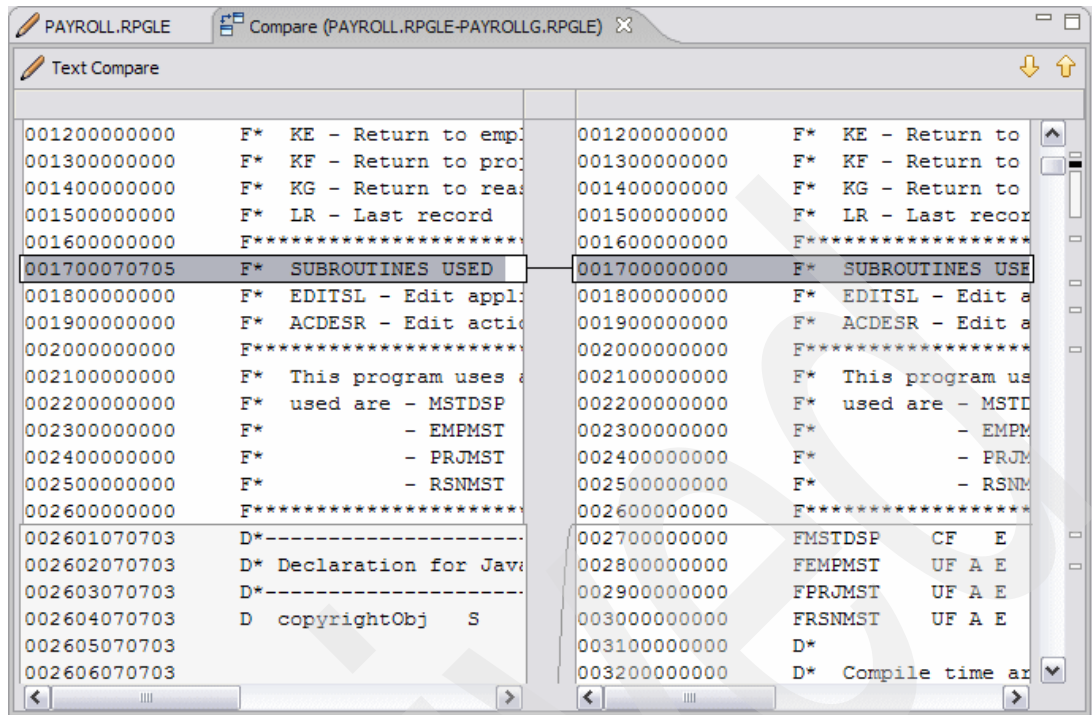


Figure 2-20 Compare editor

2.4 Online help

The Help system lets you browse, search, bookmark, and print help documentation. The documentation is organized into sets of information that are analogous to books. The help system also supplies a text search capability for finding the information you need by search phrase or keyword, and context-sensitive help for finding information to describe the particular function you are working with.

You can interact with the help system in the workbench using the **Help view** or in the separate **Help window**. The view and window provide the same information but in different ways.

Help view

The Help view provides help inside the workbench. You can open the view from the main menu by selecting **Help** → **Dynamic Help** or **Help** → **Search**. The view will open showing the Related Topics or Search page, respectively, as shown in Figure 2-21. You can use links at the bottom of the help view to turn to other pages.

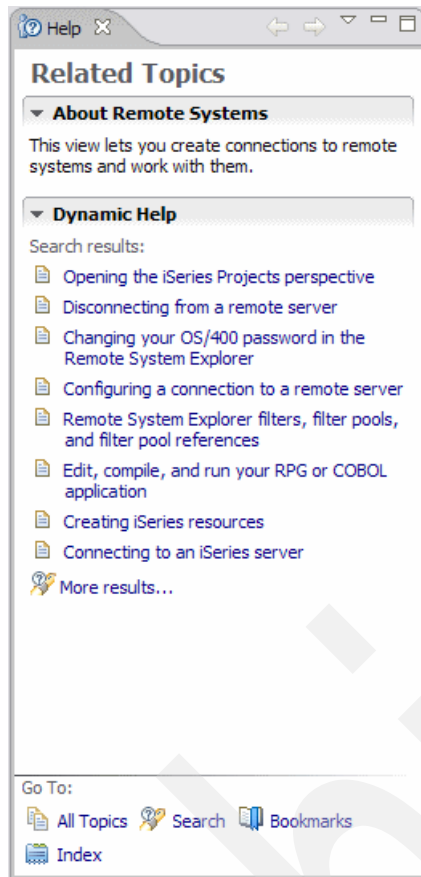


Figure 2-21 Dynamic help

Help window

The Help window provides the same content as the Help view, but in a separate window instead of in a view. You can open the window from the main menu by selecting **Help** → **Help Contents**. The first view shown in the window is called Contents, as shown in Figure 2-22. This view displays the table of contents for the product documentation. Click one of the links to expand the navigation tree for a set of documentation.

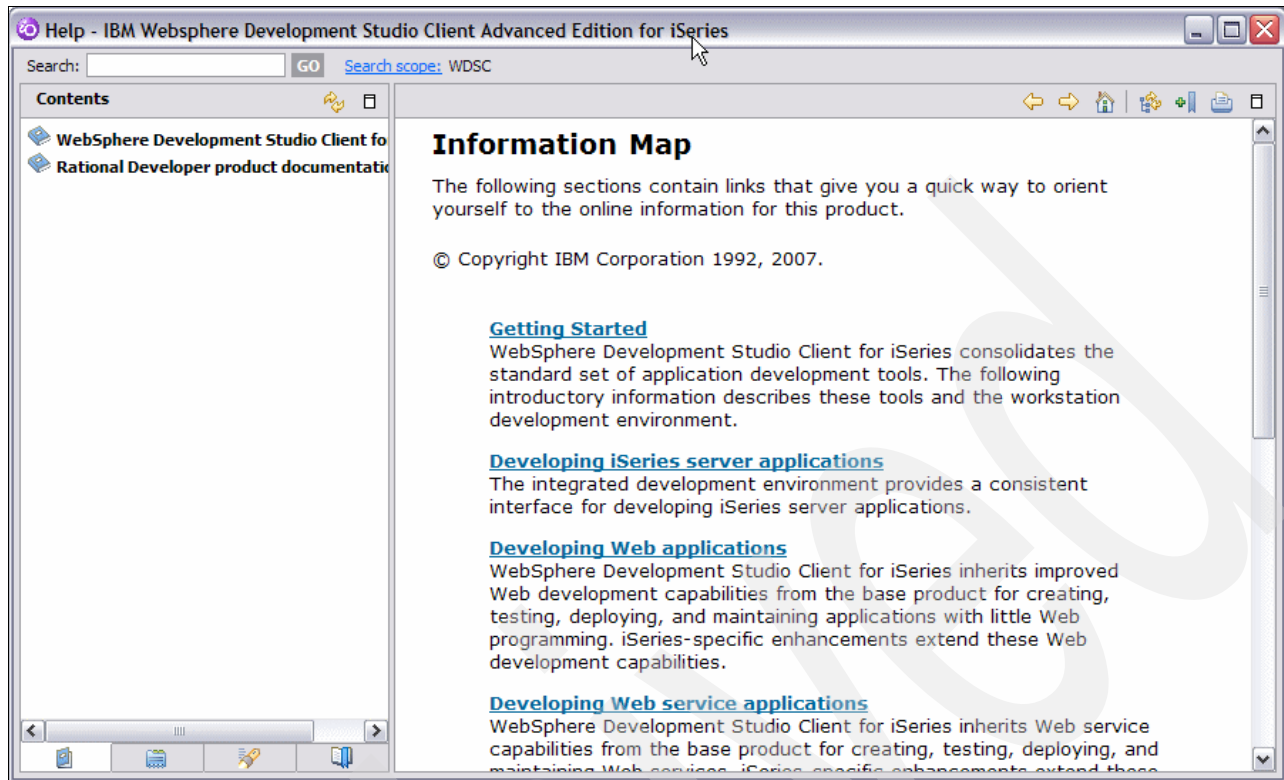


Figure 2-22 Help window

Context-sensitive help

If you are working through a task and encounter a part of the interface that you do not understand, you can summon context-sensitive help. By default, this will display the Help view and give you some specific information about the view/editor/dialog you are using, and possibly some links to topics for further help.

Context-sensitive help can be accessed by bringing focus to the interface part in question by clicking on it or using the Tab key, and then pressing F1. Alternatively, in dialogs you can achieve the same result by pressing the help button (?) in the dialog's button bar, as shown in Figure 2-23.

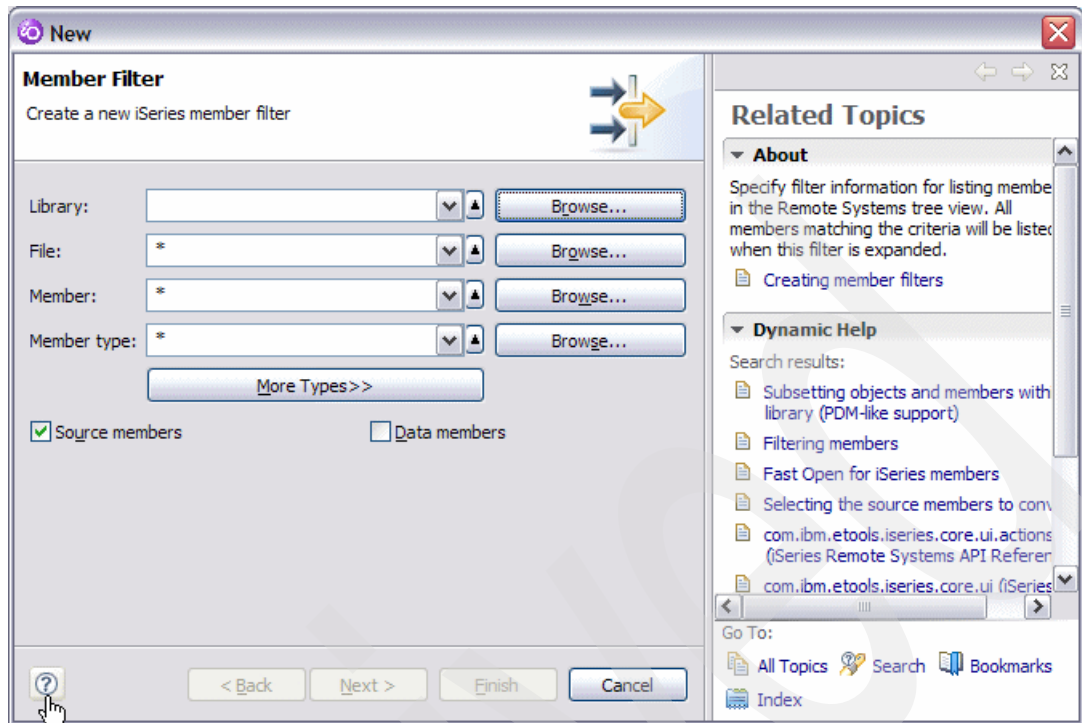


Figure 2-23 Context sensitive help

Searching help

The help system includes a search engine that can run simple or complex queries on the documentation to help you find the information you are looking for.

To search help:

1. From the main menu, select **Help** → **Search**.
2. Type in the word or phrase for which you want to search.
3. Click **GO** or press Enter. The list of results will be displayed below.
4. To view the content of a topic in the list of results, click on it.

You can also search from the Help window using the Search field at the top of the window as shown in Figure 2-24.

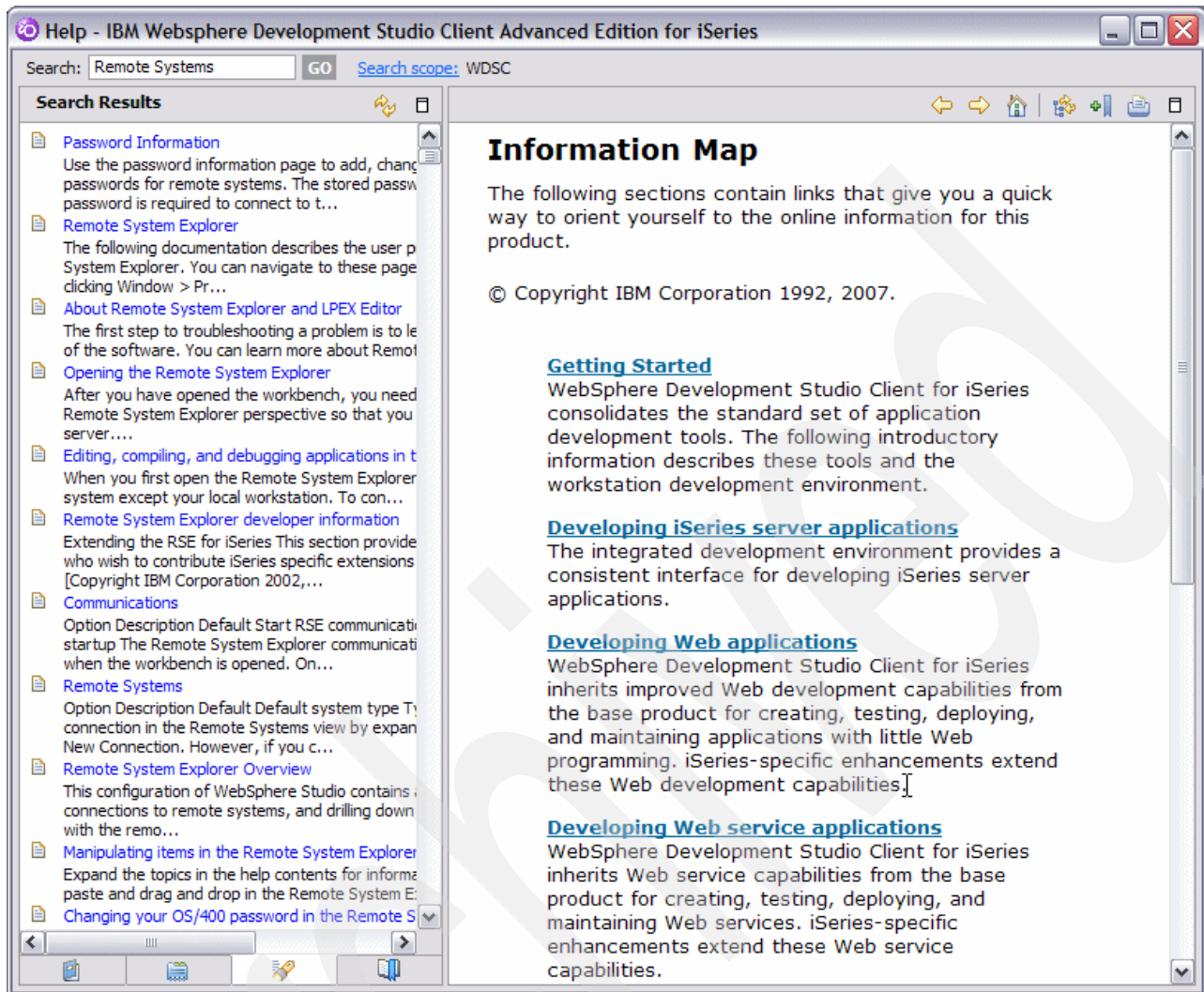


Figure 2-24 Help Window search

Archived



WebFacing, HATS, and WDHT

In this chapter we introduce the IBM Web enablers included in WDSC V7.0. You will learn how the WebFacing tool and Host Access Transformation Services (HATS) can greatly improve the productivity and flexibility of your applications by giving them a new Web interface. A list of enhancements made to each since the last edition of this book is also included at the end of the chapter.

3.1 Why Web-enable an application

Superior software applications are one of the factors that determine competitive advantages among businesses. During the process of improving your software, you may consider writing new code yourself, purchasing an application from a solutions vendor, or leveraging your existing applications through modernization.

Some of the benefits of making use of existing applications is that it saves time, reduces costs, and lowers risks by using existing skills, applications, and data to deliver new e-business solutions. Another major consideration, besides lessening cost and risk, is the performance, reliability and scalability that are preserved when existing applications are modernized instead of writing new ones.

There are several reasons why you may be interested in Web-enabling your 5250 business applications.

- ▶ A Web interface may provide a more familiar interaction between the user and the application than a green screen, particularly for newer users who are more comfortable with the Internet and point-and-click interface in general.
- ▶ Richer customization capabilities become available for Web applications than were possible as native 5250 programs, for example, the use of graphics and essentially unlimited color palettes, increased screen real-estate, and drag-and-drop functionality. The potential when using Web technologies is becoming increasingly unbounded.
- ▶ Web applications can be accessed wherever a user has access to a browser, which is usually far more common than having a 5250 emulator installed, or using a System i terminal, which can be very costly.
- ▶ Other reasons, such as business or customer requirements, may be why you are considering Web enablement. Business pressure may come from outside the IT department, from marketing initiatives, for example. If you are a solution vendor, your customers may be apprehensive about pure i5/OS solutions, or perhaps they need a Web application to be integrated with existing applications. They may also choose Web solutions out of a need to minimize the learning curve for employees and end users of their application, all of which are compelling and legitimate concerns.

All the reasons outlined above, when addressed, can lead to improved usability, reduced training costs, increased productivity, and the creation of opportunities to reach new clients—all convincing incentives to Web-enable.

3.1.1 What is blocking Web enablement of green screens

Web-enabling a green screen application can pose several real challenges. One of the highest hurdles is that complex legacy applications can be very difficult to rewrite as Web applications, and you may not immediately have the skills or resources for such a task. i5/OS developers would need to learn Web technologies, and conversely, Web developers would need to learn about the i5/OS. Both journeys can be long and arduous, and along steep learning curves. The costs of training employees, of new hardware and software, and of schedule delays are all realistic concerns voiced by those considering Web enablement.

3.1.2 Web-enabling tools in WDSC

IBM provides solutions to help customers quickly Web-enable their green screens. In this book, we look at two IBM solutions, WebFacing Tool and Host Access Transformation Services (HATS), available in WDSC. Both have been developed to make the Web enablement process as simple and seamless as possible, and are supplemented by

extensive documentation, tutorials, and cheat sheets to help you get up to speed as soon as possible.

Note: Both WebFacing and HATS have been greatly improved and enhanced over the years. If, for whatever reason, you hesitate to use either, read on.

WebFacing Tool

The IBM WebFacing Tool generates Web interfaces to 5250 applications. It converts data description specifications (DDS) and user interface manager (UIM) source into the Web user interface at development time, generating Web representations of DDS constructs using JavaServer Pages (JSP) and XML.

Since the conversion output is Web pages, you can refine them without limit. Web Settings support in WebFacing allows you to insert predefined customizations to DDS fields to tailor how your programs will look and function when accessed through a Web browser. You can use the project conversion, runtime and style properties to change the conversion settings, runtime behavior, and look and feel of your entire WebFacing applications. You may also edit the conversion output directly. Advanced features, such as Portal support and single signon, allow seamless integration with other applications.

Figure 3-1 shows a sample “before and after” of a green screen that has been WebFaced.

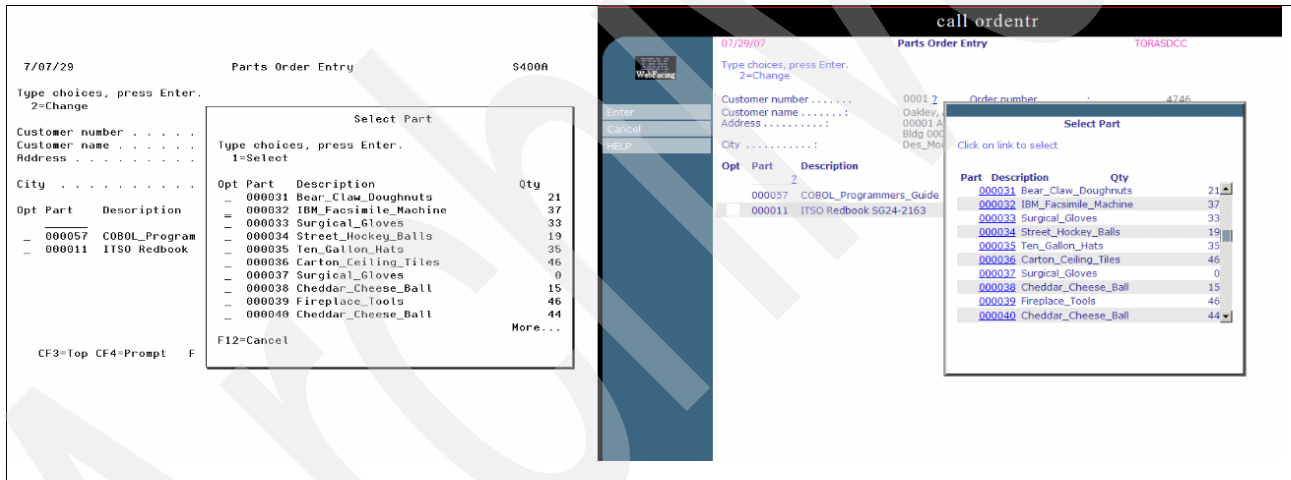


Figure 3-1 5250 screen and its WebFacing representation

HATS

HATS includes a dynamic rule-based transformation engine that quickly and easily generates a Web interface from a 5250 data stream at runtime. This enables applications to be migrated to the Web without accessing or modifying the original source code.

The resulting Web interface can be customized with the HATS Toolkit. The macro support of HATS enables you to provide programmed navigation through multiple host screens. It also allows you to combine information from multiple screens or from multiple applications into a single Web page. Other advanced features, such as Integration Objects, the use of business logic, and Portal support, provide integration capabilities with other applications.

Figure 3-2 shows a sample “before and after” of a green screen that has been transformed by HATS.

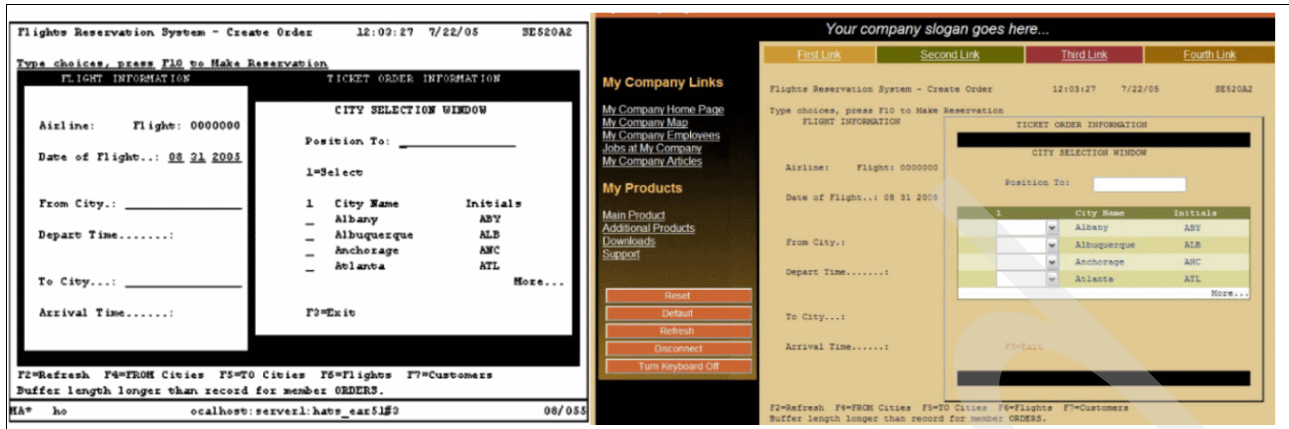


Figure 3-2 5250 screen and its HATS transformation

3.2 Web enablement using WebFacing

This section gives a basic introduction of the IBM WebFacing Tool. It shows the main features and benefits WebFacing provides. It also highlights the new features in V7.0. More detailed information about WebFacing is included in the next chapter.

3.2.1 Introduction to WebFacing

The IBM WebFacing Tool converts existing 5250 interfaces where source is available to browser-based graphical user interfaces. With little or no modification to your original System i applications, you can extend the use of your programs to the Internet or an intranet.

Figure 3-3 shows the high-level tasks and files involved in Web-enabling your applications.

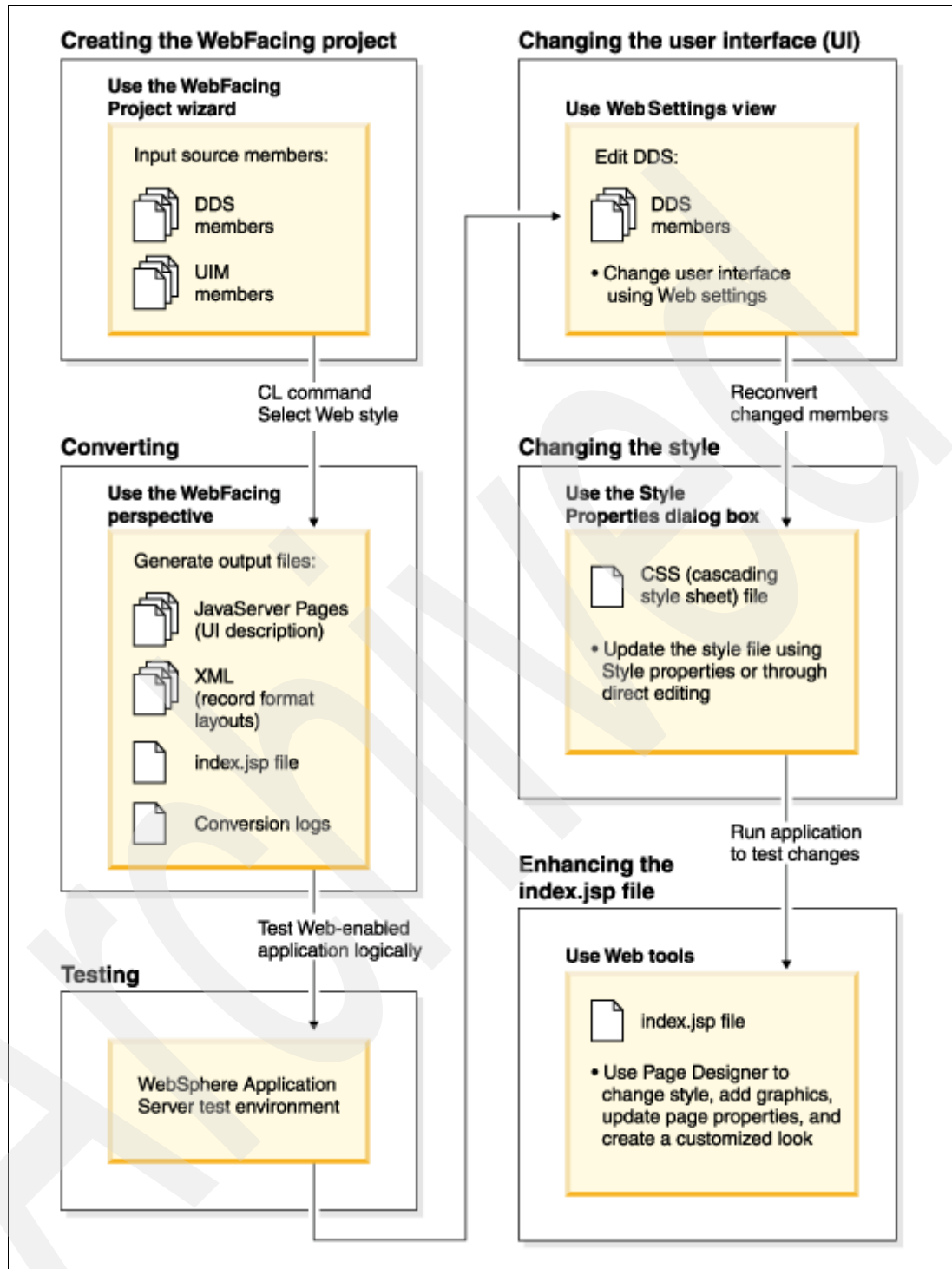


Figure 3-3 WebFacing development stages

The tool has user-friendly wizards that facilitate selecting your original application's DDS and UIM source members, and converting the source. The conversion creates JSPs and XML files that substitute for your original DDS and UIM source code and make Web access possible. After conversion, use the built-in WebSphere Application Server test environment that WDSC provides to test your converted Web application. You can change your application's user interface by editing the DDS members using the Web Settings view in the

IDE, Screen Designer (Advanced Edition), or optionally CODE Designer, then reconvert the changed members when you are done. Further customize the style using the Style properties dialog or editing CSS files directly. You can also enhance the generated index.jsp file using Page Designer to change the style, add graphics, update page properties, and create a more customized look for your Web-enabled application.

You can continue to refine your new Web pages and retest them until you have a design that you like. Finally you can deploy the new browser-based interface to your program as a WebSphere application to a production server.

Figure 3-4 shows the runtime flow when a user invokes a converted program through a Web browser interface.

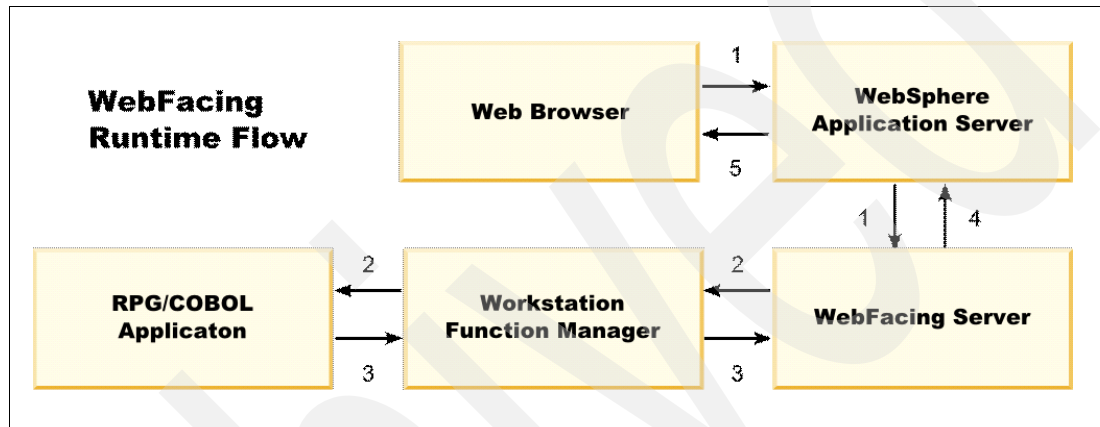


Figure 3-4 WebFacing runtime flow

1. A user invokes a converted program through a Web browser interface.
2. WebFacing server on System i passes the request to Workstation function manager (WSFM), part of the i5/OS, and then WSFM passes the data buffer to the program.
3. When the response comes back, because the job was started by the WebFacing server, WSFM knows that this is a WebFacing request and passes the data and control to the WebFacing server.
4. Control returns to the WebFacing Runtime Servlet that runs in WebSphere Application Server.
5. The WebFacing Runtime Servlet locates the appropriate JSPs and Java beans. It tells WebSphere Application Server to return the JSP back to the browser. The JSP is processed, and the resulting HTML is returned to the browser.

Note that no 5250 data stream is generated in this flow.

3.2.2 WebFacing features and benefits

WebFacing provides a development time tool to convert display files into a Web user interface and generates a Web project of your application. Since the newly converted WebFacing application is a standard Java 2 Platform, Enterprise Edition (J2EE) Web project using common Web technologies, such as JSP, XML, CSS, and JavaScript™, it is highly customizable.

Figure 3-5 shows a sample “before and after” of an out-of-the-box WebFaced page and one that has been customized. Notice that the page content has been modified to be more Web friendly, such as allowing you to click on a link to make a selection. The text “FKey12=Cancel”

has been removed and you can click the X on the top right corner to cancel the operation instead.

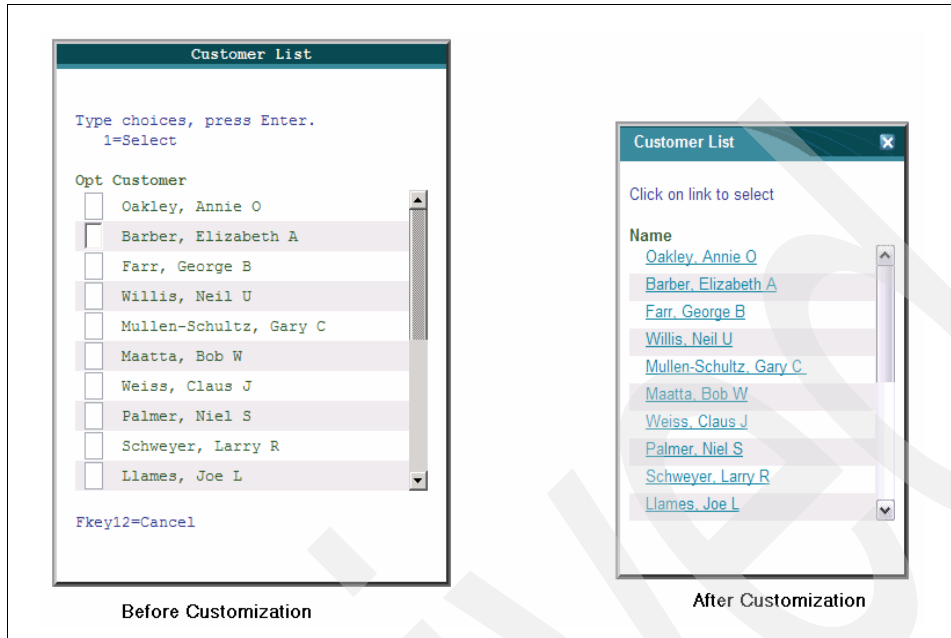


Figure 3-5 WebFacing customization

Your WebFacing application does not require online transaction processing (OLTP) capacity. It can be run entirely in batch mode using i5/OS Standard Edition. System screens or non-WebFaced parts of your application are also accessed in batch mode by dynamic transformation of the screens for the Web.

WebFacing also provides cluster support on a WebSphere Application Server to meet the high availability needs of your application, allowing for better response times, scalability, and failover safety. You can also configure additional interactive subsystems on i5/OS for your System i as the number of users of your WebFacing application grows.

The Advanced edition of WDSC V7.0 includes several additional features, including the following:

- Portal support

The WebFacing Tool generates a Struts-based portlet application that can be deployed to a WebSphere Portal server and added to portal pages. This allows multiple applications to run in the same browser window. The portlet application can also take advantage of the features available to a WebFacing Web project as illustrated in Figure 3-6.

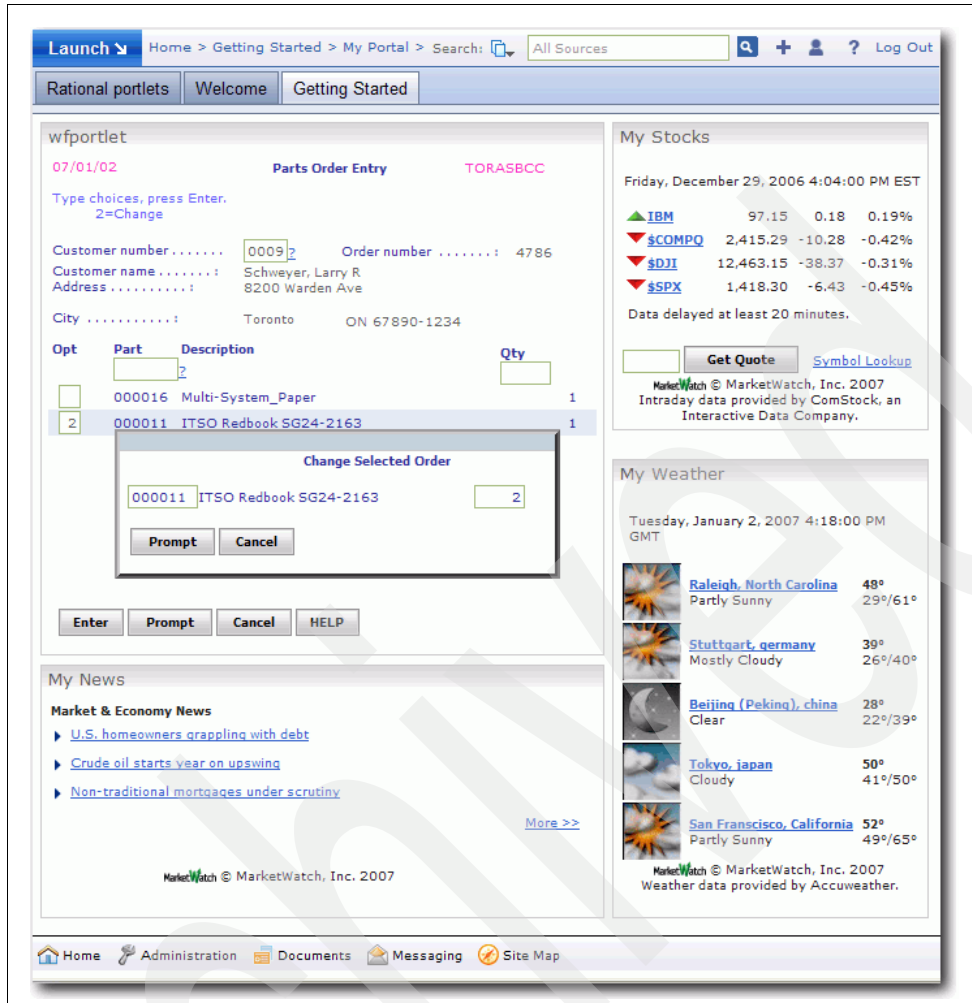


Figure 3-6 WebFacing portlet

- ▶ Single signon

This feature enables users to access more than one application and multiple platforms using one user ID and password. For example, you can integrate secured WebFacing and Web Tools applications which a user can access without requiring separate IDs and passwords to be entered for each. This support is very handy for WebFacing portlets.

- ▶ Cascading Style Sheets - Positioning (CSS-P)

This feature allows the DDS fields on the Web-enabled interface to be repositioned, unconstrained by the boundaries of the original 5250 area, and offers an almost unlimited ability to customize the Web-enabled interface.

- ▶ Command key action

You can define customized command keys in your WebFacing application to invoke a Struts-based WebFacing service, a JSP, or a static html page. These command keys override the existing function keys and can connect to a Web application URI or an external URL. You might have already defined a JSP or service using a Web diagram, in which case you can use command key actions to invoke that JSP or service.

New in V7.0

Let us now take a look at the new features and enhancements introduced by WebFacing V7.0.

WebFacing and HATS interoperability

A new type of WebFacing project, HATS/WebFacing Enabled project, is now available. You can link it with a HATS project and can take advantage of features unique to each (Figure 3-7). You will learn more about HATS in the next section. This Linked HATS/WebFacing application operates seamlessly and provides a unified, Web-enabled interface, which can be deployed to a WebSphere Application Server.

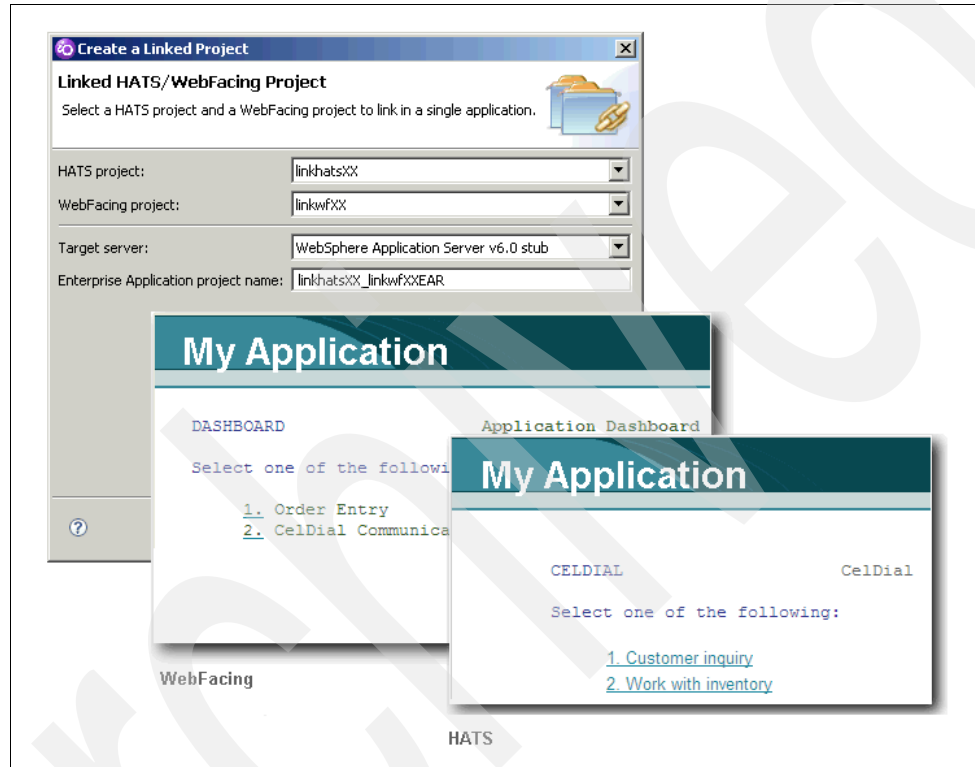


Figure 3-7 Linked HATS/WebFacing project

The topic of interoperability will be covered in greater detail later on in this chapter.

New styles

Two new styles, swirl and financialGreyRedtones, have been added to assist in developing mixed WebFacing and HATS applications (Figure 3-8). Note that these two new styles are also available to WebFacing Web Projects.

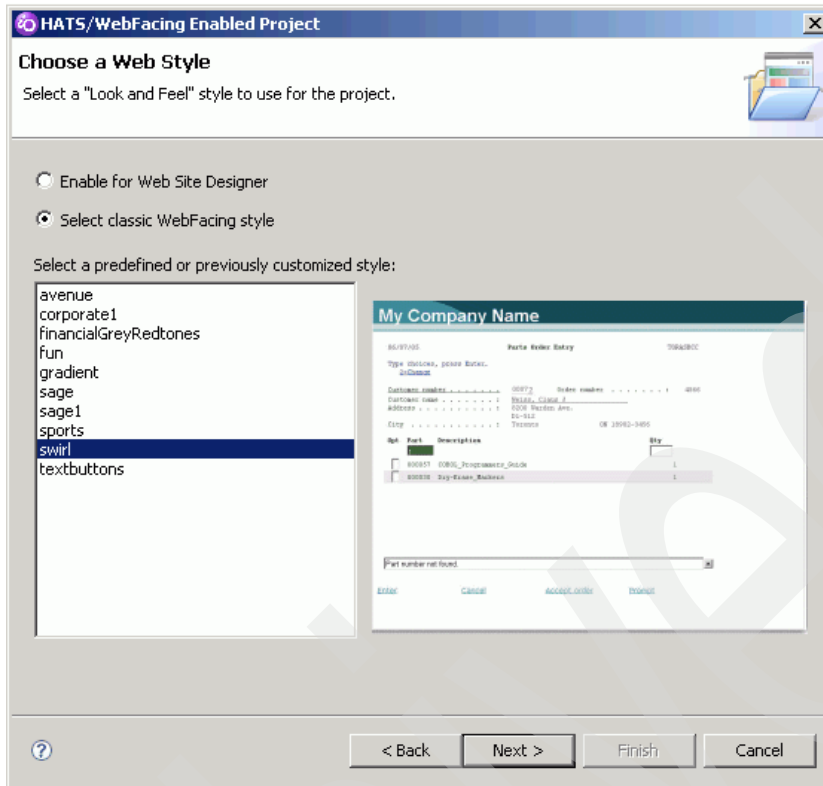


Figure 3-8 New WebFacing styles

Cheat sheets

WDCS 7.0 includes a collection of cheat sheets that act like short interactive tutorials to help users get started with common tasks. There are two WebFacing cheat sheets that detail the process of creating and testing a WebFacing Web project or portlet project (Figure 3-9). You can access System i platform cheat sheets from the First Steps page on the WDCS Welcome screen or from the Help toolbar menu.

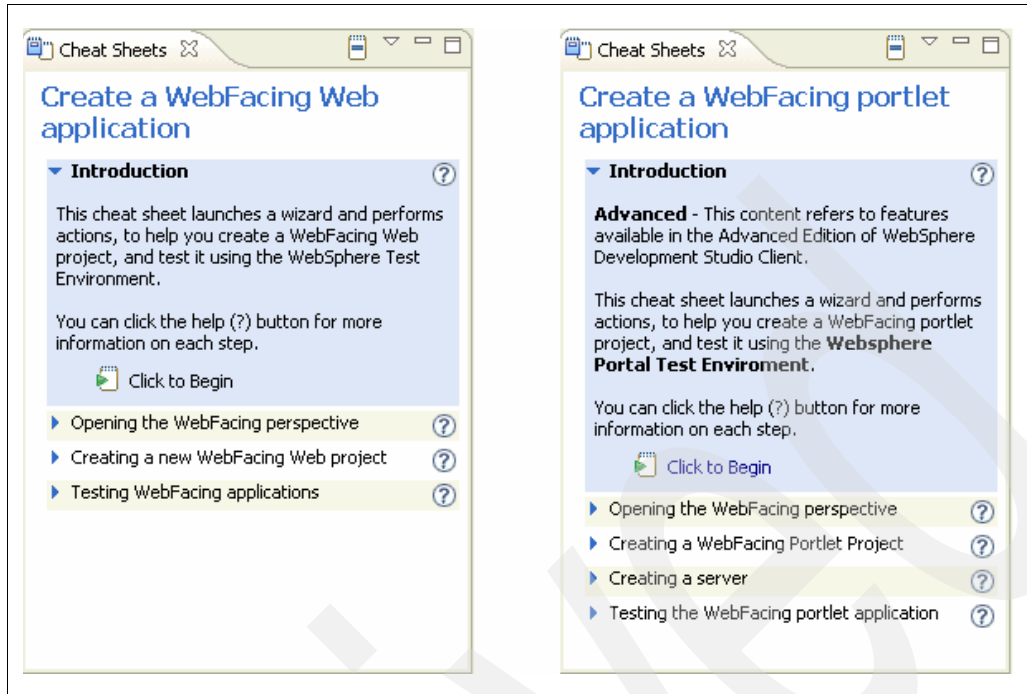


Figure 3-9 WebFacing Web and Portlet cheat sheets

Help enhancement

Several other help improvements complement the cheat sheets. One is that information is better integrated into the interface. When a user presses F1, instead of an infopop appearing above a field as in previous releases, a new side panel opens with the available information for multiple fields on the page, as well as links to related resources in the product documentation and on the Web (Figure 3-10). If you preferred infopops, there is a preference option you can set to continue using them.

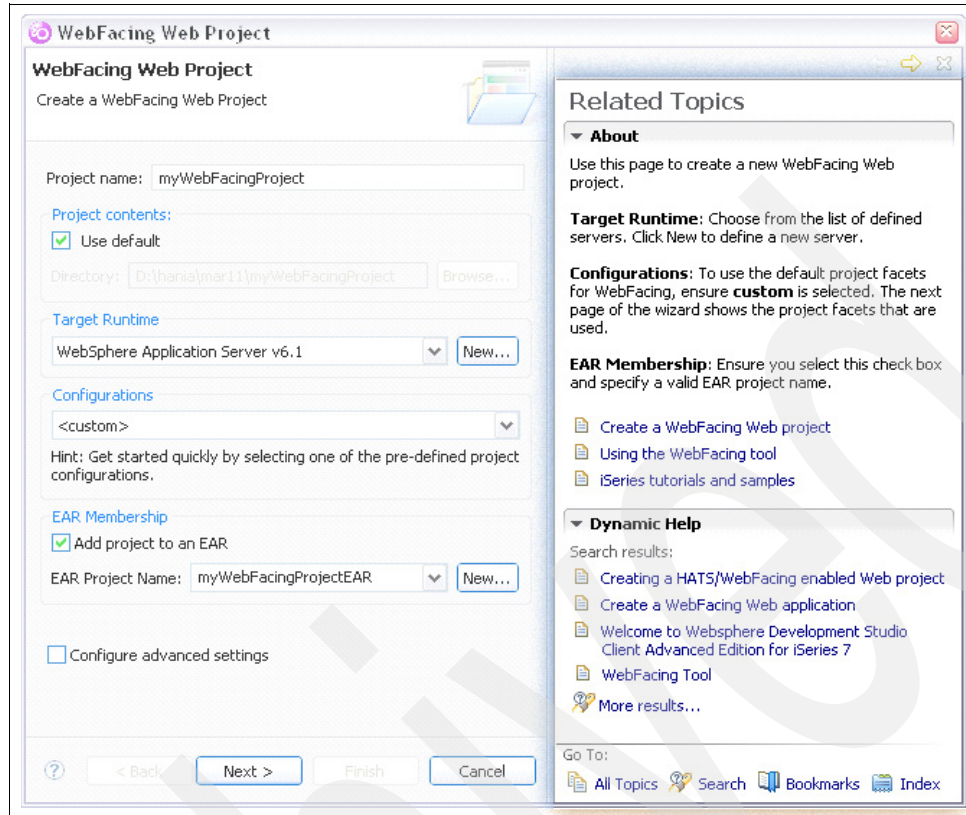


Figure 3-10 Dynamic Help panel

Web and portlet project wizards redesigned

The WebFacing project wizards (Figure 3-11) have been rebuilt over the existing J2EE project wizards, with several resulting benefits. One is that users have more control over applications while they are being created. For example, unlike previous releases, users can change the default context root of an application from within the wizard instead of after the project has been created. The redesign also makes it a lot easier to add WebFacing support to an existing J2EE project; WebFacing is simply another project facet that can be added or removed like any other facet. You'll also find many usability enhancements, such as being able to add DDS and UIM source members to convert on a single page.

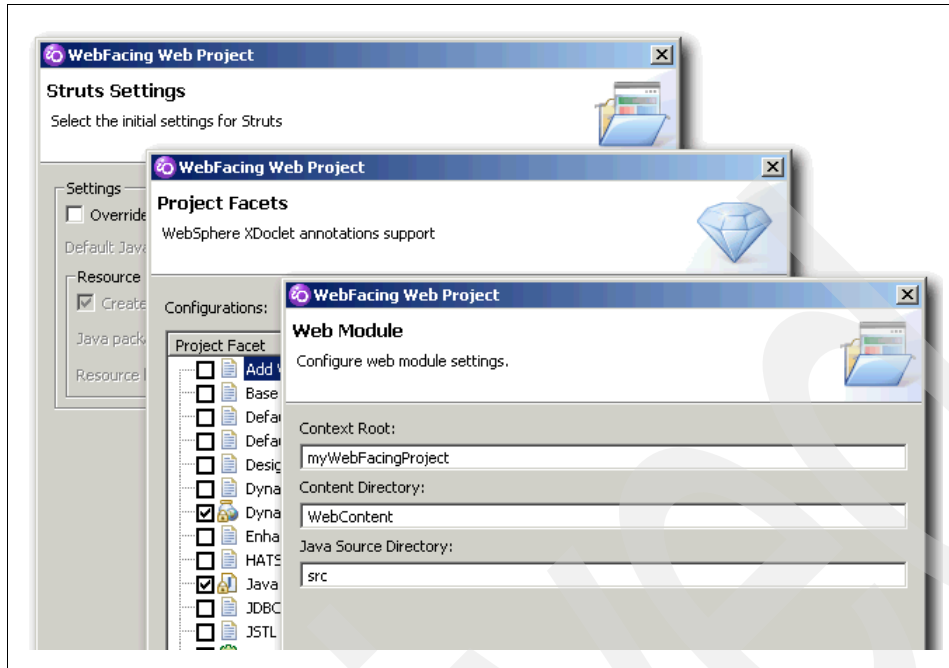


Figure 3-11 WebFacing project wizards' advanced settings pages

New project migration wizard

If you are migrating a WebFacing application to V7.0 from an earlier release, you will find that getting your projects up and running is an easier and less error-prone task than in the past. There is a new Migration wizard that you must run on your project before you can make any changes to it in V7.0, which updates all necessary project files to the current version. This reduces the chances that your WebFacing project will contain mismatched resources from multiple versions if changes and customizations are made prior to migration. Figure 3-12 shows how to access the Migration wizard.

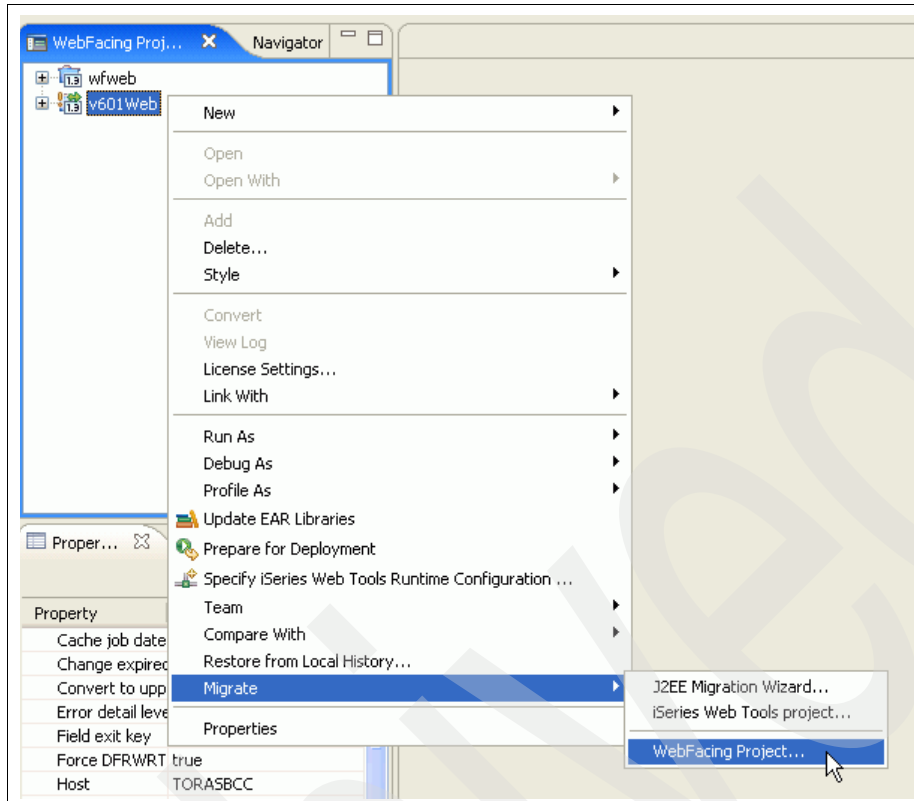


Figure 3-12 Launching the WebFacing Migration wizard

3.3 Web enablement using HATS

This section provides an overview of IBM's Host Access Transformation Services, commonly known as HATS.

3.3.1 Introduction to HATS

HATS, the other IBM Web enablement tool included with WDCS V7.0, is a rule-based transformation engine that allows customized access to host applications by dynamically creating a new Web HTML interface. Unlike WebFacing, which converts DDS and UIM source into Web pages during the development stage, HATS works at runtime, so you do not need to have the source for your application.

At runtime, each area of a green screen is transformed into a variety of Web elements, such as hyperlinks, drop-down combo boxes, input fields, and buttons. Using the default templates and transformation rules, you can get your HATS application up and running on the Web within hours. You can also create or customize the provided templates and recognition patterns, alter the application's flow using macros, and customize and combine screens from different hosts to improve the navigation of your Web application.

Figure 3-13 shows the stages of development, deployment, and runtime for a HATS Web application. If the application is a portlet, the stages are similar except that the portlet is installed and run on a WebSphere Portal system.

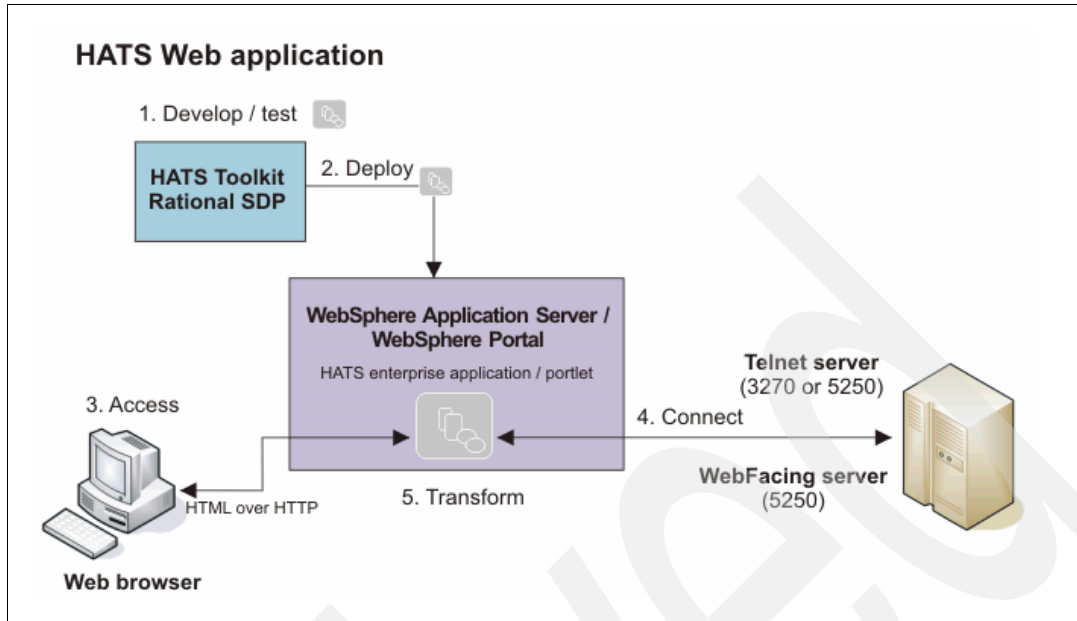


Figure 3-13 HATS Web application stages

1. The developer uses the HATS Toolkit to develop and test a HATS Web application.
2. When ready, the developer deploys the HATS application by exporting it as a J2EE application and installing and running it on a WebSphere Application Server.
3. Using a Web browser, the user accesses the HATS application.
4. The HATS runtime connects to the target host system through either a Telnet or a WebFacing server.
5. As the user interacts with the host through the HATS application, the HATS runtime transforms host screens to a GUI.

In the sections that follow, we discuss HATS in more detail, outlining its benefits, new features, and developer considerations.

3.3.2 HATS features and benefits

Of the many features and benefits we discuss here, one of the most evident is that HATS does not require any source files, which has several positive side effects.

Firstly, you may not have the source for your application at all. It may have been created long ago, and the source is no longer available, or you might have purchased it from a solutions vendor who did not provide source files.

Also, since no DDS or UIM files need to be converted during development, you can typically spend less time on creating your first project and running it. You can later customize and tweak the Web application as needed based on your test runs. No changes ever need to be made to your host application.

Another pleasant by-product is that if the host application does change, no reconversion is needed. One caveat on this point, however, is that customized screens might need adjustments, such as screen recognition criteria tweaks.

HATS comes with many customization capabilities that can be used to move the initial application beyond the default transformation. These include:

- ▶ A wide variety of editable templates to change the look-and-feel of the application.
- ▶ The ability to combine screens from multiple hosts on the same Web page in different tabs.
- ▶ Rich UI widgets such as graphs and calendar date pickers are provided for your use and can be easily integrated using provided tools (Figure 3-14).



Figure 3-14 HATS application using rich widgets

- ▶ Macros allow you to customize and control application flow by skipping screens, prompting for user input before proceeding or entering data on their behalf, or extracting host data for display on the Web. Any series of keystrokes can be captured in a macro. HATS provides full support for macros and includes a macro editor (Figure 3-15).

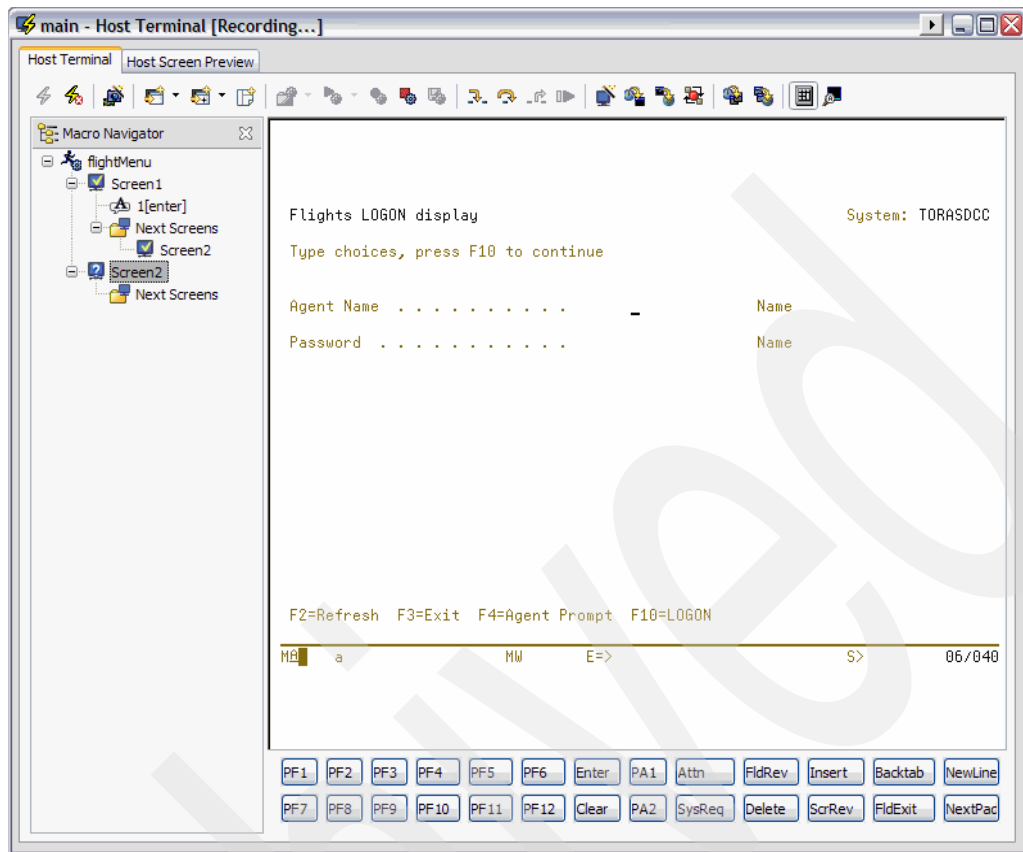


Figure 3-15 HATS macro editor

- ▶ Global variables can be used to store and retrieve data. HATS can also perform mathematical calculations on global variables.

Another benefit of HATS applications is that applications can be run in all A-level Web browsers, including Internet Explorer®, Firefox, Safari, and Opera, allowing a very wide audience of users to be reached.

HATS applications can also be transformed into Web Services using provided wizards, or run as a portlet on the WebSphere Portal Server. Portlets can take advantage of the Portal Server's credentials vault for Single Sign-On, and deliver Click-to-action support to integrate existing applications at the glass with other portlets.

New in V7.0

New functions in HATS Version 7.0 include support for running HATS applications in new environments, new capabilities for creating graphical user interfaces for host applications, and improved ease-of-use. The following describes some of these enhancements.

HATS and WebFacing interoperability

As mentioned earlier, HATS interoperability with WebFacing applications provides the ability to perform data description specification (DDS) map transformation, as well as 5250 data stream transformation, within the same Web application. This function is particularly useful for those who have built separate WebFacing and HATS applications that need to interact with each other.

Host simulation

The HATS Toolkit includes a host simulation capability that allows recording host simulation trace files to save and later use instead of a live host connection. Some of the uses of this function are:

- ▶ Creating screen captures, events, and transformations.
- ▶ Creating and testing macros.
- ▶ Testing HATS applications using the local test environment.
- ▶ Recording a host simulation file in the runtime environment that you can use, along with other traces and logs, to document a failing scenario for troubleshooting purposes.

Project themes

A new project theme wizard (Figure 3-16) provides a simple method of setting a theme for the overall appearance and behavior of a HATS application. The wizard lets you select whether the application should appear more like an emulator, more like a modern application, or somewhere in between. There are many options in the HATS project settings that change the appearance and behavior of the application. The new wizard provides a convenient method of setting the options in one place that together can be treated as a theme.

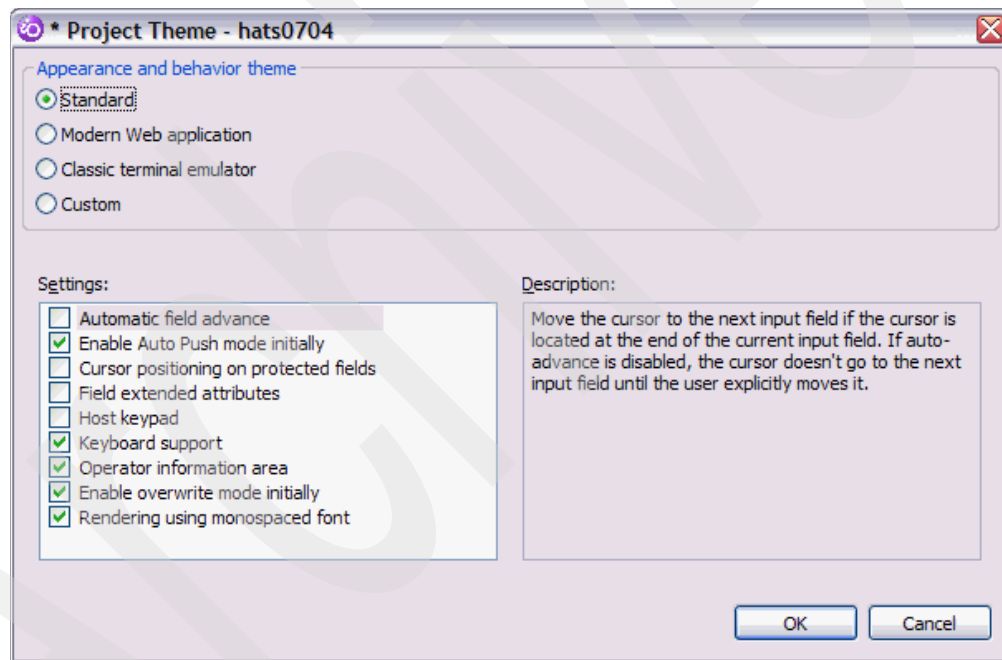


Figure 3-16 Project themes wizard

Testing modes

Various modes are now available that simplify testing your HATS project under different conditions. The *Run on Server* testing mode can be used to modify and test the runtime settings that are deployed to the production environment. The *Debug on Server* mode can be used to test without modifying these production settings. *Profile on Server* can be used for performance analysis, and enables you to locate operations that require the most time, and identify actions that are repeated, to eliminate redundancy.

New host components and widgets

Host components are HATS objects that are responsible for recognizing elements of the host screen. Widgets are HATS objects that are responsible for rendering recognized host

components in a GUI for a host screen. V7.0 includes new host components and widgets plus some new settings for existing ones. Highlights include a new Table component, List, Subfile, and a new option on the Table widget that allows you to retrieve table data in spreadsheet file format.

Support for combining screens

A new screen combination wizard and editor are provided to make it easier to combine data from multiple host screens. These provide a method of defining the screen recognition criteria for the beginning and ending screens, how to navigate from screen to screen, the screen region containing the data to gather, and the component and widget to use to recognize and render the gathered data.

Client settings configuration

In the project settings editor on the Other tab, there is a new section named Client Settings (Figure 3-17). This new section provides a central location for specifying client side settings, such as automatic field advancement, eliminating the need to modify HATS source files.

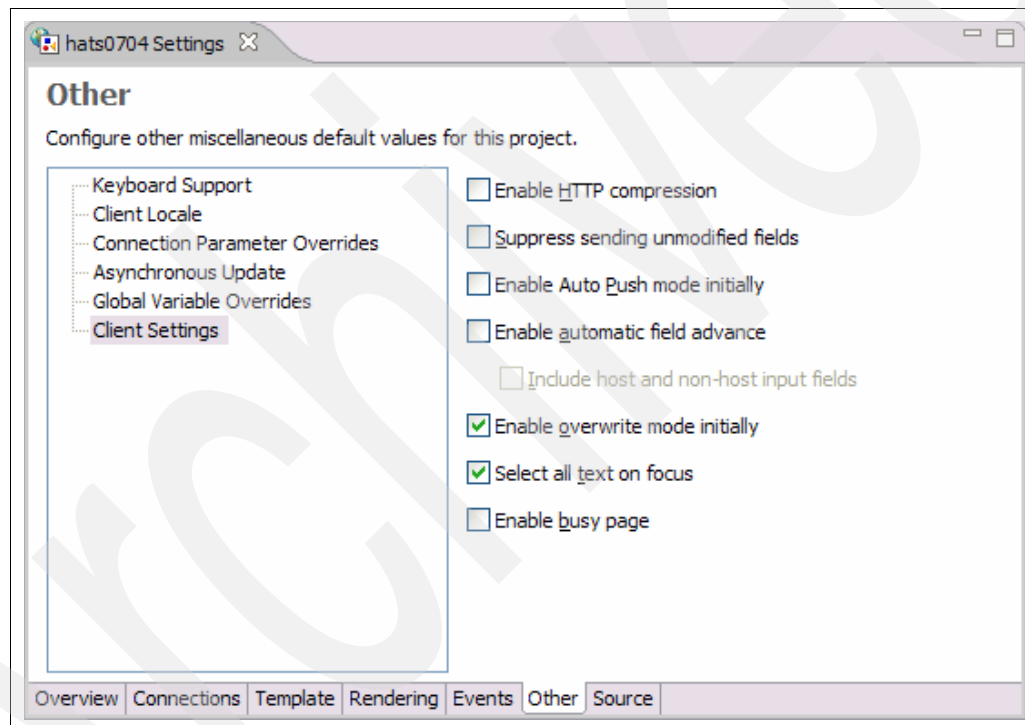


Figure 3-17 Client Settings configuration page

Rich client platform

HATS V7.0 gives you the ability to transform access to your existing host applications using rich client GUI applications that run in Eclipse Rich Client Platform (Eclipse RCP) or Lotus Expeditor Client environments. Since this chapter is about Web enablement, rich clients are outside its scope and will not be covered in further detail. For more information on HATS' rich client support, visit the HATS Information Center at:

<http://publib.boulder.ibm.com/infocenter/hatshelp/v70/index.jsp>

3.4 WebFacing and HATS interoperability

In the past, the usual criteria when choosing which IBM Web enabler to use was whether or not you had the source for your host application. If the DDS was not available, you would use HATS. If the source was available, but you did not want to use a 5250 data stream transformation, WebFacing was the tool of choice.

You may also have applications from different vendors, both with and without DDS source. As a result, you may have used both WebFacing and HATS, developing separate projects that could not work together. WebFacing and HATS interoperability introduced in V7.0 solved exactly this problem.

The process of getting the two technologies to interact or interoperate with each other actually began in V6.0.1. Before V6.0.1, WebFacing users would get a runtime error when their application tried to access screens whose display files had not been converted. WebFacing and HATS interoperability in WebFacing 6.0.1 fixed this problem. With i5/OS* V5R4, users could enable their runtime environments for dynamic data stream transformation based on HATS technology, bringing to an end “DDS not converted” runtime errors. Note that WebFacing applications using the WebFacing and HATS interoperability extension require IBM WebFacing Deployment Tool for WebSphere Development Studio with HATS technology (WDHT) V6.0.1.

For HATS users, WebFacing and HATS interoperability in HATS 6.0.4 delivered dynamic runtime transformation of the 5250 data stream with no required OLTP capacity by working through the WebFacing server on i5/OS V5R4. Deployment of HATS applications also requires WDHT V6.0.1.

In V7.0, WebFacing users can continue accessing system screens or 5250 applications that have not been Web-enabled as in version 6.0.1, or they can use the new linked HATS/WebFacing project to directly mix WebFacing and HATS applications. The new linked projects let users enjoy many customization options only available in a HATS project. For example, they can use HATS macros to automate application execution and control application flow. They can also use different rendering patterns to improve the usability and attractiveness of screens rendered by HATS.

This new linked HATS/WebFacing project is also good news for HATS developers, who can now take advantage of many unique WebFacing features. For example, WebFacing converts display files to JSP. HATS developers can now, through JSP programming, do unlimited detailed and complex application customization. For applications that have display files not easily customizable by HATS, they can use WebFacing to render just those applications and let HATS handle the rest. Additionally, WebFacing handles data validation in the browser, reducing server side calls.

The new linked application interoperates seamlessly. Within the same application, WebFacing performs DDS source transformation, while HATS performs 5250 data stream transformation. One user job is created for one browser session. Out of the box, two new WebFacing styles mentioned earlier in this chapter provide a unified look with pages rendered by HATS

3.5 WDHT

The IBM WebFacing Deployment Tool for WebSphere Development Studio with HATS technology (WDHT) V7.0 is a license enabler for the runtimes of WebFacing V7.0 and HATS V7.0. You must purchase WDHT licenses to enable the runtime environment for each. Without WDHT, all WebFacing applications, HATS applications, and linked WebFacing/HATS applications are limited to two trial connections for development and testing.

More information is available on IBM's WDHT Web site:

<http://www.ibm.com/software/awdtools/wdht>

Once you have purchased licenses and received your WDHT CD, follow the steps below to unlock the two-connection limit for your WebFacing, HATS or linked WebFacing/HATS applications.

1. Insert the WDHT CD.
2. Start WebSphere Development Studio Client for System i or WebSphere Development Studio Client Advanced Edition for System i.
3. In the WebFacing perspective, right-click anywhere in the WebFacing Projects view and select **License Settings** (Figure 3-18).

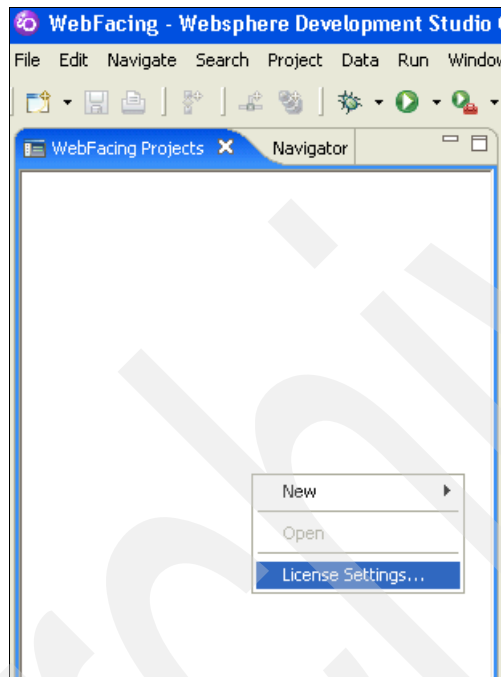


Figure 3-18 Launching the License Settings wizard from the WebFacing Projects view

Alternatively, select **License Settings** from the HATS toolbar menu in the HATS perspective (Figure 3-19).

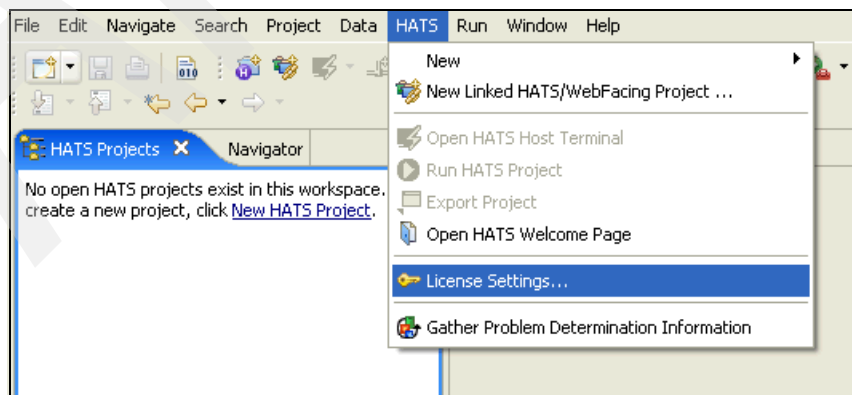


Figure 3-19 Launch License Settings from the HATS perspective

4. On the License Settings dialog, specify the location of the enablement file: <cd-rom drive>\runtimeenablement7-webrcp.jar (Figure 3-20).

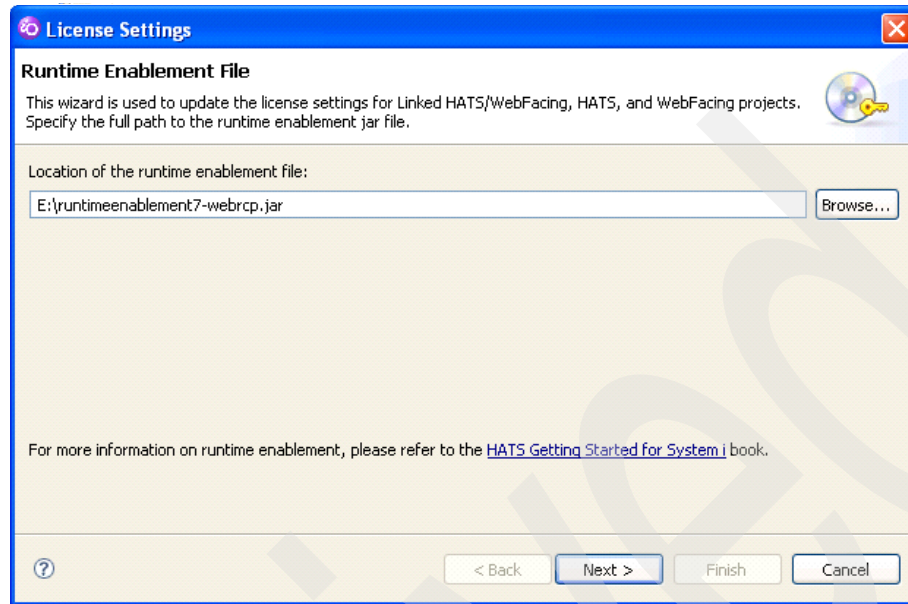


Figure 3-20 License Settings dialog pointing to a runtime enablement file

5. On the License Options panel (Figure 3-21), select the license option as dictated by your product Proof of Entitlement and click **Finish**. Options are **User** or **Processor or Processor / System i tier (WDHT only)**. If you specify **User**, you must also enter the number of licenses purchased.

To enable the runtime environment for all new projects regardless of which workspace you create them in, select **Apply to all new projects created in any workspace**. To enable the runtime environment only for projects that exist in the current workspace, select **Apply to all existing projects in the current workspace**. By default, both are selected. If you have existing projects in other workspaces, you must run the License Settings wizard in each of those workspaces.

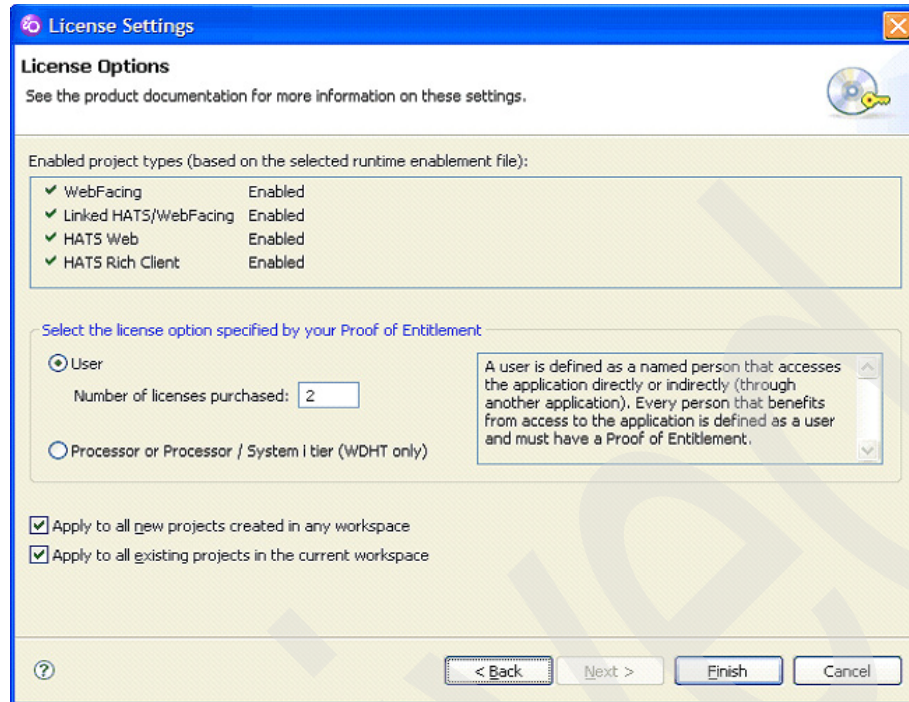


Figure 3-21 Choose licensing option and other settings

Note: WebFacing project types are listed only if WebFacing is installed, and the same applies to HATS.

3.6 V7.0 enhancements and changes summary

There have been several enhancements to the IBM WebFacing Tool since the last edition of this book, which covered V5.1.2. HATS has also included a multitude of improvements since V5, the topic of the most recent book on that product.

The following are lists of changes and enhancements made to the WebFacing Tool and to HATS since their last IBM Redbooks publications.

WebFacing

V7.0

- ▶ Linked HATS/WebFacing applications
- ▶ WDHT Licensing change
- ▶ WebFacing Web Project wizards redesigned
- ▶ New Web Styles
- ▶ New project migration wizard
- ▶ WebFacing cheat sheets in the Help menu
- ▶ Dynamic help
- ▶ WebFacing Developer role added to the Welcome view

V6.0.1

- ▶ Common Base Event logging data generated by the WebFacing Tool
- ▶ Embedded HATS Screen Rendering (HSR) of non-WebFaced screens
- ▶ Enhanced support for bidirectional languages

V6.0

- ▶ DDS outline view
- ▶ Web Settings view
- ▶ Performance improvements for conversion and runtime
- ▶ Cascading Style Sheets-Positioning (CCS-P) Web Setting (Advanced Edition)

HATS

V7.0

- ▶ Rich client platform support
- ▶ Simplified combination of multiple screens of tabular data
- ▶ HATS and WebFacing application run-time interoperability
- ▶ Integration of host simulation tool to allow easier offline development
- ▶ IBM Support Assistant (ISA) plugin support
- ▶ Conversion of HATS table component data to a spreadsheet file
- ▶ Improved rendering patterns for transformations
- ▶ New and improved host components and widgets
- ▶ New project themes wizard
- ▶ Expanded print support configuration
- ▶ Enhanced default rendering
- ▶ New client setting configurations
- ▶ New support for DBCS 3270 and 5250 fields
- ▶ Enhanced subfile column detections and settings
- ▶ Global rules enhancements

V6.0

- ▶ CICS® BMS map importer for screen customization and transformation creation without being connected to a host
- ▶ Wizard to create custom components and widgets
- ▶ JSF support for Integration Objects
- ▶ The ability to specify the “next” likely screen customizations for better performance
- ▶ Enhanced macro debugging that allows stepping into and over screens in a macro
- ▶ Recognition and rendering of 5250 screens with Enhanced Non-Programmable Terminal User Interface (ENPTUI)
- ▶ Improved 5250 subfile recognition, rendering, and customization
- ▶ Easier to arrange and edit HATS components and widgets on a transformation page



Bringing Your i5/OS applications to the Web

This chapter summarizes the features related to putting your i5/OS applications to the Web:

- ▶ Web development tools in WDSC
- ▶ Web development with JSF
- ▶ Enabling SOA through Web Services
- ▶ WebFacing, HATS, and WDHT
- ▶ Using WebFacing to create browser interface for a 5250 application

4.1 Tools for Web development

WebSphere Studio is a development environment designed to make Web application creation simple and efficient. The Web tools that are part of WebSphere Studio allow you to work with Web technologies to create Web applications that range from simple, static Web sites, to fully dynamic Web applications that access data sources.

Web perspective

The *Web perspective* contains a selection of views and editors that are customized to be most useful to a Web developer. These views include reusable drag and drop components, link structures, access to page data, your project resources, code snippets, image galleries, and more. The editors range from standard source editors with content assist features, to Web Site Designer and Page Designer, editors that are fully-featured to help create, edit, and preview Web sites and the pages within those Web sites.

The following tools are the primary means for developing Web applications in the workbench.

Wizards

The wizards are the easiest way to add static or dynamic content to your Web pages. The basic wizards allow you to create and specify settings for Web projects, HTML files, and JSP files, and use the examples provided with the product. The advanced wizards help you retrieve information from common databases, use Java bean APIs to query or update databases, and generate output forms and servlets. With these features, you can do everything from a simple table lookup to more complex interactive applications.

The wizards provide detailed graphical user interfaces to guide you through the process of creating SQL queries, choosing Java methods and properties, and integrating them into the Web pages you are creating. You do not have to be an expert at SQL syntax or Java programming, because the wizards walk you through the process step-by-step, and then generate sophisticated JSP and servlet code to run your Web applications.

Web Site Designer

Web Site Designer is a high-level editor that allows you to quickly build complex Web sites. The navigation support enables you to begin your site by designing its organization, and then populating it with the appropriate Web pages. Web Site Designer also enables easy reorganization if you decide to restructure your Web site through a drag and drop site map.

You can toggle between two modes to design your site's navigation. Web Site Designer has a Navigation view and a Details view to help you plan, organize, and create your overall Web site.

Page Designer

Page Designer is an advanced-function HTML and JSP editor that allows you to quickly build complex Web pages, both visually and textually. The dynamic element support enables you to include form elements, Java applets, embedded scripts, and JavaServer Pages (JSP) tags.

You can toggle among three modes to visually design pages, work with HTML, JavaScript or JSP content, and preview your pages. Content assist, which provides guided editing as you insert new tags, is available in the Source (text) page. To help you create the visual impact you want on your Web sites, the editor includes its own library of reusable graphics and two graphic programs for creating, editing, and animating image files.

Web deployment descriptor editor

The Web deployment descriptor editor helps you to define any of the deployment information that can exist in the Web application deployment descriptor included in a WAR file. Pages in this graphical editor provide controls to set numerous Web Application parameters related to servlets, filters, life-cycle listeners, mime mappings, parameters, references, security, and other general deployment settings.

In addition, the Web deployment descriptor editor provides an opportunity to configure WebSphere Application Server-specific bindings and extensions, as appropriate. When a new Web project is created, the generated web.xml file contains an appropriate DOCTYPE declaration. The *Sun Microsystems Java Servlet 2.3 Specification* includes the DTD for the web.xml file, along with examples.

Content assist

Content assist is a WebSphere Studio source editing tool that prompts you with a list of valid alternatives for completing the current line of code or inserting a macro. Content assist is sensitive to the context of the cursor position. It recognizes whether you are in an HTML area, a JavaScript area, or a JSP area. For HTML, content assist provides an appropriate list of proposals (tags, attributes, and attribute values) based on the context. Content assist can be invoked at anytime by pressing Ctrl+Space or selecting a menu item. You can also set a preference that automatically invokes content assist when special characters, such as “.”, “<”, or “=” are typed.

Content assist is available within the Page Designer Source page for adding HTML, WML, JSP, and JavaScript tagging. JavaScript content assist is available within a JSP and HTML files, and Java content assist is available within those JSP tags that support Java code, such as JSP scriptlet tags. Variations of content assist are also available within Java and XML source editing areas. See content assist documentation for specific source editors to understand the capabilities and limitations for specific code types.

4.2 Web development with JSF

Java Server Faces (JSF) is a new way of developing i5/OS Web applications. The System i-specific Web tools based on Struts, for example, Web Interaction wizard and Web Components, are deprecated. JSF is the new strategic direction.

Note: If you are an existing user running a migration tool on a project from a pre-V7.0 release, a deprecation warning message will be shown. Figure 4-1 shows an example of such a message. This particular message will be shown when you create a new Web project when adding Project Facet “System i Web Components Tag Library”.



Figure 4-1 Deprecation warnings message example

4.2.1 JSF-based architecture - the new way

JavaServer Faces (JSF) is a standard framework that simplifies the construction of user interfaces (UI) for server-based applications by enabling the assembly of reusable UI components in a page. JSF facilitates the connection of widgets to data sources and to server-side event handlers.

The JSF specification defines a set of standard UI components and provides an Application Programming Interface (API) for developing components, or enables the reuse and extension of the existing standard UI components.

JSF reduces the effort of creating and maintaining applications that run on a Java application server and render application UI to a target client. It facilitates Web application development through:

- ▶ Reusable components to create UI
- ▶ Migration of application data to and from the UI
- ▶ Managing UI state across server requests
- ▶ Enabling the implementation of custom components
- ▶ Providing a model for wiring client-generated events to server-side application code

JSF provides component tags for all input fields available and implements the Model-View-Controller (MVC) design framework, ensuring that applications are well designed and easier to maintain. JSF allows Java programmers to focus on backend application development that is integrated with the UI. It allows Web page creators to create UI by assembling pre-built JSF components that already contain the necessary logic.

Now you should use JSF UI components instead of System i Web components. For the JavaBean, you should now use Program Call Bean with JSF rather than Web Interaction wizard. It is more flexible and has a richer user interface and interaction.

Service Data Object (SDO)

Service Data Objects (SDO) is a data programming architecture and API for the Java platform that unifies data programming across data source types, provides robust support for common application patterns, and enables applications, tools, and frameworks to more easily query, view, bind, update, and introspect data.

Applications based on JavaServer Faces (JSF) technology require a data access method that supports the disconnected Web model. Such dynamic Web applications typically access data sources and are able to create, read, update, and delete data dynamically. This product offers a simple data access paradigm with the Faces and Faces Client components, a convenient and robust data access technology called Service Data Objects (SDO) (referred to as WDO for WebSphere Application Server version 5.x) that enables you to avoid persistent objects and to deal exclusively with data. Service Data Objects technology is:

- ▶ **Data-centric:** Results are retrieved as a structured graph of data. SDO does not support the retrieval of objects (which include associated behaviors) as is the case with EJB persistence mechanisms.
- ▶ **Disconnected:** The retrieved result (data table) is independent of any associated connections or transactions.
- ▶ **XML-integrated:** Easy conversion of retrieved data to and from XML format.
- ▶ **Source-independent result set:** The results of SDO queries are standardized because results returned from any type of data source are converted to an SDO graphical representation. For example, an SDO query against a JDBC™ data source and a query against an EJB data source both return SDO-structured results that use the same set of standard data types.

You can still use other data access methods with Faces and Faces Client components, including data access beans and Web services; however, SDO is optimized for Web applications, in particular dynamic Web applications that use JSF technology. These SDO capabilities supplement the currently available object-oriented data access mechanism provided by EJB, and the data-centric, but connected, mechanism provided by direct use of JDBC.

4.3 Enabling SOA through Web Services

SOA is a methodology that can be used in software development. As the concept of an “object” is central to Object Oriented Architecture, SOA is based on the concept of a “service”. In SOA, a service is an application that can be invoked by other applications. The concept of a service in software is similar to the concept of a service in real life. For example, if you want to relocate, you look up available services in the Yellow Pages. Then you can contact the relocation company directly and work out the details of the move. The same scenario can apply to a software interaction. A company can have software that handles relocation procedures. The software can search a public service registry for relocation services. Based on some selection criteria (price, delivery time), the relocation application can choose one of the relocation providers and work with them directly to schedule the move and arrange payment.

The described services scenario involves the following three participants:

- ▶ Service requestor: an application with a business need
- ▶ Service broker: a registry of all available services
- ▶ Service provider: an application that implements a business function

A service provider implements a service and publishes it to the service broker. A service requestor searches the registry to find a service of interest. Upon finding a service, the service requestor binds to the service provider and invokes the service with the help of an XML file describing the Web Services interface.

4.3.1 Using a Web Service

When you want to know when a plane arrives, do you also need to know who is the pilot, where it came from, what is its color and all those kind of details? Certainly not. Well, then, you don't want a big monolithic application, needing far too much time to get just one piece of information. This is why the Web Services exist.

How to describe a Web Service

An application, when it has been well designed, is composed of several small functional units. Each of these units can be called a service. When you are facing a Web application, then you can use those Web Services. They are self-contained software components.

The interface of the Web Service is described in an extensible markup language (XML) formatted language called Web Services Description Language (WSDL). XML is a standardized language: everybody can receive and translate it. WSDL describes:

- ▶ What the Web Service can do
- ▶ Where it resides
- ▶ How to invoke it
- ▶ The structure of the input and output messages (parameters)
- ▶ Which communication protocols to use

A Web Service can be an operation, or an exchange of data. It follows a strict protocol based on XML, called Simple Object Access Protocol (SOAP) to describe how it can interfere with a program or another Web Service.

The most interesting part, when using a Web Service, is that it doesn't depend on your hardware platform, your operating system, your software, or the language you are using.

How it works

An application accesses your Web Service by issuing a request. This request has to be formatted following the SOAP rules. Your Web Service receives the request, treats it and sends an answer back, still encoded with SOAP.

Installing and managing WDSC

This part covers the installation and management topics with the following chapters:

- ▶ Installation of WDSC
- ▶ Managing your WDSC instance

Archived



Installing WDSC

This chapter introduces you to the installation of WDSC V7.0 and WDSC AE V7.0. Starting with Version 7.0, the IBM Installation Manager is used to install and manage the WDSC product on your workstation. The IBM Installation Manager also replaces the Rational Product Updater that was used in version 6 of the product to maintain WDSC. It provides a selective install, and many other installation management features.

5.1 Overview of the installation process

In Version 7.0, the IBM Installation Manager is used to install and service the product. It is a program that helps you install WDSC on your workstation. It also helps you update, modify, and uninstall this and other packages that you install. A package can be a product, a group of components, or a single component that is designed to be installed by Installation Manager.

The launchpad program provides a single location to view release information and start the installation process. This section provides instructions for installing IBM Installation Manager and WDSC.

5.1.1 WDSC installation wizard

This section explains what WDSC installation wizard is and what it does.

Installation of Installation Manager

IBM Installation Manager is installed automatically by the launchpad program. When starting the installation process from the launchpad program, IBM Installation Manager starts preconfigured with the location of the repository that contains the WDSC package.

Installing Installation Manager manually

1. Run `setup.exe` from the `InstallerImage_win32` folder on the first installation disk.
2. Click **Next** on the Welcome screen.
3. Review the license agreement on the License Agreement page and select **I accept the terms in the license agreement** to accept. Click **Next**.
4. Click **Change** on the Destination Folder page to change the installation location if required. Click **Next**.
5. Click **Next** on the Setup Type page.
6. Click **Install** on the Ready to Install Program page. The Completed page opens after the installation is complete.
7. Click **Finish**.

If you install and start the Installation Manager directly, then you must set the repository preference manually.

Starting Installation Manager manually

1. Open the Start menu from the Taskbar.
2. Select **All Programs** → **IBM Installation Manager** → **IBM Installation Manager**.

Installation of RAD components

WDSC V7.0 inherits and extends the *Rational Software Delivery Platform (RSDP)* and a subset of *Rational Application Developer (RAD)*. RAD components are installed with WDSC V7.0.

Advanced WSC AE V7.0 contains the full capabilities of RAD V7.0 and WDSC V7.0.

For more information about RAD components in WDSC, and WDSC packaging, refer to 2.1, “Building blocks of WDSC” on page 16.

5.1.2 Introduction to Installation Manager

Regardless of which installation scenario you follow to install WDSC on your workstation, you always need to use Installation Manager to install your package.

IBM Installation Manager offers a number of time-saving features. It keeps track of what you are about to install, software components that you have already installed, and components that are available for you to install. It searches for updates so you know that you are installing the latest version of a WebSphere or Rational product package.

Installation Manager also provides tools for managing licenses for the product packages that it installs. It provides tools for updating and modifying packages. You can also use the Installation Manager to uninstall product packages.

The IBM Installation Manager comprises five wizards that make it easy to maintain your product packages through their life cycles:

- ▶ The **Install Packages** wizard walks you through the installation process. You can install a product package by simply accepting the defaults, or you can modify the default settings to create a custom installation. Before you install a product package, you are provided with a complete summary of the selections that you have made throughout the wizard. With the wizard you can install one or more product packages at the same time.
- ▶ The **Update Packages** wizard searches for available updates to product packages that you have installed. An update might be a released fix, a new feature, or a new version of the product. Details of the contents of the update are provided in the wizard. You can choose whether to apply an update or not.
- ▶ With the **Modify Packages** wizard you can modify specific elements of a package that you have already installed. During your initial installation of the product package, you select the features you want to install. If you find later that you require other features, you can use this wizard to add them to your product package installation. You can also remove features, and add or remove languages.
- ▶ The **Manage Licenses** wizard helps you to set up the licenses for your packages. However, you will not use this wizard for installing a WDSC package. WDSC includes a full license that is automatically installed upon installation of the product.
- ▶ The **Uninstall Packages** wizard helps you to remove a product package from your computer. You can uninstall more than one package at a time.

5.2 Planning for installation

Effective planning and an understanding of the key aspects of the installation process can help ensure a successful installation. This section first covers the necessary prerequisites, then suggests a planning checklist and discusses some migration considerations.

5.2.1 WDSC product prerequisites

This section details hardware, software, and user privilege requirements that must be met in order to successfully install and run your software.

Hardware

Before you can install WDSC and WDSC AE, verify that your system meets the minimum hardware requirements.

The following are the minimum hardware requirements for installing WDSC V7.0 or WDSC AE V7. Note that the actual hardware needed depends heavily on which components you select at installation time.

- ▶ Intel® Pentium® III 800 MHz processor minimum (higher is recommended)
- ▶ 768 MB RAM; 1 GB RAM recommended
- ▶ For the i5/OS Development Tools, Remote System Explorer and System i Projects
 - 256 MB RAM recommended
- ▶ Disk space:
 - For installing the i5/OS Development Tools feature only, 1GB of disk space is required. 500 MB is used only during the installation process and can be recovered after the installation is complete.
 - Disk space requirements can be reduced or increased depending on the features that you install.
 - Installation requires:
 - For WDSC, a minimum of 5 GB of disk space. Some of this disk space (500 MB) is used only during the installation process and can be recovered after the installation is complete.
 - For WDSC AE, a minimum of 5.5 GB of disk space. Some of this disk space (500 MB) is used only during the installation process and can be recovered after the installation is complete.
 - Additional disk space is required for developing applications.
 - Additional disk space is required if you download the product package to install this product.
 - Additional disk space is required if you use the FAT32 file system instead of NTFS.
 - An additional 500 MB of disk space is required in the directory pointed to by your environment variable TEMP.
- ▶ Display resolution: 1024 x 768 minimum
- ▶ Microsoft® mouse or compatible pointing device
- ▶ CD-ROM drive

Advanced

Software

Before you install the product, verify that your system meets the software requirements as described in the following sections.

System

The following software needs to be installed under i5/OS on your System i system:

- ▶ i5/OS V5R3, or V5R4
- ▶ WDSC and WDSC AE V7.0 support for the i5/OS release will end when that i5/OS release officially ends service, or when WDSC V7.0 officially ends service. You may be able to access OS/400 V5R2 and earlier releases, but as they are not supported any longer, any problems with WDSC accessing them may not be resolved.
- ▶ WebSphere Development Studio for System i (5722-WDS).
- ▶ WebSphere Development Studio for System i (5722-WDS) option 60 *Workstation Tools - Base*.

- Some components will need additional program temporary fixes (PTFs). The PTF information is available online at:

<http://www.ibm.com/support/docview.wss?uid=swg21044473>

To find more information about any required System i server PTFs, we recommend you perform the following steps after you installed WDSC on the first workstation:

1. Open the Remote System Explorer perspective if it is not already open. The name of the current perspective is shown in the top left corner of the title bar for the Window. To open the Remote System Explorer perspective, select **Window** → **Open Perspective** → **Other** → **Remote System Explorer** .
2. Create a connection to access i5/OS on your System i by expanding **New Connection** → **System i**.
3. Expand your new connection and right-click **System i Objects**. Select **Verify Connection** from the menu. This displays a dialog like the one in Figure 5-1 that shows which required PTFs are already installed on the system and which ones are missing.

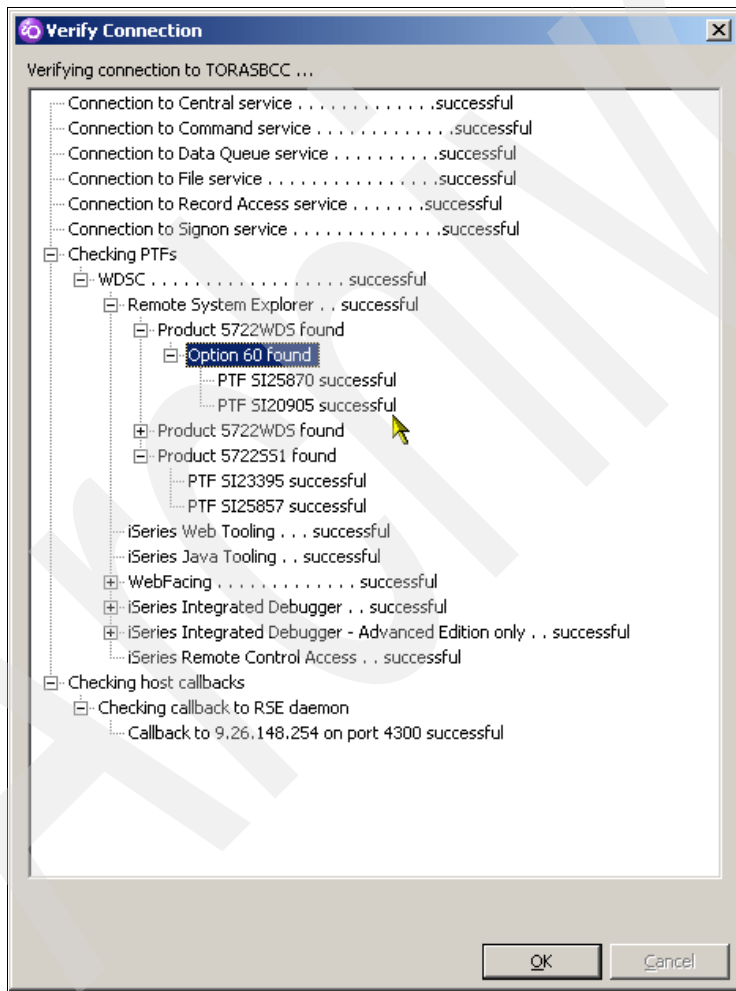


Figure 5-1 Verifying connection and prerequisite program products and PTFs

To compile RPG, COBOL, C, C++, CL, or DDS on the System i5™, using the Remote System Explorer (RSE) or CODE, i5/OS (5722-SS1) and the corresponding options of 5722-WDS must be installed on your System i5 or System i system.

Tip: In order to see the latest available PTFs, your workstation needs to have access to the Internet. Otherwise a list of PTFs known at the time when the product was packaged is used.

Necessary software on your workstation

The following operating systems are currently supported for WDSC V7.0 and WDSC AE V7.0:

- ▶ Microsoft Windows 2000 Professional with Service Pack 4 or later
- ▶ Microsoft Windows 2000 Server with Service Pack 4 or later
- ▶ Microsoft Windows 2000 Advanced Server with Service Pack 4 or later
- ▶ Microsoft Windows XP Professional with Service Pack 2 or later
- ▶ Microsoft Windows Server® 2003 Standard Edition
- ▶ Microsoft Windows Server 2003 Enterprise Edition

Additional software requirements

- ▶ One of the following Web browsers is required for viewing the Readme files and the installation guide, and to support the Standard Widget Toolkit (SWT) browser widget:
 - Microsoft Internet Explorer 6.0 with Service Pack 1.
 - Mozilla 1.6 or 1.7.
 - Firefox 1.0.x or 1.5.
- ▶ The launchpad does not support Mozilla 1.6; if your browser is Mozilla, you need version 1.7 or later to run the launchpad.
- ▶ To properly view multimedia user assistance, such as tours, tutorials, and demonstration viewlets, you must install Adobe® Flash Player Version 6.0 release 65 or later.
- ▶ For information about supported database servers, Web application servers, and other software products, refer to the online help. After installing the product, click **Help** → **Help Contents**.
- ▶ To run applications built with IBM WebFacing Tool, Internet Explorer 6.0 or later is required.
- ▶ To run applications built with i5/OS Web Tools, Internet Explorer 6.0 or later, or Mozilla V1.7 or later, is required.
- ▶ TCP/IP must be installed and configured.
- ▶ To profile applications, JRE™ 1.4, or later should be installed.
- ▶ The following runtime environments are supported for IBM WebFacing Tool, and i5/OS Web Tools applications:
 - WebSphere Application Server Express (Version 5.1, or 6.0) installed locally or remotely
 - WebSphere Application Server (Version 5.1, 6.0 or 6.1) installed locally or remotely
- ▶ **Advanced** Developing portlets in either the IBM WebFacing Tool or HATS requires portal development tools that are available in WDSC AE only.

User privileges

You must have a user ID that meets the following requirements before you can install WDSC or WDSC AE:

- ▶ Your user ID must not contain double-byte characters.
- ▶ You must have a user ID that belongs to the local Windows workstation's administrators group.

5.2.2 Installation planning checklist

This section provides a checklist that helps you in planning a consistent and complete installation of WDSC and WDSC AE.

Choosing installation media

There are a number of scenarios that you can follow when installing WDSC.

The following factors might determine your installation scenario:

- ▶ The format and method by which you access your installation files (for example, from CDs or files downloaded from the IBM Passport Advantage® Web site).
- ▶ The location for your installation (for example, you can install the product onto your own workstation, or make the installation files available to your enterprise).
- ▶ The type of installation (for example, you can use the Installation Manager GUI, or install silently).

The following section describes four installation scenarios you can choose from:

Installing from the CDs

In this installation scenario, you have CDs that contain the product package files, and typically you are installing WDSC on your own workstation.

For an overview of the steps, refer to “Using CDs” on page 87.

Installing from a downloaded electronic image on your workstation

In this installation scenario, you have downloaded the installation files from either IBM Passport Advantage or the Entitled Software Support site and you will install WDSC on your own workstation.

For an overview of the steps, refer to “Using an electronic image on your workstation” on page 87.

Installing from an electronic image on a shared drive

In this installation scenario, you can place the electronic image on a shared drive so that users in your enterprise may access the installation files for WDSC from a single location.

For an overview of the steps, refer to “Using an electronic image on a shared drive” on page 88.

Installing from a repository on an HTTP or HTTPS server

This installation scenario represents the fastest method for installing the product across a network. This scenario differs from the shared-drive installation,

In order to place product package files for WDSC on an HTTP Web server, you use the IBM Packaging Utility as we describe in 6.4, “Placing WDSC on an HTTP server” on page 122. This utility is provided with WDSC to copy the installation files in a format—a package—that can be used for installing WDSC directly from an HTTP Web server. The directory on the HTTP Web server that contains the package is called a *repository*.

Note: In this scenario, the installation files for only WDSC are placed in the package.

For an overview of the steps, refer to “Using a repository on an HTTP Web server” on page 88 and 6.4, “Placing WDSC on an HTTP server” on page 122.

Deciding which features to install

In WDC V7.0, you can customize your software product by selecting which features to install.

When you install the WDC product package using IBM Installation Manager, the installation wizard displays the features in the available product package. From the features list, you can select which to install. A default set of features is selected for you (including any required features). Installation Manager automatically enforces any dependencies between features and prevents you from clearing any required features.

Where did WDC Lite go?

With WDC 6.0.1 an option called *WDC Lite technology preview* was provided, resulting in a lightweight edit/compile/debug environment for developing native i5/OS applications.

After you installed the complete product, you could download and install an additional component of WDC. It allowed you at execution time to run a different version of the program which had only the functions of the Remote System Explorer and System i projects enabled. This resulted in much less CPU and memory usage, but did not reduce the disk space, because the full product needed to remain installed.

Now, with WDC V7.0 and WDC AE V7.0, a different approach has been implemented to allow you to reduce resources. You can now select at installation time which component you need. For example, not installing the Web Development Tools (see Figure 5-10 on page 97) drastically reduces disk space requirements, as well as memory and processor load at execution time.

Note: After you finish installing the package, you can still add or remove features from your software product by running the Modify Packages wizard in Installation Manager.

For more information, refer to 6.3, “Modifying the installation” on page 118.

Features

To check WDC V7.0 and WDC AE V7.0 features, refer to the Features section in WDC Installation Guide. The latest version of the Installation Guide for WDC V7.0 is available at:

http://download.boulder.ibm.com/ibmdl/pub/software/awdtools/wdc/v7/70/documents/install_instruction/install.html

The latest version of the Installation Guide for WDC AE V7.0 is available at:

http://download.boulder.ibm.com/ibmdl/pub/software/awdtools/wdcae/v7/70/documents/install_instruction/install.html

Upgrade and coexistence consideration

If you have a previous version of the product, or if you plan to install multiple Rational Software Development Platform products on the same workstation, review the information in this section.

Coexistence considerations for IBM product offerings

Some products are designed to coexist and share function when they are installed in the same package group. A package group is a location where you can install one or more software products or packages.

When you install each package, you select whether you want to install the package to an existing package group, or whether you want to create a new one. IBM Installation Manager

will block products that are not designed to share or do not meet version tolerance and other requirements. If you want to install more than one product at a time, the products must be able to share a package group.

At the time of release, the following products will share function when installed to a package group:

- ▶ WDSC or WDSC AE
- ▶ WebSphere Developer for zSeries®
- ▶ Host Access Transformation Services Toolkit
- ▶ Rational Application Developer
- ▶ Rational Software Architect
- ▶ Rational Functional Tester
- ▶ Rational Software Modeler
- ▶ Rational Systems Developer

Any number of eligible products can be installed to a package group. When a product is installed, its function is shared with all of the other products in the package group. If you install a development product and a testing product into one package group, when you start either of the products, you have both the development and testing functionality available to you in your user interface. If you add a product with modeling tools, all of the products in the package group will have the development, testing, and modeling functionality available.

If you install a development product and later purchase a development product with increased functionality and add that product to the same package group, the additional function will be available in both products. If you uninstall the product with the greater functionality, the original product remains.

Note: Each product installed into a unique location may be associated with only one package group. A product must be installed into multiple locations in order to be associated with multiple package groups. Rational Functional Tester can be installed into only one location on a computer.

Upgrade considerations

You cannot upgrade WDSC V6.x or WDSC AE V6.x or any previous version that is already installed on your workstation to version 7.0. However, WDSC V7.0 and WDSC AE V6 can coexist with earlier versions and will not delete any previous versions during installation.

You can migrate your workspace, projects, and artifacts from WDSC or WDSC AE version 5.1.2 or 6.x to version 7.0. For more information, refer to 5.2.3, “Migration from previous WDSC versions” on page 86.

Important: The first time you start WDSC or WDSC AE version 7.0, it defaults to a new workspace. Do not open your existing workspace in version 7.0 until you read the migration documentation.

If you have a previous version of Agent Controller, you should stop it, uninstall it, and clean up any leftover files before installing this version of Agent Controller.

Installation repositories

IBM Installation Manager retrieves product packages from specified repository locations.

Some organizations bundle and host their own product packages on their intranet. For information about this type of installation scenario, refer to “Installing from a repository on an

HTTP or HTTPS server” on page 81. Your system administrators will need to provide you with the correct URL.

By default, IBM Installation Manager uses an embedded URL in each Rational software development product to connect to a repository server over the Internet. Installation Manager then searches for the product packages as well as new features.

Setting repository preferences in Installation Manager

When you start the installation of WDSC or WDSC AE from the Launchpad program, the location of the repository that contains the product package you are installing is automatically defined in IBM Installation Manager. However, if you start Installation Manager directly (for example, installing WDSC or WDSC AE from a repository located on a Web server) then you must specify the repository preference (the URL for the directory that contains the product package) in Installation Manager before you can install the product package.

Specify these repository locations on the Repositories page of the Preferences window. By default, Installation Manager uses an embedded URL in each Rational Software Development product to connect to a repository server through the Internet and search for installable packages and new features. Your organization may require you to redirect the repository to use intranet sites.

To add, edit, or remove a repository location in Installation Manager:

1. Start IBM Installation Manager.
2. On the Start page of Installation Manager, click **File** → **Preferences**, and then click **Repositories**. The Repositories page opens, showing any available repositories, their locations, and whether they are accessible.
3. On the Repositories page, click **Add Repository**.
4. In the Add Repository window, type the URL of the repository location or browse to it and set a file path.
5. Click **OK**. If you entered an HTTPS repository location, you will be prompted to enter a user ID and password. The new or changed repository location is listed. If the repository is not accessible, a red x is displayed in the Accessible column.
6. Click **OK** to exit.

Notes:

- ▶ Before starting the installation process, be sure to obtain the installation package repository URL from your administrator.
- ▶ For Installation Manager to search the default repository locations for the installed packages, ensure that the preference “Search the linked repositories during installation and updates” on the Repositories preference page is selected. This preference is selected by default.

Package groups and the shared resource directory

When you install the WDSC or WDSC AE package with IBM Installation Manager, you must choose a package group and a shared resource directory.

Package groups

During the installation process, you must specify a *package group* for the WDSC package. A package group represents a directory in which packages share resources with other packages in the same group. When you install the WDSC package with Installation Manager, you can create a new package group or install the packages into an existing package group.

(Some packages might not be able to share a package group, in which case the option to use an existing package group will be disabled.)

When you install multiple packages at the same time, all the packages are installed into the same package group. Refer to “Coexistence considerations for IBM product offerings” on page 82 to view a list of the supported packages.

A package group is assigned a name automatically; however, you choose the installation directory for the package group.

After you create the package group by successfully installing a product package, you cannot change the installation directory. The installation directory contains files and resources specific to the WDCS product package installed into that package group. Resources in the product package that can potentially be used by other package groups are placed in the shared resources directory.

Shared resources directory

The *shared resources directory* is the directory where installation artifacts are located so that they can be used by one or more product package groups.

Important: You can specify the shared resources directory once: the first time that you install a package. We suggest that you use your largest drive for this. You cannot change the directory location unless you uninstall all packages.

Extending an existing Eclipse IDE

When you install the WDCS product package, you can choose to extend an Eclipse integrated development environment (IDE) already installed on your computer by adding the functions that the WDCS package contains.

The WDCS package that you install with IBM Installation Manager is bundled with a version of the Eclipse IDE or workbench; this bundled workbench is the base platform for providing the functionality in the Installation Manager package. However, if you have an existing Eclipse IDE on your workstation, then you have the option to extend, that is, add to the IDE the additional functionality provided in the WDCS package.

You might extend your existing Eclipse IDE, for example, because you want to gain the functionality provided in the WDCS package, but you also want to have the preferences and settings in your current IDE when you work with the functionality from the WDCS package. You also might want to work with plug-ins you have installed that already extend the Eclipse IDE.

To extend an existing Eclipse IDE: in the Location page of the Install Packages wizard, select the **Extend an existing Eclipse IDE** option.

Your existing Eclipse IDE must be version 3.2.1 or later, plus the latest updates from eclipse.org to be extended. Installation Manager checks that the Eclipse instance you specify meets the requirements for the installation package. You also require a JRE from the following Java development kit:

IBM 32-bit SDK for Windows, Java 2 Technology Edition, Version 5.0 service release 3;
Sun™ Java™ 2 Standard Edition 5.0 Update 9 for Microsoft Windows

Important: Choosing to extend an Eclipse IDE for installation supports some components of WDCS and WDCS AE.

5.2.3 Migration from previous WDSC versions

This section explains how to migrate your projects to WDSC Version 7.0 or WDSC AE Version 7.0 from previous versions of WDSC. Migration is supported from V5.1.2 and V6.x. Migration of projects earlier than V5.1.2 is not supported because of additional features that have been added.

Migration is required most often because of a change in the underlying technology from one release to another. For example, new standards in Web application directory structure may require reorganization of your projects from one release to another.

In general, migration is seamless and transparent. All you need to do is use your existing workspace and continue working with your projects with the new software version.

For some special cases, related to Java tools, Web tools, and customization of WebFacing projects, you might need to use a migration tool or do extra manual steps. In all cases, a message informs you when migration is necessary.

Uninstalling older product versions

The WDSC Version 7.0 installation process does not automatically uninstall earlier versions of the software. It is important to note, however that once you have migrated your workspace, you will no longer be able to open it in an older version of the Development Studio Client.

Locating your workspace in Version 5.1.2 or 6.x

To find your V5.1.2 or V6.x workspace, chose one of the following methods:

- ▶ If you still have the Development Studio Client V5.1.2 or V6.x installed:
 - To find the location of your current workspace, in the V5.1.2 or V6.x workbench, select **File** → **Switch workspace**. The default value in the dialog window is the directory of your current workspace location.
 - To find the location of an individual resource, such as a project:
 - i. In the V5.1.2 or V6.x workbench, switch to the **Navigator view**: click **Window** → **Show View** → **Other**. Expand **Basic** and select **Navigator**. Click **OK**.
 - ii. Right-click the name of the resource and select **Properties**.
 - iii. In the left pane, click **Info**.
- ▶ If you have already installed the WDSC V7.0 or WDSC AE V7.0:
 - If you accepted the default location when you installed V5.1.2 or V6.x, your workspace is in the directory:
X:\Documents & Settings\your userid\My Documents\IBM\wdsc\workspace, where X is the drive where your Windows installation directory is located.
 - If you did not accept the default location when you installed V5.1.2 or V6.x, search for a directory called .metadata. Every workspace directory contains a metadata directory; the directory in which .metadata appears is usually your workspace.

For more information refer to the migration documentation in the V7.0 online help for details: **Help** → **Help Contents** then look in the **WebSphere Development Studio Client for System i** → **Installing and migrating book**.

5.3 Installation of WDSC

The following steps describe installing the WDSC or WDSC AE package with the Installation Manager GUI.

5.3.1 Starting the installation process with your media of choice

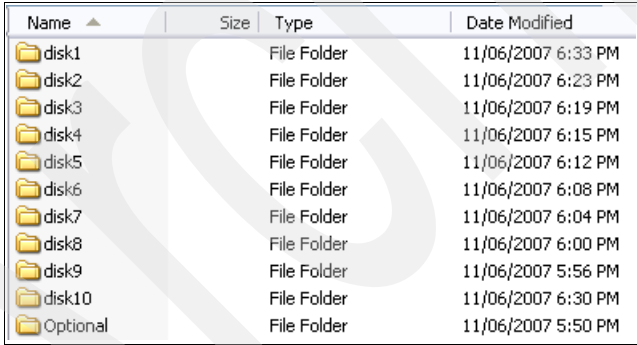
The following sections provide an overview of different installation scenarios.

Using CDs

The general steps for installing from CDs are:

1. Insert the first installation CD into your CD drive.
2. If autorun is enabled on your system, the IBM Rational Software Development Platform launchpad program automatically opens. If autorun is not enabled, start the launchpad program. To manually start up the launchpad, double-click the launchpad.exe on the first product CD.
3. Start the installation of WDSC from the launchpad program. For details, refer to “Starting from the launchpad” on page 89.
4. Follow the on-screen instructions in the Install Packages wizard to complete the installation. For details, refer to 5.3.2, “Step-by-step guide using Installation Manager” on page 89.
5. Install optional software that is included with WDSC. For more information, refer to 6.2, “Extending WDSC with optional software” on page 115.

You can also copy WDSC CDs to your local disk, and start WDSC installation. Figure 5-2 shows the structure of the local directory containing WDSC disks.



Name	Size	Type	Date Modified
disk1		File Folder	11/06/2007 6:33 PM
disk2		File Folder	11/06/2007 6:23 PM
disk3		File Folder	11/06/2007 6:19 PM
disk4		File Folder	11/06/2007 6:15 PM
disk5		File Folder	11/06/2007 6:12 PM
disk6		File Folder	11/06/2007 6:08 PM
disk7		File Folder	11/06/2007 6:04 PM
disk8		File Folder	11/06/2007 6:00 PM
disk9		File Folder	11/06/2007 5:56 PM
disk10		File Folder	11/06/2007 6:30 PM
Optional		File Folder	11/06/2007 5:50 PM

Figure 5-2 WDSC local directory disk structure

Using an electronic image on your workstation

The following are general steps in preparation for installing WDSC or WDSC AE from an electronic installation image.

Note: IBM Passport Advantage allows you to download an electronic image, which is a zip file. Entitled Software Support allows you to download ISO files, suitable for burning your own set of CDs. You can also extract the electronic image from the ISO file with an appropriate utility.

1. Ensure that your workstation has sufficient space to store the files you must download and the extracted installation image. For more information, refer to “Hardware” on page 77.

2. Download all required parts for the product image from IBM Passport Advantage or Entitled Software Support to a temporary directory.
3. Extract the installation image from the downloaded file—using a zip or ISO extraction utility—and verify that the installation image is complete.

Installing from an electronic image

The following are general steps for installing WDSC or WDSC AE from an electronic installation image.

1. Start the launchpad program: double-click the launchpad.exe inside the disk1 folder where the images have been uncompressed.
2. Start the installation of WDSC from the launchpad program. For details, refer to “Starting from the launchpad” on page 89.
3. Follow the on-screen instructions in the Install Packages wizard to complete the installation. For details, refer to 5.3.2, “Step-by-step guide using Installation Manager” on page 89.
4. Install optional software that is included with WDSC. For more information, refer to 6.2, “Extending WDSC with optional software” on page 115.

Using an electronic image on a shared drive

In this scenario, you will place the electronic image on a shared drive so that users in your enterprise can access the installation files for WDSC or WDSC AE from a single location.

The following steps are performed by the person who places the installation image on a shared drive.

1. Ensure that your shared drive has sufficient disk space to store both the files you must download from IBM Passport Advantage and the extracted installation image. For more information, refer to “Hardware” on page 77.
2. Download all required parts for the product image from IBM Passport Advantage to a temporary directory on the shared drive.
3. Extract the installation image from the downloaded files into an accessible directory on the shared drive and verify that the installation image is complete. See Verifying and extracting electronic images for details.

To install WDSC or WDSC AE from the installation files on the shared drive:

1. Change to the disk1 directory on the shared drive containing the installation image.
2. Follow the steps in “Installing from an electronic image” on page 88.

Using a repository on an HTTP Web server

In this scenario, the product packages are retrieved by IBM Installation Manager from an HTTP Web server.

These steps assume that the repository containing the package for WDSC or WDSC AE has been created on the HTTP Web server. For more details, refer to 6.4, “Placing WDSC on an HTTP server” on page 122.

To install the WDSC or WDSC AE package from a repository on an HTTP server:

1. Install IBM Installation Manager. For more information, refer to “Installing Installation Manager manually” on page 76.
2. Start Installation Manager. For more information, refer to “Starting Installation Manager manually” on page 76,

3. Set the URL of the repository containing the package of WDSC as a repository preference in Installation Manager. For more information, refer to “Setting repository preferences in Installation Manager” on page 84.
4. Start the Install Packages wizard in Installation Manager and follow the on-screen instructions in the Install Packages wizard to complete the installation. For details, refer to 5.3.2, “Step-by-step guide using Installation Manager” on page 89.
5. Install optional software that is included with WDSC. For more information, refer to 6.2, “Extending WDSC with optional software” on page 115.

5.3.2 Step-by-step guide using Installation Manager

This section provides a step-by-step guide for installing WDSC V7.0 and WDSC AE V7.0 from the launchpad.

Starting from the launchpad

If you are installing from a CD and autorun is enabled on your workstation, then the WDSC launchpad starts automatically when you insert the first installation disc into your CD drive. If you are installing from an electronic image, or if autorun is not configured on your workstation, then you must start the launchpad program manually.

To start the launchpad program:

1. Insert the IBM WDSC CD into your CD drive.
2. If autorun is enabled on your system, the IBM WDSC launchpad program automatically opens.
3. To manually start up the launchpad:
 - a. If you are installing from an electronic image, double-click the launchpad.exe inside the disk1 folder where the images have been uncompressed.

- b. If you are installing from the CD, double-click the launchpad.exe on the first product CD. The launchpad opens the dialog shown in Figure 5-3.



Figure 5-3 launchpad window for WDSC AE

4. You can choose your language from the Language selection drop-down list in the launchpad. For more information about installation steps and issues, refer to the View Release Information section in the launchpad.
5. If you are installing WDSC AE, click **Install IBM WebSphere Development Studio Client Advanced Edition for System i V7.0**. If you are installing WDSC, click **Install IBM WebSphere Development Studio Client for System i V7.0**.

If you have IBM installation Manager installed, it will start automatically with the Install Packages page. When Installation Manager starts, it searches its defined repositories for available packages. If you do not have the Installation Manager installed, follow the steps in the wizard to install it.

6. The Install page of the Install Packages wizard lists all the packages found in the repositories that Installation Manager searched; see Figure 5-4. If two versions of a package are discovered, only the most recent, or recommended, version of the package is displayed.
 - a. To display all versions of any package found by Installation Manager, click **Show all versions**.
 - b. To return to the display of only the recommended packages, uncheck **Show all versions**.

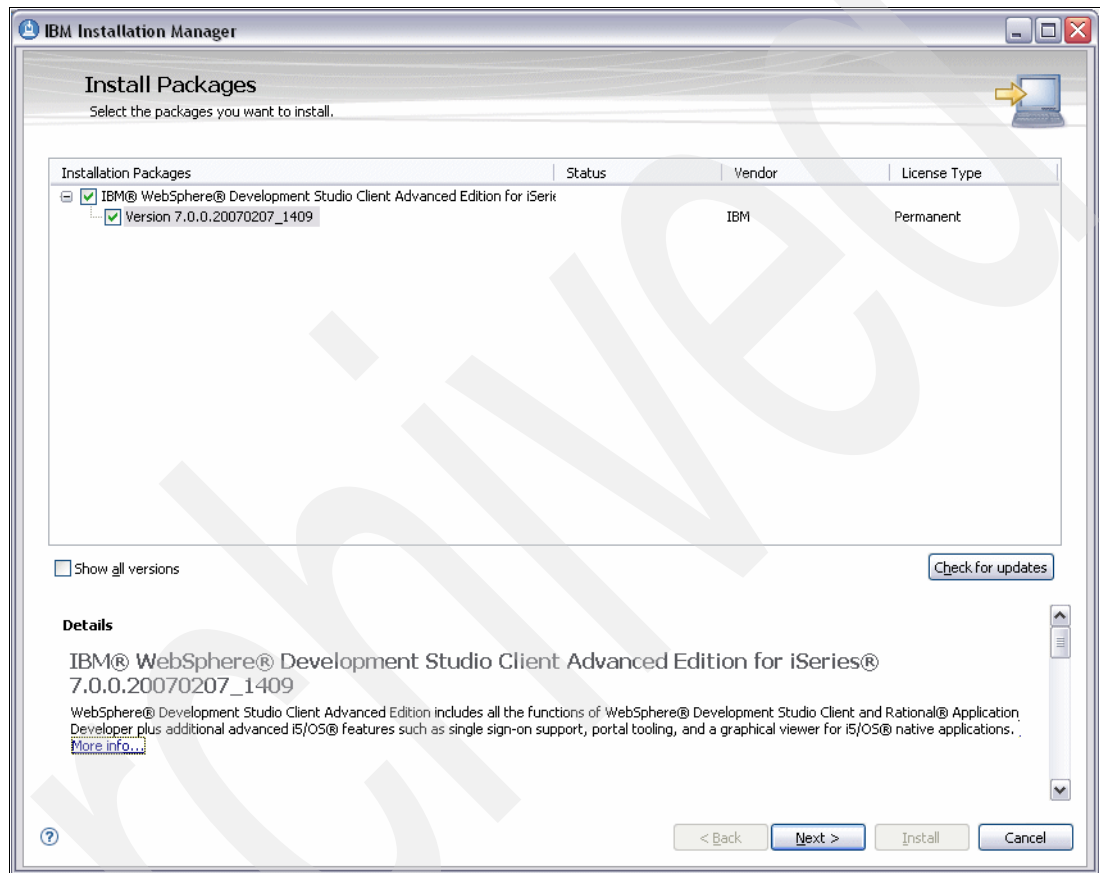


Figure 5-4 Installation Manager - Install Packages page

7. Click the WDSC package to display its description in the Details pane.
8. To search for updates to the WDSC package, click **Check for updates**.

Installation Manager searches for updates at the predefined IBM update repository for the product package. It also searches any repository locations that you have set. A progress indicator shows the search is taking place. You can install updates at the same time that you install the base product package.

If updates for the WDSC package are found, then they will be displayed in the Installation Packages list on the Install Packages page below their corresponding product. Only recommended updates are displayed by default.

Important: You need to have Internet access to be able to use the predefined IBM update repository locations. If your network uses a proxy server to control Internet access, select **File** → **Preferences** from the menu and expand **Preferences** to specify an FTP or HTTP proxy server.

Note: For Installation Manager to search the predefined IBM update repository locations for the installed packages, the preference “Search the linked repositories during installation and updates” on the Repositories preference page must be selected. The preference is selected by default.

9. Select the WDSC package and any updates to the package that you want to install. Updates that have dependencies are automatically selected and cleared together. Click **Next** to continue.

Note: If you install multiple packages at the same time, then all the packages will be installed into the same package group.

10. On the Licenses page, Figure 5-5, read the license agreement for the selected package. If you selected more than one package to install, there might be a license agreement for each package. On the left side of the License page, click each package version to display its license agreement. The package versions that you selected to install (for example, the base package and an update) are listed under the package name.
 - a. If you agree to the terms of all of the license agreements, click **I accept the terms of the license agreements**.
 - b. Click **Next** to continue.

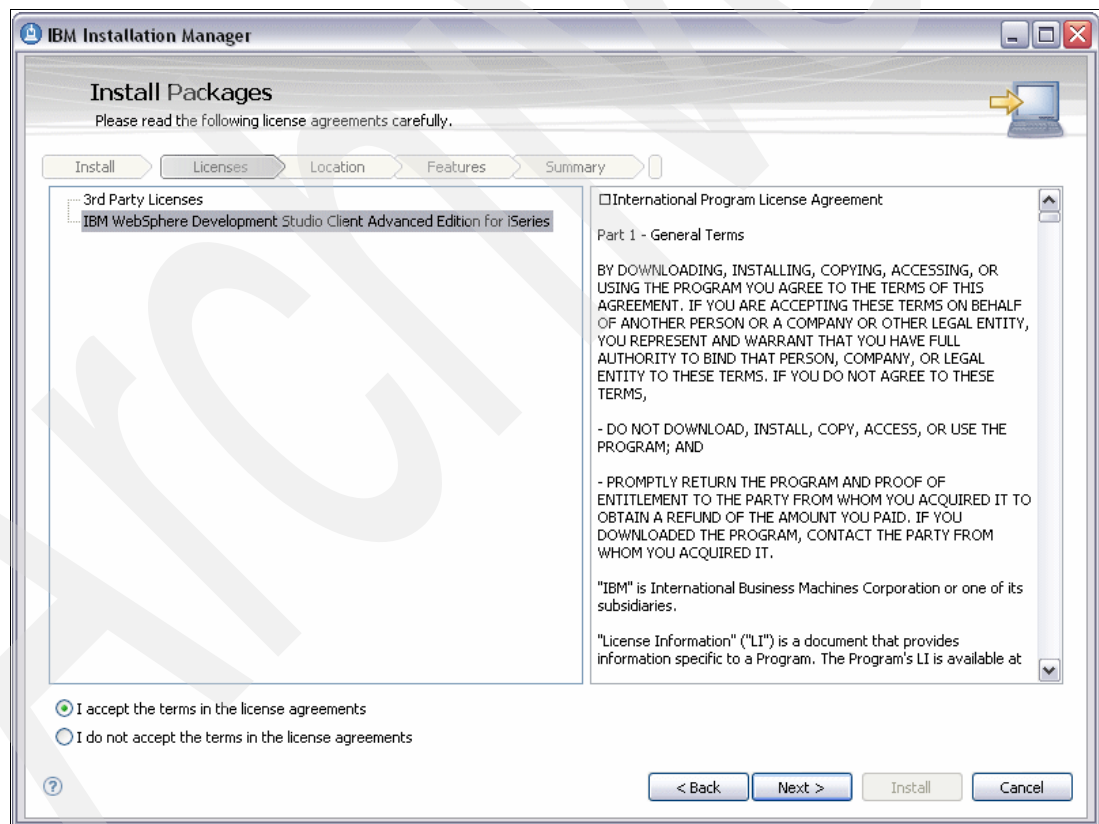


Figure 5-5 Installation Manager - License Agreement page

11. On the Location page, Figure 5-5, type the path for the shared resources directory in the Shared Resources Directory field; or, accept the default path. The shared resources directory contains resources that can be shared by one or more package groups. Click **Next** to continue.

The default path is: C:\Program Files\IBM\SDP70Shared.

Important: You can specify the shared resources directory only the first time that you install a package. Use your largest disk for this to help ensure adequate space for the shared resources of future packages. You cannot change the directory location unless you uninstall all packages.

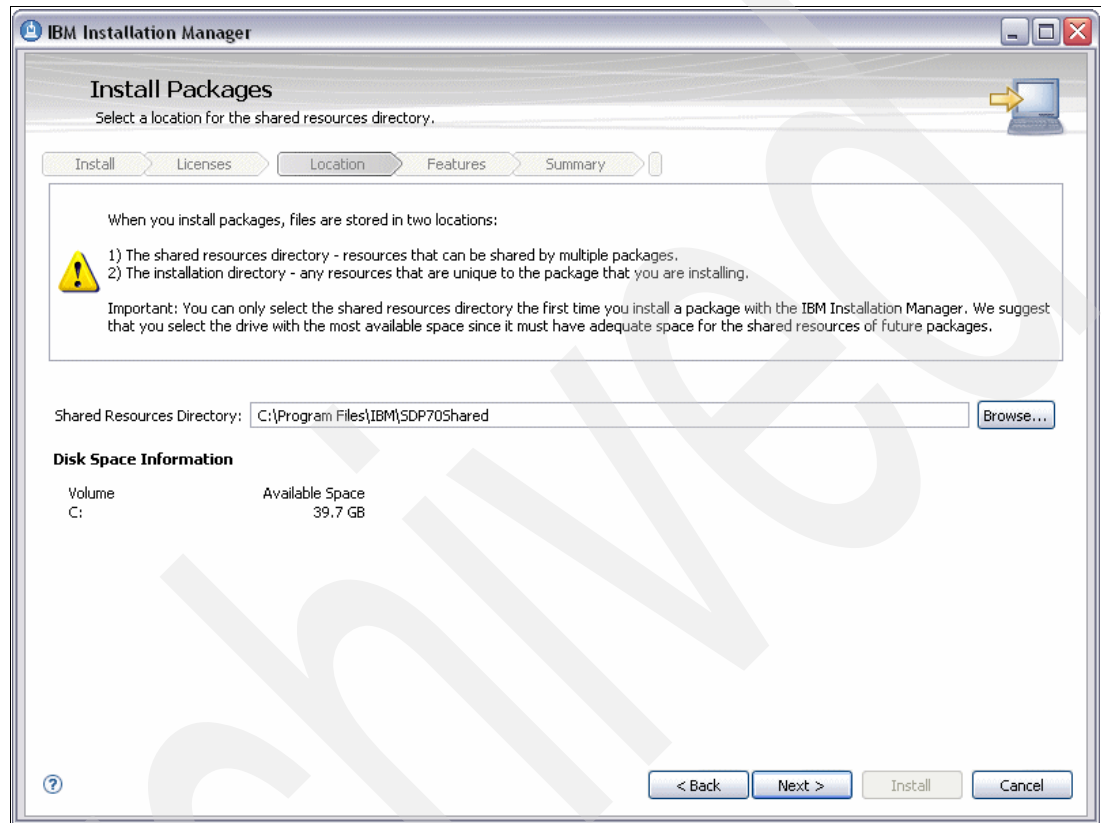


Figure 5-6 Location page - Shared resources directory

12. On the Location page, Figure 5-7, either choose an existing package group to install the WSC package into, or create a new one. A package group represents a directory in which packages share resources with other packages in the same group. To create a new package group:
 - a. Click **Create a new package group**.
 - b. Type the path for the installation directory for the package group. The name for the package group is created automatically.
The default path is: C:\Program Files\IBM\SDP70.
 - c. Click **Next** to continue.

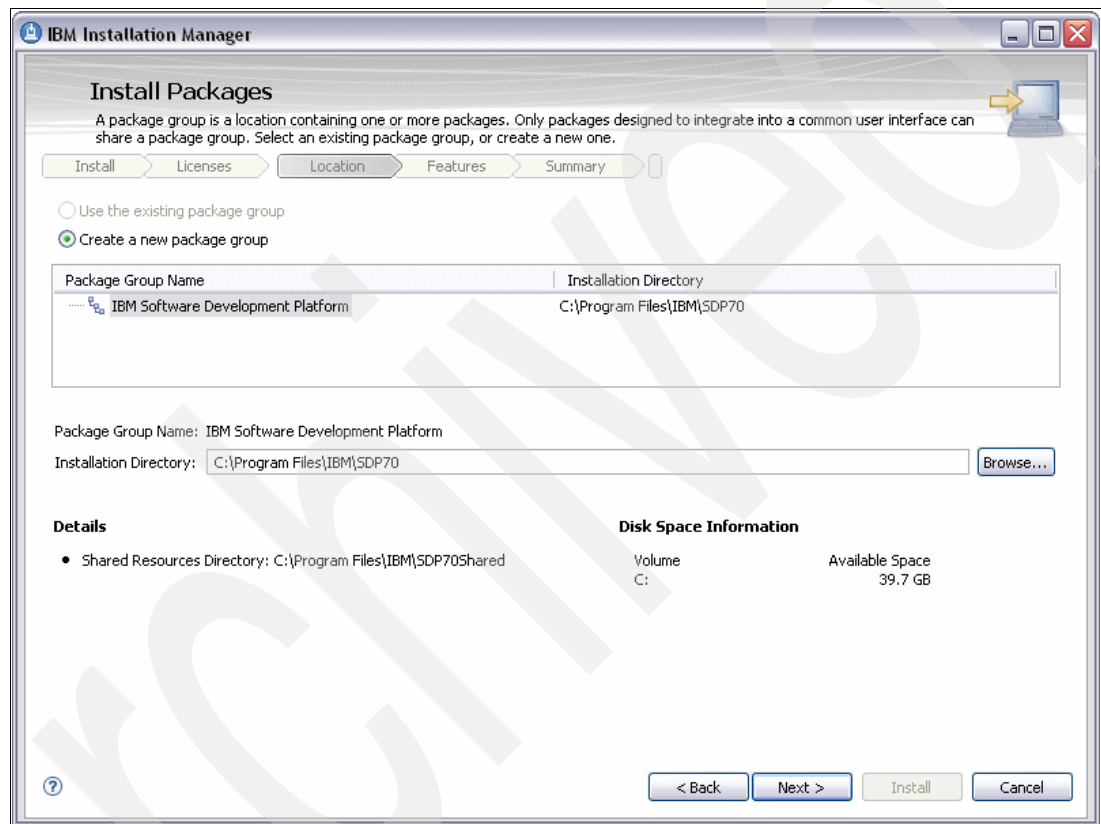


Figure 5-7 Location page - Package group

13. On the next Location page, Figure 5-8, you can choose to extend an existing Eclipse IDE already installed on your system, adding the functionality in the packages that you are installing. You must have Eclipse Version 3.2.1 or higher to select this option.
 - a. If you do not want to extend an existing Eclipse IDE, click **Next** to continue.
 - b. To extend an existing Eclipse IDE:
 - i. Select **Extend an existing Eclipse**.
 - ii. In the Eclipse IDE field, type or navigate to the location of the folder containing the eclipse executable file (eclipse.exe or eclipse.bin). Installation Manager will check if the Eclipse IDE version is valid for the package that you are installing. The Eclipse IDE JVM™ field displays the Java Virtual Machine (JVM) for the IDE that you specified.
 - iii. Click **Next** to continue.

Important: Choosing to extend an Eclipse IDE for installation supports some components of WDS and WDS AE.

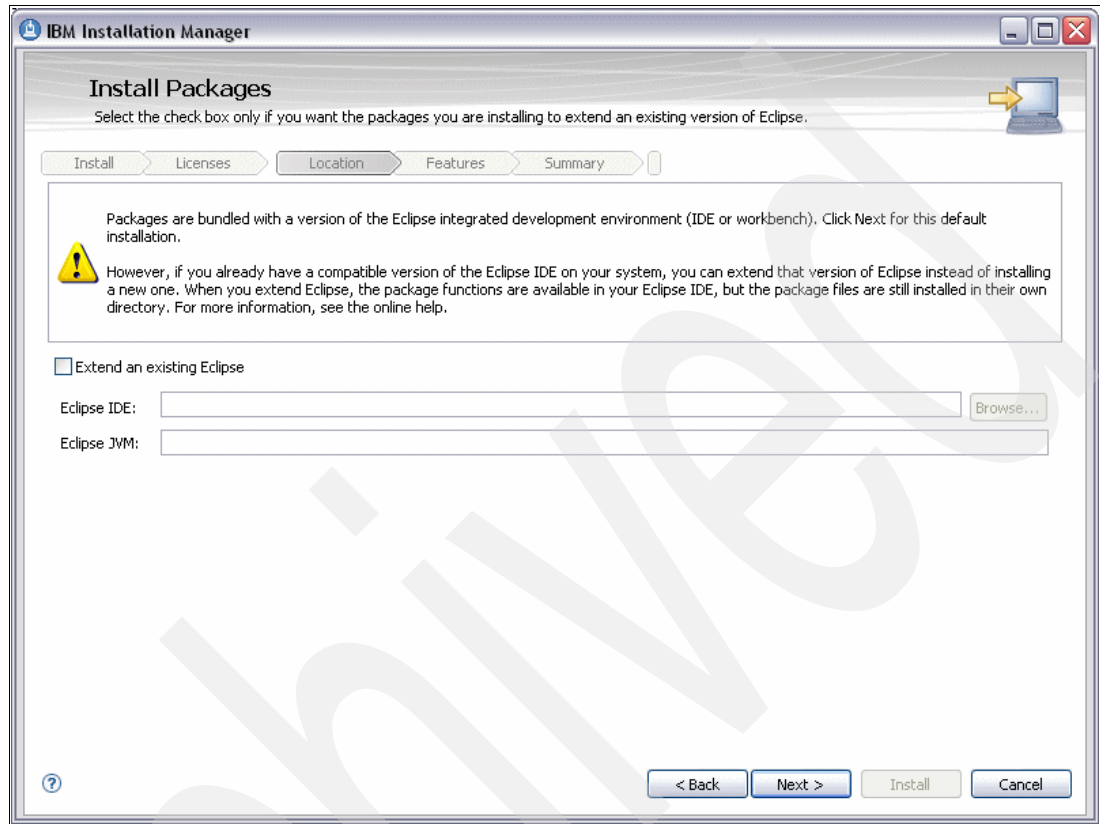


Figure 5-8 Location page - Extend an existing Eclipse

14. On the Features page under Languages, Figure 5-9, select the languages for the package group. The corresponding national language translations for the user interface and documentation for the WDSC package will be installed. Note that your choices apply to all packages installed under this package group.

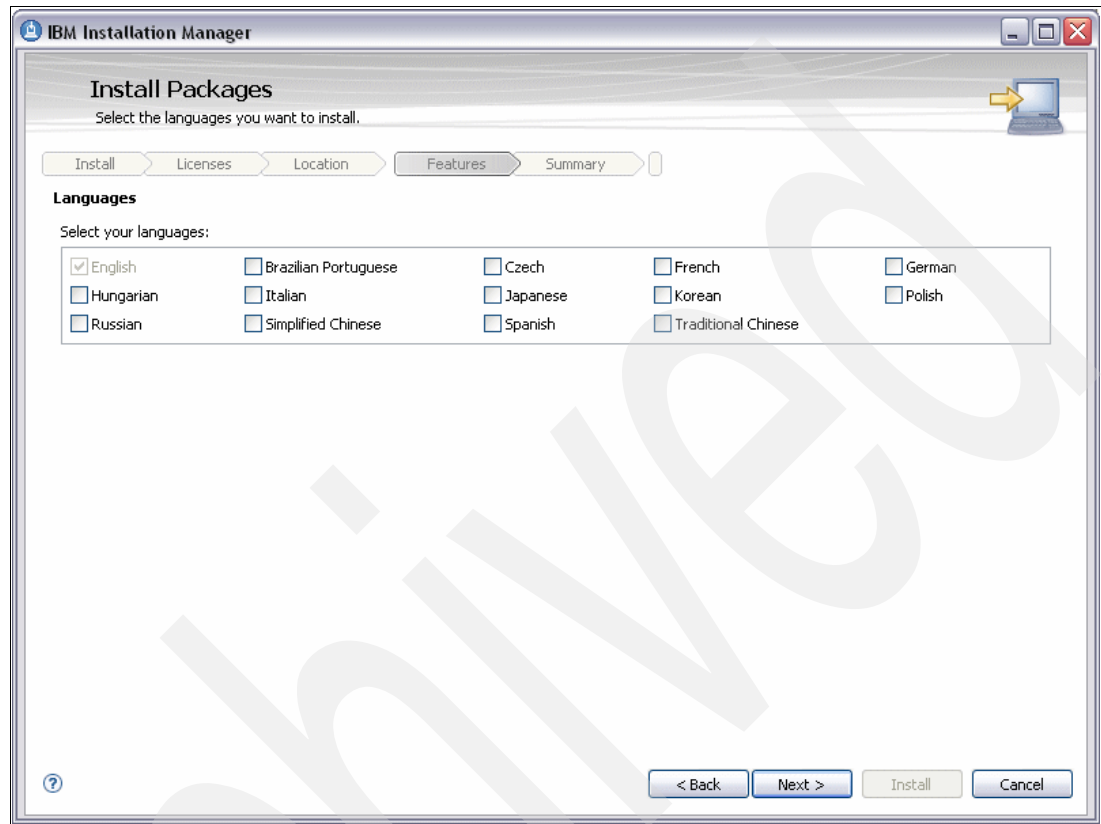


Figure 5-9 Features page - languages selection

15. On the next Features page, Figure 5-10, select the package features that you want to install. For more details about the features, refer to the Features section in *WDSC Installation Guide*. The latest version of the Installation Guide is available at:

http://download.boulder.ibm.com/ibmdl/pub/software/awdtools/wdscae/v7/70/documents/install_instruction/install.html

- a. Optional: To see the dependency relationships between features, select **Show Dependencies**. This can be important to understand why a certain component is selected automatically. For example, if you select the IBM WebFacing Tool, the Web Development must also be installed.
- b. Optional: Click a feature to view its brief description under Details.
- c. Select or clear features in the packages. Installation Manager will automatically enforce any dependencies with other features and display updated download size and disk space requirements for the installation.
- d. When you are finished selecting features, click **Next** to continue.

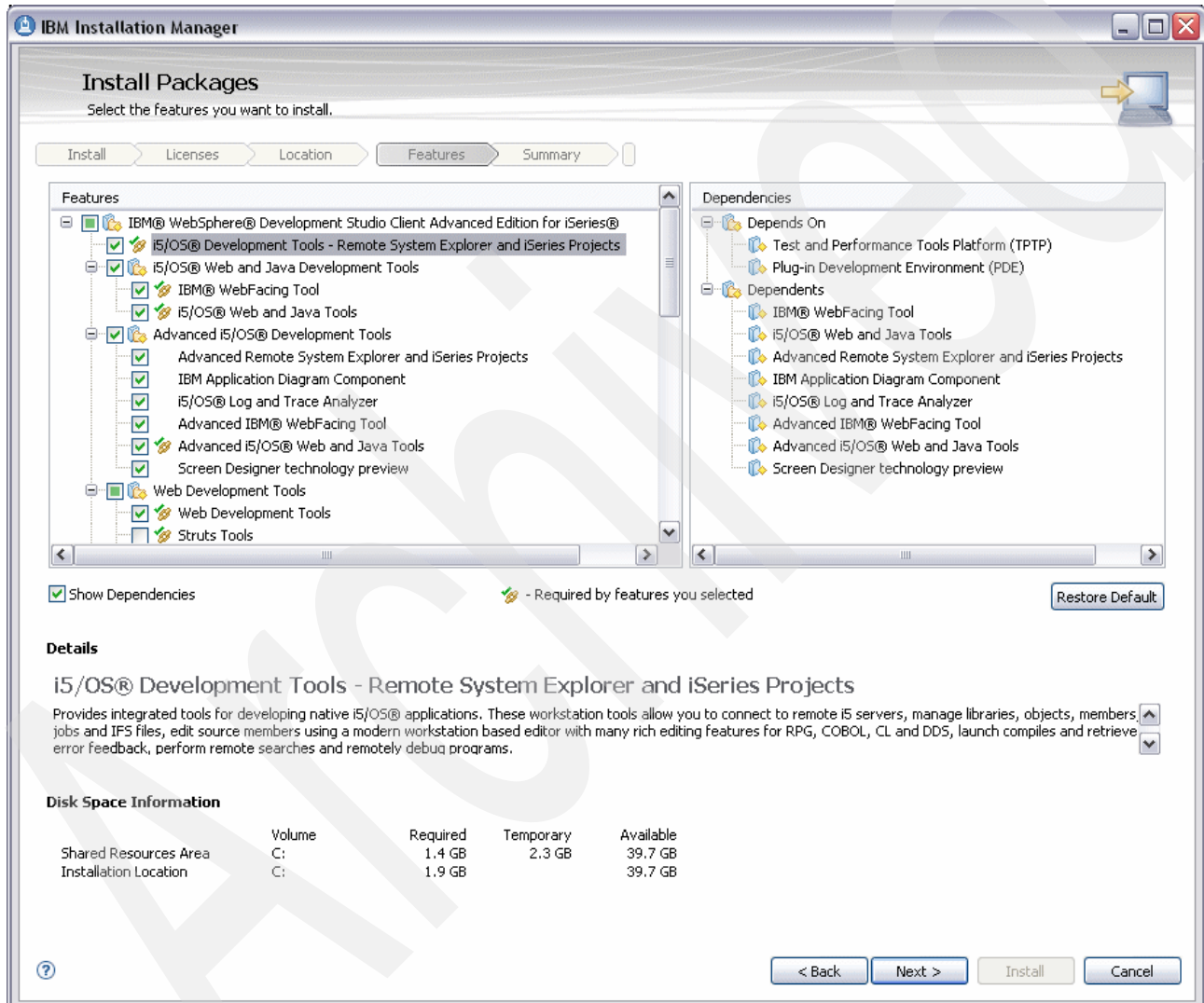


Figure 5-10 Features page - Select package features

16. On the Summary page, Figure 5-11, review your choices before installing the WDSC package. If you want to change the choices you made on previous pages, click **Back** and make your changes. When you are satisfied with your installation choices, click **Install** to install the package. A progress indicator shows the percentage of the installation completed.

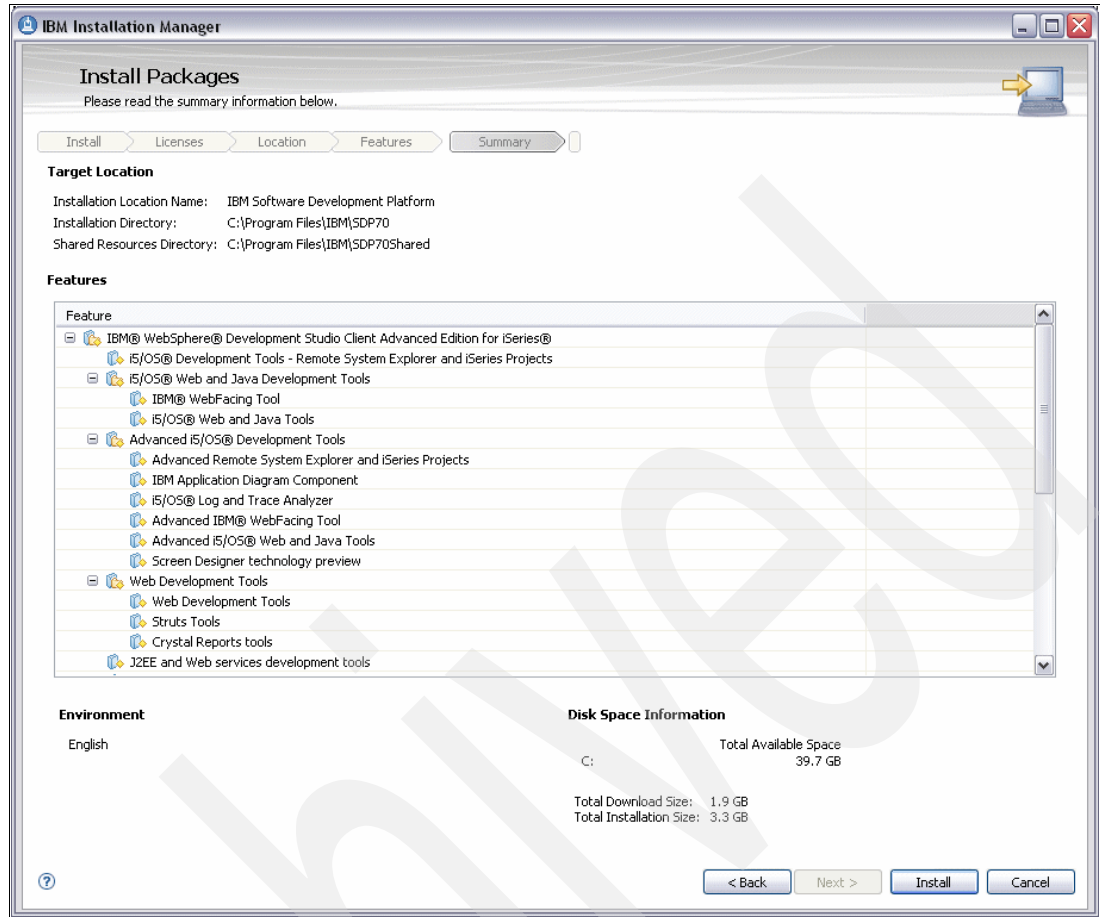


Figure 5-11 Installation summary page

17. When the installation process is completed, a message confirms the success of the process.

- a. Click **View log file** to open the installation log file for the current session in a new window. You must close the Installation Log window to continue.
- b. In the Install Package wizard, select whether you want IBM WebSphere Development Studio Client to start when you exit.
- c. Click **Finish** to launch the selected package. The Install Package wizard closes and you are returned to the Start page of Installation Manager.

5.3.3 Various installation options

Installing a lightweight option of WDSC

With Installation Manager you can customize your product by selecting which features of WDSC to install. The following steps describe installing a lightweight edit/compile/debug environment for the WDSC or WDSC AE package with the Installation Manager GUI in addition to a full (or bigger) version you already installed earlier.

1. Repeat steps 1 to 15 in “Step-by-step guide using Installation Manager” on page 89.

Important: Choose an installation location where another package group is not already installed for installing a lightweight option of WDSC. Placing a complete package of WDSC and a lightweight option of WDSC in the same installation location would result in resource sharing, thus, doing so would cause the lightweight option of WDSC to have more than just the minimum set of System i tools.

2. On the next Features page, deselect all the package features selected by default, except for the i5/OS Development Tools - Remote System Explorer and System i Projects feature. Click **Next** to continue.
3. On the next Features page, click **Next**.
4. On the Summary page, review your choices before installing the WDSC package. If you want to change the choices you made on previous pages, click **Back** and make your changes. When you are satisfied with your installation choices, click **Install** to install the package. A progress indicator shows the percentage of the installation completed.
5. When the installation process is completed, a message confirms the success of the process.
 - a. Click **View log file** to open the installation log file for the current session in a new window. You must close the Installation Log window to continue.
 - b. In the Install Package wizard, select whether you want WebSphere Development Studio Client to start when you exit.
 - c. Click **Finish** to launch the selected package. The Install Package wizard closes and you are returned to the Start page of Installation Manager.

Installing silently

You can install the WDSC product package by running Installation Manager in silent installation mode. When you do this in silent mode, the user interface is not available; instead, Installation Manager uses a response file to input the commands that are required to install the product package.

Running Installation Manager in silent mode is helpful because it enables you to use a batch process to install, update, modify and uninstall product packages through scripts.

There are two main tasks required for silent installation:

1. Create the response file.
2. Run Installation Manager in silent installation mode.

Creating a response file

You can create a response file by recording your actions as you install a WDSC product package using Installation Manager. When you record a response file, all of the selections that you make in the Installation Manager GUI are stored in a response file. When you run Installation Manager in silent mode, Installation Manager uses the response file to locate the repository that contains the package, to select the features to install, and so on.

Refer to “Sample response file” on page 100 for a sample response file.

Running Installation Manager in silent installation mode

You can run Installation Manager in silent installation mode from a command line.

You run Installation Manager in silent mode appending the **-silent** argument to the Installation Manager start command **launcher.bat** as follows:

```
launcher.bat -silent [arguments]
```

You can find the launcher.bat file in the eclipse folder of the Installation Manager's install directory. Refer to "Sample response file" on page 100 for a sample response file.

Table 5-1 describes the arguments used with the silent installation command.

Table 5-1 Argument for the Installation Manager start command launcher.bat

Argument	Description
-silent	Directs Installation Manager to run in silent mode (without the Installation Manager user interface).
-input	Specifies an XML response file as the input to Installation Manager. A response file contains commands that Installation Manager runs.
-log	(Optional) Specifies a log file that records the result of the silent installation. The log file is an XML file.

To run Installation Manager in silent installation mode:

1. On a command line, change to the eclipse subdirectory in the directory where you installed Installation Manager. For example:

```
cd C:\Program Files\IBM\Installation Manager\eclipse
```
2. Enter and run the following command, substituting your own locations for the response file and, optionally, the log file:

```
launcher.bat -silent -input c:/temp/responsefile.xml -log c:/temp/mylog.log
```

When the installation Manager runs in silent installation mode, it reads the response file and writes a log file to the directory you specified. While you must have a response file when running in silent installation mode, log files are optional. The result of this execution should be a status of 0 on success and non-zero on failure.

You can silently search for and install updates for all available products:

1. On a command line, change to the eclipse subdirectory in the directory where you installed Installation Manager.
2. Enter and run the following command, substituting your own locations for the response file and, optionally, the log file:

```
launcher.bat -silent -installAll
```

All available products known to Installation Manager are installed.

You can silently search for and install updates for all currently installed products:

1. On a command line, change to the eclipse subdirectory in the directory where you installed Installation Manager.
2. Enter and run the following command, substituting your own locations for the response file and, optionally, the log file:

```
launcher.bat -silent -updateAll
```

All available product updates known to Installation Manager are installed.

Sample response file

You can use an XML-based response file to specify predefined information such as silent installation preferences, repository locations, installation profiles, and so on. Response files

are beneficial for teams and companies that want to install installation packages silently and to standardize the locations and preferences for installation packages, such as the one shown in Example 5-1.

Example 5-1 Sample response file

```
<agent-input>

<!-- add preferences -->
<preference name="com.ibm.cic.common.core.preferences.http.proxyEnabled"
value="c:/temp"/>

<!-- create the profile if it doesn't exist yet -->
<profile id="my_profile" installLocation="c:/temp/my_profile"></profile>

<server>
<repository location=
"http://a.site.com/local/products/sample/20060615_1542/repository/"></repository>
</server>

<install>
<offering profile="my_profile" features="core" id="ies"
version="3.2.0.20060615">
</offering>
</install>
```

Silent Installation log files

You can use silent installation log files to examine the results of a silent installation session.

The silent installation function creates an XML-based log file that records the result of the silent install execution (as long as a log file path is specified using `-log <your log file path>.xml`). If your silent installation session is successful, the log file will contain just the root element of `<result> </result>`. However, if errors occur during the installation, the silent install log file will contain error elements with messages such as shown in Example 5-2.

Example 5-2 Installation error message

```
<result>
  <error> Cannot find profile: profile id</error>
  <error> some other errors</error>
</result>
```

For detailed analysis, you can look at the logs generated in the Installation Manager data area. By using a preference command, you can optionally set the data area to your preferred location, as shown in the response file topic.

5.4 Starting WDCS for the first time

You can start WDCS or WDCS AE from the desktop environment or a command line interface.

To start WDCS V7.0 from the desktop environment, select **Programs** → **IBM Software Development Platform** → **WebSphere Development Studio Client for System i** from the Start menu.

To start WDSC V7.0 from the command line, enter the following command:

```
<installation directory>\eclipse.exe -product com.ibm.etools.System  
i.wdsc.welcome.ide
```

To start WDSC AE V7.0 from the desktop environment, select **Programs** → **IBM Software Development Platform** → **WebSphere Development Studio Client Advanced Edition for System i** from the Start menu.

To start WDSC AE V7.0 from the command line, enter the following command:

```
<installation directory>\eclipse.exe -product com.ibm.etools.System  
i.wdsc.ae.welcome.ide
```

Note: The first time you start WDSC version 7.0 it defaults to a new workspace. Do not open your existing workspace in WDSC version 7.0 until you read the migration documentation. You can access the migration guide in the online help in the workbench **Help** → **Help Contents** → **Installing and migrating**.

For a description of Eclipse Startup Parameters, see the Workbench Online Help by selecting **Help** → **Help Contents** from the main menu and then **Rational Developer product documentation** → **Introduction to Rational Developer product documentation** → **Get started with the workbench** → **Tasks** → **Running Eclipse**. Then page down to Advanced Topics in Running Eclipse.

5.5 Welcome

The Welcome page, as shown in Figure 5-12, is the first page you see when you launch WDSC V7.0 and WDSC AE V7.0 for the first time, after you selected the workspace. Its purpose is to introduce you to WDSC, its features and tutorials, and to guide you through some basic tasks, with samples to get you started.

The default Welcome page includes:

- ▶ Overview
- ▶ What's New
- ▶ First Steps
- ▶ Samples
- ▶ Tutorials
- ▶ Web Resources
- ▶ Migrate
- ▶ Workbench
- ▶ Enable Roles section

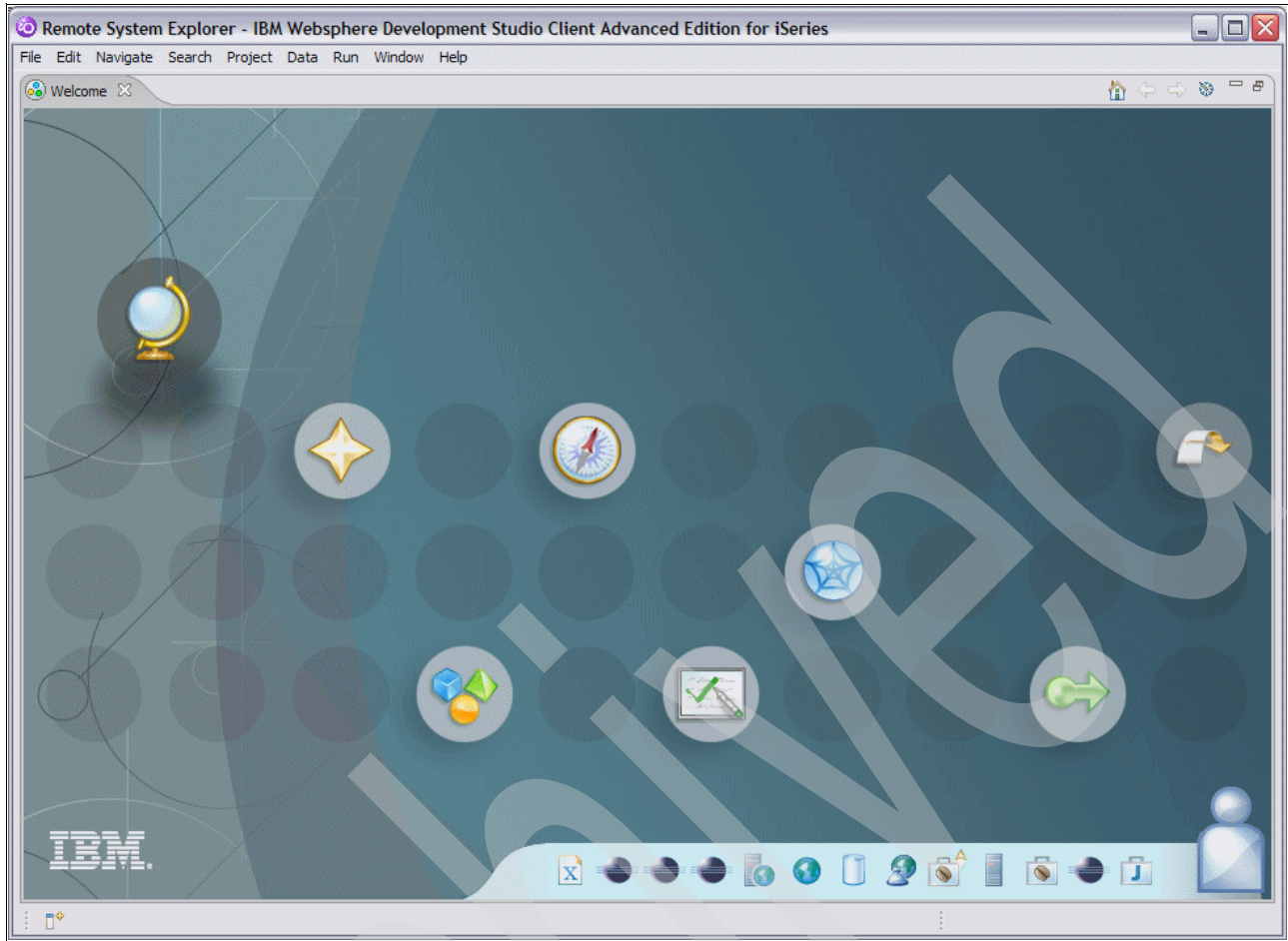


Figure 5-12 WDSC Welcome page

To go to the workbench from the Welcome page:

- ▶ Select the Workbench icon in the Welcome page; see Figure 5-13. This closes the Welcome page and opens the workbench default perspective.

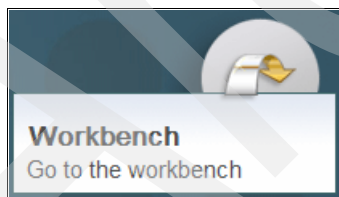


Figure 5-13 Go to the workbench

Note: You can open the Welcome page, at any time, from the workbench menu **Help** → **Welcome**.

Customizing the Welcome appearance

You can customize the Welcome appearance:

1. From the workbench menu, select **Window** → **Preferences**.
2. In the preferences dialog, select **General** → **Welcome**; see Figure 5-14.

3. You can use this preference page to select one of the predefined themes, which affects the overall look of the Welcome.
4. You can also select which pages will be displayed, and the visibility, layout, and priority of the items within each page.

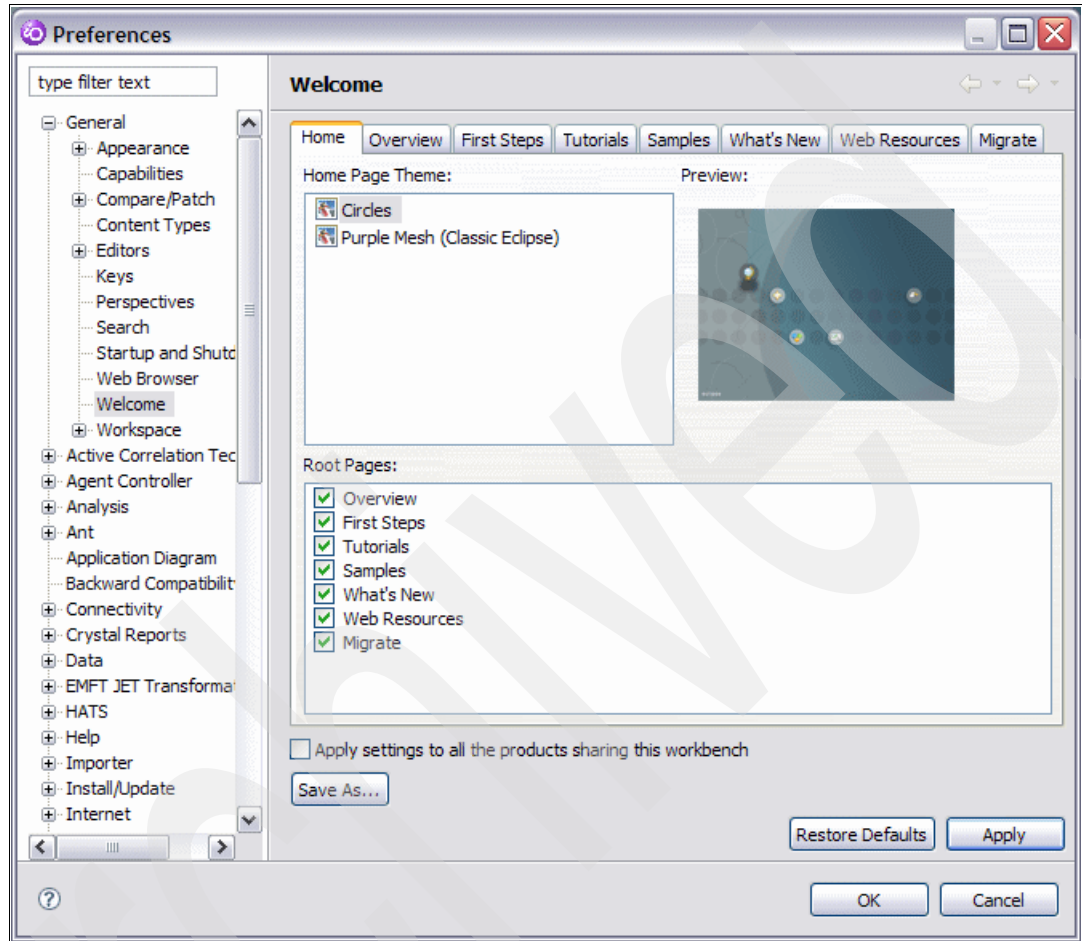


Figure 5-14 Customizing the Welcome page

Roles and capabilities

The WDSC user interface is composed of menus, toolbars, perspectives, views of information within the perspectives, wizards, and contextual help. To reduce workspace clutter, a limited set of these tools is enabled by default when you start the product. To support this streamlining, all the product features are grouped into logical sets called roles and capabilities.

As you work, you can add more tools by activating additional roles that enable additional product capabilities.

You can enable roles from the Welcome home page. To display the menu of available capabilities, click the **Enable Roles** icon in the lower-right corner of the Welcome home page; see Figure 5-15.

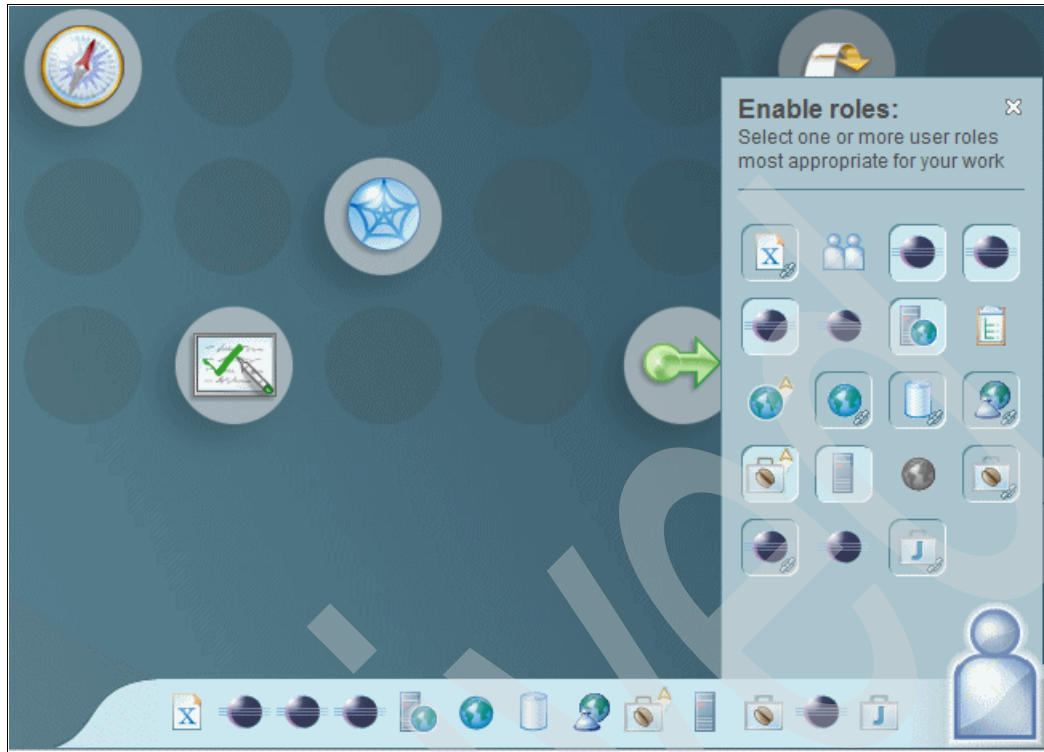


Figure 5-15 Enable Roles from Welcome

You can also find a list of capability options from **Window** → **Preferences**, and then click **General** → **Capabilities**, as shown in Figure 5-16.

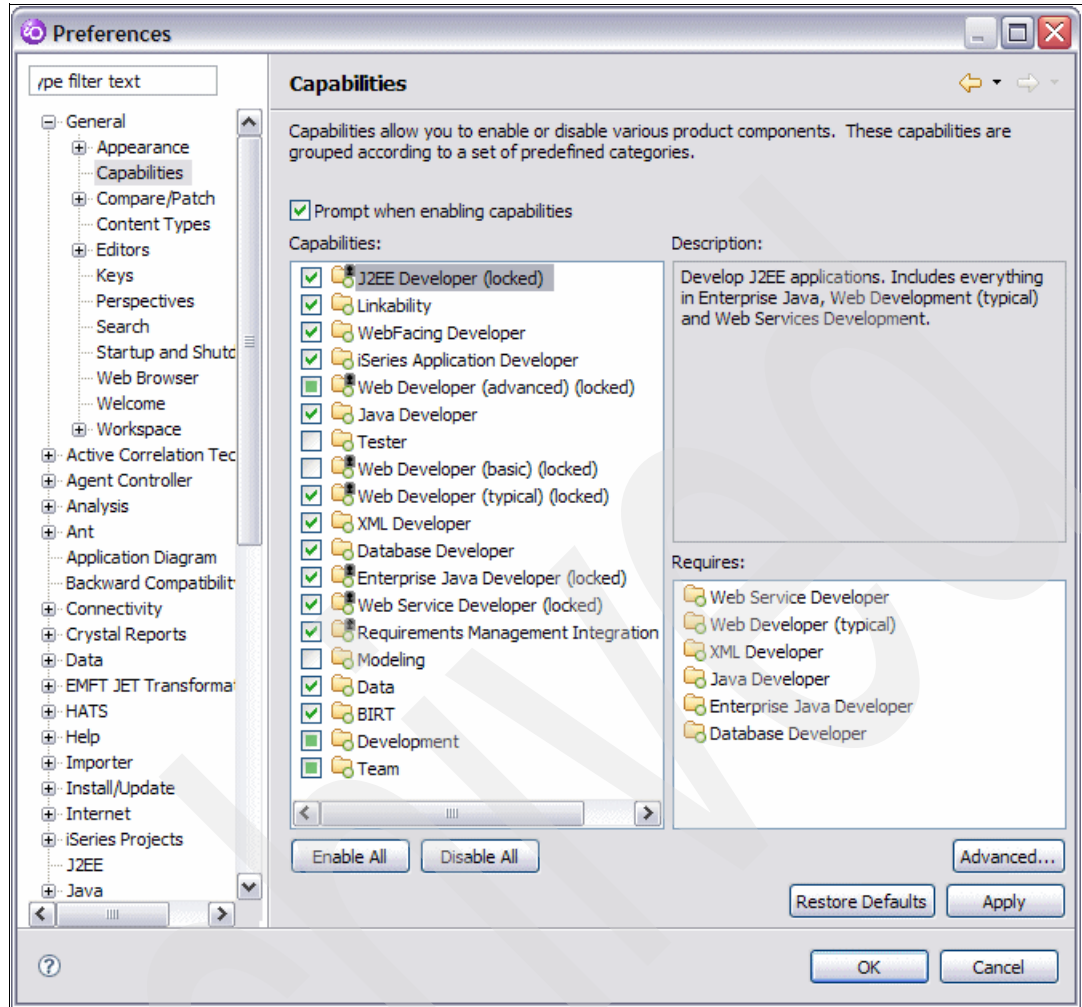


Figure 5-16 Enable/disable WDS components

To enable tools by using the predefined roles, select the checkboxes for the capabilities you want. In many cases, there are dependencies between capabilities, so enabling or disabling a capability automatically enables or disables other capabilities.



Managing your WDSC instance

This chapter describes how you can maintain your version of WDSC after you install it as described in 5.1, “Overview of the installation process” on page 76. It includes how to install optional software, such as the HATS toolkit, CODE, for example, as well as adding or removing WDSC components after you first install the product. We also explain how you can acquire and install fix packs and create packages to be installed by other team members in your organization.

6.1 Updating and installing fixes for WDSC

You can install updates for WDSC packages that were installed from IBM Installation Manager. An update might be a released fix, or a new feature of the product. Details of the contents of the update are provided in the wizard. You can choose whether to apply an update or not.

By default, Internet access is required unless your repository preferences points to your local update site. See the Installation Manager help for more information.

Each installed package has the location embedded for its default IBM update repository. For Installation Manager to search the IBM update repository locations for the installed packages, the preference **Search the linked repositories during installation and updates** on the Repositories preference page must be selected. This preference is selected by default.

To find the version of the installed packages and fixes:

1. Run IBM Installation Manager.
2. Select **File** → **View Installed Packages**; see Figure 6-1.

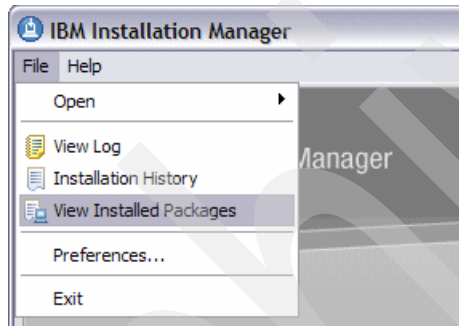


Figure 6-1 WDSC version - View Installed Packages

3. In the Installed packages and fixes list, select your package. The Details section contains information about the selected package; see Figure 6-2.

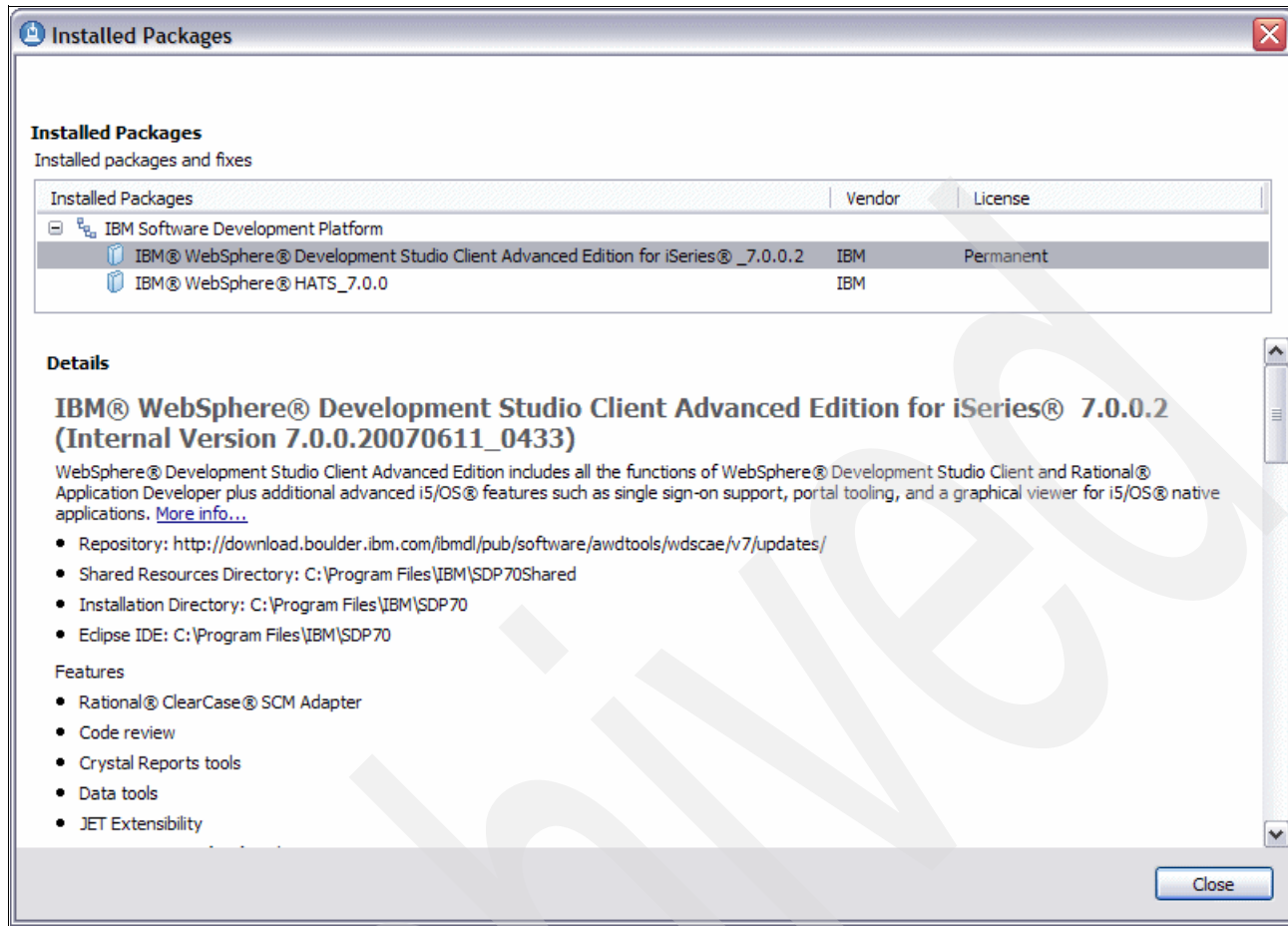


Figure 6-2 WDSC installed packages and fixes version

4. Click **Close** to close the Installed Packages window.

To find and install product WDSC updates:

Note: You have to close all programs that were installed using Installation Manager before updating WDSC.

1. From the Start page of the Installation Manager, click **Update Packages**; see Figure 6-3.

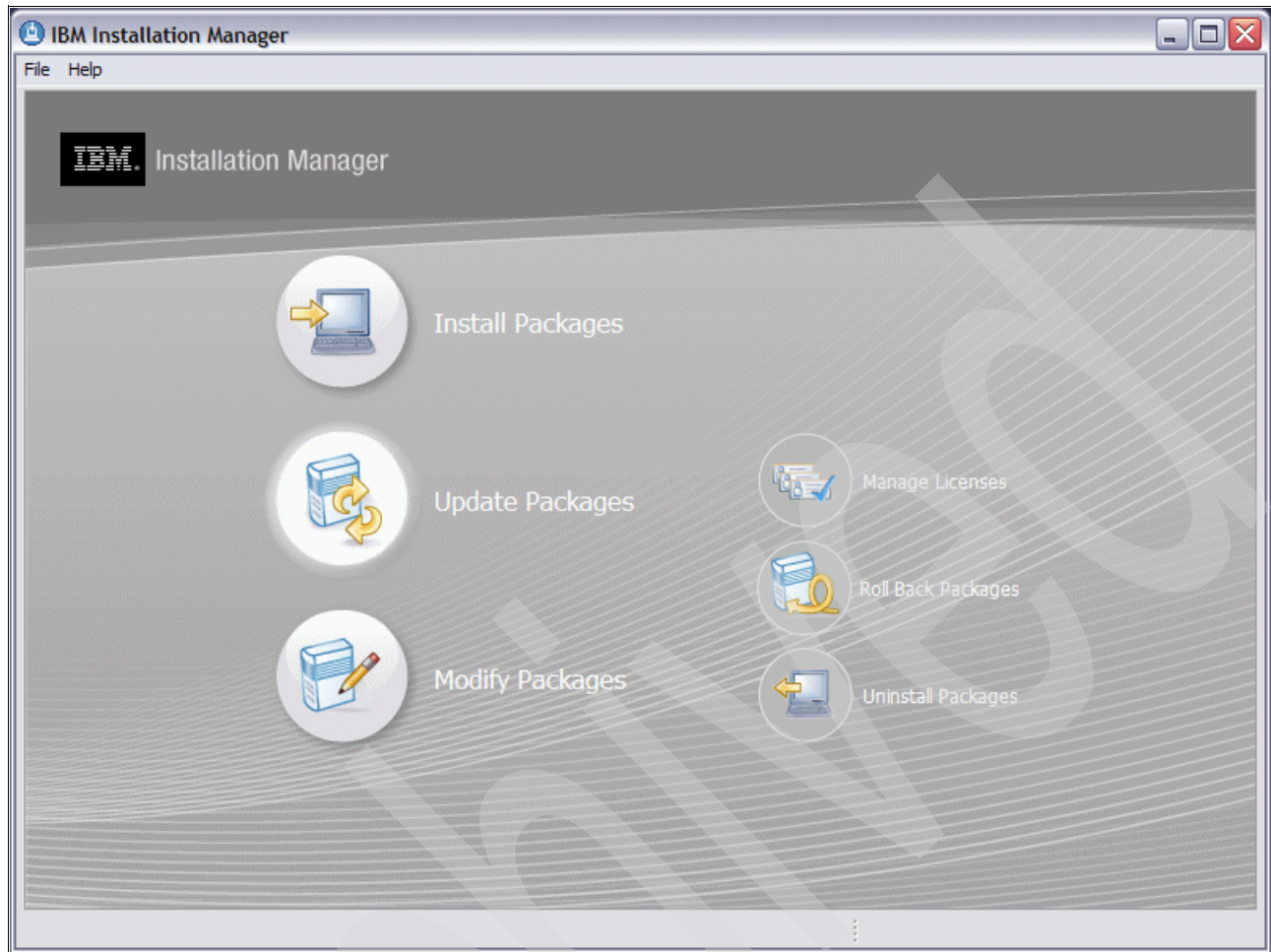


Figure 6-3 Update packages from Installation Manager

2. If a new version of Installation Manager is found, you are prompted to confirm that you want to install it before you can continue. For details about the current version and the updated version click **Details**; see Figure 6-4. Click **Yes** to proceed. Installation Manager automatically installs the new version, stops, restarts, and resumes.

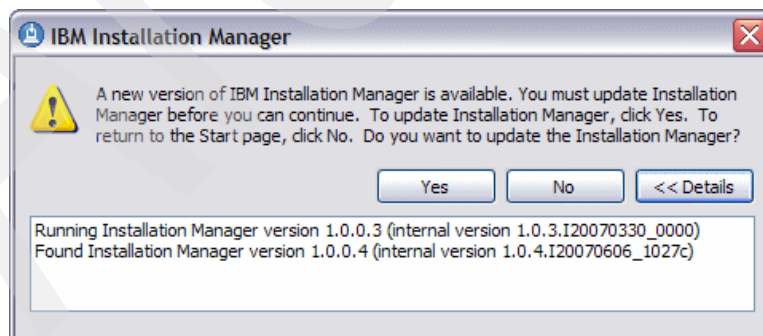


Figure 6-4 Updating Installation Manager version

3. In the Update Packages wizard, select the location where the WDSC product package is installed or select the **Update All** check box, and then click **Next**. Installation Manager searches for updates in its repositories and the predefined update sites for WDSC. A progress indicator shows the search is taking place.

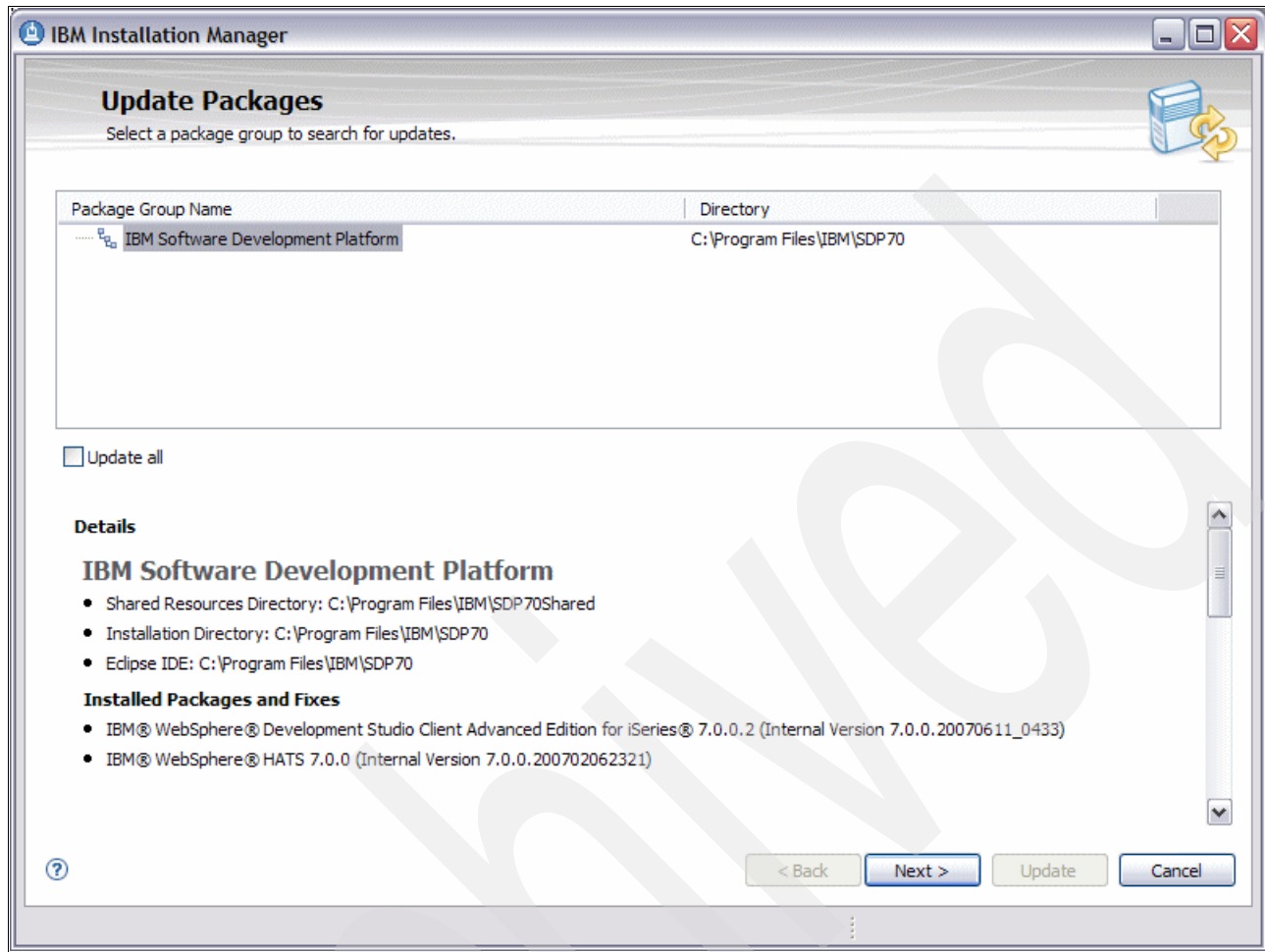


Figure 6-5 Update packages - select package to search for updates

4. If updates for a package are found, they are displayed in the Updates list on the Update Packages page below their corresponding package. Only recommended updates are displayed by default. Click **Show All** to display all available updates found for the available packages; see Figure 6-6.
 - To learn more about an update, click the update and review its description under Details.
 - If additional information about the update is available, a More info link is included at the end of the description text. Click the link to display the information in a browser. Review this information before installing the update.

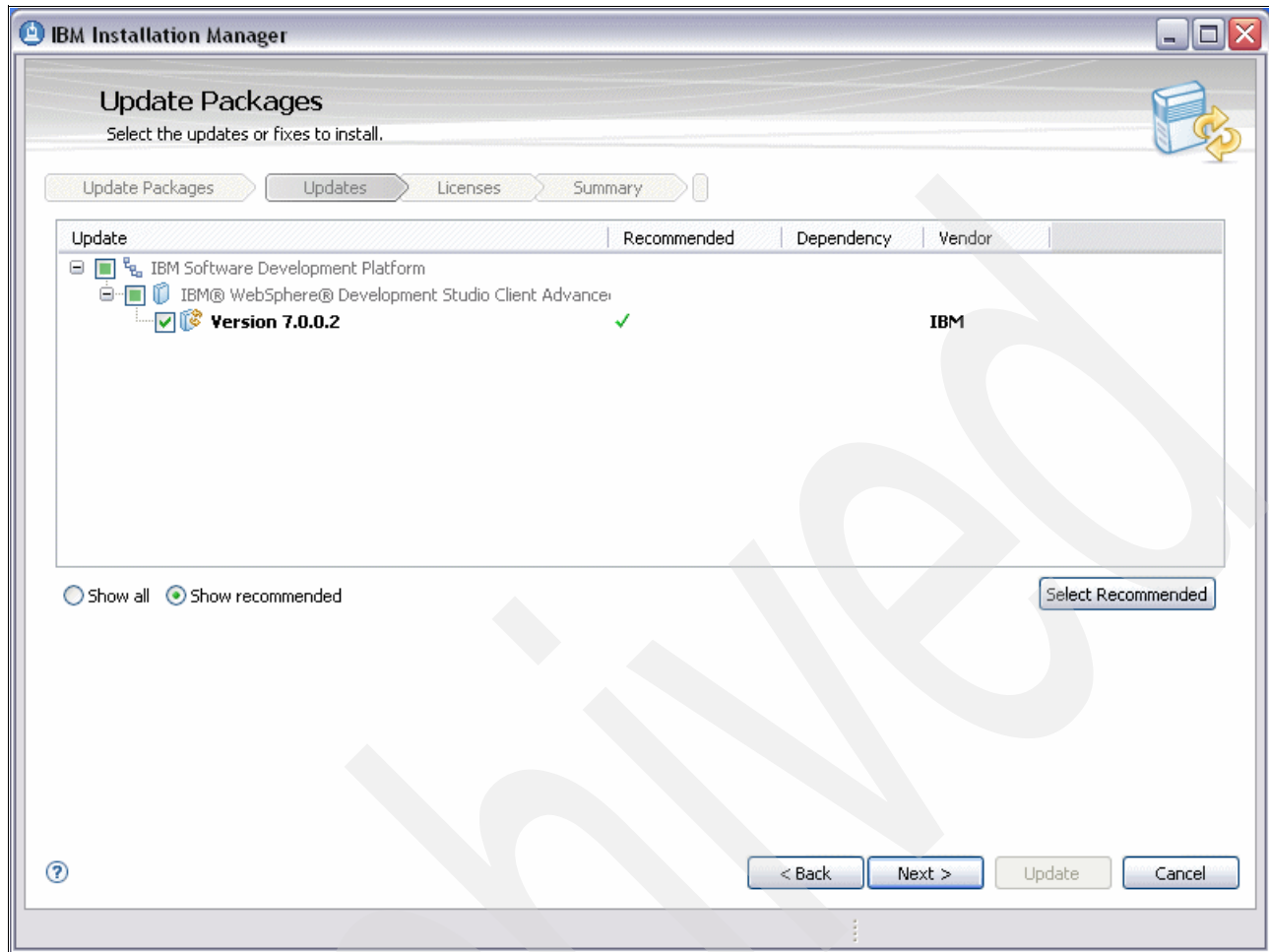


Figure 6-6 Update packages - package update list

5. Select the updates that you want to install or click **Select Recommended** to restore the default selections. Updates that have a dependency relationship are automatically selected and cleared together.
6. Click **Next** to continue.
7. On the Licenses page, read the license agreements for the selected updates. On the left side of the License page, the list of licenses for the updates you selected is displayed; click each item to display the license agreement text.
 - d. If you agree to the terms of all the license agreements, click **I accept the terms of the license agreements**, as shown in Figure 6-7.
 - e. Click **Next** to continue.

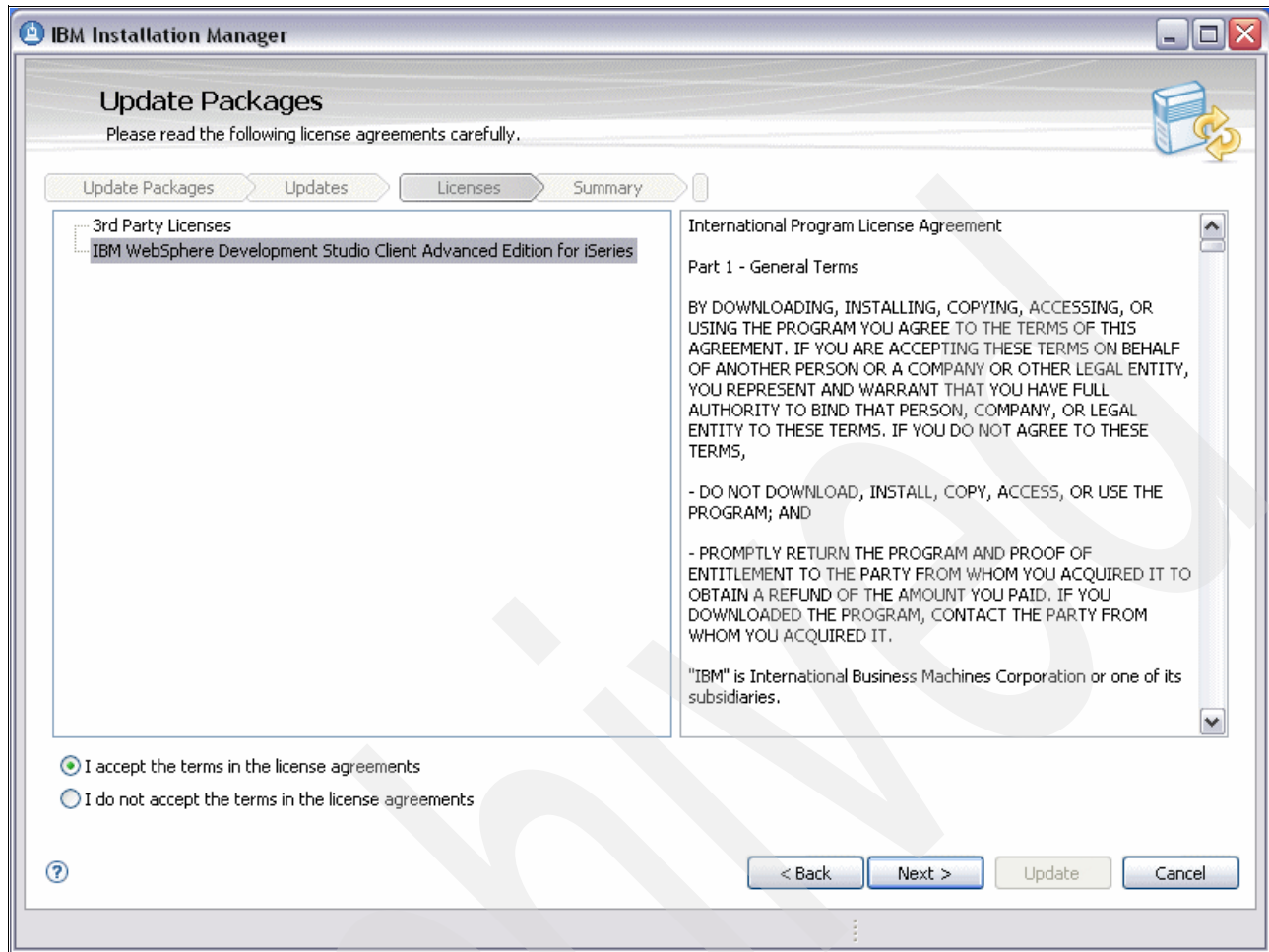


Figure 6-7 Update packages - license page

8. On the Summary page, review your choices before installing the updates; Figure 6-8.
 - a. If you want to change the choices you made on previous pages, click **Back**, and make your changes.
 - b. When you are satisfied, click **Update** to download and install the updates. A progress indicator shows the percentage of the installation completed.

Note: During the update process, Installation Manager might prompt you for the location of the repository for the base version of the package. If you installed the product from CDs or other media, they must be available when you use the update feature.

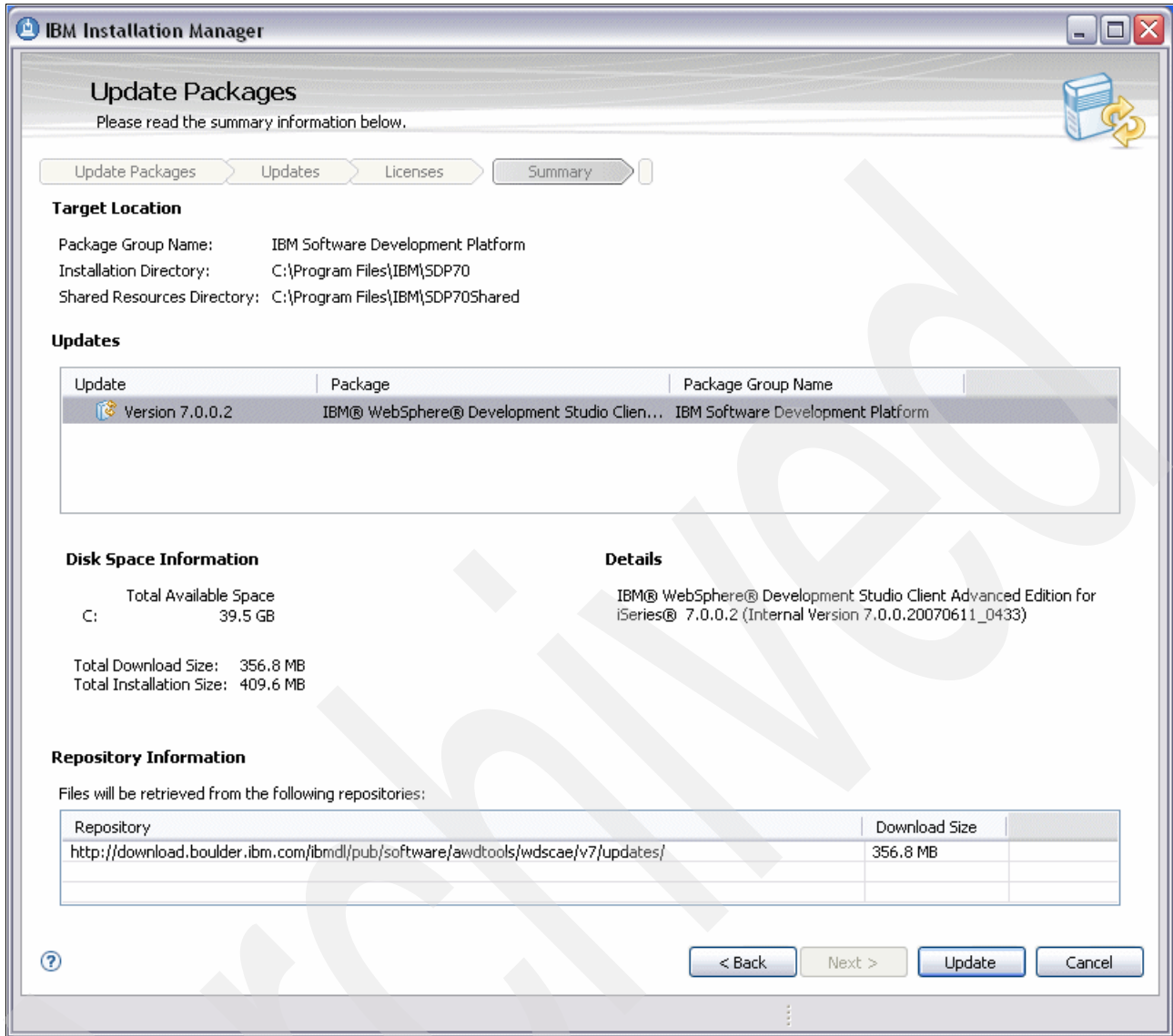


Figure 6-8 Update packages - summary page

9. Optional: When the update process completes, a message that confirms the success of the process is displayed near the top of the page. Click **View log file** to open the log file for the current session in a new window. You must close the Installation Log window to continue.

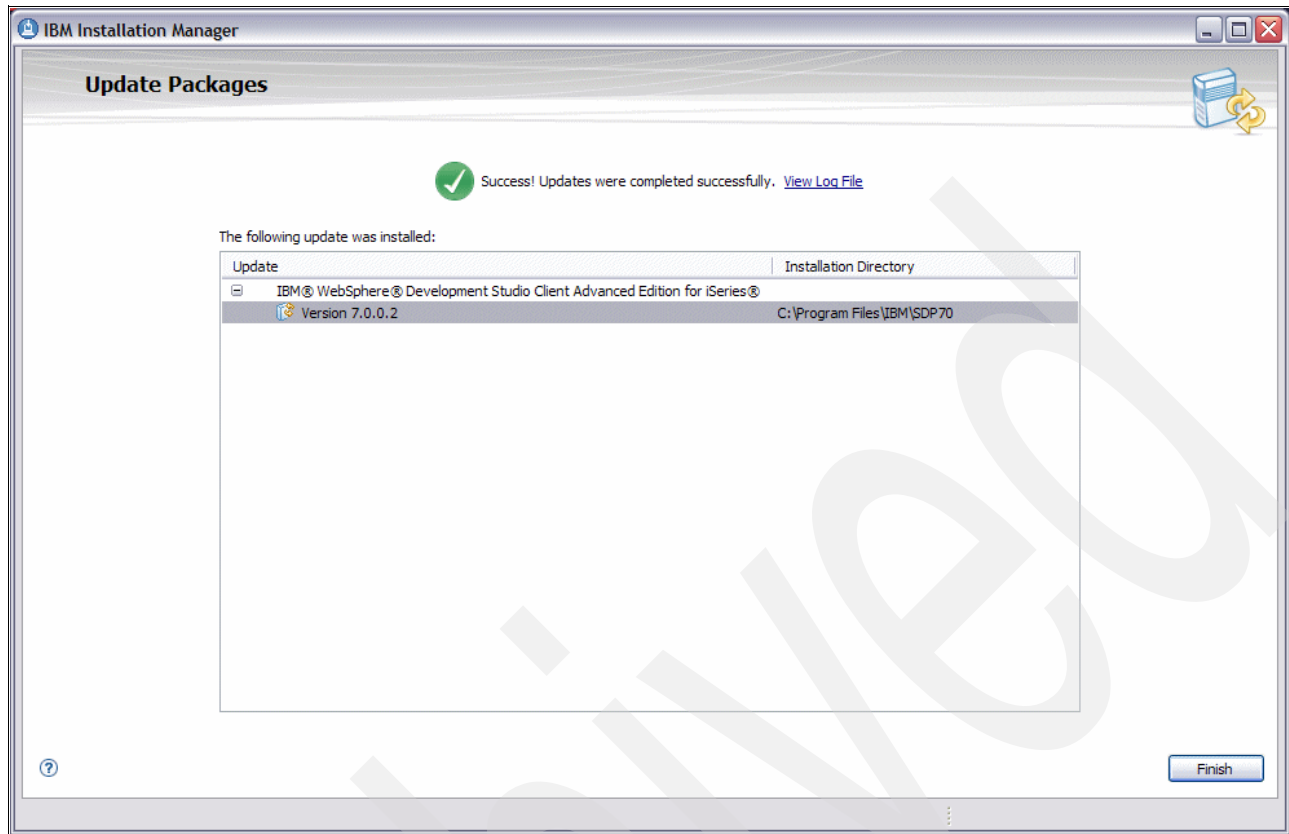


Figure 6-9 Update packages - successful update summary page

10. Click **Finish** to close the wizard.

6.2 Extending WDCS with optional software

This section explains how to extend WDCS with optional software.

6.2.1 List of optional software

The following optional software is included with WDCS for System i or WDCS AE for System i product:

- ▶ IBM Rational Agent Controller Version 7.0.1
- ▶ IBM CoOperative Development Environment and VisualAge(R) for RPG Version 6.0
- ▶ IBM Host Access Transformation Services Toolkit Version 7.0
- ▶ IBM WebSphere Application Server Version 6.1 for Windows
- ▶ **Advanced** ▶ IBM WebSphere Portal Version 6.0 test environment (Advanced Edition only)
- ▶ **Advanced** ▶ IBM WebSphere Portal Version 5.1.0.x test environment (Advanced Edition only)
- ▶ **Advanced** ▶ Crystal Reports Server XI Release 2 for Windows (Advanced Edition only)
- ▶ **Advanced** ▶ IBM Rational ClearCase LT Version 7.0 for Windows (Advanced Edition only)

For more information about each optional software, refer to the “Installing optional software” section in *WDCS Installation Guide*. The latest version of the Installation Guide is available at:

Installing Host Access Transformation Services Toolkit Version 7.0

7.0 Do the following steps to install HATS:

1. You can start HATS installation from the WDSC launchpad, or from the HATS launchpad in HATS CD:
 - a. From the WDSC launchpad:
 - i. Start the WDSC launchpad; for more information, refer to “Starting from the launchpad” on page 89.
 - ii. In the launchpad, under Install Optional Products, select **IBM WebSphere Host Access Transformation services v7.0**.
 - iii. You will be prompted to enter the HATS directory, click **Browse** and select the disk1 folder located in your HATS CD or local folder and click **OK**; Figure 6-10.

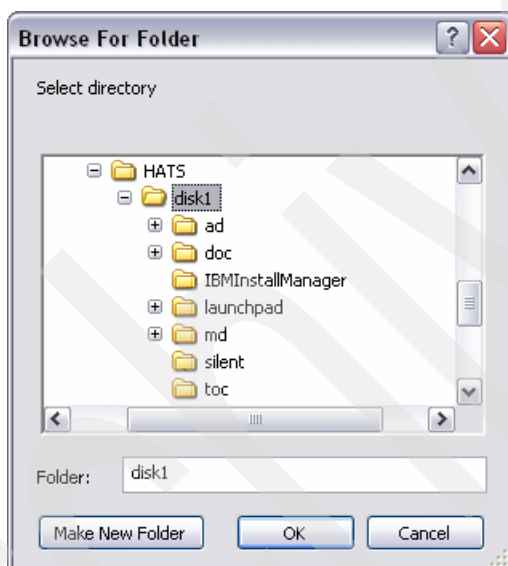


Figure 6-10 Select HATS disk1

- iv. Click **OK** in the Select directory dialog; Figure 6-11.

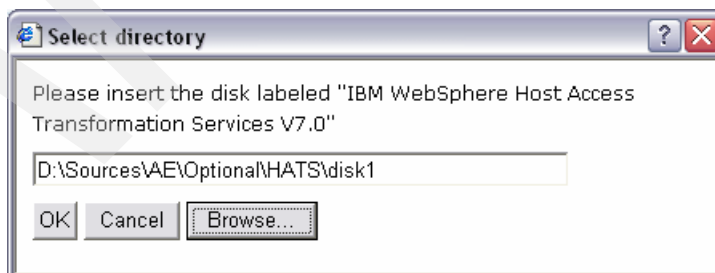


Figure 6-11 Select disk directory of HATS

- b. From the HATS launchpad:
 - i. Insert the HATS CD into the CD-ROM drive. If you do not have auto-run enabled, execute launchpad.exe from the root directory of the CD.

- i. The Welcome to IBM WebSphere HATS window appears, as shown in Figure 6-12. From this window you can view release information, prerequisite information, information resources, or install the product. Choose **Release Information** and review the Readme file, which contains last-minute information.

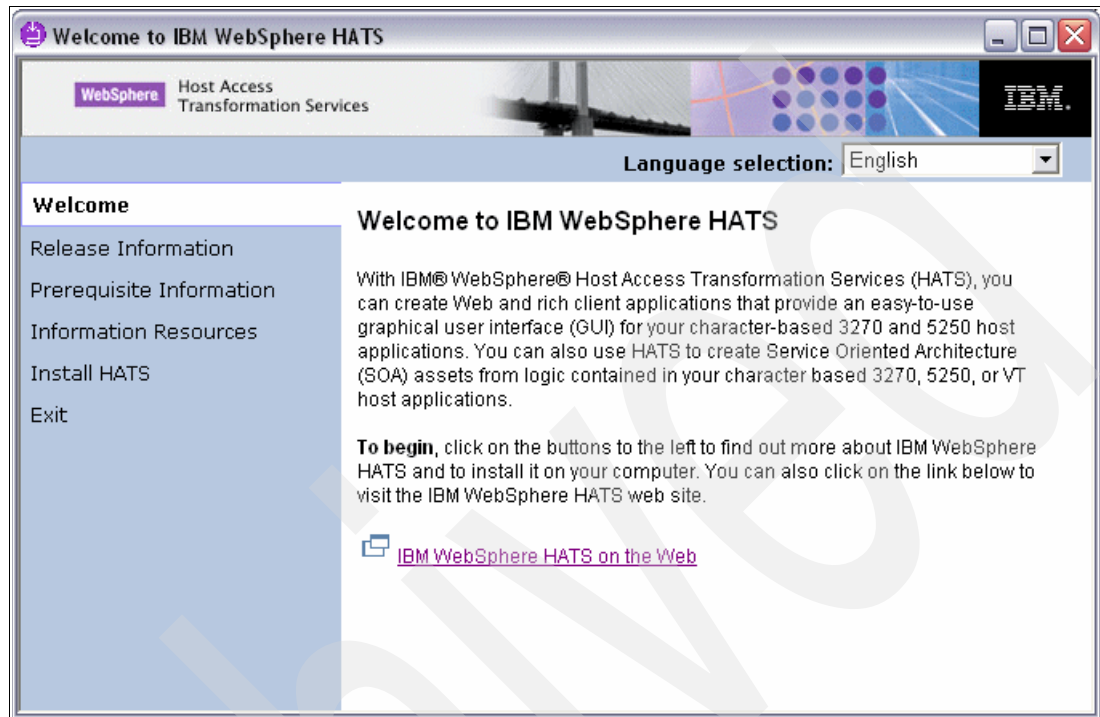


Figure 6-12 HATS launchpad

- i. When you are ready to install HATS Toolkit, select **Install HATS** and click **Launch IBM Installation Manager** to install HATS.

Note: If IBM Installation Manager is not installed, an error message is displayed.

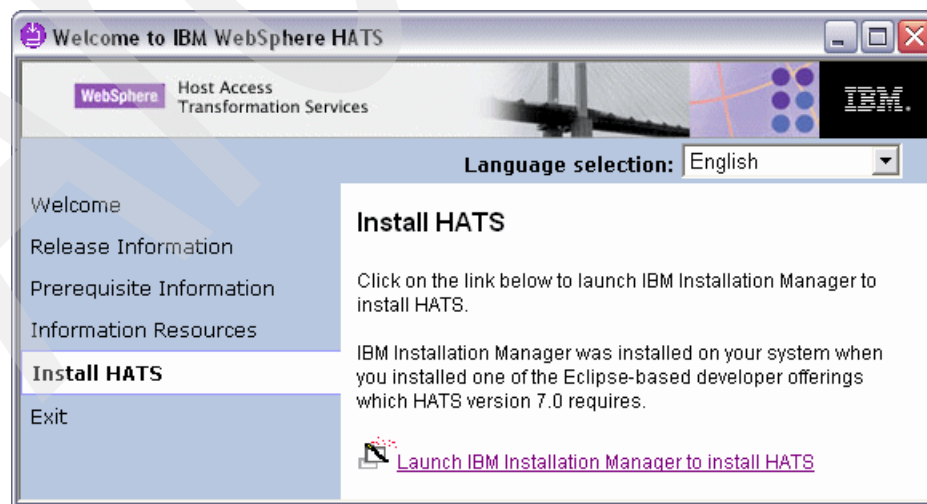


Figure 6-13 Launch IBM Installation Manager to install HATS

2. The IBM Installation Manager window appears.

3. On the Install Packages panel, select the box for **IBM WebSphere HATS** and click **Next**.
4. On the Licenses panel, accept the terms in the license agreement and click **Next**.
5. On the Location panel, under Package Group Name, select the package group where you want to install HATS, for example **IBM Software Development Platform**, and click **Next**.

Note: You must select an existing Rational SDP group so that correct prerequisite information can be determined.

6. On the Features panel, click **Next**.
7. On the Summary panel, click **Install**.
8. You will see a progress bar as the installation program installs HATS and creates Start menu items.
9. On the Install Packages panel, you are informed if the installation was completed successfully. If necessary, you can click **View Log File** to view the installation log. Click **Finish** to close the installation panel.

6.3 Modifying the installation

You can install new or additional features, or functions, and remove existing features using IBM Installation Manager.

When you install WDSC, you can install all of the features available to you, or choose to customize your installation by selecting to install only the features that you need. You can, however, choose to install the excluded optional features at a later date.

Using the Modify Packages function in IBM Installation Manager, you can install additional optional product features for the WDSC package installed on your system.

By default, Internet access is required unless the repository preferences point to a local modification site or CDs.

Important: You have to close all programs that were installed using Installation Manager before modifying your installation.

6.3.1 Adding and removing components

1. Start Installation Manager, click **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
2. From the Start page of the Installation Manager, click the **Modify Packages** icon; Figure 6-14.

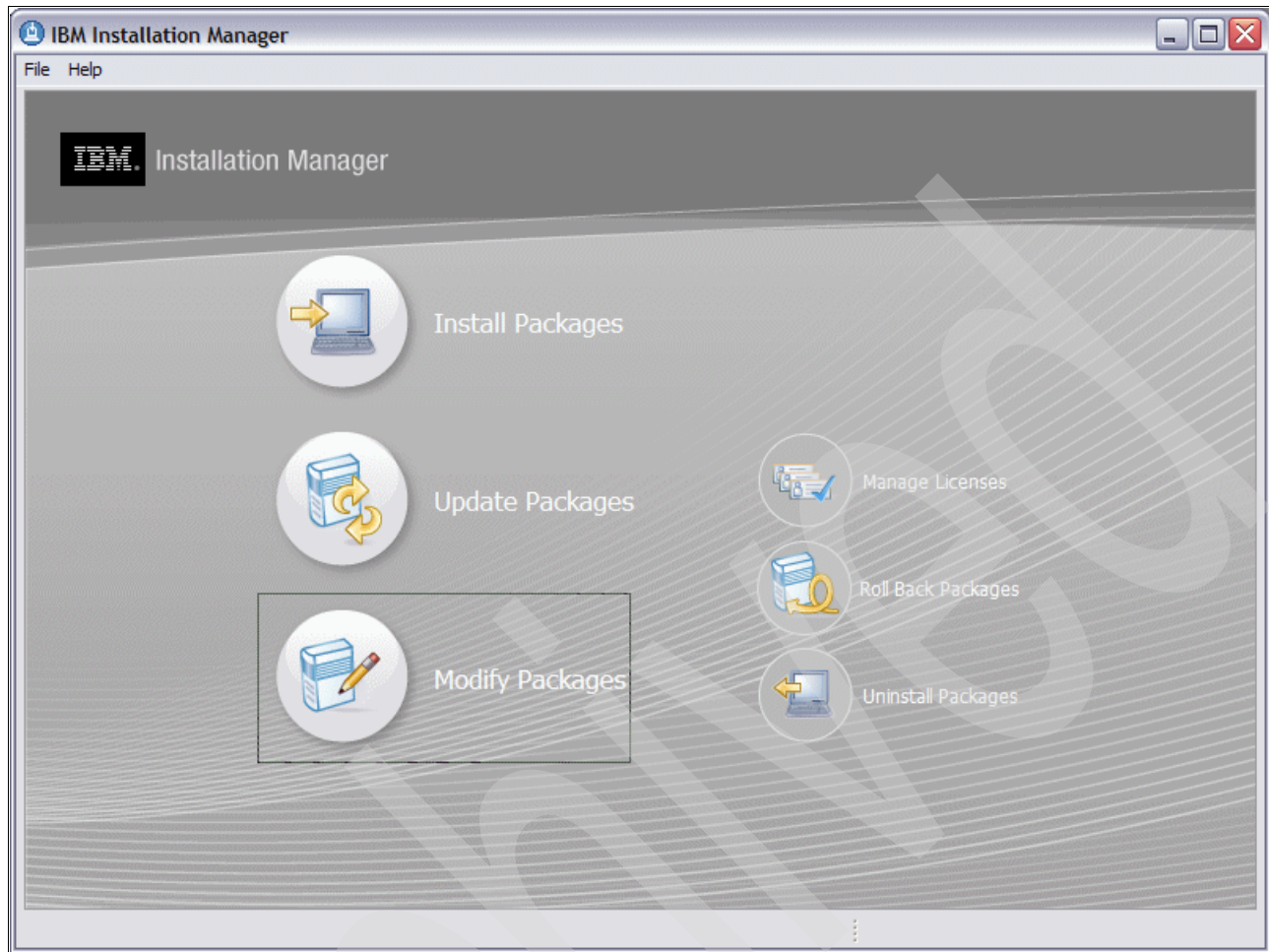


Figure 6-14 Installation Manager - Modify Product Packages

3. If a new version of Installation Manager is found, you are prompted to confirm that you want to install it before you can continue. For details on the current version and the updated version, click **Details**; Figure 6-4. Click **Yes** to proceed. Installation Manager automatically installs the new version, stops, restarts, and resumes.
4. In the Modify Packages wizard, select the installation location for the WDSC product package that you want to modify, and click **Next**; Figure 6-15.

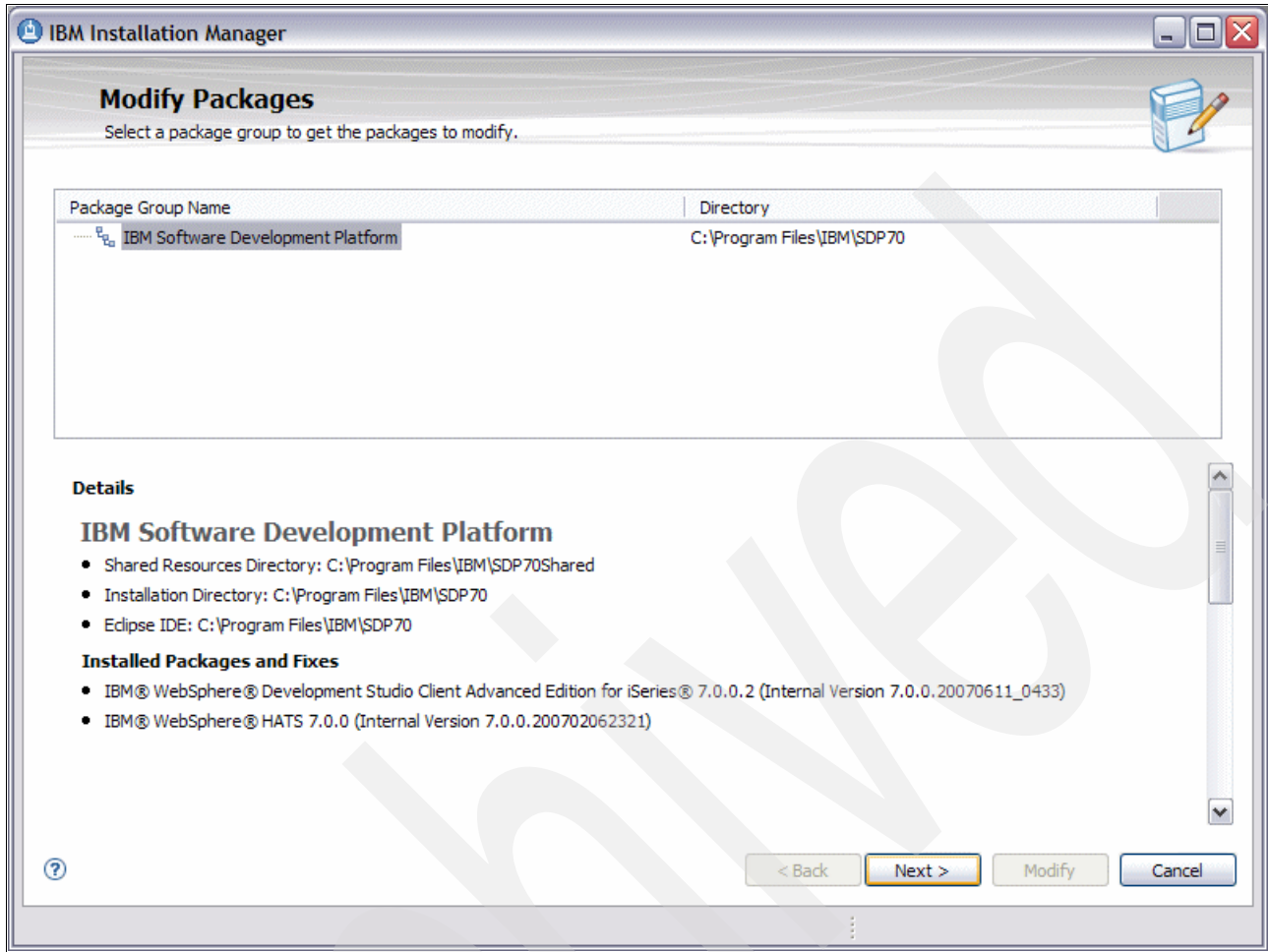


Figure 6-15 Installation Manager - Select package group

- On the next Modify Packages page, under Languages, select the languages for the package group; Figure 6-16. The corresponding national language translations for the user interface and documentation for the packages will be installed.

Note: The languages you choose apply to all packages installed under the package group that you selected in step 4.

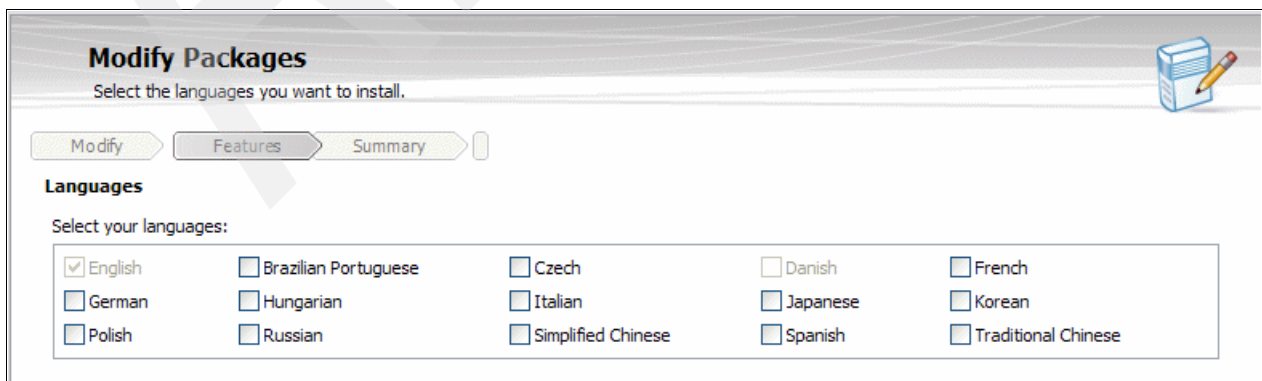


Figure 6-16 Installation Manager - Select languages

6. Click **Next**.
7. On the Features page, select the package features that you want to install or remove; Figure 6-17.
 - a. To learn more about a feature, click the feature and review the brief description under Details.
 - b. If you want to see the dependency relationships between features, select **Show Dependencies**. When you click a feature, any features that depend on it and any features that are its dependents are shown in the Dependencies window. As you select or exclude features in the packages, Installation Manager will automatically enforce any dependencies with other features and display updated download size and disk space requirements for the installation.
 - c. To return to the default features selected for the packages, click **Restore default**.

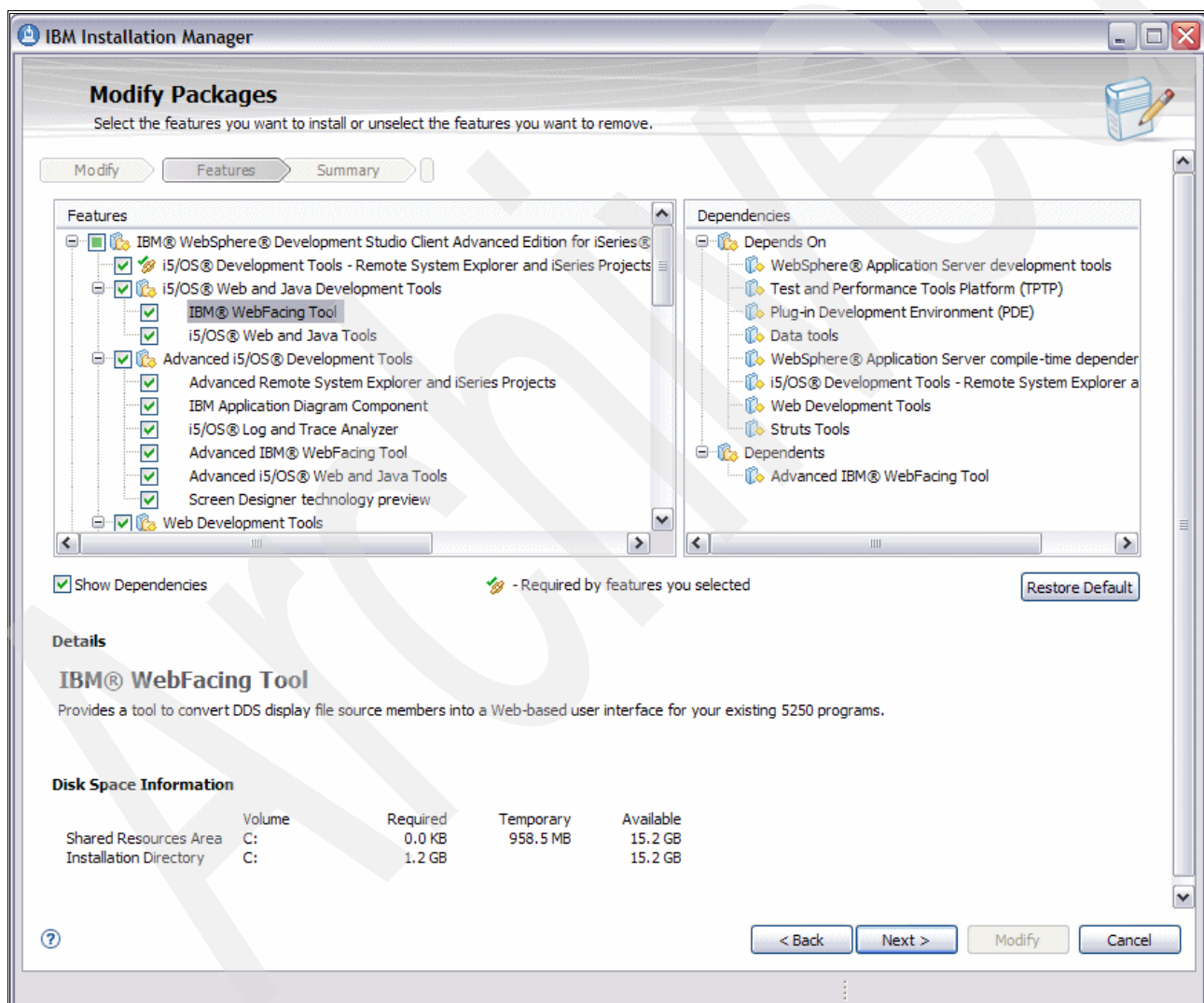


Figure 6-17 Install new features or remove existing features

8. When you are finished selecting features, click **Next** to continue.
9. On the Summary page, review your choices before installing the packages; Figure 6-18.
 - a. If you want to change the choices you made on previous pages, click **Back** and make your changes.

- b. When you are satisfied with your choices, click **Modify**. A progress indicator shows the percentage of the installation completed.

Note: The Adding Feature list displays the features that will be installed, and the Removing Feature list displays the features that will be removed.

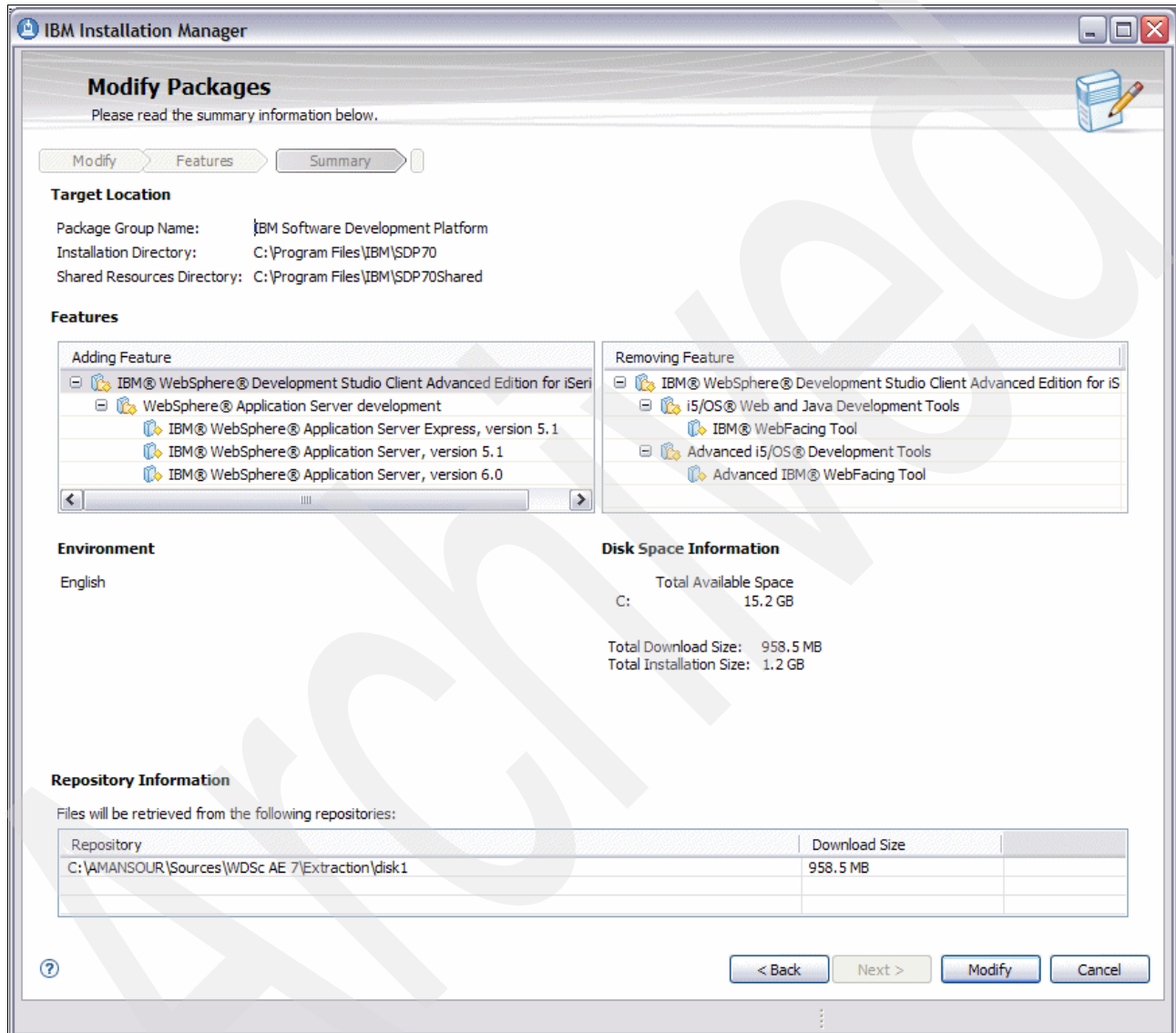


Figure 6-18 Modification summary information

10. Optional: When the modification process completes, click **View Log File** to see the complete log.

6.4 Placing WDCS on an HTTP server

You can install WDCS across a network by placing product package files for WDCS on an HTTP Web server. You must use the IBM Packaging Utility. The directory on the HTTP Web server that contains the package is called a repository.

6.4.1 IBM Packaging Utility

This utility is provided with WDSC to copy the installation files in a format that can be used for installing WDSC directly from an HTTP Web server.

The Packaging Utility software is located on the Auxiliary CD that is included with WDSC or WDSC AE. If you want to place a repository that contains a WDSC package on a Web server that will be available over HTTP or HTTPS, you must use the Packaging Utility to copy the product package of WDSC into the repository.

You can use this utility to perform the following tasks:

- ▶ Generate a new repository for product packages.
- ▶ Copy product packages to a new repository. You can copy multiple product packages into a single repository, thereby creating a common location for your organization from which product packages can be installed using IBM Installation Manager.
- ▶ Delete product packages from a repository.

Refer to the online help for the Packaging Utility for full instructions on using the tool.

Installing the Packaging Utility

The IBM Packaging Utility must be installed from the Auxiliary CD before it can be used to copy WDSC product packages.

Use the following steps to install the IBM Packaging Utility software:

1. Start the installation from the Auxiliary CD. Change to the Aux_CD_Win\PackagingUtility directory on the Auxiliary CD. PackagingUtility\PU_win32.zip and run install_win32.exe.
2. If IBM Installation Manager is not detected on your workstation, you are prompted to install it and then the installation wizard starts. Follow the on-screen instructions in the wizard to complete the installation of Installation Manager. See “Installation of Installation Manager” on page 76.
3. When the installation of Installation Manager completes, or if it is already on your computer, Installation Manager starts and automatically begins the Install Packages wizard.
4. Follow the on-screen instructions in the Install Packages wizard to complete the installation.

6.4.2 Installation using IBM Packaging Utility

To prepare WDSC or WDSC AE for installation from a repository located on an HTTP Web server, do the following steps.

Copying product packages to an HTTP server using the Packaging Utility

To copy product packages with the Packaging Utility:

1. If you are copying from a CD image, perform these tasks:
 - a. Insert the first installation CD into your CD drive.
 - b. If autorun is enabled on your system, the WDSC Launchpad program automatically opens. Close the Launchpad program.
2. Start the Packaging Utility: **Start** → **Programs** → **IBM Packaging Utility** → **IBM Packaging Utility**.

3. On the main page of the utility, click **Copy Packages**; Figure 6-19. The Prerequisite page opens, and presents two options:
 - a. I will be downloading product packages from IBM Web.
 - b. I will be obtaining the product packages from other sources.

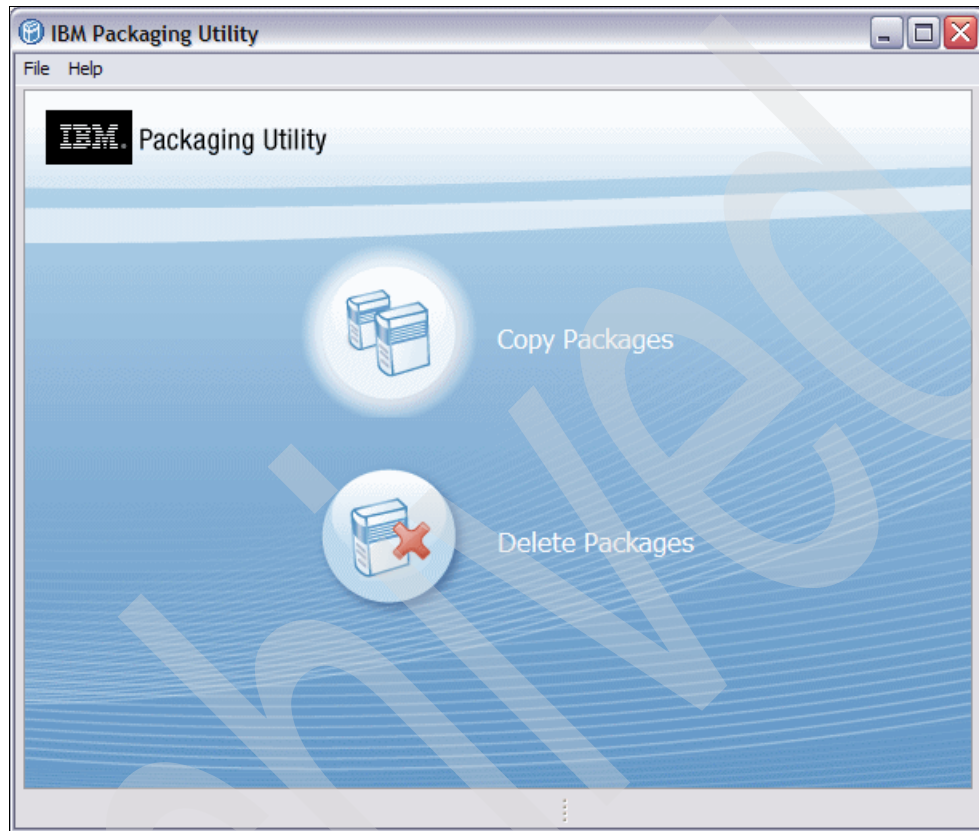


Figure 6-19 IBM Packaging Utility main page

4. Click **I will be downloading product packages from IBM Web**; Figure 6-20.

Note: You can use the “I will be obtaining the product packages from other sources” option if you have already defined an accessible repository.

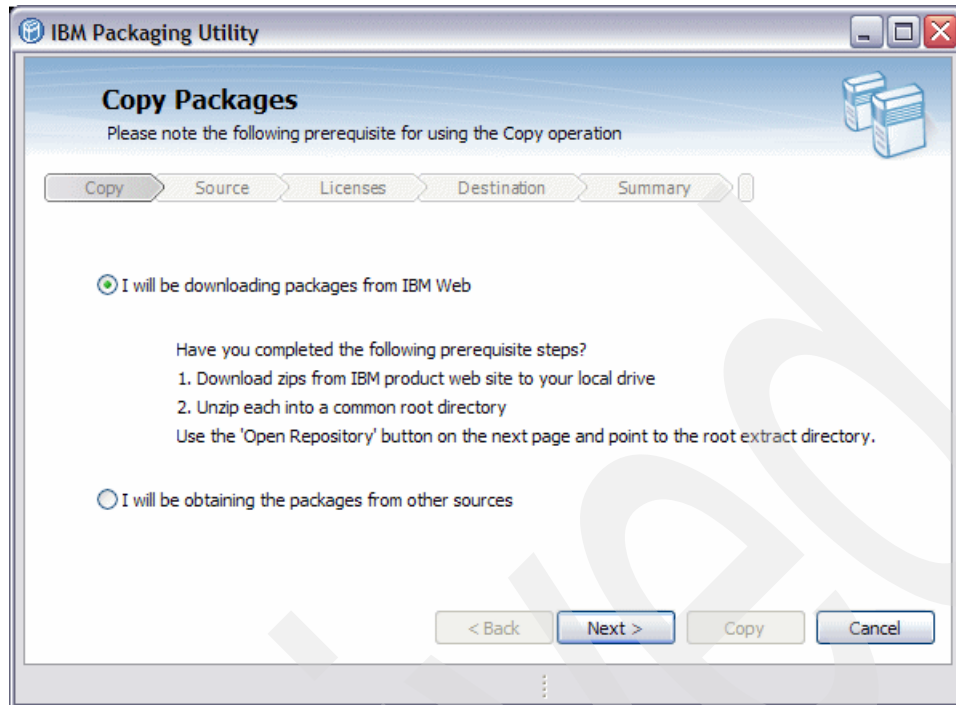


Figure 6-20 IBM Packaging Utility copy packages

5. Click **Next** to advance to the Source page. If there are no product packages to select, you must open a repository that contains product packages.
6. To open a repository, click **Open repository**. The Open Repository window opens. A repository can be a path to a directory in the file system, a disk drive containing the first CD of the product, or a URL to a directory on a server.
7. To define a repository location, click **Repository Location Browse**, and then navigate to and select the Repository location—either the common root directory that contains the electronic disk images or the drive containing the first product installation CD. For example, if the WDSC files (disk1, disk2, and so on) reside in C:\My product\unzip, you should define this location as a repository.
8. Click **OK** to define the repository location and to close the Browse to a repository directory window; Figure 6-21.

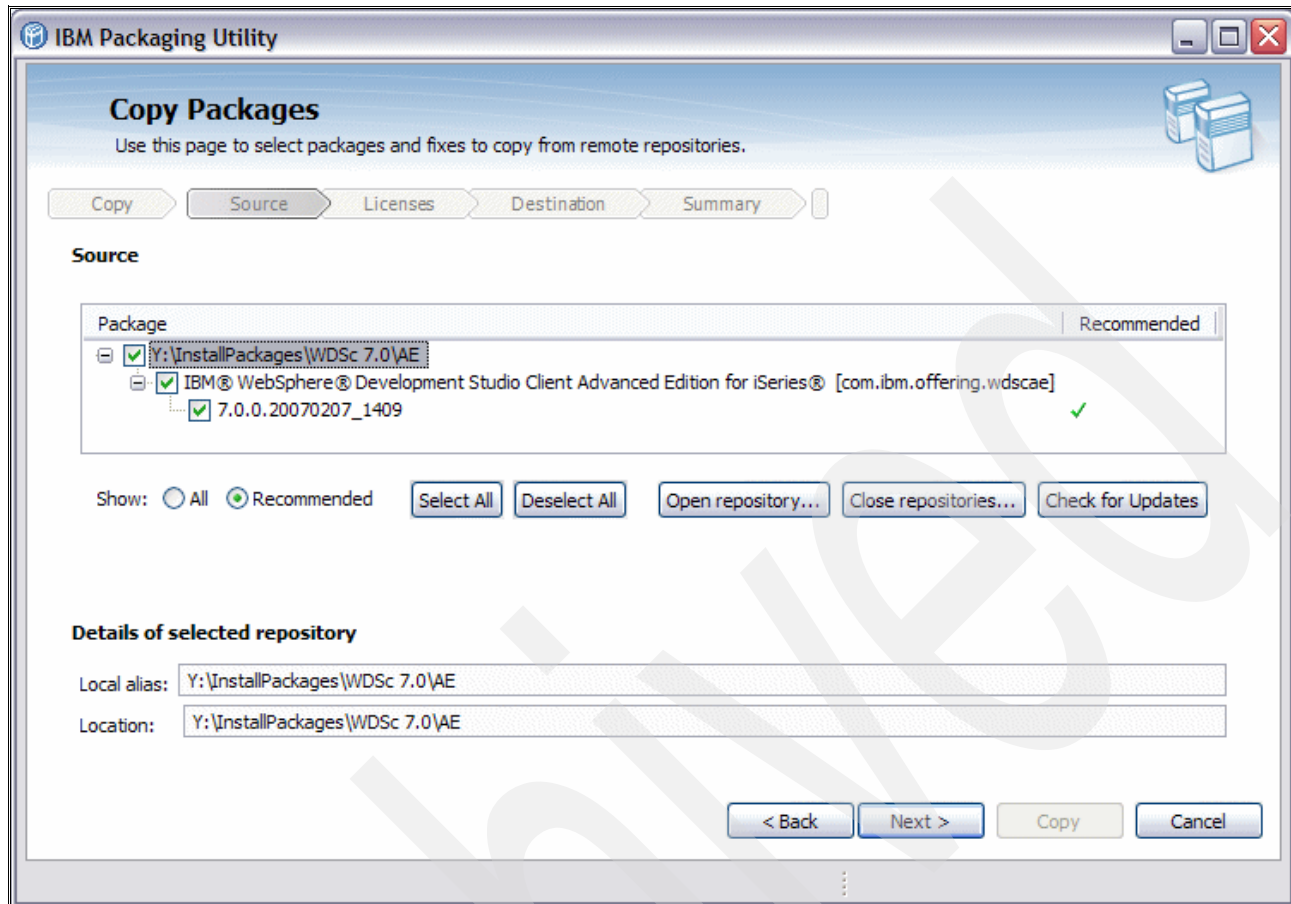


Figure 6-21 IBM Packaging Utility source packages

9. On the Destination page, click **Browse** and select an existing repository directory, or create a new folder to store the products.
10. After you specify a repository for the selected product packages and any fixes, click **OK** to close the Browse to a directory window. The file path that you just defined is listed in the Directory field on the Destination page.
11. Click **Next** to advance to the Summary page. The Summary page displays the selected product packages that will be copied into the destination repository. This page also lists the amount of storage space that the copy requires, as well as the amount of available space on the drive; Figure 6-22.

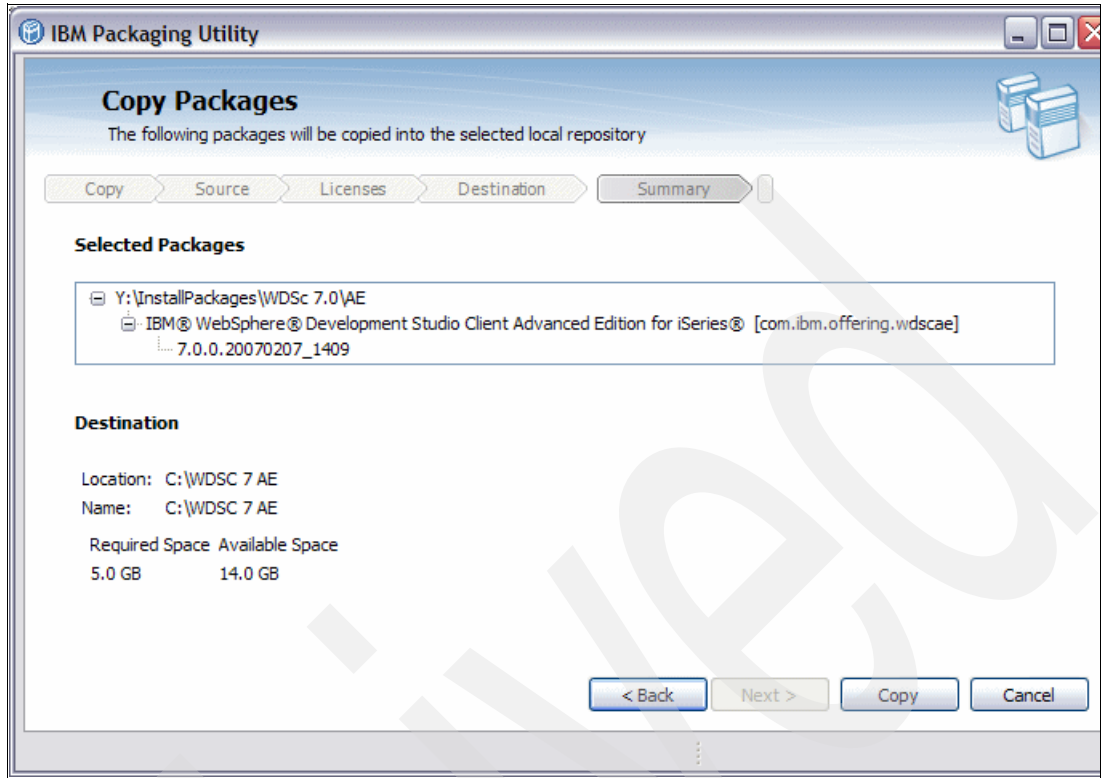


Figure 6-22 IBM Packaging Utility Summary page

12. Click **Copy** to copy the selected product packages to the destination repository. A status bar opens at the bottom of the wizard indicating how much time is remaining in the copy process. After the copy process is finished, a Complete page opens and displays all of the product packages that were copied successfully.
13. Click **Done** to return to the Packaging Utility main page.

Now that you have used the Packaging Utility to copy the WDSC installation files into a repository, you can place the repository on a Web server and make the directories and files available over HTTP. (The repository can also be placed on a UNC drive.)

6.5 Uninstalling WDSC

The Uninstall Packages option in the Installation Manager enables you to uninstall packages from a single installation location. You can also uninstall all the installed packages from every installation location.

Note: To uninstall the packages, you must log in to the system using the same user account that you used to install the product packages.

To uninstall the packages:

1. Close the programs that you installed using Installation Manager.
2. Start Installation Manager: **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
3. On the Start page of the Installation Manager, click the **Uninstall Packages** icon.

4. In the Uninstall Packages page, select the WDCS package that you want to uninstall.

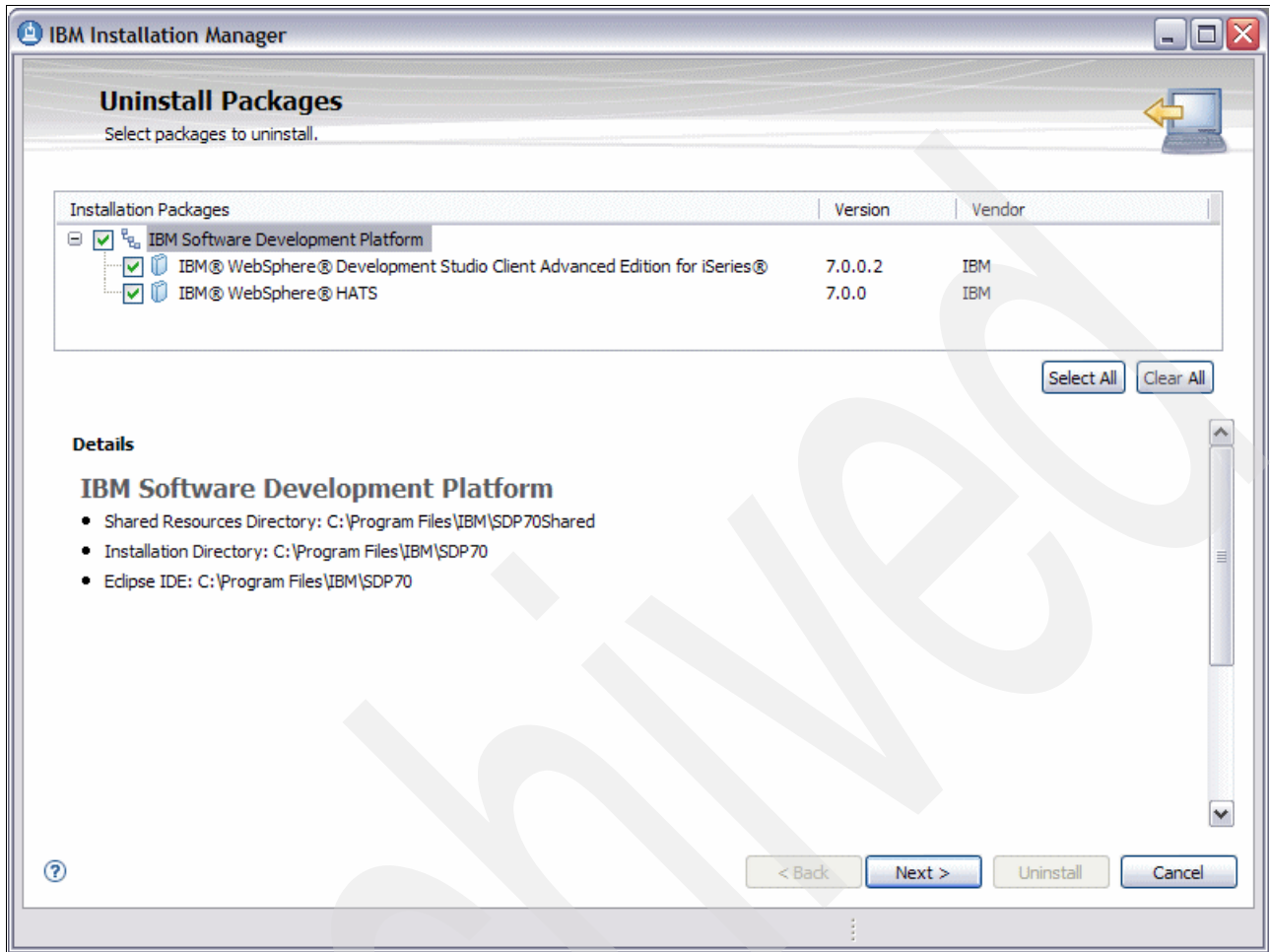


Figure 6-23 Uninstall WDCS packages

Important: You must remove the installed optional software package before removing the WDCS base package, or you can remove all the packages at the same time. The Next button in the Installation manager will be disabled, if you tried to remove the WDCS package while another depending package is still available; Figure 6-24. For information about optional software, refer to 6.2.1, “List of optional software” on page 115.

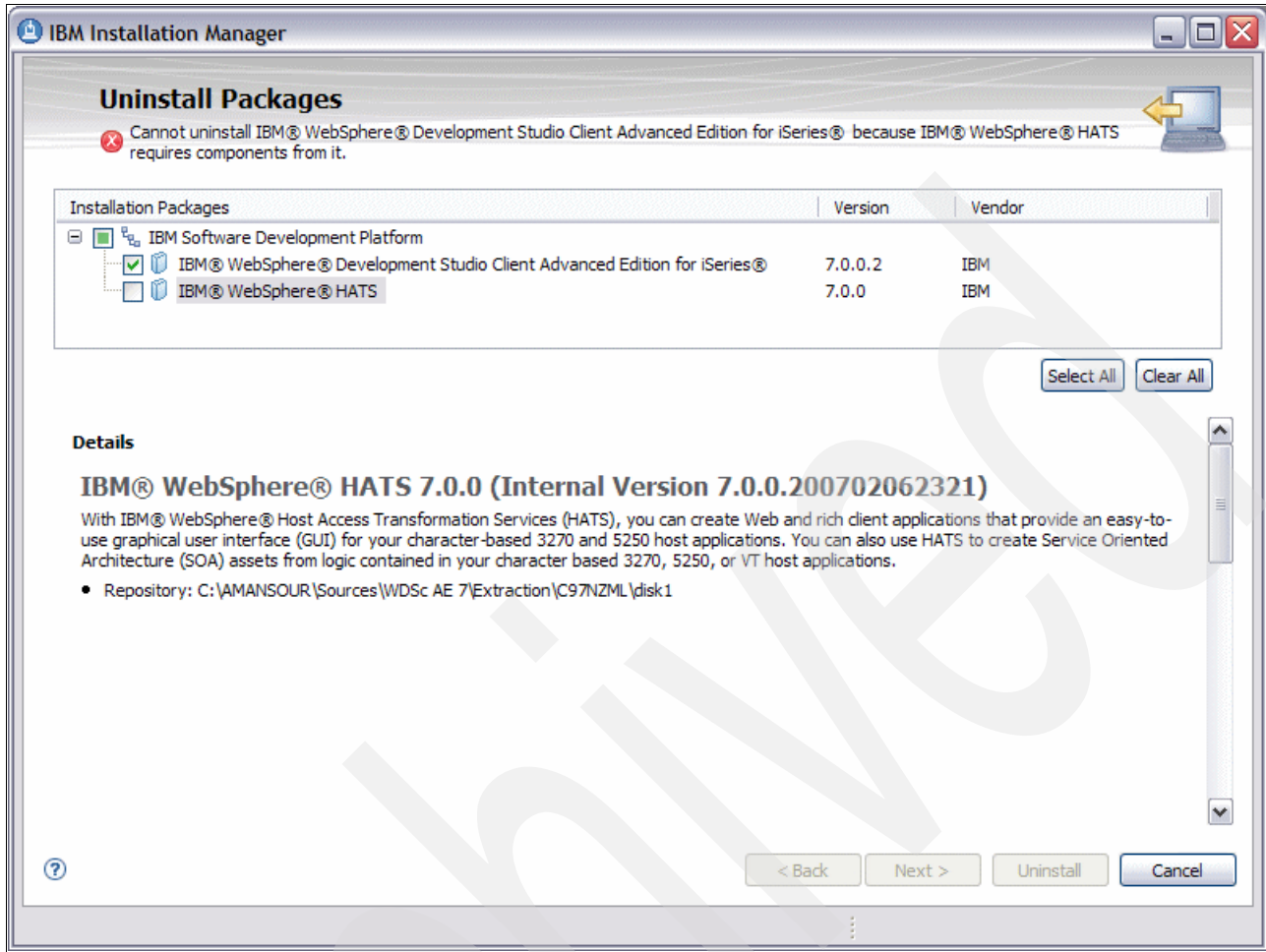


Figure 6-24 Uninstalling WDSC dependency warning

5. Click **Next**.
6. In the Summary page, review the list of packages that will be uninstalled and then click **Uninstall**; Figure 6-25. A progress indicator shows the percentage of the uninstallation completed.

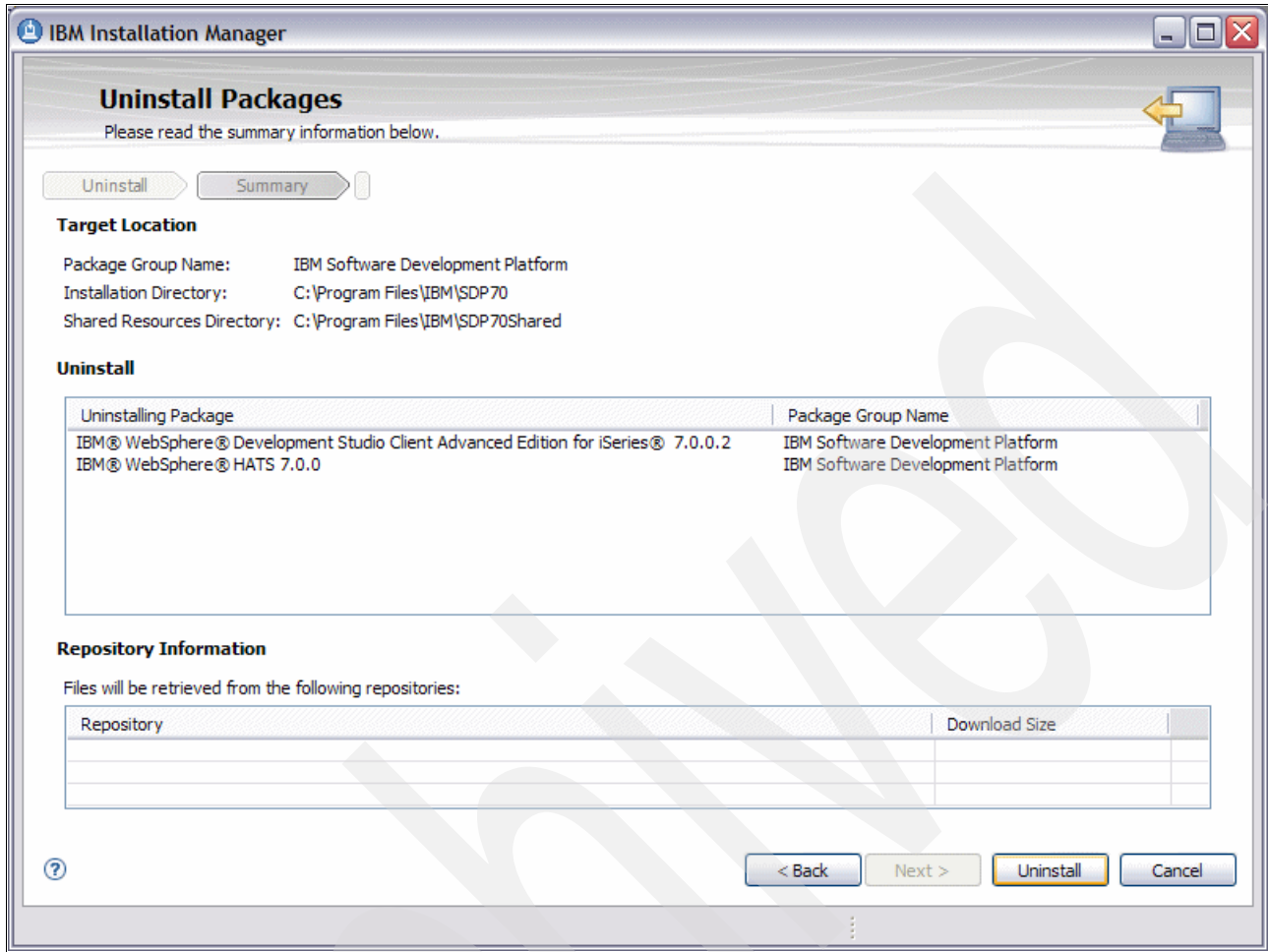


Figure 6-25 Uninstall summary page

7. When the uninstallation process is completed, the Complete page opens and confirms success of the uninstallation process.
8. Click **Finish** to exit the wizard.

WDSC troubleshooting

To find the version of the Installation Manager:

1. Start Installation Manager: **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
2. Select **Help** → **About IBM Installation Manager**; Figure 6-26 and Figure 6-27.

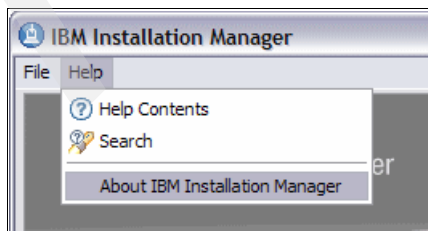


Figure 6-26 About IBM Installation Manager

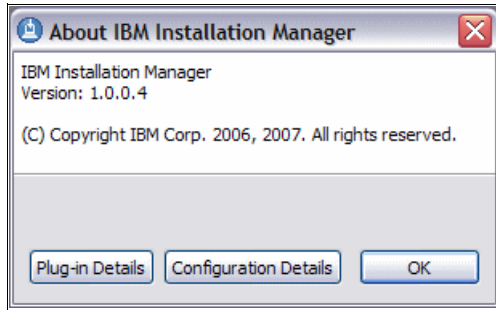


Figure 6-27 IBM Installation Manager Version

Version of plug-ins

To find the version of installed plug-ins:

1. Start WDS.
2. Select **Help** → **Software Updates** → **Manage Configuration**; Figure 6-28. This will open the Product Configuration window.

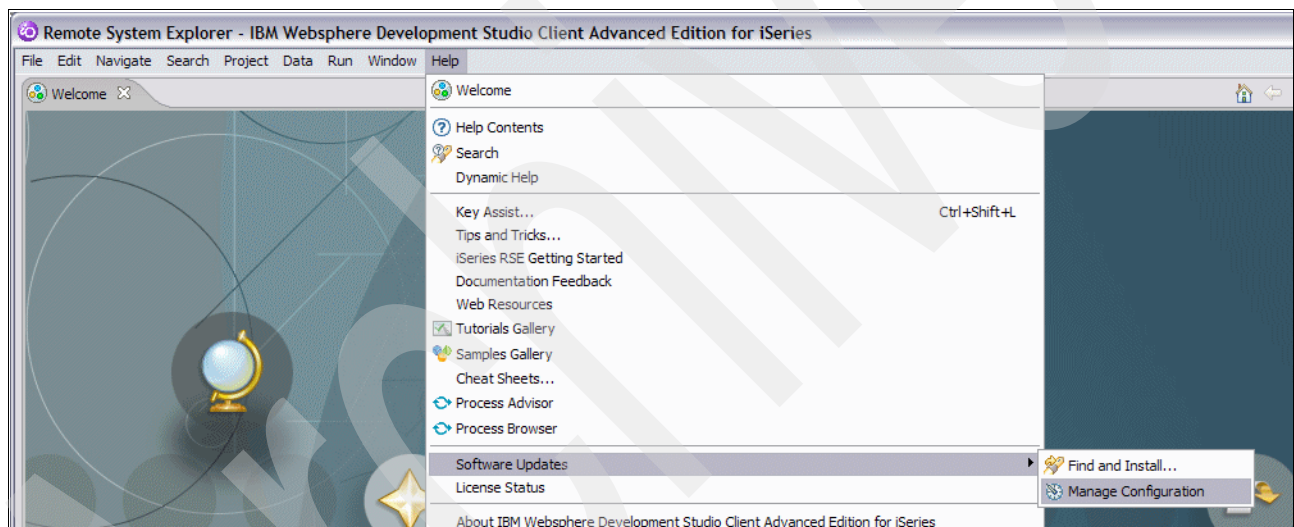


Figure 6-28 Manage WDS configuration

3. From the Product Configuration menu, expand the WDS shared installation directory and select a plug-in. This will show the available tasks for this plug-in; Figure 6-29.

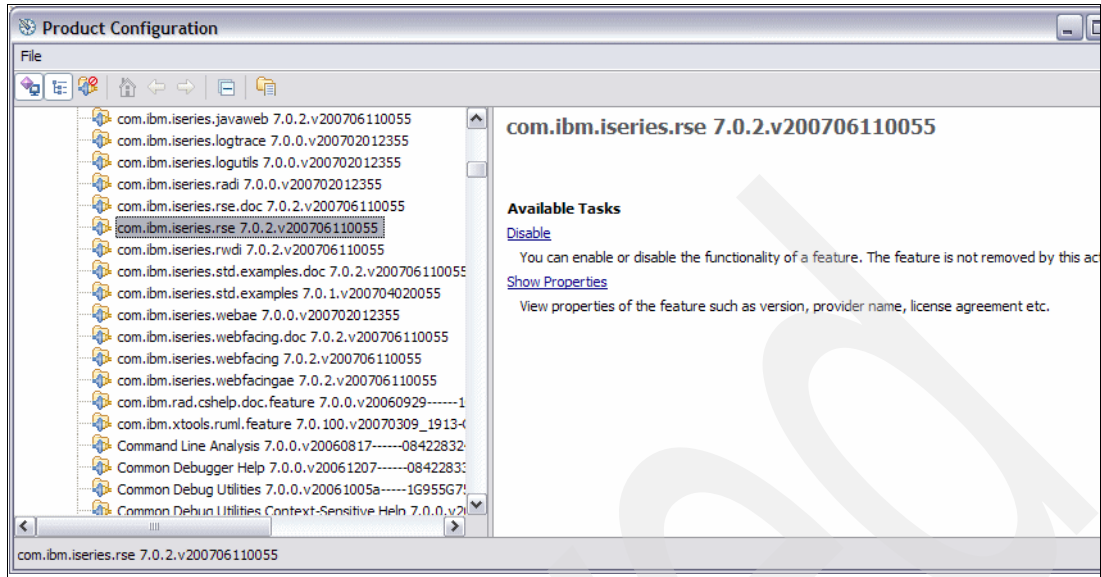


Figure 6-29 Product configuration - plug-in tasks

- To show the feature version and properties, click **Show Properties**; Figure 6-30.

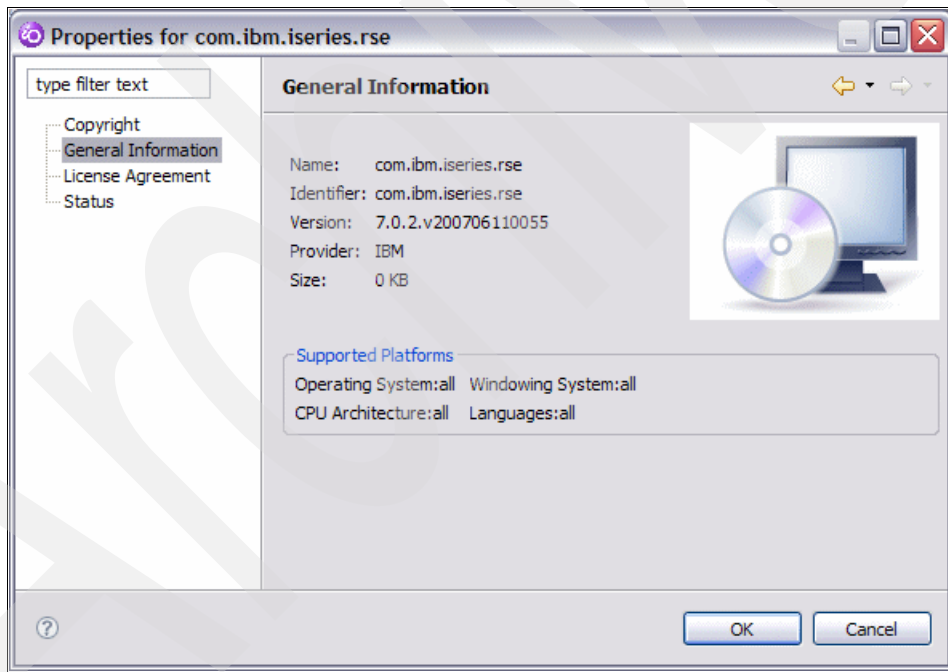


Figure 6-30 Product configuration - Feature properties

Installation log files

- Start Installation Manager: **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
- Click **File** → **View Log**; Figure 6-31.

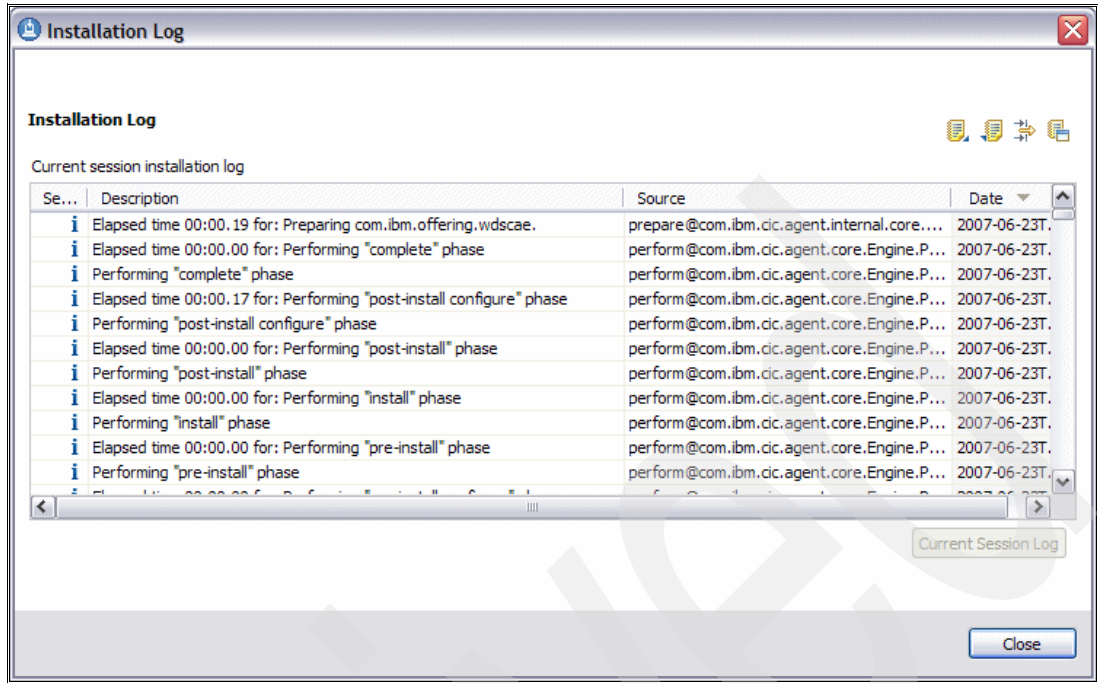


Figure 6-31 Installation Log

3. Click **File** → **Installation History**; Figure 6-32.

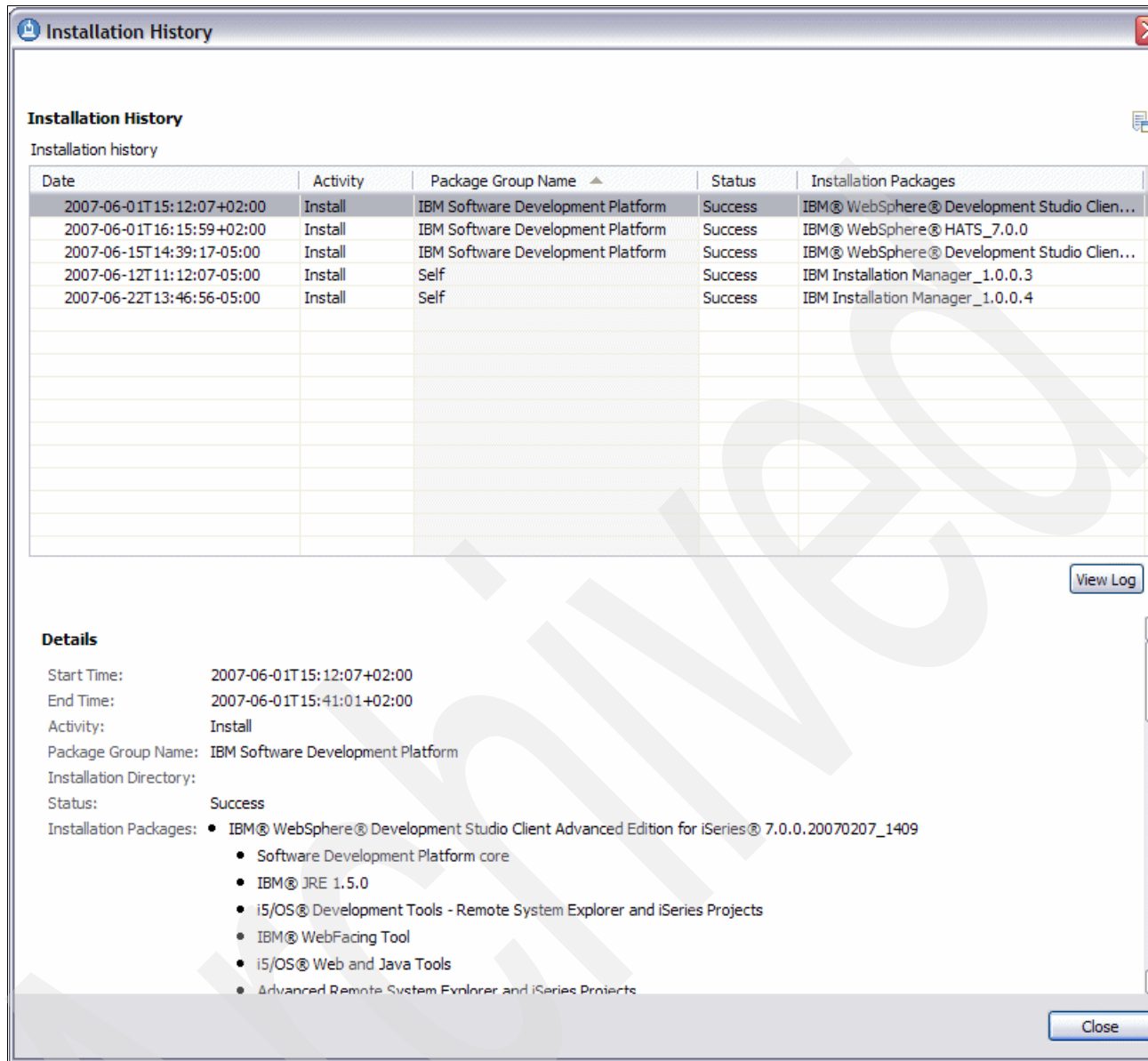


Figure 6-32 Installation history

Workspace log file

This file logs what errors occur when the IDE is started up, including problems with loading the plug-ins. You can find the workspace log file in the .metadata folder inside the WDSC workspace folder:

C:\Documents and Settings\Administrator\IBM\rationalSDP7.0\workspace\.metadata\.log

Uninstalling Installation Manager

You can also uninstall Installation Manager by the control panel:

1. Go to the **Control Panel** → **Add or Remove Programs**.
2. Select the entry for IBM Installation Manager and click **Remove**; Figure 6-33.

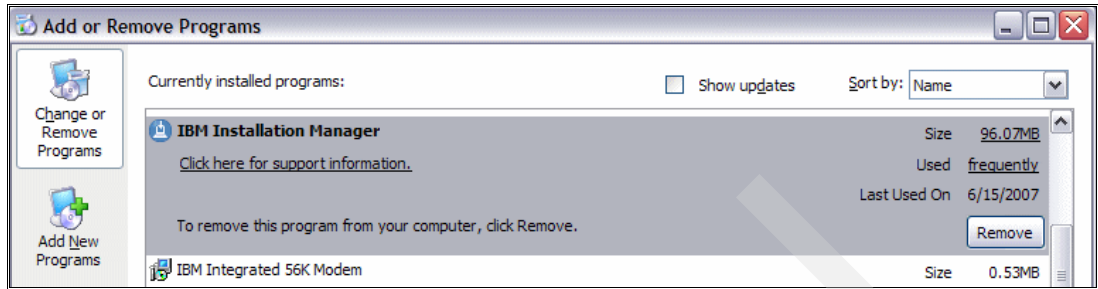


Figure 6-33 Removing IBM Installation Manager

Manual cleanup

To manually clean up a WDCS installation, do the following:

1. Make sure that in your folder options you can see hidden files and folders.
2. Go to C:\Program Files\IBM and remove InstallationManager and SDP70 folders.
3. Go to C:\WINDOWS and remove the .nifregistry file, if found.
4. Go to C:\WINDOWS and remove the vpd.properties file, if found.
5. Go to C:\Documents and Settings\- 6. The C:\Documents and Settings\- 7. Go to C:\Documents and Settings\- 8. Go to C:\Documents and Settings\All Users\Application Data\IBM and remove the InstallationManager folder.

Archived

The Remote System Explorer and System i Projects

This part concentrates on the discussion of RSE and System i Projects with the following chapters:

- ▶ Working with i5/OS-based development resources using the Remote System Explorer
- ▶ Remote Systems LPEX Editor: The modern SEU and much more
- ▶ Screen Designer Technical Preview - the modern Screen Design Aid (SDA) in WDSC AE 7.0
- ▶ Application Diagram in WDSC AE - visualize program structures
- ▶ System i Projects
- ▶ Integrated System i Debugger

Archived

Working with i5/OS-based development resources using the Remote System Explorer

This chapter introduces you to the Remote System Explorer (RSE), the modern PDM. It discusses and demonstrates how to use this Workbench perspective to manage your System i development resources in a productive manner.

Several sections in this chapter have been taken from the previous, WDSC V5.1.2-based version of this book. Changes and additions are marked by a vertical bar on the left of the page.

This chapter covers the following topics:

- ▶ Introduction to the Remote System Explorer
 - A Guided tour of RSE
 - What's new in WDSC
- ▶ Connecting to your System i host
 - Profiles
 - Connections
- ▶ Subsystems
 - System i objects
 - System i commands
 - System i jobs
 - IFS files
 - Qshells
- ▶ Managing your System i objects, including filters and filter pools
- ▶ Working with your library list
- ▶ System i table views
- ▶ User actions, including named types
- ▶ Drag and drop functionality

References to more information

There is also a tutorial, which you can find under **Help** → **Tutorials Gallery** and then expand.

Do and Learn

→ **System i**


→ **Develop System i applications with the Remote System Explorer**

You may either follow the online tutorial or download and print a PDF attached to the first page. If you do the latter, make sure to read and perform at least the steps described under **Introduction** → **Prerequisites** → **Restore the RSELABXX savefile (RSELABXX.savf) on an System i system** → **Import the RSELABXX project to your workspace**, since it provides the sample application which you can install onto your own System i.

7.1 Introduction to Remote System Explorer

The Remote System Explorer is more than just a graphical user interface (GUI). It is a Workbench perspective that provides access to all development resources of your System i server. You can think of it as an enhanced and more flexible workstation version of the Programming Development Manager (PDM). It allows you to effectively manage and organize all the System i resources and applications needed by application developers in an easy and user-friendly way.

When you first open the workbench, the RSE is open by default. If this is not the case, follow these steps to open the RSE perspective:

1. From the Workbench menu, select **Window** → **Open Perspective**. You can also click the **Open Perspective**() button on the toolbar on the right side of the Workbench window.
2. Select **Remote System Explorer**. If this option does not appear on this submenu, try these steps:
 - a. Select **Other**. The Select Perspective window opens.
 - b. Select **Remote System Explorer (default)**.
 - c. Click **OK**.

By default, the RSE perspective consists of the following editors and views as shown in Figure 7-1 on page 142:

- ▶ A tabbed notebook at the upper left containing:
 - Remote Systems view
 - Team view
- ▶ A tabbed notebook at the lower left containing:
 - Properties sheet
 - Remote Scratchpad
- ▶ An area for editors in the middle, showing the System i RSE Getting Started page in Figure 7-1 (new in WDSC V7.0 and WDSC AE V7.0).
- ▶ Outline
- ▶ A tabbed notebook containing:
 - Remote System Details
 - Tasks
 - System i Table View (visible in Figure 7-1)
Command section below the Series Table View (see mouse pointer in Figure 7-1)
 - System i Commands Log

Note: For more information about perspectives and views and how to work with them, see 2.2.3, “Perspectives” on page 21, and 2.2.4, “Views” on page 23.

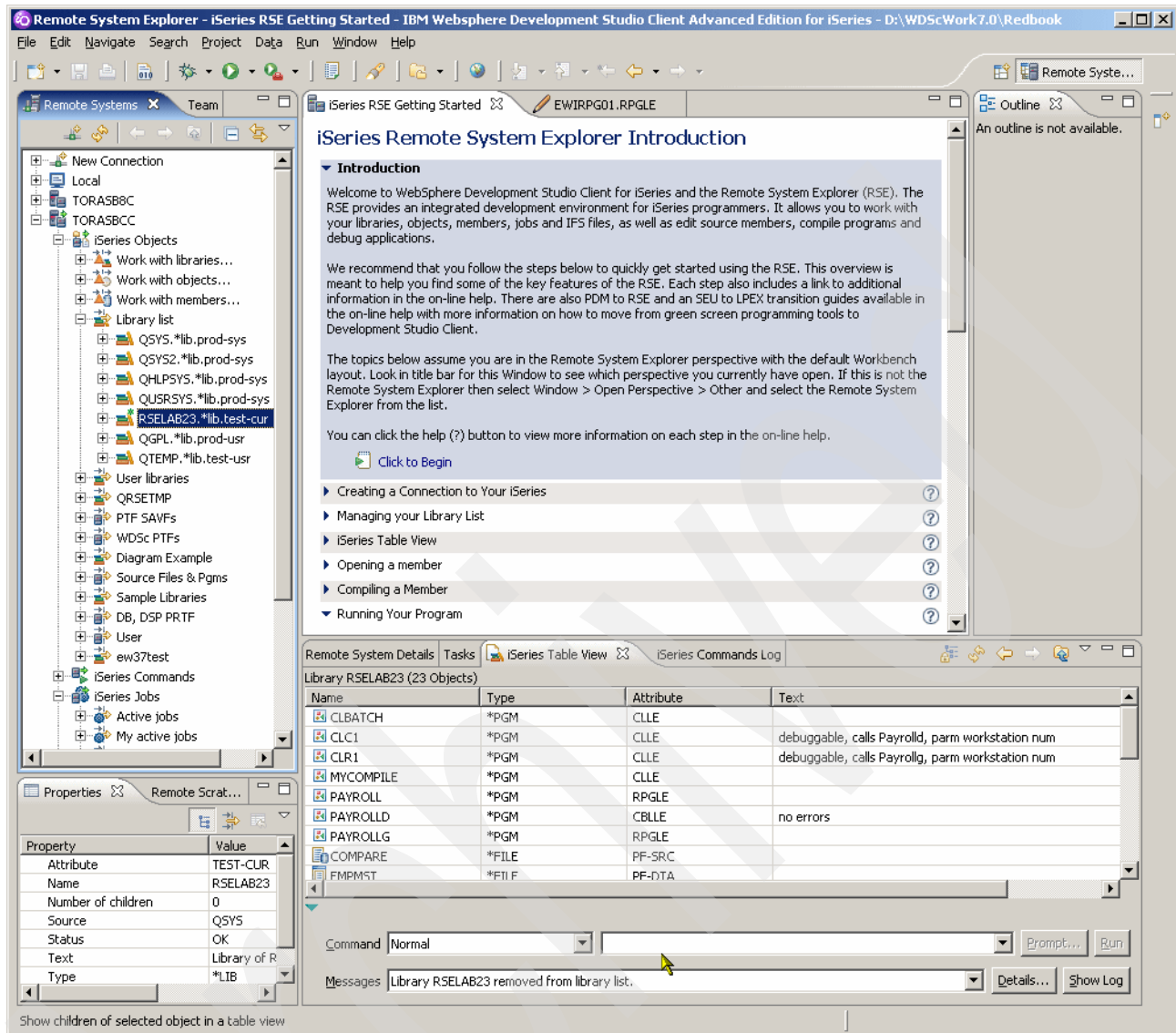


Figure 7-1 The RSE perspective and default views

7.1.1 A guided tour of RSE

The following sections provide a quick introduction to start using the Remote System Explorer, briefly called RSE in the rest of the book.

One of the enhancements in WDSC V7.0 and WDSC AE V7.0 was the addition of a System i RSE Getting Started page that guides you through the steps when you use RSE for the very first time. The following sections show the same examples along with figures of some of the dialogs and more background information.

Creating a connection to your System i

The first step in using the RSE is to create a connection to your System i:

1. Locate the Remote Systems view. By default this is located on the left-hand side of the workbench.
2. Expand the New Connection entry in the tree, and then expand System i.

3. You will be prompted for a profile name if this is your first time creating a connection. This is not your i5/OS user profile, this is a name that can be used to group and share RSE information with the rest of your team Figure 7-2. Enter a name that uniquely identifies you or accept the default name. Click **Next**.



Figure 7-2 New RSE profile name

4. Enter a Connection name and the TCP/IP Host name for this connection. The Connection name can be any name you want to associate with this connection and will appear in the Remote Systems view. Typically this would be the system name of your i5/OS, but it does not have to be that name. Note, however, you can create multiple connections for the same i5/OS host—for example, to use a different user profile, library list or other settings when working with different applications.
5. Click **Finish**.

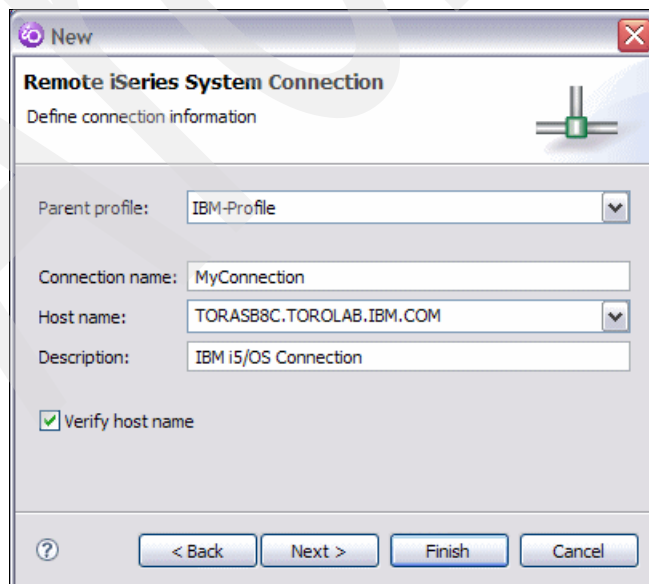


Figure 7-3 New System i connection information

6. Your connection now appears in the Remote Systems view; Figure 7-4.

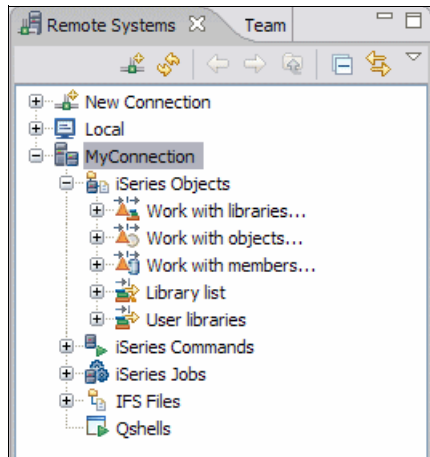


Figure 7-4 System i connection in Remote Systems view

When you expand your connection you will see entries under the connection that allow you to access different i5/OS resources. These entries are called subsystems in the RSE (they are not related to i5/OS job subsystems). The System i Objects subsystem allows you to access and work with your libraries, objects and members.

Managing your library list

To manage your System i library list:

1. Expand the Library list filter below the System i Objects subsystem to see the library list being used by the RSE connection; Figure 7-5.

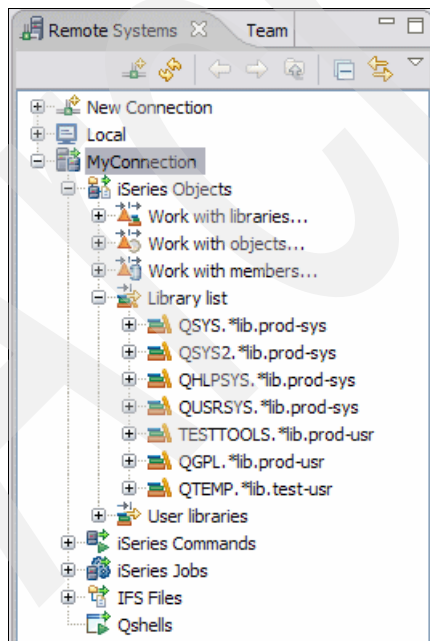


Figure 7-5 RSE connection library list

2. If you are already not connected to your system, you will be prompted for a User ID and Password as shown in Figure 7-6.

Note: The library list for the RSE connection is set based on the job description that is associated with your i5/OS user profile. However, we describe in 7.5, “Working with your library list” on page 204 how you can modify that library list.

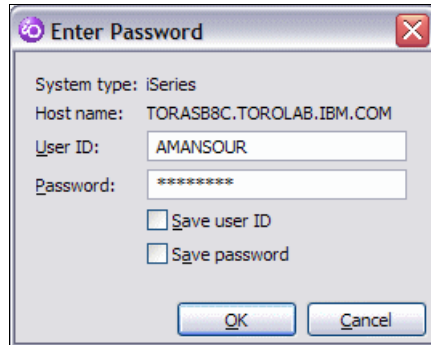


Figure 7-6 Connecting to the System i system

3. Enter your user ID and password, and click **OK**.

You can customize the library list in one of two ways:

- ▶ Using the context menu actions to change the library list immediately
- ▶ Using the Initial Library List property page for the subsystem

Using the pop-up menu actions for the library list filter

In the library list filter, you can perform several actions to modify your library list. These actions correspond with the *Edit Library List* (EDTLIBL) i5/OS command. When we say *your* library list, we actually mean the automatically started server job which accesses i5/OS on behalf of you while you work in the Remote Systems view or Remote Systems LPEX editor. The changes you make are only in effect until the RSE connection is disconnected.

- ▶ To add a library to the library list of your server job, right-click the Library List filter and select **Add Library** from the context menu, as shown in Figure 7-7.
- ▶ To change the current library of your server job, right-click the Library List filter and select **Change Current Library** from the context menu, as shown in Figure 7-7.

We talk about this in more detail in 7.5, “Working with your library list” on page 204.

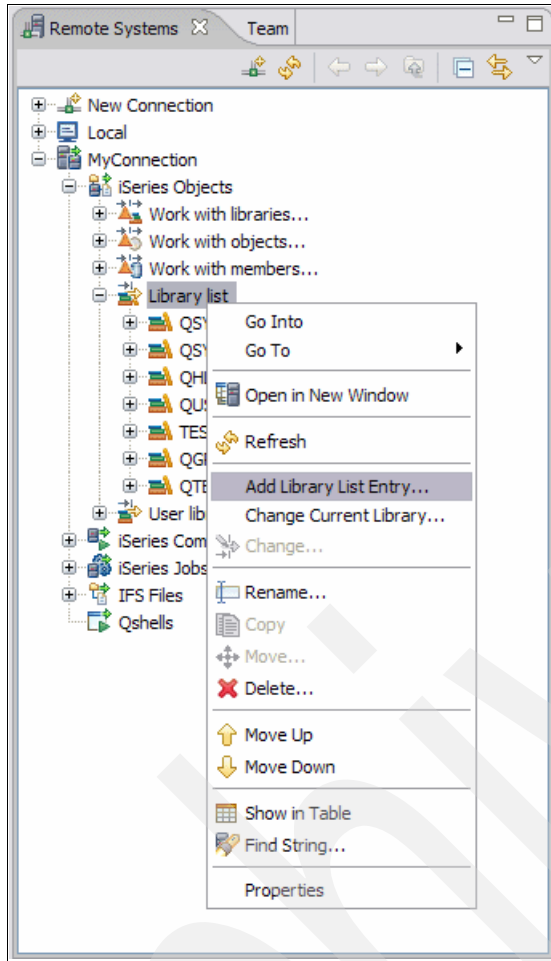


Figure 7-7 Library list pop-up menu functions

System i Table view

The Remote Systems view shows your i5/OS libraries, objects and members using a tree structure. If you would prefer a table view similar to PDM then you can use the System i Table View. You can perform the same actions with the items in a System i Table as you can in the Remote Systems view. Different is the amount of information you see for each individual library, object, or member and how you navigate to different items.

The Table view adds the ability to view descriptions and other attributes, and perform more PDM-like actions such as defining a subset of the items you want to work with. You can also sort items easily by clicking the heading of any of the columns.

We describe System i Table views in more detail in 7.6, “System i Table views” on page 211.

To use the table view, right-click a filter, library, program or file in the Remote Systems view and select **Show in Table** from the context menu, as shown Figure 7-8 on page 147.

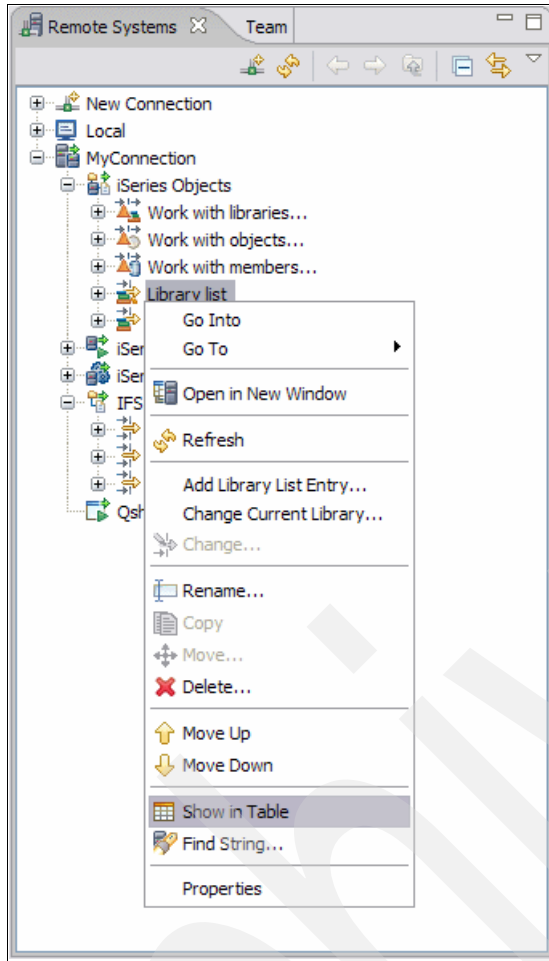


Figure 7-8 Show in Table view

Working with items in the Table view through the context menu

In the System i table you can right-click any library, object or member and get the same pop-up menu actions as you would see in the Remote Systems view; see Figure 7-9.

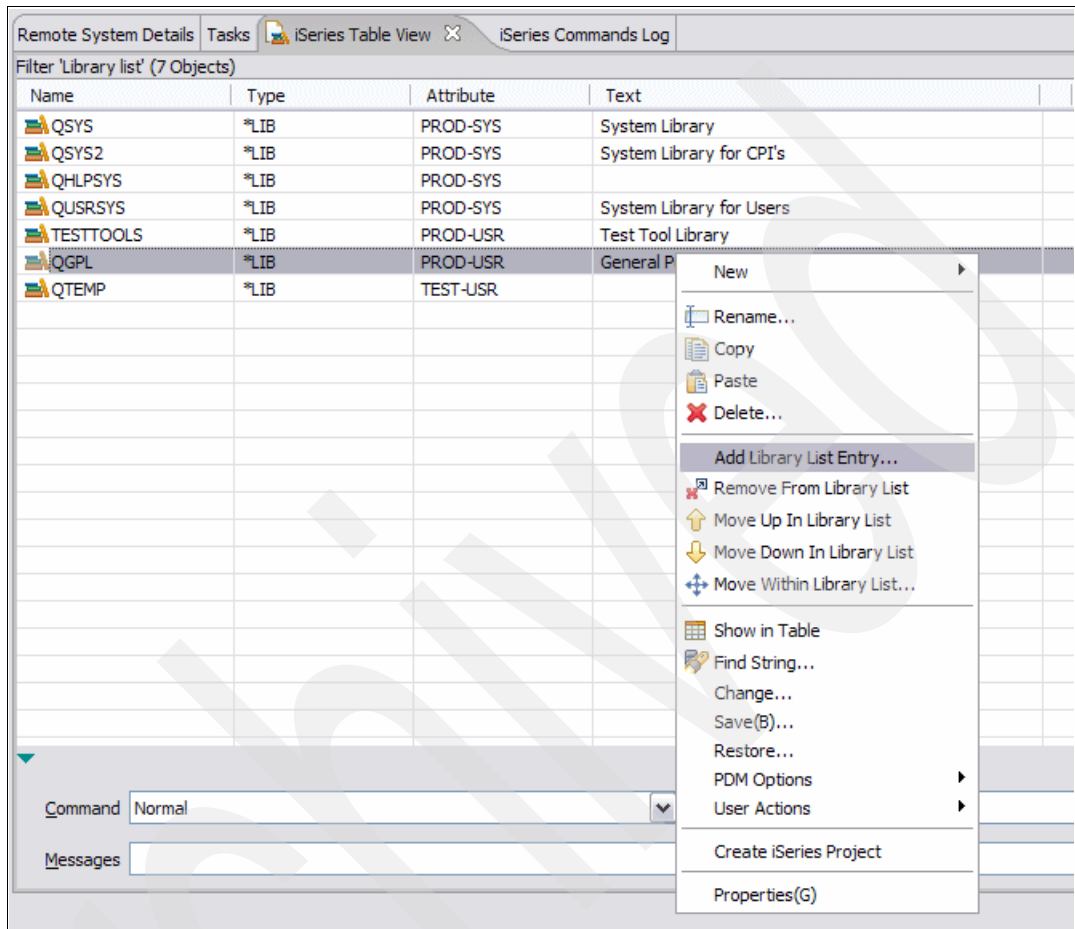


Figure 7-9 Table view pop-up menu actions

Double-clicking a library or physical file will display the contained objects or members in the table. Double-clicking a member will open the member for editing in the Remote Systems LPEX Editor, as shown in Figure 7-10.

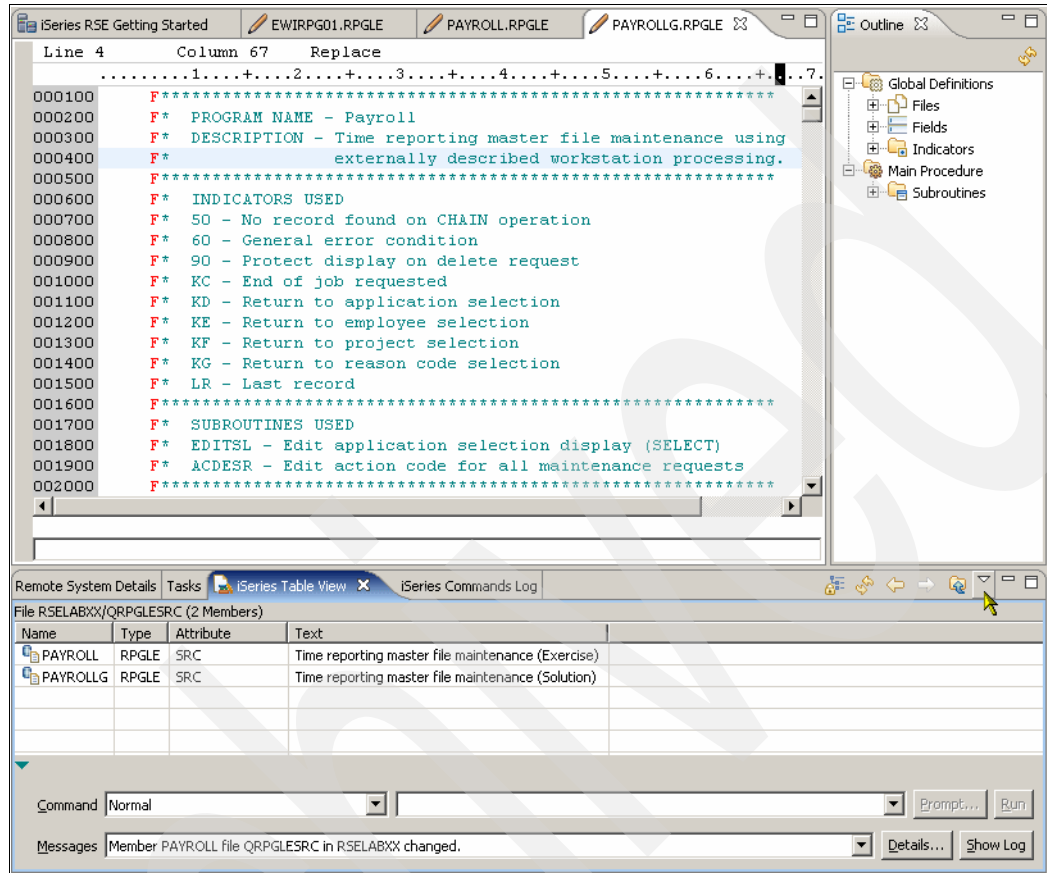



Figure 7-10 Opening a source member from Table view

Table view pull-down menu

You can also perform additional actions from the view's pull-down menu. To open the pull-down menu, click the upside-down triangle in the top right corner of the view  as indicated by the mouse pointer in Figure 7-10.

The Table view pull-down menu actions are shown in Figure 7-11 on page 150; we briefly describe them here:

- ▶ You can use the Work With actions in the System i Table View's pull-down menu to select different types of items (such as libraries, objects, or members) to be shown in the Table view. In the same submenu you may also select a set of items you worked with previously.
- ▶ You may define a Subset of items to be shown in the table, similar to using F17 in PDM. Having done so, select **Show All** to show all items again.
- ▶ To show all the information for objects and members, from the pull-down menu select **Show Columns** → **All** or you may select your own choice of columns defined in the preferences (see Figure 7-84 on page 213) by selecting **Show Columns** → **Customized**.
- ▶ The option Position to... allows you to position objects in the System i Table view similar to the “position to” and “position to type” functions in PDM by entering the first characters of the name or type.

- ▶ You can select an item in the Table view, and from the pull-down menu select **Print** or **Subset** actions.

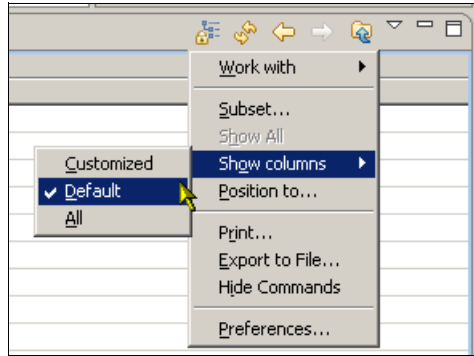


Figure 7-11 Table view work with actions

- ▶ The option **Export to File** allows you to export the contents of the table view to a file, for use in another format for the data, such as a spreadsheet. You can export all contents in the table view, or just selected items. You can also export the data with or without the header and footer information.
- ▶ Depending on whether you currently see the command view area (shown in Figure 7-12), you either hide it or show it again by selecting **Hide Commands** or **Show Commands**.
- ▶ When you select **Preferences...** you may define a customized set of columns and some more preferences as explained in 7.6.1, “Populating System i Table views” on page 212.

Table view command area

The command area at the bottom of the System i Table view allows you to run CL commands and view the resulting messages; see Figure 7-12.

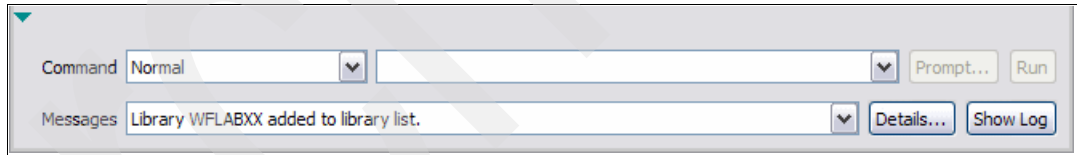


Figure 7-12 Table view command area

We describe System i Table views in more detail in 7.6, “System i Table views” on page 211.

Opening a member

You can open a member of a source physical file for editing by right-clicking its name either in the Remote Systems view (as shown in Figure 7-13) or the System i Table view and selecting **Open With** → **Remote Systems LPEX Editor** from the context menu.

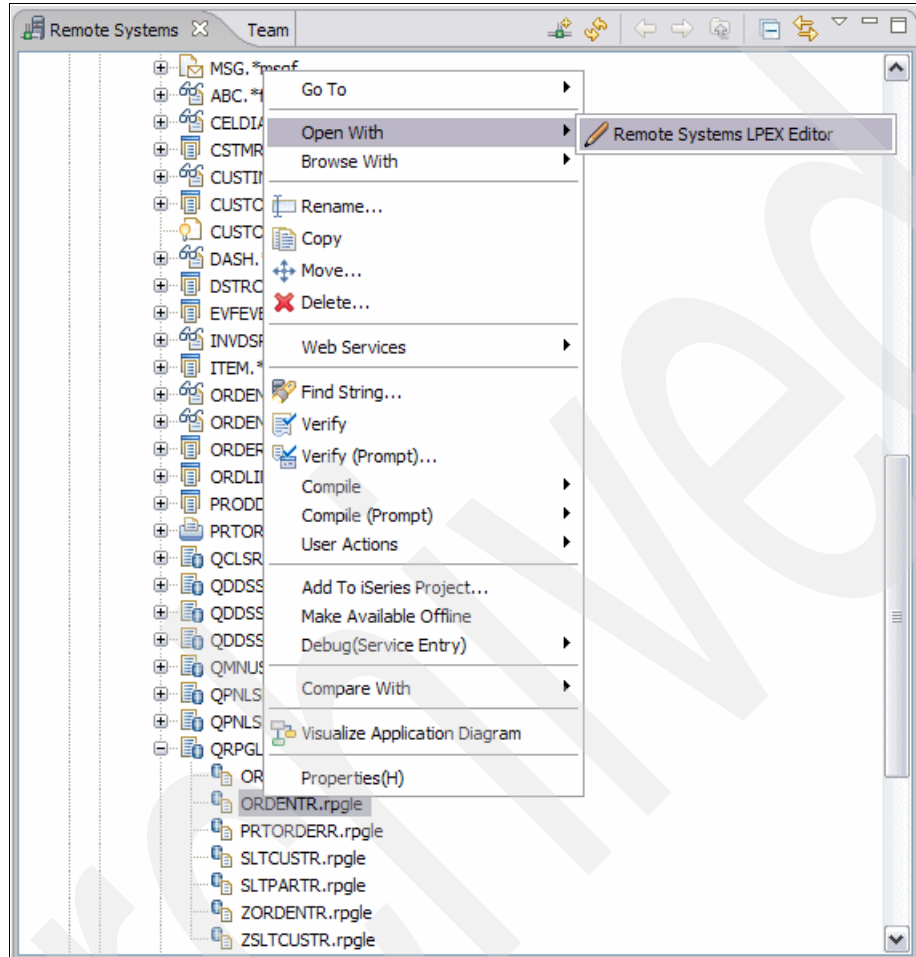


Figure 7-13 Open member in LPEX Editor

This locks the member in i5/OS and opens it in the local Remote Systems LPEX Editor. After you are done making changes you can save the member using either the **File** → **Save** menu option or the **Ctrl+S** keyboard shortcut. The locks on the member are released when you close the editor.

If you know the exact library, file and member name for the member you want to open, then you can also use the Open Member dialog to enter the information instead of selecting the member in the Remote Systems view.

You can open the Open Member dialog in two ways: Either right-click the System i Objects subsystem and select **Open Member** as shown in Figure 7-14, or use the Ctrl+Shift+a keyboard shortcut, which brings you directly to the Open Member dialog shown in Figure 7-15.

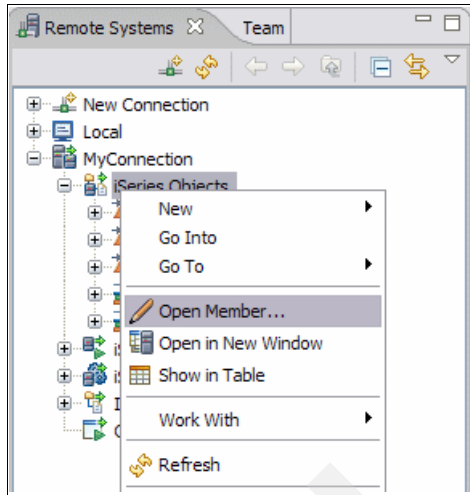


Figure 7-14 Open Member from the Remote Systems view

In the Open Member dialog, enter the library, file and member names and click **OK**; see Figure 7-15.

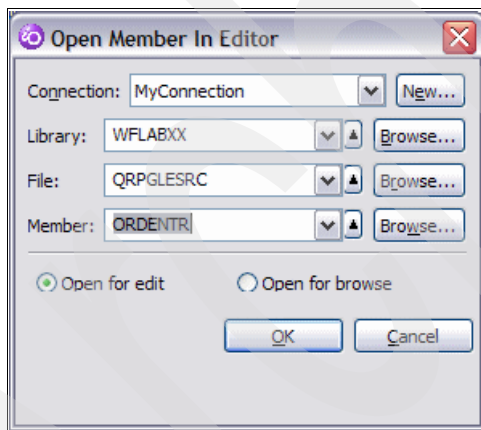


Figure 7-15 Open Member in the Editor dialog

Compiling a member

You can invoke compiles from within the editor or using the context menu actions on a member in the Remote Systems view. While editing a member in the Remote Systems LPEX Editor you will see a Compile menu in the Workbench menu bar, as shown in Figure 7-16. It contains two cascading menus: Compile and Compile (Prompt).

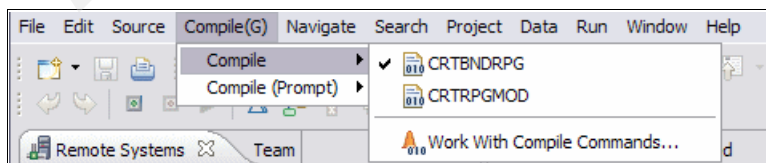


Figure 7-16 Workbench compile menu

Use the **Compile (Prompt)** menu if you wish to prompt the compile command before it runs. Each of these menus cascades to show the standard IBM compile commands associated with the member's source type. You can also use the **Work with Compile Commands** action to customize the compile commands or add your own.

The compile commands are submitted to batch on System i. After the compile completes, the RSE automatically downloads a list of all messages issued by the compiler and displays them in the System i Error List view. Double-clicking entries in this view automatically positions the editor to the line causing the error and inserts the error message into the editor.

Use the **Source** → **Remove Messages** action to clear all inserted messages from the editor. Note that the option **Source** → **Refresh (Ctrl+F5)** also removes the error messages, but in addition also removes any filters and clears any text you may have entered in the prefix area.

Running your program

You can also start a program directly from the Remote Systems view by selecting a choice from the context menu of any program object. How you start it depends on whether it is a batch or 5250 application.

Starting an interactive (5250) application from the Remote Systems view

The easiest way is to open a 5250 emulator outside of the workbench and run your program the way you normally would. However, you can also start the program directly from the Remote Systems view. See “Starting an interactive (5250) application from the Remote Systems view” on page 153 for details on how to do so.

Start a batch program from the Remote Systems view

You can right-click the program in the Remote Systems view and select a run action from the Run As cascading menu; see Figure 7-17.

- ▶ **Batch**

The Batch action runs your programming using the Submit Job (SMBJOB) CL command from the RSE connection.

- ▶ **Multi-Threaded**

The Multi-Threaded action is only applicable to multi-threaded C/C++ applications.

- ▶ **System i application in an RSE job**

The System i application in an RSE job runs your program in the server job used by the RSE connection. This can be useful if you want to call a program to perform some job setup for the RSE connection. However, you will not be able to use the RSE connection while the program is running.

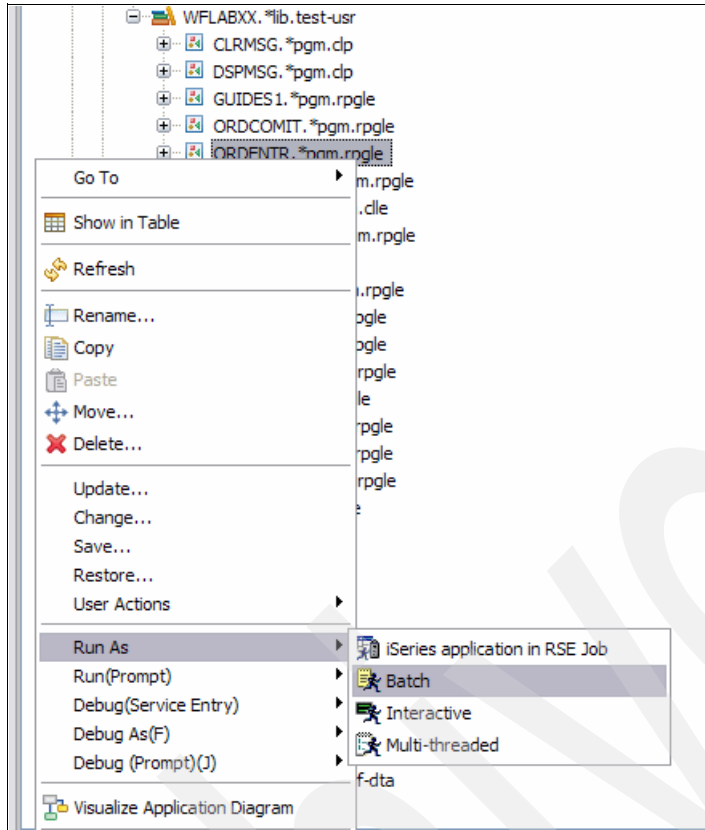


Figure 7-17 Run batch programs from the Remote Systems view

Debugging your program

There are many ways to launch your application for debugging with the Integrated System i Debugger in Development Studio Client. One of the easiest ways is to use Service Entry Points. A Service Entry Point can be thought of as a system-wide breakpoint set on a program, service program, module, or procedure that is specific to jobs running under your user profile. Here are the steps for using Service Entry Points:

1. In the Remote Systems view, select the program, service program, module or procedure that you want to debug. You can find modules and procedures by expanding program and service program objects in the Remote Systems view.
2. Right-click and select **Debug (Service Entry)** → **Set service Entry Point**; see Figure 7-18.

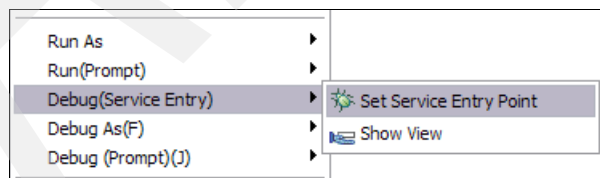


Figure 7-18 Set Service Entry Point

3. This sets the Service Entry Point and adds it to the System i Service Entry Points view; see Figure 7-19. You can use this view later on to remove, disable, or refresh the Service Entry Point.

Library	Program	Program Type	Module	Procedure	User ID	Connection	Enabled
WFLABXX	ORDENTR	*PGM	*ALL	*ALL	AMANSOUR	MyConnection	Yes

Figure 7-19 System i Service Entry Points View

- Run your application. You can start it using one of the RSE options described in “Running your program” on page 153 or any other method to invoke a program under i5/OS. However, you *cannot* run it as a System i application in an RSE job, since it will cause the workbench to hang.
- When the program, service program, module or procedure that you set the Service Entry Point on is called, the i5/OS job will suspend and the Integrated System i Debugger inside of the Development Studio Client will attach to the job. This causes the Debug perspective to open.
- Inside of the Debug perspective, the Debug view in the top left corner shows all of your active debug jobs and the current call stack for each one that is currently suspended; see Figure 7-20. You can use the icons in the toolbar for the Debug view to control the debugger or use the corresponding Step actions under the Workbench's Run menu.

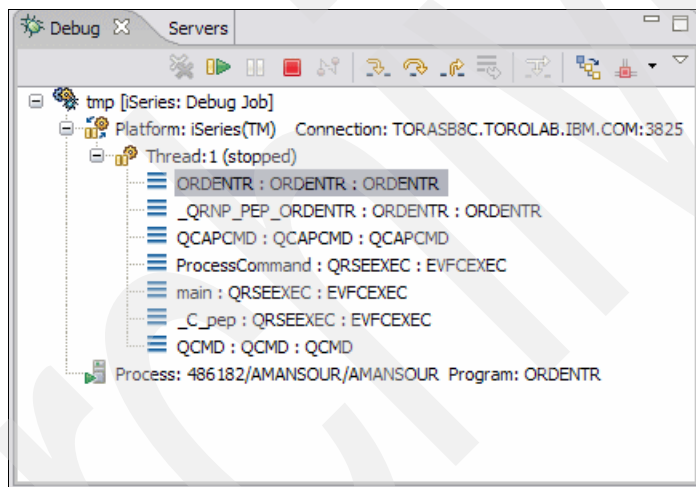


Figure 7-20 Debug view

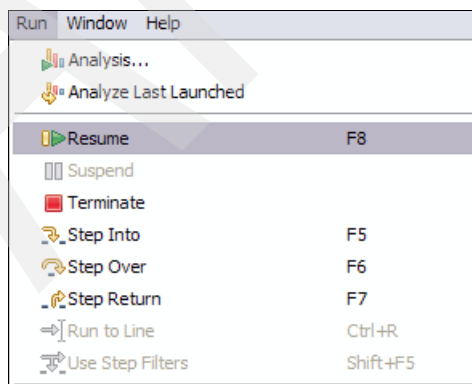


Figure 7-21 Debugging step actions from the Run menu

7. To set a line breakpoint, select any executable line in the source, then right-click and select **Add Breakpoint** from the pop-up menu. All breakpoints are shown in the Breakpoints view.
8. To view or change the value of a field, select the field in the source, then right-click and select **Monitor Expression@** from the pop-up menu. This adds the field to the Monitors view. As you step through your program, variables in the Monitors view will be automatically updated to show their current values.
9. The RSE perspective stays active while you are in the Debug perspective. All open perspectives are shown in the top right corner of the workbench. You can switch to another open perspective by selecting it there (you may need to click the >> symbol to see a complete list of open perspectives). Alternatively you can use the Ctrl+F8 shortcut key to toggle between open perspectives.

Using filters

Working with large lists of objects or members in the RSE is not advisable for both practical (it is difficult to scroll through a list of hundreds of items) and performance reasons. This is why the RSE was designed with very powerful filtering capabilities.

The System i Objects subsystem allows you to create library, object, and member filters. These are roughly equivalent to these PDM commands:

- ▶ Work with Libraries Using PDM (WRKLIBPDM)
- ▶ Work with Objects Using PDM (WRKOBJPDM)
- ▶ Work with Members Using PDM (WRKMBRPDM)

7.2 Connecting to your i5/OS server

Before you connect to your System i host via the RSE, you must define a profile and a connection.

The first connection

See 7.1.1, “A guided tour of RSE” on page 142 for an easy step-by-step guide to create your very first connection from WDSC to your i5/OS system. The following sections provide more detailed information on RSE connections.


7.2.1 Profiles

Profiles are RSE entities that help you organize and share the information about RSE connections when you have a number of connections. Profiles can be *private* (previously called *personal*) or *shared*. Shared profiles are known as *team profiles*. Team profiles allow RSE resources, such as connections and filters, to be shared among members in a development team.

In order to enable team support, you need to have an Eclipse-enabled change management repository such as CVS, IBM Rational ClearCase, or similar. The team profiles contain all the connections, filters, user actions and compile commands. Whenever these items are created, you are prompted for the profile to create them in. Whenever they are shown, the totals from all active profiles are shown. By default, team members only have their own profile and the Team profile active.

To create a new profile in the RSE, follow these steps:

1. Switch to the RSE perspective.

2. Click the **Menu**  button in the toolbar of the Remote Systems view. Select **Work With Profiles**. You can also click the Team tab in the Remote Systems view.
3. Right-click **RemoteSystemsConnections** and select **New Profile**, as shown in Figure 7-22.

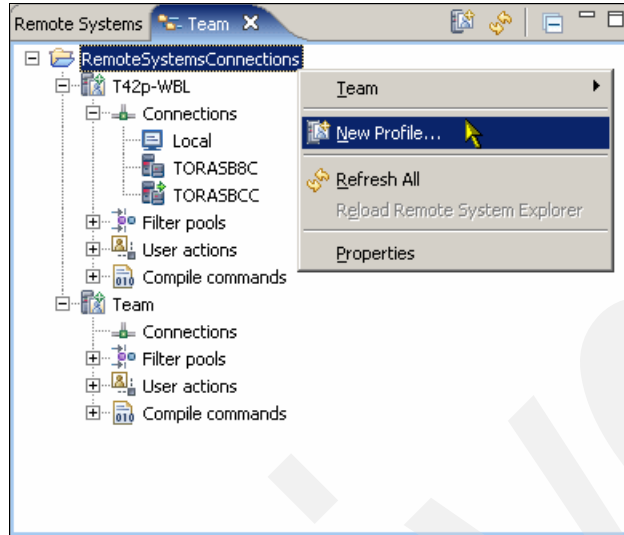


Figure 7-22 Create a new profile in the RSE

4. The New Remote System Profile window (Figure 7-23) opens. Complete the following actions:
 - a. Enter the new profile name in the Name field.
 - b. Select the **Make active** box to make this profile active.
 - c. Click **Finish**.

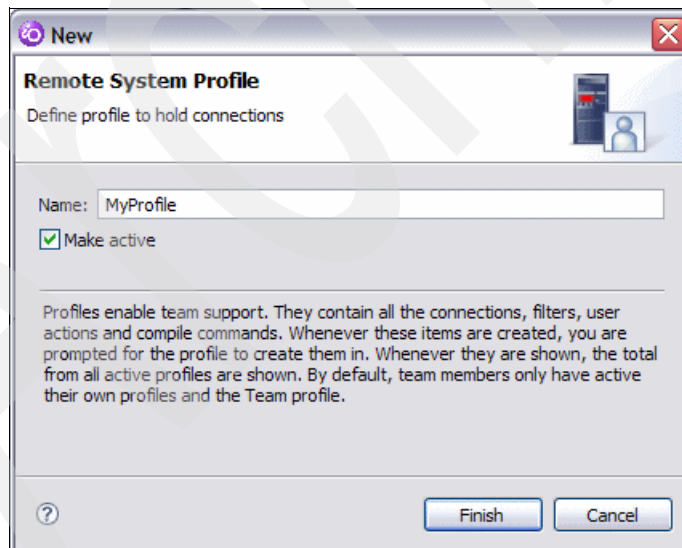




Figure 7-23 Defining a new profile

Profiles defined in the RSE perspective are unrelated to System i user profiles. These profiles are used to hold connections.

Important: Only connections for active profiles are displayed in the Remote Systems view.

To deactivate a profile, follow these steps:

1. Switch to the Team tab in the remote systems view.
2. Right-click any of the active profiles.
3. Active profiles have a decorator next to their icon . Click **Make Inactive** to deactivate it and the decorator disappears .

To delete a profile, follow these steps:

1. Switch to the Team tab in the remote systems view.
2. Right-click any profile and select **Delete**.


Important: Deleting a profile also deletes the connections held by that profile.

7.2.2 Connections

RSE connections allow you to access your System i source members and objects. You can create different connections with different names to the same host. Each connection has its own server job so you have multiple connections, each with different specifications such as library list, set of filters, and so on.

Now that you have created a profile, you can create connections to one or more System i host systems.

To create a new connection in the RSE, follow these steps:

1. Switch to the RSE perspective.
2. Open a New Connection window using one of the following options:
 - In the Remote Systems view, expand the **New Connection** node and then expand the **System i** node.
 - Click the **Menu**  button in the Remote Systems view and select **New Connection**.

The New - Remote System Connection window (Figure 7-24) opens.

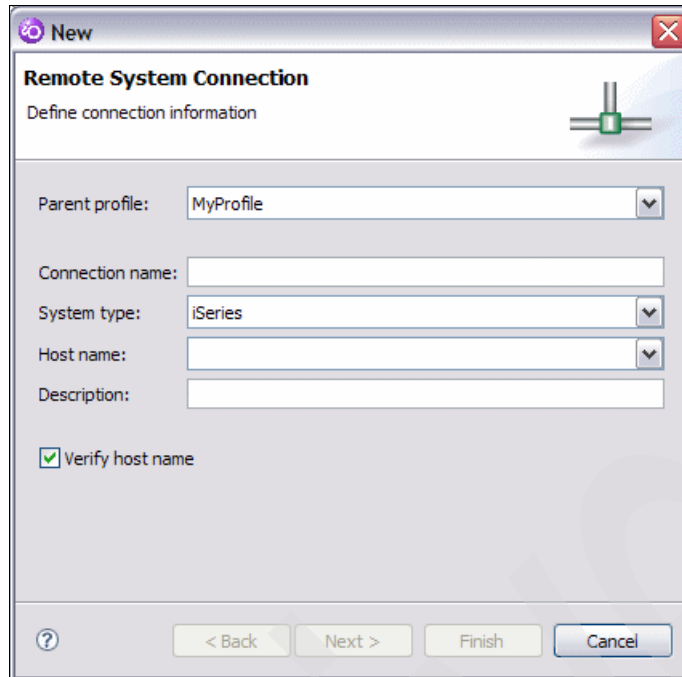


Figure 7-24 New -Remote System Connection window for defining a connection

3. Click the **Parent profile** drop-down list. To keep your resources private, select a profile you defined (see 7.2.1, “Profiles” on page 156). To share your RSE resources, select the **Team profile**. Note, connections created in your non-private profile do not have any default filters created, that is, the subsystems will have nothing underneath them until you explicitly create a filter.

Important: Only active profiles are displayed in the Parent profile drop-down list.

4. Enter values for your connection as follows (see Figure 7-25):
 - a. In the Connection name field, specify a name for your connection.
 - b. Click the System type drop-down list and select the remote host operating system.

Note: The System type drop-down list only shows if you selected the New Connection option from the Menu button in the Remote Systems view.

- c. In the Host name field, specify the name of the System i host. This can either be the system name or the IP address.
- d. Optionally enter some text in the Description field that describes your connection.
- e. Select the **Verify host name** box. This step is optional and the box is checked by default. We recommend that you keep the default action. Selecting this option ensures that an error message is displayed at the time you define your connection, in the event that your host name is specified incorrectly.

Figure 7-25 Entering data for the new connection

5. Click **Finish**. The connection that you defined appears in the Remote Systems view.
6. Connect to the remote System i host:
 - a. Right-click either **System i Objects**, **System i Commands**, **System i Jobs**, **IFS Files**, or **Qshells** and select **Connect**. These entities are known as *RSE subsystems* (see 7.3, “RSE subsystems are not i5/OS subsystems” on page 165).
 - b. Enter your user ID and password when prompted.

When you successfully connect, the Connected property in the Properties view (located in the bottom left corner of the RSE perspective) indicates yes.

Tip: At this stage, we recommend that you select the **Verify Connection** option by right-clicking any of the subsystems. This action prompts you for your user ID and password to the System i host. Then it checks for any missing PTFs. However, to check for the most recent PTFs available, your workstation must have a connection to the Internet.

Saving your System i sign-on information

You can save your System i sign-on information within the RSE. This saves you from having to sign on to your System i host when you request a connection through an implicit action such as expanding a subsystem filter. The stored passwords are then retrieved whenever a password is required to connect. When you save your password for a particular remote system, you will not be prompted to sign on when you try to connect to that system.

When you first try to connect to a remote system, you will be prompted with the dialog shown in Figure 7-26. To save your sign-on information, follow these steps when you are prompted to provide your user ID and password via the Enter Password window:

1. In the User ID field, enter your System i host user profile.
2. Select the **Save user ID** check box. This associates your System i user profile with the RSE connection.

3. In the Password field, enter your System i host password.
4. Select the **Save password** check box. This saves and associates your password information to your user ID in the RSE.
5. Click **OK**.



Figure 7-26 Saving System i sign-on information

The next time you connect to the remote system, you will not be prompted to sign on again.

Clearing the saved password

If you saved your password with the RSE connection as described in step 4 on page 161, there may be occasions when you want to remove the password again, for example, if another person needs to use your workspace. To do so, perform the following steps:


1. Switch to the Remote System Explorer perspective.
2. Make sure you are *not* connected to the System i server by right-clicking a subsystem in the Remote Systems view and selecting **Disconnect** as in Figure 7-30 on page 164. If you are already connected, the option is **Connect**, so you do not need to take action.
3. Right-click one of your subsystems, such as System i Objects, and select **Clear Password** as shown in Figure 7-30.

When you connect to your i5/OS the next time, you need to specify your password again.

Changing the saved password

If you saved your password with the RSE connection as described in step 4 on page 161, you may want to change the stored password some time, for example after you changed it directly on I5/OS through the System i Navigator, using the Change Password (CHGPWD) CL command on a 5250 command line, or any other technique.

There is a way to maintain the stored password and user profile names for all your RSE connections by performing the following steps:

1. Open the Remote System Explorer preferences (Figure 7-27 on page 162) by doing one of the following:
 - In any perspective of the workbench, select **Window** → **Preferences** from the menu and expand **Remote Systems**.
 - Click the drop-down menu  button from the upper right corner of the Remote Systems view, and select **Preferences** → **Remote Systems**.

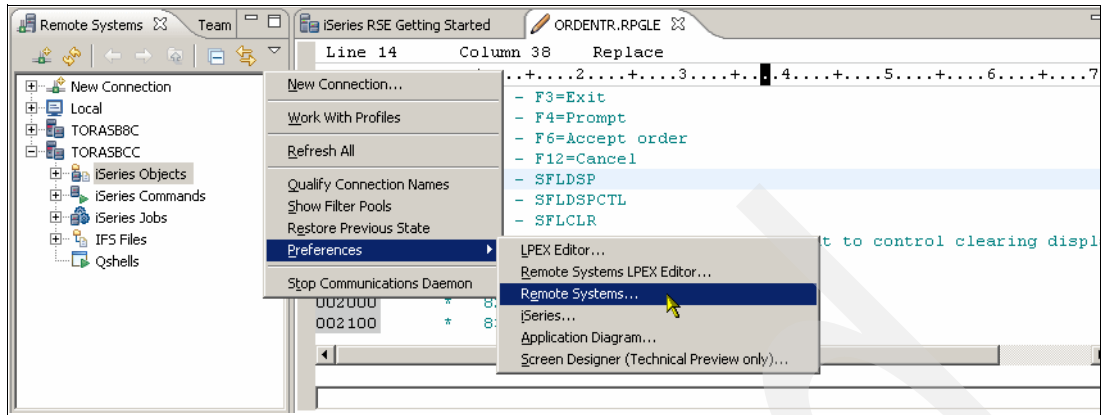


Figure 7-27 Select Remote Systems Preferences

2. Expand **Remote Systems** → **Passwords** as shown in Figure 7-28.

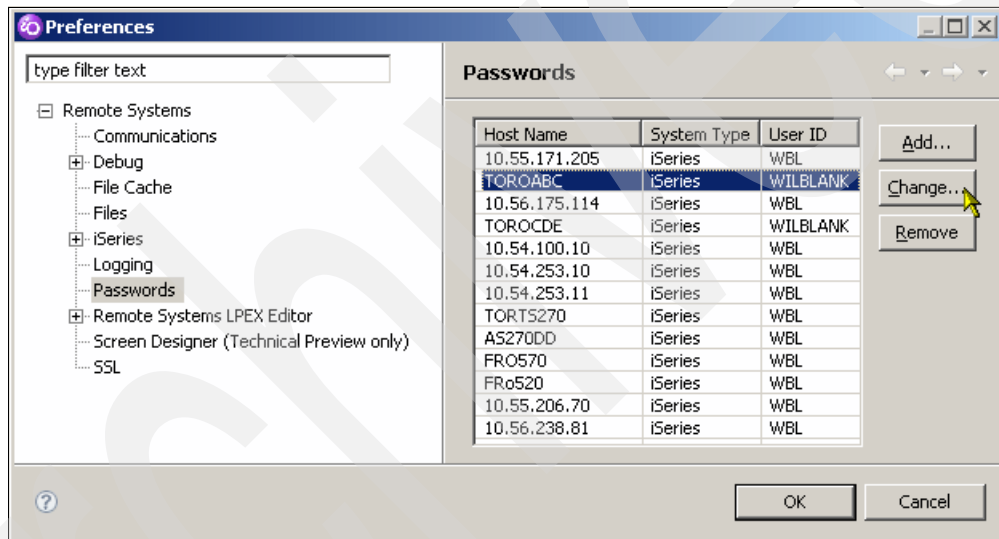


Figure 7-28 Remote Systems Password preferences

3. In the Remote Systems Passwords dialog you may add, change or remove the:

- Host name (or IP address)
- System type (System i, in our case)
- User Id (name of the I5/OS user profile)
- Password

by selecting one of the entries and clicking **Add...**, **Change...**, or **Remove**. To change an existing entry, the dialog shown in Figure 7-29 appears.



Figure 7-29 Change Saved Password

Note, the *host name* is not the RSE connection name, but rather the actual communication address or host name the connection is referring to (as defined in the Host name field in Figure 7-25 on page 160). If you change the host name here, you must also change the host name in the connection properties.

Important: Changing the saved password, as described above, does *not* change the password in your i5/OS user profile. This needs to be done by other methods, such as described in “Changing your i5/OS password in the Remote System Explorer” on page 163, through the System i Navigator, or using the Change Password (CHGPWD) CL command.

Changing your i5/OS password in the Remote System Explorer

You can change the password for your i5/OS user profile that you use to connect to your System i server:

1. Switch to the Remote System Explorer perspective.
2. Make sure you are connected to the System i server by right-clicking a subsystem in the Remote Systems view and selecting **Connect**. If you are already connected, the option is **Disconnect** as in Figure 7-30, so you do not need to take action.

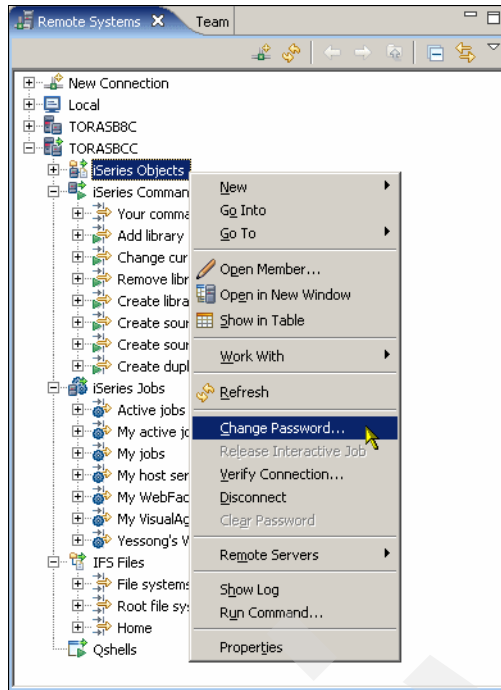


Figure 7-30 Change Password for an RSE connection

3. Right-click one of your subsystems, such as System i Objects, and select **Change Password** as shown in Figure 7-30.
4. Enter your current and your new password in the appropriate fields, as shown in Figure 7-31, and click **OK**.



Figure 7-31 Change Password dialog

The action described above performs the same as the Change Password (CHGPWD) CL command on a 5250 command line.

7.2.3 TCP/IP ports required for WDSC

WDSc uses TCP/IP to communicate between the local system and the remote System i server. The following tables list the TCP/IP ports that are required for the various functions in Development Studio Client. These are the default ports and may have been changed on your System i.

The ports listed in Table 7-1 are all for TCP/IP connections originating from Development Studio Client to the remote System i. You need the first set for Development Studio Client to function properly. If you do not require the last three functions, then you do not need those ports open.

Table 7-1 TCP/IP ports required for WebSphere Development Studio Client for System i

Development Studio Client Function	Required TCP/IP Port
Remote System Explorer (basic functions for retrieving lists, editing, compiling and running).	449 (server mapper host server) 446 (DRDA® (record access) host server) 8470 (central host server) 8475 (remote command host server) 8476 (signon host server)
RSE - Integrated File System access	8473 (file host server)
RSE - Interactive job support (via the Start RSE Server (STRRSESVR) command)	8472 (data queue host server)
WebFacing (Runtime only)	4004 (WebFacing server)

There are two places in Development Studio Client where “call-backs” are used, where a TCP/IP connection originates from the remote System i back to Development Studio Client; see Table 7-2.

Table 7-2 Call-back ports

Development Studio Client Function	Required Local PC Port
RSE - Interactive job support (via STRRSESVR command)	3001 (RSE communications daemon)
Integrated Debugger (WDS Sc V5.0 and V5.1 only)	8001 (Debug daemon) and 3001

Note: All ports (i5/OS and local PC) can be changed. To change the local PC ports use the Workbench preferences pages:

Remote Systems → Communications
Run/Debug → Debug Daemon

Ports and online help

The online help system uses a port to communicate with a local server (at localhost or 127.0.0.1). This port is generated and never conflicts with any other port in use, but a firewall may be configured to block all communication to localhost. The firewall would need to be reconfigured if this is the case in your environment and you want to access online help.

7.3 RSE subsystems are not i5/OS subsystems

Subsystems in the Remote Systems view divide the System i server into functional areas. Each System i connection contains four subsystems:

- ▶ System i Objects
- ▶ System i Commands
- ▶ System i Jobs
- ▶ IFS Files
- ▶ Qshells

Similar to how the functions of PDM are structured, the first subsystem provides access to the libraries, objects, and source members on your System i or AS/400 server as explained in 7.3.1, “System i Objects subsystem” on page 166. In addition, through the other three subsystems, you have the capability to work with commands, jobs and Integrated File System (IFS) files, which are all the artifacts an application developer on OS/400 or i5/OS is dealing with.

These subsystems are displayed as nodes in a tree in the Remote Systems view (Figure 7-32). To view the subsystems for a particular connection, expand the connection node by clicking the plus (+) sign.

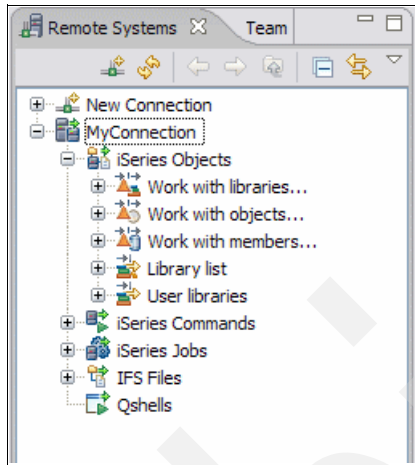


Figure 7-32 Expanding the connection node

Note: Subsystems in the RSE are different from i5/OS subsystems on the System i host.

You can expand each subsystem node by clicking the plus (+) sign.

7.3.1 System i Objects subsystem

This subsystem allows you to manage i5/OS objects in QSYS on your System i host. You can create, copy, rename, delete, and retrieve information about objects. You can also run programs and compile source members.

Expanding this subsystem reveals a set of *prompting filters* that let you create your own filters for viewing libraries, objects, and members. These prompting filters are analogous to the Work with Libraries Using PDM (WRKLIBPDM), Work with Objects Using PDM (WRKOBJPDM), and Work with Members Using PDM (WRKMBRPDM) commands that most System i developers are familiar with. However, unlike the PDM, with RSE you can assign names to your filters, save and reuse them later.

This subsystem contains two default predefined filters for viewing libraries in your Library list and User libraries; see Figure 7-33.

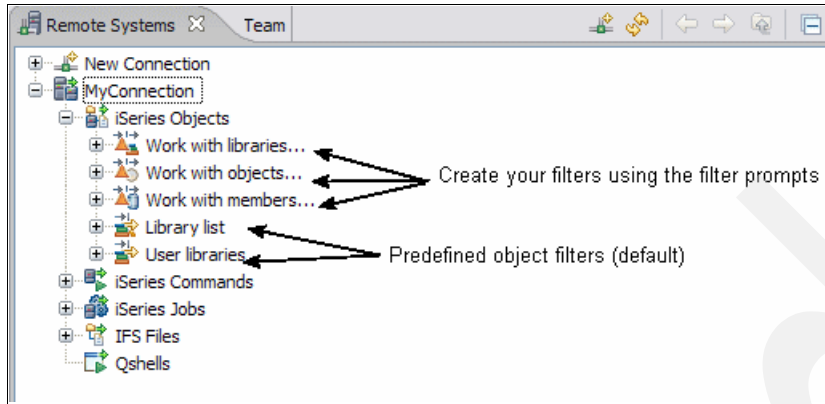


Figure 7-33 The expanded System i Objects subsystem

7.3.2 System i Commands subsystem

This subsystem contains predefined *command sets* that are ready to run when they are expanded in the Remote Systems view. Command sets can contain one or more commands. If the command set has more than one command, then all the commands are run in sequence when the command set is expanded. Output messages from commands are displayed in the System i Commands Log view, by default in the bottom pane of the RSE perspective.

Running command sets

This section explains how to run a command set by executing one of the predefined commands that creates a source member. To execute a command set, follow these steps:

1. In the RSE view, expand your connection by clicking the plus (+) sign.
2. Under your connection, expand the **System i Commands** subsystem.
3. Click the plus (+) sign to expand the **Create source member** command. Alternatively, you can also right-click the command and select **Run(Prompt)**. The Add Physical File Member (ADDPFM) window (Figure 7-34) opens.

If you are not connected to your System i host, you are prompted to enter your sign-on details at this point.

4. Enter your own values for the command parameters. Click **OK** to run the command. The output message from the command displays in the System i Commands Log view.

Notice that you can also select the **Advanced** check box for more command parameters (similar to F10 with a CL command prompt) or the **All Parameters** check box for *all* command parameters (similar to F9).

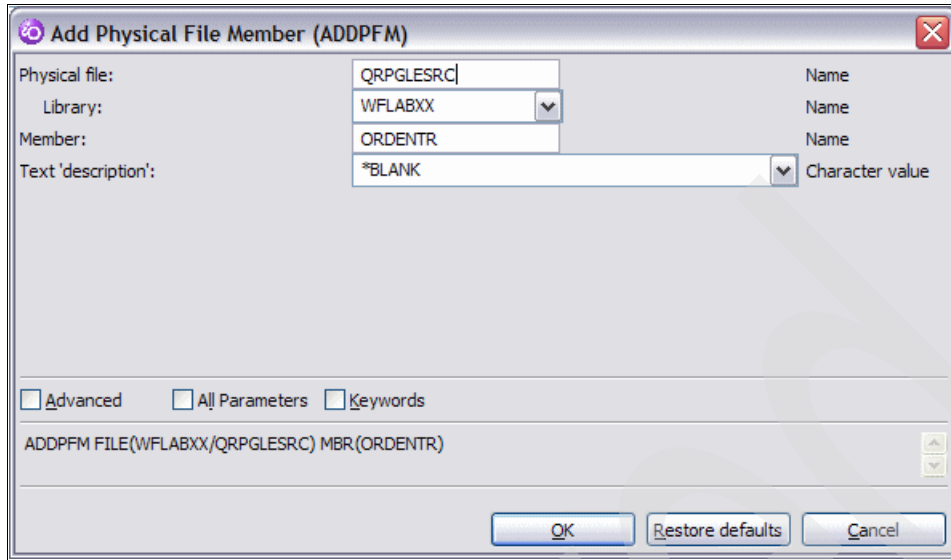


Figure 7-34 Entering values for the ADDPFM command

Creating your own command set

In addition to the predefined command sets, you can create your own command sets. We explain this procedure by creating a command set for the Add Binding Directory Entry (ADDBNDDIRE) command.

To create your own command set, follow these steps:

1. Either expand the **Your command(s)** node or right-click the **System i Commands** node and select **New** → **Command Set**. This action opens a new window for defining your command set.
2. In the New - Command Set window (Figure 7-35), in the Command field, type the ADDBNDDIRE command or you can use the **Browse** button to display the available commands. You may also click **Prompt** to check the syntax of the command you entered and optionally provide parameters for that command.
3. Select the **Prompt when run** check box. This option displays a prompt window for the command when it is run to allow changing or adding values for the command parameters before the command actually executes.
4. Click **Normal**. This option specifies that the command should run in the Remote System server job on the System i host.

Note: You can also choose to run the commands in a batch or interactive job. For more information on running interactive, refer to “Running interactive commands” on page 170.

5. Click **Next**.

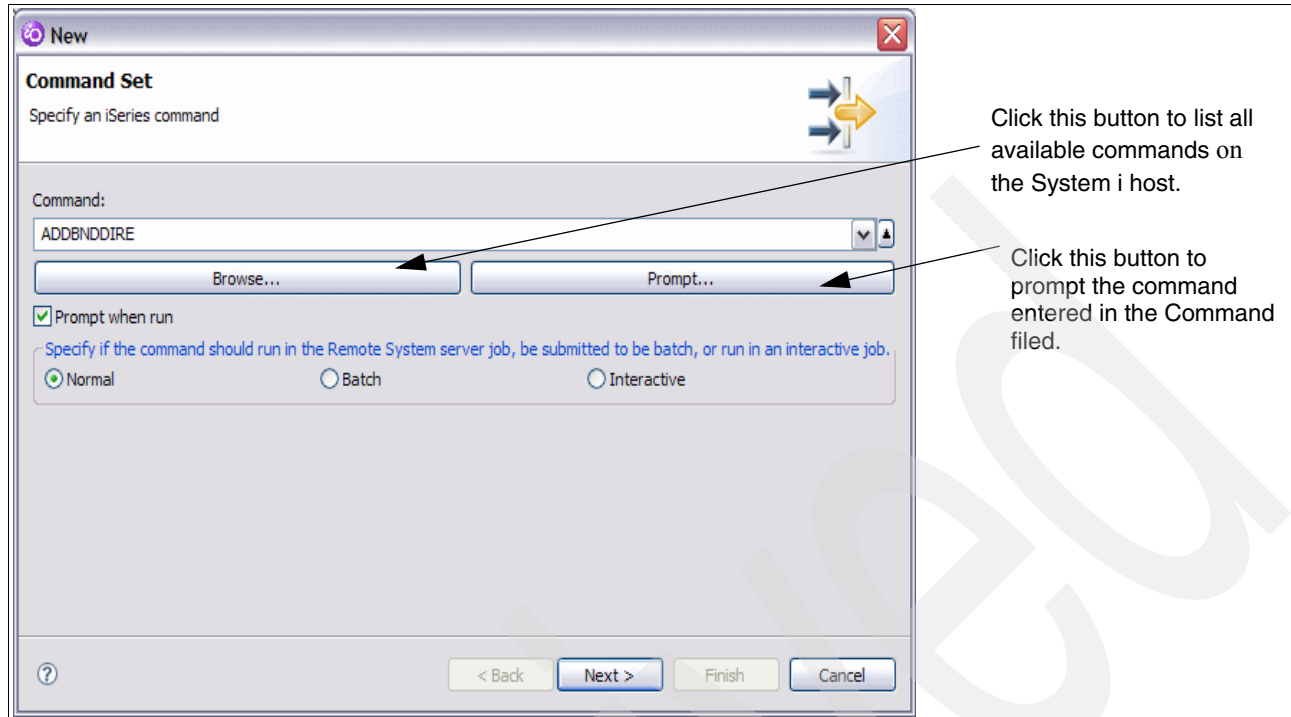


Figure 7-35 Creating the ADDBNDDIRE command set example

6. The New - New Command Set window opens (Figure 7-36). Enter the name of the command set in the Command set name field. In the example, we name this command set Add Binding Directory Entry.

If you select the “Only create command set in this connection” check box, it is not available in connections other than the one you created it in.

You can select either your own profile or the team profile to own this new command set. That way you can choose it to be shared by the team or not.

7. Click **Finish**.

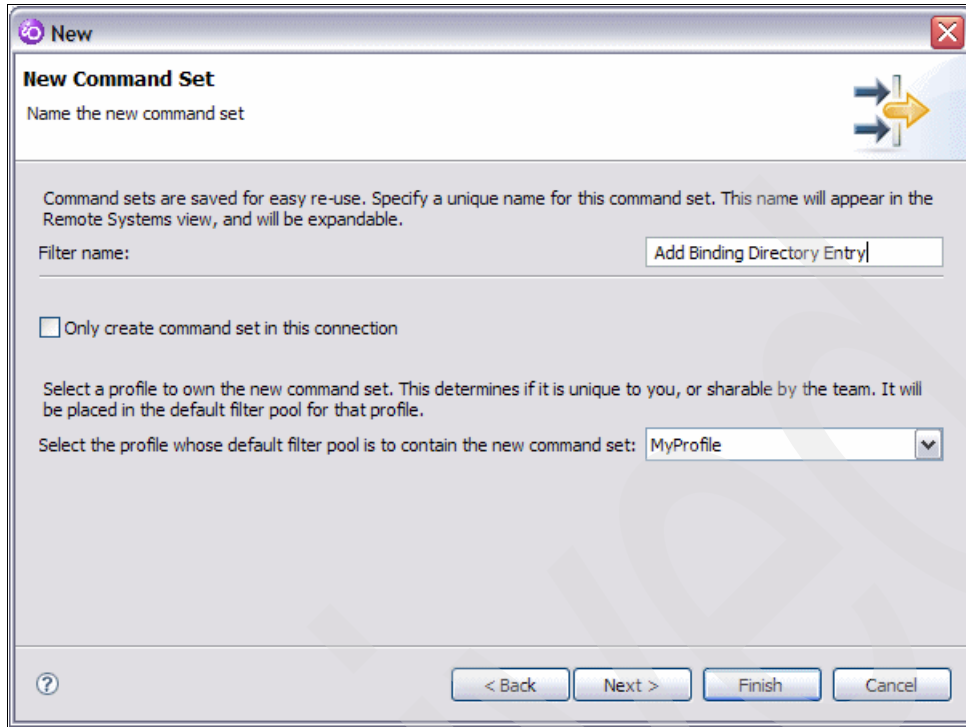



Figure 7-36 Naming the command set

8. At this point, if you are connected to your System i host, the command runs and you are prompted to enter values for the command parameters. For now, you can cancel the command prompt by clicking **Cancel**.
9. The command that was just created appears as a child node called Add Binding Directory Entry to the System i Commands subsystem. The command runs when you either:
 - Expand the command set node by clicking the plus (+) sign.
 - Right-click the command set node and select **Run(Prompt)**.

Note: It might seem surprising that we talk about creating command *sets* here and only specified a single command in the previous example. After you created the command set (with a single command only), you can add more commands by right-clicking its name and selecting **Change** from the context menu. In the Change Command Set dialog you can select **New command** in the Commands field and add new commands to the set.

Running interactive commands

If you run interactive commands, you *must* ensure that you associate an interactive job with your RSE connection:

1. Start the System i communications daemon if it is not already running. Click the drop-down menu  button from the upper right corner of the Remote Systems view. Select **Start Communications Daemon**.
2. Open a 5250 emulator.
3. Run this CL command:


```
STRSESVR NAME(your-connection-name)
```

Note that the connection name is case sensitive here and must exactly match the connection name you used in your workbench. See also “Run an interactive program from the Remote Systems view” on page 242.

Working with command sets

You can change, rename, copy, move, and delete command sets. You can also move them up or down within the list under the subsystem. To execute these tasks, right-click the command set and then select the appropriate action from the menu. Select the **Change** task to add more commands to a command set.

You can find more information about command sets in the online help. From the Help Contents, select **WebSphere Development Studio Client for System i** → **Developing System i server applications using Remote System Explorer** → **Managing commands and shells**.

7.3.3 Starting and stopping the remote servers

The RSE view also allows you to start and stop the various host servers running under i5/OS. To do so perform the following steps:

1. In the Remote Systems view, expand your connection and then right-click **System i Objects**, **System i Commands**, or **System i jobs** and select **Remote Servers**. Then select any remote server whose state you want to change. In this example, we choose to stop the WebFacing server.

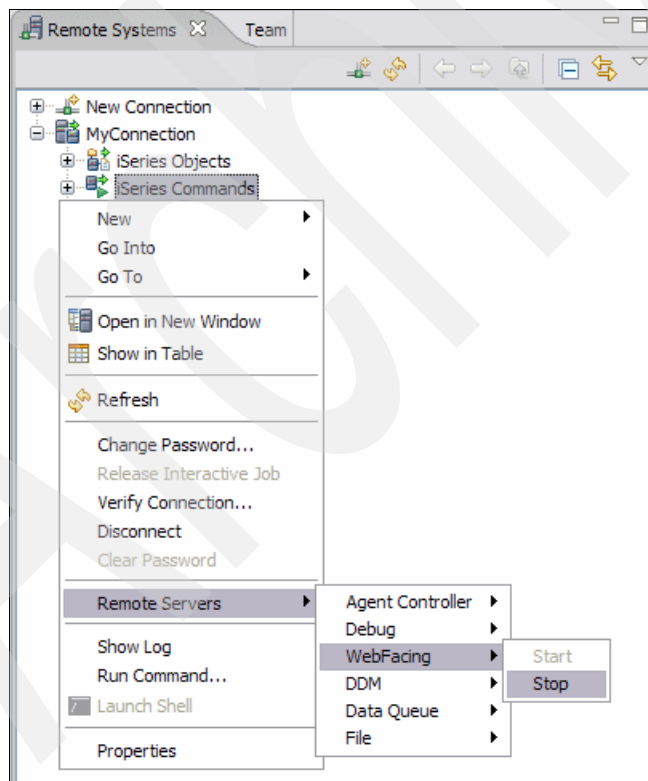


Figure 7-37 Stopping the WebFacing Remote Server

2. Select **WebFacing** → **Stop**, as shown in Figure 7-37.

By looking at the context menu, you can tell whether the selected host server is currently active or not. If it is active, the Start option is greyed out. If it is not active, the Stop option is greyed out. If both options are greyed out, your RSE is not connected to your System i server.

7.3.4 System i jobs subsystem

This subsystem allows you to monitor and manage jobs running on your System i host. Expanding the System i jobs node reveals six predefined filters that are created when you define a connection. These predefined filters are (see Figure 7-38):

- ▶ **Active jobs:** Shows *all* active jobs on your System i host. This is equivalent to running the Work with Active Jobs (WRKACTJOB) CL command.
- ▶ **My active jobs** (previously called Your active jobs): Shows all *active jobs* running under the user profile you used in the currently active connection.
- ▶ **My jobs** (previously called Your jobs): Shows *all* your jobs regardless of their status.

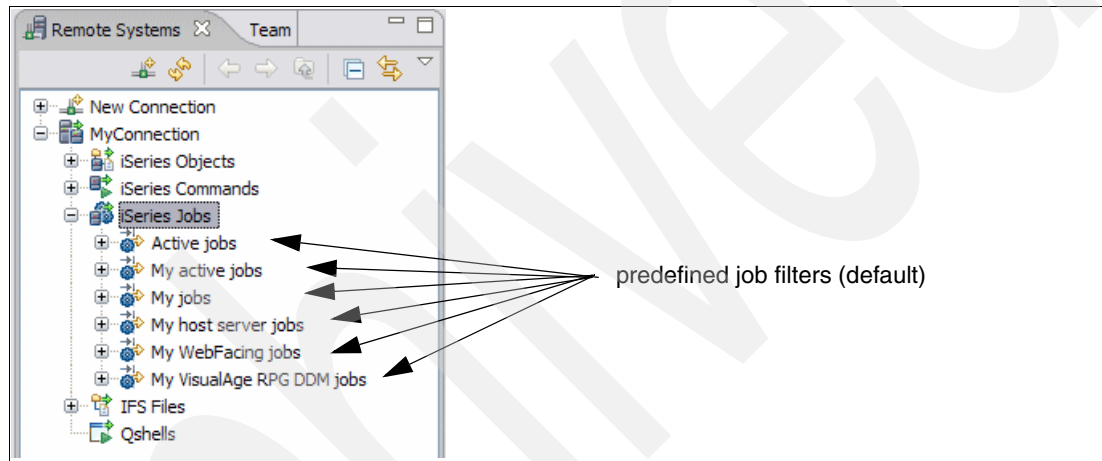


Figure 7-38 The expanded System i Jobs subsystem

- ▶ **My host server jobs:** Shows all jobs serving *your* user profile that start with QZRC initiated by the remote System Explorer and any server job using the remote command host server, such as Java applications, client access, and so on.
- ▶ **My WebFacing jobs:** Shows all jobs serving *your* user profile initiated by a WebFacing project.
- ▶ **My VisualAge RPG DDM jobs:** Shows all jobs serving *your* user profile initiated by VisualAge RPG.

A function not available to CL commands!

Note that the latter three job filters show jobs serving *your* user profile. Unlike the My jobs filter or the Work with User Jobs (WRKUSRJOB) CL command, they most likely do not show your user profile as part of the job name. You rather see names similar to 138567/QUSER/QZRCRSVS.

The reason for this is that many of these servers are implemented as pre-started jobs which are able to serve several different users before they end. To do so, they temporarily switch to the client's job user profile without starting a new job.

Using a 5250 screen, you need do run the Display Job (DSPJOB) CL command for each individual job to find out which user profile it is currently serving. This is a major advantage of the RSE, since there is no single CL command to list all jobs serving a certain user profile.

The jobs shown in the System i jobs subsystem are grouped by job subsystem and if a job is not associated with a subsystem, it appears under “Other jobs”.

Working with System i jobs

To list the jobs on your System i host, expand the job filter. Individual job entries in the tree view are identified by their *job number*, *user*, and *job name*. After you expand the filter, you can perform such tasks as:

- ▶ End jobs
- ▶ Hold jobs
- ▶ Release jobs
- ▶ Add jobs to job status view
- ▶ Display job logs for active jobs

These actions are accessible through a pop-up menu when you right-click a job entry. See Figure 7-39.

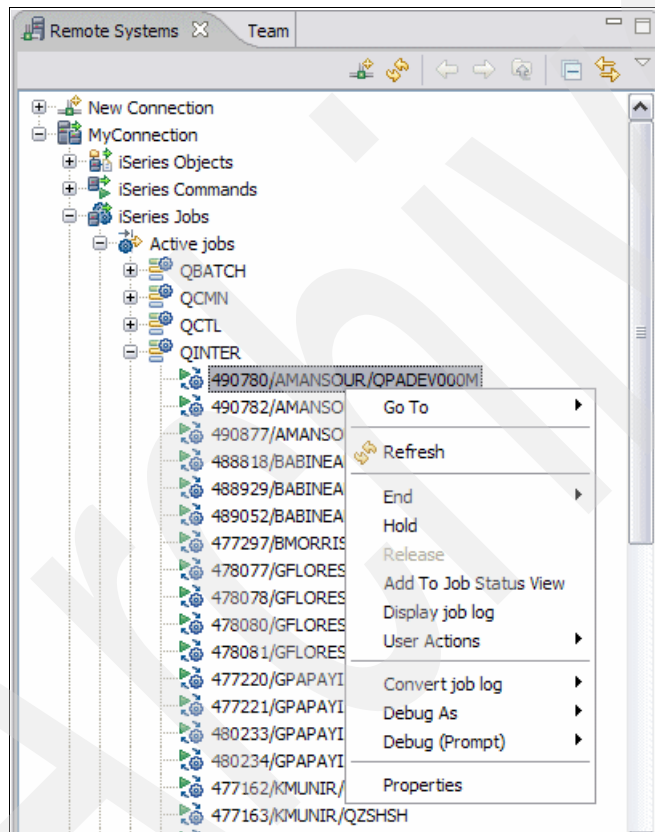


Figure 7-39 Working with jobs in the System i jobs subsystem

Tip: After you complete any tasks in the System i jobs subsystem, we recommend that you refresh the window. Right-click the **System i jobs** subsystem, the job filter, or the actual job itself. Then select **Refresh** (you can also click **Refresh** on the RSE toolbar). This ensures that updated information is displayed.

Properties or attributes pertaining to a particular job are displayed in the Properties view. See Figure 7-40.

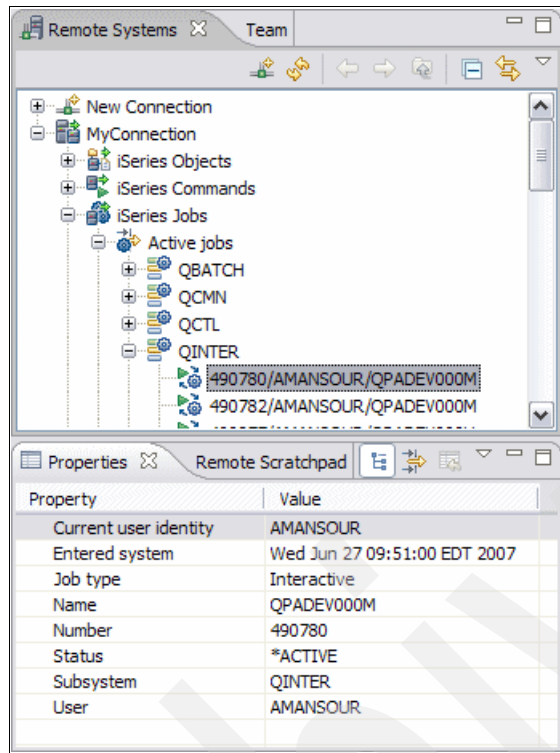


Figure 7-40 Viewing job details in the Properties view

The Properties view only displays a minimal amount of information related to the job. You can find more details about a particular job by right-clicking the job and selecting **Properties**.

Creating your own job filters

In addition to the predefined job filters, you can create your own. This section explains how to do this by creating a filter that displays all active administration jobs that run in the IBM HTTP Server subsystem.

To create a job filter, complete the following steps:

1. Right-click the **System i jobs** subsystem.
2. Select **New** → **Job Filter**.
3. The New - Job Filter window (Figure 7-41) opens.

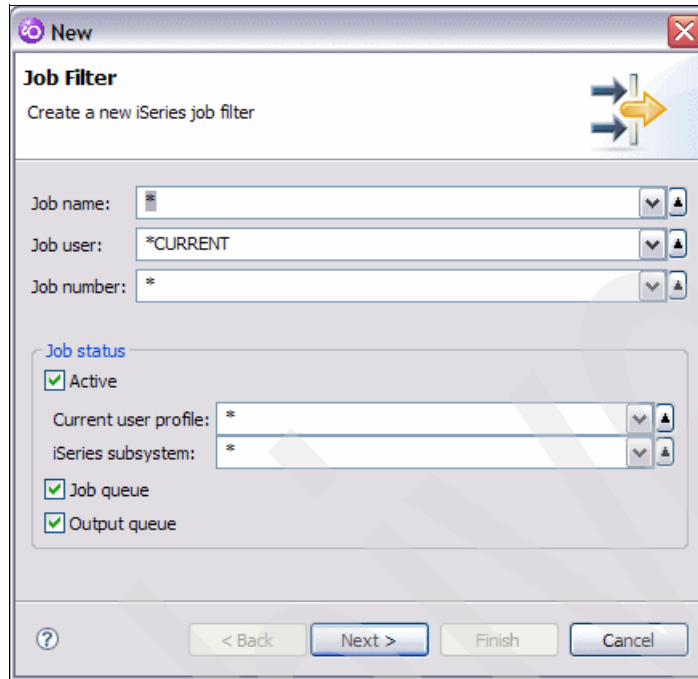


Figure 7-41 The New - Job Filter window for creating job filters

4. Enter values for the new job filter. At this point, you may choose to create a job filter using different values depending on what jobs you have on your system:
 - a. In the Job name field, enter ADMIN.
 - b. In the Job user field, enter QTMHHTTP.
 - c. Deselect the **Job queue** and **Output queue** boxes since we are only interested in displaying active jobs.
 - d. Click **Next**.

5. The next page of the New - Job Filter window as shown in Figure 7-42 opens. Follow these steps:

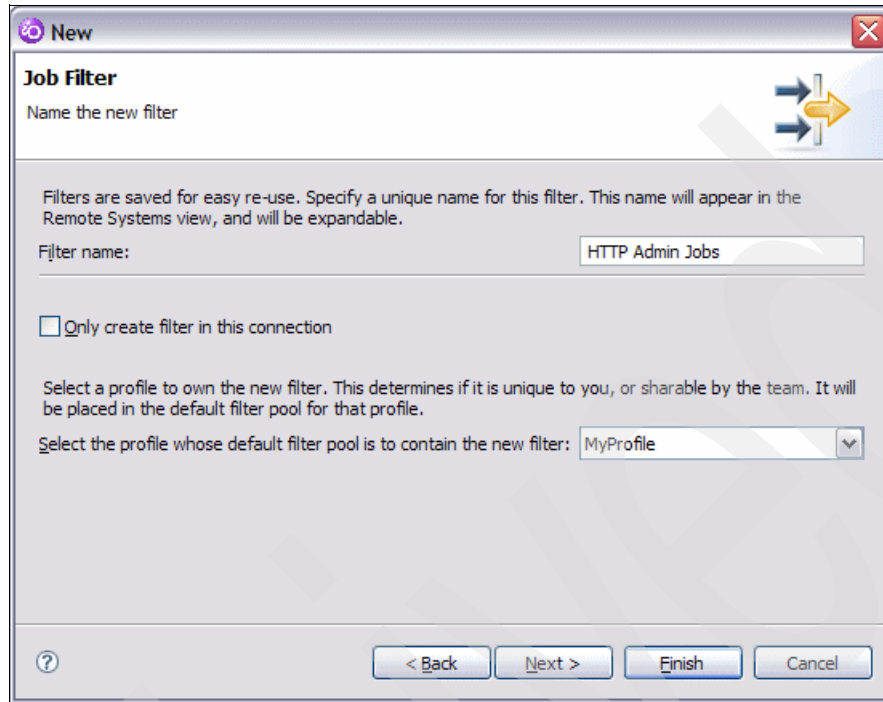


Figure 7-42 Naming the job filter

- a. In the Filter name field, enter the name for the job filter. In this example, we enter HTTP Admin Jobs.
- b. Click the **Owner profile** drop-down list and select a profile to own the filter.
- c. Click **Finish**.

The newly created job filter displays as a child node to the System i Jobs subsystem. You can now expand the filter to display the jobs. See Figure 7-43.

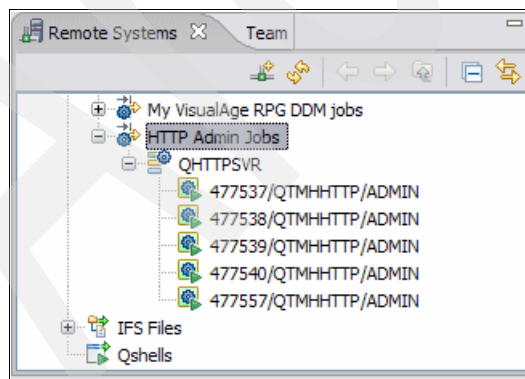


Figure 7-43 Displaying the new job filter

Job statuses are also indicated visually with icons to the left of the job entry.

Working with job filters

You can change, rename, copy, move, and delete job filters. You can also move them up or down in the list under the subsystem. To execute these tasks, right-click the job filter and select the appropriate action from the pop-up menu.

7.3.5 Integrated File System Files subsystem

Access to the System i Integrated File System (IFS) is supported via this subsystem. The IFS provides a common interface to the different file systems that are available on your System i host. Expanding the subsystem reveals three predefined filters (see Figure 7-44):

- ▶ **File systems:** Contains all of the file systems that are available on the System i server.
- ▶ **Root file system:** Displays all folders and files available in this file system. This is the most used file system of all those available besides the QSYS “traditional” library-based file system.
- ▶ **Home:** This is a filter that displays folders and files under the /home folder.

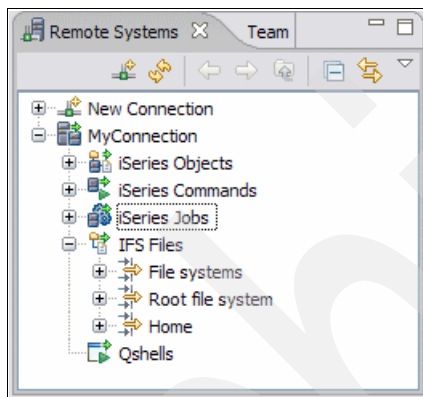


Figure 7-44 The expanded IFS Files subsystem

Creating your own IFS filters

Besides the predefined IFS filters, you can also create your own. We demonstrate this process by creating a filter that displays all files and folders in the path /QIBM/UserData/mqm.

To create your own IFS Files filter, follow these steps:

1. Right-click the **IFS Files** subsystem.
2. Select **New** → **Filter**. The New - File Filter window (Figure 7-45) opens.
3. Enter values for the new file filter. At this point, you may choose to create a file filter using different values depending on what is available in your IFS:
 - a. In the Folder field, enter the path name for the filter to act upon. The easiest way to do this is to click **Browse** and select your path from there.
 - b. Select your subsetting options. You can subset by *file name* or *file type*. Select the appropriate radio button representing these options and then enter your filter information directly. In the case of *file types*, click **Select** to display a list of file types for your selection. You can select *more than one* file type.
 - c. Select the **Show files only** check box if you don't want folders to appear in your filter. For this demonstration, we want this filter to display both folders and files, so we do not select this box.
 - d. Click **Next**.

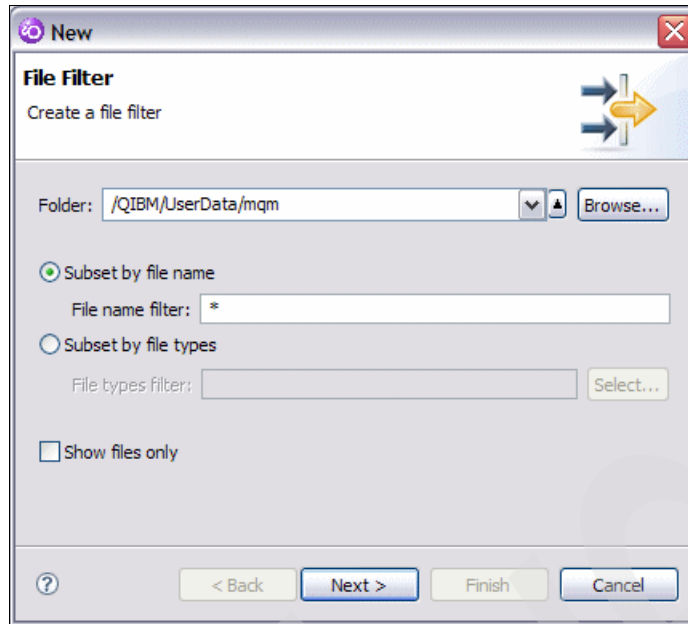


Figure 7-45 Entering information for the IFS File Filter

4. On the next New - File Filter window (Figure 7-46), in the Filter name field, enter the name for the file filter. In this example, we enter WebSphere MQ Files. Click the **Owner profile** drop-down list and select a profile to own the filter. Click **Finish**.

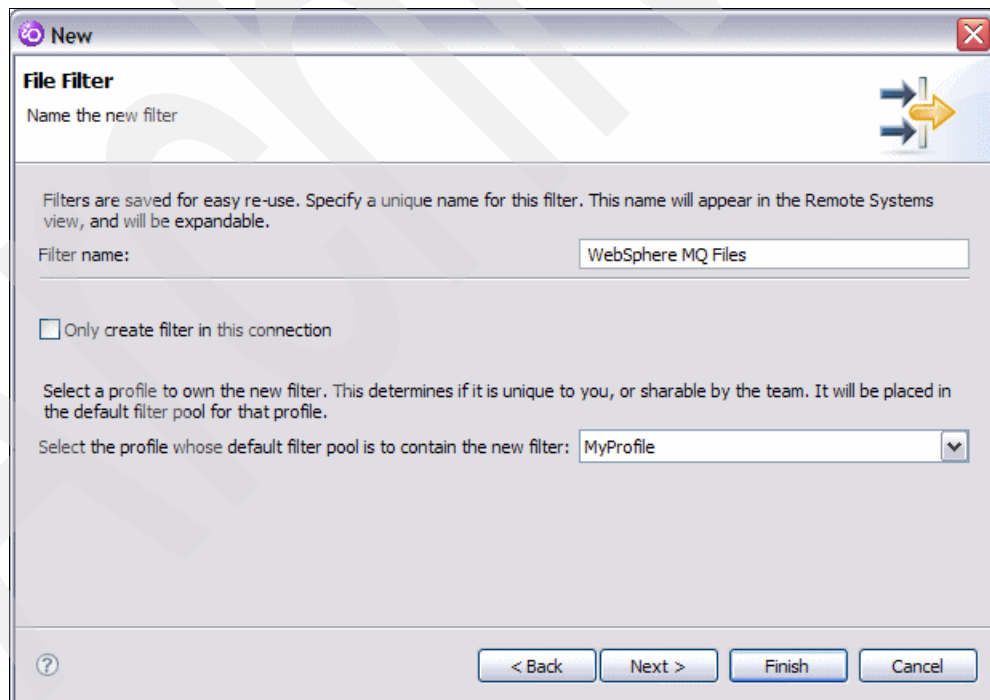


Figure 7-46 Naming the IFS file filter

The newly created IFS file filter displays as a child node to the IFS Files subsystem. You can now expand the filter to display the folders and files specified by this filter. See Figure 7-47.

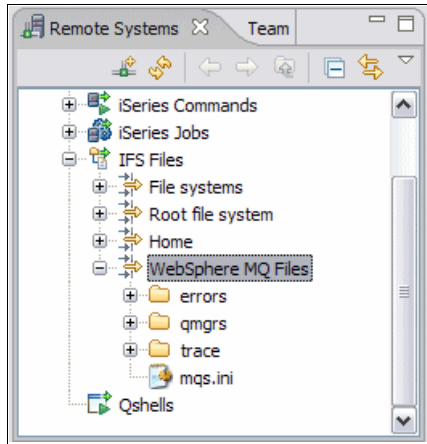


Figure 7-47 Displaying the new IFS file filter

Working with folders and files

You can create, rename, copy, move, and delete folders and files. These tasks are accessible by right-clicking the folder or file and then selecting the appropriate action from the pop-up menu.

Opening files

To open files with an associated editor within the RSE perspective, you can choose one of the following options:

- ▶ Double-click the file. This opens the file using the associated editor. For example, if you double-click a Java file, it opens the file with the Java editor.
- ▶ Right-click the file and select **Open With**. From the submenu, select the editor you want to open the file with.

Searching for folders and files

You can search for files (and folders) in a particular folder. To do this, follow these steps:

1. Right-click the folder. In this example, we search for ".ini" files in the qmgrs folder that appears under the WebSphere MQ Files filter created in "Creating your own IFS filters" on page 177.
2. Select **Search**. The Search window (Figure 7-48) opens.
3. Click the **Remote Search** tab. Enter your search pattern in the File name patterns field. You can also search for a certain string inside the file in the Search string field. Select the **Search subfolders** check box. Click **OK**.

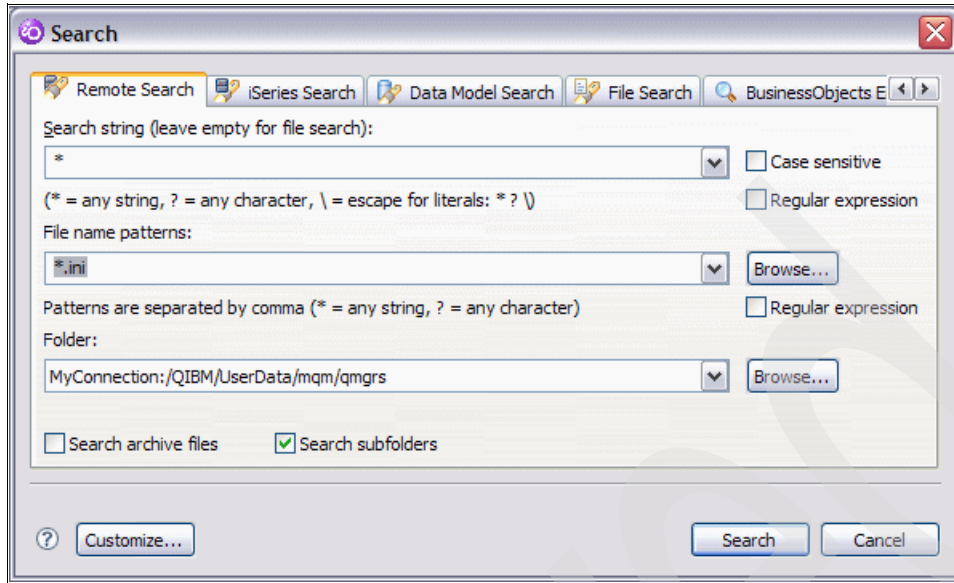


Figure 7-48 Entering a search pattern in the Search window

The result of the search appears in the Remote Search view (Figure 7-49) as part of a tabbed notebook of views.

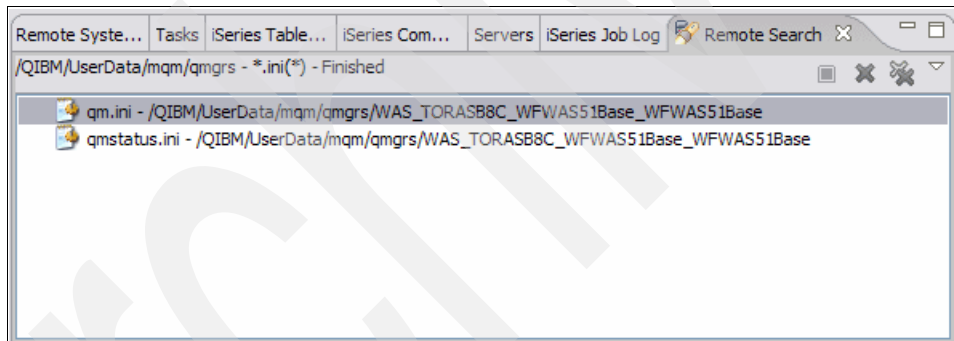


Figure 7-49 Search results displayed in the Remote Search view

Working with file filters

You can change, rename, copy, move, and delete IFS file filters. You can also move them up or down within the list under the subsystem. Simply right-click the job filter and then select the appropriate action from the menu that appears.

Launching command shells

From the IFS Files subsystem, you can launch a command shell to the i5/OS Qshell environment. To launch a command shell, follow these steps:

1. In the Remote Systems view, right-click **IFS Files**.
2. Select **Launch Shell**.
3. The Remote Shell view opens in a tabbed notebook of views. Enter your commands in the Command field and press Enter. The output of the command displays in the output area above the command entry field.

For more information about command shells, refer to 7.3.6, “Qshells subsystem” on page 181.

7.3.6 Qshells subsystem

From the Remote System Explorer, you can use the Remote Shell view to run and interact with commands and command shells on various remote servers.

You can use the view to:

- ▶ Run commands in a command shell.
- ▶ Display and interpret the output of a program.
- ▶ Enter input to a program.
- ▶ Run multiple commands in a command set (created under System i commands) as a single action.
- ▶ Use content assist to complete the code.

The Qshells subsystem lists the active running Qshells for the connection. You can also use this subsystem to start a Qshell.

To launch a shell from the Qshells subsystem:

1. Right-click the Qshells subsystem in the Remote System view.
2. Select **Launch Shell** from the context menu Figure 7-50.

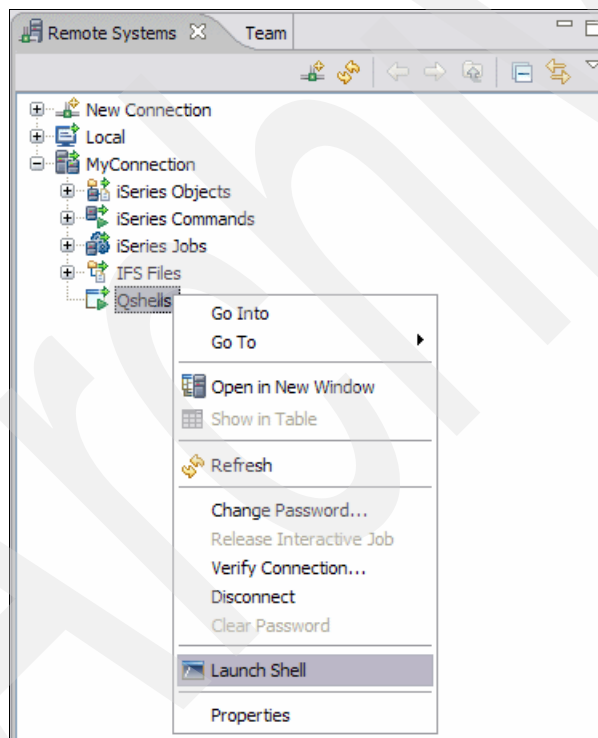


Figure 7-50 Launch Shell from the Qshells subsystem

3. The Remote Shell view (Figure 7-51) opens in a tabbed notebook of views. Enter your commands in the Command field and press Enter. The output of the command displays in the output area above the command entry field.

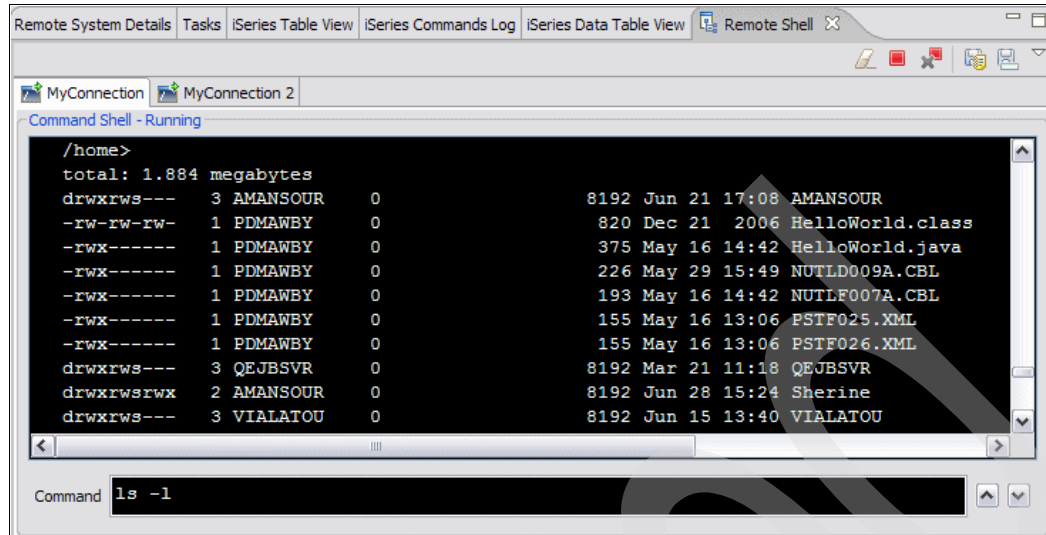


Figure 7-51 Remote shell view

You can launch additional shells in the Remote Shell view. To launch additional command shells, click the **Menu** button on the Remote Shell view toolbar. Then select **Launch Shell** → **your connection** → **Qshells** or right-click **Qshells** and select **Launch Shell** from the context menu. The new command shell appears as an *additional tab* near the top of the Remote Shell view.

To terminate the selected command shell, click the **Terminate the selected shell** button in the Remote Commands view toolbar.

To terminate and remove the command shell from the view, click the **Terminate the selected shell and remove it from the Remote Shell view** button in the Remote Commands view toolbar.

The launched Qshells appear also in the Remote Systems view; see Figure 7-52. You can expand the Qshells subsystem to show them. You can also right-click a shell in the list and perform the following functions:

- ▶ Show in Remote Shell View
- ▶ Cancel Shell
- ▶ Remove Shell
- ▶ Export Shell Output
- ▶ Export Shell History

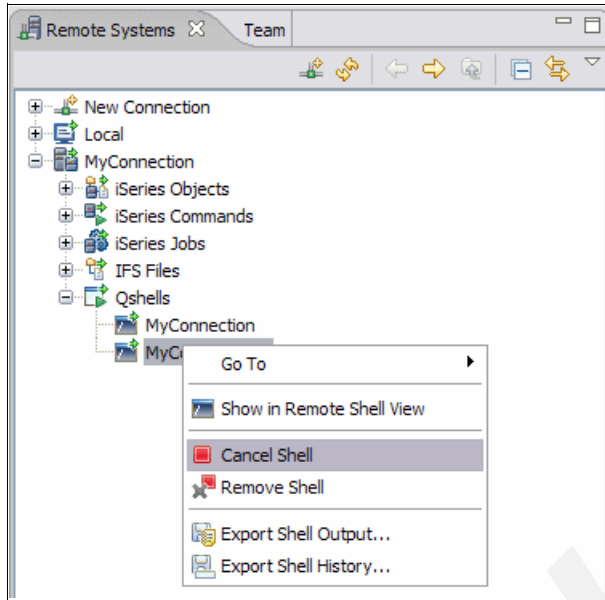


Figure 7-52 Qshells subsystem functions

You can also launch shells from IFS Files subsystem. For more details, refer to “Launching command shells” on page 180.

7.4 Managing your System i objects

This section discusses the use of filters and the different types of objects that you can create in the System i Objects subsystem in the RSE.

7.4.1 Types of filters in the System i Objects subsystem

The use of filters allows you to easily view and manage subsets of your System i objects. After you create a filter, you can save it under a unique name and reuse it. Four types of filters are available in the System i Objects subsystem, as explained in the following sections.

Library filters

Library filters list a set of libraries from your System i host in the Remote Systems view.

To create a library filter, follow these steps:

1. In the Remote Systems view, expand your connection and then expand **System i Objects**.
2. Expand **Work with libraries**. You can also right-click **System i Objects** and select **New** → **Library filter**. Expanding Work with libraries corresponds with using the WRKLIBPDM command.
3. The New - Library Filter window opens (Figure 7-53). Select a generic or specific library name from the Library drop-down list. Or, enter RSE*, for example, to list all libraries starting with “RSE”. You can also browse to locate libraries by clicking **Browse**.
4. Click **Next**.

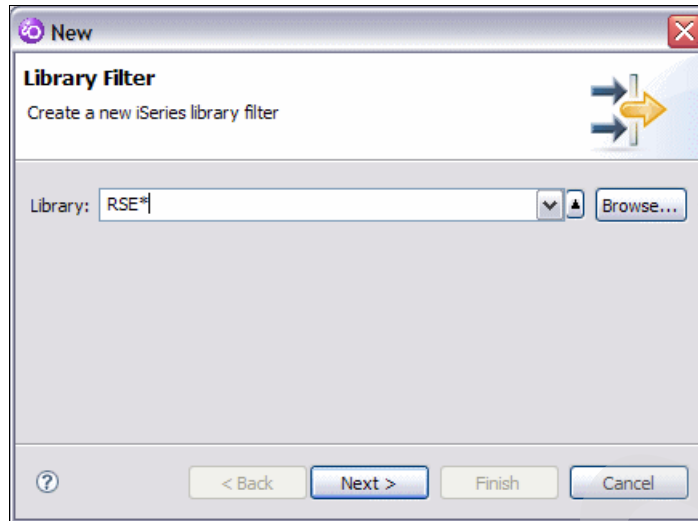


Figure 7-53 New Library Filter window

5. On the next window, enter a name for your filter in the Filter name field. Select the profile you want to use. Use an individually created profile to keep your work private or select the team profile to share with others.
6. Click **Finish**.

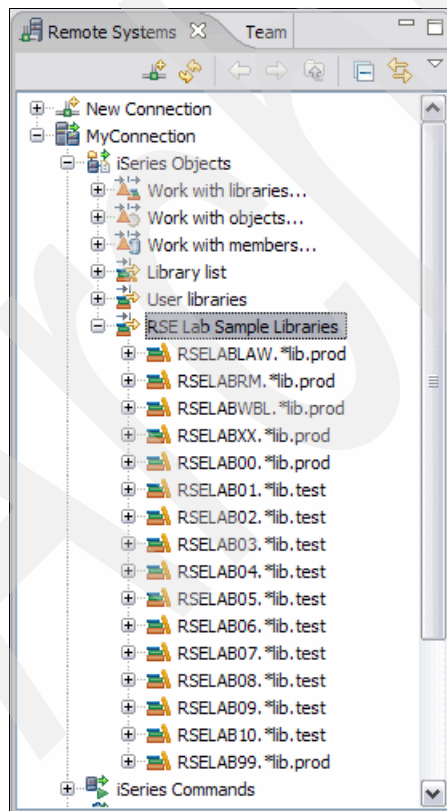


Figure 7-54 Expanding the newly created library filter

Your new library filter displays as a child node to the System i Objects subsystem. Expanding the newly created filter displays the libraries specified by this filter. Further expanding any library under the filter displays the objects in that library, as in Figure 7-54 on page 184.

Library list filters

Library list filters display the *LIBL library list from your System i host in the Remote Systems view. When expanded, it supplies the same actions as a library filter, with additional actions for manipulating the library list such as moving a library up or down the library list as we describe in more detail in 7.5, “Working with your library list” on page 204. If you are using your default profile, the wizard tells you that you already have a library list. Therefore, this action only applies if you are using a team profile or an additional personal profile.

To create a library list filter, follow these steps:

1. In the Remote Systems view, expand your connection and then expand **System i Objects**.
2. Right-click **System i Objects** and select **New → Library List Filter**.
3. The New - Library List Filter window opens. Enter a name for the filter in the Filter name field and switch the owner profile if necessary. Click **Finish**.

Your new library list filter displays as a child node to the System i Objects subsystem.

Temporarily subsetting all objects in a library by their type

When you expand a library node within a library filter or library list filter as shown in Figure 7-54 on page 184, you see all objects in that library. Depending on the number of objects, this can be quite a long list and you may have to page down several times to find a specific object.

One way of improving the time to search for an object is to use the *Expand to* function described here:

1. Right-click any library name in a library filter or library list filter.
2. Select **Expand to** in the context menu.
3. The resulting submenu, shown in Figure 7-55 on page 186, allows you to select one of the following options:
 - Source Files
 - Data Files
 - Device Files
 - Message Files
 - Save Files
 - Programs and Service Programs
 - Modules
 - Binding Directories
 - Data Queues
 - Data Areas
 - All

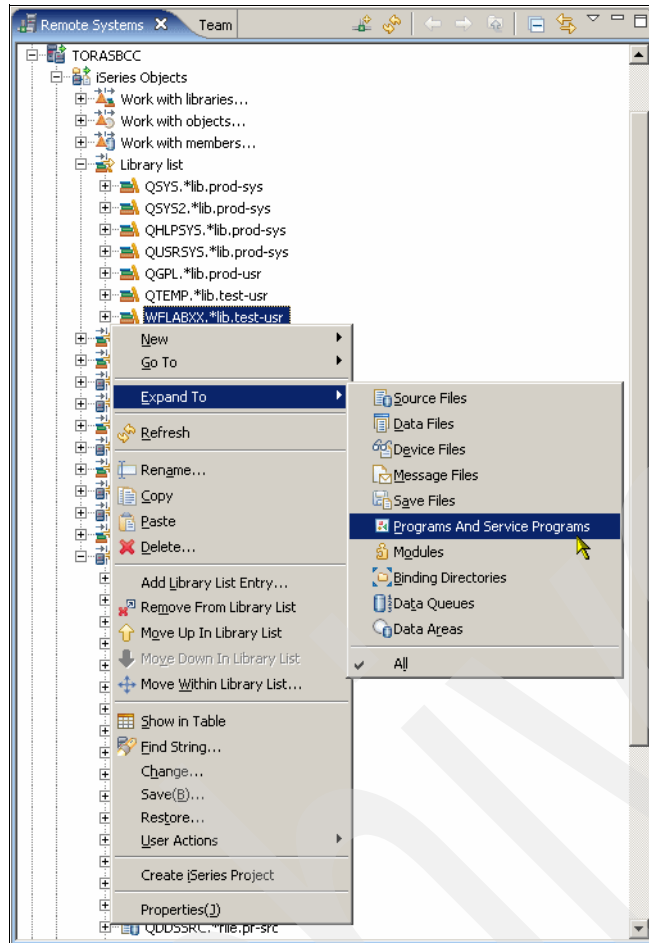


Figure 7-55 Expand a library to specific object types

Note, the selection list shown in Figure 7-55 is neither really a list of i5/OS object types nor a complete list of these. It rather shows those types of items which are frequently used by programmers. For example, the file object type *FILE is further broken down by its attribute (source file, data file,... and so on), whereas *Programs and Service Programs* contains two object types. Many other i5/OS object types, such as configuration descriptions, cannot be selected. However, you can use Object filters, as described in “Object filters” on page 186, for more granular selections, or you can use the System i Table view as described in “Subsetting objects and members in the Table view” on page 214.

Subsetting objects in a library by their names

Library list filters do not allow you to select only a subset of the objects in a library based on their names. However, you can use Object filters, as described next, for more granular selections or you can use the System i Table view as described in “Subsetting objects and members in the Table view” on page 214.

Object filters

Object filters list a set of i5/OS objects (that is, objects in QSYS or in one or more libraries in QSYS) in the Remote Systems view.

Note: Object filters contain objects in one *or more libraries*. You cannot see, either in the Remote Systems view or in the System i Table view, which library a particular object belongs to. To find out, either click an object and the Source field in the Properties view, or right-click the object and select **Properties** from the context menu and then select **Object Info**.

To create an object filter, follow these steps:

1. In the Remote Systems view, expand your connection and then expand **System i Objects**.
2. Expand **Work with objects**. You can also right-click **System i Objects** and select **New** → **Object Filter**. Expanding Work with objects corresponds with the Work with Objects Using PDM (WRKOBJPDM) command. The New - Object Filter window opens. For this demonstration, we create an object filter that displays all program objects in the library RSELABxx.
3. In the New - Object Filter (Figure 7-56), complete these tasks:
 - a. Browse, specify, or select from the Library drop-down list the library that contains your objects.
 - b. Enter a generic or specific object in the Object field, or click **Browse** to select. The default is * (asterisk), and you can use up to ten characters.
 - c. To define *object type*, browse, specify, or select from the drop-down list, or click **More Types>>** to select from all known object types.
 - d. For Object attribute, browse, specify, or select from the drop-down list. Note that you can only enter a *single object type* and *attribute pair per filter*, until you change a filter to add more filter strings.
 - e. Click **Next**.

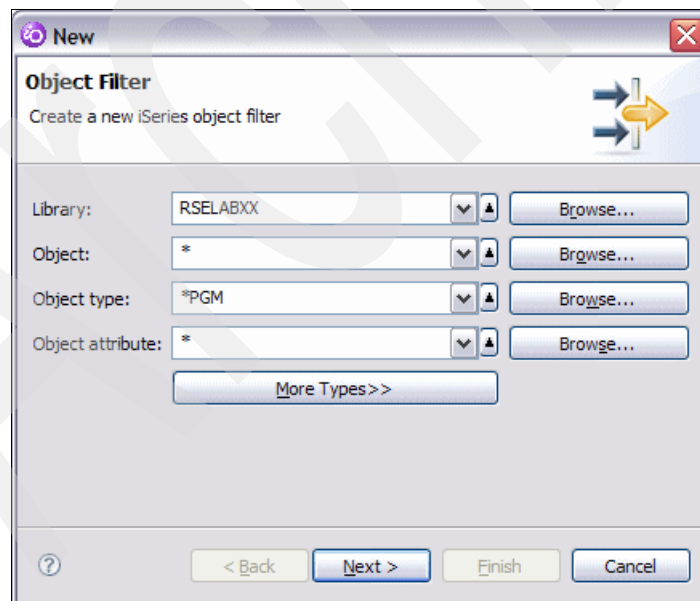


Figure 7-56 New Object Filter window

4. On the next window, enter a name for your filter in the Filter name field. Select the profile you want to use. Use an individually created profile to keep your work private or select the Team profile to share with others. Click **Finish**.

Your new object filter displays as a child node to the System i Objects subsystem. Expanding the newly created filter displays the objects specified by the filter. In our example, it displays all program objects in library RSELABxx. The object type and attribute also appear next to the object name in the list. See Figure 7-57.

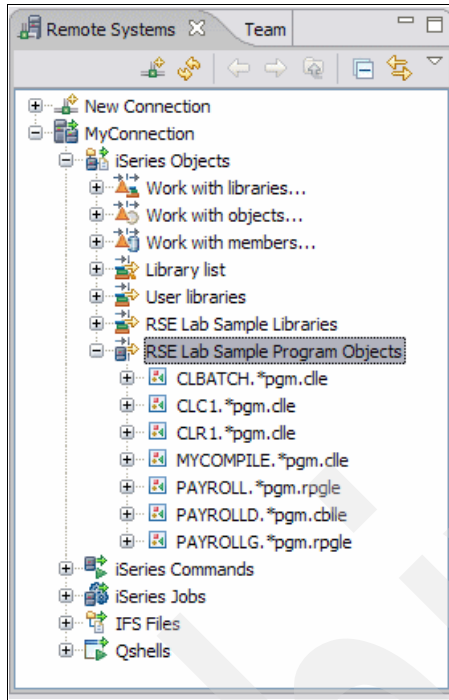


Figure 7-57 Expanding the newly created object filter

Member filters

Member filters list a set of source and data members from your System i host in the Remote Systems view.

To create a member filter, follow these steps:

1. In the Remote Systems view, expand your connection and then expand **System i Objects**.
2. Expand **Work with members**. You can also right-click **System i Objects** and select **New** → **Member filter**. Expanding Work with members corresponds with the WRKMBRPDM command. The New - Member Filter window opens. For this example, we create a member filter that displays all members in the QCLSRC source file in library RSELABxx.
3. In the New - Member Filter window (Figure 7-58), complete these tasks:
 - a. Browse, specify, or select from the Library drop-down list the library that contains your source members.
 - b. Browse, specify, or select from the File drop-down list the file within the library that contains the appropriate source members.
 - c. Enter a generic or specific member name in the Member field. The default is * (asterisk), and you can use up to 10 characters.
 - d. To select a specific member type, enter the type in the Member type field, or click **More Types** to select from all known object types. This step is optional.

- e. Select the **Source members** and **Data members** check boxes to indicate whether you want to list source members, data members, or both. You *must* select at least one. These options are only available if you specify a generic file name in the File field.
- f. Click **Next**.

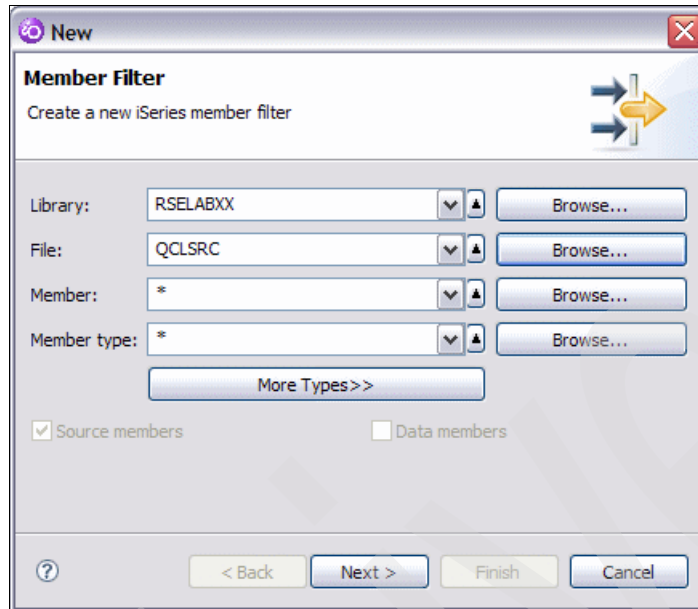


Figure 7-58 New Member Filter window

4. In the next window (Figure 7-59), enter a name for your filter in the Filter name field (this name is what appears in the Remote Systems view). Select the profile you want to use. Use an individually created profile to keep your work private or select the Team profile to share with others.



Figure 7-59 Name of new member filter

5. Click **Finish**.

- Expand your filter in the Remote Systems view to see the results, or right-click the filter and select **Show in Table**.

Filters are actually made up of one *or more* filter strings. When you first create the filter, the wizard prompts you for the filtering criteria for only one filter string. After the filter is created you can right-click it and select **Change** from the pop-up menu to modify the filter string or add additional filter strings. This flexibility allows you to create filters that capture only the objects or members you are interested in using either naming patterns (if possible) or explicitly listing each one.

Your new member filter displays as a child node to the System i Objects subsystem. Expanding the newly created filter displays all the members specified by the filter. In this case, it displays all the source members in the QCLSRC source file located in library RSELABxx. The member type appears next to the member name in the list. See Figure 7-60.

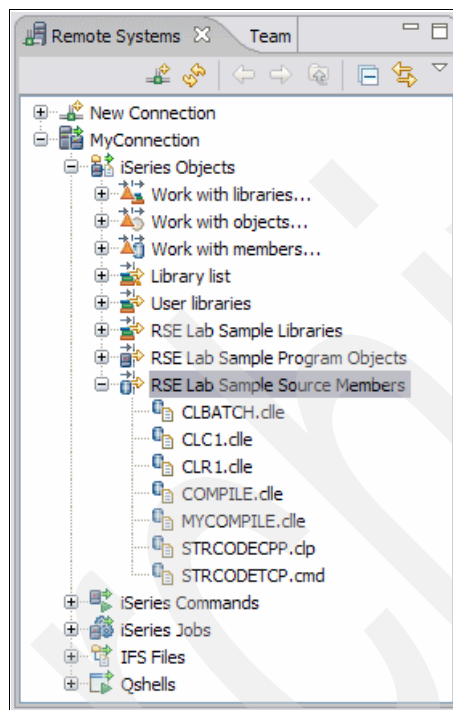


Figure 7-60 Expanding the newly created member filter

7.4.2 Working with System i Objects filters

You can change, rename, copy, move, or delete System i Objects filters. You can also move them up or down within the list under the subsystem. To perform these tasks, right-click the object filter and then select the appropriate action from the pop-up menu.

You *change* a filter when you want to alter the System i host items it displays in the Remote Systems view or System i Table view. When you change a filter, you can also add extra filter strings to the filter.

Filter pools

If you have been using the workbench for some time, it is quite possible that your workspace may contain too many filters to navigate easily. In this case, you can group filters into *filter pools*. You can group filters by project, release, connection, task, etc. For example, a filter

pool may contain a combination of two library filters, one object filter, and six member filters. Or it might contain a series of object filters and nothing else.

Also filter pools allow you to show filters under only certain connections. For example, you can create a filter pool called “Human Resources application” and add related filters to this filter pool. Then you can choose to only show this filter pool under a specific System i connection.

To use filter pools, follow these steps:

1. Click the **Menu** (▾) button on the toolbar of the Remote Systems view, and select **Show Filter Pools**.
2. Expand **System i Objects**. You should now see your filters listed under *Profile name* Filter Pool. See Figure 7-61

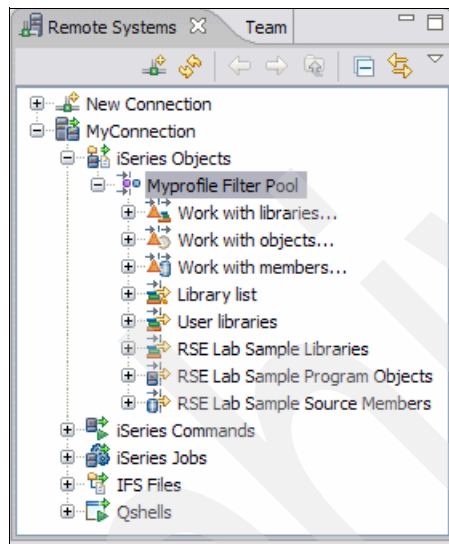


Figure 7-61 Showing filter pools

3. Right-click **System i Objects** and select **New** → **Filter Pool**. From here, you can create a new filter pool under the System i Objects subsystem.
4. Right-click **Profile name Filter Pool** and select **New**. From here, you can create a new Library List, Library, Object, or Member filters within a specific filter pool.
5. To switch back to regular filters, click the **Menu** (▾) button on the toolbar for the Remote Systems view, and deselect **Show Filter Pools**.

The ability to use filter pools is available for all four subsystems. You can copy filters or move them from one filter pool to another.

You can also copy or move filter pools from one profile to another within the subsystem they reside in.

7.4.3 Creating i5/OS objects in the System i Objects subsystem

This section describes the built-in wizards in the RSE that enable the creation of the following i5/OS objects (that is, objects in QSYS or in a library in QSYS):

- ▶ Data areas
- ▶ Data queues
- ▶ Message files

- ▶ Source physical files
- ▶ Members (*not* objects; for the purposes of this discussion, the term “object” will suffice)

The following sections explain how to create each object type.

Creating data areas

In the RSE, you can create data areas in libraries on your System i host. This function corresponds to the Create Data Area (CRTDTAARA) CL command.

To create a data area, follow these steps:

1. In the Remote Systems view, expand your connection. Then expand **System i Objects** → **Library list**.
2. Right-click the library in which you want to create the data area. You can select any library within your subsystem that is part of a user-created filter as well. We chose a library from the predefined library list filter. Select **New** → **Data Area**. The New - System i Data Area window (Figure 7-62) opens.
3. In the New - System i Data Area window, complete the following tasks:
 - a. Enter the name of the data area you want to create in the Data area field.
 - b. Select the type of data area you want to create using the radio buttons in the Type section of the window.
 - c. Optionally, add a description for the data area in the Text field.

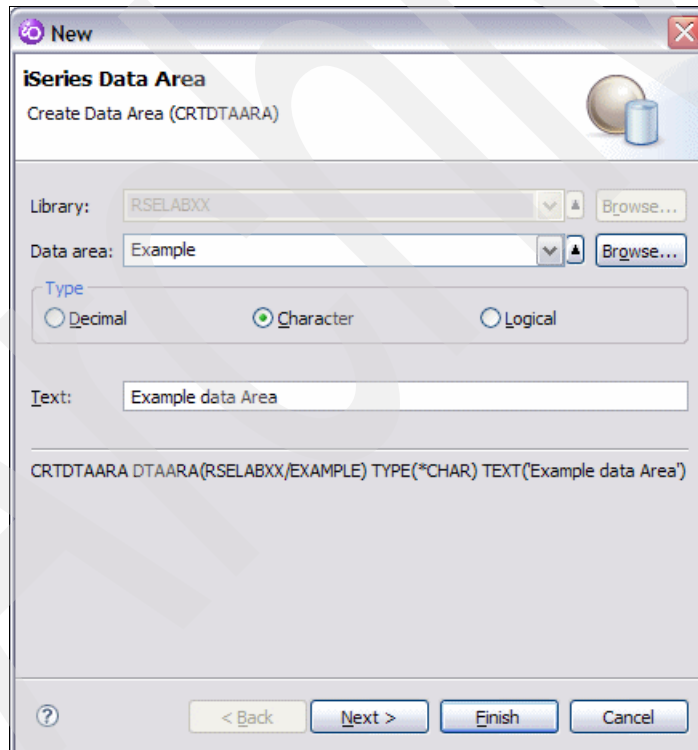


Figure 7-62 Creating a data area

4. Optionally click **Next** to specify further options for your data area, such as authority, length information, or the initial value.
5. Click **Finish**. The new data area displays as a child node to the library in which it was created.

Basic information about the data area is displayed in the Properties view, which is generally located under the Remote Systems view. Further information is available by right-clicking the data area and selecting **Properties(B)**.

To *change* the value contained in the data area, right-click the data area and select **Change**. Enter the new value in the New value text box and specify any substring requirements. Click **OK**.

Creating data queues

In the RSE, you can create data queues in libraries on your System i host. This function corresponds to the Create Data Queue (CRTDTAQ) CL command.

To create a data queue, follow these steps:

1. In the Remote Systems view, expand your connection. Then expand **System i Objects** → **Library list**.
2. Right-click the library in which you want to create the data queue. You can select any library within your subsystem that is also part of a user-created filter. We chose to use a library from the predefined library list filter. Select **New** → **Data Queue**. The New - System i Data Queue window (Figure 7-63) opens.
3. In the New - System i Data Queue window, complete the following tasks:
 - a. Enter the name of the data queue you want to create in the Data queue field.
 - b. Alter the size of the data queue entry length in the Maximum entry length field if necessary.
 - c. Optionally add a description for the data queue in the Text field.
4. Optionally click **Next** to specify further options for your data queue, such as authority, force to auxiliary storage, sequence, queue size, and auto-reclaim.
5. Click **Finish**. The new data queue displays as a child node to the library in which it was created.

Basic information about the data queue is displayed in the Properties view that is generally located under the Remote Systems view. Further information is available by right-clicking the data queue and selecting **Properties(B)**.

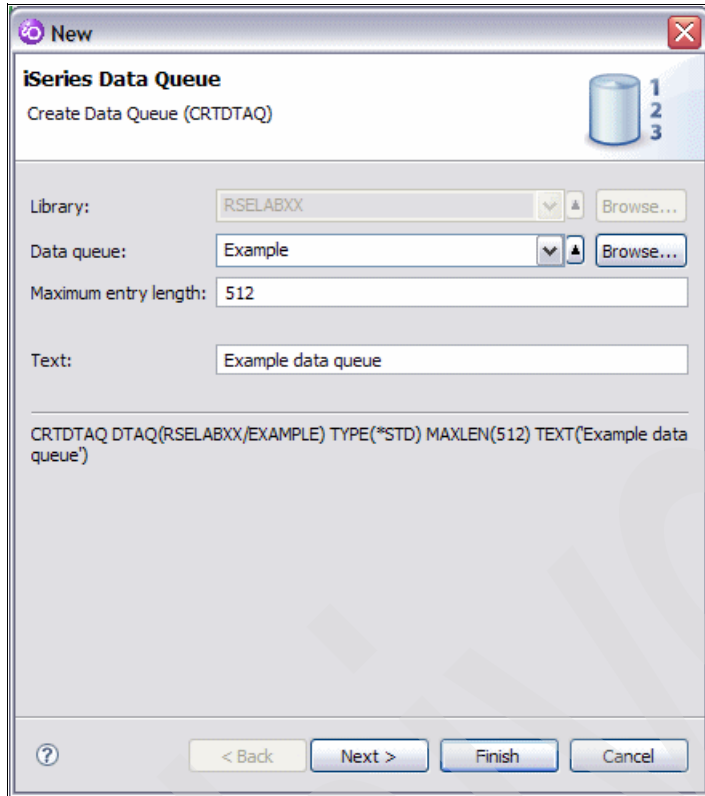


Figure 7-63 Creating a data queue

Creating message files

In the Remote System Explorer, you can create message files in libraries on your System i host. This function corresponds to the Create Message Files (CRTMSGF) CL command.

To create a message file, follow these steps:

1. In the Remote Systems view, expand your connection. Then expand **System i Objects** → **Library list**.
2. Right-click the library in which you want to create the message file. You can select any library within your subsystem that is also part of a user-created filter. We chose to use a library from the predefined library list filter. Select **New** → **Message File**. The New - System i Message File window (Figure 7-64) opens.
3. In the New - System i Message File window, complete the following tasks:
 - a. Enter the name of the message file you want to create in the Message file field.
 - b. Optionally add a description for the message file in the Text field.

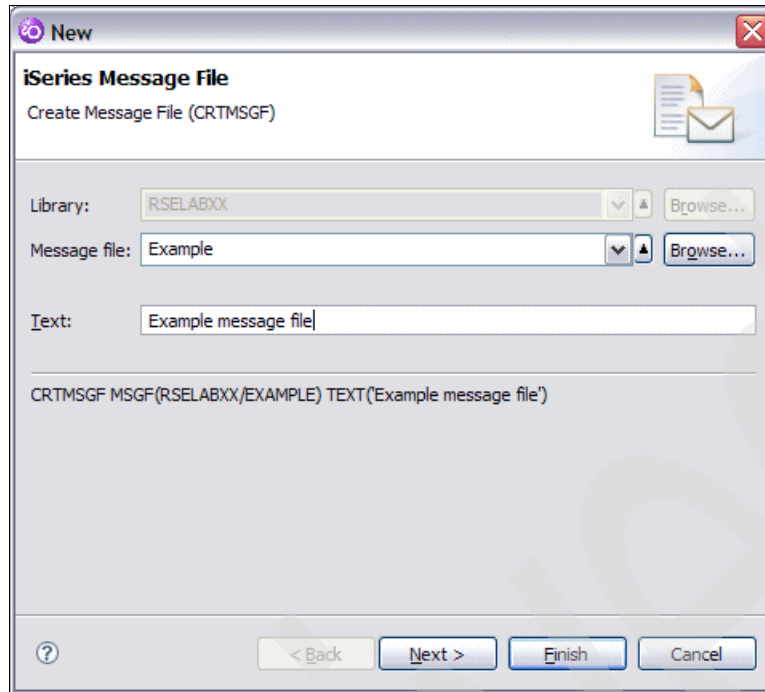


Figure 7-64 Creating a message file

4. Optionally click **Next** to specify further options for your message file, such as authority, coded character set ID, or file size.
5. Click **Finish**. The new message file displays as a child node to the library in which it was created.

Working with message descriptions

To *add* a message description to a message file, complete the following steps:

1. Right-click the message file and select **New** → **Message Description**. The Add Message Description window opens.
2. Enter details for the new message description.
3. Select the **Advanced(L)** check box to specify any advanced options.
4. Click the **OK** button to create the message description.

To display message descriptions in a message file, you can do either of the following tasks:

- ▶ Double-click the message file.
- ▶ Expand the message file.

The messages are displayed as child nodes to the message file in the Remote Systems view.

To *change* or *delete* a message description, right-click the message description and select the appropriate action from the pop-up menu.

Creating source physical files

In the RSE, you can create source physical files in libraries on your System i host. This function corresponds to the Create Source Physical Files (CRTSRC PF) CL command.

To create a source physical file, follow these steps:

1. In the Remote Systems view, expand your connection. Then expand **System i Objects** → **Library list**.
2. Right-click the library in which you want to create the source physical file. You can select any library within your subsystem that is part of a user-created filter as well. We chose to use a library from the predefined library list filter. Select **New** → **Source Physical File**. The New - System i Source Physical File window (Figure 7-65) opens.
3. In the New - System i Source Physical File window, complete the following tasks:
 - a. In the File field, enter the name of the source physical file you want to create.
 - b. In the Record length field, enter the record length for the source physical file.
 - c. Optionally, in the Text field, add a description for the source physical file.

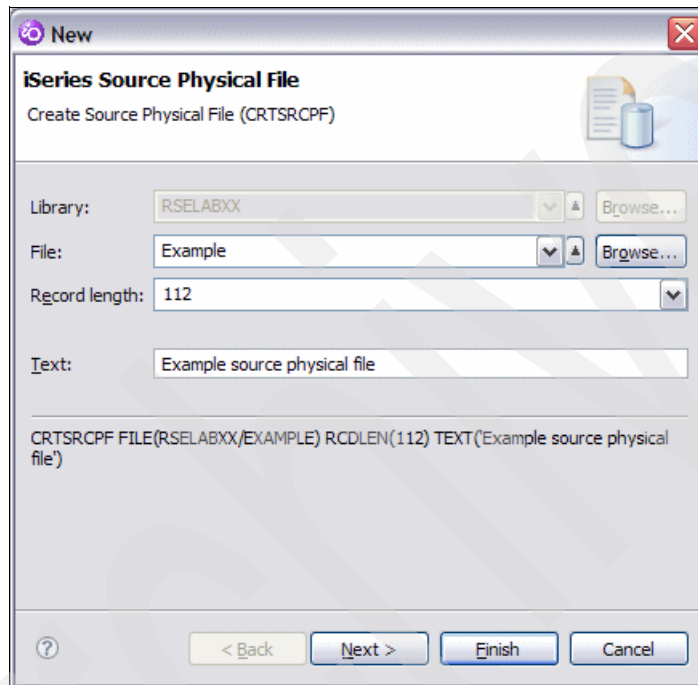


Figure 7-65 Creating a source physical file

4. Optionally click **Next** to specify further options for your source physical file, such as authority, coded character set id, or IGC data.
5. Click **Finish**. The new source physical file displays as child node to the library in which it was created.

Note: You can also create source physical files from the System i Commands subsystem. To do this, follow these steps:

1. In the Remote Systems view, expand your connection. Then expand **System i Commands** → **Create source file**.
2. Specify or select parameters for the source file. Select the **Advanced** check box for extra parameters.
3. Click **OK**.

Creating members

In the RSE, you can create members in source physical files on your System i host. This function corresponds to the ADDPFM CL command.

To create a member, follow these steps:

1. From the Remote Systems view, expand your connection. Then expand **System i Objects** → **Library list**.
2. Select and expand the library and source physical file where you want to create a member. You can select any library your subsystem that is part of a user-created filter as well. We chose to use a library from the predefined library list filter.
3. Right-click the source physical file and select **New** → **Member**. The New - System i Source Member window (Figure 7-66) opens.
4. In the New - System i Source Member window, complete the following tasks:
 - a. In the Member field, enter the name of the member you want to create.
 - b. In the Member type field, enter the member type, or select from the available items in the list.
 - c. Optionally, in the Text field, add a description for the member.
 - d. Click **Finish**.

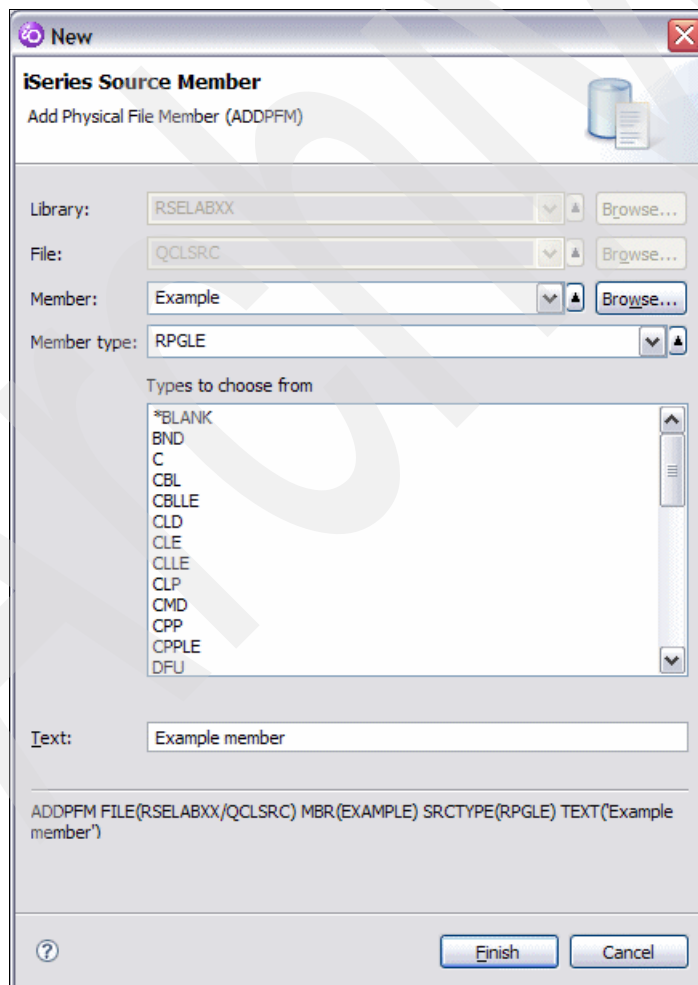


Figure 7-66 Creating a member

The new member displays as a child node to the selected source physical file. The new member (empty) also opens in the default editor.

Note: You can also create members from the System i Commands subsystem. To do this:

1. In the Remote Systems view, expand your connection. Then expand **System i Commands** → **Create source member**.
2. Specify or select parameters for the source member. Select the **Advanced** check box for extra parameters.
3. Click **OK**.

7.4.4 Searching operations

You can search for text strings on your System i server from the Remote System Explorer perspective.

Search for text strings on the System i server

This action corresponds to the *Find String Using PDM* (FNDSTRPDM) command in PDM, which would search for strings in one or more members of a source file. In addition to that, the search option in the Remote System Explorer can search multiple libraries and filters as well.

There are two ways to search:

- ▶ From a selection in RSE
- ▶ From the central search dialog

When you search from a selection, you navigate through your libraries and files until you find the item you want to search.

When you search from the central search dialog, you do not begin with any selection or containing item. Your search is as broad as you want it to be, and you can narrow the search by specifying certain libraries, files, or member names to search.

Tip: Before you do a Search operation in the source editor, save the file you are searching. The search function works from the most recently saved version of the file rather than from the contents that you see in the editor area. You do not need to save your file before you do a Find/Replace operation.

To search for strings from a selection

1. In the Remote Systems view, expand your System i connection until you reach the item that you want to search. Or, navigate through items displayed in the System i Table view.
2. Select the containing item by clicking the item.
3. Right-click and select **Find string**.
4. In the **Find String** window, specify the string that you want to search in the Search string field. Select the **Case sensitive** check box if applicable.

The Scope area displays the targets that you have selected, as shown in Figure 7-67.

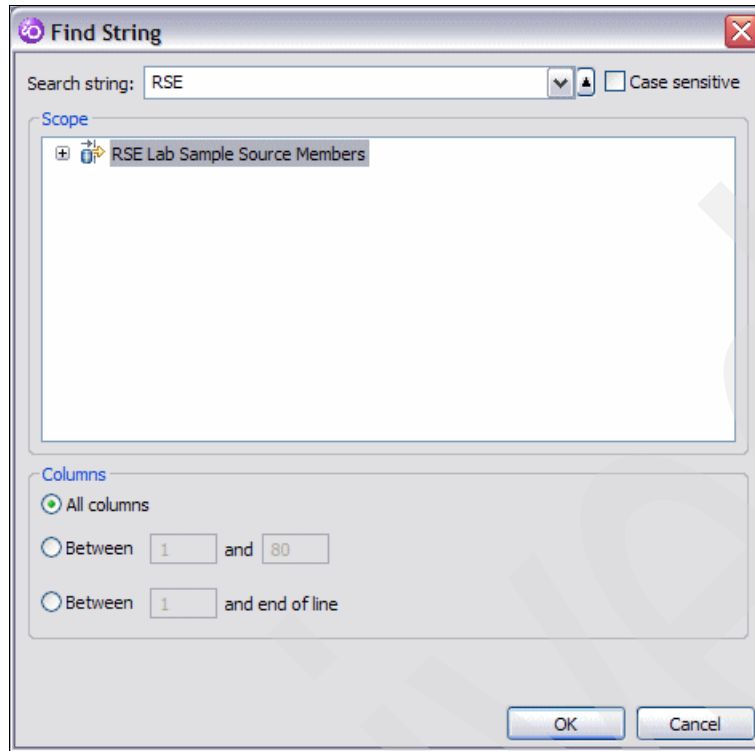


Figure 7-67 Find String window showing the Search string and the Scope

5. If applicable, specify columns to search with the Columns radio buttons. By default, **All columns** are searched.
6. Click **OK**. The matching strings display in the Remote Search view, as shown in Figure 7-68.

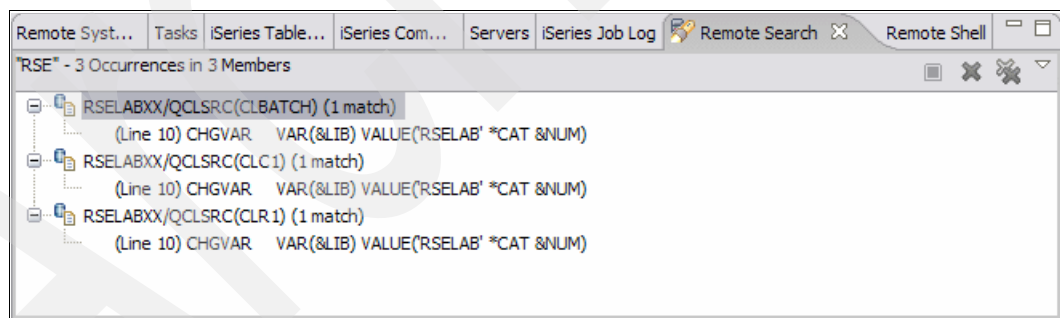


Figure 7-68 Remote Search window showing the search results

To search for strings from the entire System i server

1. Click **Search** → **System i** from the Workbench menu.
2. In the System i Search tab of the Search window, specify the string that you want to search in the Search string field.
3. Select the **Case sensitive** check box if applicable.
4. Use the Target area to specify the target that you want to search. The names can be generic.
5. If the file name is generic, two check boxes beneath this area are enabled: **Source members** and **Data members**. Source members is checked by default. Check Data

members if you also want to search data members. Searching data members takes a longer amount of time compared to searching source members, which is why it is unchecked by default, as shown in Figure 7-69.

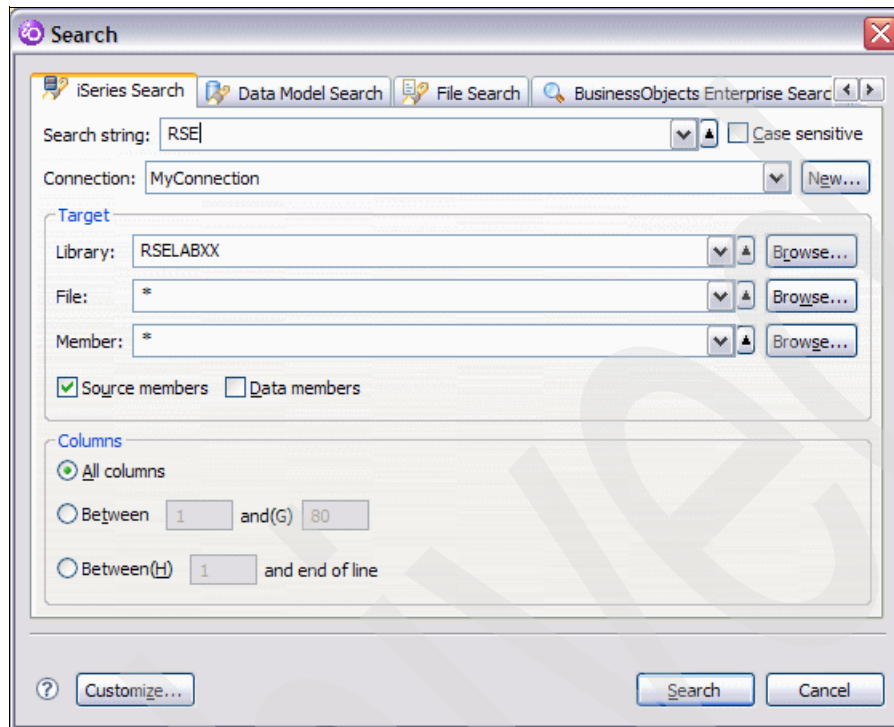


Figure 7-69 System i Search window

6. If applicable, specify columns to search with the Columns radio buttons. By default, **All columns** are searched.
7. Click **OK**. The matching strings display in the Remote Search view.

Tip: The string searched for can be a character or hexadecimal value.

7.4.5 Working with save files

Most i5/OS and OS/400 object types can be saved not only directly onto a tape or CD ROM but also into another object type called save file (object type *FILE with attribute SAVF). As a common practice, these save files can also be transferred to and stored on other platforms such as a Windows or Linux workstation. This format also allows you to make i5/OS and OS/400 objects available for download from Web sites.

This was mostly done by sending the SAVF to the workstation by means of the File Transfer Protocol (FTP) or mapping the System i storage as a network drive to your workstation and using drag and drop in a file explorer or a copy command to transfer the saved objects to a binary file.

RSE allows you to exchange i5/OS save files in a much easier way. The key capability of RSE to enable this, is the fact that you can copy and paste between different connections within your RSE workbench. That is, you may copy a save file from an i5/OS library into a directory on your local workstation (using the default “local” connection for example).

Download a save file from i5/OS to your PC

So to download a save file from your System i to your PC simply right-click an object with the extension *.file.savf and select **Copy** from the context menu (or select the object and press Ctrl+c). Then right-click the target directory in a local system connection and select **Paste** from the context menu (or select the object and press Ctrl+v).

Upload a save file from your PC to i5/OS

Conversely, you can upload the save file from your PC to an i5/OS library by copying it from a local system connection to a System i connection.

Restoring objects or a complete library directly from a PC to i5/OS

The biggest advantage appears when you want to restore the content of a save file residing on your workstation. With RSE, you can do so in a single step! Just right-click any local file with extension .savf or .sav and select **Restore to System i** from the context menu, as shown in Figure 7-70.

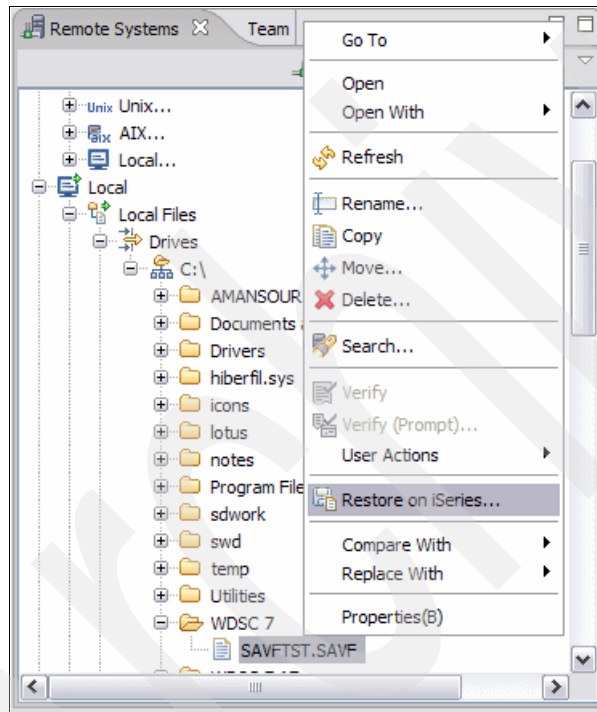


Figure 7-70 Restore i5/OS or OS/400 objects from a local save file

You see the Restore System i Save File dialog shown in Figure 7-71, which gives you the choices to:

- ▶ Restore objects into an existing library or restore a complete library, if the Save Library (SAVLIB) command was used to fill the save file.
- ▶ Specify the name of a library and the save file.
- ▶ Specify the name of the library that the objects should be restored to. The default is *SAVLIB, but you can enter any other library name, as shown by the mouse pointer in Figure 7-71.
- ▶ Select to see a prompt for the restore command (RST, RSTOBJ, or RSTLIB) in case you want to override any of the default parameters.

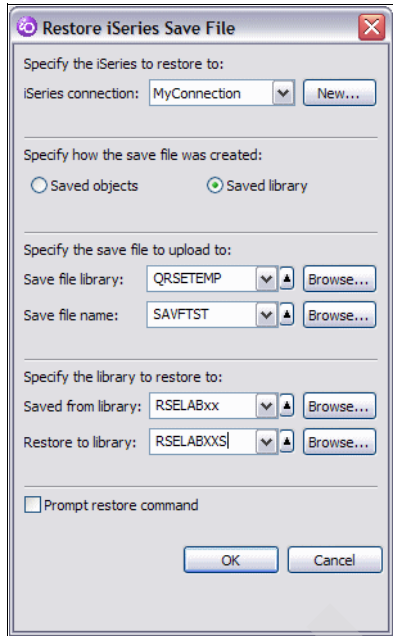


Figure 7-71 Restore System i Save File

During the upload and restore process you can see the progress in the bar at the bottom of the dialog, similar to Figure 7-72.

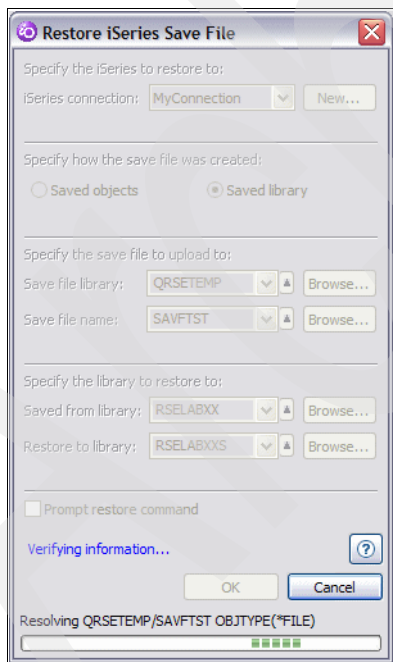


Figure 7-72 Uploading a Save File

If you selected to be prompted for the restore command, the prompt, similar to the one shown in Figure 7-73, appears *before* the save files were actually uploaded onto the System i. You may select the **Advanced(1)** check box to see more parameters or the **All Parameters(2)** to see all parameters.

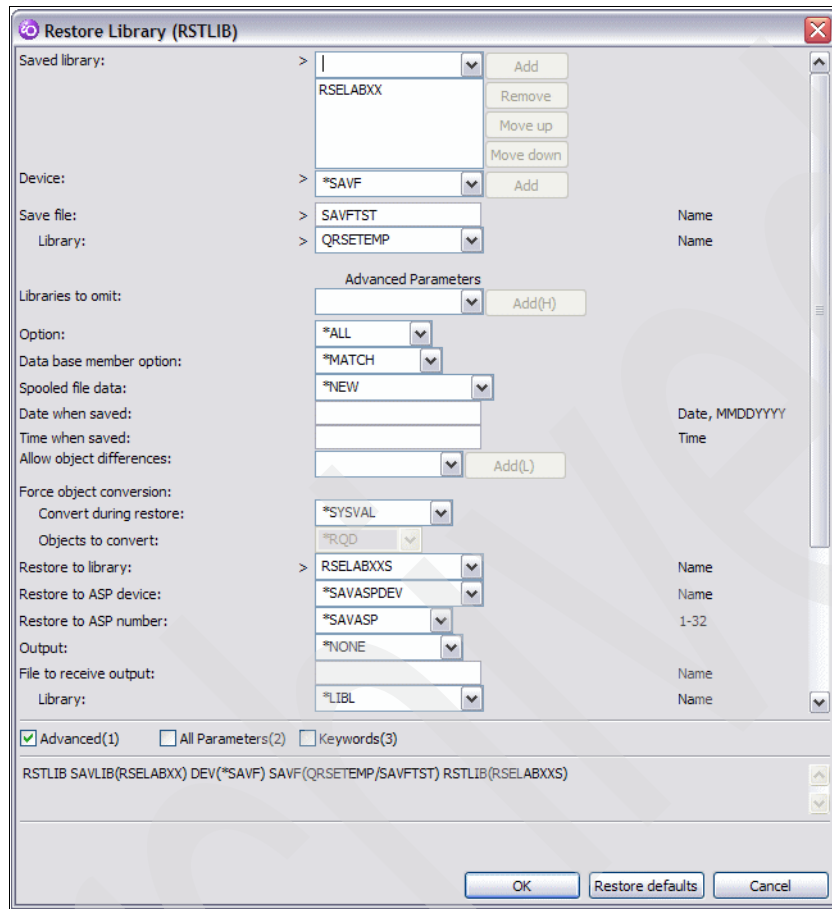


Figure 7-73 Restore Library (RSTLIB) prompt

After the restore process completes or ends abnormally, you can see the informational or error message from the corresponding job log in the System i Commands Log (by default in the lower pane of the RSE perspective), as shown in Figure 7-74.

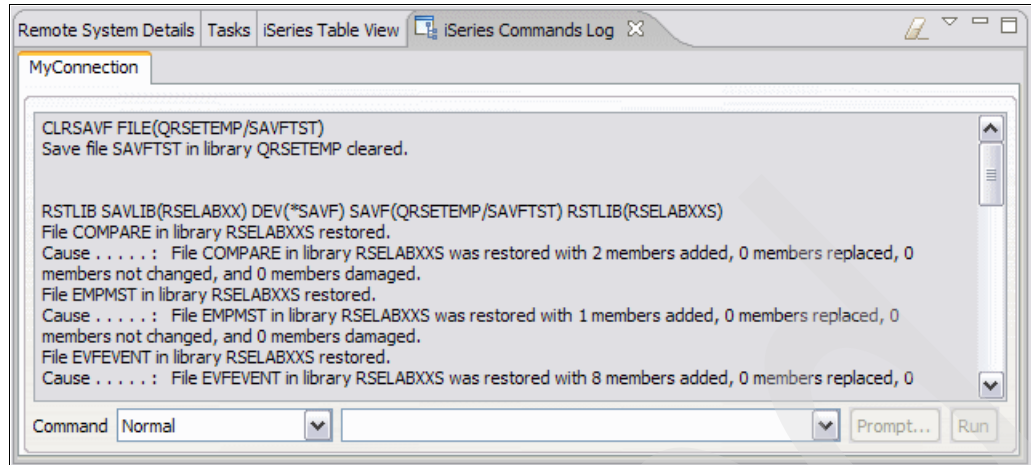


Figure 7-74 System i Commands Log after restoring a library

Working with zipped files

Also, as a common practice, you may compress a save file into a .zip file. The Remote Systems view of the RSE perspective allows you to expand a .zip file and see which files or directory are stored in there. However, you cannot directly restore a save file to the System i from a zip file without first extracting the content.

7.5 Working with your library list

This section explains the actions that you can perform on your library list in the RSE perspective.

Using the pop-up menu actions for the library filter or library list filter

Within the library list filter, you can perform several actions to modify the user portion of your library list. These actions correspond with the Edit Library List (EDTLIBL) i5/OS command.

Note, when we say *your* library list, we actually mean the *RSE communication server* job which accesses i5/OS on your behalf, while you work in the Remote Systems view or Remote Systems LPEX editor. The changes you make are only in effect until the RSE connection is disconnected.

- ▶ To add a library to the library list of your server job, right-click the Library List filter and select **Add Library** from the context menu, as shown in Figure 7-75.
- ▶ To change the current library of your server job, right-click the Library List filter and select **Change Current Library** from the context menu, as shown in Figure 7-75.

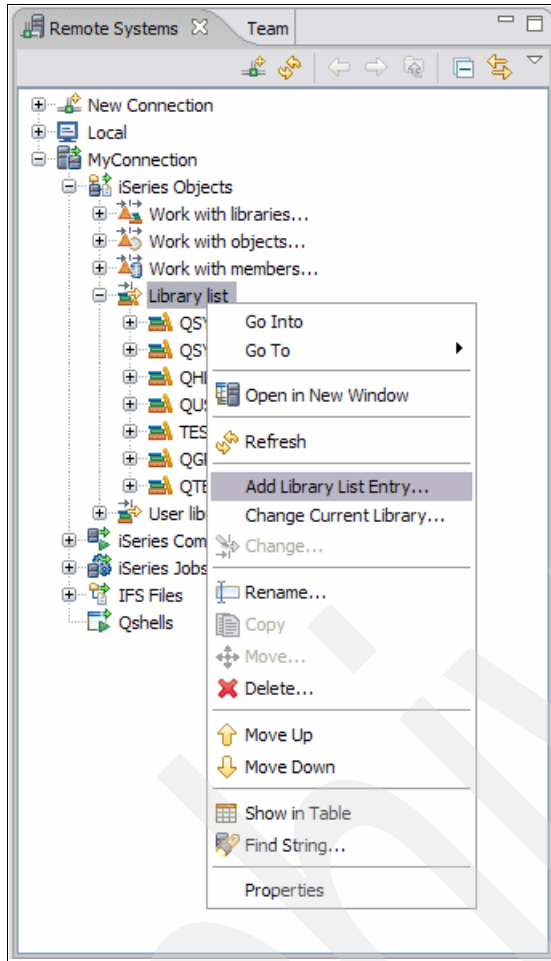


Figure 7-75 Library list pop-up menu functions

Important: When you perform the following actions with your library list, the changes only apply to the current session in the current communications server job.

Using the Initial Library List property page for the subsystem

A more permanent solution for modifying your library list is to use the Initial Library List property page for the System i Objects subsystem. Changes made here will always be added to the library list for the RSE connection (modifying the library list derived from the job description defined in your user profile).

The changes only take effect the next time the RSE connection is actually established. To enforce the change immediately, select **Disconnect** from the context menu shown in Figure 7-76 and then expand the tree under **System i Objects** again, to connect to i5/OS again and perform the actions defined in the properties shown in Figure 7-77 on page 207.

To set this option, perform the following actions:

1. Right-click on System i Objects (and select **Properties** from the pop-up menu Figure 7-76).

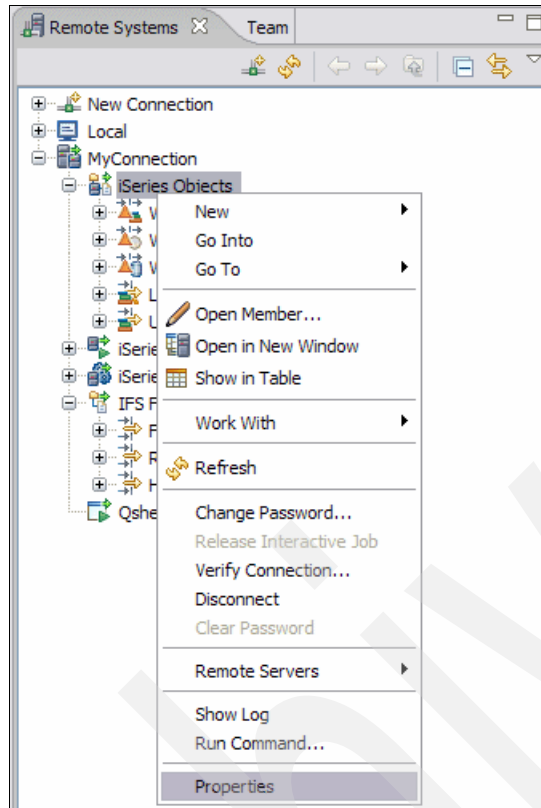


Figure 7-76 System i Objects pop-up menu

2. Select the Initial Library List page in the Properties dialog Figure 7-77. After you finish the changes, click **Apply**, then **OK**.

Note: Each RSE connection has its own library list. If you are working on multiple projects that require different library lists, you can create a connection with an individual library list setting for each project.

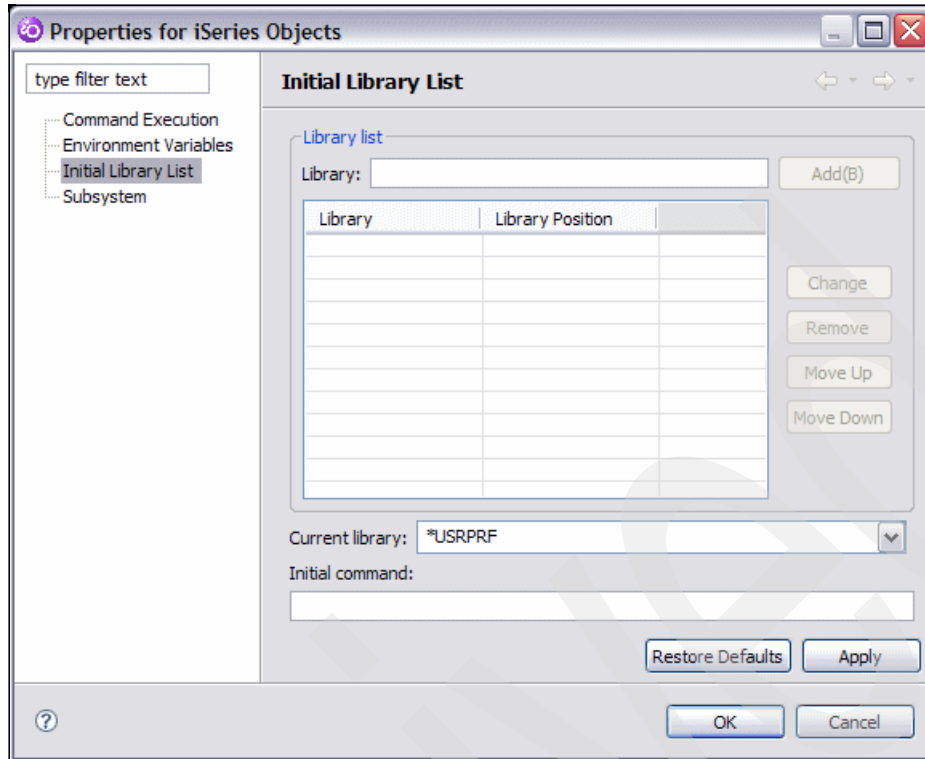


Figure 7-77 System i Objects pop-up menu

You can also specify a current library to be set, as well as an initial command to be executed, when RSE connects to the server.

7.5.1 Adding a library list entry

To add a library to the user portion of your library list, follow these steps:

1. In the Remote Systems view, expand **System i Objects**.
2. Right-click the **Library list** filter and select **Add Library List Entry**. The Add Library List Entry window (Figure 7-78) opens.

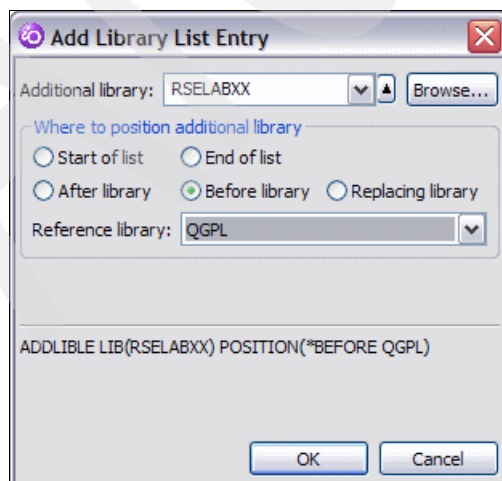


Figure 7-78 Adding a library to the library list

3. In the Add Library List Entry window, complete the following tasks:
 - a. In the Additional library field, enter a library name, or click **Browse** to navigate to a library.
 - b. Select one of the radio buttons that specify where to position the library relative to a reference library.
 - c. From the Reference library drop-down list, select a reference library, if applicable.
 - d. Click **OK**.

Tips:

- ▶ An alternative method for adding an entry to the user portion of the library list is to right-click a library in a predefined or user-created filter and then select **Add To Library List**.
- ▶ As mentioned above, you can only add a library to the user portion of your library list. If you need to change the system library list, you can enter the Change System Library List (CHGSYSLIBL) in the command line below the System i Table view to add a library or remove it from the system portion of the library list.

7.5.2 Changing the current library

To change the current library, follow these steps:

1. In the Remote Systems view, expand **System i Objects**.
2. Right-click the **Library list** filter and select **Change Current Library**. The Change Current Library window opens, as in Figure 7-79.
3. In the Current library field, enter a new library name, or click **Browse** to select a library. Then click **OK**.

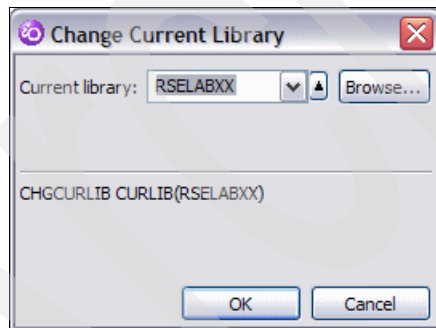


Figure 7-79 Changing the current library

7.5.3 Removing a library list entry

To remove a library from the user portion of your library list, follow these steps:

1. In the Remote Systems view, expand **System i Objects** → **Library list**.
2. Right-click the user library you want to remove and select **Remove From Library List** as shown in Figure 7-80.

Note: You can only remove a library that belongs to the user portion of your library list.

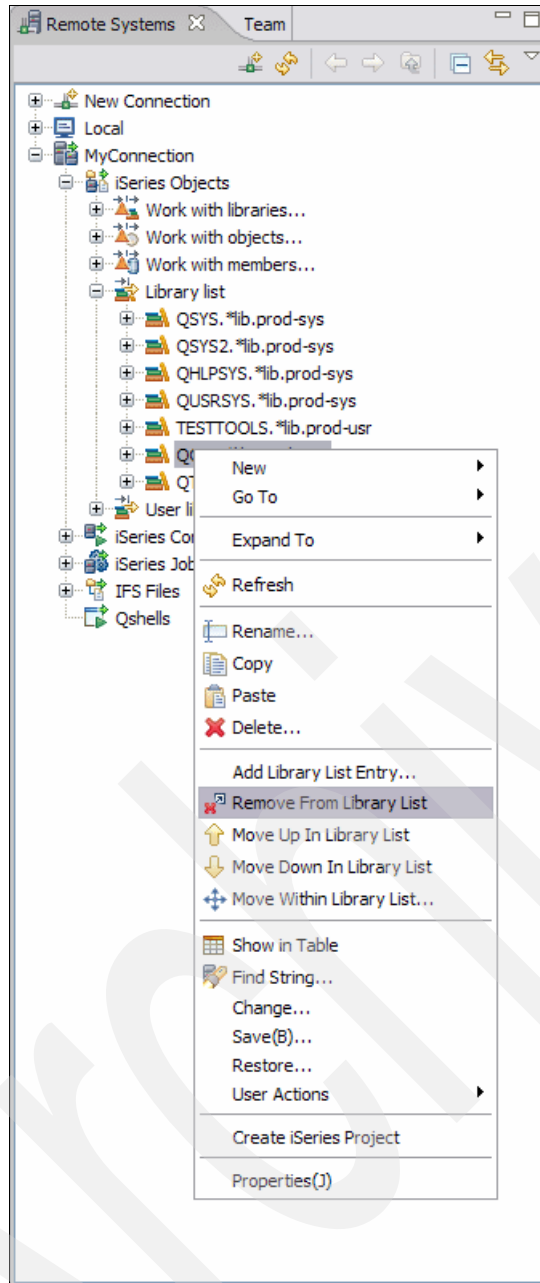


Figure 7-80 Library context menu functions

7.5.4 Moving libraries within your library list

You may also change the sequence of the libraries in the user portion of your library list by right-clicking any library under the library list filter and selecting **Move Up In Library List** or **Move Down in Library List**, as shown in Figure 7-80 on page 209.

Up or down in the library list

To move a library up or down in the library list, follow these steps:

1. In the Remote Systems view, expand **System i Objects** → **Library list**.

2. Right-click the *user* library you want to move. Select **Move up in library list** or **Move down in library list** from the context menu shown in Figure 7-80.

Moving libraries within the library list

To move libraries within the library list, follow these steps:

1. In the Remote Systems view, expand **System i Objects** → **Library list**.
2. Right-click the *user* library you want to move and select **Move Within Library List**. The Move Library List Entry window opens (Figure 7-81).
3. In the Move Library List Entry window, complete these tasks:
 - a. From the Reference library drop-down list, select a reference library.
 - b. Select one of the radio buttons that specifies where to position the library relative to the reference library.
 - c. Click **OK**.

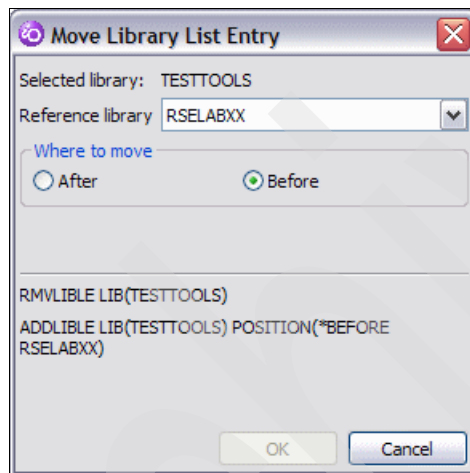


Figure 7-81 Moving a library within a library list

7.5.5 Specifying initial libraries and commands

In the RSE, you can specify libraries to be added to the initial library list, specify the current library at connect time, and specify to run a command every time you connect.

To specify initial libraries and commands, complete these steps:

1. In the Remote Systems view, expand your connection name.
2. Right-click any subsystem node and select **Properties**.
3. The Properties window (Figure 7-82) for the subsystem opens. Complete the following tasks:
 - a. In the left side pane, click **Initial Library List**.
 - b. Enter the name of the library to be added to the library list in the Library field. Click **Add(B)** to add the library into the list below. Do this for all libraries that you want to add to the library list.
 - c. After you add one or more libraries to the list and select one of them, the Remove button becomes active and allows you to remove the libraries from the list.
 - d. After you added more than one library, and select one of them, the Move Up and Move Down buttons become active and allow you to position the libraries within the list.

- e. In the Current library field, optionally specify or select the current library (optional) from the drop-down list. The default is *USRPRF.
- f. In the Initial command field, optionally enter a command to be executed every time the connection is made.
- g. Click **OK**.

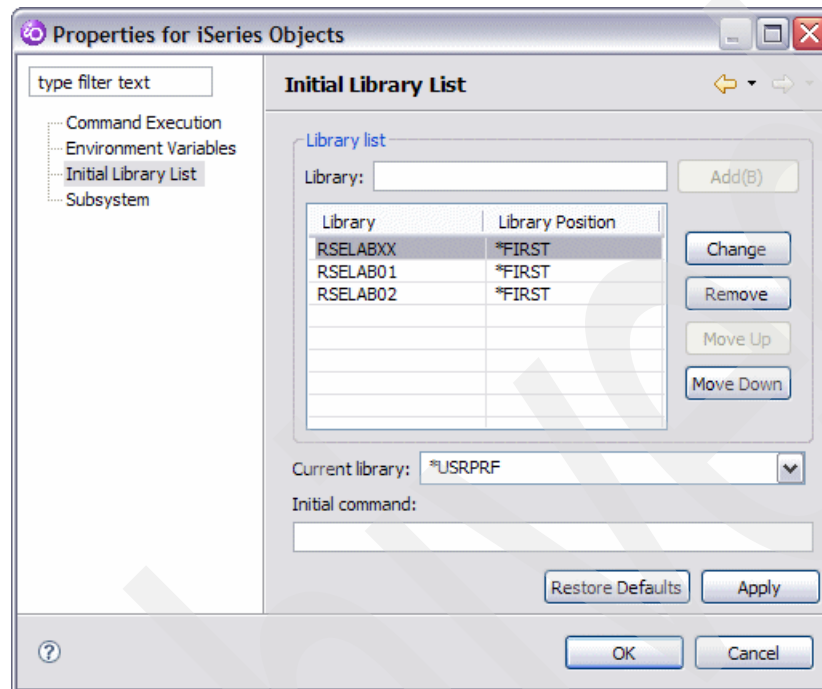


Figure 7-82 Specifying initial libraries and commands

These library list changes take place every time you connect the RSE connection, in addition to whatever is specified in the job description for the user profile that signs on. If you are frequently working with different environments, for example maintaining multiple applications, each having its own distinct library list, you can create multiple connections pointing to the same System i but using a different library list.

7.6 System i Table views

System i Table views are an alternative for displaying information within the RSE. Table views take the currently selected object in the System i Objects, System i Jobs, or IFS Files subsystems as input and display the contents in a table. With this view, you can see item properties at the same time since they are displayed as rows in a table.

You can perform the following tasks with the System i Table view:

- ▶ Navigation
- ▶ Command prompting and running
- ▶ Direct editing of table entries
- ▶ Object and source subsetting and positioning
- ▶ View locking and unlocking
- ▶ Programming Development Manager actions
- ▶ Displaying of additional information
- ▶ List printing

With the table view, you can list and sort libraries, objects, and members. You can also change, copy, rename, delete, edit, compile and run items in the view from the pop-up menu (by right-clicking the item). The System i Table view is similar in functionality to PDM.

Difference between System i Table and Remote Systems Details view

There are two very similar but yet different types of views:

- ▶ The System i Table view
- ▶ The Remote Systems Details view

Both views automatically appear in the lower pane whenever you open the Remote System Explorer perspective or reset it to its original settings (**Window** → **Reset Perspective**).

By double-clicking a row in the Remote Systems Details view you can drill down to the subsystems, from there to the filters, objects and members. However, it does not only contain i5/OS and OS/400 objects, but also everything else, such as jobs, IFS files and IFS directories. The Remote Systems Details view also shows files of non-System i connections, for example the files on your local workstation.

The System i Table view is also located in the lower right pane, but does not contain any content, unless you select objects to be displayed as described in 7.6.1, “Populating System i Table views” on page 212. It is specifically for displaying and working with objects and members from the QSYS file system and is designed for easy migration to the RSE for PDM users and support of PDM actions and a command entry area, which the Remote Systems Details view does not.

The System i Table view and the Remote Systems view share the same windows for the same functions. These functions are also synchronized. For example, if you rename an item in the table view, the change is reflected in the Remote Systems view.

7.6.1 Populating System i Table views

Before you can use any of the functions provided by table views, you *must* populate the table. To populate a table view from the Remote Systems view, right-click an *appropriate* item in the Remote Systems view and select **Show in Table**.

Note: Appropriate items that can be displayed in a table include:

- ▶ Library, object, and member filters
- ▶ Libraries
- ▶ Source files
- ▶ System i Jobs filters
- ▶ IFS Files filters and folders

The table view displays as part of a tabbed notebook of views. For this example, we populate a table view with items from a library called LIB6961; see Figure 7-83.

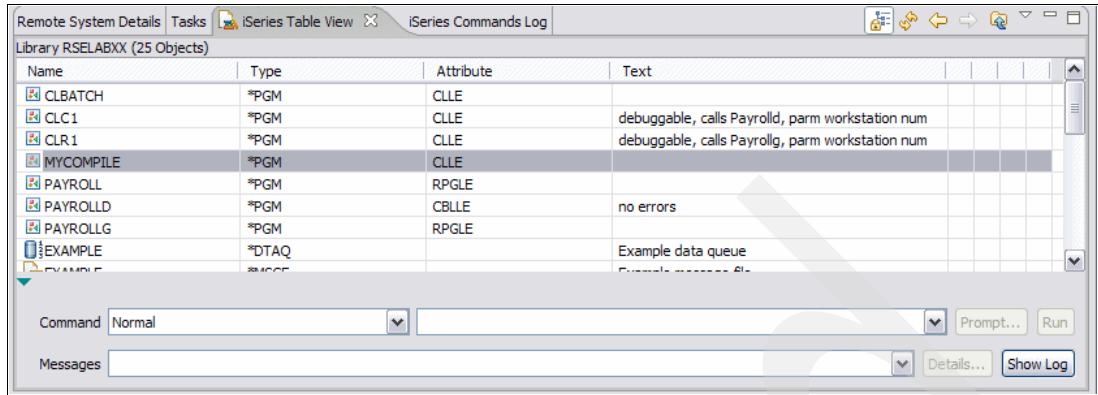


Figure 7-83 Populating an System i Table view

Column titles in the table view change depending on the type of items being displayed. To view more information in the table view, click the **System i Table Menu** (☰) button on the toolbar and select **Preferences**.

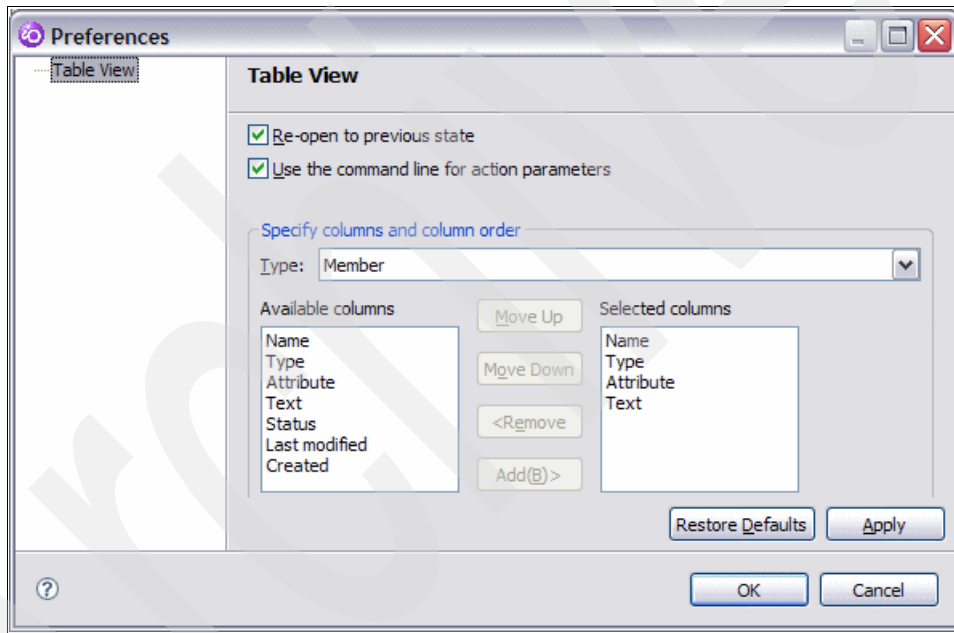


Figure 7-84 Table View Preferences

In the right pane of the **Preferences Table View** dialog, as shown in Figure 7-84, you may add or remove columns to be displayed in Member or Object tables.

7.6.2 Working with items in a table view

Here are some tips for working with items in the table view.

- ▶ Click any of the column titles to *sort* by that attribute.
- ▶ To *print* the list of items in the view, click the **Menu** (☰) button on the toolbar and select **Print**.
- ▶ You can *navigate* through items in the view by double-clicking items that contain other items. For example, you can double-click source files to repopulate the view with the members in the source file. You can double-click IFS folders to display other folders and

files. You can even double-click message files to display message descriptions contained within the message file. When you access editable items, such as a member or Windows file, you can double-click the item to open it in the default editor for that type of item. During the course of your session, if you populate table views, you can revisit each view in history by using the “forward” and “back” buttons on the table view toolbar. See Figure 7-85.

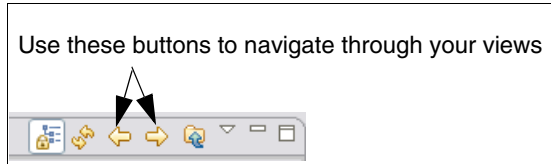


Figure 7-85 Table view navigation buttons

Subsetting objects and members in the Table view

When dealing with long lists of items, it is generally helpful to subset and reorganize the items in the view.

To specify subsetted lists for objects or members, follow these steps:

1. In the populated System i Table view, click the **Menu** (☰) button from the Table view toolbar and select **Subset**. Different Subset windows open depending on the type of items displayed in the Table view. For members, see Figure 7-86. For objects, see Figure 7-87.

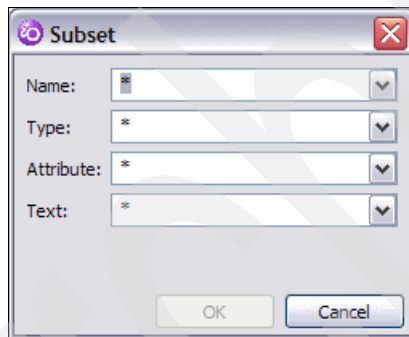


Figure 7-86 Subset window for members

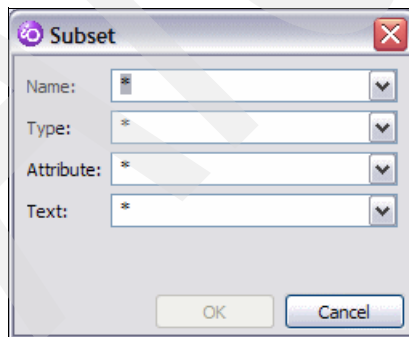


Figure 7-87 Subset window for objects

2. Subset your information by Name, Type, Attribute, or Text.
3. Click **OK**. The table displays the information specified in the Subset window.

After you subset a list, the title bar changes to reflect the new set of items. To revert back to the original display of all items in the table, click the **Menu**(▾) button on the Table view toolbar and select **Show All**.

Positioning objects and members in the Table view

The positioning of items in the System i Table view is similar to the “position to” and “position to type” functions in PDM.

To *position to* items in the Table view, follow these steps:

1. In the populated System i Table view, click the **Menu**(▾) button on the Table view toolbar and select **Position to**. The Position To window (Figure 7-88) opens.
2. In the Position To window, complete these tasks:
 - a. In the Name field, click the drop-down list and select ***TOP** or ***BOTTOM**. Or enter the name (or partial name) of the item that you want to position to.
 - b. In the Type field (only for objects), leave the default as asterisk, or enter the type. If you do not specify the type, then the first type that is displayed or visible at the top of the view is used.
 - c. Click **OK**.

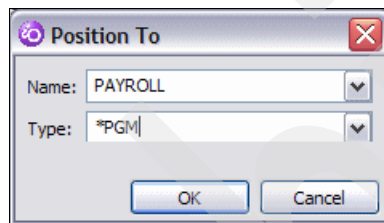


Figure 7-88 Position To window

Tip: You can also invoke the Position To function by entering the first letter of the name that you want to position to in the table. In addition, if you press Shift+*first letter* repeatedly, the wizard cycles through all the names that start with that letter.

Direct editing of table view items

In the System i Table view, you can directly edit properties of members and objects. This function corresponds to direct editing in PDM.

You can edit the following item properties directly:

- ▶ For objects, you can edit the Name and Text properties.
- ▶ For members, you can edit the Name, Type, and Text properties.

To directly edit item properties, follow these steps:

1. Click the row in the System i Table view that contains the property you want to edit.
2. In the selected row, click the cell you want to edit.
3. Make the change directly in the cell and press Enter. If you press the Escape key instead of the Enter key while inside the cell, no change takes place. You may need to refresh (**File** → **Refresh**) the screen for the changes to be updated on the Workbench views or simply unlock the Table view.

Locking and unlocking table views

The System i Table view toolbar contains a lock button that controls the correlation between the Remote Systems view and the System i Table tree view. See Figure 7-89.



Figure 7-89 Lock button

If the lock is *disabled*, then whenever you click an object or library in the Remote Systems view, the associated contents of that item automatically populate the System i Table view.

If the lock is *enabled* (the default state), then clicking various items in the Remote Systems view does *not* change the input to the System i Table view.

To enable or disable the lock, click the button once to change its state.

Field Table view

You can not only show libraries, objects and members in an System i Table view, for externally described files you can even show the fields for each record format by right-clicking the file name in the Remote Systems view or System i Table view and selecting **Show in Table** → **Fields** from the context menu. Figure 7-90 shows an example for a physical file.

You may also show the fields for other externally defined files, such as display and printer file. If the file contains more than one record format, you can either right-click the file name to see all fields in all record formats, or right-click an individual format to see only the fields used in that format.

Name	Record	Type	L...	Text	Alias
CID	CSRCD	Character	4	Customer ID	
CDID	CSRCD	Packed Decimal	3.0	District ID	
CWID	CSRCD	Character	4	Warehouse ID	
CFIRST	CSRCD	Character	16	First Name	
CINIT	CSRCD	Character	2	Middle Initial	
CLAST	CSRCD	Character	16	Last Name	
CLDATE	CSRCD	Zoned Decimal	8.0	Date of DB Build	
CADDR1	CSRCD	Character	20	Address Line 1	
CCREDIT	CSRCD	Character	2	Credit Status	
CADDR2	CSRCD	Character	20	Address Line 2	
CDCT	CSRCD	Packed Decimal	5.4	Discount Percentage	
CCITY	CSRCD	Character	20	City	
CSTATE	CSRCD	Character	2	State	
CZIP	CSRCD	Character	10	Zip Code	
CPHONE	CSRCD	Character	16	Phone Number	
CBAL	CSRCD	Packed Decimal	7.2	Customer Balance	
CCRDLM	CSRCD	Packed Decimal	7.2	Credit Limit	
CYTD	CSRCD	Packed Decimal	13.2	Customer YTD	
CPAYCNT	CSRCD	Packed Decimal	5.0	Customer Payments	
CDELCNT	CSRCD	Packed Decimal	5.0	Customer Deliveries	
CLTIME	CSRCD	Zoned Decimal	6.0	Time of DB Build	
CDATA	CSRCD	Character	500	Customer Data	

Figure 7-90 Field Table view for a physical file

New feature of WDSC V7: The Data Table View

The System i Data Table View allows you to display the contents of data physical file member using the System i Data Table View. This view applies only to data physical files, not source, and operates only on one member of a data physical file at a time.

The file records in the member are displayed as rows in a table, with columns in the table according to the fields in the record format of the file, and the field values presented in a formatted fashion according to the field data type.

To show the contents of data physical file member in Data Table View:

1. Right-click the member in the Remote systems view.
2. From the context menu, select **Show in Table** → **Data**; see Figure 7-91.

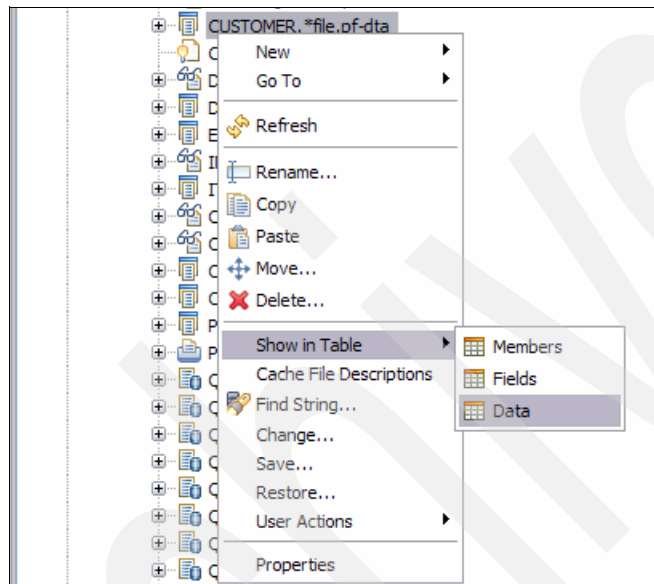


Figure 7-91 Show in Data Table View

3. The System i Data Table View is opened and the data of the physical file member is represented in it; see Figure 7-92.

*RC...	C...	CUSTNA	R.	CONTAC	CPHONE	CFA	CADD
1	0...	Meriden Electronics Limited	4...	Alfredo Bayonne	432-789-67...	206-865-40...	10423
2	0...	Royal Hardware Supplies	2...	Arnie Podell	905-619-20...	905-619-20...	Maple
3	0...	Webster Appliances	1...	Bob Wolfstadt	619-549-52...	619-549-52...	7350 M
4	0...	ProLine Building Supplies	1...	Bud Dobbs	905-403-40...	905-403-40...	73 Mar
6	0...	Universal Communications Ltd.	6...	Dave Franken	415-545-50...	415-545-50...	720 H
7	0...	Baker Electronics	1...	Dave Matthison	818-715-20...	818-715-20...	17150
8	0...	Village Telephone	1...	Dayle Swigger	707-367-45...	707-367-45...	2752 E
10	0...	BelAir Communications incorp.	1...	Doreen Coin	914-765-80...	914-765-80...	302 W
11	0...	Burnham Trading Inc	4...	Efrem Helassie	613-225-07...	613-225-07...	91 Bas
12	0...	Calderone Imports	0...	Elsie Pons	407-392-70...	407-392-70...	702 S.

Figure 7-92 Data representation in System i Data Table View

7.7 User actions

User actions are host commands that you define to run against i5/OS libraries, objects, jobs, and members. They can also be defined for folders and files in any remote UNIX®, Windows, Linux, local, or IFS system.

Each user action consists of the following:

- ▶ A unique name.
- ▶ A command (that may have substitution variables) - The user action specifies in which mode the command is run (normal, batch, or interactive). For more information about running modes, see “Creating your own command set” on page 168 and 7.11.2, “Running modes” on page 241.
- ▶ Substitution variables that are replaced with contextual information when the command is run.
- ▶ An optional comment used to describe the action in more detail.
- ▶ Various options that affect what happens when the command is run, for example, whether to prompt the command first.
- ▶ One or more file types to scope the resource types that apply to the action. For example, a command to start Screen Design Aid (SDA) would be scoped to members of type DSPF. The action is only shown for remote members that are of type DSPF.

7.7.1 Creating user actions

User actions can be created from either the Remote Systems view or from the System i Table view. In this section, we use the Remote Systems view. In the following procedure, we create a user action that adds service programs to a binding directory.

To create a user action, complete the following steps:

1. In the Remote Systems view, expand the **System i Objects** subsystem. Expand any filters if required.
2. Right-click any library, object, or member and select **User Actions** → **Work With User Actions**. The Work With User Actions window shown in Figure 7-93 opens.

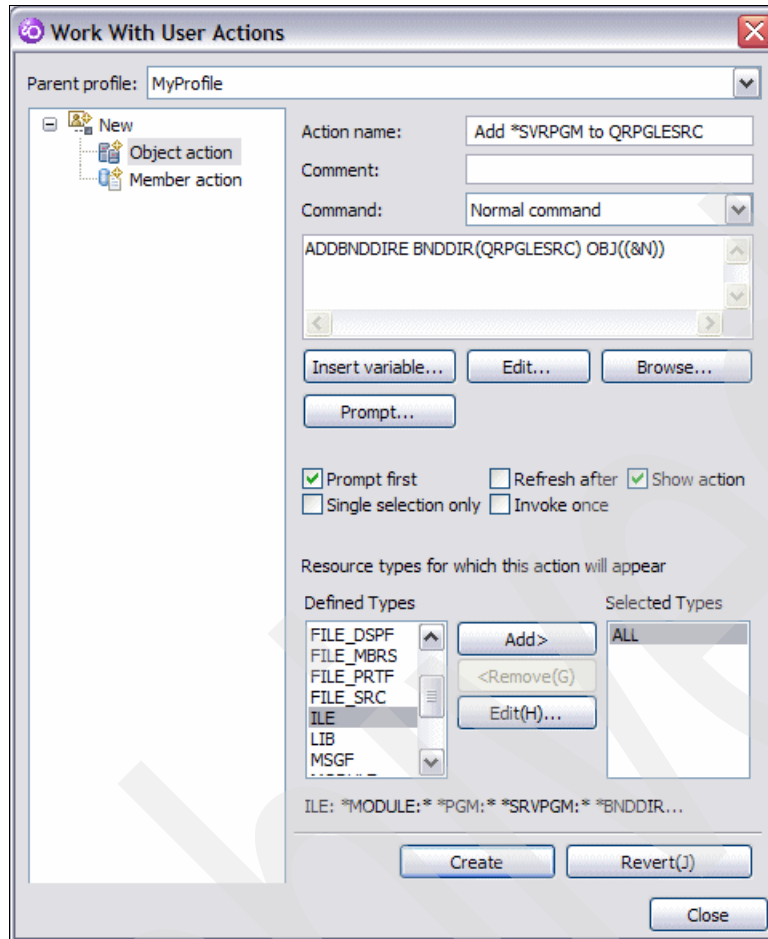


Figure 7-93 Creating a user action

3. In the Work With User Action window, complete the following tasks:
 - a. In the left side pane of the window, click **Object action** under the New node. Entry fields for specifying the user action appear on the right side pane of the window. If you are creating a user action that is to be applied to members, click **Member action** at this point.
 - b. In the Action name field, enter a name for the user action.
 - c. Optionally, in the Comment field, enter a comment that may further describe the purpose of the action.
 - d. From the Command drop-down list, select the execution environment in which this user action will run.
 - e. In the text box underneath the Command text label, enter the command that this user action will run. At this point, you can also:
 - Click **Insert variable** to insert substitution variables into your command string.
 - Click **Browse** to display a list of available i5/OS commands. From here, you can select a command.
 - Click **Prompt** to prompt the command string entered in the text box. This prompt assists you in formatting the command string correctly. When prompting a command, you can also click **Help** in the prompt window to receive further assistance on the command.

- f. Select the appropriate boxes for your user action. These options affect what happens when the command is run. In this example, we selected **Prompt first** and **Show action**. With these selections, the command is prompted when the user action is run and the user action is displayed in the pop-up menu when applicable.
- g. From the Defined Types list, select the object types to which this user action applies. Click the type in the list and then click **Add** to add the types to the Selected Types display area. In this example, we selected the ILE type, which applies to *SRVPGM objects.
- h. Click **Create**, then **Close**.

7.7.2 Invoking user actions

This section is based on the user action we created in 7.7.1, “Creating user actions” on page 218. To invoke a user action, follow these steps:

1. From the Remote Systems view (or a System i Table view), right-click a service program object in the System i Objects subsystem and select **User Actions**. The user action displays in the next pop-up menu; see Figure 7-94.

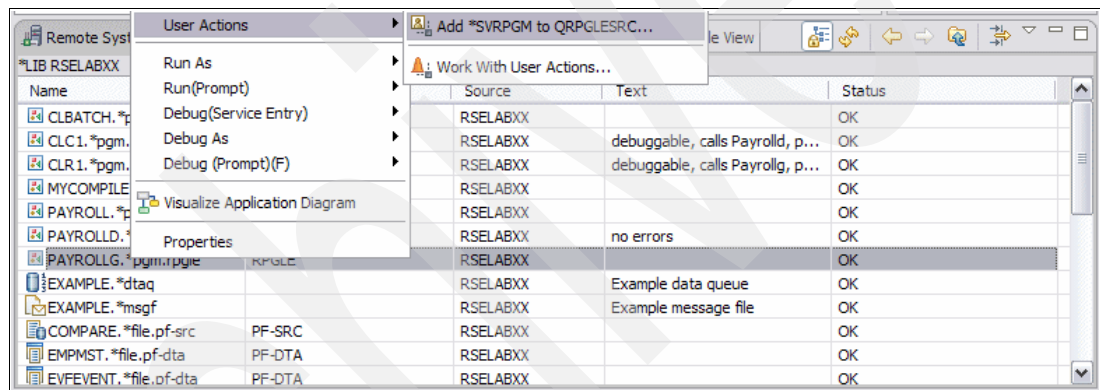


Figure 7-94 invoking a user action

2. Select the user action that appears. In this example, the user action displays as Add *SRVPGM to QRPGLESRC. The command associated with this user action (ADDBNDDIRE) is prompted.
3. From the command prompt window, click **OK** to run the command.
4. Output messages from the command are displayed in the System i Commands Log view as part of the tabbed notebook of views.

7.7.3 Named types

In 7.7.1, “Creating user actions” on page 218, we specify object and member types to which the user action is scoped. In that section, we only chose from a list of pre-supplied types. However, we can also define our own named types.

Named types are collections of resource types that are either object or member types. Named types enable user actions to be scoped to one or more resource types.

Creating a named type

In 7.7.1, “Creating user actions” on page 218, we created a user action for the ADDBNDDIRE command. A resource type of ILE was chosen. The user action created was intended to be used on service program objects only. However, the predefined ILE resource type applies to

modules, programs, and binding directories as well. In this section, we create a new named type that applies to service program objects only.

To create a new named type, follow these steps:

1. Right-click any of the RSE subsystems (except System i Commands) and select **Work With → Named Types**. The Work With Named Types window (Figure 7-95) opens.
2. In the left side pane of the window, there are three expandable nodes. They are:
 - **New**: Expanding this node reveals options for creating either *object types* or *member types*.
 - **Object**: Expanding this node reveals all the *named object types* defined. This includes predefined and user-created named types. Selecting a named object type from the tree allows you to change details pertaining to the type. You can only edit user-defined named object types.
 - **Member**: Expanding this node reveals all the *named member types* defined. This includes predefined and user-created named types. Selecting a named member type from the tree allows you to change details pertaining to the type. You can only edit user-defined named member types.

In the left side pane of the window, expand **New** and select **Object**.

3. Entry fields appear on the right side pane of the window. In the window, complete the following tasks:
 - a. In the Name field, enter a unique name for the object type.
 - b. In the Object type field, specify the object type. Click **Browse** to display a set of valid object types.
 - c. In the Object attribute field, specify the object attribute. Click **Browse** to display a set of valid object attributes.
 - d. Click **Add**. The object type/attribute pair appear in the Object Types display area.
 - e. Repeat steps b through c as many times as required if you want to add more than one object type/attribute pair.
 - f. Click **Create**.
 - g. The Object node in the left side pane expands and displays the newly created named object type at the end of the list. This new type is now available as a resource when scoping user-defined actions.

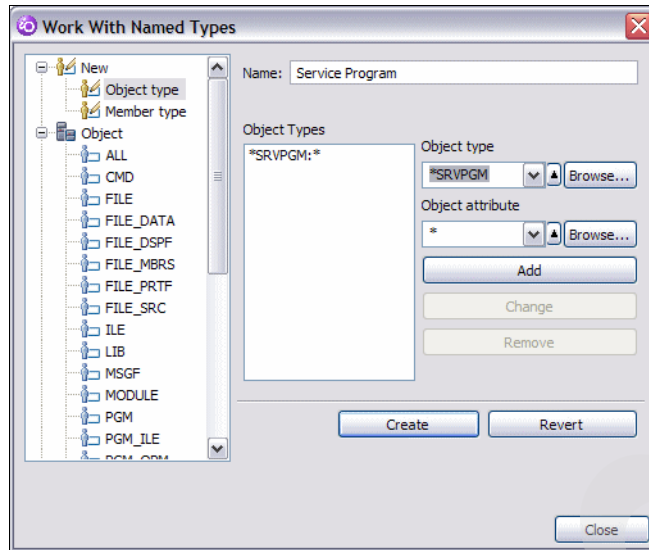


Figure 7-95 Creating a new named object type

Removing a named type

To remove a *user-defined* named type, follow these steps:

1. Right-click any of the RSE subsystems (except System i Commands) and select **Work With** → **Named Types**. The Work With Named Types window opens.
2. In the Work With Named Types window, expand **Object** or **Member**.
3. Right-click the user-defined type from the list that you want to remove. Select **Delete** as shown in Figure 7-96.
4. Click **Yes** on the confirmation window.

Note: IBM pre-supplied named types *cannot* be removed.

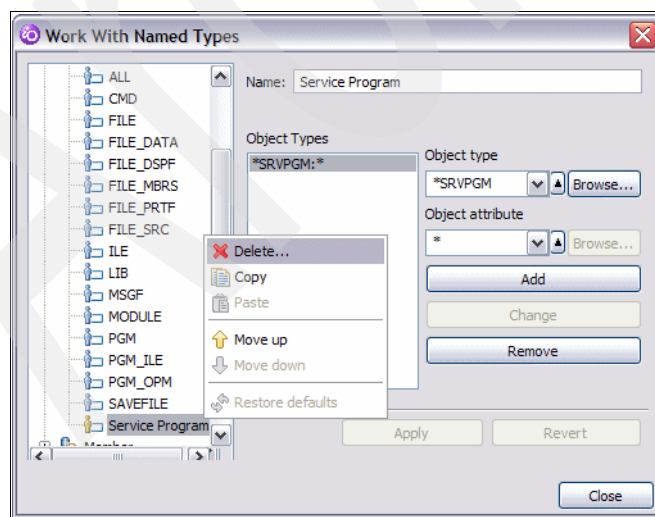


Figure 7-96 Removing a named type

7.8 Additional parameters with actions issued from a table view

You can specify additional parameters in the Table view that will be used for action commands (user actions, PDM actions). This function exists in PDM, is heavily used there, and was a frequent customer request.

In PDM (as well as many other “Work with” panels), before an option is executed, you can type parameters on the command line corresponding to options you typed in the list. These parameters are then added to the command when the option is performed. This function is now available for actions initiated from the Table view. The way it works is the same as in PDM, plus the ability to modify existing parameters.

This capability is very convenient, if you need to perform the same action on many objects or members. The following sections describe sample scenarios.

Changing attributes of multiple files at once

Assume you need to change one or more attributes to the same value for more than one object. For example, you want to set the *Maximum file wait time* and the *Maximum record wait time* to 180 seconds for all physical and logical files in your library. You can do that easily from the RSE view by right-clicking each file and selecting **Change(B)** from the context menu.

However, doing so for more than two or three files can become tedious and you might make mistakes by keying the wrong numbers. Using a Table view can help here if you take these steps:

1. In the Remote Systems view, right-click the library and select **Show in table** from the context menu shown in Figure 7-97 on page 224. All objects of that library are shown now in the Table view in the lower pane of the Workbench.

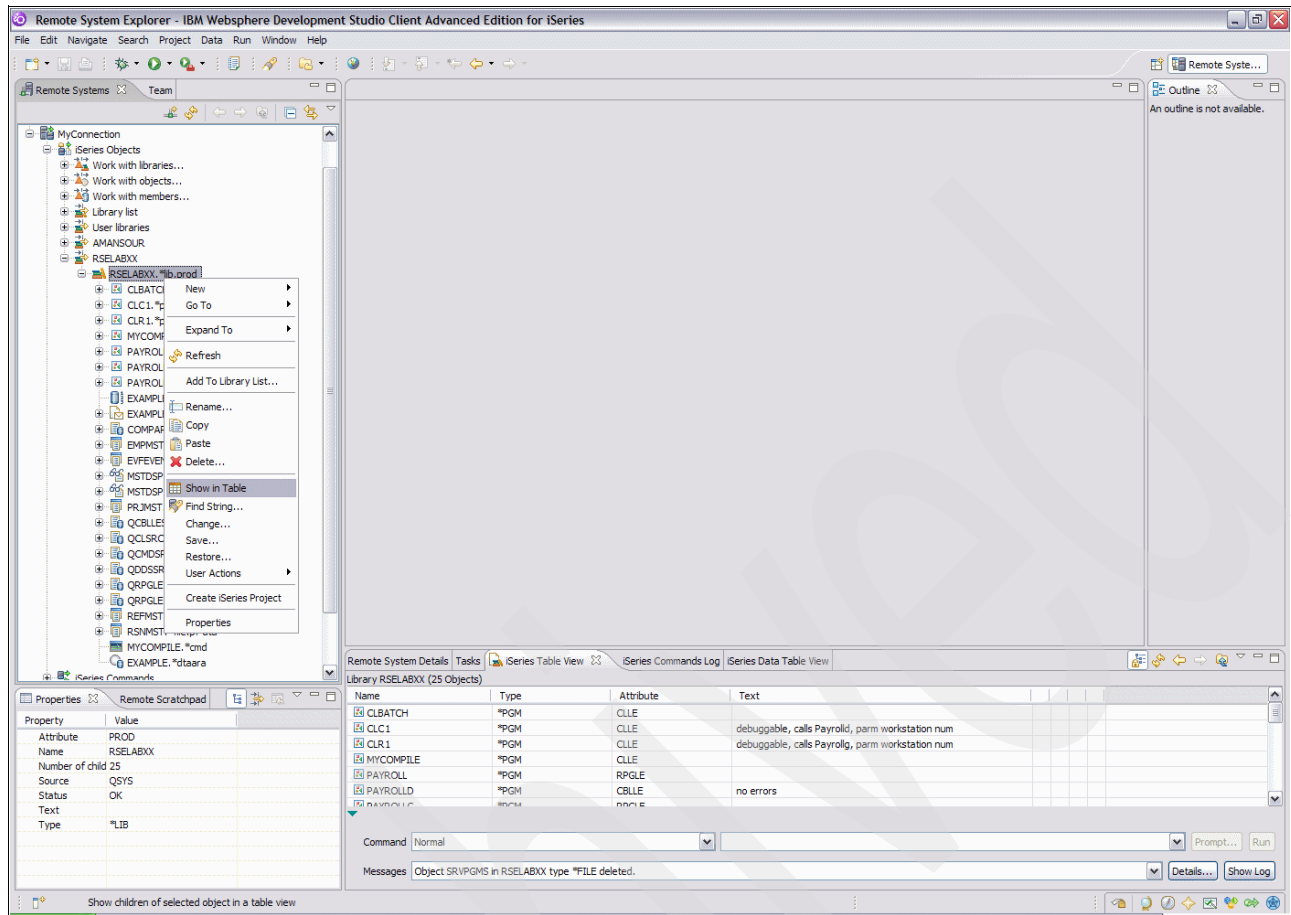


Figure 7-97 Show objects in a table

2. You can now subset the objects as described in , “Subsetting objects and members in the Table view” on page 214 to see only physical and logical files, or you can simply sort all objects by their type and attributes. To do the latter, simply click the header of the *Attribute* column and scroll down till you see the first logical (LF) or physical file (PF-DATA), as shown in Figure 7-98.

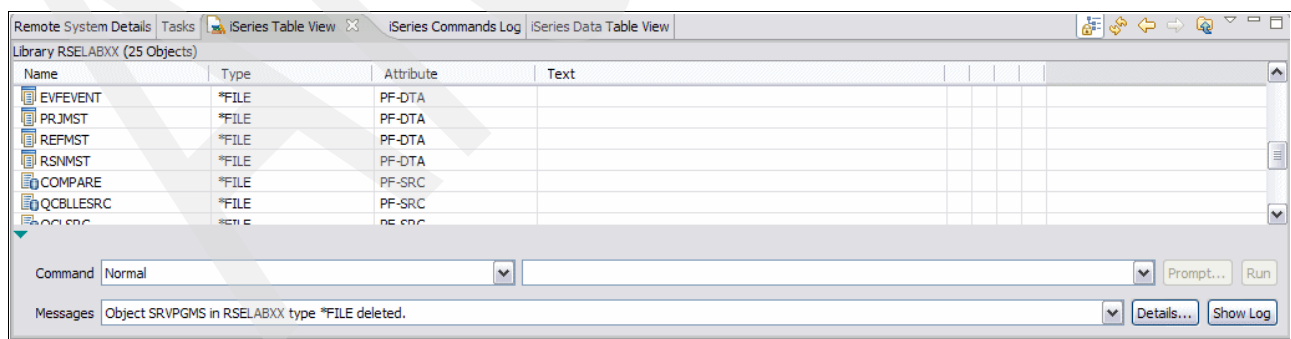


Figure 7-98 Sort table view by object attribute

3. Before you can change all files, you need to find out the names of the parameters to change. To do so, right-click one of the files and select **Change** from the context menu and the Change Logical File dialog appears.

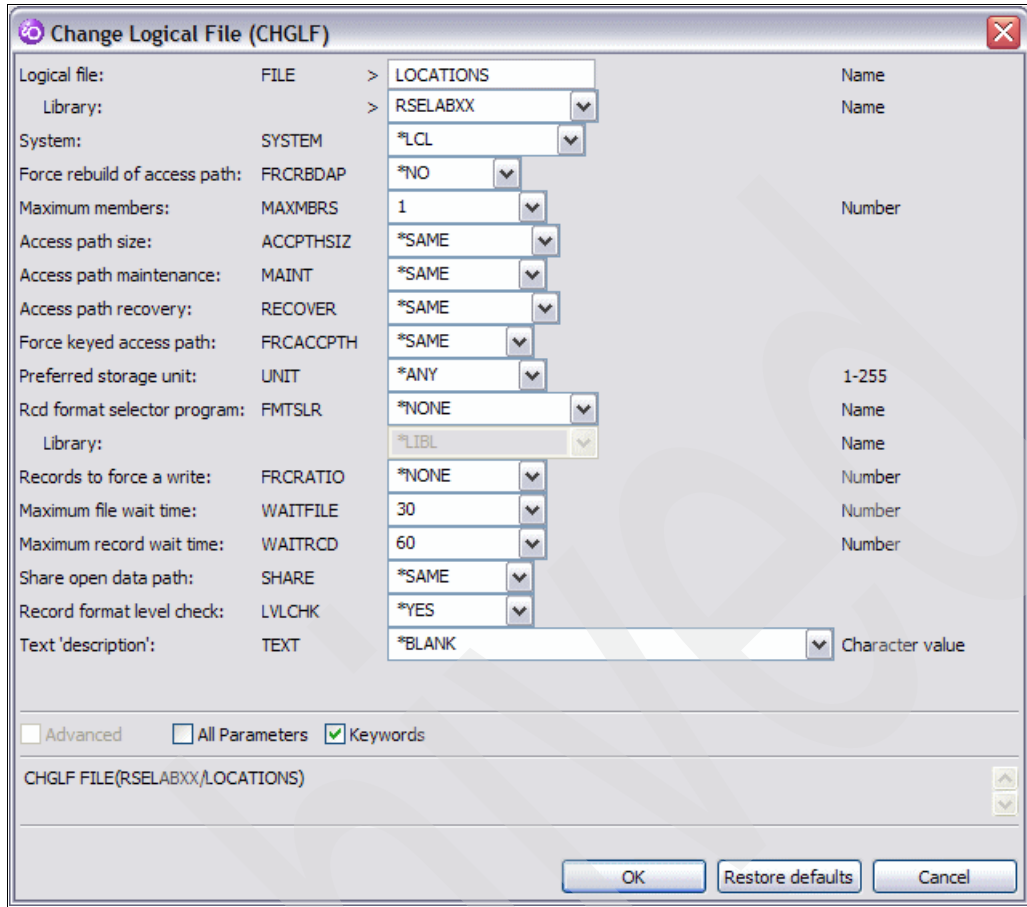


Figure 7-99 Change Logical File dialog

- In our example, we want to change the *Maximum file wait time* and the *Maximum record wait time* to 180 seconds. To see the name (as opposed to the description) of the parameters, select the **Keywords** check box as shown in Figure 7-99. The names of the parameters appear in the column between the parameter descriptions and the input file for the values of the parameters. As you can see in Figure 7-99, for this example the *Maximum file wait time* is called **WAITFILE** and the *Maximum record wait time* has the name **WAITRCD**.
- Since there is no need to change only this particular file, click **Cancel**.

- Back in the Table view enter the parameters along with their values in parentheses into the command line as highlighted in Figure 7-100.

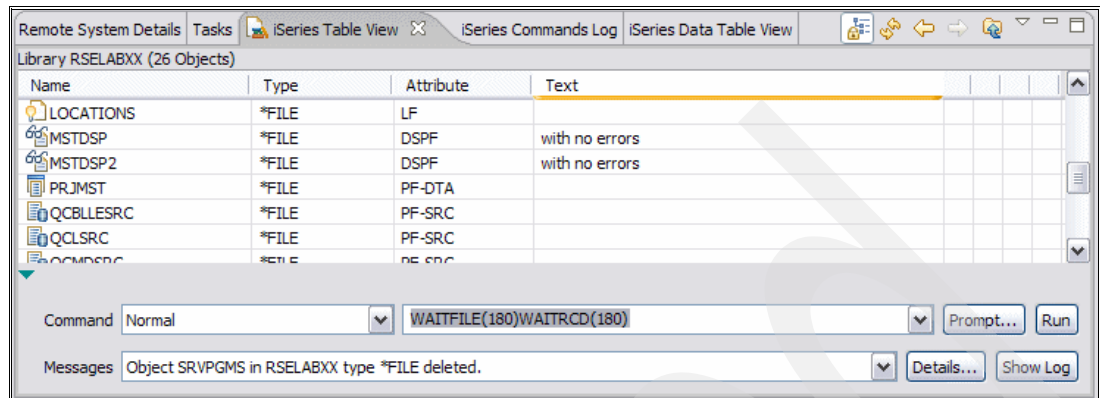


Figure 7-100 Specify additional parameters

- Now select all objects you want to change, right-click them and then select **Change** from the context menu.

Specifying additional parameters applies only to those actions that require executing a command, including user actions. The exceptions are the Paste and Copy actions, which are not supported because they can be run across different views.

Changing the owner of multiple objects

Another good example to use this capability is the situation when you need to change the owner of many or all objects in a library. Since there is no predefined action to change the owner of an object, you need to create a user-defined action by performing the following steps:

- Right-click a library or object in the Remote Systems view or System i Table View and select **User Actions** → **Work With User Actions...** from the context menu and the Work With User Actions dialog appears.

- In the left pane, under **New**, select **Object Action** as shown in Figure 7-101.

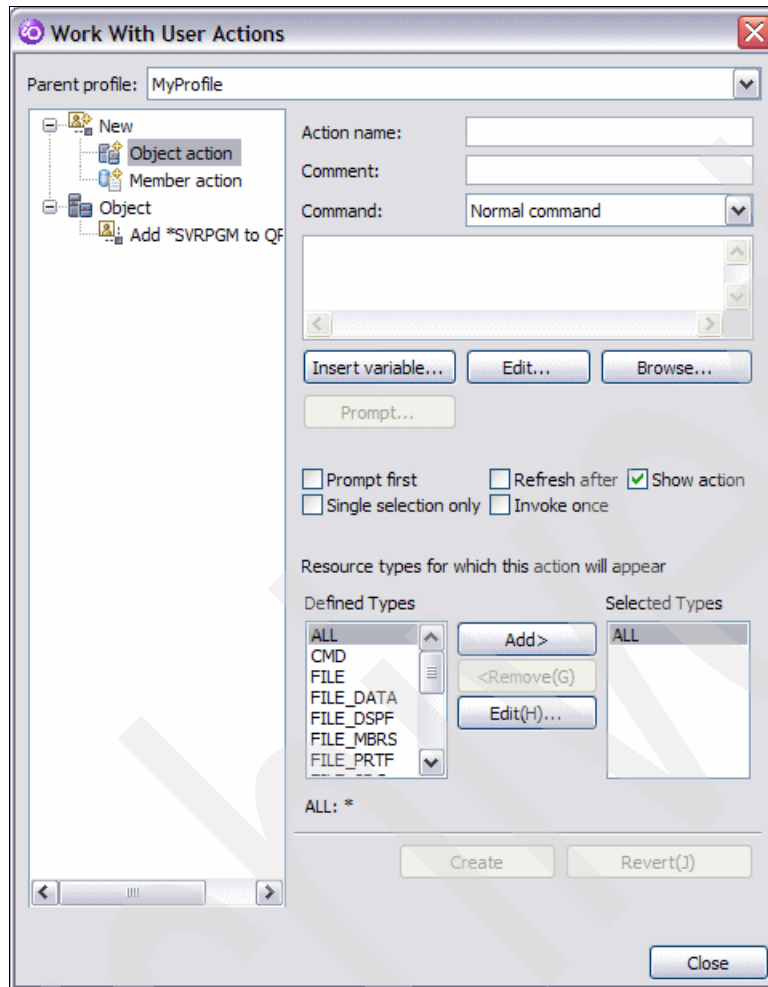


Figure 7-101 Work with user actions

- On the right side of the pane, in the field *Action name*: enter the name you want to use for the new action, for example Change Object Owner.
- In the field *Comment*:, you may enter a more detailed description, for example Change the owner of the selected objects.
- Enter the name of the command to be executed in the *Command*: field. In our example, it is the CHGOBJOWN command.
- To make sure the names of the selected objects are used as parameters, you should insert variables. To do so click **Insert Variable** and select **&L - Object or Member Library Name**, which inserts the name of the library for the selected object.
- On the right of the &L in the *Command*: field, key in a slash (/) as the delimiter between library and object name.
- Click **Insert Variable** again and select **&N - Name of selected resource**, which inserts the name of the library for the selected object.
- On the right of the &N in the *Command*: field, key in a space as the delimiter between object name and object type.
- Click **Insert Variable** again and select **&T - Object or member type with asterisk**, which inserts the name of the library for the selected object.

11. To make sure you used the correct syntax for the Change Object Owner (CHGOBJOWN) command, click **Prompt** and the Change Object Owner prompt shown in Figure 7-102 appears.

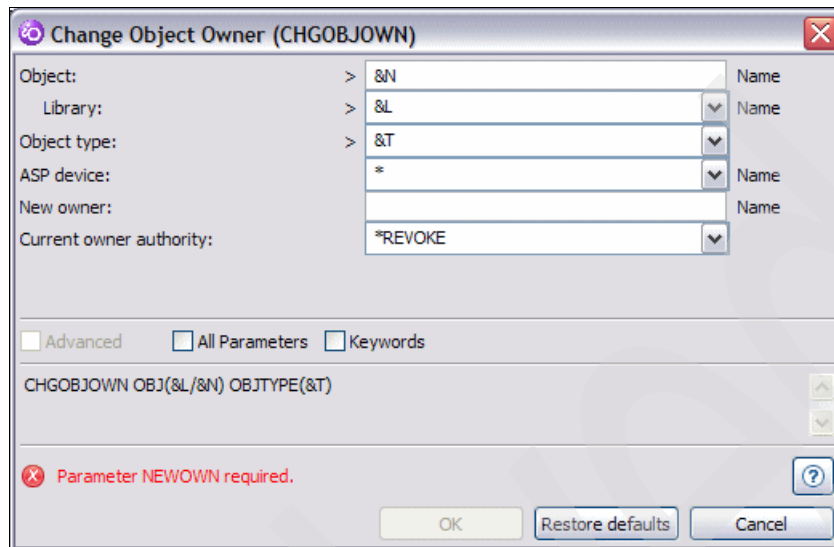


Figure 7-102 Change Object Owner prompt

As you can see in Figure 7-102, the New Owner parameter is required and needs to be inserted when the action is executed. If you want to provide a default value, such as QPGMR, you can enter it here. You can also use the substitution variable &U to insert the name of the user profile used for the RSE connection at the time when this user action is executed.

12. To see the names of the parameters, select the **Keywords** check box.
13. Click **Cancel** to close the prompt dialog.
14. If you want a command prompt to appear before the action is executed, select the **Prompt first** check box. Note that the command is prompted for *each* object, if you run the action against multiple objects.
15. Click **Create** to create your new user action.
16. Click **Close**.
17. Enter the parameter NEWOWN(*MyProfile*), where *MyProfile* is the name of the new owner, into the command field as shown in Figure 7-103.

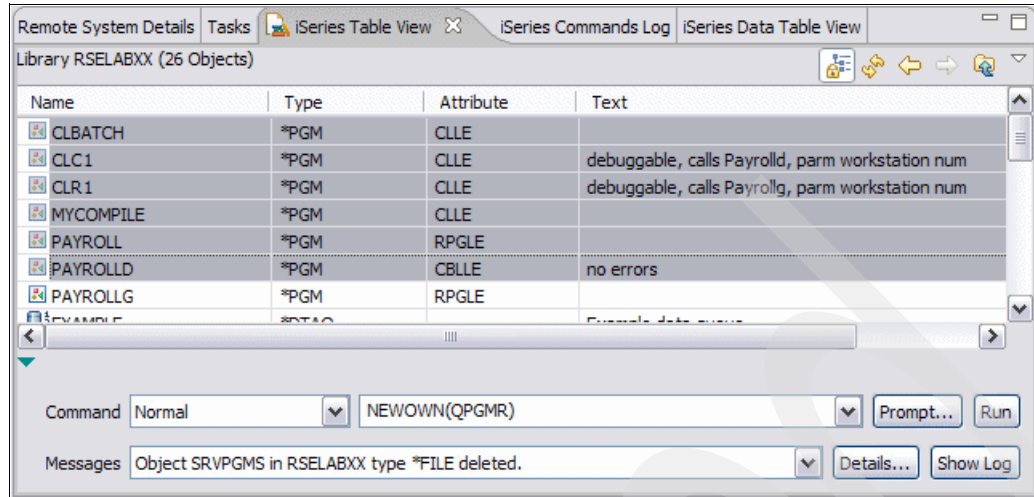


Figure 7-103 Specify a new owner parameter

18. Select one or more objects in the Table view, as shown in Figure 7-103, right-click, and select **User Actions** → **Change Object Owner** from the context menu. The command CHGOBJOWN is run against all selected objects.

7.9 Drag and drop functionality

The RSE supports drag and drop functionality. Dragging and dropping within the RSE *copies* items between source and target locations. This functionality provides you with the ability to drag and drop the following:

- ▶ Objects and members between libraries and source files on the same System i host.
- ▶ Source files and members between libraries on different System i hosts.
- ▶ Between different views: For example, you can drag and drop items between the Remote Systems Tree view and the System i Table view.
- ▶ Drag and drop also works for IFS files and directories.
- ▶ You can also drag and drop (or copy and paste) between different system types, it doesn't have to be between two System i systems. For example, you can drag a save file from your System i server to a directory on your local PC.

7.10 Compiling source members in Remote System Explorer

This section describes the process for verifying and compiling source members in the RSE perspective. It also explains how to configure the compile commands to your requirements.

7.10.1 Verifying source members in RSE

When you use RSE for developing your applications, you do not always need to perform a full compile for your programs. As long as you are debugging compile time errors and do not need a program object to be created, verifying the source is sufficient.

The source verify function checks the validity of your source code without creating a compiled object on the System i host. Source verification can be performed on RPG, COBOL, and DDS source members. Verification can also be done offline.

To invoke the verify function, follow these steps:

1. Drill down through the Remote Systems view or System i Table view to find the source member you want to verify.
2. Right-click the source member and select either **Verify** or **Verify (Prompt)**. The difference Between Verify and Verify (Prompt) is that in Verify (Prompt) you are prompted to enter parameters for the verification process; see Figure 7-104. You can also select **Source** from the Workbench menu and select **Verify** or **Verify With Prompt**. To invoke the verifier in this manner, the source member *must* already be opened in the Remote Systems LPEX Editor.

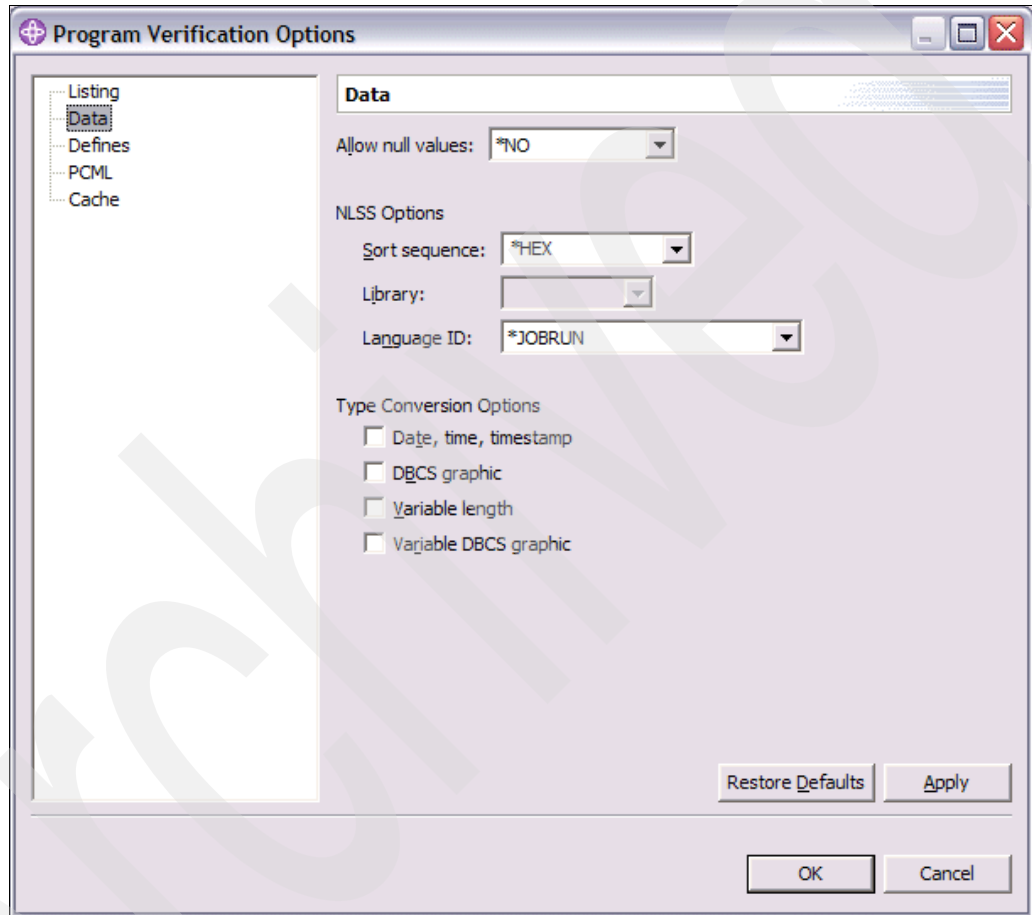


Figure 7-104 The Program verifications options window prompting for RPG

3. The source verifier launches and runs. Any errors found by the verifier are reported in the System i Error List view. Details, such as message ID, message text, severity, and source line number, are displayed for each error found. This view usually appears as part of a tabbed notebook of views similar to the one in Figure 7-105.

ID	Message	Sev...	Line
RNF7030	The name or indicator CSTDTL is not defined.	30	118
RNF7066	Record-Format CSTDTLF not used for input or output.	0	1
RNF7031	The name or indicator DADDR1 is not referenced.	0	5
RNF7031	The name or indicator DADDR2 is not referenced.	0	6
RNF7031	The name or indicator DCITY is not referenced.	0	7
RNF7031	The name or indicator NAME is not referenced.	0	25
RNF7031	The name or indicator DNXTOR is not referenced.	0	12
RNF7066	Record-Format DSRCD not used for input or output.	0	4
RNF7031	The name or indicator DSTATE is not referenced.	0	8

Figure 7-105 The System i Error List after verification errors

4. Double-click an error in the System i Error List view. The source member opens in the editor with the cursor positioned at the source line in error. An error message is also displayed directly beneath the offending source line. Correct the error and the message is removed.
5. Repeat the previous step for each error in the System i Error List view.

Global verifier preferences

Preferences for the source verifier can be set globally. In this example, we set a global preference that instructs the verifier to generate a listing after verification is completed. We set this preference for RPGLE source members.

1. From the Workbench menu, select **Window** → **Preferences**.
2. In the left side pane of the Preferences window, expand **Remote Systems** → **System i** → **Program Verifiers** → **ILE RPG**.
3. In the right side pane of the Preferences window, select the **Generate listing** box.

Now, when a source verification is invoked on an RPGLE source member, a source listing is generated in the System i Listings view, as shown in Figure 7-106.

Name	Date	Time
ORDENTR.RPGLE	Apr 24, 2003	4:23:1
MBR2.RPGLE	Apr 24, 2003	9:03:4
CSTDTLR.RPGLE	Apr 7, 2003	1:18:2

```

Generate program interface . . . : *NO
IBM WebSphere Development Studio Client for iSeries C
Line <----- Source Specifications -----
Number . . .1. . .+ . . .2. . .+ . . .3. . .+ . . .4. . .+ . . .5. . .+ . . .6. . .

                S o u r c e   L i s t i n g

      1 FCstdtld  CF  E                WORKSTN
      2 F                Infds (@F$$)
  
```

Figure 7-106 Source listing in the System i Listings view

To override any global settings temporarily, select the **Verify (Prompt)** option.

7.10.2 Compiling RPGLE source members

This section presents an example that focuses on compiling an RPGLE source member. Most of the concepts and procedures discussed in this section also apply to other System i member types.

This section also focuses on performing these compile-related tasks from the Remote Systems view. This is our choice only for purposes of the demonstration. You can easily perform these tasks from the System i Table view as well.

All compile commands are accessible through the pop-up menu for source members. The pop-up menu is invoked by right-clicking the source member. Only compile commands that are applicable to the member type are shown in the subsequent submenu that opens when you select the compile option.

The compile process

You have two options when compiling your source members in the RSE:

- ▶ **Compile (no prompt):** You are not prompted to edit any parameters for the compile command before the compile starts.
- ▶ **Compile (with prompt):** Before the compile starts, you are prompted to edit parameters for the compile command.

The following example uses the “with prompt” option so that we can show the prompted compile command.

For members of type RPGLE, you can choose to create:

- ▶ A bound RPGLE program: This corresponds to the CRTBNDRPG i5/OS command.
- ▶ An RPGLE module object: This corresponds to the CRTRPGMOD i5/OS command.

To create a bound RPGLE program or RPGLE module object, follow these steps:

1. Navigate the filters in the Remote Systems view until you find the member you want to compile.
2. Right-click the member and perform one of the following actions:
 - Click **Compile (Prompt)** → **CRTRPGMOD** to create a module.
 - Click **Compile (Prompt)** → **CRTBNDRPG** to create a bound program. The Create Bound RPG Program (CRTBNDRPG) command interface window (Figure 7-107) opens.
3. Change or specify command parameters. You can select the **Advanced(5)** check box for further command options, or even check the **All Parameters(6)** check box to show all the parameters available for this command. Select the **Keywords(7)** check box to display parameter keywords.

Tip: The (5) after the word Advanced means that you can also select the Advanced(5) check box using the keyboard shortcut ALT+5. This applies also to letters appearing in parentheses behind any menu item.

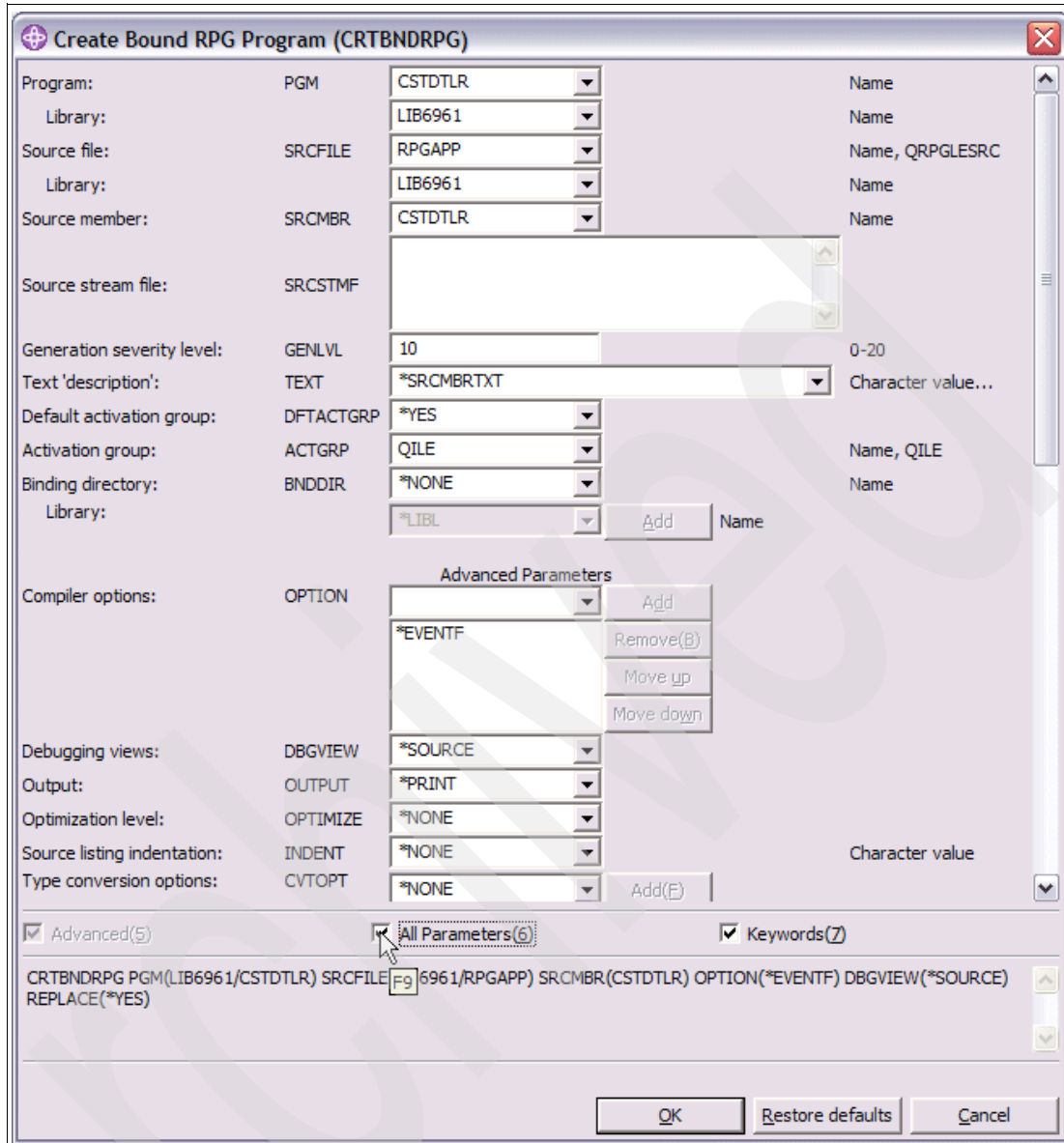


Figure 7-107 Command interface for the CRTBNDRPG command

4. Click **OK** to start the compile process. Messages issued by the compile command are displayed in the System i Commands Log view. See Figure 7-108.

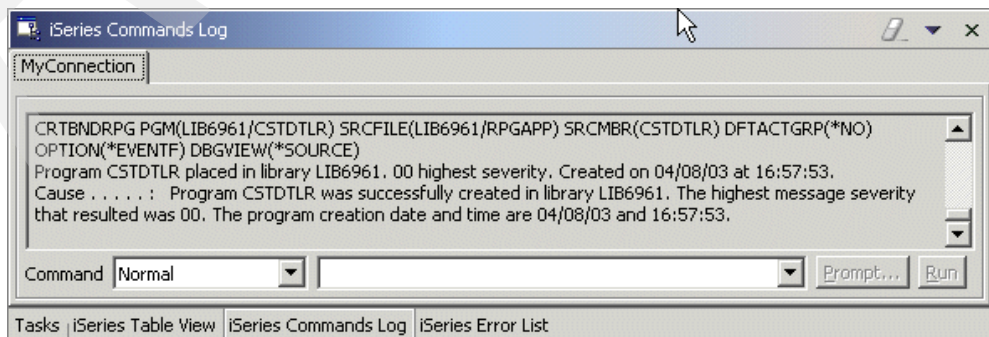


Figure 7-108 Output from the compile command

Any errors found during the compile process are reported in the System i Error List view. This outcome is identical to that of the source verification process. See 7.10.1, “Verifying source members in RSE” on page 229 for details.

5. From the System i Error List, double-click the error entry. The editor opens the member and positions the cursor on the offending line of source. Again, this procedure is identical to the one followed after performing a source verification. See 7.10.1, “Verifying source members in RSE” on page 229 for details.
6. After a successful compile, refresh the Remote Systems view so you can see the newly created program or module object.

Creating programs and service programs from modules

Modules are the building blocks for ILE programs and service programs. The RSE options to create programs and service programs correspond with the CRTPGM and CRTSRVPGM CL commands.

Creating programs

To create a program from a module object, follow these steps:

1. Drill down through your filters in the Remote Systems view until you find the first module required to create the program. You can use the System i Table View if you prefer. We find it convenient to choose the module that is to be the *program entry procedure (PEP)* module. This is the first module that is called after the program is activated.
2. Right-click the module and select **Create** → **Program**. The Create Program (CRTPGM) window opens.
3. Enter or select values for the command parameters. Click **Advanced** to display and specify more command parameters.
4. Click **OK** to create the program. Messages issued by the command are displayed in the System i Commands Log view.

Creating service programs

To create a service program from a module object, follow these steps:

1. Drill down through your filters in the Remote Systems view until you find the first module required to create the service program. You can use the System i Table View if you prefer.
2. Right-click the module and select **Create** → **Service Program**. The Create Service Program (CRTSRVPGM) window opens.
3. Enter or select values for the command parameters. Select the **Advanced(L)** check box to display and specify more command parameters, or even check the **All Parameters** check box to show all the parameters available for this command. Select the **Keywords** check box to display parameter keywords.
4. Click **OK** to create the service program. Messages issued by the command are displayed in the System i Commands Log view.

Run-time options for compiles

You can set run-time options for compiles either *globally* or specifically for a *connection only*. Specific connection settings override the corresponding global settings.

Setting global run-time options

To set global run-time options for compiles, follow these steps:

1. From the Workbench menu, select **Window** → **Preferences**. The Preferences window shown in Figure 7-109 opens.

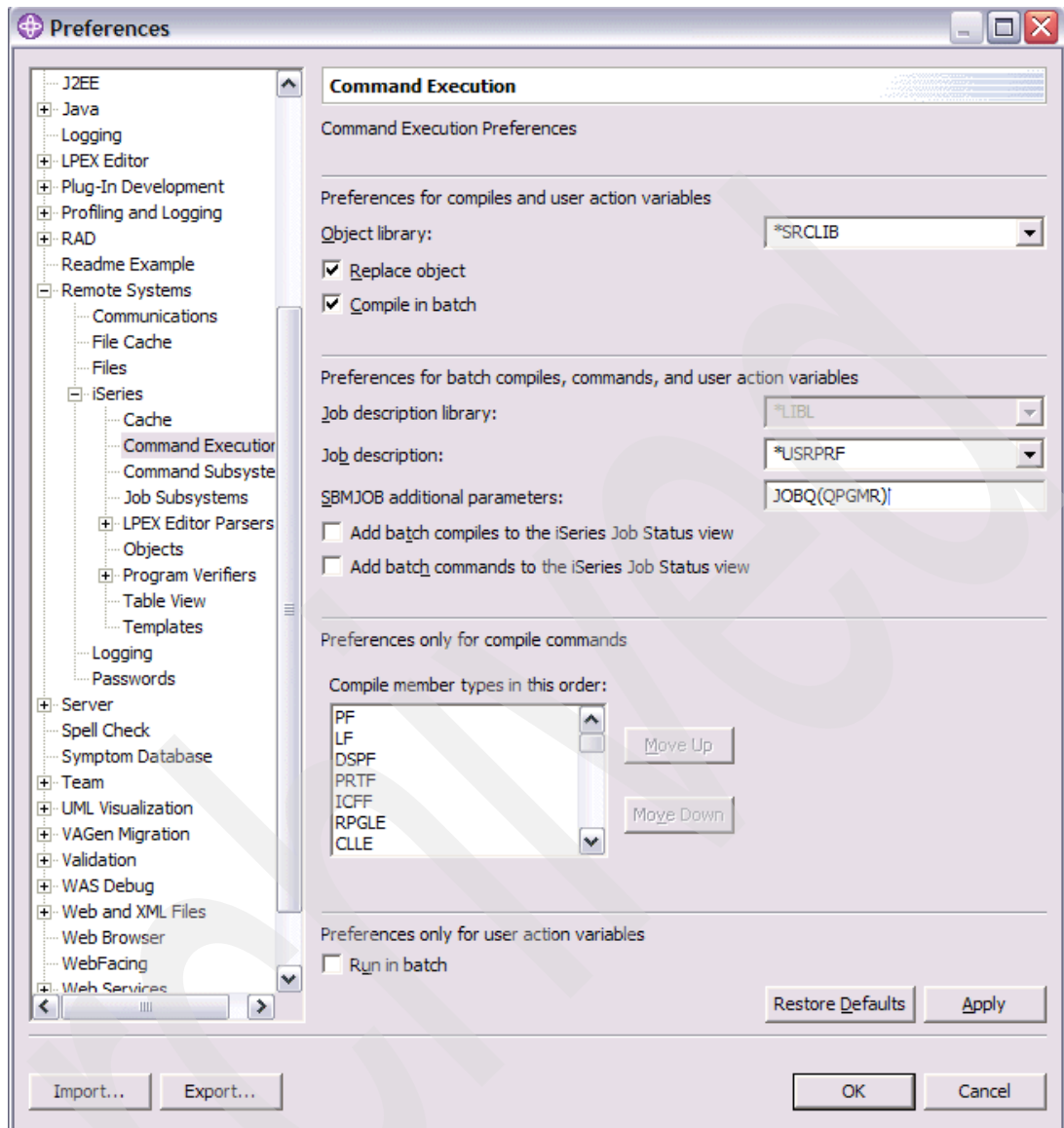


Figure 7-109 Setting global preferences for run-time compiles


2. In the left side pane of this window, expand **Remote Systems** → **System i** → **Command Execution**.
3. In the right-hand side of the window, complete the following tasks:
 - a. In the Preferences for compiles and user action variables section:
 - i. Enter or select the object library where compiled objects are placed. The default is the library where the source is stored (*SRCLIB).
 - ii. Select the **Replace object** check box to specify whether a compiled object is replaced if one exists.
 - iii. Select the **Compile in batch** check box to specify whether compiles are to run in batch. If batch is not selected, compiles run in the RSE communications server job.
 - b. In the Preferences for batch compiles, commands, and user action variables section:
 - i. In the Job description field, enter the name of the job description to be used with the batch job. The default is the job description specified in the user profile (*USRPRF).

- ii. In the Job description library field, enter the name of the library where the job description resides. This field can only be changed if you specify a job description other than *USRPRF.
 - iii. In the Submit Job (SBMJOB) additional parameters field, specify additional parameters to append to the SBJJOB command for batch compiles.
4. Click **OK** to save your changes.

Setting connection-specific run-time options

To set connection-specific run-time options for compiles, follow these steps:

1. From the Remote Systems view, right-click your connection and select **Properties**. The Properties for *your-connection-name* window opens.
2. In the left side pane of this window, click **Subsystems**.
3. The options and fields in the Properties window look similar to those in the Preferences window discussed in “Setting global run-time options” on page 234.

For connection-specific changes, click the toggle **arrow** () button on each option that you want to change so that the arrow on the button points to the right. When the toggle button is in this state, the corresponding option is activated, which allows you to change its value. When the arrow on the toggle button points to the left, the corresponding option is disabled (grayed out) and the global setting is used. See “Setting global run-time options” on page 234.

4. Click **OK** to save your changes.

7.10.3 Working with compile commands

You can configure compile commands to suit your requirements. Configuration usually involves changing parameter values on these commands. This section demonstrates this procedure by changing two command parameters for the Create Bound RPG Program (CRTBNDRPG) command.

To change compile command parameters, follow these steps:

1. Drill down through your filters in the Remote Systems view until you find an RPGLE source member. You can use the System i Table view as well if you prefer.
2. Right-click the member and select **Compile** or **Compile (Prompt)**. Then select **Work With Compile Commands**. The Work With Compile Commands window, shown in Figure 7-110, opens.

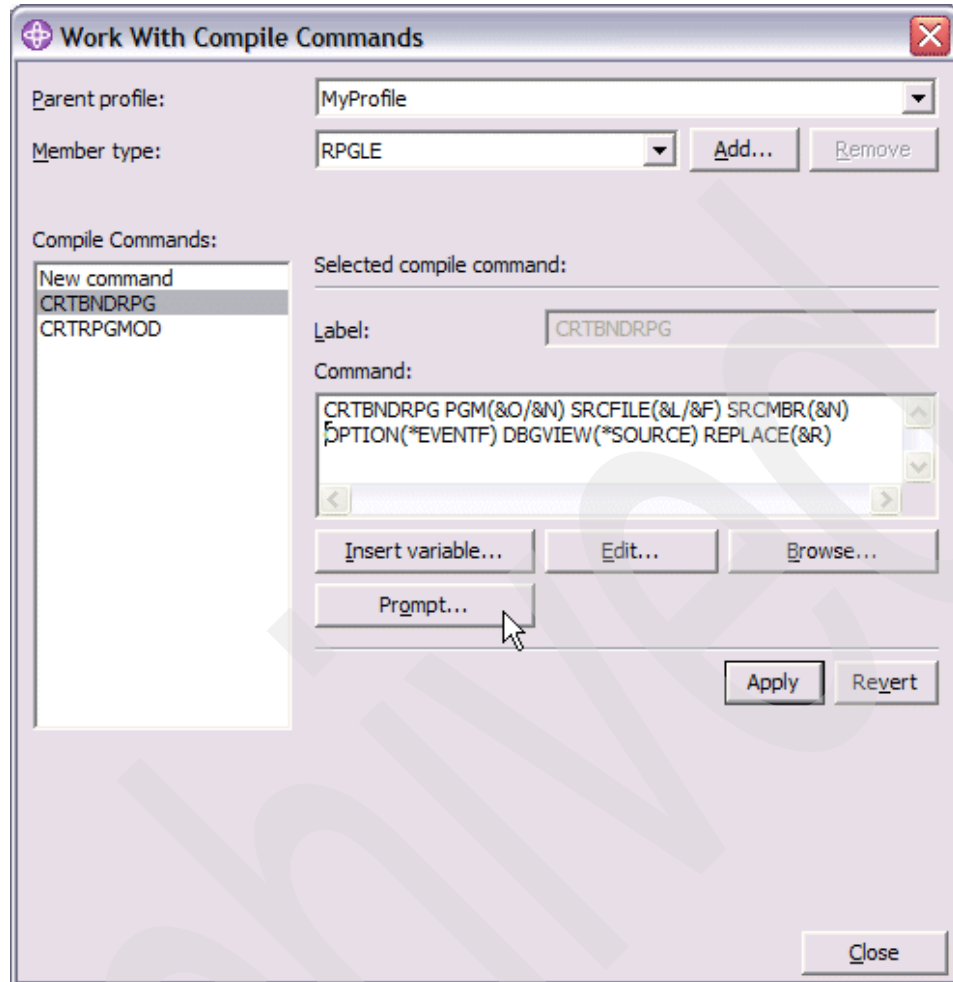


Figure 7-110 Work with Compile Commands window before changes are made

3. In this window, complete the following tasks:
 - a. Make any appropriate changes to the Parent profile field. If this is a change for your own personal profile, you will not need to change this field.
 - b. Leave the Member type field set to RPGLE.
 - c. In the Compile Commands list, select **CRTBNDRPG**. The Label field is populated with the value CRTBNDRPG (this cannot be changed) and the Command text box is populated with the i5/OS command string currently associated with the label.
 - d. Click **Prompt**. The prompt window for the CRTBNDRPG command opens.
 - e. In the prompt window, complete the following tasks:
 - i. Click the **Default activation group** drop-down list, and select ***NO**.
 - ii. Select the **Advanced** check box to display and specify more command parameters, or even check the **All Parameters** check box to show all the parameters available for this command. Select the **Keywords** check box to display parameter keywords.
 - iii. Scroll down, click the **Debugging views** drop-down list, and select ***STMT**.
 - iv. Click **OK** to save your changes.

The CRTBNDRPG command string in the Command text box reflects the changes you made in the command prompt window from the previous step; see Figure 7-111. You can also edit the command string directly in the text box if you prefer.

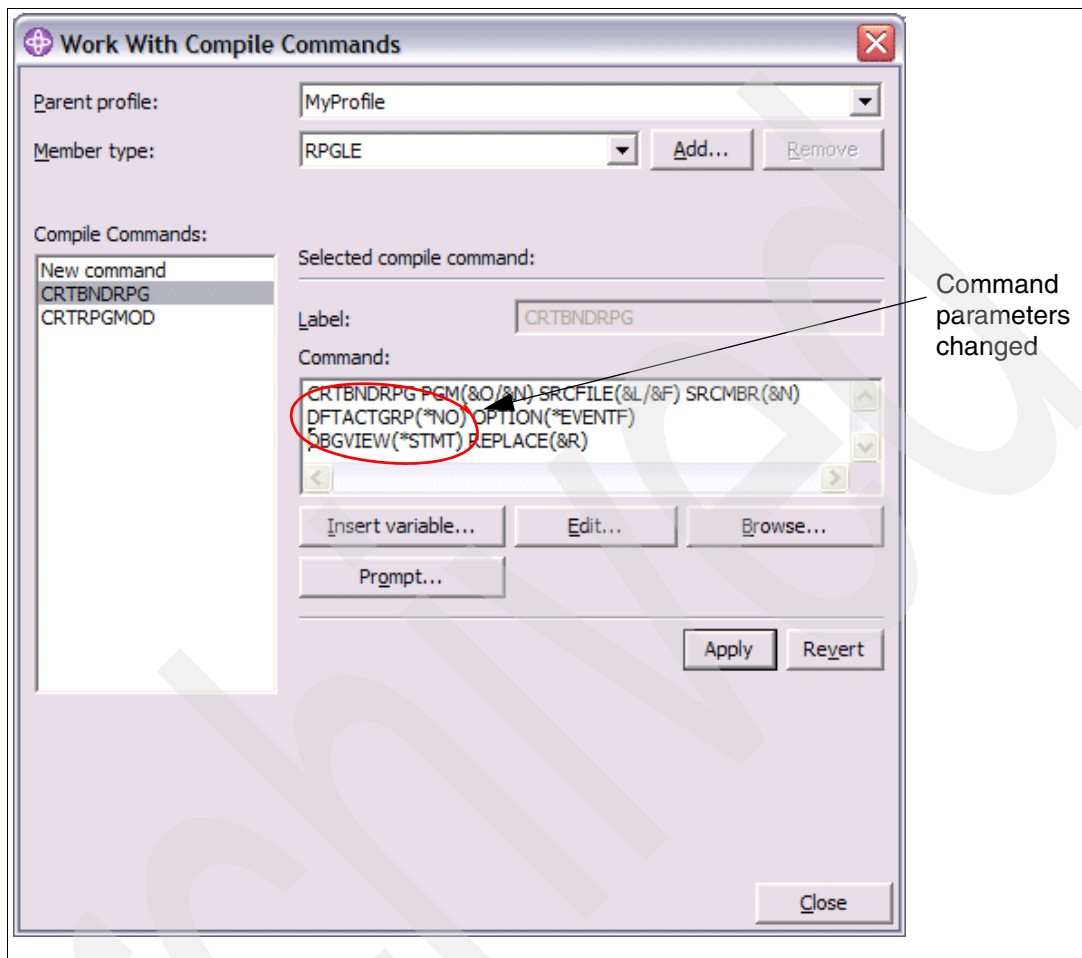


Figure 7-111 Work with Compile Commands window after changes are made

4. Click **Apply** to save your changes.
5. Click **Close** to close the window.

Changes made to compile commands are persistent and scoped to a profile. The preceding procedure is analogous to using the Change Command Defaults (CHGCMDDFT) command to change default values for command parameters. However, within the RSE, different profiles can have different default values for the same compile commands. If you need a different value for a particular parameter, which is rare, you can always select the “compile with prompt” version of the compile command and change the parameter in the prompt window before you run the command.

Important: Do *not* remove the *EVENTF value from the OPTION parameter of any compile command. This option instructs the compiler to create an “events” file, which is used to populate the System i Error List view if errors are found during the compile process.

7.11 Running programs from the Remote System Explorer

This section describes the process for running your programs from the RSE perspective. It also discusses the options available with regards to the mode you choose to run your programs in.

You have two options when running your programs from the RSE:

- ▶ **Run As(B)**: You are not prompted to edit any command parameters before the program runs.
- ▶ **Run(Prompt)(F)**: Before the program runs, you are prompted with a command window allowing you to specify or change command parameters.

In the following example, we use the “with prompt” option so that we can show an example of the prompted run command.

7.11.1 Running a program: An example

To run a program from the RSE, follow these steps:

1. Drill down through your filters in the Remote Systems view until you find a program (*PGM) object that you want to run. You can use the System i Table view if you prefer.
2. Right-click the program and select **Run(Prompt)(B)**. A submenu is displayed that offers four choices for modes in which to run the selected program. Select **System i application in RSE job** for now. See 7.11.2, “Running modes” on page 241, for more information about the different ways to run your programs.

My System i program for run introductory window opens; see Figure 7-112. You can change this name easily by editing in the Name field.

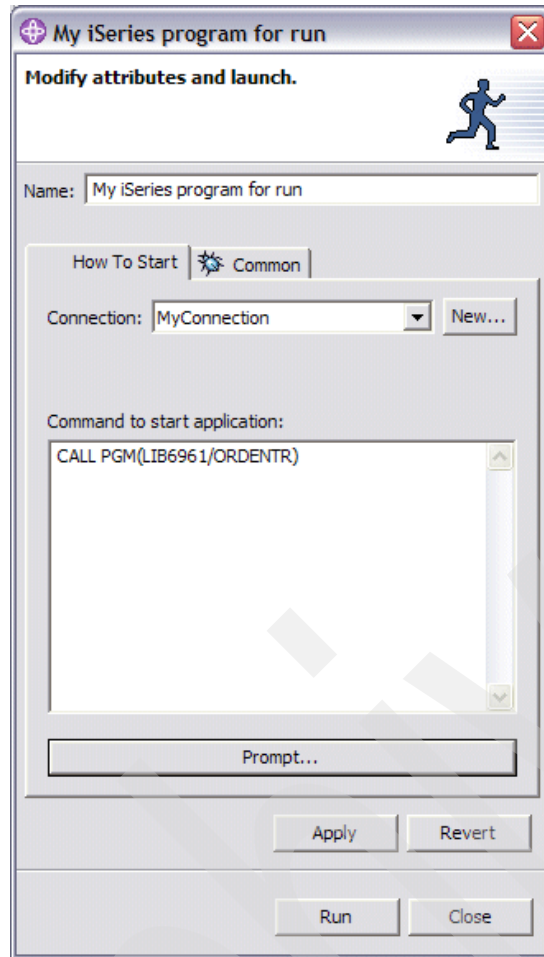


Figure 7-112 My System i program for run introductory window

3. Click **Prompt** and the Call Program window appears; see Figure 7-113. The Program and Library fields are already populated reflecting your program selection.
4. Enter any required program parameters in the Parameters field. Add, remove, and manipulate the parameter list using the Add, Remove, Move up, and Move down buttons.

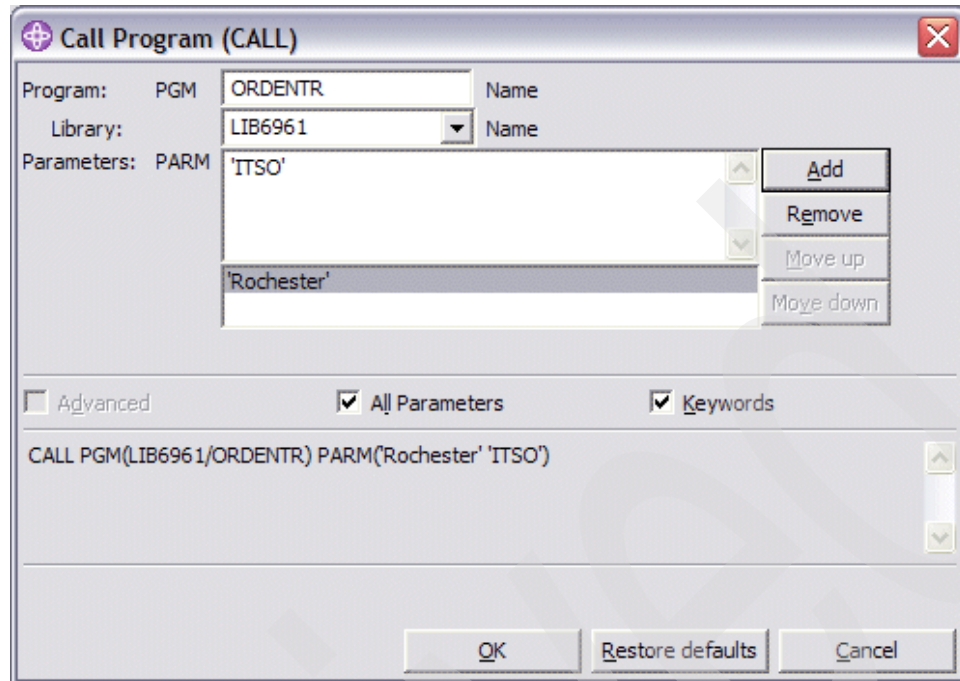


Figure 7-113 Running a program as System i application in the RSE Job

5. Click **OK** to get back to Figure 7-112.
6. Click **Run** to run the program. Messages issued by the CALL command are displayed in the System i Commands Log view.

7.11.2 Running modes

As mentioned in the previous section, you can run your programs in one of four modes. These are:

- ▶ System i application in RSE Job
- ▶ Batch
- ▶ Interactive
- ▶ Multi-threaded

The following sections explain each mode.

System i application in RSE Job

If you choose to run your programs in normal mode, then the programs are run in the RSE communications server job. It is easier to monitor the status of your jobs with this option, although you tie up the communications server. If you have a long-running job, this option may not be the best choice.

When you run your programs in the RSE communications server job, the CALL CL command is used. You are prompted to enter parameters (if required) for your program if you choose the “run with prompt” option. See 7.11.1, “Running a program: An example” on page 239.

Batch

Running your programs in batch mode requires no initial setup and works as you would expect. When you run your programs in a batch job, the SBMJOB CL command is used. You

are prompted to make any changes to the SBMJOB command if you choose the “run with prompt” option.

Messages issued from the SBMJOB command are displayed in the System i Commands Log view, as shown in Figure 7-114.

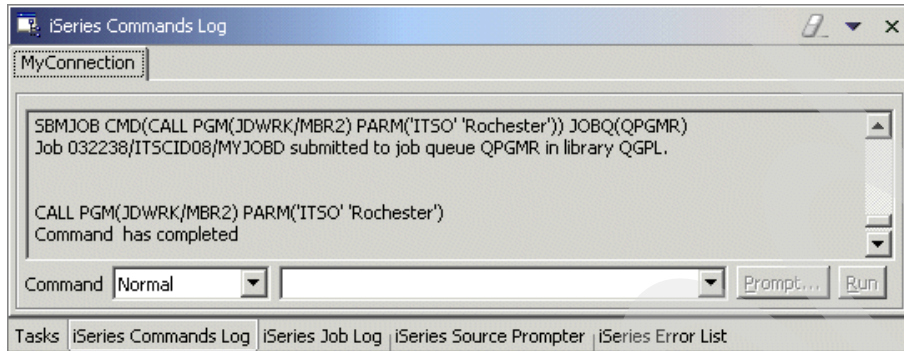


Figure 7-114 Messages issued by the SBMJOB command to the System i Commands Log

Run an interactive program from the Remote Systems view

To run programs in an interactive job, you need a 5250 green-screen emulator. You *must* associate the emulator with a connection in the RSE communications server by issuing the Start RSE Server (STRRSESVR) command in the emulator:

STRRSESVR NAME(*your-connection-name*)

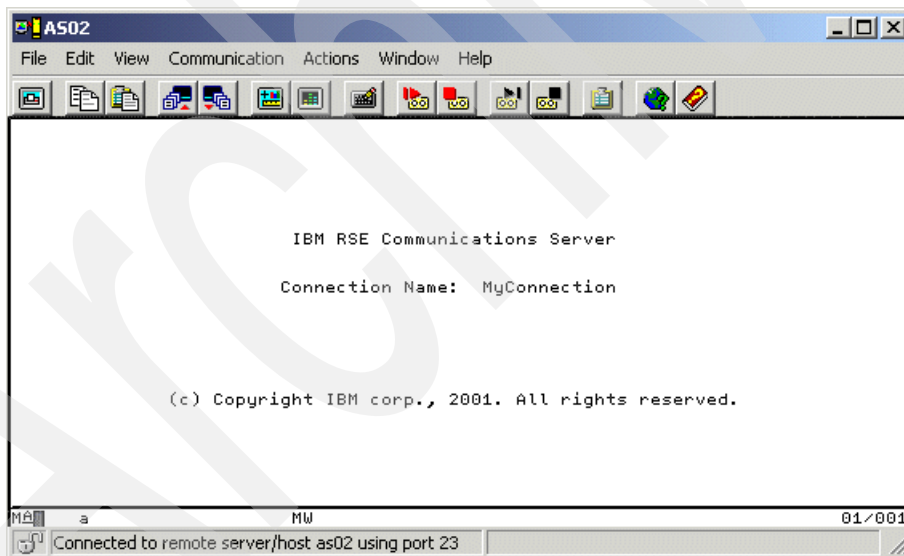


Figure 7-115 Emulator window after issuing the Start RSE Server (STRRSESVR) command

Your-connection-name is a connection that *must* already be defined in the RSE. After you type this command, a panel similar to Figure 7-115 is shown in the emulator session. Note that it is case-sensitive and needs to be specified exactly as in your RSE perspective. You can also specify the following parameters, if you press F4=Prompt:

- ▶ A working library (WRKLIB) specifies the library where the STRRSESVR command creates a data queue to communicate with the RSE server. The default (*DFT) selects QGPL to be used.
- ▶ The remote location name (RMTLOCNAME). It specifies the TCP/IP hostname or IP address of the client machine where the Remote System Explorer System i

communications daemon is running. The special value *PRV ensures to use the value from the last invocation of this command. *RESOLVE attempts to automatically determine the correct address and is recommended when using DHCP.

In some network environments, the client's address cannot be determined and you might receive error message RSE2001 (Error connecting to Remote System Explorer). In this case you may try to enter the IP address of your workstation with parameter RMTLOCNAME.

- ▶ The TCP/IP port number (PORT). The default port number is 4300. If you want to use a different port, you also need to change the value for the *RSE communications daemon port number* under **Preferences** → **Remote Systems** → **Communications**.

Important: The connection name is case-sensitive and needs to be specified exactly as in your RSE perspective.

If you attempt to run a program in an interactive job without performing the previous association, a message window opens with instructions on how to associate an interactive job with your RSE connection, as shown in Figure 7-116.

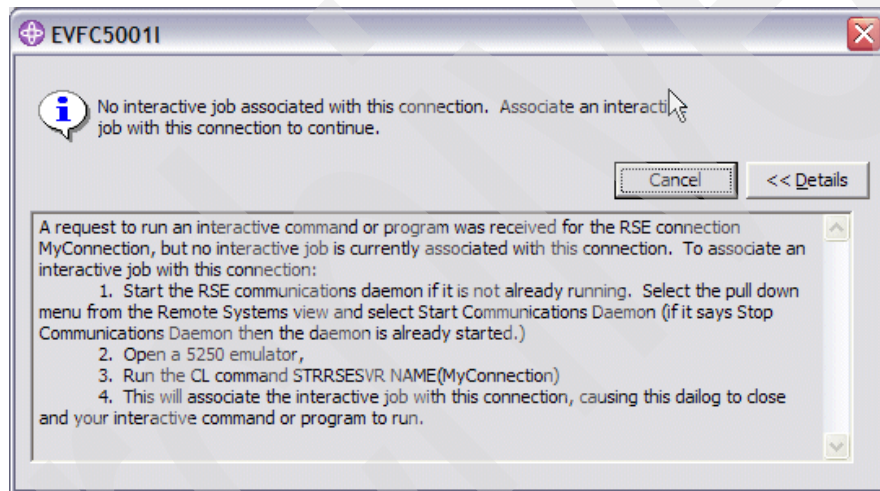


Figure 7-116 Message window for associating an interactive job with a connection

When you run a program in an interactive job, a message is sent to the System i Commands Log (Figure 7-117). The message tells you to switch to the emulator where the STRRSESVR command was issued.

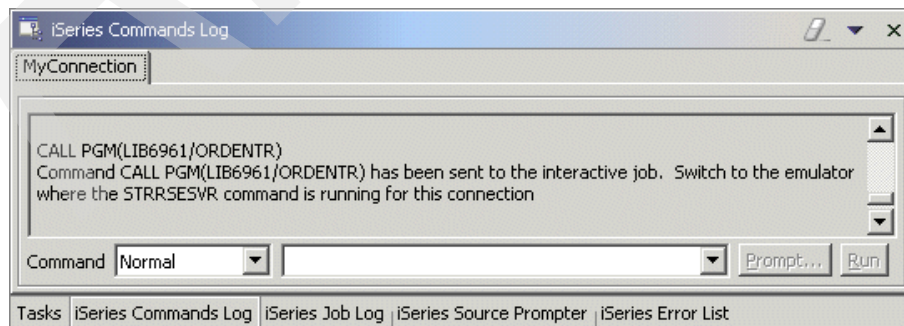


Figure 7-117 Message issued after running a program in an interactive job

Switching to the associated emulator session allows you to view output and interact with the program, as shown in Figure 7-118.

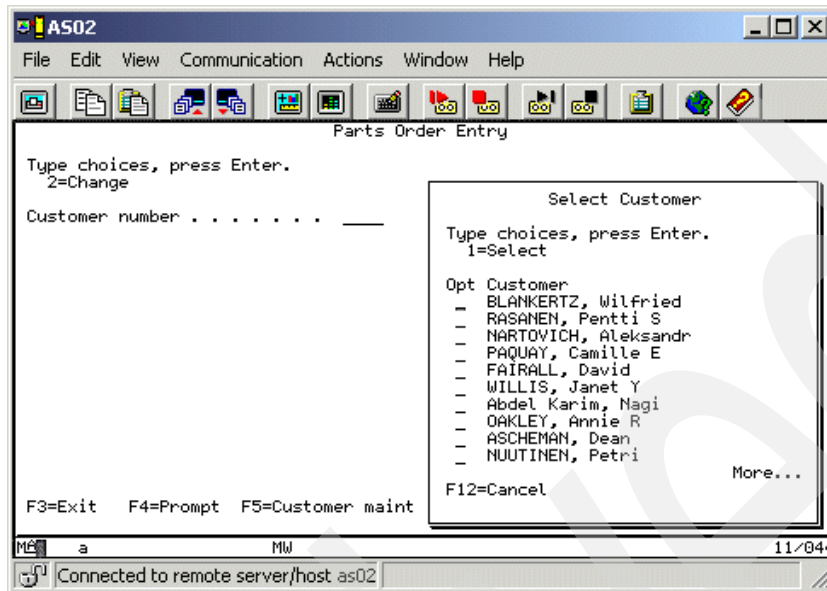


Figure 7-118 Interactive program running in the emulator session

When you end your interactive program, the emulator session shows the display in Figure 7-115 on page 242 again.

Multi-threaded

Programs that spawn new threads have to be started using this option. It is otherwise very similar to the other run modes.

7.11.3 Ad hoc commands

You can run ad hoc commands from the RSE in the System i Commands Log view (Figure 7-119). To run an ad hoc command from this view, follow these steps:

1. From the first drop-down list, select the mode in which the command is to run.
2. Enter your command string in the second entry field. After you enter a command string or partial command string, click **Prompt** to prompt the command. Then enter the rest of the command details in the prompt window.
3. Click **Run** or press Enter to run the command.

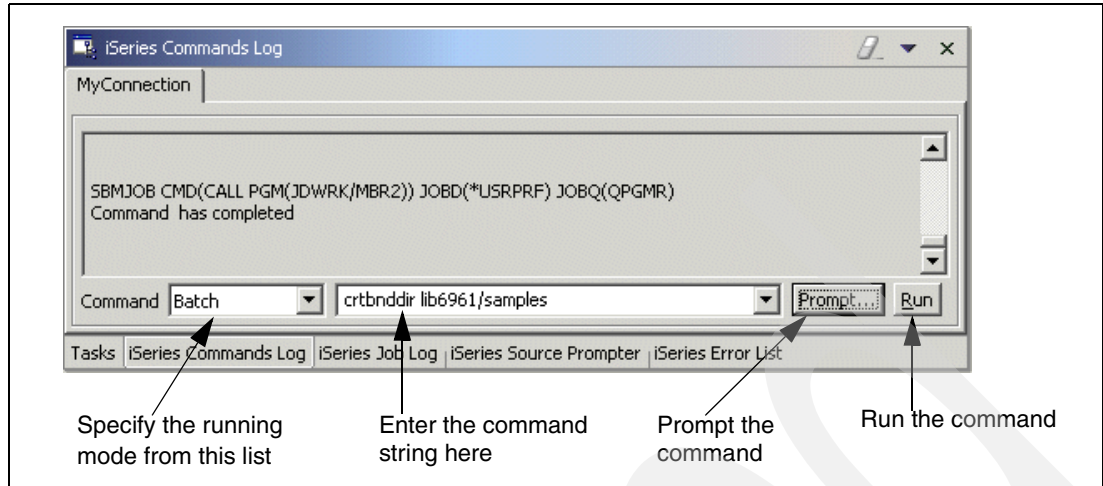


Figure 7-119 Running ad hoc commands in the System i Commands Log view

7.12 Debugging programs in the RSE

For information about the debuggers that are available and how to debug your programs in the RSE, see Chapter 12, "Integrated System i Debugger" on page 407.

7.13 Tips for using the Remote System Explorer

Here we include some miscellaneous tips to help you get started using the RSE.

7.13.1 Use the right mouse button

As a general rule, if you want to perform an action against something in the RSE or debugger, right-click it and look for the action in the pop-up menu. For example, if you need to search a list of members for a specific string, multi-select the members, right-click and select **Find String** from the pop-up menu; see Figure 7-120.

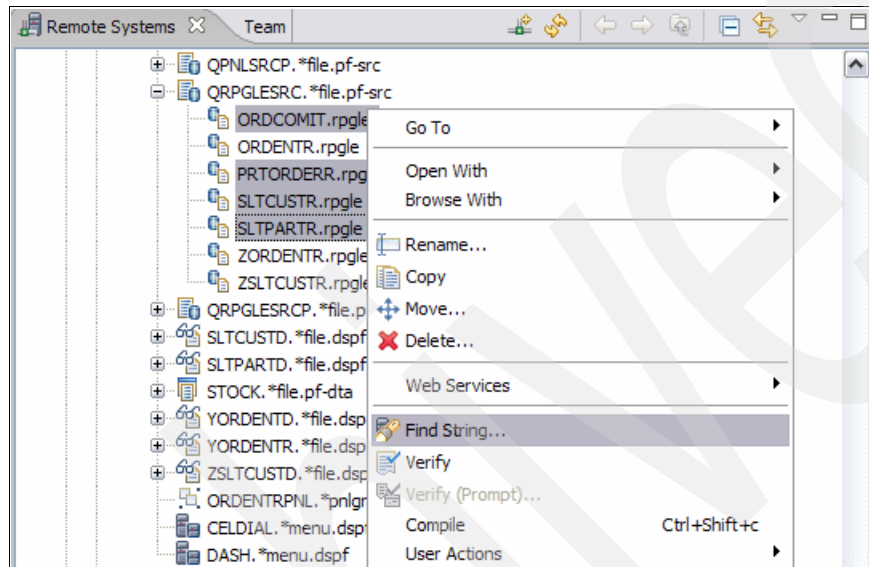


Figure 7-120 Search members for a string

7.13.2 Standard workstation shortcut keys

Most of the standard workstation shortcut keys also work within the workbench. For example; Ctrl+Home or Ctrl+End to go to the top or bottom of a view or editor, Ctrl+c (copy), Ctrl+v (paste), Ctrl+z (undo), Ctrl+y (redo), Ctrl+s (save), using the arrow keys to navigate through a tree or table view, and positioning in a tree or table view by just starting to type the name.

Note: Occasionally the standard shortcuts for copy and paste as mentioned above do not work, but when you right-click and select those actions from the context menu, it does work.

7.13.3 Additional shortcut keys to remember: Ctrl+F6, Ctrl+F7, and Ctrl+F8

Some additional shortcut keys to remember: Ctrl+F6 to switch between open editors, Ctrl+F7 to switch between open views, Ctrl+F8 to switch between open perspectives, and Ctrl+Shift+a to open an i5/OS source member.

7.13.4 Context-sensitive help through F1

Pressing F1 provides context sensitive help based on the currently active view. Typically this help gives you a quick overview and some hyperlinks into the online help.

Also, starting with WDCS V7.0 and WDCS AE V7.0, most dialogs contain a question mark button on the lower left corner. When you click the button, a help window opens at the right of the dialog.

7.13.5 Missing a view? Reset Perspective!

You can drag and drop views around within the Workbench window using the mouse to customize the layout to your liking. Use the **Window** → **Reset Perspective** if you want to get back to the default layout; see Figure 7-121.

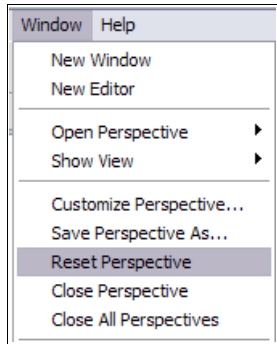


Figure 7-121 Reset perspective

7.13.6 Maximize the view by double-clicking the title bar

Double-clicking the title bar for a view or editor will maximize to take up the entire area of the workbench. Double-clicking it again will minimize it back to the original size. The shortcut key for this is Ctrl+m.

Note: The shortcut key Ctrl+m does not work inside the Remote Systems LPEX Editor because that key combination is assigned to a different action.

Archived

Remote Systems LPEX Editor: The modern SEU and much more

This chapter builds on the concepts and techniques introduced in Chapter 7, “Working with i5/OS-based development resources using the Remote System Explorer” on page 139. It explains how to use the Remote System Explorer (RSE) for specific System i development activities such as the editing, compiling, running, and debugging of your applications. The Remote Systems LPEX Editor is considered to be the modern source entry utility (SEU).

This chapter covers the following topics:

- ▶ Editing source members in the RSE
 - Remote Systems LPEX Editor
 - Editor functions
- ▶ Compiling source members in the RSE
 - Run-time compile options
 - Working with compile commands
- ▶ Running programs from the RSE, including the running modes normal, batch, and interactive
- ▶ Debugging programs in the RSE

Chapter 7, “Working with i5/OS-based development resources using the Remote System Explorer” on page 139, serves as an introduction to the RSE. We *highly* recommend that you read it before you continue with this chapter.

The chapter was written originally based on WDS*c* 5.0 and now updated with the enhancements for WDS*c* 7.0.

8.1 Editing source members in the RSE

This section introduces you to some basic concepts for editing your source members in the default editor within the RSE perspective. The default editor is the *Remote Systems Live Parsing Extensible Editor (LPEX) Editor*. The Remote Systems LPEX Editor is a powerful language-sensitive editor that can create and edit many kinds of files, including programs, documentation, and text files.

However, we focus on using the Remote Systems LPEX Editor for editing System i source members, such as RPG, COBOL, Data Description Specification (DDS), and CL. We also explain and demonstrate some common editing related functions with this editor.

By definition, the LPEX editor is the default editor for source files in the Remote System Explorer, and can be used to create and edit many kinds of files, including program source files, documentation, and data files. In addition to basic editing functions, the LPEX Editor offers language parsing, location marking, elaborate search, and key stroke recording facilities.

8.1.1 Launching the Remote Systems LPEX Editor

To open source members using the Remote Systems LPEX Editor, follow these steps:

1. Navigate the filters in the Remote Systems view until you find the member or members you want to edit. You can also do this from an System i table view.
2. Double-click the member, or right-click the member, and select **Open With** → **Remote Systems LPEX Editor**. The editor launches and loads the member or members. See Figure 8-1.

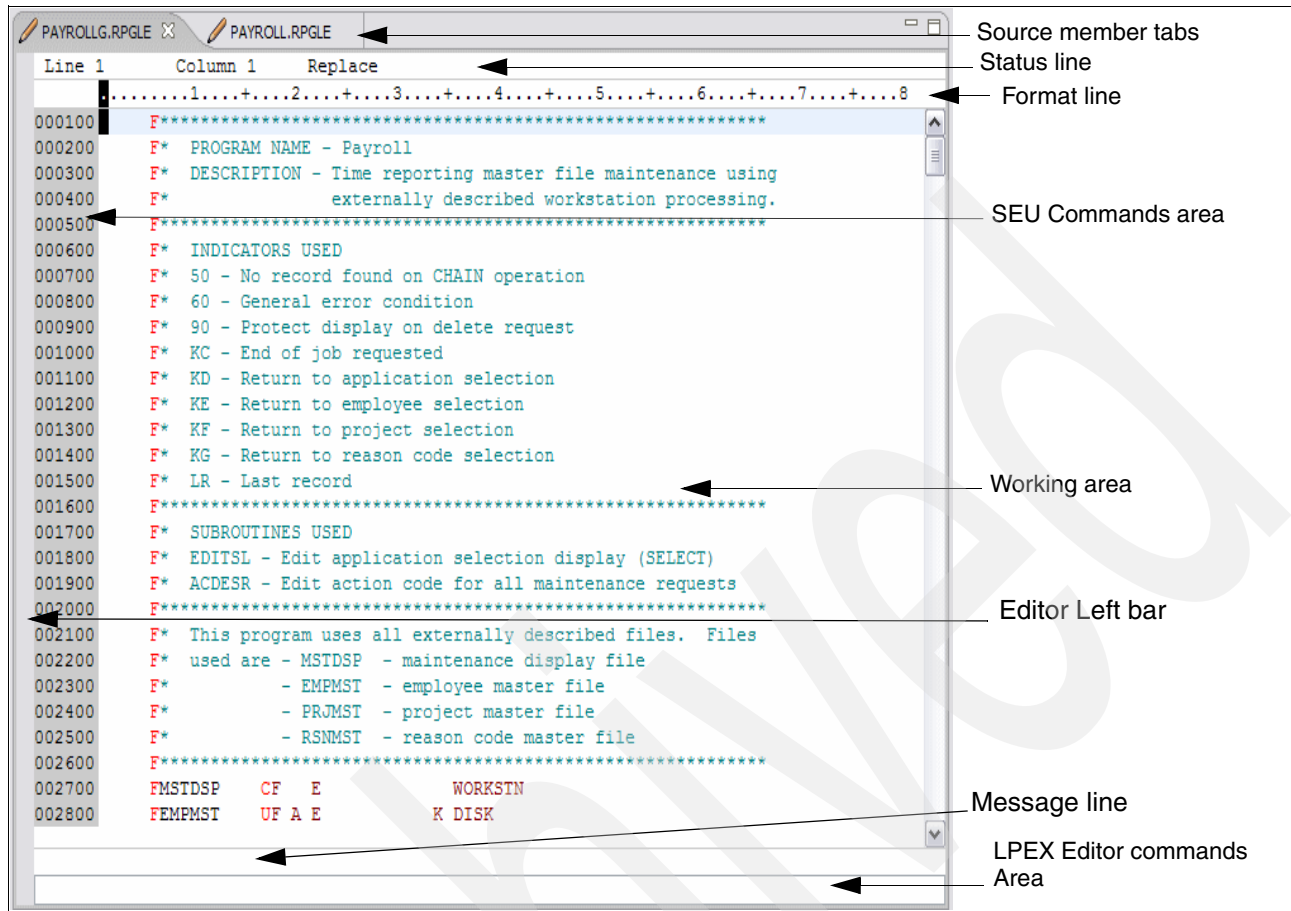


Figure 8-1 Launching the Remote Systems LPEX Editor

In reference to Figure 8-1, the following list highlights some areas of the Remote Systems LPEX Editor:

- ▶ **Source member tabs:** As shown in Figure 8-1, more than one instance of the editor can be launched so that multiple source members can be opened. To edit a particular member, click the source member's tab to place focus on that member.

Tip: You can also use the keyboard shortcut, Ctrl+F6, to switch between editor instances.

- ▶ **Status line:** This line provides information about the current status of the editor. For example, the *Line* and *Column* numbers indicate the current position of the cursor in the editor area. This line also shows which mode the editor is in. When the editor is in *Replace* mode, the cursor appears as a solid block. Any text overlaid by the block is replaced by any new text that you type. When the editor is in *Insert* mode, the cursor appears as a thin vertical line. Any text that you type is inserted into the member at the cursor position, and existing text to the right of the cursor is shifted to the right. You toggle back and forth between the two modes by pressing the Insert key.

Note: When column sensitive editing is enabled it will limit insertion and deletion to the columns specified for the language.

- ▶ **Format line:** This line displays the grid line that shows marked fields based on the format type of the line being edited. Useful for RPG and DDS type source members.
- ▶ **Working area:** You click in this area to enter your text and perform other general editing tasks.
- ▶ **SEU commands area:** This is where SEU commands are entered. These are the commands that most System i developers are familiar with when working with the SEU editor in OS/400 green-screen mode. To execute these commands, you click in the designated area, type the command, and press Enter.
- ▶ **Editor Left bar:** This bar lies on the left side of SEU commands area. You can right-click on this bar, and from the pop-up menu select **Add breakpoint**, **Add Bookmark** or **Add Task**.
- ▶ **Remote Systems LPEX Editor commands area:** You enter editor default or user-defined commands in this area. To execute these commands, you click in the LPEX editor command area, type the command, and press Enter.

For example, when you enter the following command into the command area, two new lines are added to the file:

```
add 2
```

Tip: You can put the cursor at any location where you need more help and press F1.

8.1.2 Working with the Remote Systems LPEX Editor

LPEX is an editor provided with the workbench based on the Eclipse open source project that is not development language specific. However, it can be extended with parsers that provide this functionality. So when you are editing an ILE RPG source member in the workbench, you are really using the LPEX editor *plus* the ILE RPG parser. The parser is what handles the token highlighting, syntax checking, F1 help, content assist, and all other language-specific features. Parsers are automatically used by LPEX based on the file type.

In the preferences dialog you see a section for basic LPEX Editor, **Window** → **Preferences** → **LPEX Editor** → **Parsers** to configure basic editor features as shown in Figure 8-2; and another section for language-specific LPEX Editor, **Window** → **Preferences** → **Remote Systems** → **Remote Systems LPEX Editor** to customize language-specific features, as shown in Figure 8-3.

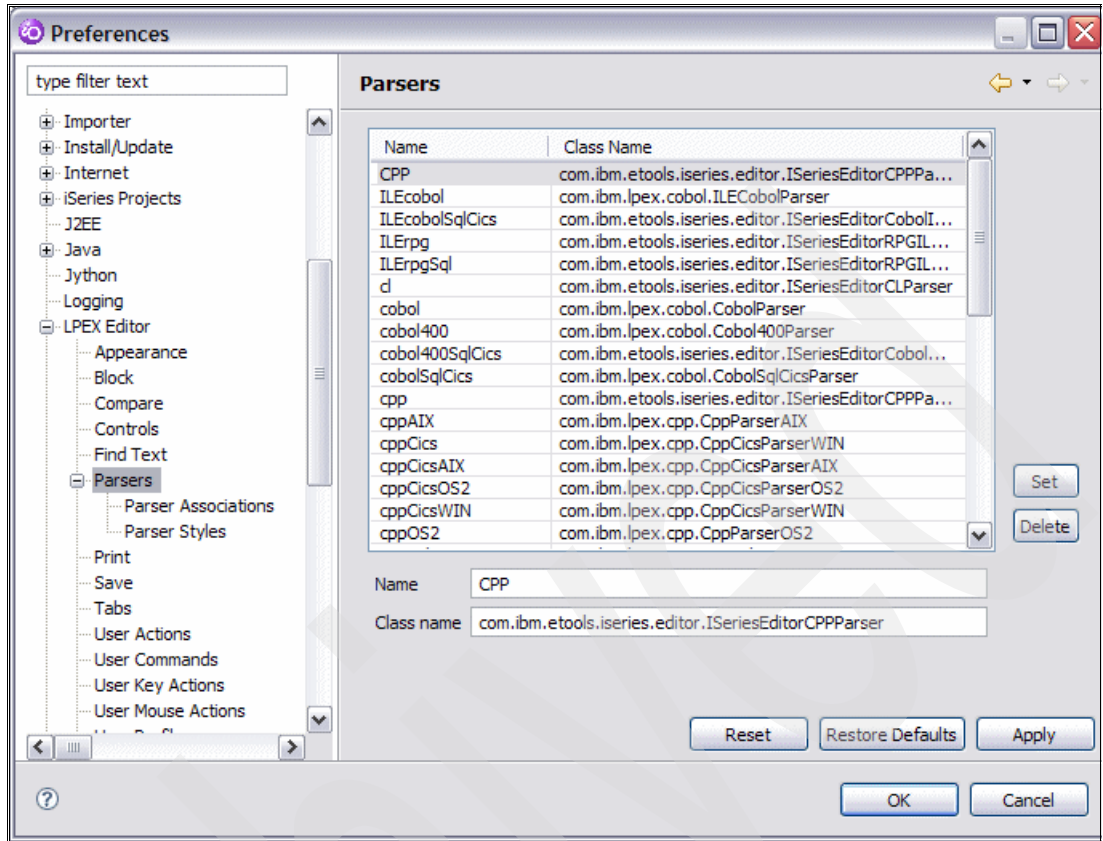


Figure 8-2 Customizing basic editor features

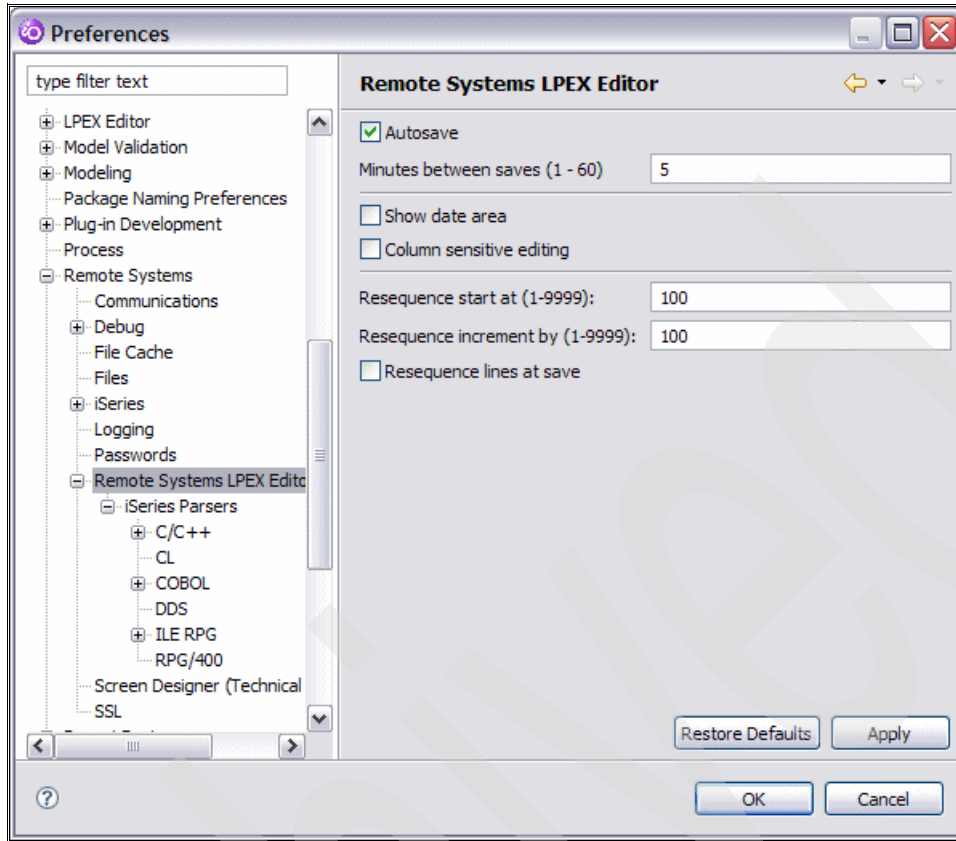


Figure 8-3 Customizing language-specific features

Many basic and advanced functions are available with the Remote Systems LPEX Editor. These include:

- ▶ Cut, copy, and paste operations
- ▶ Block marking of lines, characters, or rectangles with copy, move, and delete operations
- ▶ Powerful find and replace functionality
- ▶ Unlimited undo and redo operations
- ▶ Token highlighting: Different language constructs are highlighted using different colors and fonts.
- ▶ SEU-like format line rulers to show the purpose of each column for column sensitive languages like RPG and DDS: These rulers can automatically update themselves to reflect the current specification.
- ▶ SEU-like source line prompting for RPG and DDS source members
- ▶ Sequence numbers that allow SEU-style commands in the prefix area
- ▶ Intelligent tabbing between columns for column-sensitive languages
- ▶ Automatic uppercase for languages that expect uppercase
- ▶ Commands to simplify text insertions and deletions for column-sensitive languages
- ▶ Filtered views to display specific source member content in the editor
- ▶ Indentation display of control structures for column-sensitive languages
- ▶ A show fields function for viewing field details of any database, workstation or printer files referenced in the source member

- ▶ Source verification
- ▶ Online language reference help

Syntax checking for RPG, Cobol and CL You can find more information about these functions in the online help. From the workbench menu, follow these steps:

1. Click **Help** → **Help Contents**.
2. In the Help application window, expand **WebSphere Development Studio Client for System i** → **Developing System i server applications using Remote System Explorer** → **Editing, compiling, and debugging applications in the Remote System Explorer** → **Editing** → **LPEX Editor** → **Working with text**.

You can customize the Remote Systems LPEX Editor to your requirements. You can change Tab settings, tailor keyboard mappings to your requirements, and modify the general appearance and layout of the editor. For information about editor customization, see the online help. From the workbench menu, follow these steps:

1. Click **Help** → **Help Contents**.
2. In the Help application window, expand **WebSphere Development Studio Client for System i** → **Developing System i server applications using Remote System Explorer** → **Editing, compiling, and debugging applications in the Remote System Explorer** → **Editing** → **LPEX Editor** → **Customizing the editor**.

You can also issue editor-specific commands in the commands area of the editor. For more information about LPEX editor commands and how to issue them, see the online help. From the workbench menu, follow these steps:

1. Select **Help** → **Help Contents**.
2. In the Help application window, choose either of the following actions:
 - Use these steps to reach the default editor commands help:
 - i. Expand **Developing System i server applications using Remote System Explorer**.
 - ii. Expand **Editing, compiling, and debugging applications in the Remote System Explorer**.
 - iii. Expand **Editing**.
 - iv. Expand **LPEX Editor**.
 - v. Expand **Default editor commands**.
 - Or you can follow these steps to reach the issuing editor commands help:
 - i. Expand **Developing System i server applications using Remote System Explorer**.
 - ii. Expand **Editing, compiling, and debugging applications in the Remote System Explorer**.
 - iii. Expand **Editing**.
 - iv. Expand **LPEX Editor**.
 - v. Expand **Working with text**.
 - vi. Expand **Issuing editor commands**.

8.1.3 Remote Systems LPEX Editor functions

This section discusses and demonstrates a cross-section of the functions that are available with the Remote Systems LPEX Editor.

Source line prompting

Source line prompting assists you in formatting your source code lines correctly. This function corresponds to the F4 prompting function within the SEU editor. This feature is available for RPG, DDS, and CL source members.

To prompt a source line in the editor, follow these steps:

1. Open the source member in the editor. See 8.1.1, “Launching the Remote Systems LPEX Editor” on page 250.
2. Click the source line you want to prompt.
3. Right-click and select **Prompt** or simply press F4.
4. The prompt window displays as the *System i Source Prompter* view. This view is usually part of a tabbed notebook of views. See Figure 8-4.

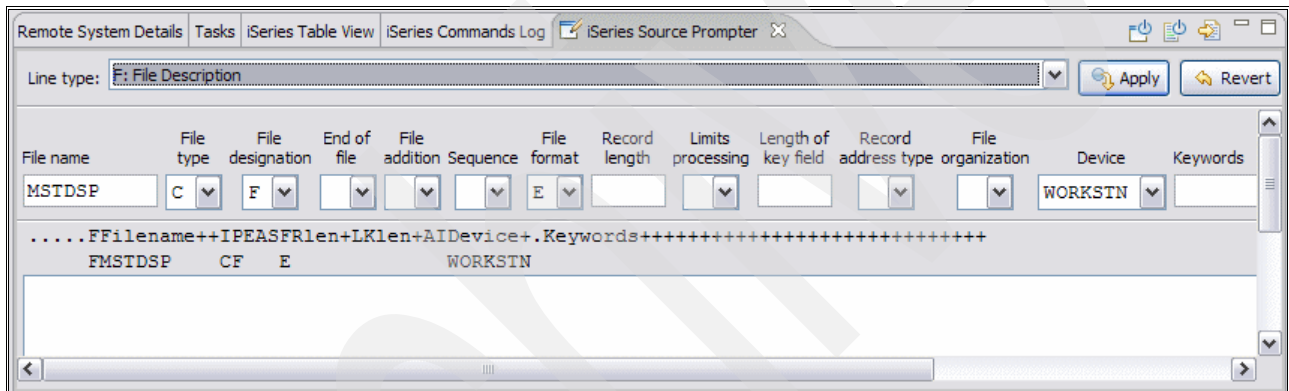


Figure 8-4 The System i Source Prompter view

5. Enter, or select from the drop-down lists, appropriate values for the fields in the prompted line and press Enter.

After the System i Source Prompter view is active, the data within this view changes to reflect the source line the cursor is currently positioned on.

The prompt window is different for CL type members. When you prompt a line in a CL source member, a prompt for the CL command displays whereby you can enter details for the command and then click OK to return to the editor.

Tip: In older editions of WDSC, you can invoke the System i Source Prompter via the keyboard shortcut Shift+F4. But now, you can invoke it simply via F4.

Syntax checking

As the title of this section suggests, this function checks your source code for correct syntax based on the source member type. You can perform syntax checking by:

- ▶ Syntax check a source line
- ▶ Syntax check a member
- ▶ Syntax check selection

To syntax check a source line, follow these steps:

1. Open the source member in the editor. See 8.1.1, “Launching the Remote Systems LPEX Editor” on page 250.
2. Click the source line you want to prompt.
3. Right-click and select **Syntax Check Line**.
4. If there is an error within the source line, a message is displayed directly below the offending line. See Figure 8-5.

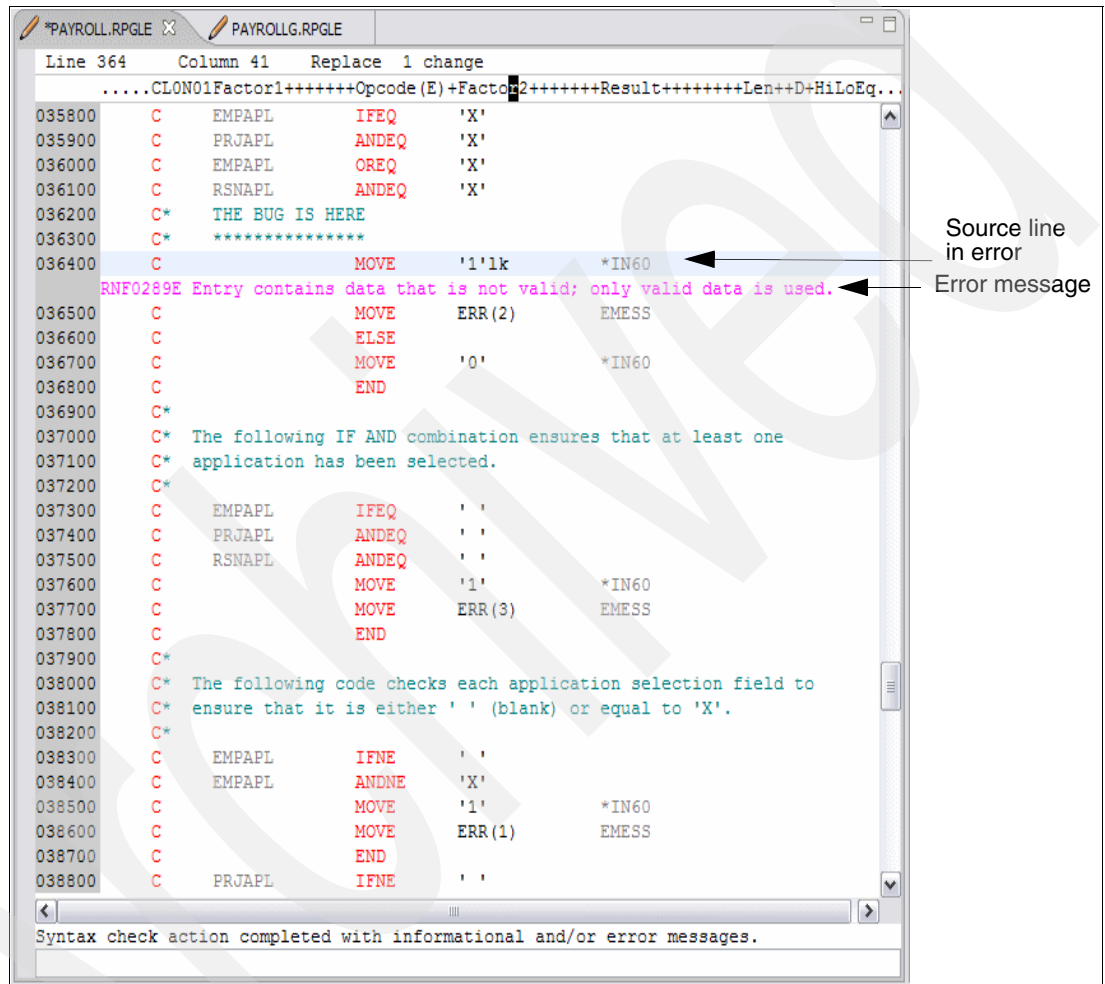


Figure 8-5 Syntax checking a source line

5. Position the cursor on the line that has the error and make the necessary correction. When you move to the next line, the previous line is checked again for errors and the error message is removed from the editor working area if there were no errors in the line.

To syntax check the member, place focus on the member by clicking its tab. From the workbench menu, select **Source** → **Syntax Check All**.

Tip: The syntax checking function is invoked *automatically* by default for RPG, COBOL, DDS, and CL source members when you move the cursor off a source line. To disable the auto-checking feature, follow these steps:

1. From the workbench menu, select **Window** → **Preferences**.
2. In the left side pane of the **Preferences** window, expand **Remote Systems** → **Remote System LPEX Editor** → **System i Parsers**.
3. Select the source member type for which you want to turn this feature off.

In the right side pane, deselect the **Automatic syntax checking** check box.

Filter views

A filter view is a useful function that allows you to specify which sections of the source member you see in the editor. The filter criteria changes depending on the type of member being edited.

In the following scenario, we specify that the filter view only displays, in the editor, subroutines defined in an RPG member.

To specify a filter view, follow these steps:

1. Open the RPG source member in the editor. See 8.1.1, “Launching the Remote Systems LPEX Editor” on page 250.
2. Right-click anywhere in the editor working area and select **Filter view** → **Subroutines**.
3. The view in the editor changes, displaying only subroutine names within the source member. You can expand each subroutine by clicking the plus (+) sign to the left of the source line sequence number to reveal the source contained within the subroutine. See Figure 8-6.

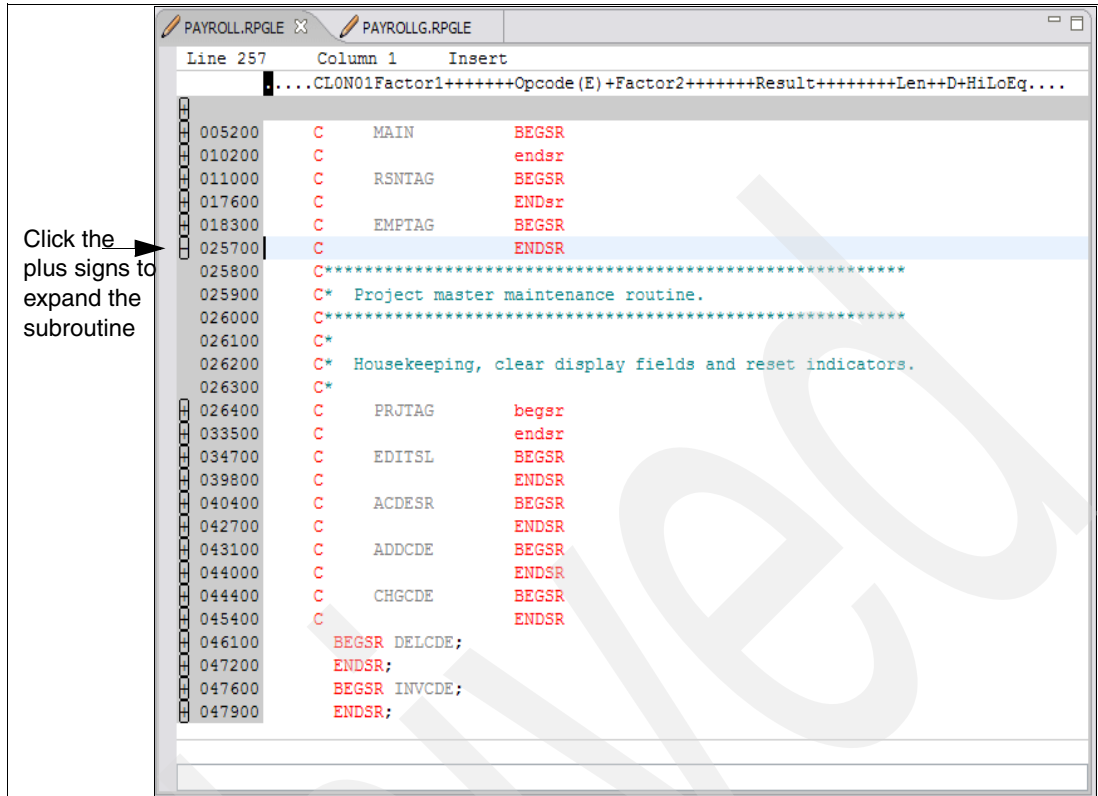


Figure 8-6 Filter view of subroutines in an RPG member

- To revert back to the view that displays all the source lines of the member, right-click in the editor working area and select **Show all**.

The list of filter criteria changes depending on the member opened in the Editor. This example used the filter criteria “subroutines” on a SQLRPGLE member. You can also filter on other items contained within the source member, such as:

- ▶ **By Date** - You may choose a date or a pair of dates to show only lines which have been changed before, after or between those dates.
- ▶ **Code** - All comment lines are hidden.
- ▶ **Comments** - Only comment lines are shown.
- ▶ **Control** - Only flow control statements, such as If, Do and so on, are shown.
- ▶ **Procedures** - Only procedures are shown.
- ▶ **SQL statements** - Only embedded SQL is shown.
- ▶ **Errors** - Only error messages from the last compile or verify are shown.

Show fields

Show fields is another useful function that allows you to view field details of database, workstation, and printer files that are defined in RPG or CL source members. In this section, we invoke this function on an RPG source member.

To use the Show fields function, follow these steps:

- Open the RPG source member in the editor. See 8.1.1, “Launching the Remote Systems LPEX Editor” on page 250.

- Click a file specification line in the RPG member. Note: to be able to show the fields of an externally described file, you need to make sure that the library containing that file is part of the library list.

Tip: To quickly add a library to the library list of your RSE server job, right-click the library in the RSE view and select **Add to Library List ...** from the context menu. You can add the library name at the start or at the end of the library list, or before or after replacing another name you can specify.

- Right-click and select **Show fields**.
- The System i Field Table View opens and displays all fields defined in the file. Field details are displayed as rows in the table view. This view is usually part of a tabbed notebook of views. See Figure 8-7.

The screenshot shows a window titled 'iSeries Field Table View' with a tabbed interface. The active tab is 'iSeries Field Table View' and the file being viewed is 'File WFLABXX/ORDENTD (47 Fields)'. The table below lists the fields:

Name	Record	Type	Length	Text	Alias
*IN03	PROMPT	Character	1		
*IN04	PROMPT	Character	1		
*IN12	PROMPT	Character	1		
*IN80	PROMPT	Character	1		
CID	PROMPT	Character	4		
ORDNBR	PROMPT	Zoned Decimal	5.0		
*IN04	ORDSFL	Character	1		
*IN12	ORDSFL	Character	1		

Figure 8-7 The System i Field Table View

- To view the field's properties, right-click the field in the table and select **Field Properties**. The Field Properties window (Figure 8-8) opens to display further attributes associated with the field.

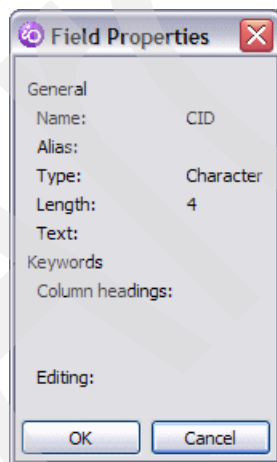


Figure 8-8 The Field Properties window

Online help

Two types of online help are available:

- ▶ Context-sensitive help
- ▶ Content assist

Context-sensitive help

To invoke context-sensitive help, click the item for which you need help and press the F1 key. The help documentation opens in a new window and you are taken to the section of the Help that is associated with the item you requested help for.

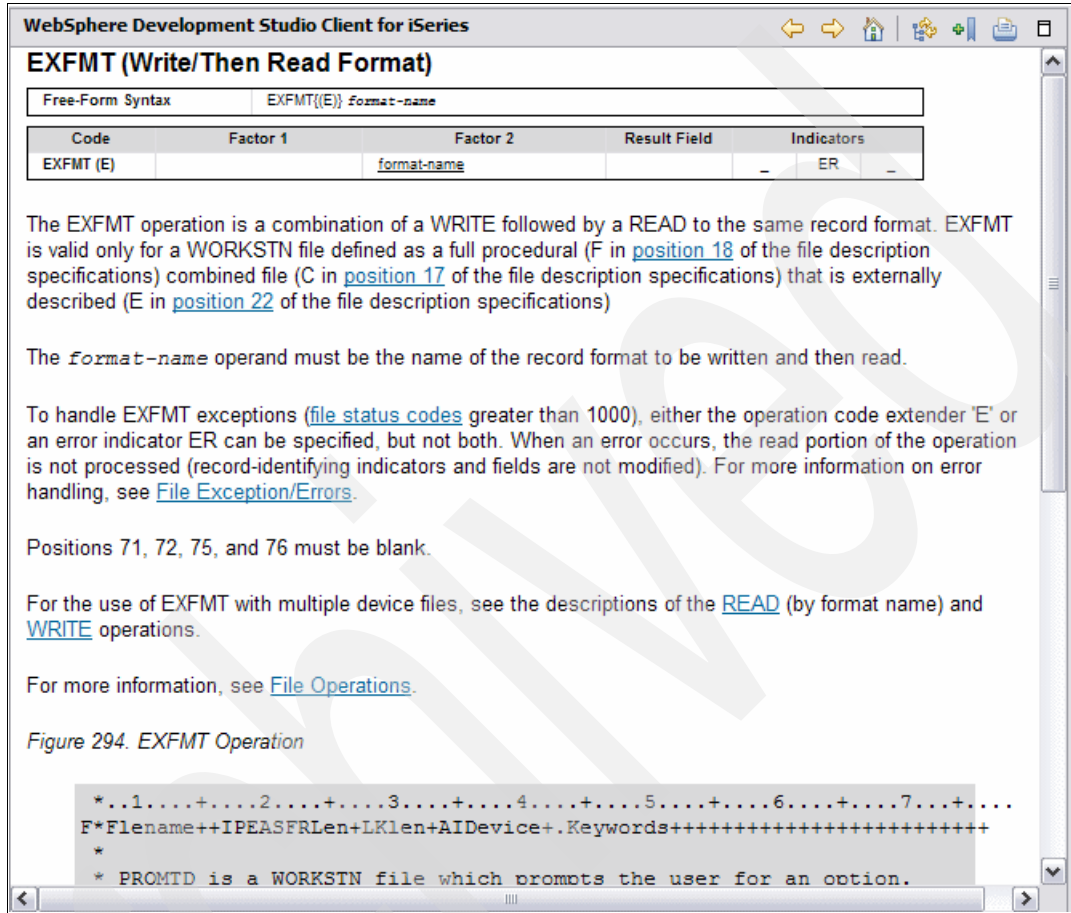


Figure 8-9 Context-sensitive help invoked for RPG opcode EXFMT

For example, clicking the EXFMT opcode in an RPG member and pressing F1 opens the help documentation to the page with details on this opcode. Figure 8-9 shows an example of context-sensitive help invoked for the EXFMT opcode.

Content assist

Content assist is a “short form” version of help. It can be used with RPG source members. This function can be used for opcodes, built-in functions, and user-defined tokens such as fields and subroutines.

To invoke content assist, click the token you need assistance with and then follow these steps:

1. Either press Ctrl+Spacebar, or from the workbench menu, select **Edit** → **Content Assist**.
2. A scrollable list of keywords that match the first letter of the token you selected is displayed. Select the keyword of interest to you. Then a description of the token is displayed in an adjoining box. Figure 8-10 shows an example of content assist invoked for the EXFMT opcode.
3. To return to the editor, either press the Esc key, or click in the editor working area.

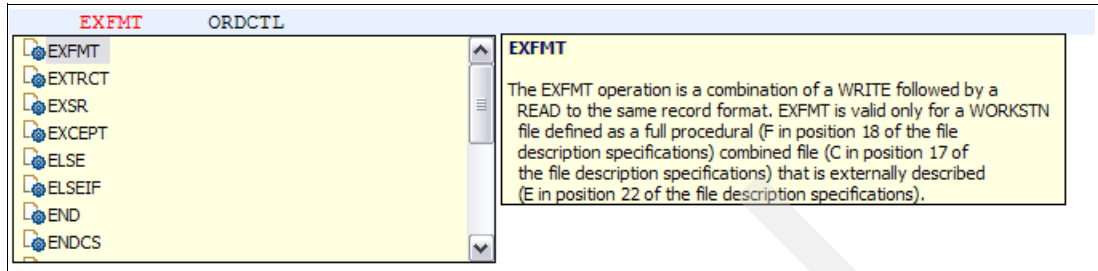


Figure 8-10 Content assist invoked for RPG opcode EXFMT

Content assist can also be used to insert entries into your source member:

1. Type one or more characters of the item you want to insert into the source member.
2. Press Ctrl+Spacebar, or from the workbench menu, select **Edit** → **Content Assist**.
3. From the scrollable list, select the item you want to insert.
4. Press Enter to insert the item.

Note that code assist not only helps you find the correct keywords according to the language syntax but also *user-defined tokens*, such as the names of variables, subroutines, and files which would be allowed to enter at this point. Figure 8-11 shows such an example for the ILE COBOL language, which is also supported for ILE RPG, but not for CL or non-ILE languages sources. For user-defined tokens, you *must* refresh the Outline view. For more information about the Outline view, see 8.1.4, “Outline views” on page 273.



Figure 8-11 Content assist for COBOL

Figure 8-11 shows content assist for user-defined tokens in ILE COBOL. Remember, you must refresh the Outline view in order to see user-defined tokens.

Source verify

The *source verify* function provides a full range of syntactic and semantic checking for your code, similar to a compiler of your source code, without creating a compiled object on the

System i host. Source verification can be performed on RPG, COBOL, and DDS source members. Verification can also be done offline.

To invoke the verify function, follow these steps:

1. Drill down through the Remote Systems view or System i Table view to find the source member you want to verify.
2. Right-click the source member and select either **Verify** or **Verify (Prompt)**. The difference Between Verify and Verify (Prompt) is that in Verify (Prompt) you are prompted to enter parameters for the verification process. See Figure 8-12. You can also select **Source** from the workbench menu and select **Verify** or **Verify With Prompt**. To invoke the verifier in this manner, the source member *must* already be opened in the Remote Systems LPEX Editor.

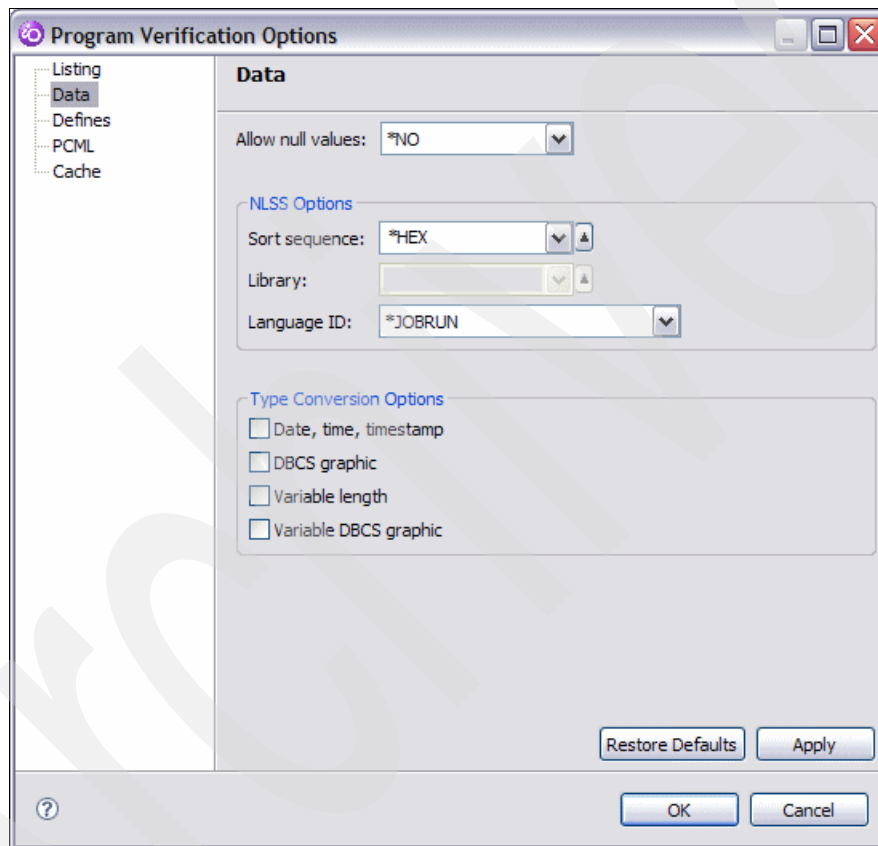


Figure 8-12 The Program verifications options window prompting for RPG


3. The source verifier launches and runs. Any errors found by the verifier are reported in the System i Error List view. Details, such as message ID, message text, severity, and source line number, are displayed for each error found. This view usually appears as part of a tabbed notebook of views similar to the one in Figure 8-13.

ID	Message	Severity	Line	Location	Connection
RNF0401	An error occurred during communication with the host.	50	27	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF2120	External descriptions for file MSTDSP not found; file is ignored.	40	27	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator ACODE is not defined.	30	114	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator EMPAPL is not defined.	30	56	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator EMPMNT is not defined.	30	220	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator EMPSEL is not defined.	30	200	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator PRJAPL is not defined.	30	57	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator PRJMNT is not defined.	30	297	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator PRJSEL is not defined.	30	280	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator RSNAPL is not defined.	30	58	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection
RNF7030	The name or indicator RSNMNT is not defined.	30	139	RSELABXX/QRPGLESRC(PAYROLL)	MyConnection

Figure 8-13 The System i Error List after verification errors

4. Double-click an error in the System i Error List view. The source member opens in the editor with the cursor positioned at the source line in error. An error message is also displayed directly beneath the offending source line; correct the error.
5. Repeat the previous step for each error in the System i Error List view.

Note: You can also invoke the source verifier using the keyboard shortcut Ctrl+Shift+V.

The Error List view also filters what errors you want to see. Click the drop-down arrow  near the top right-hand corner of the view, and choose **Show Severity**, and then choose the severity type. For example, if you choose **Show severity** → **Severe**, only messages with severity of type Severe would display.

You can also choose how double-clicking on error messages gets inserted in the corresponding editor by selecting **Messages**, and then selecting one of the options: **Insert None**, **Insert Selected Only**, **Insert All Showing**, or **Insert All**. **Insert All Showing** inserts only the messages that appear in the error list view. **Insert All** inserts all messages in the events file, even if the messages have been filtered out due to their severity.

Global verifier preferences

Preferences for the source verifier can be set globally. In this example, we set a global preference that instructs the verifier to generate a listing after verification is completed. We set this preference for RPGLE source members.

1. From the workbench menu, select **Window** → **Preferences**.
2. In the left side pane of the Preferences window, expand **Remote Systems** → **System i** → **Program Verifiers** → **ILE RPG**.
3. In the right side pane of the Preferences window, select the **Generate listing** box.

Now, when a source verification is invoked on an RPGLE source member, a source listing is generated in the System i Listings view as shown in Figure 8-14.

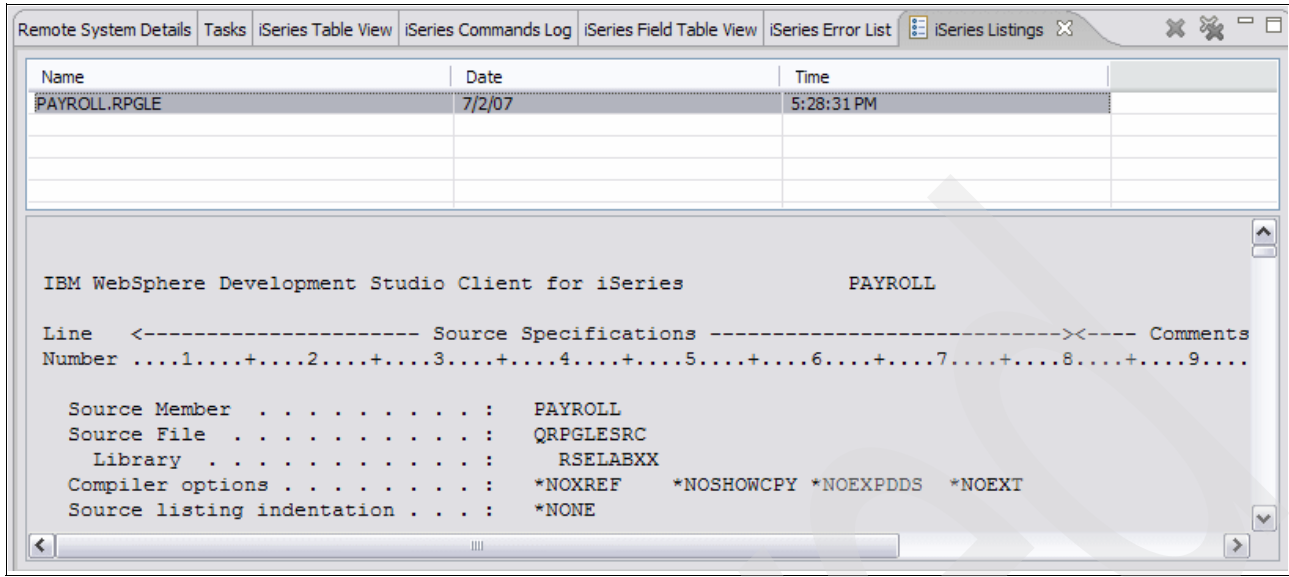


Figure 8-14 Source listing in the System i Listings view

To override any global settings temporarily, select the **Verify (Prompt)** option.

Show indentation

The *show indentation* function applies to RPG source members. Invoking this function shows an indented view of any RPG language control structures such as IF, DOW, and FOR. This feature is useful for analyzing fixed format RPG source members that contain deeply nested control structure code.

To use the show indentation function, follow these steps:

1. Open the RPG source member in the editor. See 8.1.1, “Launching the Remote Systems LPEX Editor” on page 250.
2. From the workbench menu, select **Source** → **Show Indentation**.

An indented view of the source member is displayed in the System i Indent view. See Figure 8-15.

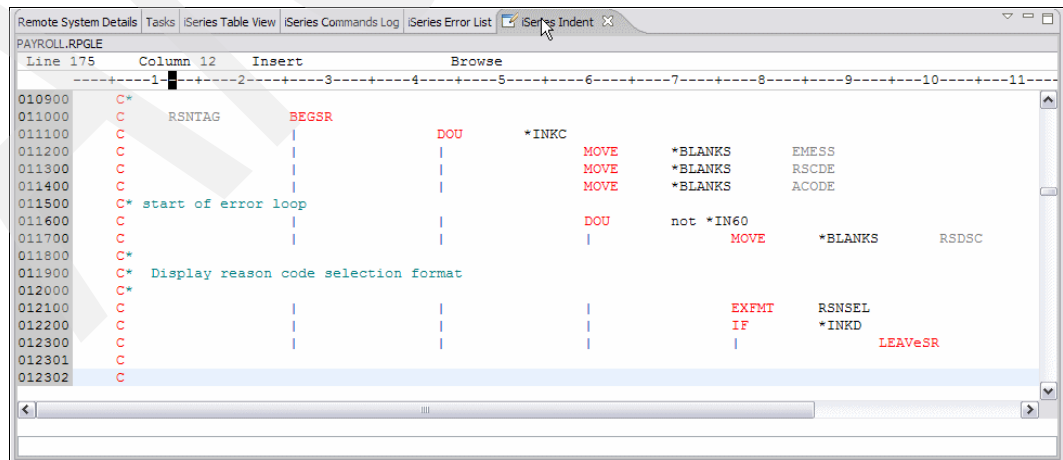


Figure 8-15 The System i Indent view

Block Nesting

You can use Block Nesting using Ctrl+O, or from the Source menu in the pop-up menu, you can display an arrow indicating the nesting level at the cursor location, as shown in Figure 8-16.

```
Line 456      Column 23      Replace
.....CLON01Factor1+++++OpCode (E)+Factor2+++++Result+++++Len++D+HiLoEq..
043800      C                      ENDF
043900      C                      ENDF
044000      C                      ENDSR
044100      C*
044200      C*   The following code verifies the change request.
044300      C*
044400      C   CHGCDE          BEGSR
044500      C                      IF          *IN50
044600      C                      EVAL      *IN60 = *ON
044700      C                      MOVE     ERR(7)      EMESS
044800      C                      ELSE
044900      C                      IF          NOT *IN50 AND ACREC = 'D'
045000      C                      EVAL      *IN60 = *ON
045100      C                      MOVE     ERR(8)      EMESS
045200      C                      ENDF
045300      C                      ENDF
045400      C                      ENDSR
045500      C   /FREE
045600      C   /**
045700      C   /**   The following code verifies the delete request.  The field
045800      C   /**   protect indicator *IN90 is first set on (equal to 1) to not
045900      C   /**   allow changes to existing data on a delete request.
```

Figure 8-16 Show Block Nesting

Displaying and editing /COPY members and copy books

This function enables you to open any members that are referenced by RPG's /COPY compiler directive and COBOL's COPY compiler directive. The referenced members are opened in the Remote Systems LPEX Editor with their own tab.

To invoke this function on an RPG member, follow these steps:

1. Open the RPG source member in the editor. See 8.1.1, "Launching the Remote Systems LPEX Editor" on page 250.
2. Position the cursor on a source line that contains the /COPY directive.
3. Perform either of the following steps:
 - Right-click and select **/COPY Member** → **Edit** to open the member in the editor in change mode.
 - Right-click and select **/COPY Member** → **Browse** to open the member in the editor in browse mode.

For COBOL members, select **Copy Book** and then **Edit** or **Browse**.

Saving your changes

When you edit a source member, the number of changes that you make during your editing session are recorded in the status line. The changes you make are temporary until you decide to save them. Members with unsaved changes are denoted by an asterisk (*) next to the source member's name tab. See Figure 8-17.

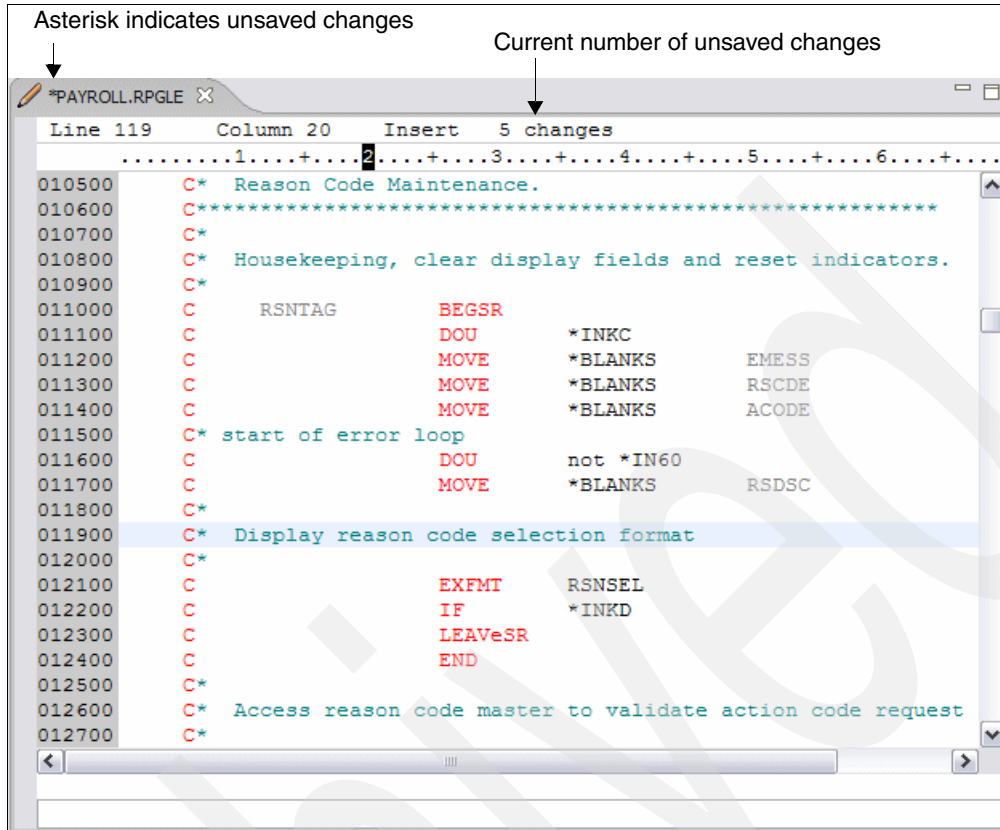


Figure 8-17 Unsaved changes in the editor


When you save your changes, both the asterisk and number of changes indicator are cleared from the window.

To save changes made to your source member, perform one of the following steps:

- ▶ Right-click in the editor work area and select **Save**.
- ▶ From the workbench menu, select **File** → **Save**.
- ▶ Press Ctrl+S.

If multiple editor instances are open, you can save all the members in one action. To do this, you can select **File** → **Save All** from the workbench menu, or simply press Ctrl+Shift+S.

Autosave while editing

To enable or disable autosave while editing in the Remote Systems LPEX editor, you need to change an editor preference. From the Remote Systems view, click the drop-down menu  and select **Preferences** → **Remote Systems LPEX Editor**. In the preference window, select the Autosave check box. The default value for Minutes between saves is set to 5. You can specify a value between 1 and 60 minutes. If the workbench closes unexpectedly while you are editing a file, restart the workbench and reopen the file you were editing. If there is a backup of that file, you will be prompted to open it for editing.

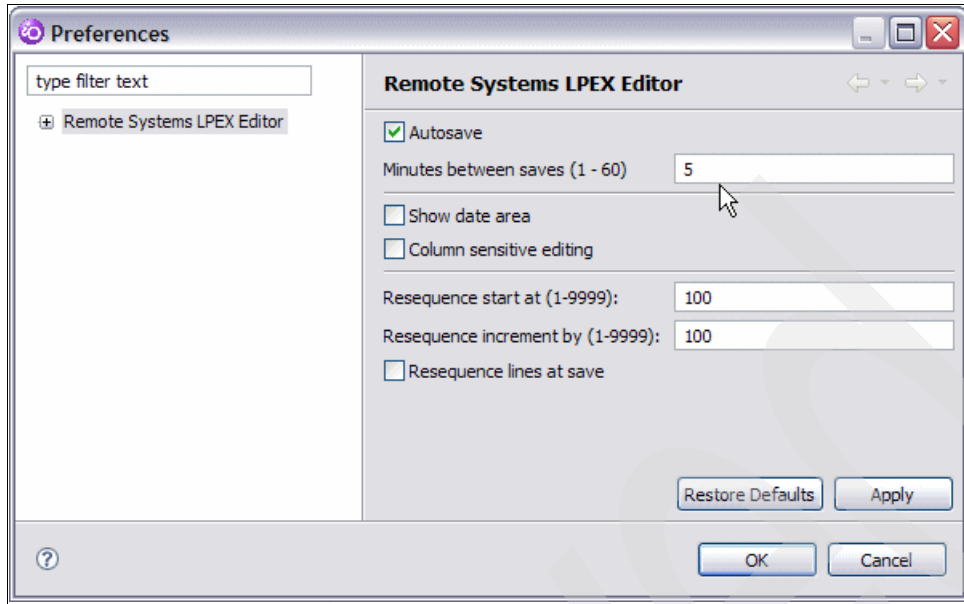



Figure 8-18 LPEX Editor autosave

Compare a file in the Remote Systems LPEX Editor

Comparing (PDM option 54) in Remote Systems LPEX Editor requires you to open a member in the editor. Once open, you can easily compare that member to another by selecting the Compare button in the toolbar  or through the **Edit** → **Compare to a file** workbench menu option.

Once the compare has been triggered, source appears merged with different lines flagged in color. Pink is the color used for the source being compared to, and yellow is the color for the opened source as shown in Figure 8-19.

Unlike on System i, where you have to flip between the spooled file and the source opened in SEU, comparing in Remote Systems LPEX Editor will allow you to continue to modify the member that was opened originally. Use **Ctrl+Shift+N** to navigate to the next mismatch and **Ctrl+Shift+P** to the previous mismatch. If you do modify source, you can use **Ctrl+Shift+R** to refresh the compare and finally right-click on the file opened in the Editor and from the pop-up menu select **Compare** → **Clear**.

You can also specify additional preferences in **Window** → **Preferences** → **LPEX Editor** → **Compare**.

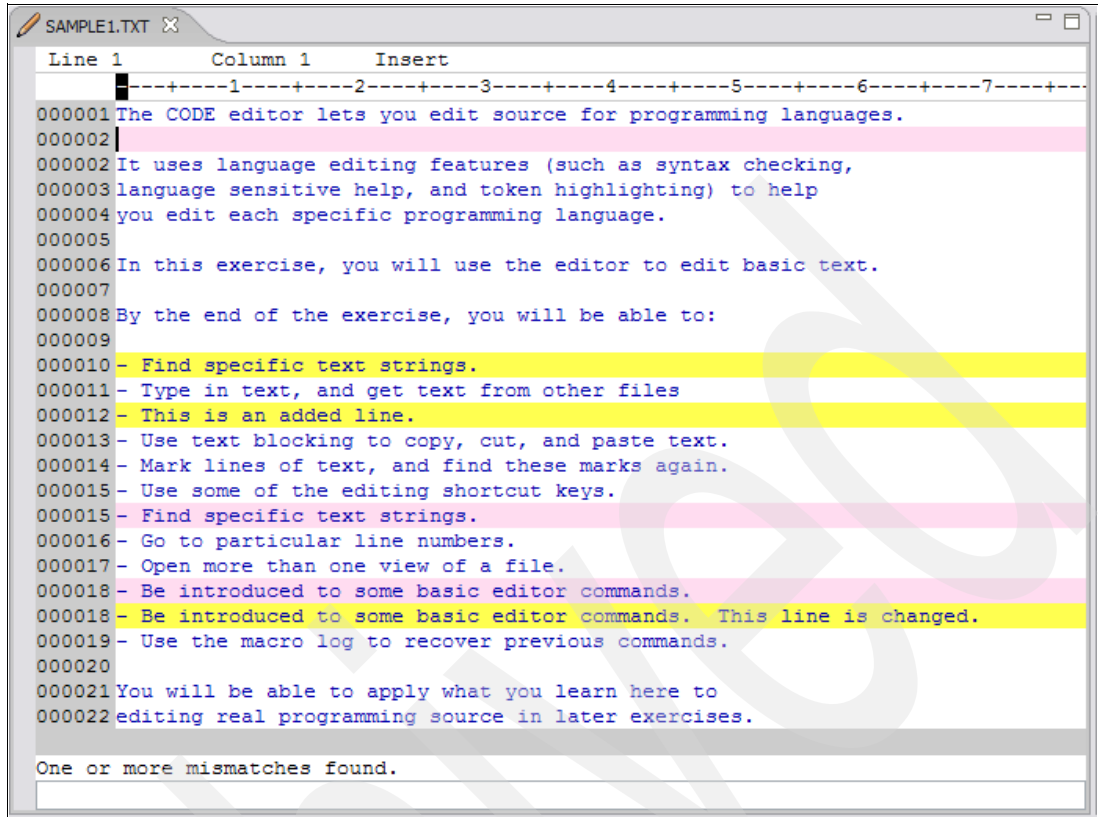


Figure 8-19 Compare files in LPEX editor

Note: Unlike other compare tools in eclipse, Remote Systems LPEX Editor is sequence number aware, and will not mismatch lines just because the sequence number has been modified.

Pop-up menu has a Compare submenu when in an active compare session, to navigate, clear, and toggle compare settings for the current view, as shown in Figure 8-20.

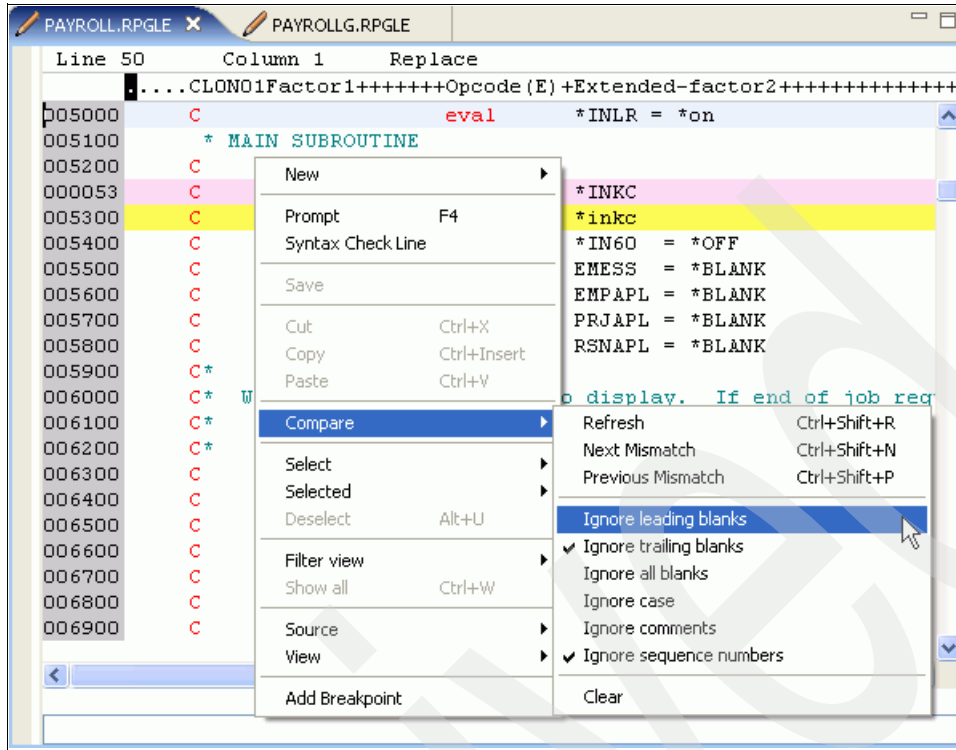


Figure 8-20 Member compare submenu

Opening the same member in multiple Editor views

You can open the same member in multiple LPEX Editor views. This will enable you to view different parts of the file at the same time. The changes made to the file in one of the views are automatically reflected into the other opened view.

To open a member in a new view:

1. Drill down through your filters in the Remote Systems view until you find the source member. Double-click on the member to open it in LPEX Editor.
2. Right-click in the source member and from the pop-up menu select **View** → **Open new view** as shown in Figure 8-21.

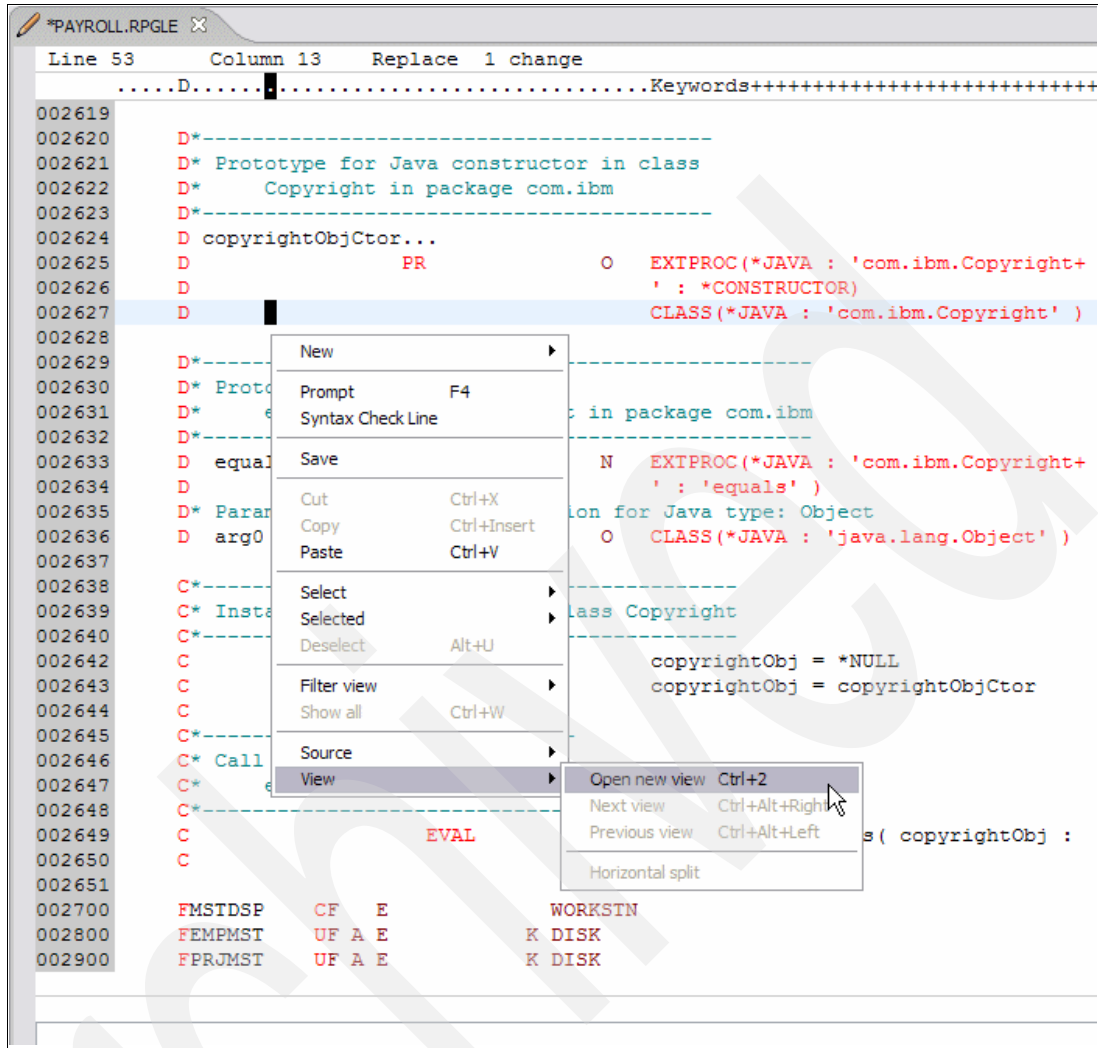


Figure 8-21 Open new LPEX editor view

3. The member is opened in a new view. You can work with the member in both views, as shown in Figure 8-22.

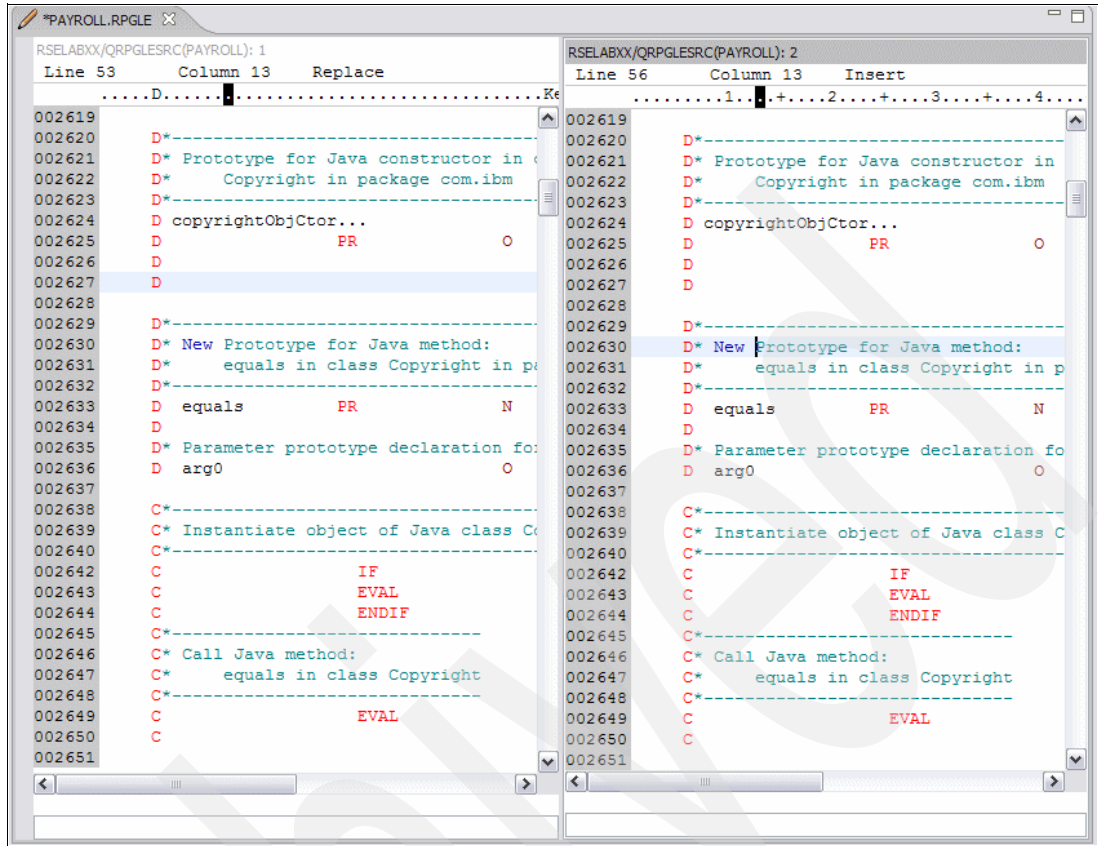


Figure 8-22 Editing member in multiple views

4. You can also Open a new view, browse to the next or previous view, split views horizontally and close view by right-clicking in the editor area and choosing the appropriate function from the view submenu. Figure 8-23 shows the Close view function. You can use this function to close all the opened views except the primary view.

Note: You can open up to five views to the same source member in the LPEX Editor.

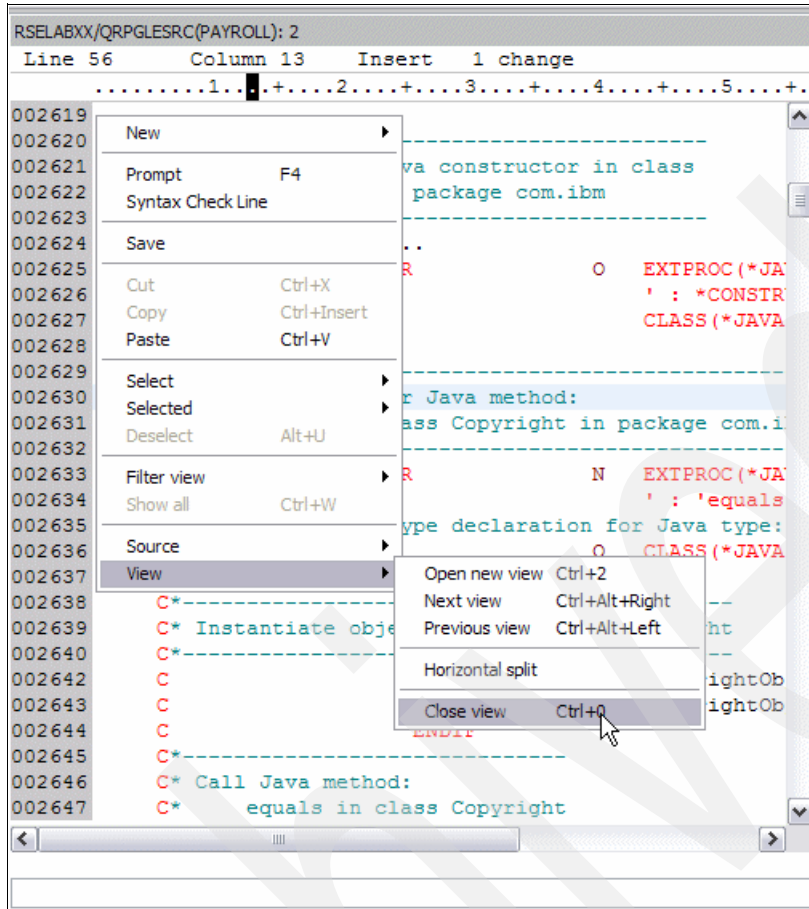



Figure 8-23 Close the LPEX Editor view

Closing editors

To close an editor, perform one of the following actions:

- ▶ Click the **Close** () button in the member tab of any open editor.
- ▶ From the workbench menu, select **File** → **Close**.
- ▶ Press Ctrl+F4.

If multiple editor instances are open, you can close all instances in one action. To do this, select **File** → **Close All** from the workbench menu, or simply press Ctrl+Shift+F4.

8.1.4 Outline views

Although Outline views are not editing functions, they prove most useful when used in conjunction with the editor. Outline views display the outline of a structured member that is currently open in the editor area and list the structural elements of the member.

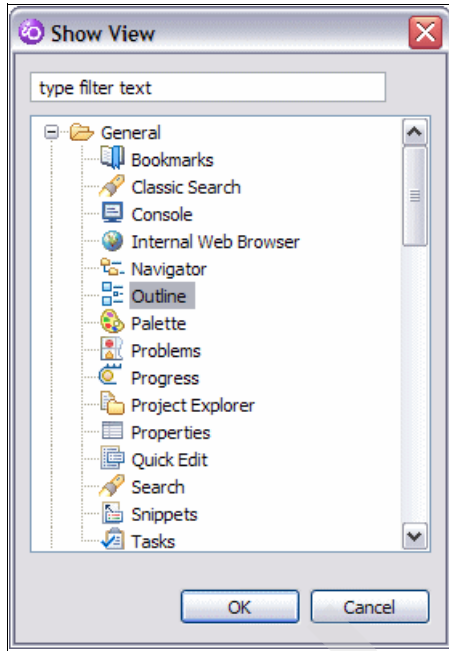


Figure 8-24 Show View window

These views are applicable to RPG and COBOL source members, as well as other types of sources in other perspectives, such as Java, HTML, Cascading Styles Sheets (CSS). By default, when you open a source member in the editor, the Outline view appears as a separate view in the RSE perspective. If the member type supports Outline view, then an Outline view is shown for the member. Otherwise, the view is empty.

If the Outline view does not display because it was closed from a previous workbench session, for example, you can restore the original appearance of the Remote Systems Perspective. To do so, select **Window** → **Reset Perspective**.

If you just want to open the Outline view without changing the size and position of the other panes, select **Windows** → **Show View** → **Other**. This will open the Show View window as shown in Figure 8-24. Expand **General** and choose **Outline**, then click **OK**. You can also open the Outline view in any other perspective, which does by default not show an outline, such as the WebFacing perspective.

The Outline view usually opens besides the editor area, but this may vary depending on your workbench preferences. Figure 8-25 shows the Outline view for an RPG source member.

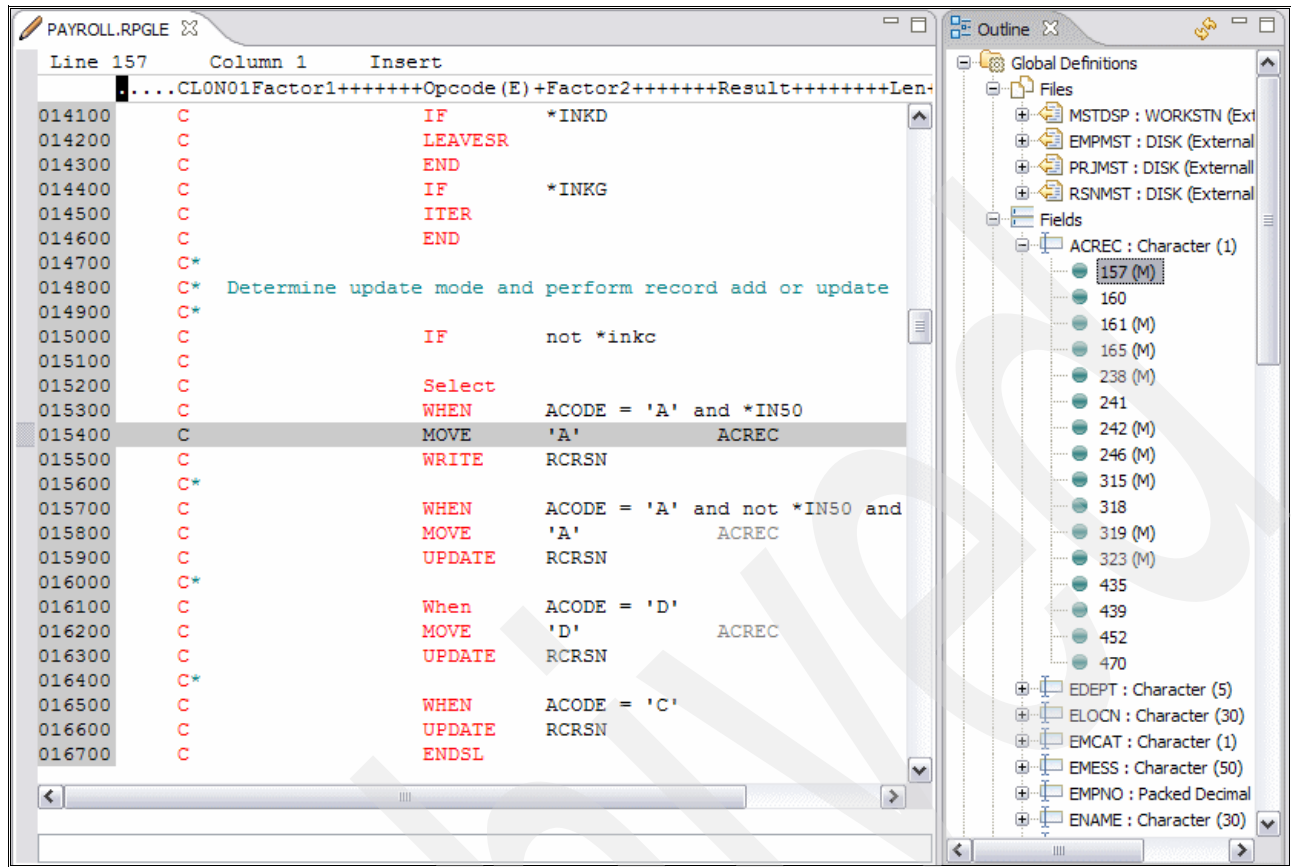


Figure 8-25 Editor pane and Outline view for an RPG member

In Figure 8-25, the Outline view represents all the structural elements of the RPG source member in an easy-to-navigate tree. Expanding any of the elements reveals further child elements contained within.

For example, expanding a file element reveals a record format element. Expanding this element displays all the fields defined within the record format. Clicking any of the elements also positions the cursor to the definition of that element within the source member if applicable.

The outline view not only shows where each element is defined, but also where it is referenced. Again, clicking one of the circles ● in the outline highlights the corresponding line in the editor pane, as indicated in Figure 8-24.

Note that the line numbers shown at the right of the icon ● are different from the sequence numbers shown on the left of the highlighted line in the Editor pane. This happens if you inserted or deleted lines and decided not to resequence the line numbers under **Window** → **Preferences** → **Remote Systems** → **Remote Systems LPEX Editor** and deselected the “Resequence lines at save” check box.

To refresh the contents of the tree in the Outline view at any time during an editing session, click **Refresh** (🔄) in the toolbar of the Outline view.

External description is not available

If your source member contains externally described files or data structures, you might see the text External description is not available when you expand a file in the outline view.

Very likely, this is caused by the fact that your RSE server job does not have the appropriate library name for that file in its library list, as shown in Figure 8-26.

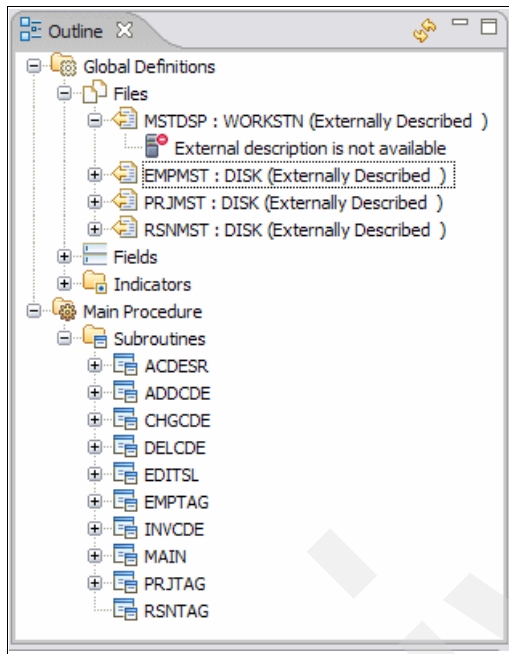


Figure 8-26 External description is not available

To quickly add that library to the library list, right-click the library in the RSE view and select **Add to Library List ...** from the context menu. Then click **Refresh** (🔄) in the toolbar of the Outline view.

Using the Fast View

If you prefer to edit your source member in a maximized editor pane (by double-clicking the tab on top of the frame), you can still use the Remote Systems view. The trick is, to show it in a *Fast View*. To do so, right-click the bar on top of the Outline view and select **Fast View** from the context menu, as shown in Figure 8-27.

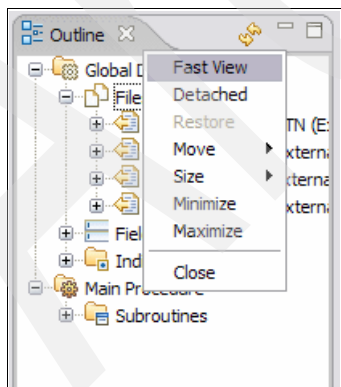


Figure 8-27 Selecting Fast View

The result is that the Outline view disappears and the icon representing it now appears on the shortcut bar at the bottom left of the window, as shown in Figure 8-27.

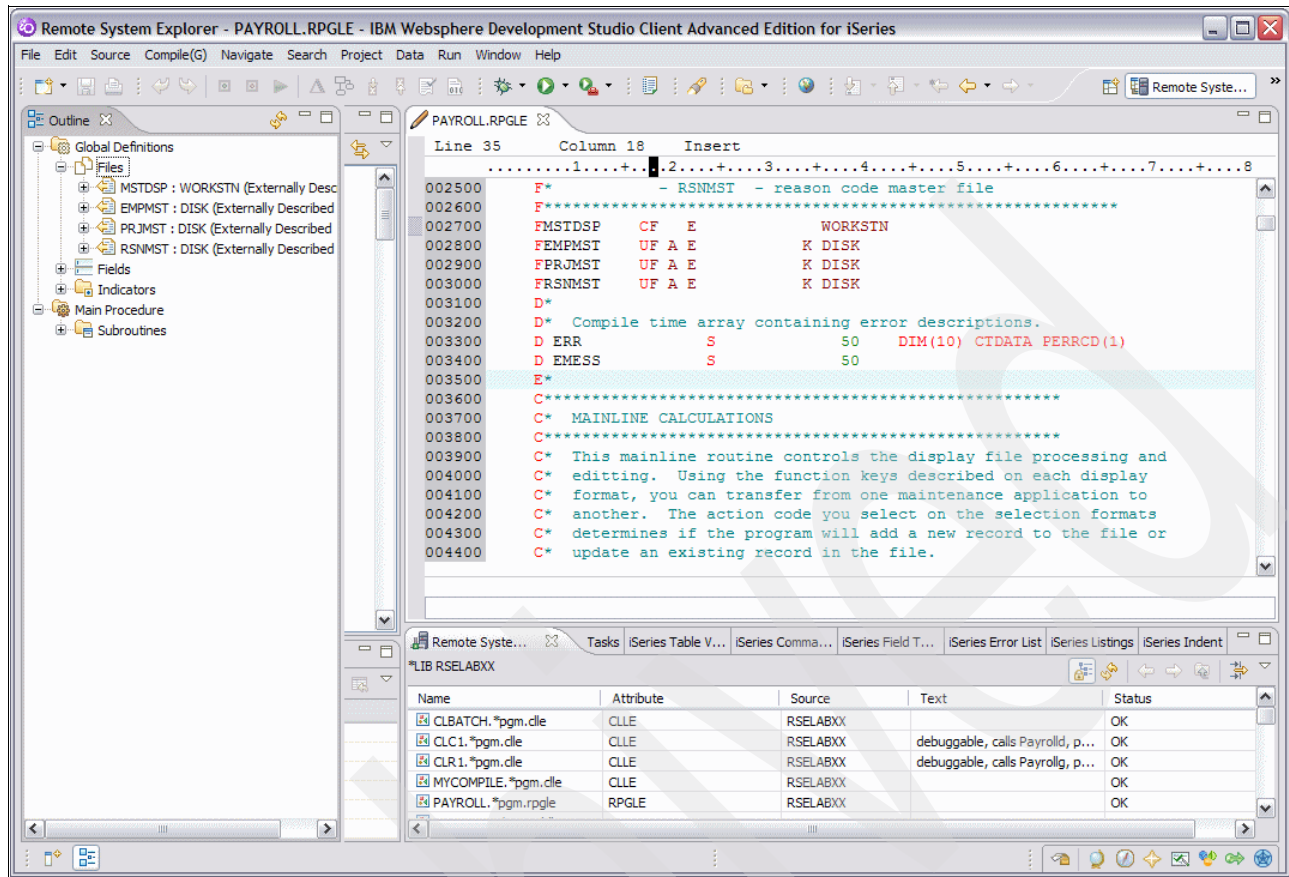


Figure 8-28 Outline as Fast View

You can now maximize the editor pane. If you want to look at the Outline view, click the icon in the shortcut bar at the bottom left of the window and the outline overlaps the editor window until you click the button again.

Additional Remote Systems LPEX Editor parser action and preferences

Additional preferences and actions are available for the System i languages:

- ▶ **Column Sensitive Editing:** This function is useful for column-sensitive languages like RPG and DDS. Normally in a Windows application, inserting and deleting text will push the remaining text left or right, and for these languages results in a syntax error. Enabling column-sensitive editing will limit insertion and deletion to the columns specified for the language.
- ▶ **Signatures:** Available for RPG and DDS, enabling this feature will automatically flag each line with the specified signature. Note that modified lines in Remote Systems LPEX Editor have the date changed as in SEU, regardless of member type.
- ▶ **Automatic-uppercasing:** Uppercases modified lines. Available for CL, DDS, RPG, COBOL members.
- ▶ **Automatic-indent:** Indents the cursor when Enter is pressed on the following line to help pretty-print your source. Available for CL, RPGLE.
- ▶ **Automatic-formatting:** Formats your source as you enter it, according to specified preferences. Available for CL and free-form SQLRPGLE.

- ▶ Open/Browse the /COPY member or Copy book: For RPG and COBOL languages, you can open or browse members referred to in the source through the pop-up in the Editor menu.
- ▶ Show fields: When a file is referenced in a program, you can use this menu option from the pop-up menu to show the fields in the file in the System i Table View. Available in RPG, COBOL, and CL.

8.1.5 Wizards in the Remote Systems LPEX Editor

The Remote Systems LPEX editor also provides several wizards to make programming easier. Some of them are described in the following sections.

Creating an RPG procedure

A procedure is any piece of code that can be called with a bound call, in essence, the CALLP operation code. You can create an RPG program, main procedure, or subprocedure while editing any RPGLE member in the Remote Systems LPEX editor.

To create the procedure, perform the following steps:

1. Expand your connection until you find the file in which you want to create the RPG procedure.
2. Double-click the file to open it in the Remote Systems LPEX editor.
3. With your cursor inside the editor view, right-click and select **New** → **Procedure**. Or select **Source** → **New** → **Procedure** from the workbench menu.
4. In the New RPG Procedure window select the procedure type from the Procedure type drop-down menu as indicated by the mouse pointer in Figure 8-29. Note that your selection automatically completes the External Name (EXTPROC) field.

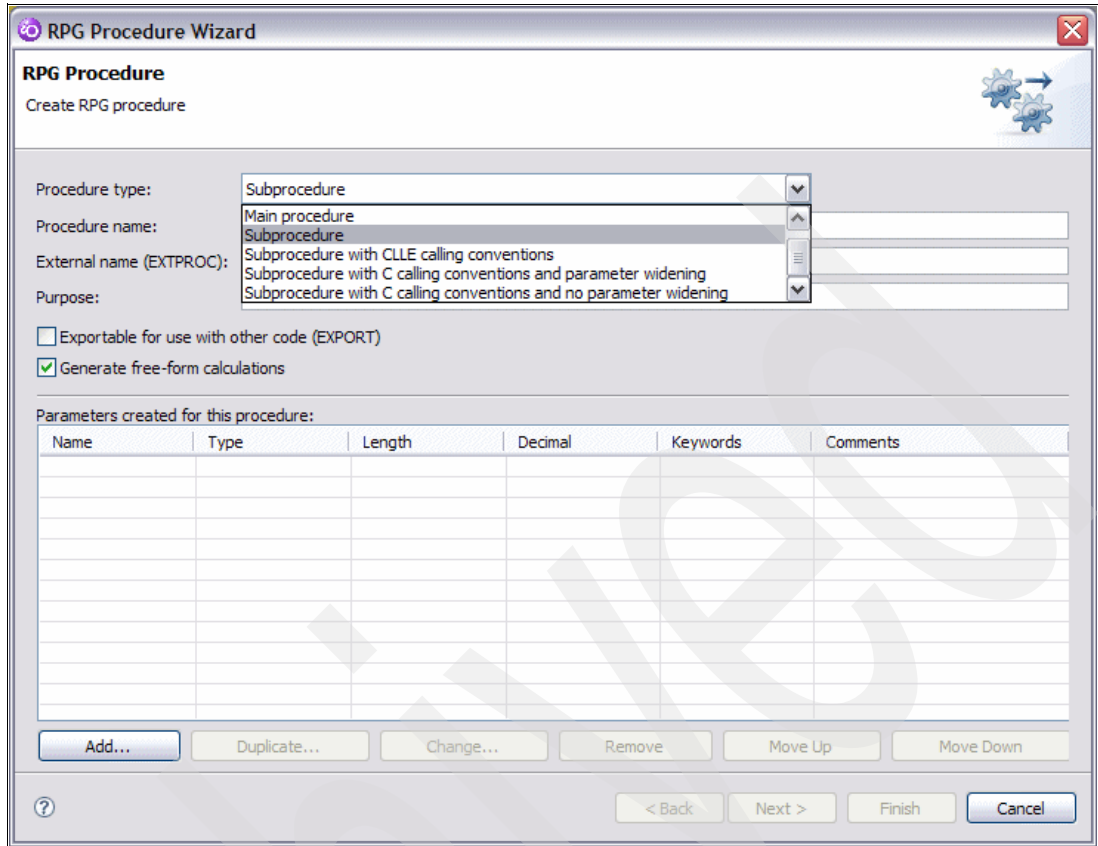


Figure 8-29 RPG procedure wizard

You have the following choices:

- Program (EXTPGM)
 - Main procedure (EXTPROC)
 - Subprocedure (EXTPROC)
 - Subprocedure with:
 - CLLE calling conventions (EXTPROC(*CL))
 - C calling conventions and parameter widening (EXTPROC(*CWIDEN))
 - C calling conventions and no parameter widening (EXTPROC(*CNOWIDEN))
5. Enter a valid ILE RPG name in the Procedure name field.
 6. (Optional) Modify the external name in the External Name (EXTPROC) if you want to override the default value.
 7. (Optional) Enter a descriptive purpose for the procedure. This description will appear in the code's header comments.
 8. Select the **Exportable for use with other code (EXPORT)** check box if this procedure is exportable for use with other code. Note, however, that if you select this check box, the wizard places both the prototype and procedure code into the current file. After you finish the wizard, you need to manually move the prototype into a COPY file, so that procedures can call this prototype. You also need to enter a COPY statement into the procedure code, at the place where you removed the prototype code. Click **Add** to add a parameter to your procedure.
 9. (Optional) Select the **Return a value** check box if you want to define a Return value. This action enables the Next button. Otherwise, click **Finish**.

If you selected **Return a value**, you also need to perform the following steps:

1. Click **Next** to go to the Return Value page of the RPG Procedure wizard.
2. Select a return value type from the Type drop-down list. The type that you select automatically completes a number of the other fields in this page of the wizard, and keeps some of the fields disabled if they are not applicable for the type you selected. You can modify any of the values in the enabled fields if you want to. Make sure to check the F1 help in the various fields of this wizard if you are not sure what to enter, or you can click the help icon (?) at the bottom left of the wizard.

Note: If the type you selected is date or time, the format label will be set to DATFMT (for date) or TIMFMT (for time). Then you can select the format and separator for your date or time label. If the type you selected is data, a Class Entry field is enabled where you can specify a valid Java class.

3. (Optional) Enter a description for the return value. This description will appear in the code's header comments. Click **Finish**.

Creating RPG D-Specifications

You can use a New RPG *D-Specification wizard* to help you create a Definition Specification from any ILE RPG file in the Remote Systems LPEX editor; Figure 8-30.

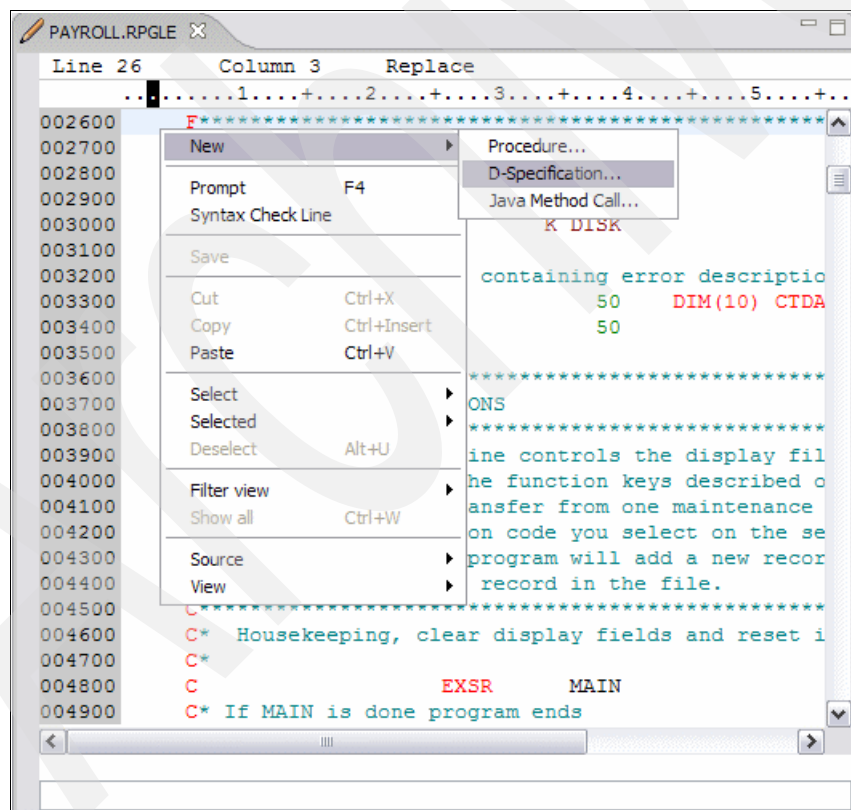


Figure 8-30 Starting the D-Specifications wizard

A D-Specification defines items used in your program, such as standalone fields, named constants, and data structures with or without a subfield.

To create the D-Specification, perform the following steps:

1. Expand your connection until you find the ILE RPG file in which you want to create the D-Specification.
2. Double-click the file to open it in the Remote Systems LPEX editor.
3. With your cursor inside the editor view, right-click and select **New** → **D-Specification** as shown in Figure 8-30 or select **Source** → **New** → **D-Specification** from the workbench menu.
4. In the New RPG D-Specification wizard, as shown in Figure 8-31, you may specify a name and purpose for the D-Specification.
5. Select the type of D-Specification from the Type drop-down list. Your selection determines the content of the rest of the pages in the New RPG D-Specification wizard. Use the F1 help throughout the rest of the wizard for more information about each option, or you can click the help icon (?) at the bottom left of the wizard.

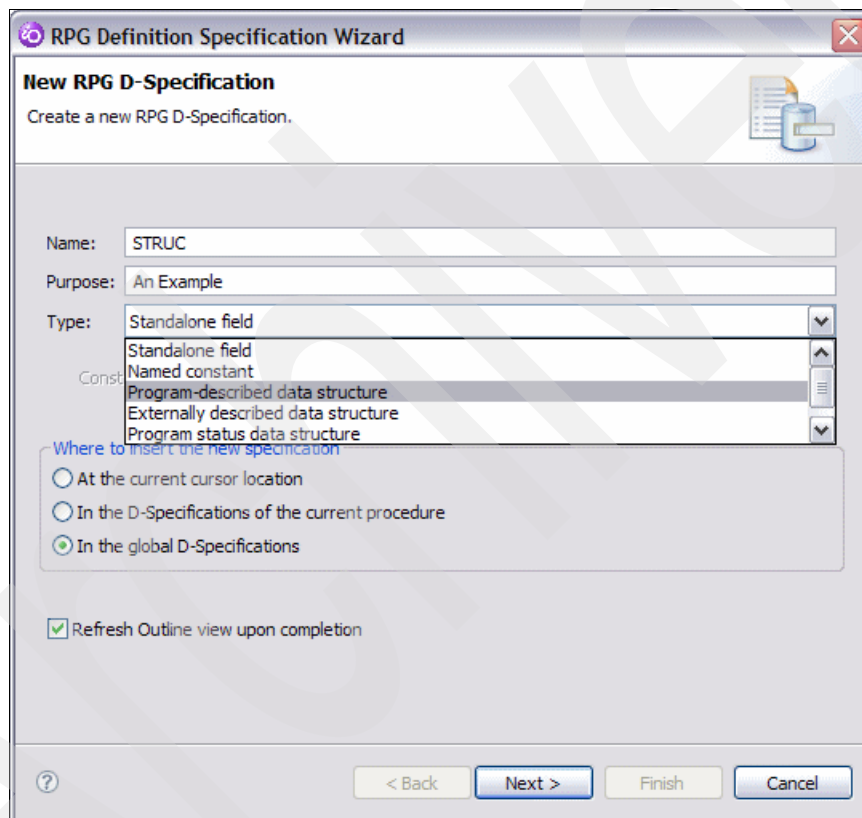


Figure 8-31 First page of the D-Specifications wizard

You have the following choices:

- *Standalone Field*: to create a standalone field type on a field-type page or field-type keyword page, that contains field types and their keywords.
- *Named Constant*
- *Program-described data structure*: to create a Simple data structure keyword page or a Simple data structure subfield page, where the keywords and subfields are created separately.

- *Externally-described data structure*: to create an External data structure page or an External data structure subfield page, where the keywords and external subfields are created separately.
 - *Program status data structure*: to create a Program status data structure subfield prefix page, and to specify the keywords and subfield prefixes for the Program status data structure.
 - *File information data structure*: to create a File information data structure subfield prefix page, and to specify the keywords, subfield prefixes, and the File information data structure type.
6. Select from the three radio button choices underneath *Where to insert the new specification*:
- *At the current cursor location*: to simply insert the code at the current location in your source.
 - *In the D-Specifications of the current procedure*: to append the code to the D-Specification of the procedure where your cursor is located.
 - *In the global D-Specifications*: to insert the code right before the first found Input specification, which could be a C-Specification, O-Specification, P-Specification, “/free”, “***”, or the end of the file.
7. Click **Next** and the second page of the D-Specifications wizard, as shown in Figure 8-32, appears.

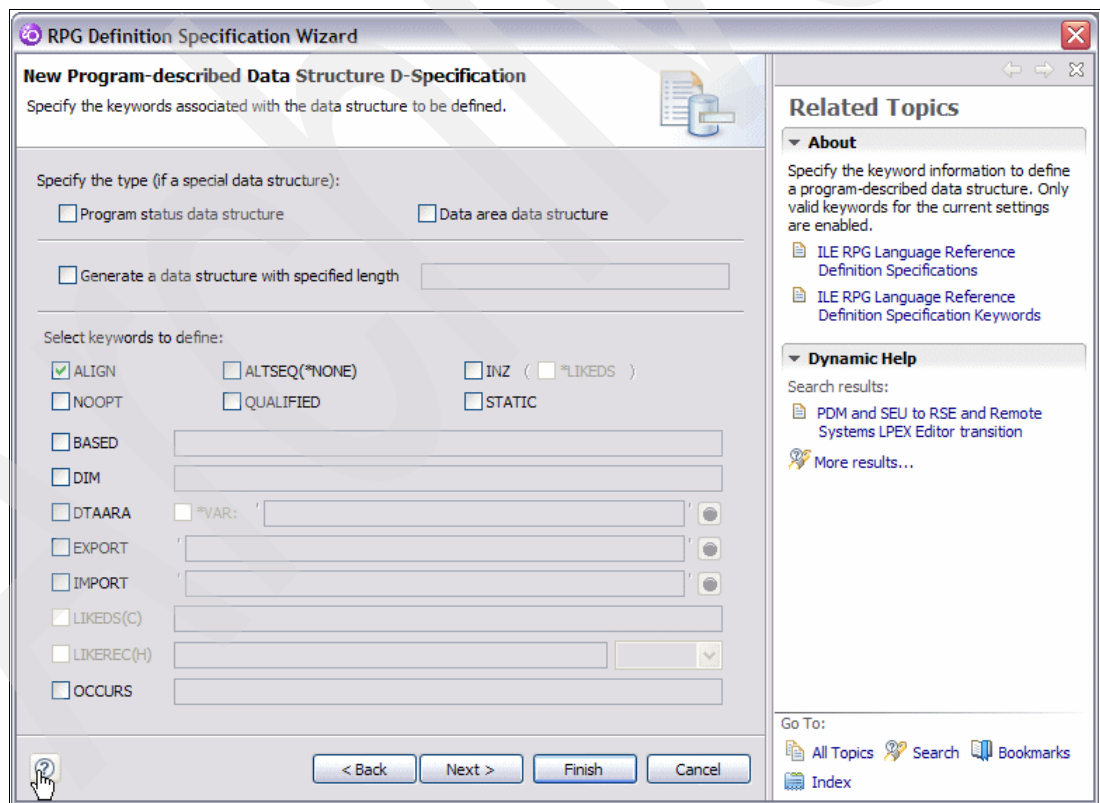


Figure 8-32 Second page of the D-Specifications wizard

8. Complete the rest of the wizard, using the F1 help for tool tips and suggestions for all of the options, or you can click the help icon (?) at the bottom left of the wizard, as indicated by the mouse pointer in Figure 8-32.

Creating an RPG Java Method Call

You can now create an RPG Java Method call from any ILE RPG file, when you use the Remote Systems LPEX editor. An RPG Java method call is a piece of ILE RPG code that calls various Java methods, so you can call Java methods from your RPG program. When you specify various Java class and method information, the method call creation wizard creates valid ILE RPG code that calls the Java methods from your RPG program.

The New Java method call wizard will change dynamically, depending on the selections that you make. Use the F1 help for more information about each option as you go through the wizard. To create the RPG Java method call, perform these steps:

1. Expand your connection until you find the file in which you want to create the method call.
2. Double-click the file to open it in the Remote Systems LPEX editor.
3. With your cursor inside the editor view, right-click and select **New** → **Java Method Call**. Or select **Source** → **New** → **Java Method Call** from the workbench menu.
4. In the *New Java Method Call* wizard, specify the Java package and the class you want to use for the method call. Use the **Browse project**, **Browse local**, and **Browse remote** buttons to search for packages and classes. When you search for class files on your local system, you can click **Add JAR file** to find a file that contains the class you want to use. Use the F1 help for more information about each option.
5. Click **Next**.
6. Select the Java method that you want to use. The methods listed are the ones available from the class selected in the previous page of the wizard.
7. Click **Next**.
8. Select the Java method parameters that you want to use. For each parameter, you can generate an RPG field, specify its RPG name, type, length, or dimension, and whether to pass the parameter by reference or to have a read-only reference. You can click each cell in the table to edit the value directly.

If you selected a method that does not contain parameters, the page will be blank, yet considered complete, with the Next button enabled. Click **Next**.

Note: When you produce an RPG Java method call from ILE RPG, the ILE RPG code is mapped into Java code. For example, “char” in Java is mapped to “1C” in ILE RPG. See the related reference topic for more information about mapping between standard Java types and ILE RPG types.

If the Java method you selected is not static, then:

- a. Specify the Java constructor you want to use to construct the Java object. (This page, and the next one, do not appear if you previously selected a static method, as no constructor is necessary.)
 - b. Click **Next**.
 - c. Specify parameters for the Java constructor selected. For each parameter, you can generate an RPG field, specify the name, type, length, dimension, and whether to pass the parameter by reference or to have a read-only reference.
 - d. Click **Next**.
9. Specify the options for code generation. You can specify whether or not to generate an RPG object for your Java class, whether or not to create a method prototype, and where to place the generated RPG code.
 - If you decide to generate an RPG object, you need to specify an RPG name for the object and an RPG name for the object constructor.

- If you decide to generate an RPG method prototype, you need to specify a prototype name.
- If you decide to generate a method call code, you need to specify an RPG return field name.

10. You have two choices for specifying where you want to place the generated code:

- *Distributed appropriately*: to generate D-Specifications in the D-Specification area of your code, and to generate calculations (the code to call the method) after the cursor.
- *All after the current line*: to generate and append the code after the cursor.

11. Click **Finish**.

Converting ILE RPG code from fixed form to free form

You can convert fixed form ILE RPG code into free form ILE RPG code (C-specifications) in the Remote Systems LPEX editor.

To convert your source, with your ILE RPG source open in the Remote Systems LPEX editor, select **Source** → **Convert All To Free Form** from the workbench menu. This will overwrite the original source.

Tip: You can use **Edit** → **Undo** from the workbench menu, or press Ctrl+Z, to undo your changes to the source.

If you only want to convert a section of your source (and not the entire file), select the source you want to convert, right-click and select **Convert Selection To Free Form** as shown in Figure 8-33. In this instance, the selected source is replaced with the converted source.

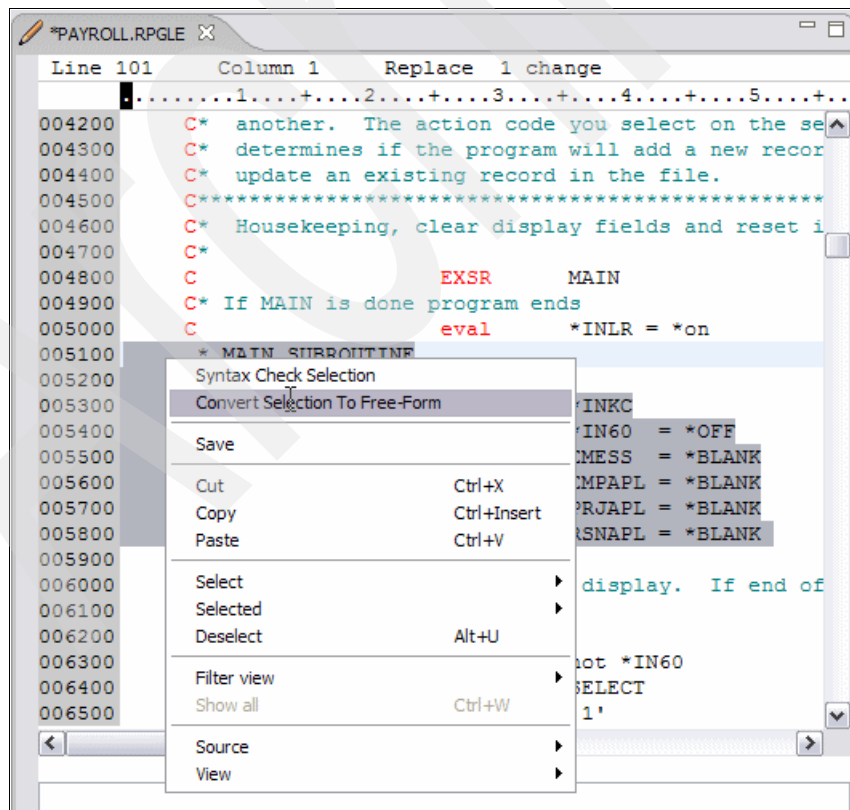


Figure 8-33 Convert selection to Free-Form

Note: When you convert your source to a free format, the default preference for automatic indenting determines how your source is indented. To change this setting:

1. From the Remote Systems view, click the drop-down menu (▾) and select **Preferences** → **Remote Systems LPEX Editor**.
2. In the Preferences window, expand **Remote Systems** → **LPEX Editor** → **System i Parsers** and select **ILE RPG**. In the Preferences area, verify that Automatic Indent is checked, and adjust the value in the Blanks (1-20) field as shown in Figure 8-34.

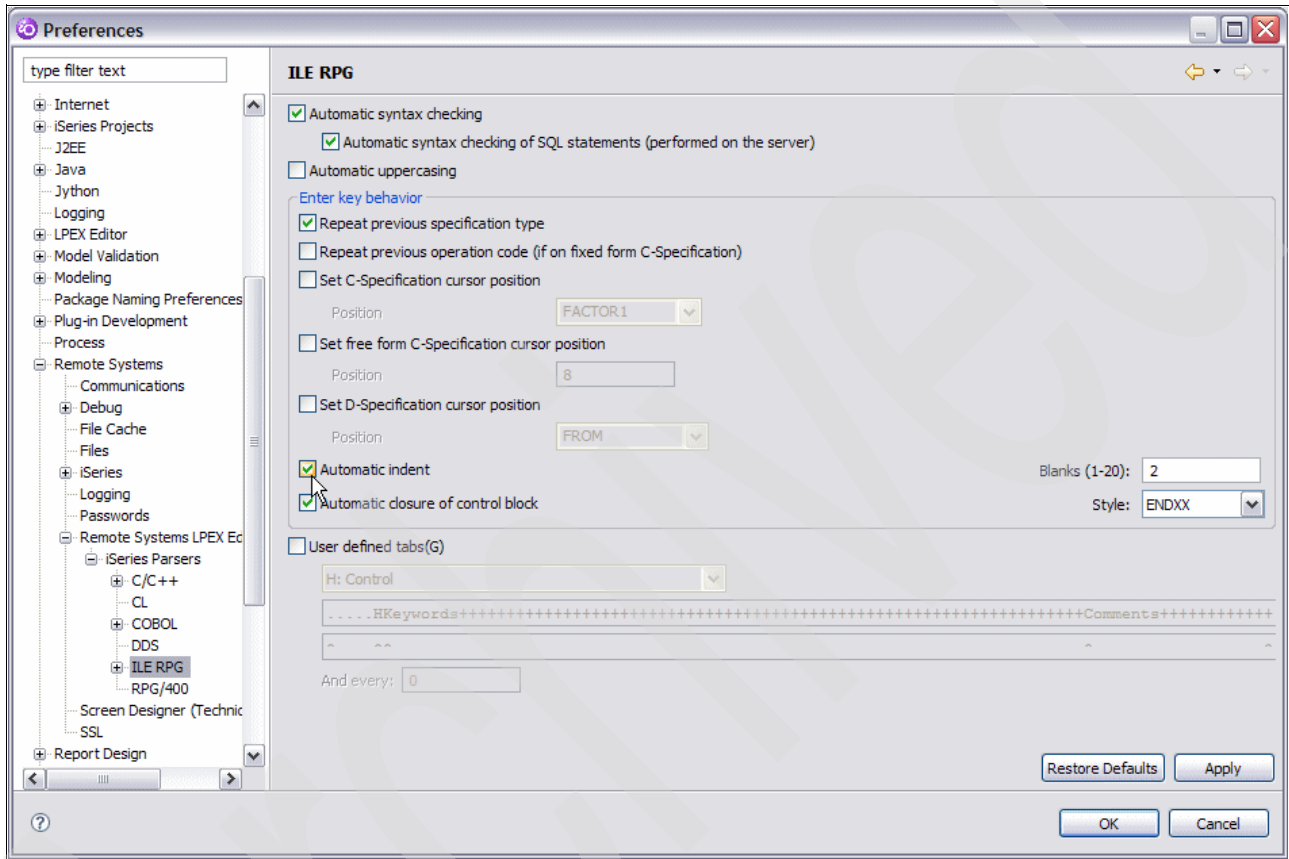


Figure 8-34 Automatic indent settings

Automatic indenting when entering free form

On a free-form line, when Enter is pressed, automatically indent the position of the cursor for the new line if the current line has one of the following opcodes (line does not need to have a semi-colon):

- ▶ DOU
- ▶ DOW
- ▶ FOR
- ▶ IF
- ▶ MONITOR
- ▶ BEGSR
- ▶ SELECT
- ▶ WHEN
- ▶ ELSE
- ▶ ELSEIF
- ▶ OTHER
- ▶ ON-ERROR

Auto-indent is controlled by a preference setting as described in “Customizing the automatic formatter” on page 286. The indent value is specified in the ILE RPG Parser Preference page. Valid range is 1 to 20 with a default of 2. The preference is on by default. The preference is disabled if the “Repeat previous specification type” preference is off.

Auto closer when editing RPG sources

When Enter is pressed, the corresponding ENDxx statement is automatically added after the new line if the current line has one of the following opcodes (line does not need to have a semi-colon):

- ▶ DO (fixed form only)
- ▶ DOU
- ▶ DOUxx (fixed form only)
- ▶ DOW
- ▶ DOWxx (fixed form only)
- ▶ FOR
- ▶ IF
- ▶ IFxx (fixed form only)
- ▶ BEGSR
- ▶ MONITOR
- ▶ SELECT

This applies to fixed and free form.

The case for the ENDxx statement is controlled by a Style subpreference to the Auto-closure preference. The possible values are ENDXX, EndXx, Endxx, endXx, and endxx. The default is ENDXX.

Auto-closure is controlled by a preference setting. The preference is off by default. The preference is disabled if the “Repeat previous specification type” preference is off.

8.1.6 Customizing the automatic formatter

When editing free-form RPG, CL, CLP, CLLE, and CMD source with the Remote Systems LPEX editor, the source is automatically formatted. You can customize this formatter to suit your needs.

When you enter source into the editing area of a CL file, the text is formatted when you go to a new line or move your cursor to a different line. For example, if you begin your line of text with a command such as CRTCMD, the beginning of the label is automatically positioned to column 2. However, you can specify a different column position in the editor preferences.

1. From the Remote Systems view, click the drop-down menu (▾) and select **Preferences** → **Remote Systems LPEX Editor** or select **Window** → **Preferences** from the workbench menu and then select **Remote Systems** → **Remote Systems LPEX Editor** → **System i Parsers** in the left pane of the Preferences dialog.
2. In the Preferences window, select **CL** (or any other language you want to customize the formatter for); see Figure 8-35.

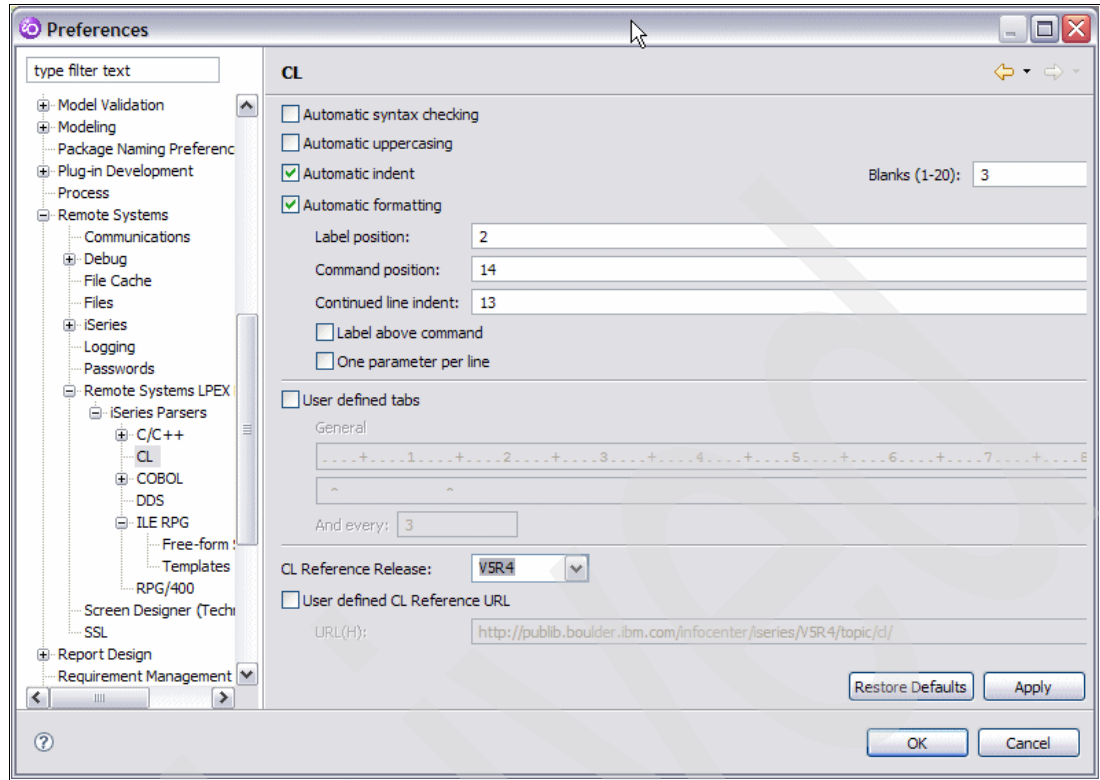


Figure 8-35 Customize the automatic formatter for CL source

3. The check box for *Automatic formatting* should be selected by default. You can then edit the entries in the fields below to customize the formatter:
 - The value in the *Label position* field specifies the column position for the label.
 - The value in the *Command position* field specifies the column position for the command. This value must be greater than the Label position value, unless you select the *Label above command* check box, in which case, the command can appear directly beneath or even to the left of the label, on the next line.
 - The value in the *Continued line indent* field specifies the column where the text will be continued if it spans more than one line.
 - Check the *Label above command* check box if you want the label to appear above, rather than to the left, of your command string.
 - Check the *One parameter per line* check box if you only want one parameter to appear on each line.

8.1.7 New in V7: Language template locations

In WDSC 7 the templates are grouped by language: one for COBOL, one for C/C++, and one for ILE RPG. The template ILE RPG FREE-FORM is renamed to FREE-FORM.

To create, edit, or remove templates:

1. From the workbench menu select **Window** → **Preferences**.
2. In the preferences window expand **Remote Systems** → **Remote Systems LPEX Editor** → **System i Parsers**.

3. You can expand C/C++, COBOL or ILE RPG and then choose **Templates**. Figure 8-36 shows the templates for ILE RPG.

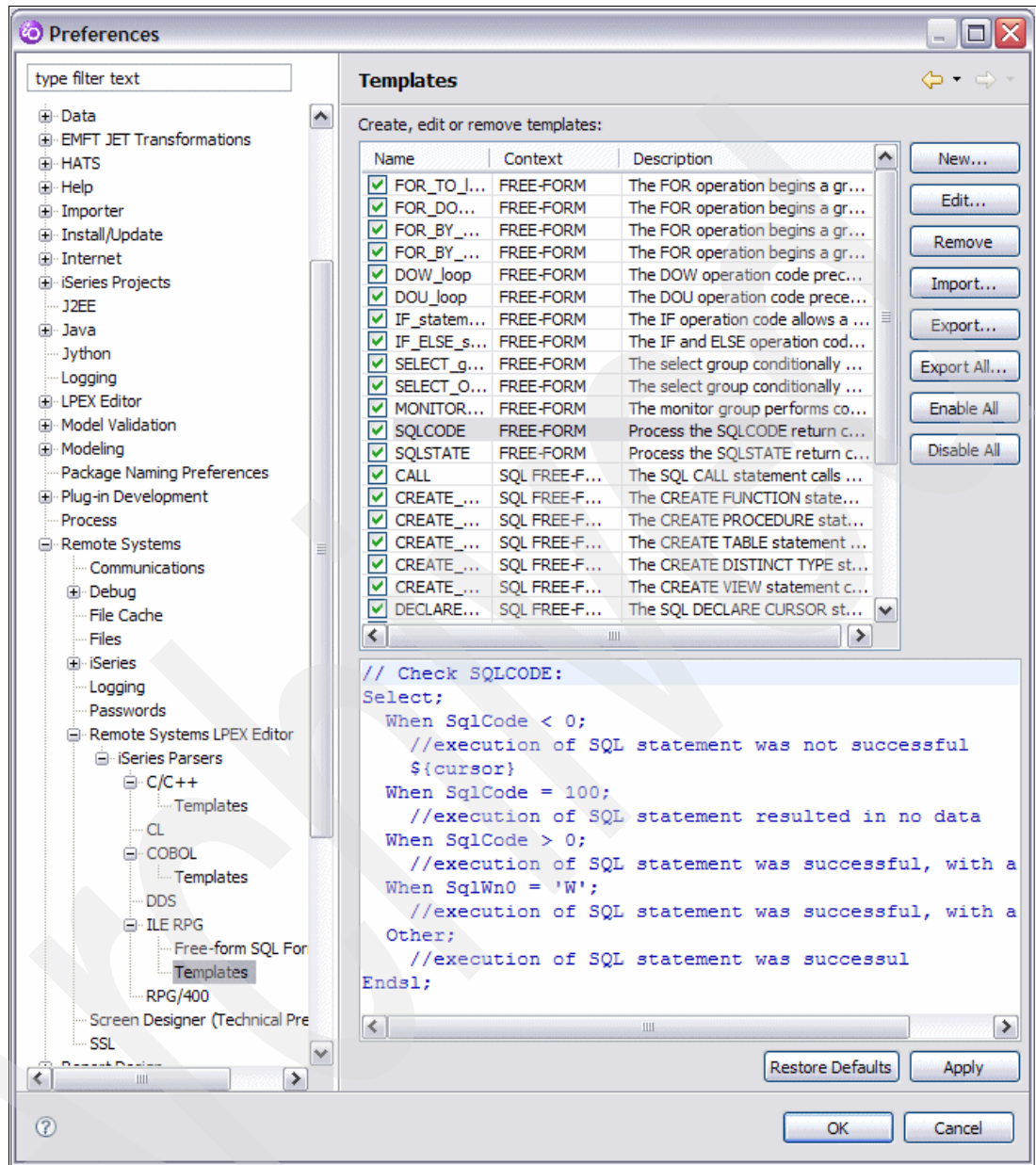


Figure 8-36 ILE RPG SQL parser templates

Templates for SQL statements embedded in free-form ILE RPG are added. There is a new context called SQL FREE-FORM.

8.1.8 New in V7: ILE RPG SQL parser

Templates allow you to define and manage your code templates. Templates may be used in Remote Systems LPEX Editor in the supported language contexts. To use templates, in the Remote Systems LPEX Editor, position the caret in a place where you want to insert a template and press Ctrl+Space. Templates appear in the presented list. Note that the list is filtered as you type, so typing a few first characters of a template name will reveal it. A preview is also presented for each selected template.

You can now perform Syntax checking for source members of type SQLRPGLE. This also applies to other source types that have been associated with the ILErpgSql parser. These source members now include:

- ▶ Support for Tokenizing embedded SQL Statements
- ▶ Support for Uppercasing, Indenting and Formatting of SQL Statements in free format ILE RPG
- ▶ Commonly used templates to assist in editing SQL statements in free format ILE RPG

To create a member with type SQLRPGLE:

1. In the Remote Systems view right-click the file, and select **New** → **Member**.
2. In the New member wizard select **SQLRPGLE** from the members type list as shown in Figure 8-37.

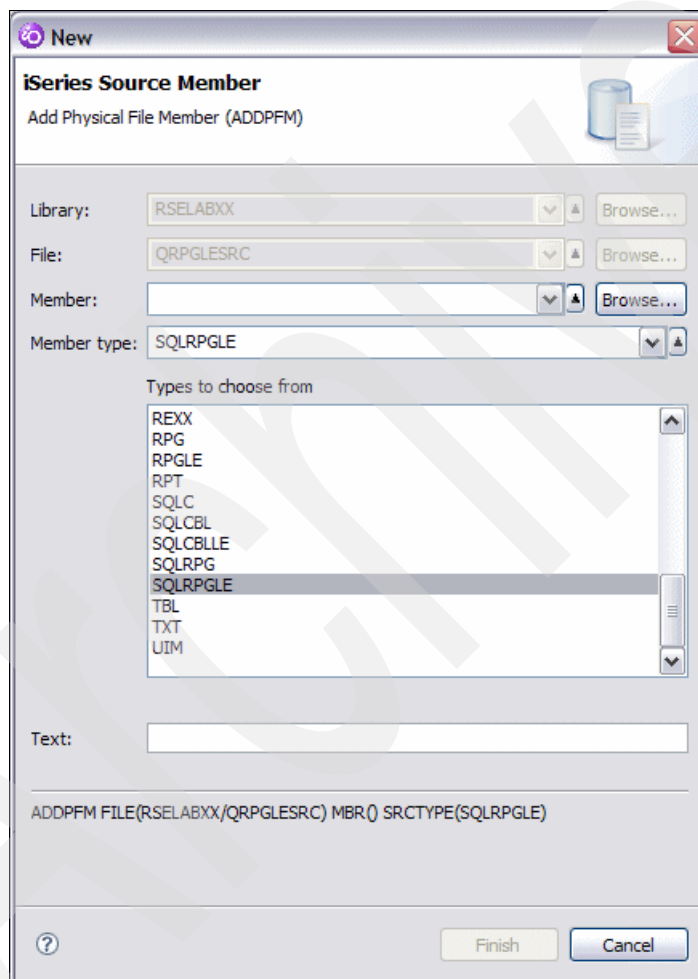


Figure 8-37 New SQLRPGLE member

For ILE COBOL, a new source type SQLCBLLE is added. It is associated with the ILEcobolSqlCics parser.

To associate a SQLRPGLE source member with ILErpgSql parser:

1. From the workbench menu, select **Window** → **Preferences**.
2. In the preferences dialog, select **LPEX Editor** → **Parsers** → **Parser Associations**.

3. In the parsers list select **sqlrpgle** and from the Document parser drop-down list select **ILERpgSql**, as shown in Figure 8-38.
4. Click **Set**, then **Apply**.

Important: Make sure to press the Set button in the **Parser Associations** dialog before clicking **Apply**, or closing the Preferences dialog.

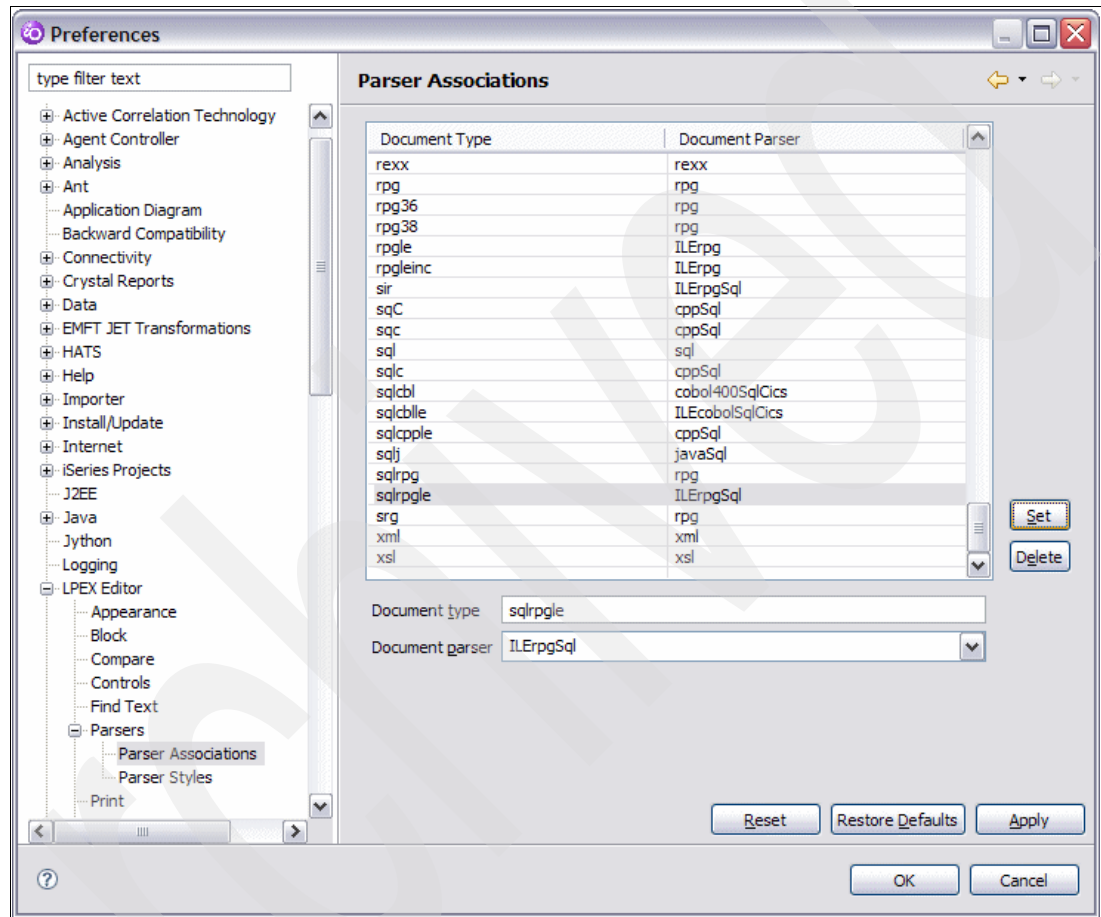


Figure 8-38 Set parser association

8.1.9 Other editing enhancements in V7

There are a number of enhancements to the LPEX Editor:

- ▶ New Incremental Find dialog check box in the **LPEX Editor** → **Find Text** preference page to set the behavior of the Find dialog; see Figure 8-39. In incremental (or live) find mode, the editor starts changing the cursor location and highlighting the search result for each character that you type in the Find dialog. This is the default. If Incremental Find is not selected in the preferences, the search will start after you type your search argument and press Enter.

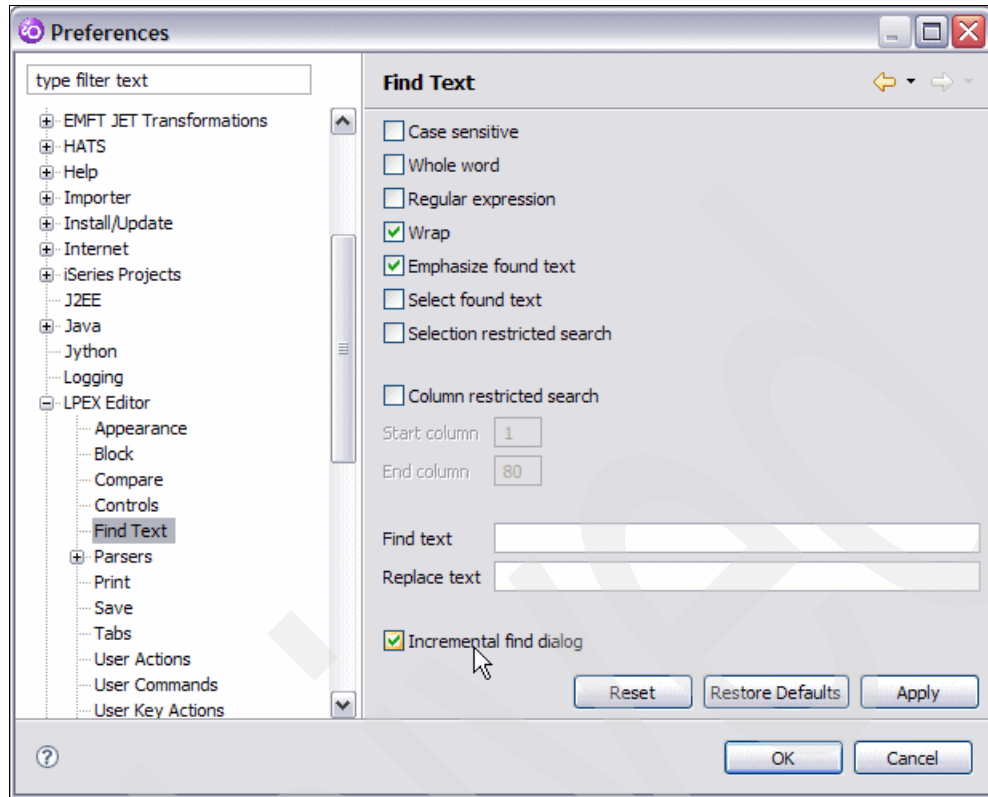


Figure 8-39 Incremental Find dialog check box

- ▶ New print.sequenceNumbers editor parameter, and new sequenceNumbers argument for the print command with a corresponding option in the LPEX Editor → Print preference page (Sequence numbers check box); see Figure 8-40.

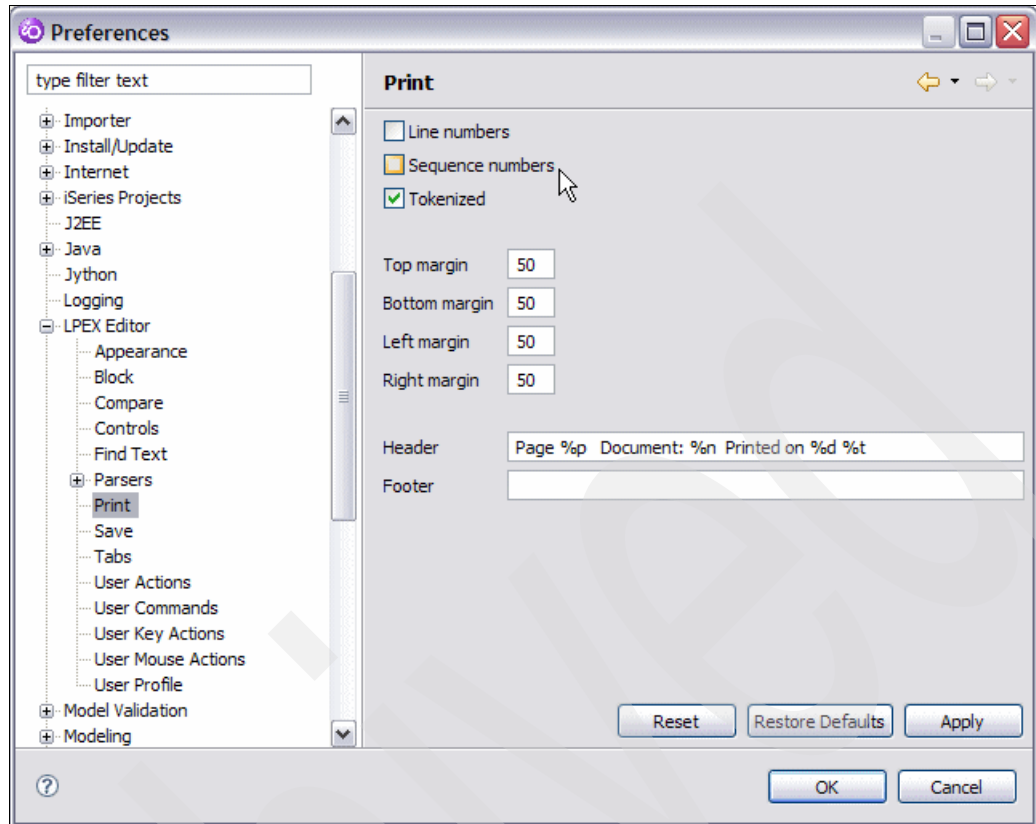


Figure 8-40 Print Sequence numbers argument

- ▶ New status line information on incomplete (pending) prefix commands. Double-click **Pending** to move the cursor to the prefix command in question, as shown in Figure 8-41.

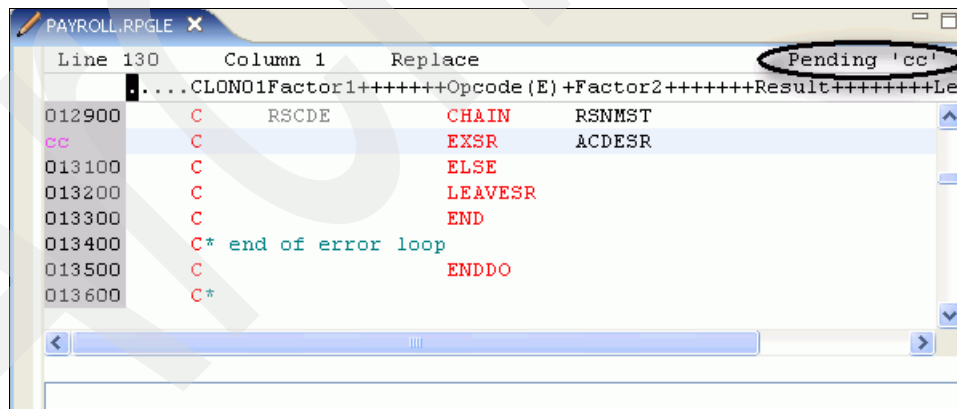


Figure 8-41 Incomplete prefix information feature

Screen Designer Technical Preview - the modern Screen Design Aid (SDA) in WDSC AE 7.0

Screen Designer is a technical preview that provides the capability to graphically design and modify the content of DDS display files. The purpose of the Screen Designer in its final implementation will be a modern alternative to the Screen Design Aid (SDA), the text based (5250) design tool for display files, as well as the CODE Designer with its graphical user interface for developing 5250 formats. However, Screen Designer does not support designing printer files or physical files.

CODE Designer was first introduced with CODE/400 and later with the advent of WDSC delivered together with it, but it needs to be installed separately and is not based on Eclipse. CODE Designer is still delivered with WebSphere Development Studio Client for iSeries (WDSC) 7.0 and WebSphere Development Studio Client Advanced Edition for iSeries (WDSC AE) 7.0. It may be deprecated in the future whenever the fully supported version of Screen Designer will become available.

Note: The purpose of a *technical preview* is to provide a general idea about how the product may be implemented in the future. You should use it as much as possible to get an understanding. However, you should *not* use it for real production work. Also keep in mind while reading this chapter that any part of the current implementation may change.

Advanced In Version 7.0, this capability is only provided as a technical preview with WebSphere Development Studio Client Advanced Edition for iSeries (WDSC AE 7.0).

References to more information

To find information about the Screen Designer, select **Help** → **Help Contents**, then expand:

- ▶ **WebSphere Development Studio Client for System i**

- ▶ → **Developing System i server applications using Remote System Explorer**
- ▶ → **Screen Designer - Technical Preview.**

This chapter assumes that you have some basic experiences in describing 5250 displays through *data description specifications (DDS)* and that you are familiar with terminology used to design 5250 displays, such as *Display file (DSPF)*, *record format*, *field* and others. You can find more information on DDS by selecting **Help** → **Help Contents**, then:

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Reference**
- ▶ → **Language Reference**
- ▶ → **System i programming information**
- ▶ → **Data Definition documentation**
- ▶ → **DDS for Display Files**

9.1 Screen Designer overview

The Screen Designer is a so-called “What you see is what you get” (WYSIWYG) editor and provides much more than what you can do by editing your DDS sources with Remote Systems LPEX editor, which is always an alternative to using the Screen Designer.

The “grandfather” of Screen Designer is the Screen Design Aid (SDA). It was first introduced with the IBM System /34 and IBM System /38 in 1978 and 1979 and still available today as part of WebSphere Development Studio (5722-WDS) for i5/OS V5R4, without any recent enhancements, however.

SDA also supported the concept of a WYSIWYG editor from the beginning. The restricting factor, however, is the fact that SDA needs to use the same “canvas”—a 24 by 80 (or 27 by 132) grid of characters to display the “what you get” while at the same time using it as an input device for accepting the user’s commands and directions.

The power of a graphical user interface (GUI) in general and the Eclipse framework in particular here, is the capability of using multiple views (or sub-windows or panes) at the same time. That way, one of the views (the *Design Area* described below) is not only used to show how the result of your work will look, but also allows changes by clicking, dragging and dropping. At the same time the Source view (the Remote Systems LPEX editor) can display, and allows you to change, the generated or manually entered source statements.

Other panes contain more tools to easily develop and enhance the display file. The contents of all these views are synchronized. For example, changing any of the keywords in the Properties view immediately changes its appearance in the Design Area and the contents of the editor. Likewise, clicking any of the elements in the Outline view immediately changes the selection in the Design area and the cursor position in the Remote Systems LPEX editor (the “Source” page).

In some respects, the Screen Designer uses similar concepts for designing character oriented 5250 formats, as the Page Designer does for creating HTML-based browser pages. However, as the results are quite different, so are the tools in detail.

The following sections show how to start Screen Designer and briefly introduce the views used by it.

9.2 Launching the Screen Designer

You may launch the Screen Designer either from the Remote Systems view or the iSeries Projects Navigator view, by right-clicking any DDS source member of a source type Display File (DSPF) or DDS menu (MNUDDS) and selecting **Open With** → **Screen Designer (Technical Preview only)** from the context menu, as shown in Figure 9-1 on page 296.

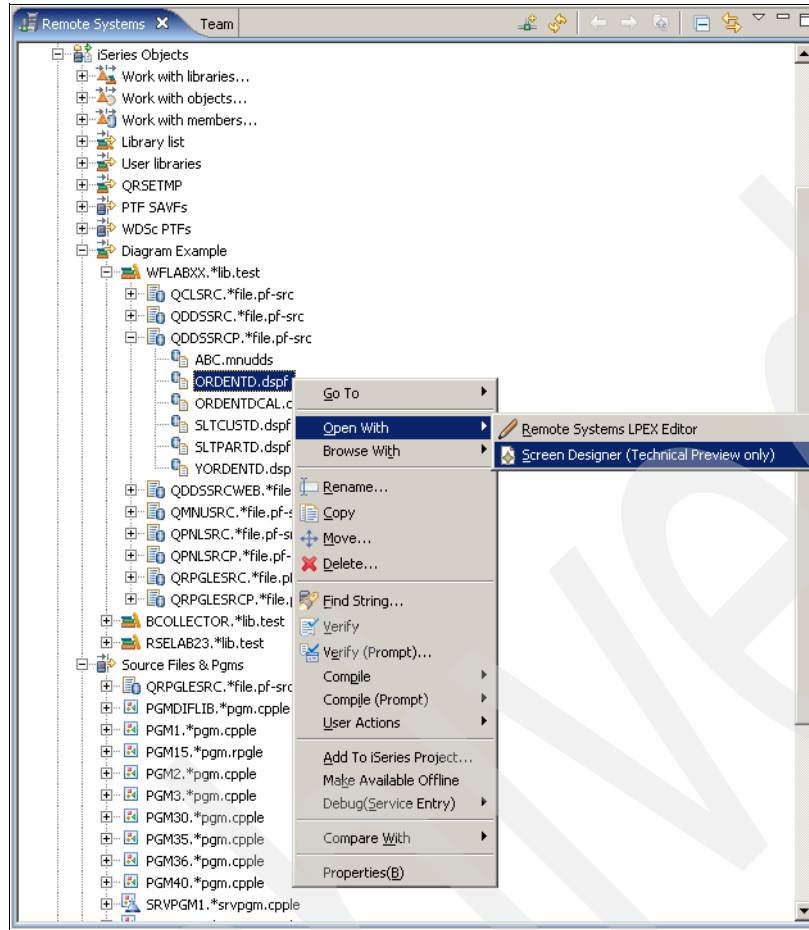


Figure 9-1 Launching the Screen Designer from the Remote Systems View

Note that the Screen Designer does not just open a single new view, but rather adds three tabbed pages to the editor area, and the Outline and Properties views get populated as you can see in Figure 9-2 on page 297.

The tabbed pages in the editor area are:

- ▶ *Source* page to view and edit source code
- ▶ *Preview* page, which is not yet available as technical preview in WDSC AE 7.0
- ▶ *Design* page (currently open in Figure 9-2 on page 297) which is further subdivided into:
 - The *Design area* in the lower left showing the layout of the panel to be designed
 - An internal *Palette*, on the right of the Design area
 - A set of *toolbar buttons*, immediately above the design area
 - The *Screen Controls*, at the top of the pane

We describe the above elements in more detail in the following sections after describing the Screen Designer perspective as an alternative to the Remote System Explorer perspective, when using the Screen Designer.

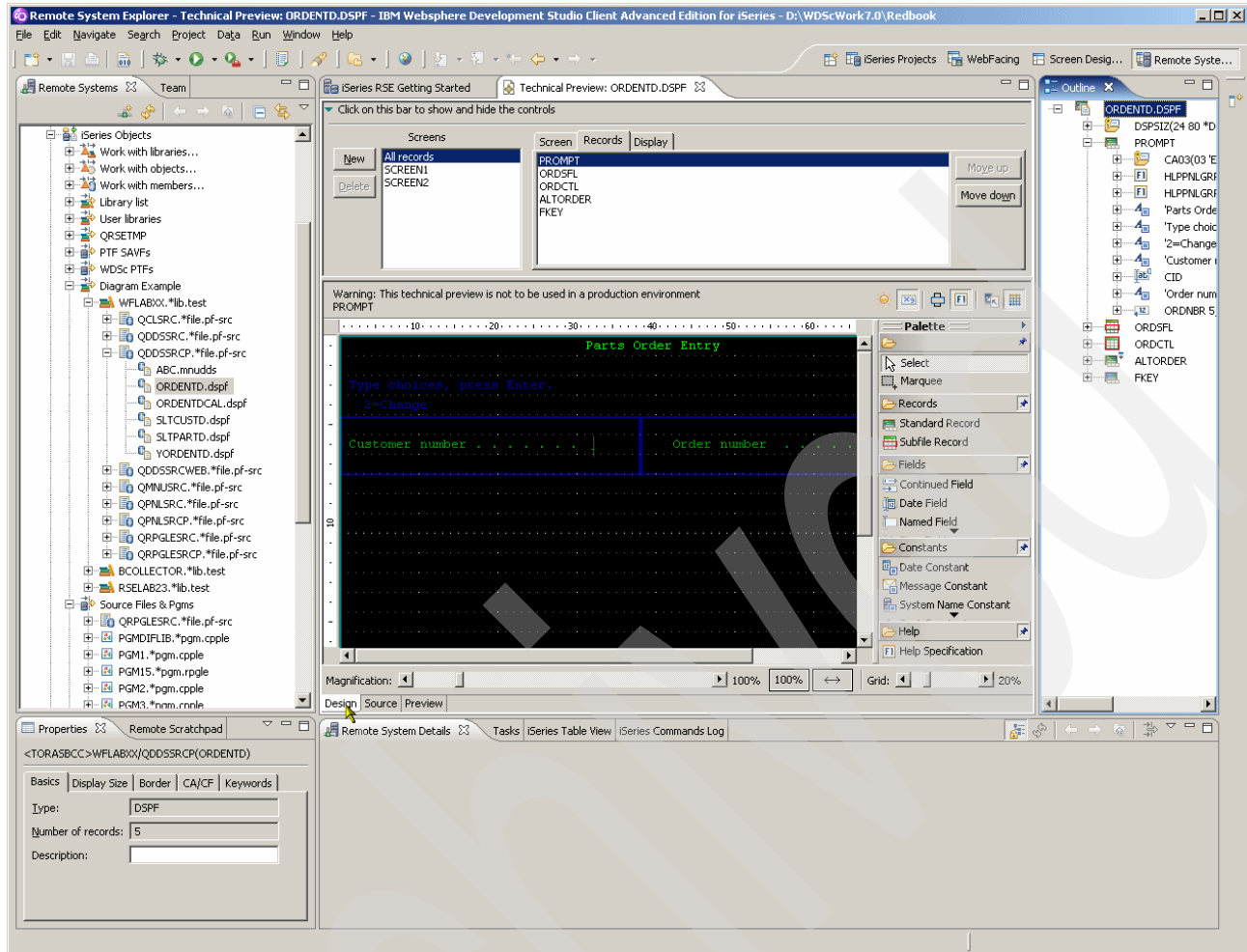


Figure 9-2 The Screen Designer in the Remote System Explorer perspective

9.3 Understanding the Screen Designer perspective

the Screen Designer either from the Remote Systems view or the iSeries Projects Navigator view. Unless you created or modified your own perspectives, the Remote Systems view is part of the *perspective* and the *System i Projects perspective* and the is part of the System i Projects perspective.

Either of those two perspective can be used to work with . However, if you prefer to use a perspective specially designed for the → → → You may compare both perspectives in Figure 9-2 on page 297 and Figure 9-3. As mentioned earlier, the Screen Designer uses some of the ideas of the Page Designer, such as using three overlaying editor pages, Design, Source, and Preview. Comparing the Web perspective and the Screen Designer perspective, you will also find similarities, for example the Palette on the upper right and the Outline view in the lower left pane.

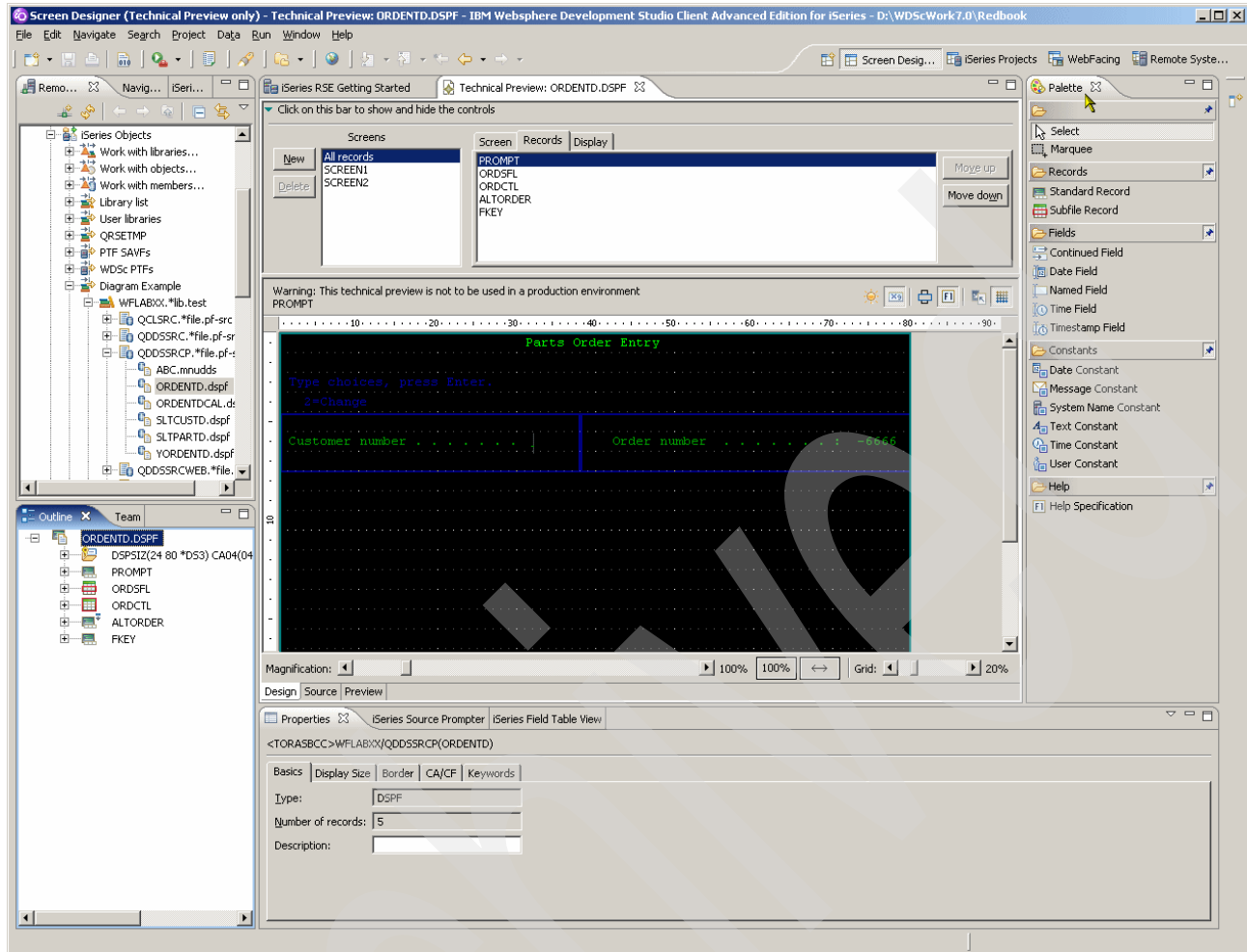


Figure 9-3 The Screen Designer in the Screen Designer perspective

Note: The default layout of the Screen Designer perspective uses the full Workbench Palette, whereas the RSE perspective by default does not open a Palette view and the integrated Palette as part of the Design area is shown instead. Compare the location and appearance of the Palettes in Figure 9-2 on page 297 and Figure 9-3. We talk about how to use the Palette in 9.4.2, “Adding new items to your display file using the palette” on page 301.

9.4 Using the Screen Designer editor pages

You find all the central and most important components of the Screen Designer in the editor area of your workbench. It consists of three tabbed pages:

- ▶ *Source* page to view and edit source code
- ▶ *Preview* page (not yet available as of the technical preview in WDCS AE 7.0)
- ▶ *Design* page (currently open in Figure 9-3) which is further subdivided in:
 - The *Design area* in the middle showing the layout of the record format to be designed
 - An internal *Palette*, on the right of the Design area (if the full Palette view is not open)
 - A set of *toolbar buttons* immediately above the design area
 - The *Screen Controls*, at the top of the pane

The following sections describe each of these components in more detail. Later in 9.5, “See everything at a glance and navigate quickly with the Outline view” on page 312 and 9.6, “Working with detailed attributes in the Properties view” on page 314 we cover the remaining views being used for Screen Designer.

9.4.1 Design area for WYSIWYG DDS development

When the Screen Designer is opened, one of two tabbed pages may be used to edit a display file. (In a future release, the third page, the Preview page, may contain just a read-only image of the display.) The Design page can be used to graphically modify screens, records and fields. Whereas the Source page (described in “Seeing and editing your DDS statements in the Source page” on page 312) is available for directly editing the display file source code.

After you start the Screen Designer, the Design area contains the first record format (indicated by the mouse pointer in Figure 9-4) of the display file formatted very similar to how it will appear on a 5250 display or emulator session.

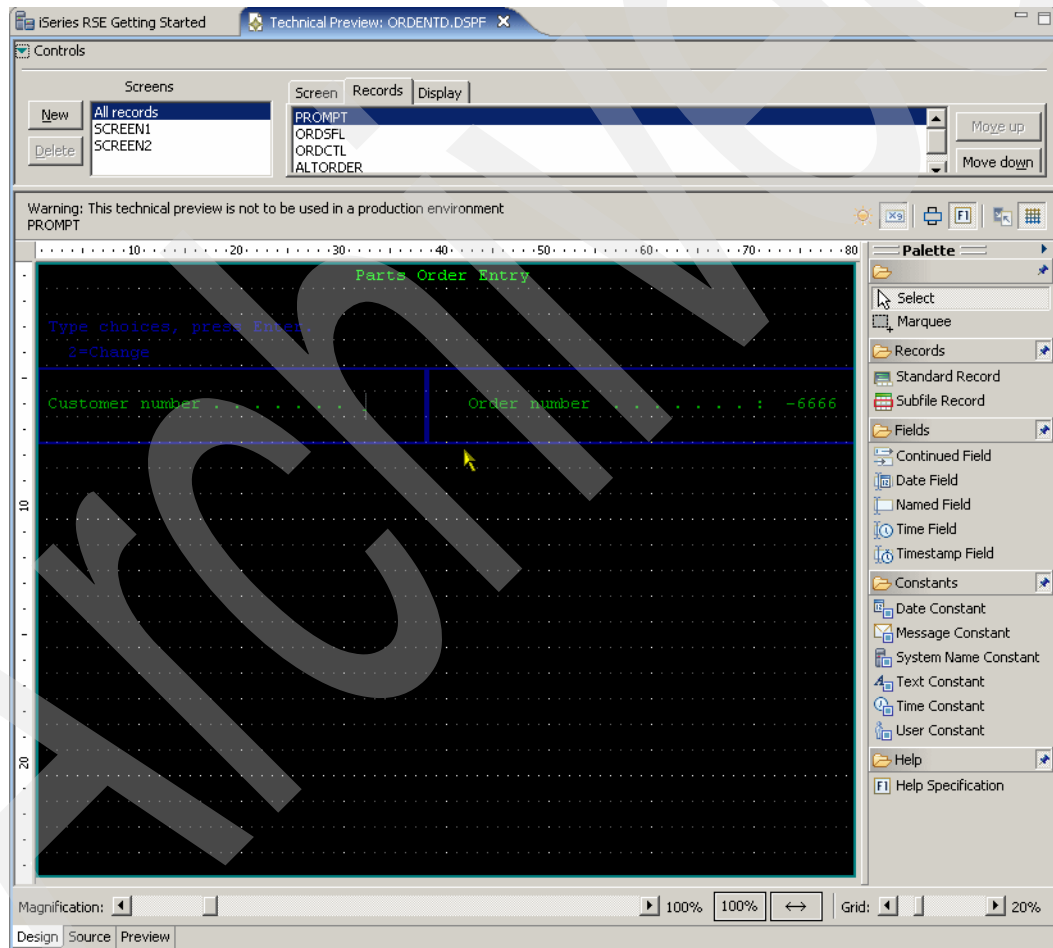


Figure 9-4 Showing the first record format of display file ORTDENTD in the Design area after starting Screen Designer

As you can see in Figure 9-4, any attempt is made to make the appearance in the Design area as close as possible to what the result would look like on a 5250 display. In addition to that, some aids are provided, such as a white grid and the rulers at the top and the left edge of the frame, which allow you to easily locate any given line and column number, or the blue frame above the mouse pointer to indicate the location of a help area.

Later in this chapter, in 9.5, “See everything at a glance and navigate quickly with the Outline view” on page 312, we describe how you can switch to a different record format and even how to see multiple formats at the same time in the Design area (in 9.4.7, “Seeing and editing your DDS statements in the Source page” on page 312). This section continues to explain some basics on how to use the Design area.

The Design area not only shows you how your 5250 panel will look at the end, you can also use it to navigate to each field of the format, add new fields, or change constants.

Select a field in the design area

To select any field, click anywhere in the area occupied by that field and frame delimited by eight small white squares, as shown in Figure 9-5.

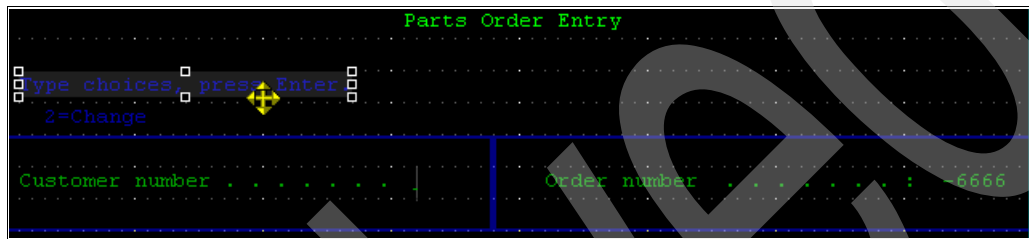



Figure 9-5 Constant field selected in the Design area

Screen Designer updates all related views immediately whenever changes are made to the Design page or even when you just select a field. That way you can use the Design area as a starting point to navigate to a certain field and then use the Properties view to work with it. Later in this chapter, in 9.6, “Working with detailed attributes in the Properties view” on page 314, we describe how you can change all of the definitions of the selected field.

Tip: If you have difficulties selecting a field, it may be caused by the fact that the field is located within a help area (the blue frame in the lower half of Figure 9-5). If you want to select such a field (the constant `Customer number` for example), you need to switch off the *Show rectangular help areas* setting by clicking the  icon in the toolbar on the right at the top of the design area. See 9.4.5, “Design Area Toolbar” on page 311 for more information.

You can also see the mouse pointer in Figure 9-5 changing its shape to move pointer whenever you touch one of the borders of that field. The shape of the cursor may be different from the one shown in Figure 9-5, depending on your workstation settings.

Moving a field using drag and drop

Once your mouse pointer is inside the selected field, you can move that field by keeping the left button depressed and moving the mouse to the desired target. As you can see in Figure 9-6, a dotted frame shows the new position and a little pop-up window specifies the exact coordinates.

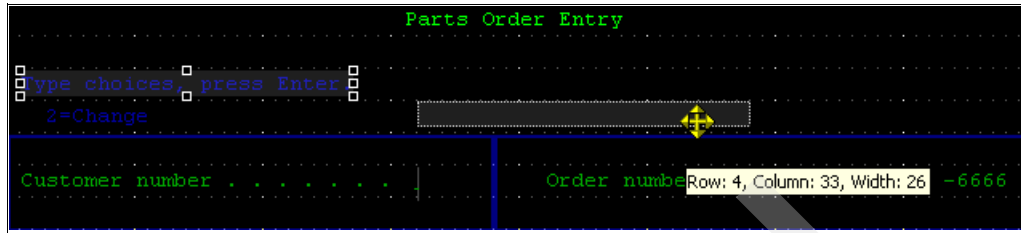


Figure 9-6 Moving a field using drag and drop

Tip: If you unintentionally moved a field, you can either press the Esc key as long as you have not released the mouse button yet, or “undo” the action by selecting **Edit** → **Undo** from the main menu, or simply pressing Ctrl+z.

Modifying a constant directly within the Design area

Constants may be modified without leaving the design area by clicking a selected constant. To do so, carefully perform the following steps:

1. Click the desired constant field.
2. After the eight small white squares show the limits of the field, click *once* again.
3. At this point do *not* enter any characters (unless you want to erase the entire text of the constant), instead either click a character you want to change or press the left or right arrow key to go there.
4. Type the new characters.
5. Press Enter to complete the change.

Note: you can delete (using the Del or backspace key) and then insert as many characters as you deleted, but you cannot increase the length of the field by inserting more characters.

9.4.2 Adding new items to your display file using the palette

To add any fields to your record format you have to use the *palette*. The Screen Designer design page provides an integrated palette. Alternatively, the Palette view of the Workbench may be used. While both provide the same functions, the Palette view provides more flexibility for resizing and moving it to different places, while the internal palette remains visible when you maximize the design page (by double-clicking the tab on top of it or pressing Ctrl+m).

Whenever the Palette view is closed, the Screen Designer's integrated palette is shown. Whenever the Palette view is opened, the Screen Designer's integrated palette is hidden. Figure 9-3 on page 298 shows on its right the Palette view of the Workbench, whereas Figure 9-7 shows the integrated palette on the right of the Design area.

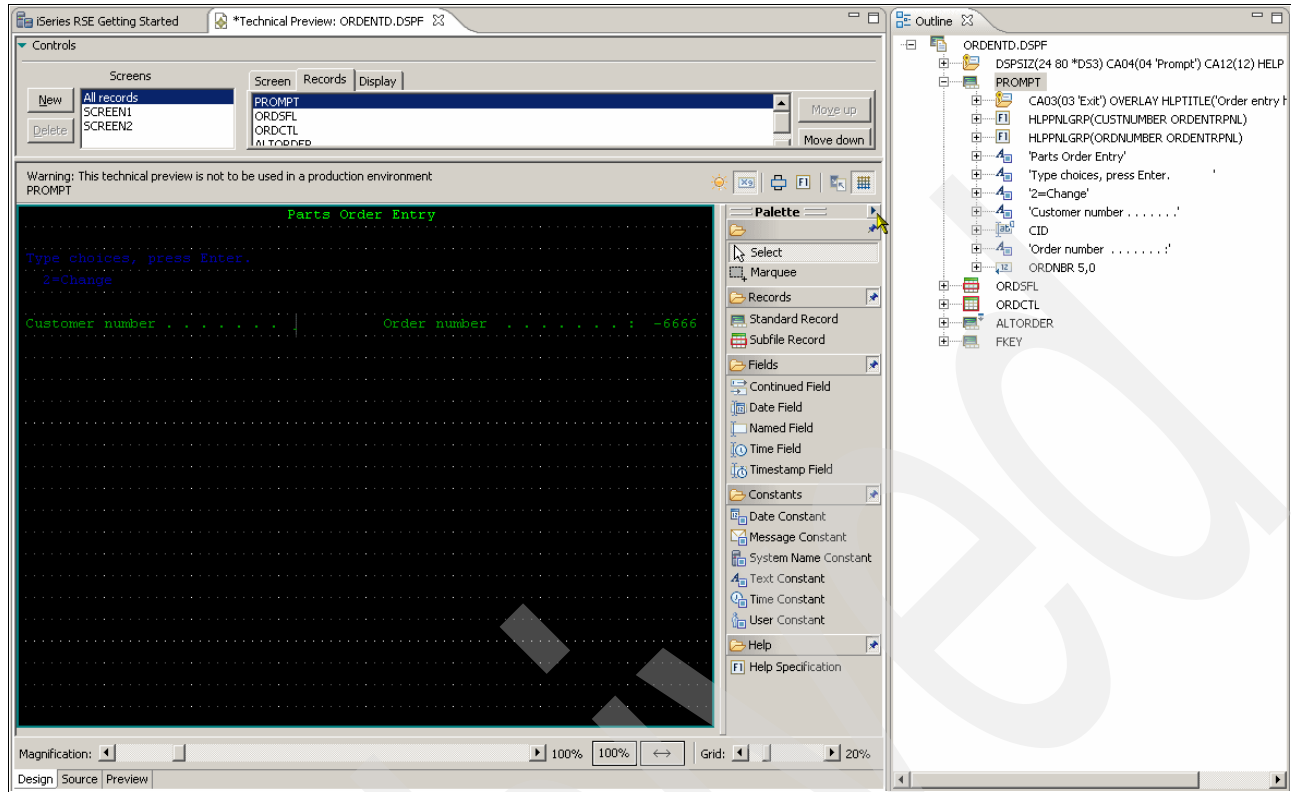


Figure 9-7 Design page with integrated palette

You can also “hide” the integrated palette by clicking the triangle on the right of its title bar (indicated by the mouse pointer in Figure 9-7). However, doing so will not hide the integrated palette, a vertical bar remains, allowing you to show it again by clicking the triangle once more.

As mentioned above, the purpose of both palettes—the integrated one as well as the Palette view—is to add new items to your display file. To do so, you use the same technique as some other graphical tools, the Page Designer for example. To add an item from the palette, you drag the button from the palette and drop it onto the design area, by performing the following steps:

1. Click the desired button in the palette.
2. Release the mouse button.
3. Move the mouse pointer over the design area where you want to place the new item.
4. Click the target.

Alternatively, you can also drag an item from the palette and drop it onto the design area. That is, not releasing the mouse button in step 2, but releasing it at the destination location.

While you move the mouse, the pointer changes its shape and an attached tool tip indicates the exact current position and the length of the field, as shown in Figure 9-8.

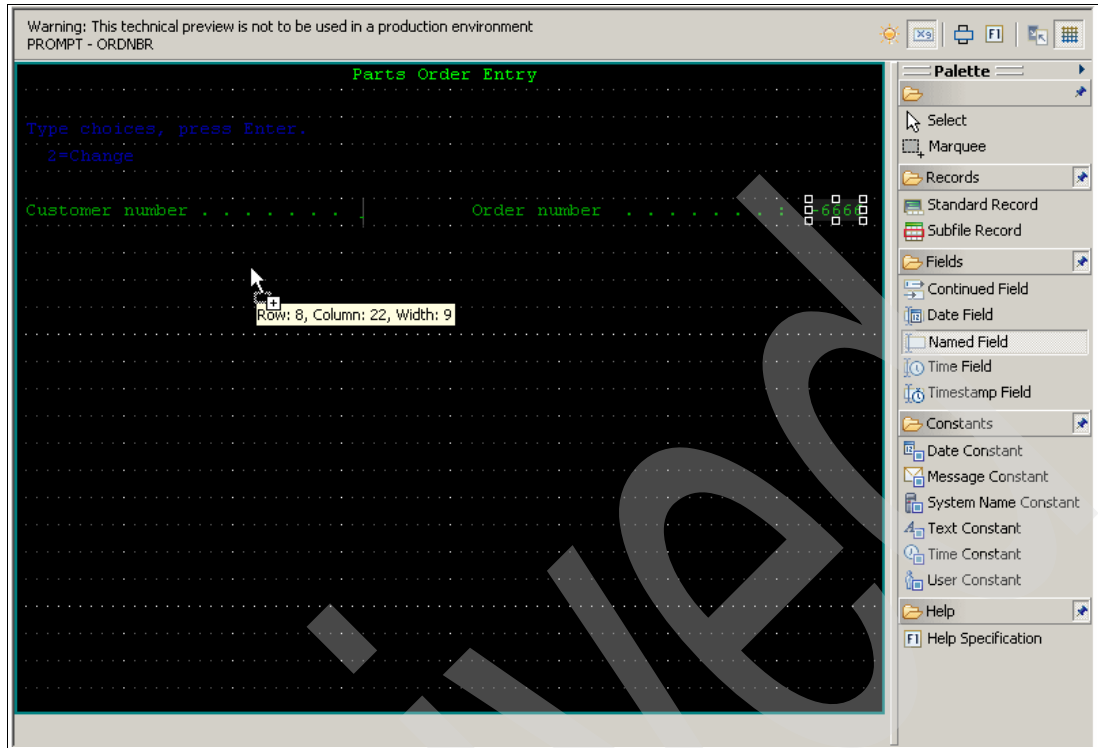


Figure 9-8 Dragging an item from the integrated palette to the design area

What type of items can be added using the palette?

The palette contains five *drawers*. Each drawer can be opened or closed by clicking its title bar. Figure 9-9 on page 304 shows the integrated palette next to the workbench palette view with all drawers open.

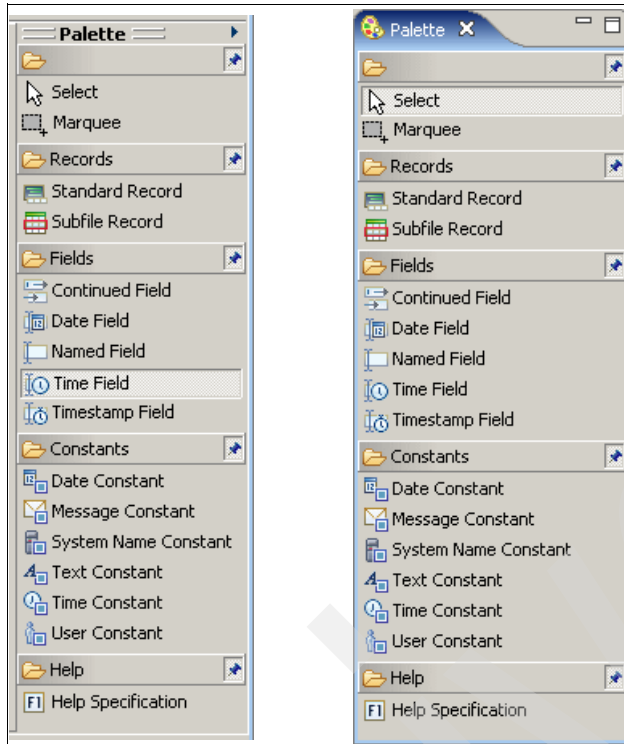


Figure 9-9 Screen Designer palette with all drawers open

The top drawer (with no name in its title bar) contains the following selection tools:

- ▶ **Selection** - The Selection tool is the default cursor tool used for selecting existing components in the Design area. You can select multiple components by holding down the Ctrl key.
- ▶ **Marquee** - The Marquee tool is a selection tool that you use to drag a rectangular selection area. All components that lie completely within the rectangle area are selected.

The other four drawers all contain items you can add to your display file:

- ▶ **Records** allows you to create:
 - Standard Record
 - Subfile Record
- ▶ **Fields** allows you to create:
 - **Continued Field** - which can wrap the user's input into several rows in a rectangular area on the design area.
 - **Date Field** - which allows you to store date information.
 - **Named Field** - that your program can read and update at runtime.
 - **Time Field** - which allows you to store time information.
 - **Timestamp Field** - which allows you to store timestamp information.
- ▶ **Constants** allows you to create:
 - **Date Constant** - a constant field which contains the job date.
 - **Message Constant** - which displays a message from a message file.
 - **System Name Constant** - a system constant which contains your i5/OS system name.
 - **Text Constant** - which displays a specified text string.

- **Time Constant** - to display the current system time as a constant (output-only) field.
- **User Constant** - which displays the user profile name for the current job as a constant (output-only) field.
- ▶ **Help** allows you to create:
 - **Help Specification** - to provide cursor-sensitive help for an area of the display by associating the area with text that is stored elsewhere.

Obviously, the items under Records allow you to add new records to the display file, while all other items in the drawers below add a field or constant to an existing record format.

Using the method described in 9.4.2, “Adding new items to your display file using the palette” on page 301, you can add any of those items to your display file. Later in this chapter, in 9.6, “Working with detailed attributes in the Properties view” on page 314, we describe how to specify the additional details for each record format and field.

9.4.3 Adding a database reference field to your record format

Besides dragging new fields from the palette, you can also create database *reference fields* by dragging them from the *System i Field Table View* and dropping them onto a record in the design area.

Reference fields are typically used to refer to the attributes of a field already defined in a physical or logical data base file (or a column in an SQL table) when defining a field for a display file or other file types. That way, you avoid having to repeatedly code length, type and other attributes of the field, and you ensure to keep these attributes consistent.

Opening the System i Field Table View

The System i Field Table View allows you to easily access the field definitions of a file from the Remote Systems view or from the Remote Systems LPEX editor. There are several ways to open *and* populate this view (just selecting **Menu** → **Show View** would open an empty view):

- ▶ From the Remote Systems LPEX editor, if you have an externally described file in your source, place your cursor at the location of the externally described file name and do one of the following:
 - From the workbench menu, click **Source** → **Show fields**.
 - You can also right-click inside your source and select **Show fields**.

The System i Field table view then opens and displays the fields from the externally described file. Note that the library where your externally described file resides must be in your library list, otherwise the view will be empty and you see message EVFR0026E in the LPEX message area. See 7.5, “Working with your library list” on page 204 for more information on how to set the library list.

- ▶ From the Remote Systems view or the System i Table view perform the following steps:
 - a. Right-click a physical or logical file object.
 - b. Select **Show in table** → **Fields**.
- ▶ Or, again from the Remote Systems view or the System i Table view, right-click a display file, a logical file, an externally described printer file, or a record format, and select **Show in table** from the context menu.

Note that when you select a display file, the Field Table View contains all fields of all record formats within that file. When you select an externally described printer file or logical file, you see a list of record formats. Clicking one of the record formats shows the fields within that format.

See “Field Table view” on page 216.

Dragging a field from the System i Field Table View

If you have the System i Field Table View open while working with the Screen Designer, you can easily create a new field in your display file by dragging it from the System i Field Table View and dropping it onto a record in the design area.

This is different from dragging an item from the palette—you need to keep the mouse button depressed while dragging,

9.4.4 Screen and record selection

Each display file contains one or more record formats which define how input fields, output fields, and constants are laid out on the 24 lines by 80 columns (or 27 by 132) 5250 display panel. While each format can define the entire panel, it is very common that a single format only occupies a portion of it and the whole display (as seen by the user) is built by combining multiple record formats.

The concept of a screen

When you design a display file, you can only work with a single record format at a time. However, when you do that, you still want to be able to see how the entire display will look. For this reason Screen Designer has the concept of a *Screen*, which is only a design time concept and defines a group of record formats. (CODE Designer uses a similar concept, but calls it a *Group*).

The All records screen

To work with this concept, you can define one or more Screens, each containing one or more record formats. Above the design area, you can see a Controls area for defining and managing screens. An **All records** screen, which contains all records, is always present and cannot be deleted. It is also the only screen shown when you open the DDS source member with Screen Designer for the very first time and have not changed it with CODE Designer before, as you can see in Figure 9-10 on page 307.

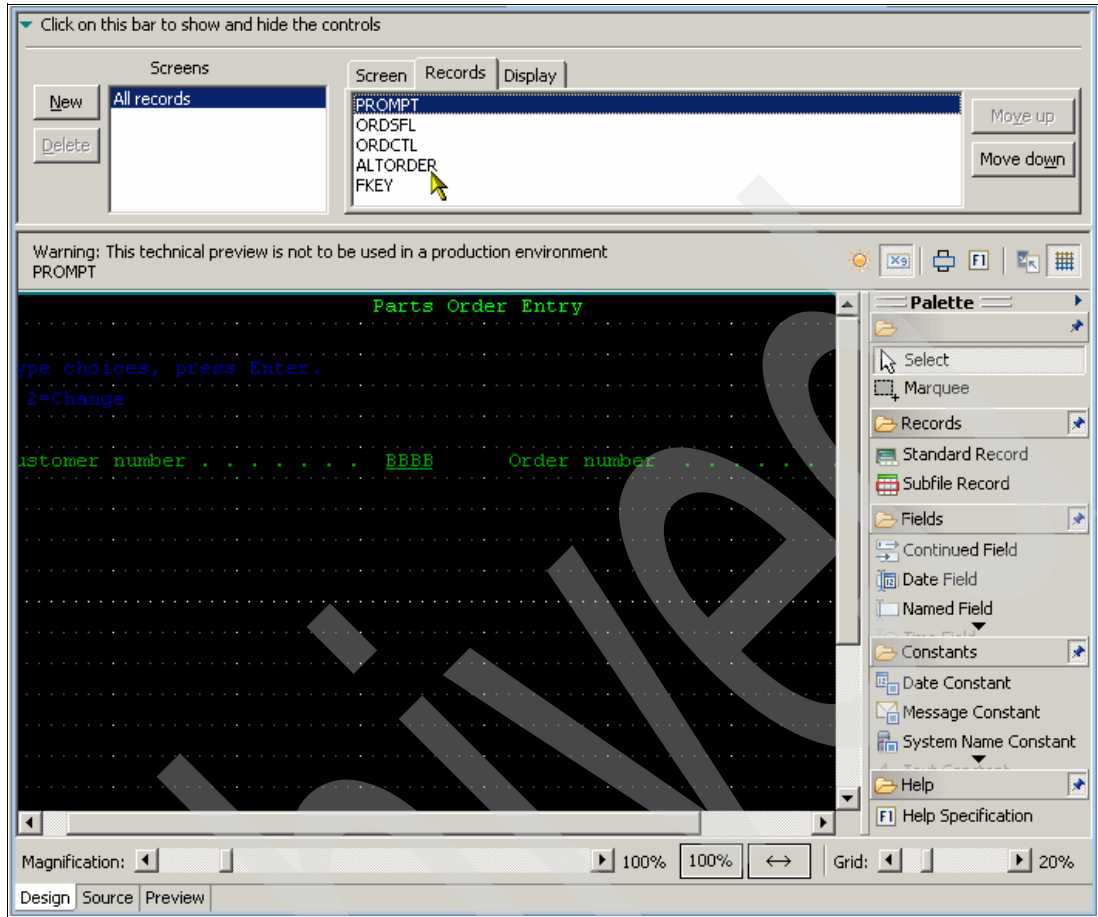


Figure 9-10 Initial look of the Controls area with only the All records screen

Using the Controls area

The Controls area is located at the top of the Design area, shown in Figure 9-10. It allows you to select which of the record formats (PROMPT in Figure 9-10) should be shown in the Design area. You can show or hide the Controls area by clicking the triangle at the upper left corner of it.

When you click any of the record formats listed in the Records list (see mouse pointer in Figure 9-10), that format will be shown in the Design area, as in Figure 9-11 on page 308.

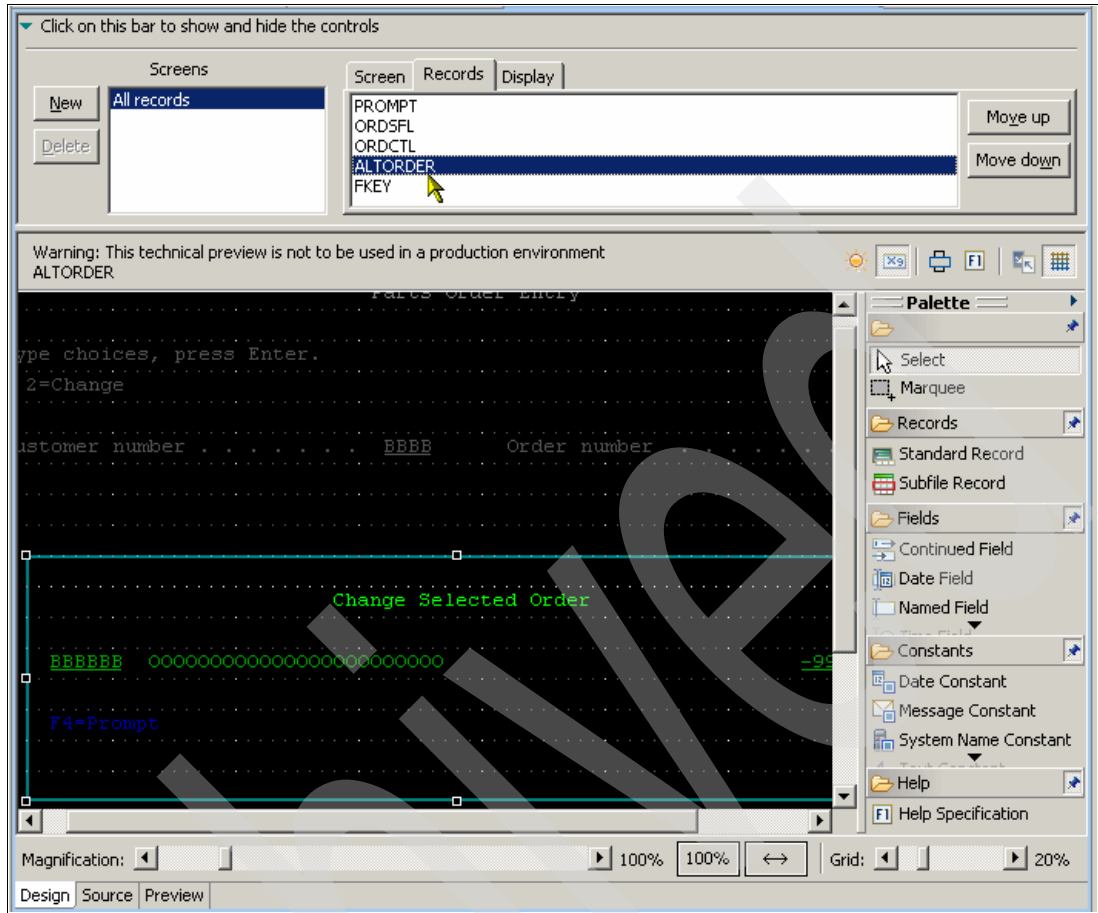


Figure 9-11 Design area with format ALTORDER selected

Create a new screen

To create a new screen, perform the following steps:

1. Click **New** at the left of the Controls area and specify a name, for example Screen1, in the Name field under the Screen tab.
2. Click the **Records** tab and you see the list of record formats again, as shown in Figure 9-12.

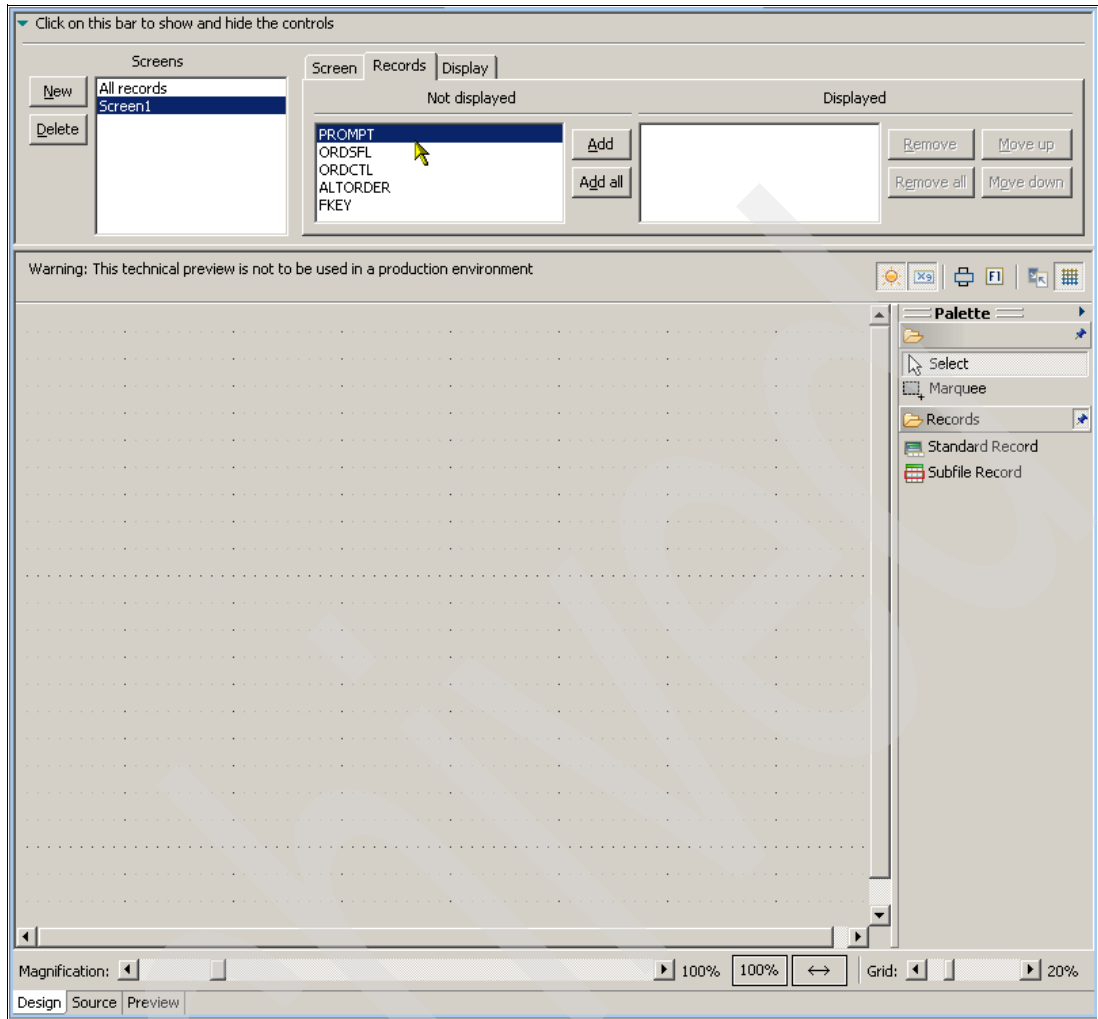


Figure 9-12 Select record formats for a new screen

3. Now click the name of the first record format you want to add to the screen.
4. Click the **Add** button.
5. Repeat steps 3. and 4. for each additional record format.

You can see the result of adding the PROMPT and FKEY format in Figure 9-13 on page 310.

If you add a subfile or subfile control record, the pair are both added at the same time.

All screen definitions except for “All records” are saved with the Display File as comments, for example:

```
A*%%GP Screen1 2
```

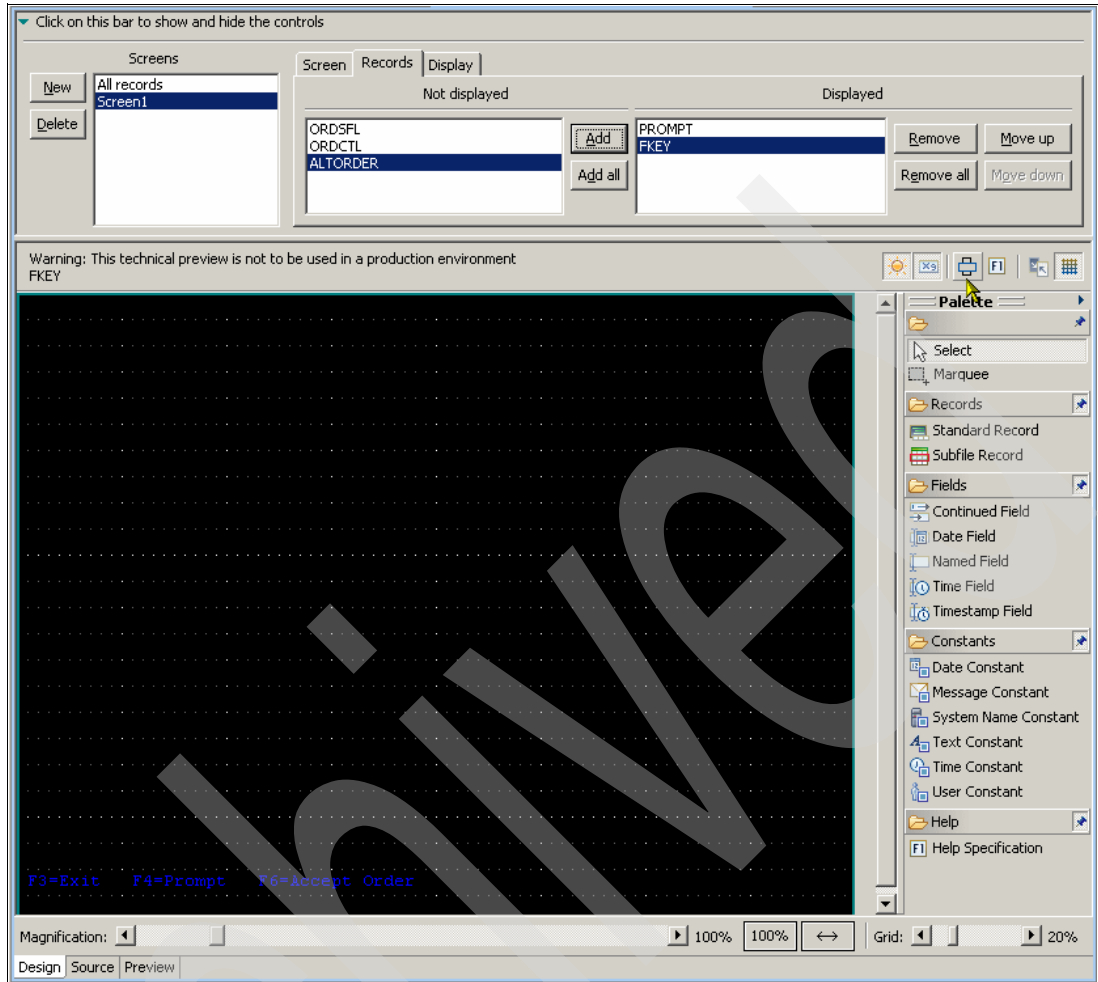



Figure 9-13 Screen 1 with two record formats

In spite of what we said earlier in “The concept of a screen” on page 306, by default you can see only a single record (FKEY) in Figure 9-13. You can switch it on or off by clicking the  button (see also at the mouse pointer in Figure 9-13) and make the selected record visually transparent, as shown in Figure 9-14 on page 311.

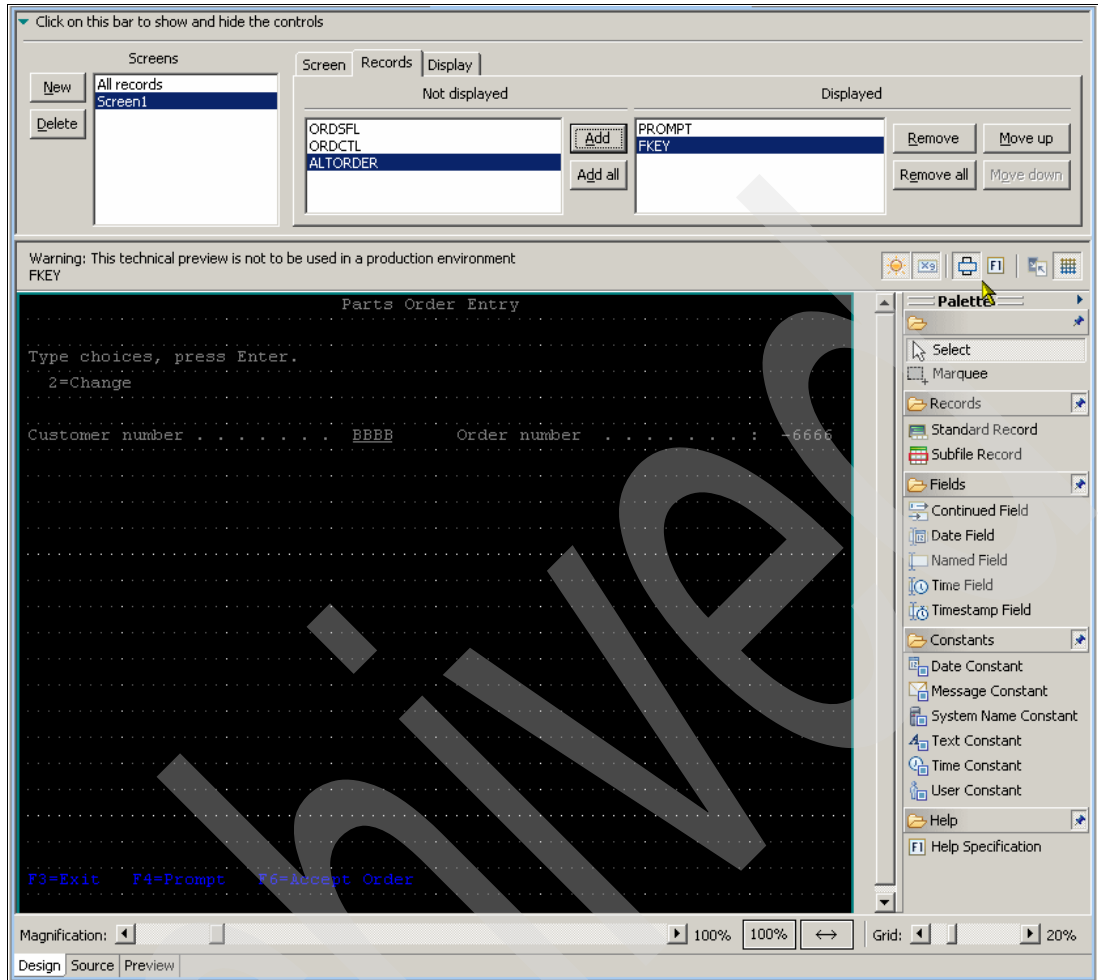


Figure 9-14 Make selected record visually transparent

As we said earlier in “The concept of a screen” on page 306, you can only change and work on a single record format at any time. The Controls area shown in Figure 9-14 is also used to select which record format you want to work on by clicking its name in the Displayed field.

9.4.5 Design Area Toolbar

Immediately above the design area, a set of toolbar toggle buttons, as shown in Figure 9-15, is provided (buttons described from left to right):

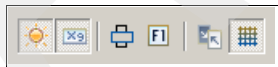


Figure 9-15 Design Area Toolbar

- ▶ Display normal intensity text as highlighted text in the design area.
- ▶ Show sample data in fields.
- ▶ Make the selected record visually transparent so fields of other records belonging to the current screen show as well (described above with Figure 9-13 on page 310).
- ▶ Show rectangular help areas, so that fields under the help area may be selected.
- ▶ Show the design area in black and white mode.

- ▶ Show the grid.

Tip: Switching off the *Show rectangular help areas* setting can be very important. Otherwise you will not be able to select any of the fields within that area.

9.4.6 Changing the magnification of the design area and the grid brightness

When you enlarge or reduce the size of the editor page, the design area is not adjusted to the new size, but cropped with scroll bars at the right and the bottom or it covers only a part of the area. However, you can change the magnification of the design view to either adjust it to the pane size or specifically enlarge a certain portion of it.

You can do so by using the controls below the design area. There is also a scroll bar to change the brightness of the grid. The text displayed at all magnification settings is dependent on the font size that is chosen in the preferences page.

9.4.7 Seeing and editing your DDS statements in the Source page

So far we only talked about using the WYSIWYG Design area to work with your display file. At any time, you can also switch to the Source page to see or edit the DDS statements directly.

What you can do in the Source page is exactly the same as what is possible when you open the source member with the Remote Systems LPEX editor. However, the big advantage in using the Screen Designer is the *combination* with the Design area, as well as the Properties view (as we describe in 9.6, “Working with detailed attributes in the Properties view” on page 314).

By “combination” we mean two things:

- ▶ Any change in either the Source page, the Design page, or the Properties view is immediately reflected in the other views and editors.
- ▶ When you select a field in the Design area and then switch to the Editor page, the cursor will be positioned at the line defining that field. When you click a certain line of the source member in the editor and switch to the Design area, the corresponding field is selected there. In both cases, the contents of the Properties view are updated as well.

9.5 See everything at a glance and navigate quickly with the Outline view

The Outline view acts as an excellent navigation tool when you want to edit DDS source (as well as ILE RPG, COBOL, or CL) in the Remote Systems LPEX editor. It displays a structural outline of record formats, fields, keywords and parameters of the display file you are currently working with.

While the Outline view is also available when you use the Remote Systems LPEX editor, it is particularly useful when you work with the Screen Designer. You can expand the file structure in the Outline view, and click various elements in the view to jump to that location in the source itself or highlight the corresponding field in the design area, whichever is currently visible in your workbench, provided it is part of the currently displayed screen (see 9.4.4, “Screen and record selection” on page 306). At the same time, the content of the Properties view is updated accordingly.

Unlike the Outline view for other languages, such as RPG, COBOL or CL, which provides a more or less high-level overview, you see keywords and parameters in the Outline view for a DDS source member.

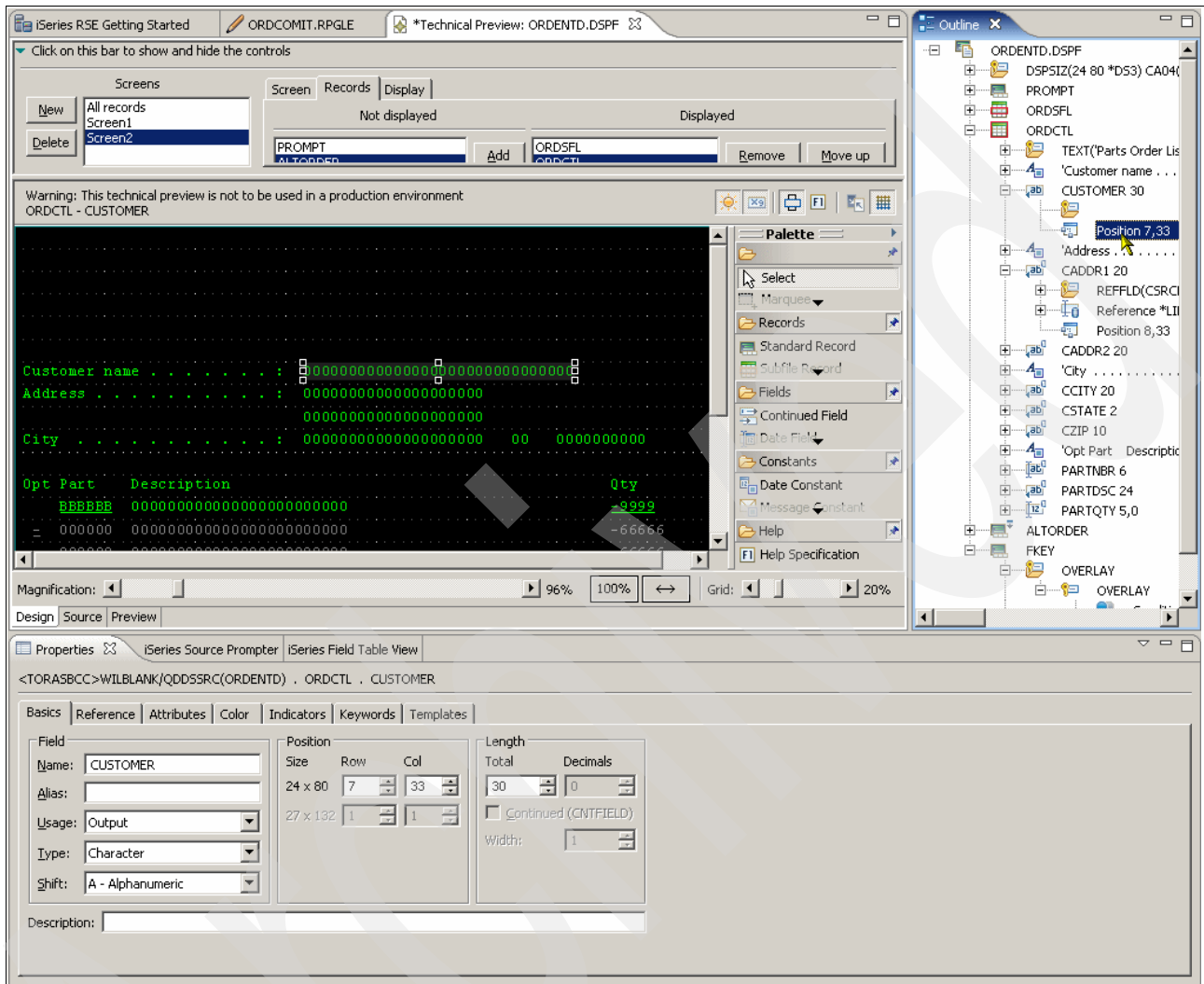


Figure 9-16 Outline view in Screen Designer

Tip: If you do not see the Outline view (by default to the right of the Palette) because you previously closed it, there are three possibilities to restore it:

- ▶ Select **Window** → **Reset Perspective** from the main menu. This will reset the default layout of the Perspective.
- ▶ Select **Window** → **Show View** → **Other** → **General** → **Outline**. This will open the Outline view regardless of which perspective you currently see.

By default the Remote System Explorer and the System i Projects perspective show the Outline view in the upper right pane, whereas the Screen Designer perspective has it in the lower left.

9.6 Working with detailed attributes in the Properties view

The Properties view displays the current attributes of a file, record format, field and keyword, and may also be used to modify those attributes. Those attributes relate directly to the keywords, parameters and value you can code in the DDS source statements to describe the appearance and behavior of the 5250 display. Any changes made to the Properties view are automatically reflected in the design area, in the Outline view, and in the DDS source code of the Source page.

9.6.1 Changing field attributes

Figure 9-16 on page 313 shows the Properties view for the CUSTOMER field, whereas Figure 9-17 shows the same entry with the Source page visible.

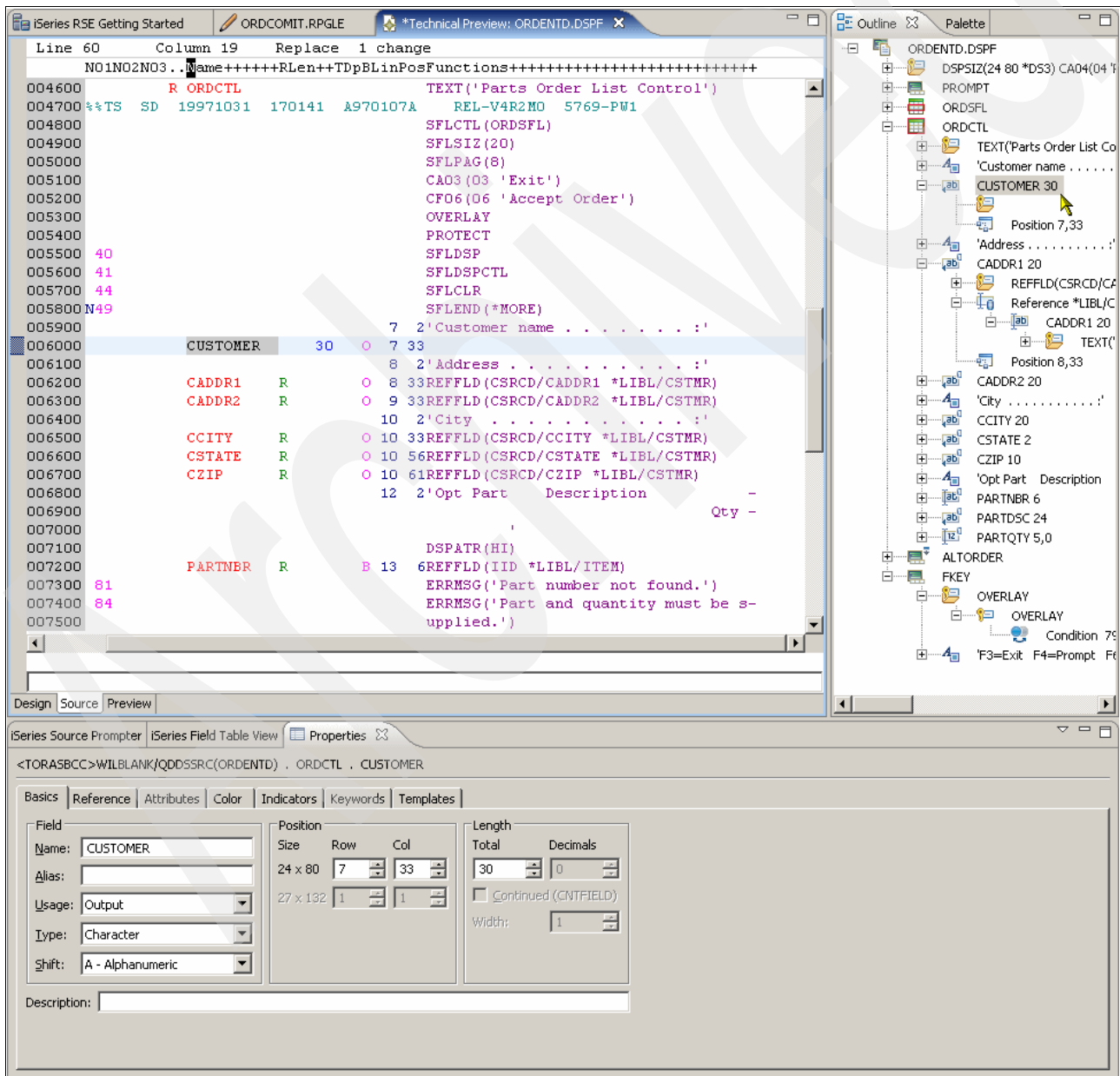


Figure 9-17 Properties, Source page and Outline view

You can click any of the tabs in the Properties view to change the values of the DDS statement.

9.6.2 Changing file attributes

The property pages not only allow you to change or add keywords and values for DDS statements defining fields, but also for record formats and the entire display file. To change or add those parameters, you need to use the Outline view to navigate to the appropriate properties.

For example, if you wanted to change the primary and secondary screen size (which is an attribute of the entire display file), do one of the following:

- ▶ Click the root of the Outline view showing the name of the display file and then click the Display Size tab.
- ▶ Expand the branch directly under the root of the Outline view, click the DSPSIZE(24 80 *DS3) node representing the DSPSIZE keyword. The Display Size tab in Outline view opens automatically, as shown in Figure 9-18.

You can now change the size of the primary display or add a secondary display.

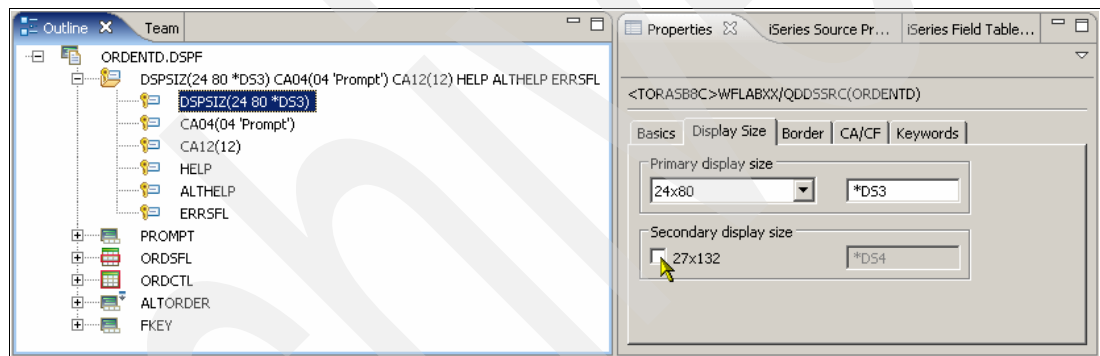


Figure 9-18 Changing primary or secondary display size

9.6.3 Working with reference fields

Under the Reference tab, you can also add or change database or source references. When you open that tab, you have the choice of selecting one of the following as Reference type:

- ▶ **None** means all attributes of the field must be defined in this DDS.
- ▶ **Database reference** refers to definition in another file.
- ▶ **Source reference** refers to another field within the same DDS source.

When you select **Database reference**, you can click **Browse** (see mouse pointer in Figure 9-19 on page 316) to find the reference file on your i5/OS system.

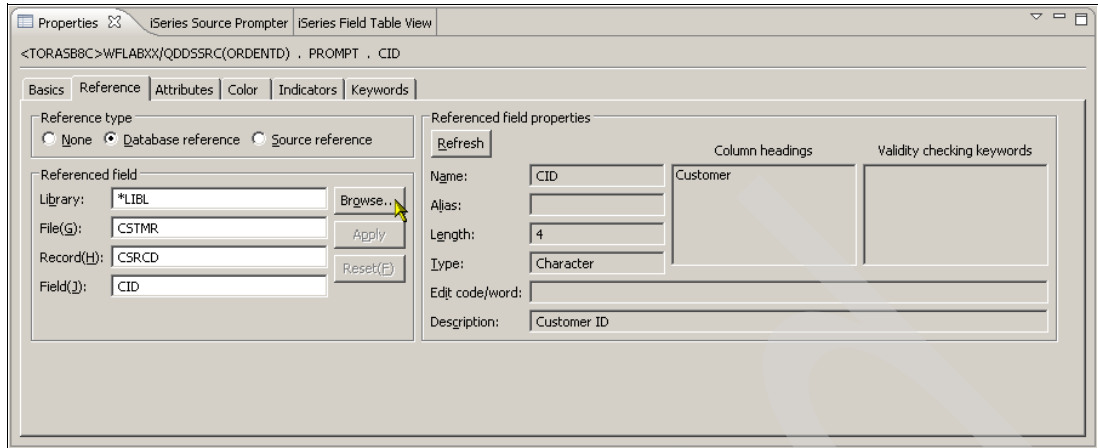


Figure 9-19 Defining a referenced field

9.6.4 Using a properties template

If the same attributes need to be defined for several fields, you can create a template based on the definition of one field and apply the template field to others, as shown in Figure 9-20.

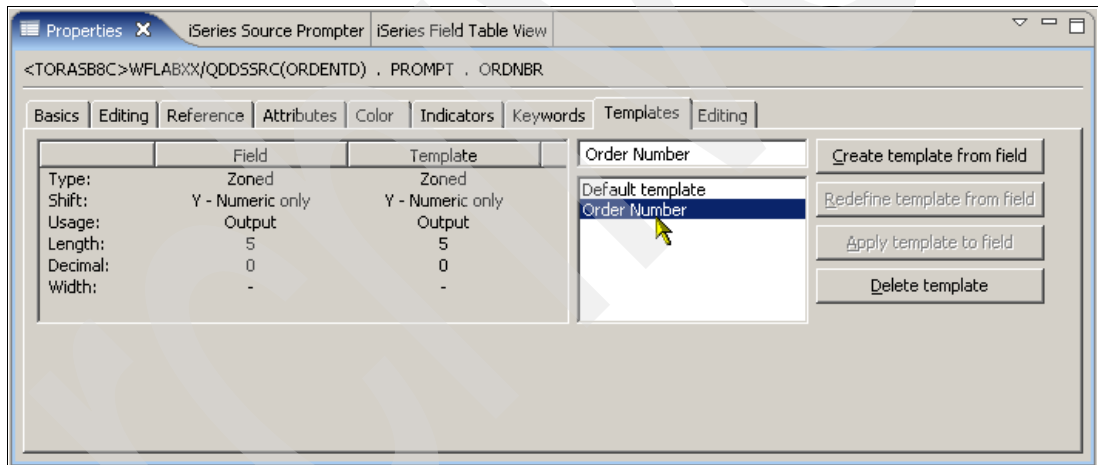


Figure 9-20 Create a template from a field

For complete information on DDS for display files, see **Help** → **Help Contents**, then:

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Reference**
- ▶ → **Language Reference**
- ▶ → **System i programming information**
- ▶ → **Data Definition documentation**
- ▶ → **DDS for Display Files**

9.7 Setting your preferences

Screen Designer provides a Preferences page that may be accessed through the Remote Systems view menu **Preferences** → **Screen Designer** or by selecting **Window** → **Preferences** from the main menu and then **Remote Systems** → **Screen Designer**.

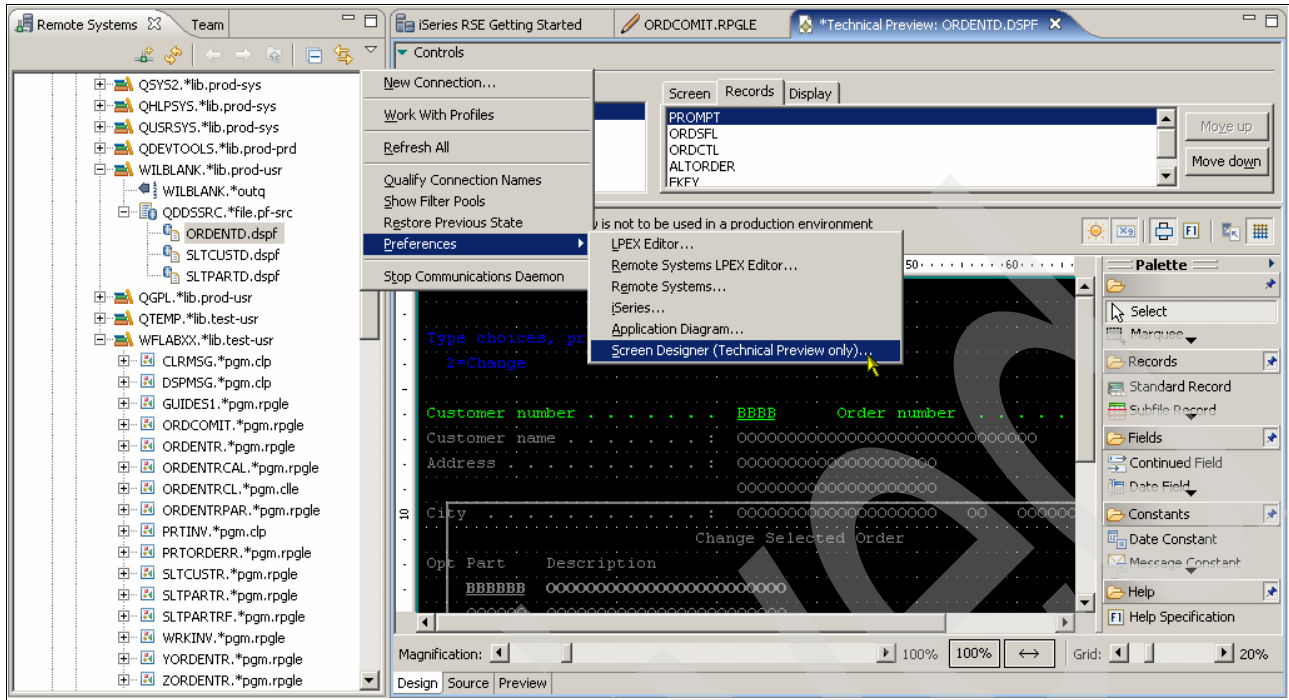


Figure 9-21 Accessing the Screen Designer preferences from the Remote Systems view menu

You can choose a new default for (see Figure 9-22 on page 318):

- ▶ The background color; the initial value is black.
- ▶ Whether or not to show rulers at the left and on top of the Design area.
- ▶ The font for constants and sample values in the fields; the default is Courier New with size 10.

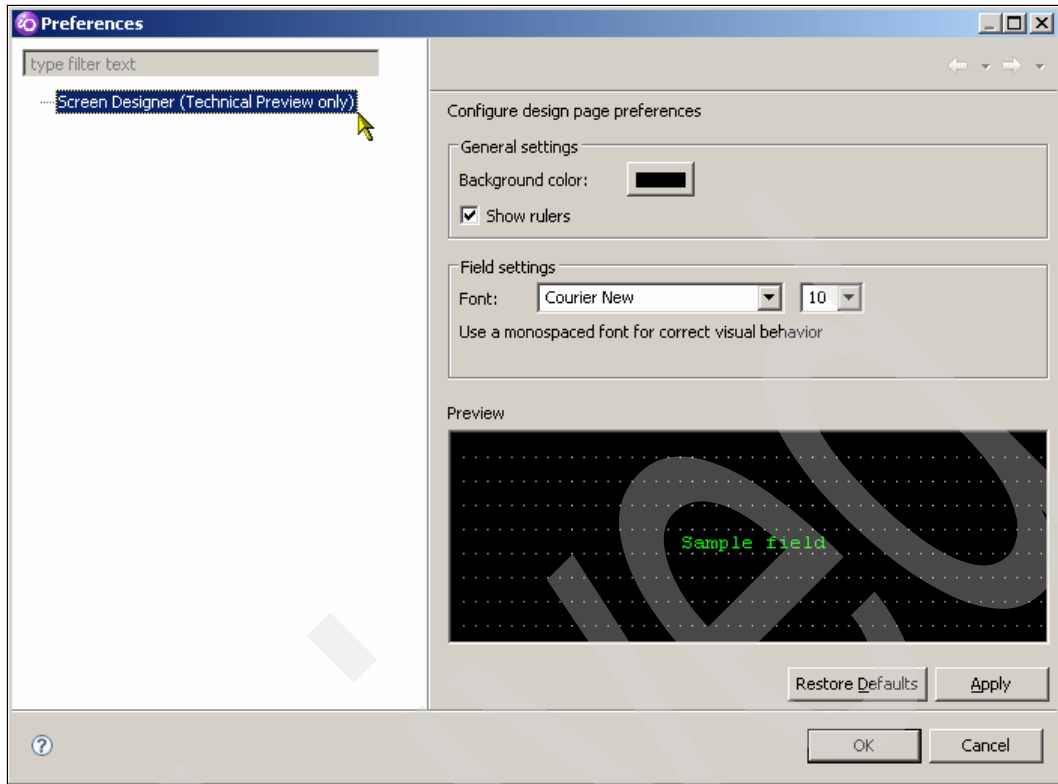


Figure 9-22 Screen Designer preferences

Application Diagram in WDSC AE - visualize program structures

The *Application Diagram*, one of the enhancements which came with WebSphere Development Studio Client Advanced Edition for iSeries V7.0, provides a graphical view of the different resources in an i5/OS application and their relationships to each other. There are two different diagrams that you can look at in the Application Diagram view:

- ▶ The Source Call Diagram
- ▶ The Program Structure Diagram

The Source Call Diagram takes ILE RPG and ILE COBOL source as input and displays a call graph showing subroutine and procedure calls. The Program Structure Diagram takes program and service program objects as input and displays the binding relationships between them as well as the modules bound into each program and service program.

Advanced In Version 7.0, this capability is only provided with WebSphere Development Studio Client Advanced Edition for iSeries (WDSC AE 7.0).

References to more Information

You can find additional information about the Application Diagram in the online help. From the menu select **Help** → **Help Contents**, then expand

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Developing System i server applications using Remote System Explorer**
- ▶ → **Building a graphical overview of your i5/OS application.**

There is also a tutorial which you can find under **Help** → **Tutorials Gallery** and then expand

- ▶ **Do and Learn**
- ▶ → **System i**
- ▶ → **Developing System i server applications using Remote System Explorer**
- ▶ → **Understand programs using the Application Diagram Viewer**

Note: you need to have at least fixpack 7.0.0.2 of WDSC AE 7.0 in order to get the tutorial. You may either follow the online tutorial or download and print a PDF attached to the first page. If you do the latter, make sure to read and perform at least the steps described under

Introduction → **Prerequisites** → **Restore the WFLABXX savefile (WFLABXX.savf) on an System i system** → **Import the WFLABXX project to your workspace**, since it provides the sample application that you can install onto your own System i.

Archived

10.1 The Application Diagram

When you need to change an existing i5/OS native application, you can gain a better understanding of the connections between resources by viewing a graphical representation rather than looking at source code. When you create a new application from scratch, you can also create diagrams for documentation purposes.

If you had used CODE or CODE/400 before, you can look at the Application Diagram as a replacement for the CODE Navigator. However, it is designed to be much more.

The Application Diagram Viewer helps you navigate easily among related i5/OS resources to allow viewing and editing in the Remote Systems LPEX editor. This includes a diagram to display the call graph for ILE RPG and ILE COBOL resources and a diagram to show the binding relationships for program and service program objects.

An application diagram is made up of nodes and connections. As shown in Figure 10-1, nodes represent different resources in your application; connections represent the relationship between those resources.

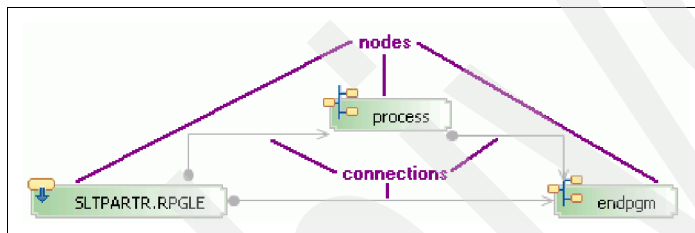


Figure 10-1 A simple Source Diagram with nodes and connections

There are two types of diagrams:

- ▶ A Source Call Diagram visualizes the structure of subroutine and procedure calls.
- ▶ A Program Structure Diagram shows you the ILE binding relationships between programs and service programs.

You can build a Source Call Diagram and Program Structure Diagram based on multiple source members and ILE programs. The following sections provide a brief overview of the two types of diagrams.

10.1.1 Source Call Diagram to analyze ILE RPG or ILE COBOL sources

The Source Call Diagram takes ILE RPG and ILE COBOL source as input and displays a call graph showing subroutine and procedure calls. Figure 10-2 on page 322 shows an example of a Source Call Diagram for a single RPG source member. Note, however, that a diagram can show the structure of multiple source members, as well as a combination of Source Call and Program Structure Diagrams.

Each node in Figure 10-2 represents a subroutine or procedure, whereas the arrows (“connections”) represent the calling relationships between them. See 10.3.1, “Understand the Application Diagram Viewer legend” on page 328 for information on the meaning of the various icons.

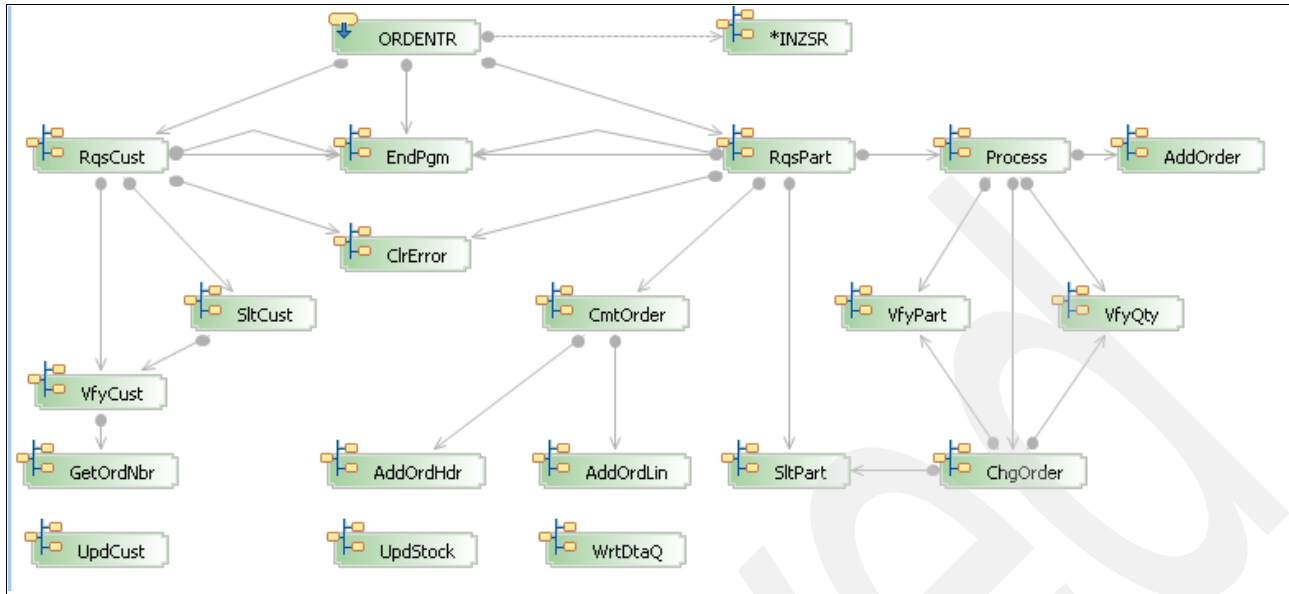


Figure 10-2 Example of a Source Call Diagram

You may see the Source Call Diagram as yet another way to visualize the structure of an ILE RPG or ILE Cobol program, going one step further than the Outline view of the Remote Systems LPEX editor. See 8.1.4, “Outline views” on page 273 for more information about using the Outline view.

Tip: The Application Diagram uses the LPEX parser associations for the ILErpg, ILErpgSql and ILEcobolSqlCics parsers to determine if a selected source member, IFS file, System i project member or local file is either ILE RPG or ILE COBOL. By default the standard types (RPGLE, SQLRPGLE, CBLLE, and SQLCBLLE) are associated with one of these parsers.

If you are using a non-standard type, then the Visualize Application Diagram action will not appear in the pop-up menu for your source members until you associate the type with one of the above three parsers. You can do this using the **LPEX Editor** → **Parsers** → **Parser Associations** preference page.

Note: When multiple local files in your Remote Systems view (that is, files on your PC disk drive) are scanned, the Application Diagram will not find calls between different files since it only looks for /COPY members on the server. However, this does not apply for local System i project members. Also, if ILE RPG subroutines, procedures or ILE COBOL procedures do not start and end in the same source member, calls from them might not be shown correctly in the Application Diagram or the properties of the calls may be incorrect.

10.1.2 Visualize ILE program dependencies with Program Structure Diagrams

The Program Structure Diagram takes program and service program objects as input and displays the binding relationships between them as well as the modules bound into each program and service program. Figure 10-3 on page 323 shows an example of a Program Structure Diagram showing six program and six service program objects, each containing one module, except SRVPGM2, which contains two modules.

The arrows (“connections”) represent the ILE binding relationships between programs and service programs. When you click one of the nodes the color of the connections to (“used by”) and from (“using”) connections changes to red and green, respectively, as you can see in

Figure 10-3. See 10.3.1, “Understand the Application Diagram Viewer legend” on page 328 for information about the meanings of the various icons.

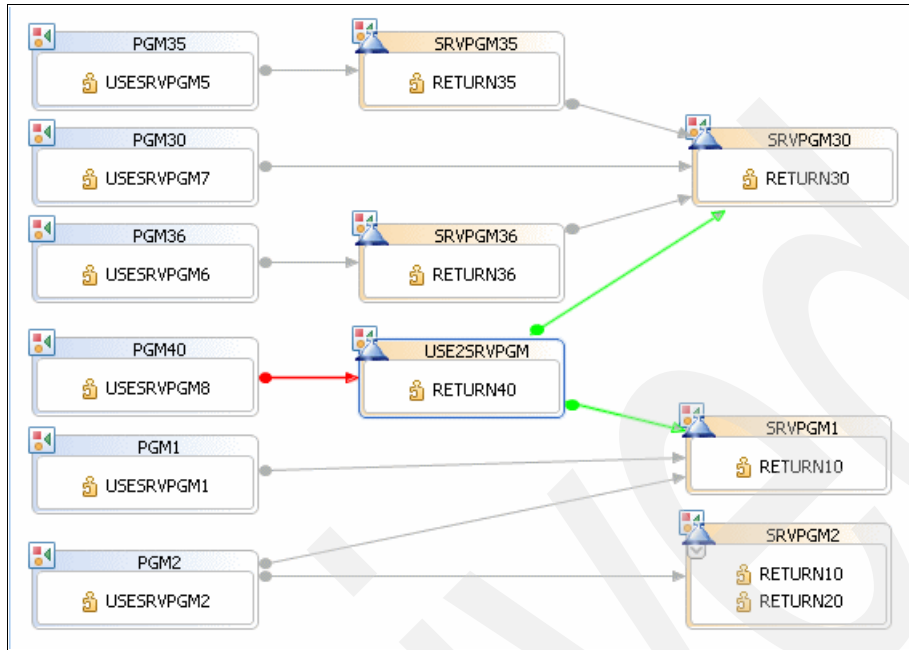


Figure 10-3 Example of a Program Structure Diagram

10.1.3 The Application Diagram Viewer

To actually see and work with Source Call and Program Structure Diagrams, you use the Application Diagram viewer. It opens up in the editor area of the workbench. The viewer consists of a main canvas, a palette, as well as a Properties view and an Outline view. Note that the content of this Outline view is *not* the same as the Outline view of the Remote Systems LPEX editor (described in 8.1.4, “Outline views” on page 273).

The Application Diagram Viewer does not have its own perspective, but rather creates canvas and palette in the editor area and shows new content in the Outline and Properties views.

The main canvas shows either the Source Call Diagram for one or more source ILE RPG or ILE Cobol members, the Program Structure Diagram for typically multiple program and service program objects, or both, as described on the previous sections.

In Figure 10-4 on page 324 you can see an example of Application Diagram Viewer showing a Source Call Diagram as well as Program Structure Diagram. The following sections describe in more detail how to create and work with Application Diagrams.

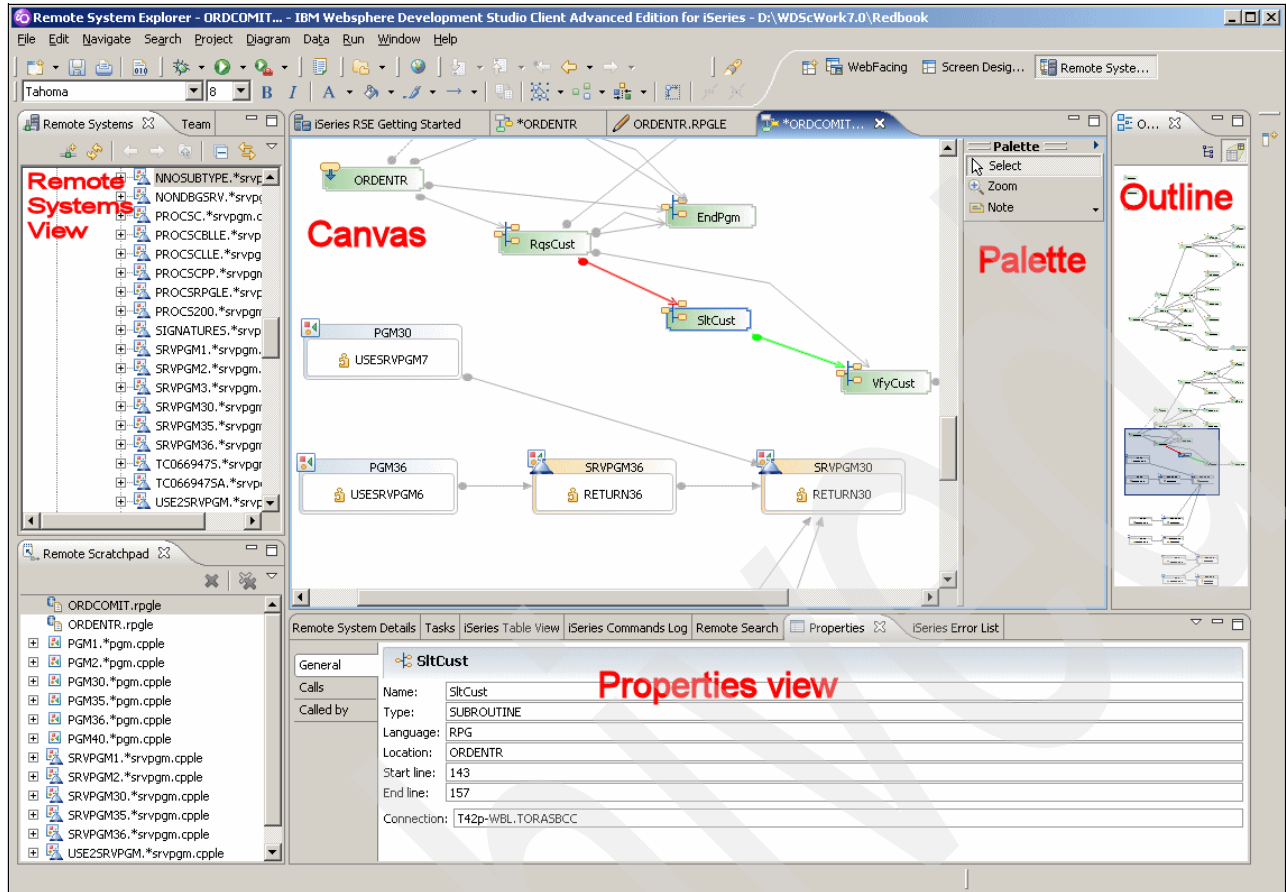


Figure 10-4 Components of the Application Diagram Viewer within the RSE perspective

Note: In Figure 10-4 the Properties view is shown in the lower right pane. By default, it appears in the lower left pane of the Remote System Explorer perspective, which usually is much smaller, as the one on the right. In 10.3.4, “Get details from the Properties view” on page 335, we show you how to move the view to a different location.

10.2 Ways to create an Application Diagram


There are several possibilities to create and open an application diagram:

- ▶ In the Remote Systems LPEX editor, when editing source code
- ▶ From the Remote System Explorer perspective
- ▶ From the System i Projects perspective

The following sections describe each method in more detail.

10.2.1 Create an Application Diagram while editing a source member

Launching from the Remote Systems LPEX editor is useful when you want to edit the source and view its application diagram at the same time. To do so, make sure the editor area has current focus (the title tab shows in blue rather than in white) and the ILE RPG or ILE Cobol source member you want to create a diagram for shows in the editor. If the title tab shows in white, click in the editor area to bring it in focus.

Then select the **Source** → **Visualize Application Diagram** action as shown in Figure 10-5 on page 325. You can also click the tool bar icon .

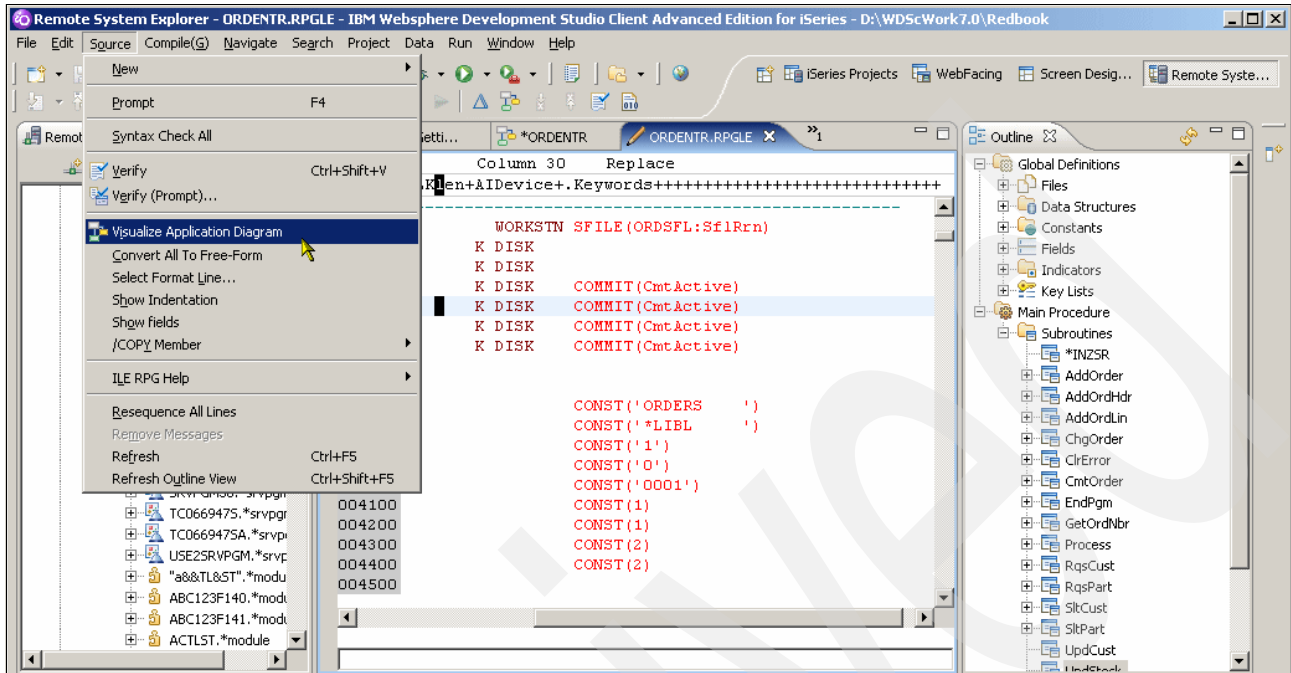


Figure 10-5 Create an Application Diagram while editing a source member

If you create an Application Diagram while editing a source member as described above, you can only show the structure of a single ILE RPG or ILE COBOL program.

Note: If you build the Application Diagram as described here, it uses the *current* contents of the editor whether or not any changes have been saved. This is a different behavior than launching from the RSE view or System i projects (where only saved changes are used but you are prompted to optionally save any open editors first).

A progress bar similar to the one in Figure 10-6 shows that the diagram is being built.

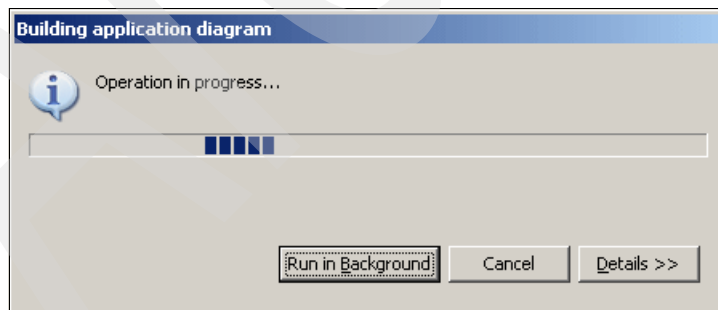


Figure 10-6 Building application diagram progress bar

10.2.2 Using the Remote Systems view

If you want to create a Source Call Diagram of more than one source member or if you want to build a Program Structure Diagram (based on program and service program objects), you can use the Remote Systems view to do so. This action appears on the context menu when you do a right mouse click for the following resources:

- ▶ ILE RPG and ILE COBOL source members
- ▶ IFS files and local files
- ▶ IFS folders (all ILE RPG and ILE COBOL source files in the folder are included)
- ▶ Local folders (all ILE RPG and ILE COBOL source files in the folder are included)
- ▶ Member filters
- ▶ Program and service program objects
- ▶ Object filters

For all of the above resources, so you can select more than one item by holding the Ctrl or Shift key and clicking each resource before performing the right mouse click. For example, select five ILE RPG source members as well as two program objects and four service program objects. Those resources can also reside in different libraries or even on different i5/OS systems.

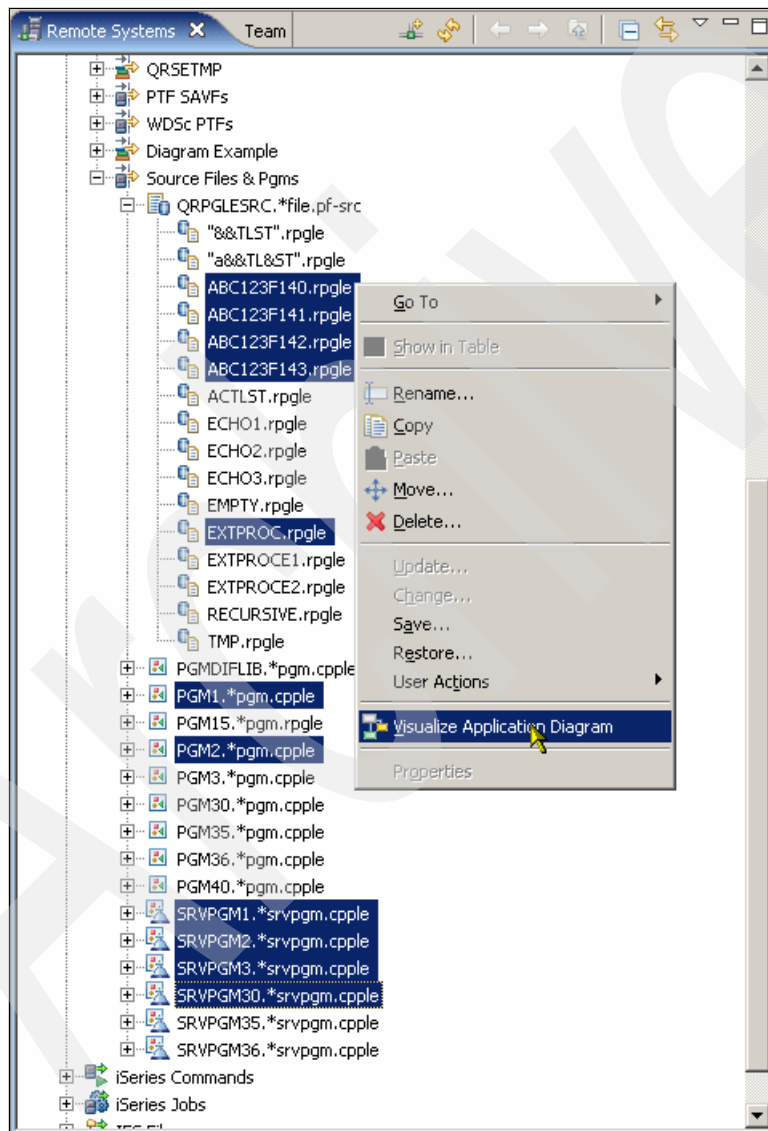


Figure 10-7 Selecting multiple resources for an Application Diagram

After selecting **Visualize Application Diagram**, you see a progress bar similar to the one in Figure 10-6 on page 325, showing that the diagram is being built. You can click the **Details >>** button and watch which steps are performed, as shown in Figure 10-8 on page 327.

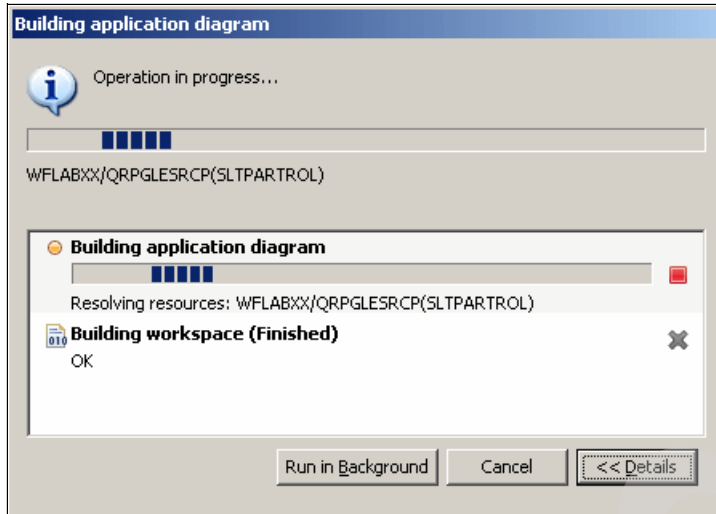
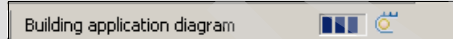


Figure 10-8 Building application diagram progress bar - Details

Tip: To continue working while the diagram is being built, click **Run in Background**. It is useful when an application diagram takes a long time to build due to the number of resources to work on. A moving blue bar in the lower right of the perspective indicates that it is still active:



If you selected more than ten members or program objects, a warning (INAVL1005Q) will display and give you the choice to cancel the process before it starts, as shown in Figure 10-9. The reason is that the diagram would take a long time (and a lot of CPU) to lay out, because the algorithm tries to create a diagram such that all connections run in one direction and do not overlap or run through other nodes.

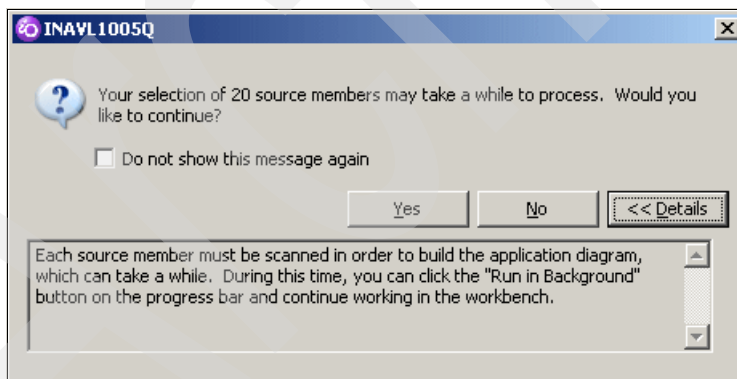


Figure 10-9 Warnings when the diagram tends to become very big

A second check occurs after the diagram is built before it is opened and a message (INAVL1007Q) similar to Figure 10-10 on page 328 is displayed if the number of nodes and connections exceeds 500.

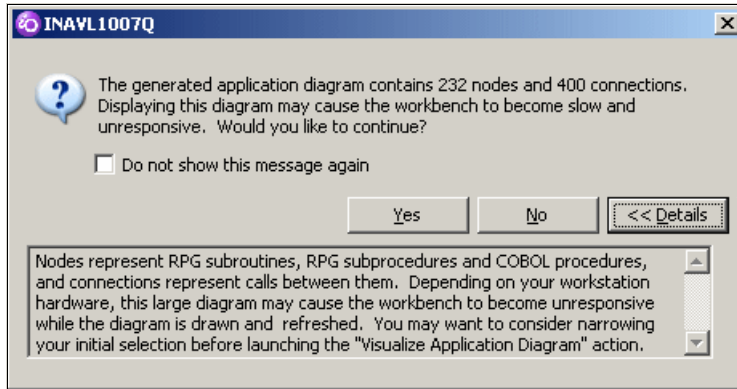


Figure 10-10 More than 500 nodes and connections (INAVL1007Q)

10.2.3 Creating an Application Diagram from a System i Project

Similar to what we described in 10.2.2, “Using the Remote Systems view” on page 325, you can also create an Application Diagram based on a System i Project. Since you can store source members or save files in a System i Project, you can only create Source Call Diagrams.

Besides selecting **Source** → **Visualize Application Diagram**, while you edit a source in the Remote Systems LPEX editor, you can also select that option from the context menus from:

- ▶ One or more source members
- ▶ One or more source files
- ▶ One or more entire System i Projects

Note that selecting multiple source members might take a very long time as described for Figure 10-9 on page 327.

10.3 Working with an Application Diagram




Now that you created an Application Diagram, it is being displayed in the Application Diagram viewer, similar to Figure 10-5 on page 325. Besides analyzing it visually, you have several possibilities now to manipulate and work with it. This section describes what exactly you can do. We start by looking at the meaning of the various graphs and icons.

10.3.1 Understand the Application Diagram Viewer legend

Besides the symbols for nodes and connections, some icons (also called decorations), as well as different colors are used to provide more information at a glance. Table 10-1 lists the various icons and their meaning.

Table 10-1 Icons used for Application Diagrams

Icon	Description
	Application Diagram viewer and visualize action
	Main entry point
	ILE RPG subprocedure or COBOL procedure
	ILE RPG subroutine

Icon	Description
	Program object
	Service project object
	Module bound into a program or service program object

In addition to the icon describing programs and service programs, there can also be a little arrow below it, sometimes called a “Module Toggler” as shown in Figure 10-11. It is only present if the program or service program contains more than one module.

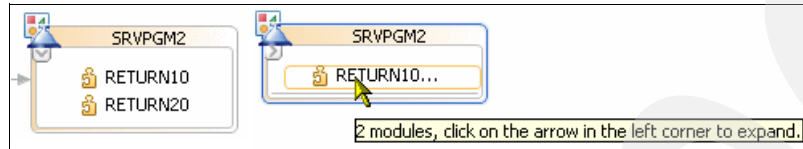


Figure 10-11 Module Toggler for a service program

You can click the arrow to expand or collapse the list of modules. Figure 10-11 also shows another interesting feature: When you move your mouse pointer (“hover”) over some of the graphics, you see more information describing the particular item. Figure 10-12 shows another example of a hover tip.

Note that Original Program Model (OPM) programs do not show any modules.

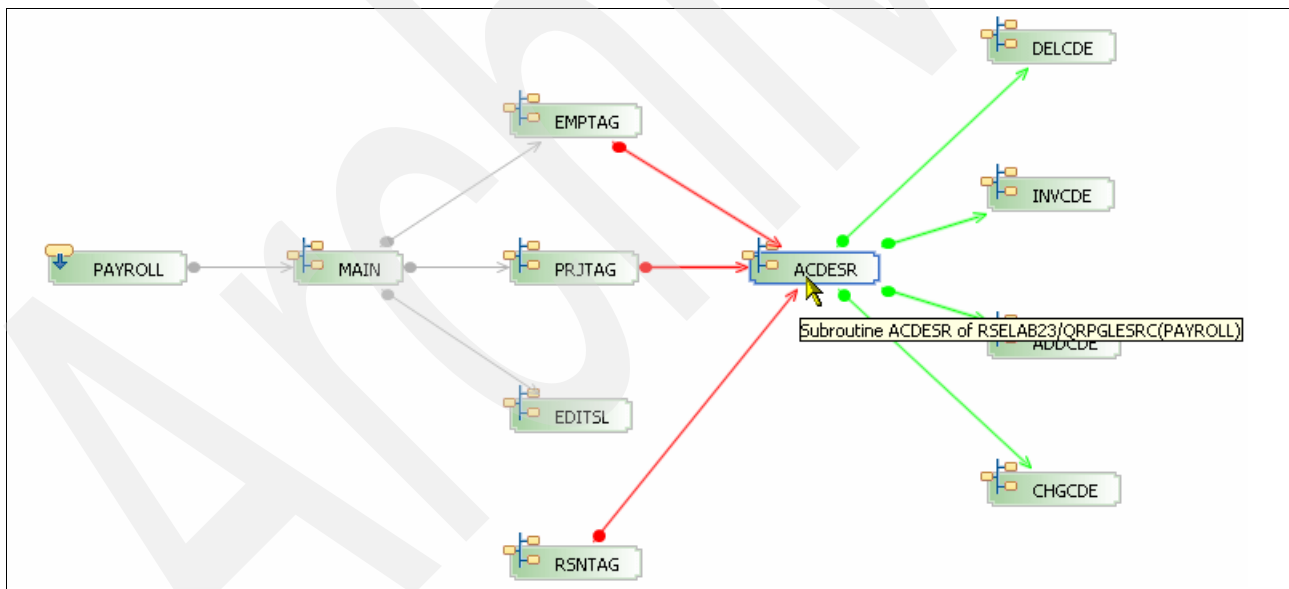


Figure 10-12 Source Call Diagrams with one node selected

The connections in Figure 10-12 represent invocation relationships between procedures or subroutines. The dot marks the calling element and the arrow head points to the called procedure or subroutine.

If you click a node to select it, you not only see the full name of the member, but also the connections to and from this node are highlighted in red and green, respectively. The red and green colors are defined by default. You may choose to select different colors, by changing your diagram settings, as described in 10.3.14, “Setting your preferences” on page 353.

While Figure 10-12 shows all traditional subroutines, you can see two procedures in Figure 10-13.

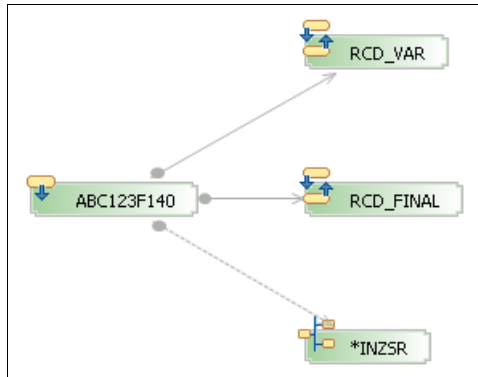


Figure 10-13 Source Call Diagrams with indirect call and procedures

A connection shown as a dashed line represents an implicit invocation, for example the one pointing to *INZSR in Figure 10-13.

There is one more notation to be mentioned: a service program node with a dashed border line. It means the information for these nodes is not completely retrieved. The reason is that these resources were not included in what you selected to launch the application diagram, but they are bound to one of the selections. For example, in Figure 10-14 we selected only PGM1 to create the diagram and SRVPGM1 shows a dotted border line. In this case, the service program does not show a module list.



Figure 10-14 Information for a node is not completely retrieved

10.3.2 Zoom in or out

The Palette view contains tools for zooming the diagram. You can enlarge or reduce the diagram by performing the following steps:

1. Click the Zoom button in the palette (shown in Figure 10-15).



Figure 10-15 Select the Zoom tool from the Palette

2. Move the mouse pointer to the area you plan to enlarge or reduce (Figure 10-16).

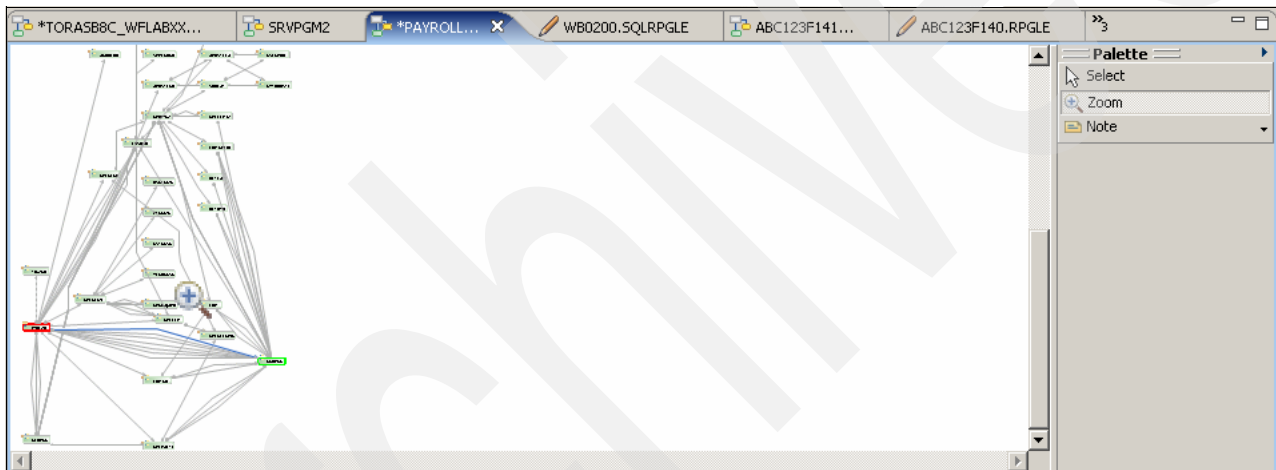


Figure 10-16 Move the mouse pointer to the area in the diagram

3. Zoom in (enlarge) by clicking (Figure 10-17 on page 331).

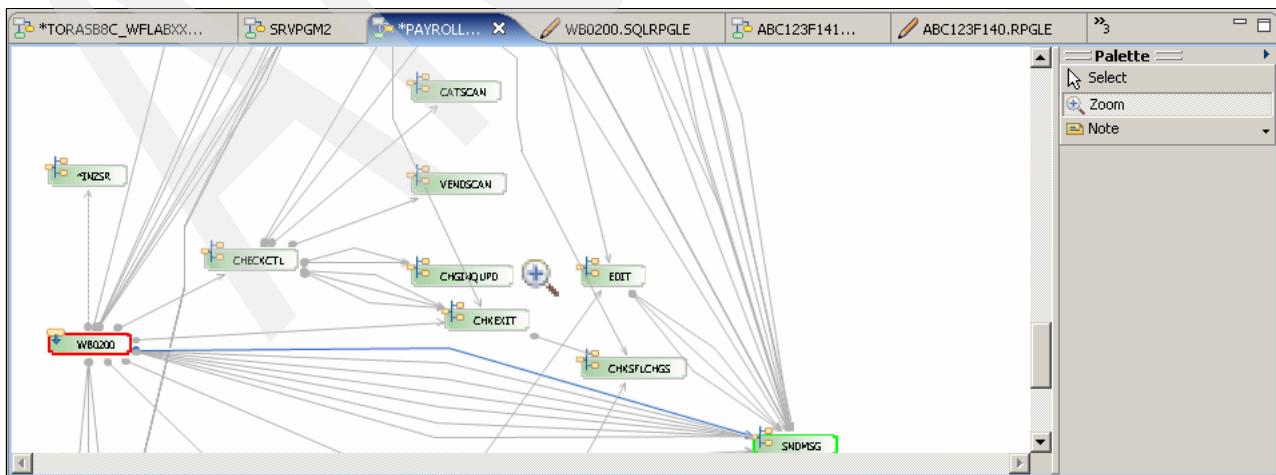


Figure 10-17 Zoom in (enlarge) by left clicking

4. Zoom out (reduce the size) by holding down the Shift key and clicking.

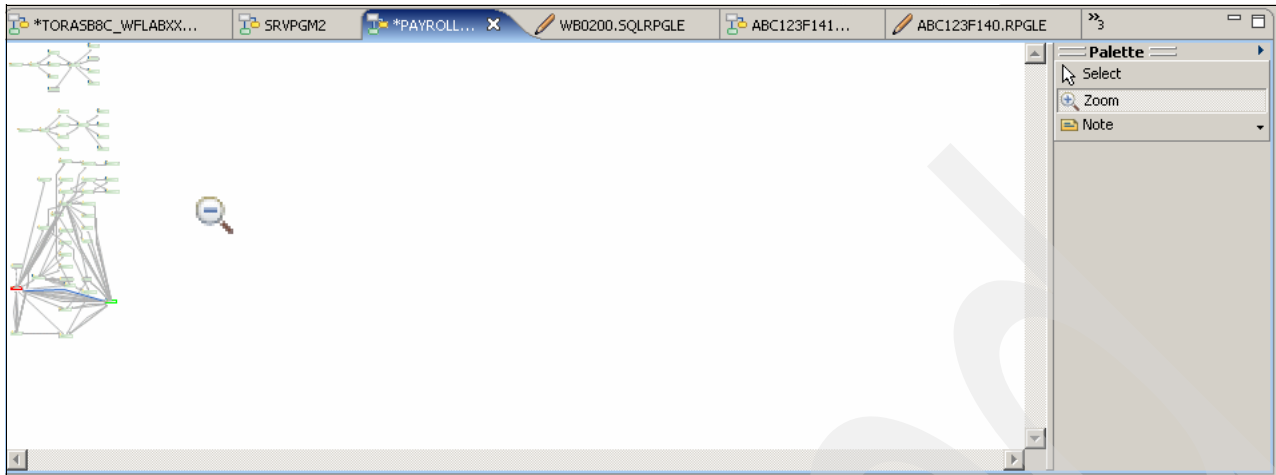


Figure 10-18 Zoom out (reduce the size) by holding down the Shift key and left clicking

When the diagram appears in the desired size, click **Select** at the top of the Palette to end the zoom mode.

Another way to zoom is the drop-down list from the tool bar as shown in Figure 10-19. The drop-down list not only allows you to zoom in and out or to select some specific percentages directly, it also provides the capability to scale the diagram according to its size and the size of the window. In our example we sized it so that the entire diagram fits into the window.

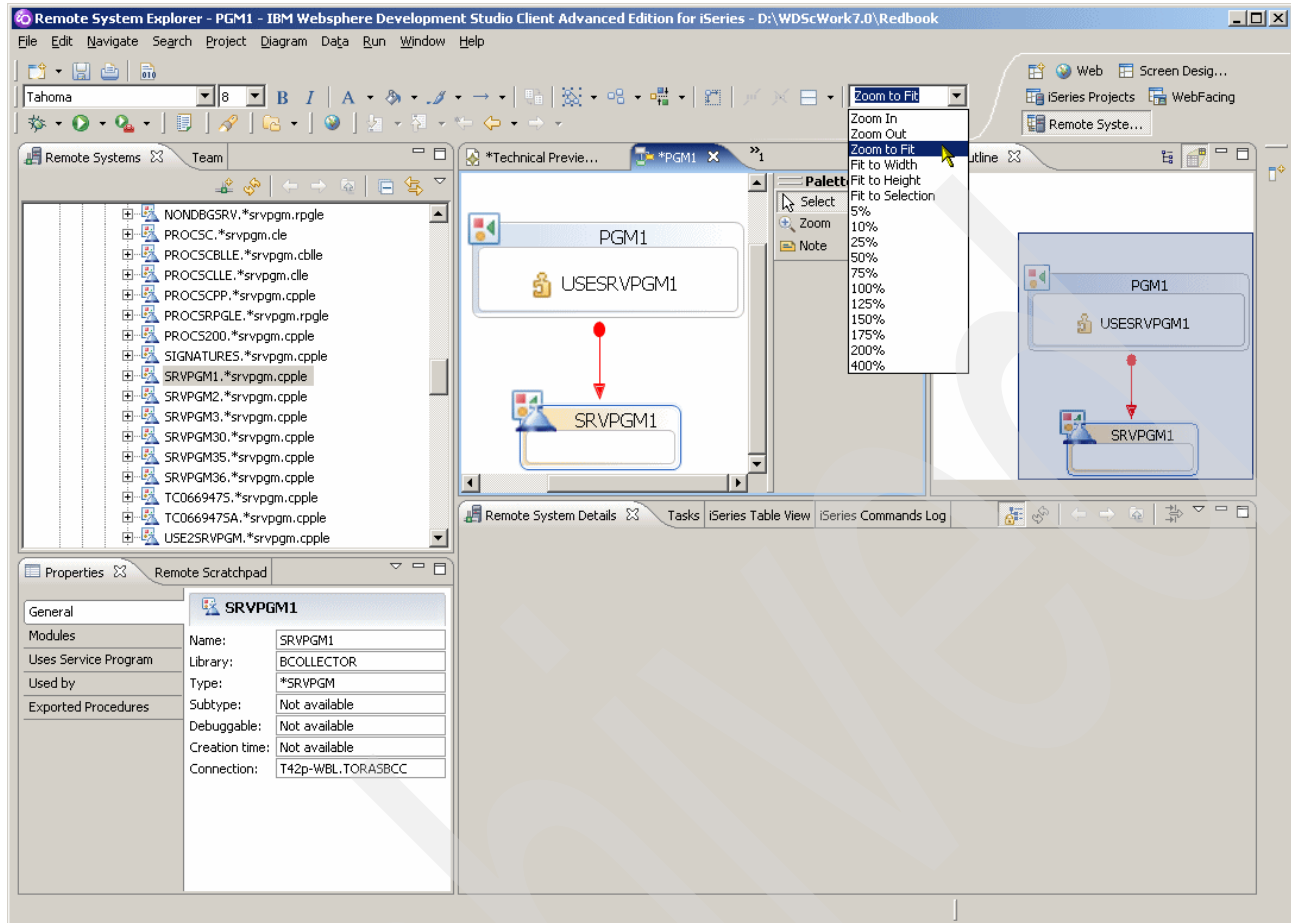


Figure 10-19 Drop-down in the task bar to zoom an application diagram

10.3.3 Application Diagram at a glance in the Outline view

The Application Diagram Viewer also populates the Outline view. There are two view modes: Outline and Overview.

Tip: If you do not see the Outline view (by default at the right of the Palette) because you previously closed it, there are two possibilities to restore it:

- ▶ Select **Window** → **Reset Perspective** from the main menu. This will reset the default layout of the Perspective.
- ▶ Select **Window** → **Show View** → **Other** → **General** → **Outline**. This will open the Outline view regardless of which perspective you currently see,

The overview mode of the outline view

Initially the Outline shows a thumbnail view, the so-called *overview mode* of the entire diagram. This can be useful when the entire diagram does not fit within the visible space of the viewer. Figure 10-20 on page 334 shows an Outline with thumbnail view of a complex diagram.

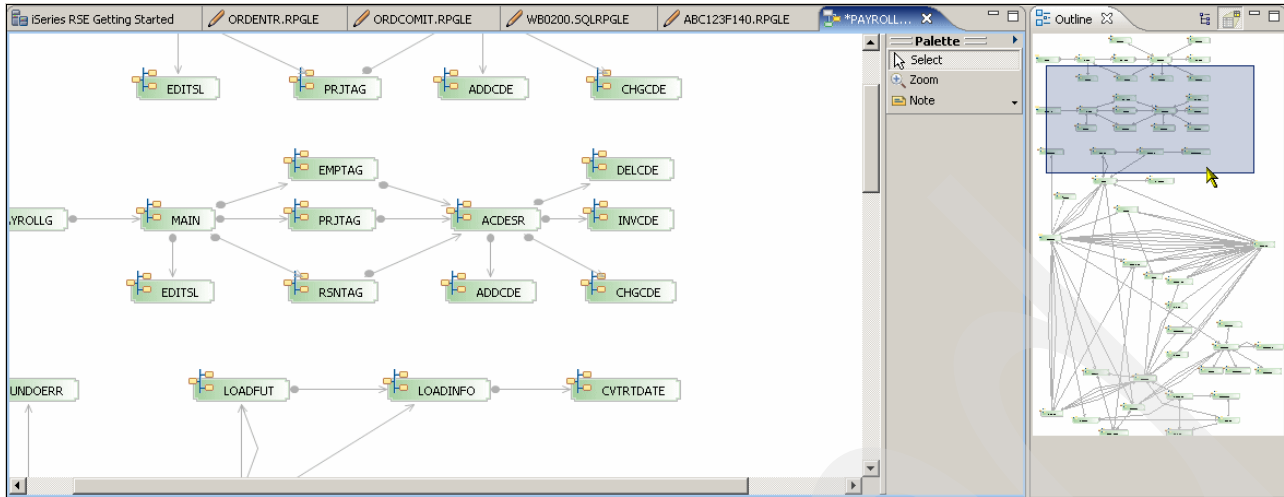


Figure 10-20 Outline showing a thumbnail view of a complex diagram

As you can see in Figure 10-20, the thumbnail view shows the entire diagram along with a shaded box (indicated by the mouse pointer) that represents the part of the diagram that is currently visible in the canvas on the left of Figure 10-20.

Using the mouse you can drag this shaded box around the thumbnail view to position the viewer. If you resize the canvas or shift the horizontal or vertical slide bars, or use the zoom function explained in the previous section, the shaded area is adjusted accordingly.

The outline mode of the outline view

The Outline view also contains a text view, the so-called *outline mode* that lists the sorted names of all the nodes in the diagram. This is useful when you want to find a specific node in the diagram—simply select it in the list and the viewer is repositioned so the node is visible. Use the icons in the toolbar (see the mouse pointer in Figure 10-21) at the top of the outline view to switch between the thumbnail and text views.

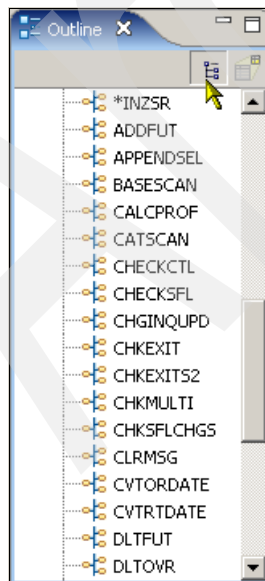


Figure 10-21 Outline showing the outline (text) view of a complex diagram

Tip: If the window size of the outline view is too narrow, so that you see only one of the icons, you may want to enlarge the view by dragging its border.

10.3.4 Get details from the Properties view

The Properties view can provide you with additional information for each node or connection you select in the Application Diagram. For some nodes the Properties view contains multiple tabs, with each tab showing different types of information.

The type of information and the number of tabs you may see here depends on the type of element in which diagram type you selected.

Properties of elements in a Source Call Diagram:

- ▶ For a node:
 - General: Name, type, location, start and end line number, connection.
 - Calls: Procedures or subroutines the selected artifact calls
 - Called by: Procedures or subroutines that call the selected artifact
- ▶ For a connection:
 - General: General information about caller, called resource and relationship type

Properties of elements in a Program Structure Diagram:

- ▶ For a node:
 - General: Name, type, debuggable attribute, creation time, connection.
 - Modules: A table displays all bound modules
 - Uses Service Program: The service program(s) used by selected program or service program

For service programs there are two extra tabs:


 - Used by: The program or service programs that use the selected service program.
 - Exported procedures: Exported procedures from the service program.
- ▶ For a connection:
 - General: Caller, called resource and relationship type

You can also select a module within a *PGM or *SRVPGM node to see details about the module in the Properties view.

If you do not see the Properties view, right-click any part of the diagram and select **Show Properties View** from the context menu.

By default the Properties view opens in the bottom left corner of the Remote System Explorer or System i Objects perspective. Occasionally this area does not provide sufficient horizontal space for some of the tables displayed in the Properties view when using the Application Diagram. You can move the Properties view to the group of views that contains the System i Table View (to the immediate right) in order to avoid having to scroll horizontally.

You achieve this by clicking the tab on top of the properties view (shown on the left of Figure 10-22 on page 336), keeping the left mouse button depressed and dragging the

pointer to the title row of the pane on the right. Make sure the mouse pointer changes its shape to a stack of folders  as shown in the upper right of Figure 10-22.

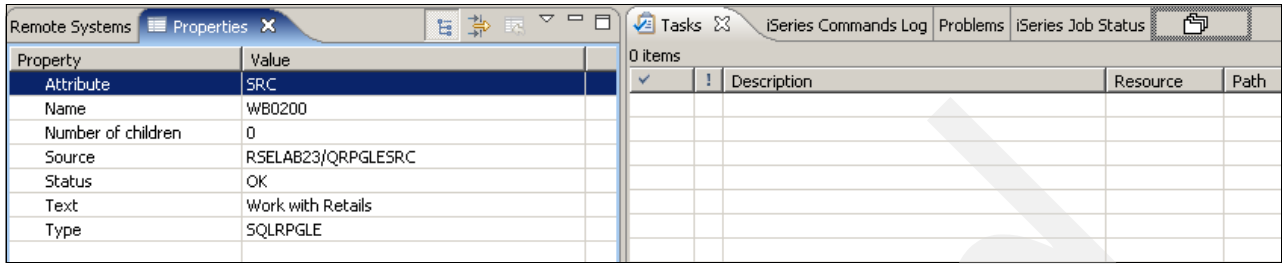


Figure 10-22 Dragging the properties view to a different location

After doing so, the lower part of the perspective will look as in Figure 10-23.

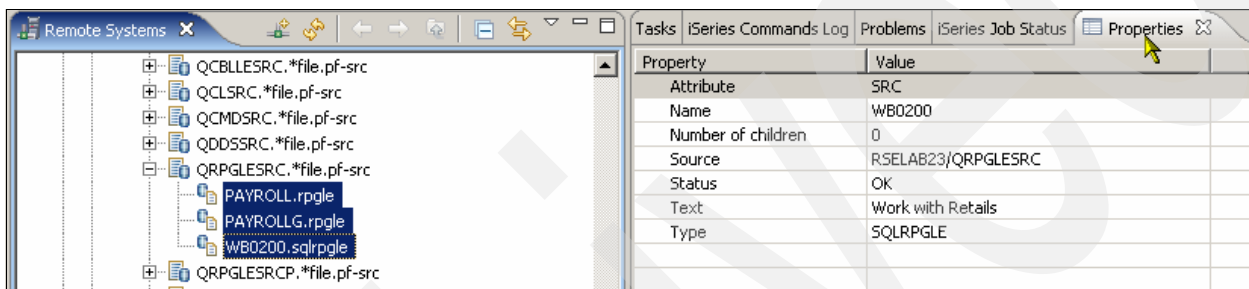


Figure 10-23 Properties view after dragging it to the right

If you click the white space of the canvas, rather than selecting a node or line, the properties view allows you to change the appearance of the Application Diagram viewer. As you can see in Figure 10-24, you can change the font, font size, as well as the colors of the text, canvas background and lines.

The changes apply only to the currently open diagram. To permanently change the appearance properties, see 10.3.14, “Setting your preferences” on page 353.

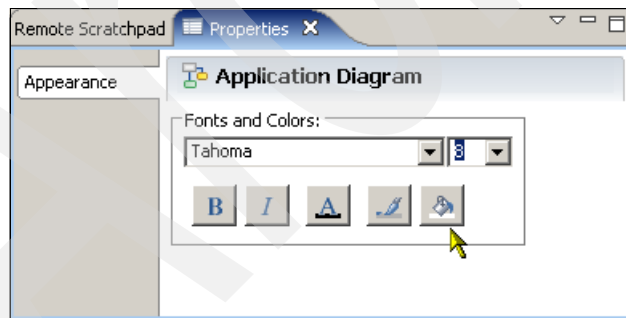


Figure 10-24 Appearance Properties of a diagram

10.3.5 Find the main entry point of your program

When there are many nodes and lines in the Source Call Diagram, it is not easy to find the main entry point. You can right-click on any blank section within the diagram and select **Find Main Entry Point** from the context menu, as in Figure 10-25.

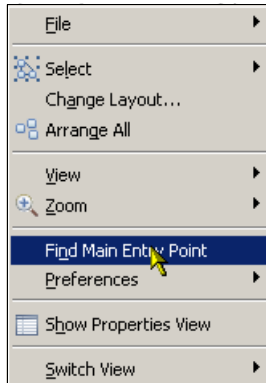


Figure 10-25 Find Main Entry Point

If your Source Call Diagram is based on more than one source member with each having a main entry point, a dialog like the one in Figure 10-26 lets you choose which one to select.

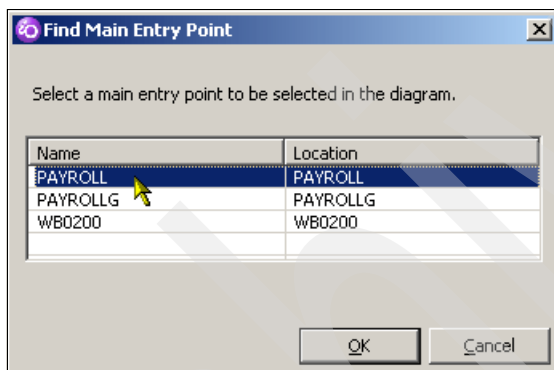


Figure 10-26 Select a main entry point to be shown in the diagram

The main entry point node will be found and highlighted, usually recognizable by the fact that at least one green line is originating at the node. If that node was not visible before, the diagram will shift so that the main entry point is now part of the visible area.

Tip: If your diagram consists of many source members, finding the main entry point allows you to quickly jump to a specific program.

10.3.6 Change the automatic layout of the diagram

When the application diagram first opens, it attempts to lay out the diagram in a way that makes it the most readable. You can configure how the diagram is laid out through the line style and orientation settings.

Diagrams with oblique lines

The choices for line style are either *Oblique* or *Rectilinear*. The Oblique line style draws connections between nodes taking the shortest route possible while attempting to avoid having lines that intersect other nodes as shown in Figure 10-27. Often this results in lines that are diagonal.

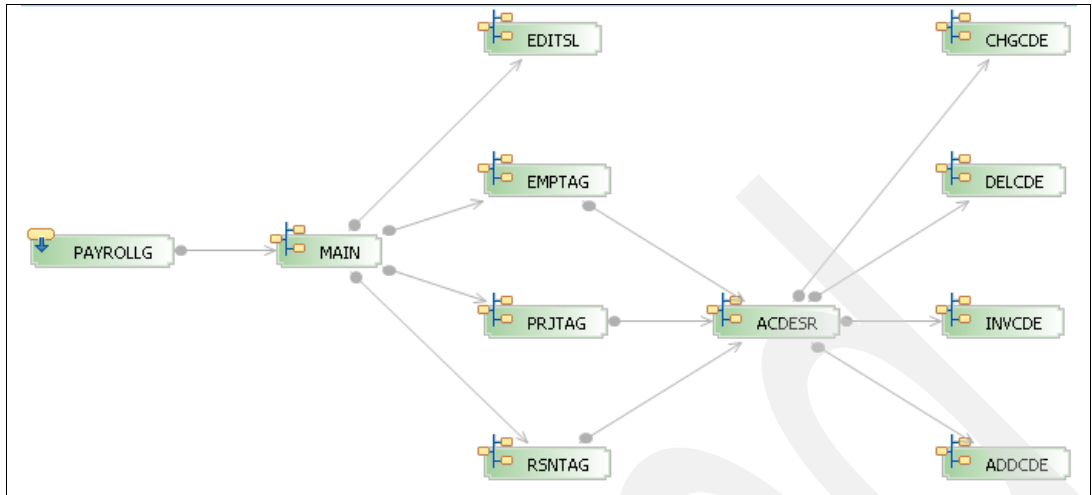


Figure 10-27 Horizontal diagram with oblique lines

Diagrams with rectilinear lines

The *Rectilinear* line style draws connections between nodes using only horizontal and vertical lines, as our example in Figure 10-28.

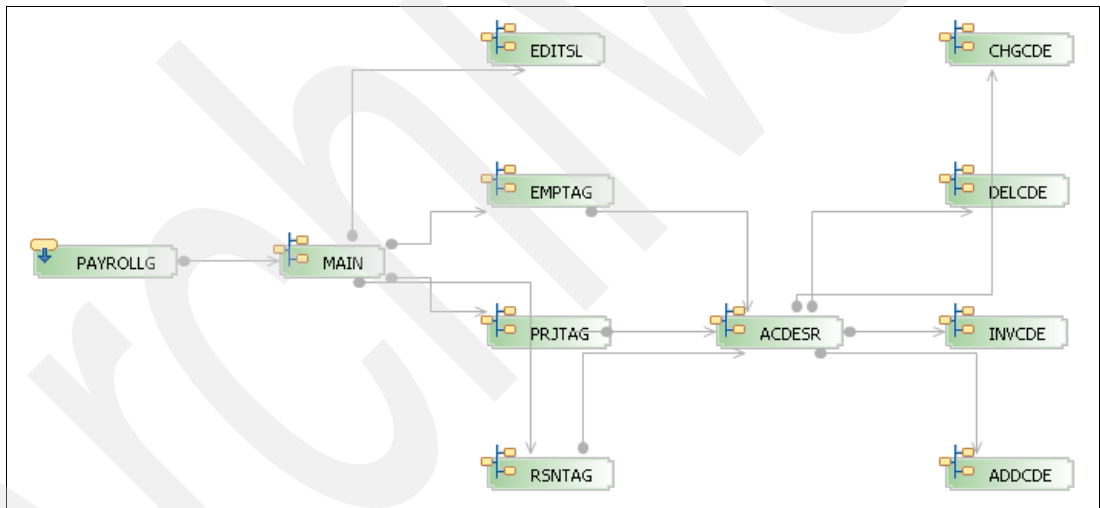


Figure 10-28 Horizontal diagram with rectilinear lines

Using a horizontal vertical orientation

The choices for orientation are horizontal and vertical. A horizontal orientation draws the diagram starting at the left and growing as required to the right. A vertical orientation draws the diagram starting at the top and growing as required to the bottom. Figure 10-29 shows the same diagram as in the two previous examples, now in a vertical orientation.

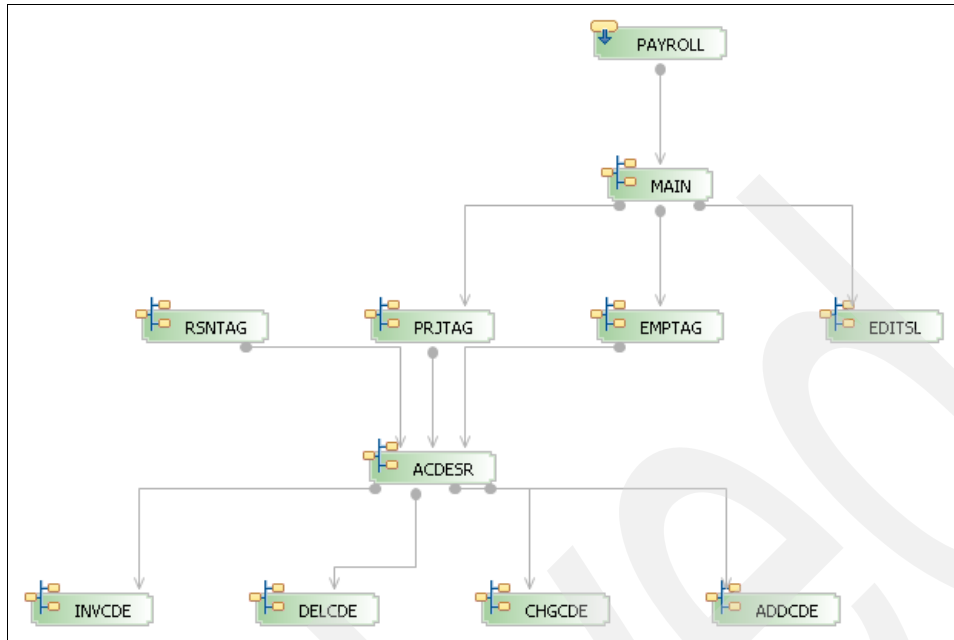


Figure 10-29 Vertical diagram with rectilinear lines

Our last example in Figure 10-30 is again oriented vertically but with oblique lines.

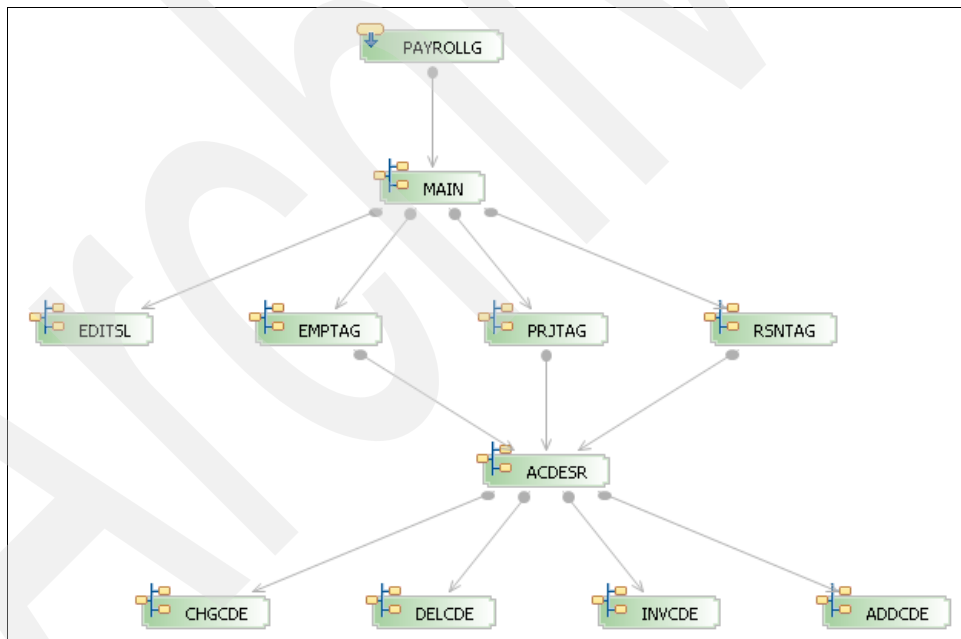


Figure 10-30 Vertical diagram with oblique lines

The default line style and orientation are Oblique and Vertical as shown in Figure 10-27 on page 338. To change the two layout parameters, for a single diagram right-click on a blank section of the diagram and select the **Change Layout** action from the pop-up menu, as shown in Figure 10-31.

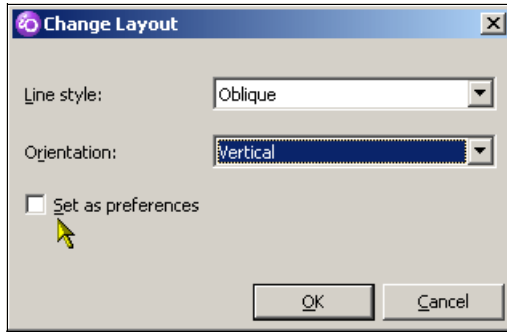


Figure 10-31 Change layout dialog

If you want to change the defaults for all diagrams, you can either select the **Set as preferences** checkmark in Figure 10-31 or on the Application Diagram preference page as we describe in 10.3.14, “Setting your preferences” on page 353.

10.3.7 Modifying the diagram layout manually

After building a diagram and having it laid out automatically based on the options explained in 10.3.6, “Change the automatic layout of the diagram” on page 337, you still might not be completely satisfied with the result in all cases.

Moving a single node to a different location within the diagram

For example, if the diagram has many nodes and connections, lines may cross each other several times, making it difficult to understand, such as the portion of a bigger diagram pointed to by the mouse pointer in Figure 10-32. In other cases you just may want to group several nodes differently, based on the nature of the application, and keep the diagram for documentation purposes by performing the following steps.

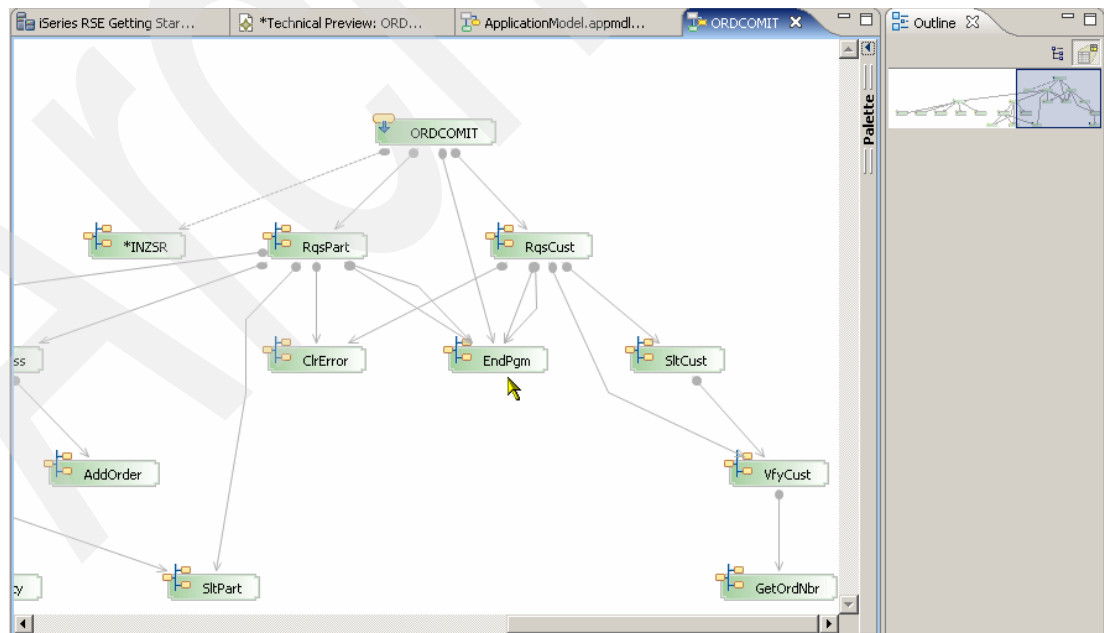


Figure 10-32 Portion of a complex diagram with crossing lines

1. To arrange the complex of nodes in the middle, you just click the node, EndPgm in our example, and drag it while keeping the left button depressed to a different location, as shown in Figure 10-33.

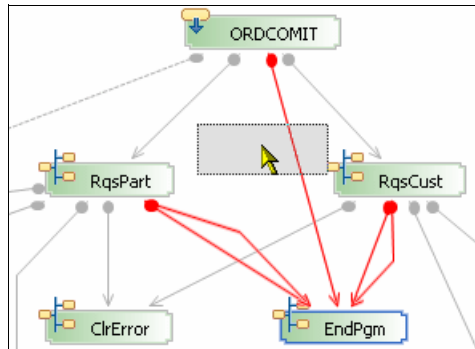


Figure 10-33 Dragging a node to a different location within the diagram

2. When the shaded rectangle reaches the desired position, release the mouse button and the node is placed there with all lines connecting to it adjusted accordingly, as you can see in Figure 10-34.

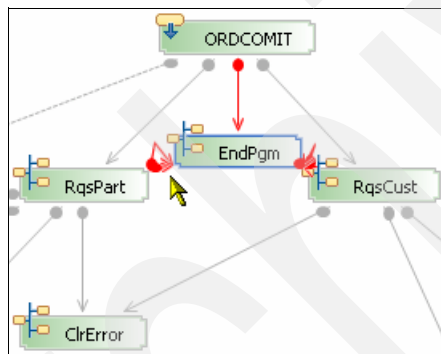


Figure 10-34 Node dropped at a different position in the diagram

Moving multiple nodes to a different location at once

However, the diagram shown in Figure 10-34 can still be somewhat improved by moving the three nodes RqsPart, ClrError, and RqsCust a little further down. You can do that in a single operation by performing the following steps:

1. Position the mouse pointer anywhere on the white space, click the left button, leave it pressed and move the mouse so there is a dashed rectangle just around the nodes you want to move. The three nodes you want to move must be *completely* inside the rectangle, as shown in Figure 10-35 on page 342. Other nodes may be partially inside it.

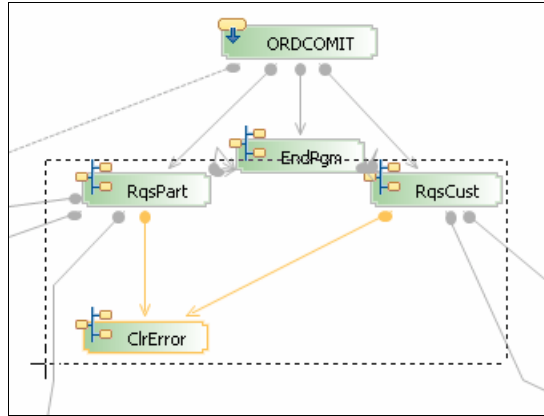


Figure 10-35 Selecting multiple nodes by drawing a rectangle

2. Release the mouse button. Only the three nodes should be highlighted, as in Figure 10-36 now.

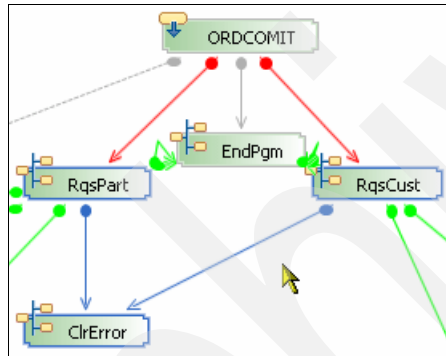


Figure 10-36 Multiple nodes highlighted for dragging

Note the blue borders of the nodes, the blue lines connecting selected nodes, and the red and green lines connecting to not selected nodes in Figure 10-36. Make sure only the desired nodes and all of them are selected. Otherwise, click somewhere in the diagram and repeat the process.

You can also add or remove nodes from the selected group by pressing and holding the Ctrl key and clicking nodes. Each click toggles the status from “selected” to “not selected” and vice versa.

3. Click one of the selected nodes, keep the button depressed, drag all three nodes down to the position you want the group to be moved to, as shown in Figure 10-37 on page 343.

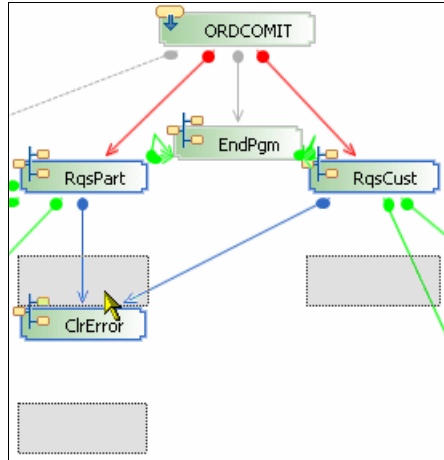


Figure 10-37 Dragging multiple nodes at once

Figure 10-38 shows the diagram after moving the three nodes down.

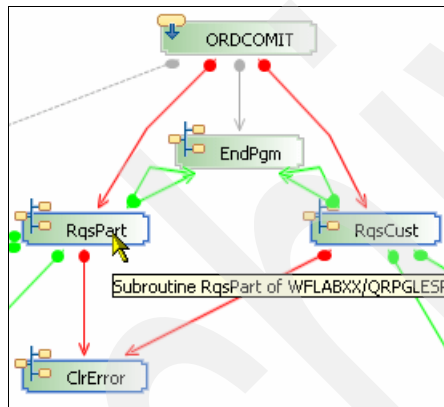


Figure 10-38 Result of dragging multiple nodes

The toolbar shows various tools to arrange the nodes in your diagram. Click the triangle on the right of each button to find more tools. For example, in Figure 10-39 on page 344 we selected three other nodes and chose **Align Middle** to place all three horizontally on the same height within the diagram.

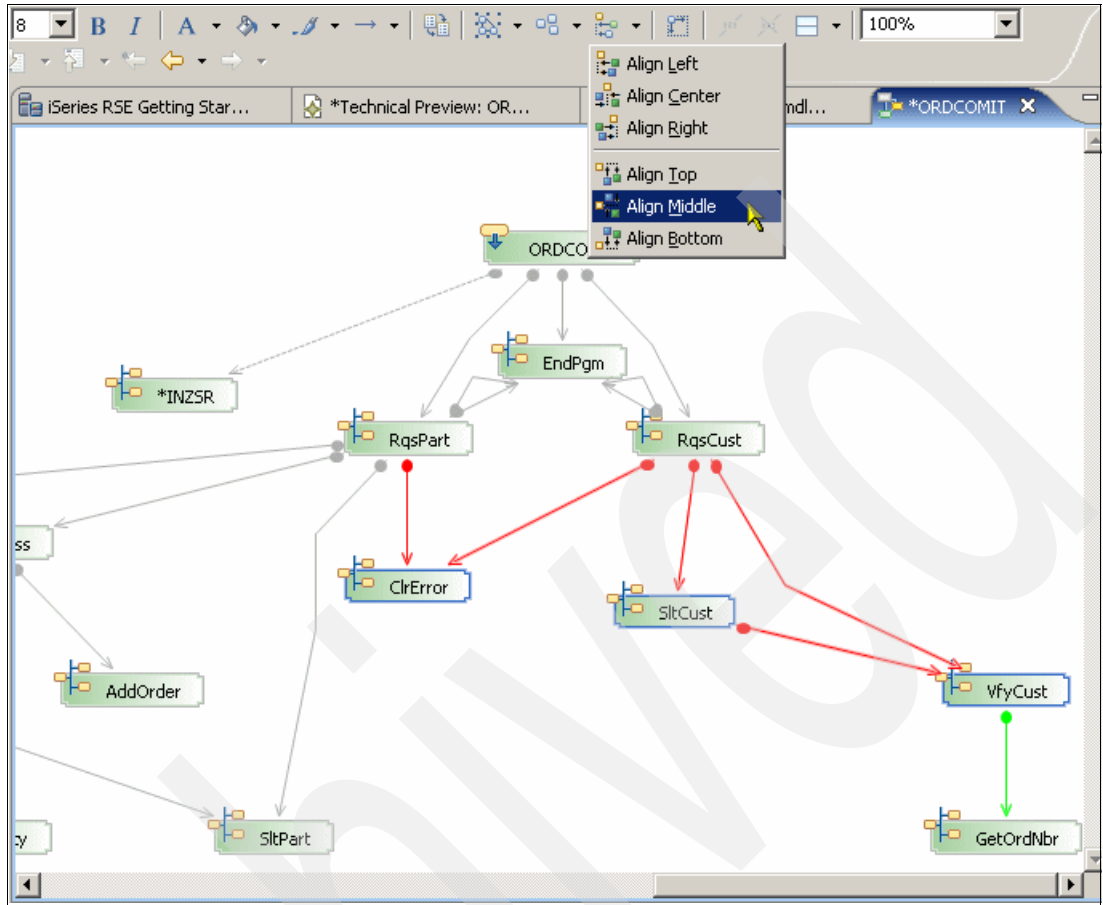


Figure 10-39 Aligning multiple nodes horizontally

The result of the above action is shown in Figure 10-40. Using these tools for aligning nodes rather than moving each individual node makes the diagram look much cleaner and “professionally” done.

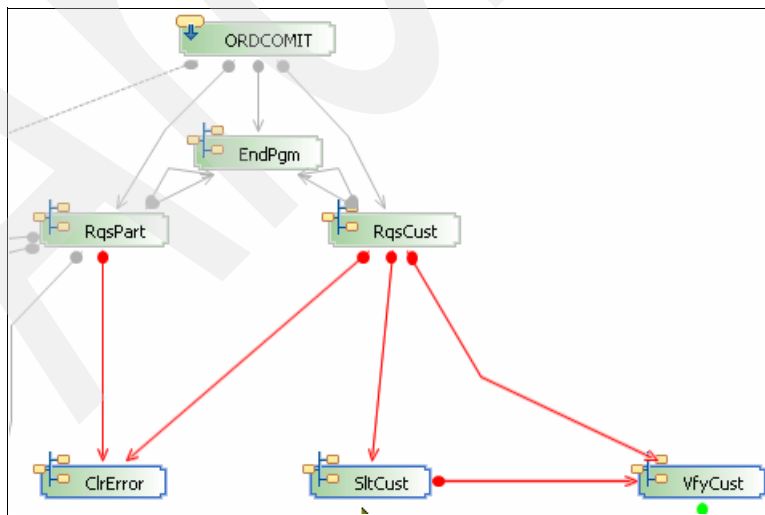


Figure 10-40 Result after aligning three nodes horizontally

Tip: If you are not satisfied with moving one or more nodes, you can always undo the previous action by selecting **Edit** → **Undo** from the main menu or simply pressing the Ctrl+z keys. Note, however, that you can only undo a single step.

It is also a good practice to save the diagram, as described in “Save as a workspace file” on page 349 before, during and after these kinds of manipulations. Saving as a text file would not be of any help here; saving as an image file (page 352) may be used to document the diagram, but you cannot change it later.

10.3.8 Switch the view

As we explained earlier in this chapter, there are two types of diagrams:

- ▶ The Source Call Diagram
- ▶ The Program Structure Diagram

One or more of each type can be combined into a single model, if you select multiple source members, program objects, and service program objects. The diagram viewer has three view modes:

- ▶ Showing both types of diagrams (the default)
- ▶ Only the Source Call Diagram
- ▶ Only the Program Structure Diagram

You can switch between these modes by right-clicking on a blank section of the diagram and selecting one of the following from the context menu, as shown in Figure 10-41:

- ▶ **Switch View** → **All**
- ▶ **Switch View** → **Source Call Diagram**
- ▶ **Switch View** → **Program Structure Diagram**

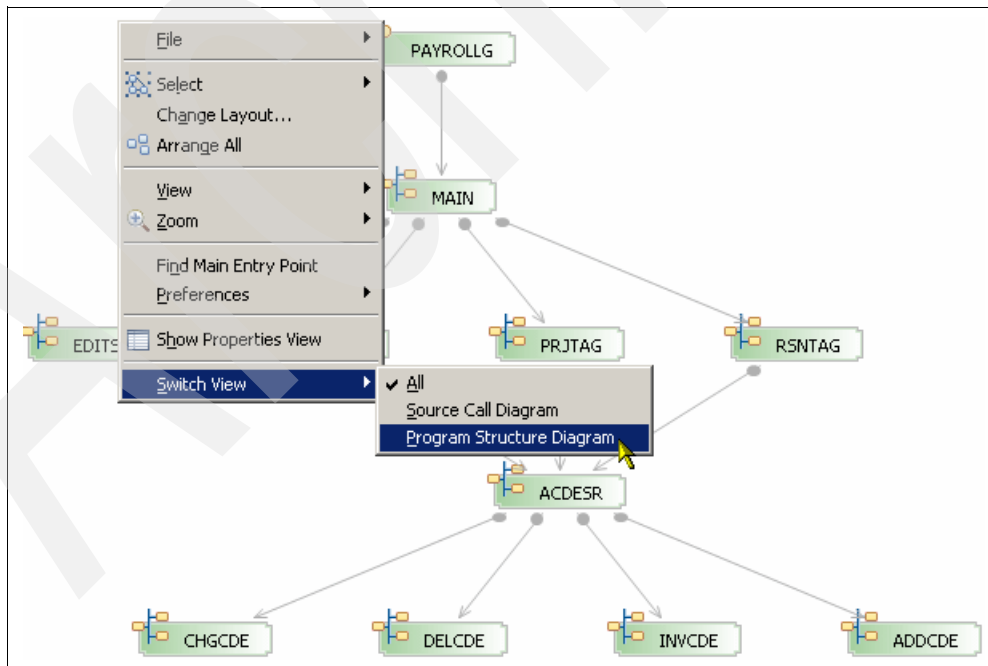


Figure 10-41 Switching the view

After you selected either **Program Structure Diagram** or **Source Call Diagram**, the view may become empty if there is no applicable resource for the selected view.

10.3.9 Using the Application Diagram Viewer Palette

The Palette view contains tools for zooming and adding notes to the diagram. The palette can be minimized and maximized by clicking the black triangle in the upper right corner of the palette.

If you prefer to use the Workbench Palette view instead of the palette embedded inside the viewer, then open the full palette view by selecting **Window** → **Show View** → **Other** → **General** and the embedded palette will disappear. Closing the full Palette view will cause the embedded palette to reappear. When using the Zoom tool from the palette, you zoom in by clicking and zoom out by holding down the Shift key and clicking.

10.3.10 Edit or browse the source

You can edit or browse the source for some of the nodes in the Application Diagram such as modules, RPG subroutines and procedures, and COBOL procedures. Double-clicking the node automatically opens the editor to edit the source. Alternatively, you can right-click and select either **Edit Source** or **Browse Source** from the pop-up menu.

You can also edit or browse the source from the Properties view, by selecting the Modules tab, right-clicking any table entry and selecting from the context menu.

You can also double-click connections that represent subroutine, procedure, or procedure calls to have the source for the calling node opened in the editor and positioned to the line where the call is performed.

When you use a Program Structure Diagram, you can also edit or browse the source member related to any of the modules shown here. Note that you must select a single module, rather than the entire node (the program or service program object).

10.3.11 Adding notes and descriptive text to the diagram

You can add documentation to the generated Application Diagram by adding notes to it from either the context menu or palette. Notes can either be standalone or attached to a node or connection in the diagram.

Important: In order to keep notes, you must explicitly save the diagram as we describe in 10.3.12, “Storing an application diagram for future use” on page 349. If you close the diagram without saving it, then all notes will be lost.

Annotating a procedure, subroutine, or program node

To add a note to a node, first select the node and then right-click and select **Add Note** from the pop-up menu.

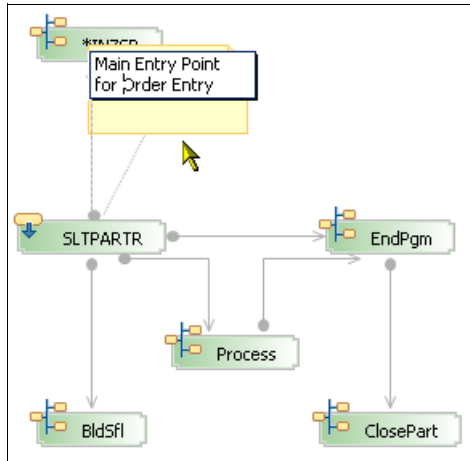


Figure 10-42 Adding a note to a node

You can now enter a description for the procedure, subroutine, program, or service program in the attached note in the white area above the mouse pointer in Figure 10-42.

Tip: To start a new line for your text similar to our example in Figure 10-42, press and hold Ctrl and press the Enter key.

Add a note to a connection

To annotate the call of another subroutine or procedure in a Source Call diagram or the bind relationship in a Program Structure Diagram, right-click the connection line joining two nodes (process and endpgm in our example in Figure 10-43) and click **Add Note**. This action creates an attached note to the connection.

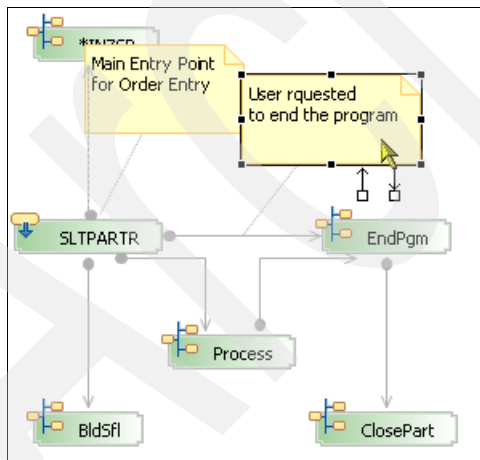


Figure 10-43 Add a note to a connection

Connecting a note to more than one object

If you hold the mouse pointer over the newly created node, two arrows pointing in opposite directions appear (below the mouse pointer in Figure 10-43). They allow the note to be attached to more than one diagram object. To attach an existing attached note to another connection, drag and drop the down arrow to another line or node. In our example, connecting the nodes Process and EndPgm. See the mouse pointer in Figure 10-44.

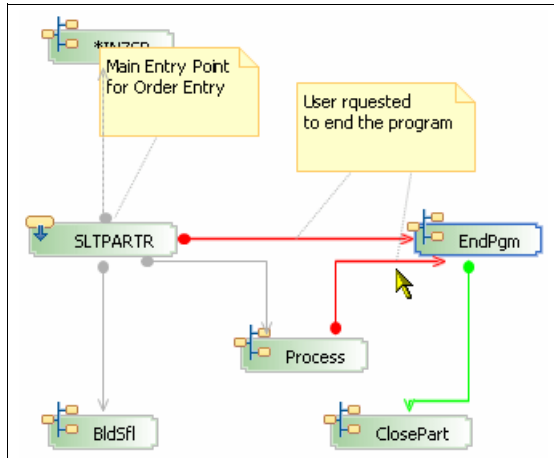


Figure 10-44 Attach an existing note to another connection

Creating a standalone note

To create a standalone note (a note that is not attached to any node or connection), click **Note** in the Palette menu, as shown in Figure 10-45.

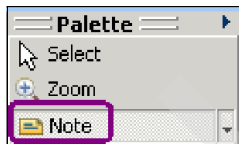


Figure 10-45 Select the note tool from the Palette

Then drag the note object and drop it anywhere on the diagram canvas. A standalone note is created. A standalone note is not connected to any object within the diagram.

You can create text that is also standalone, but has a transparent background. To do so, click the drop-down arrow in the Palette view, and select **Text**, as shown in Figure 10-46.

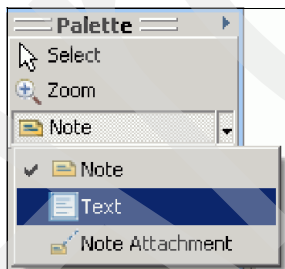


Figure 10-46 Select the text tool from the Palette

The text tool allows you to place any text without any surrounding frame or background color different from the canvas to appear anywhere in the diagram, for example as shown in Figure 10-47 above the mouse pointer.

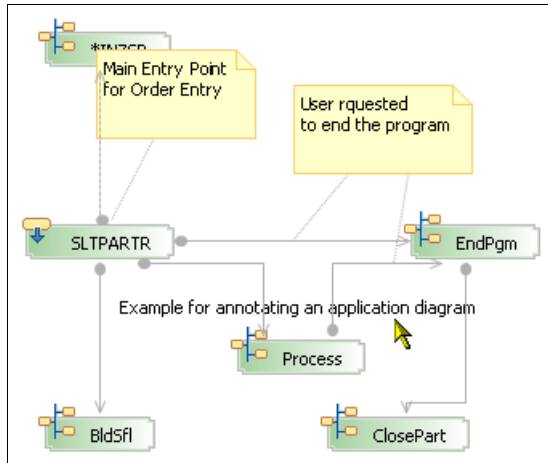


Figure 10-47 Standalone text appearing in a diagram

To change the text font and colors, right-click a blank area in the main canvas and click **Show Properties View** as described with Figure 10-24 on page 336.

Important: In order to keep notes, you must explicitly save the diagram as we describe in 10.3.12, “Storing an application diagram for future use” on page 349. If you close the diagram without saving it, all notes will be lost.

10.3.12 Storing an application diagram for future use

The Application Diagram can be saved to a workspace file, an image file, or a text file. Saving to a workspace file allows the diagram to be reopened at a later time. Saving to an image file allows the diagram to be easily sent to another person or embedded within another document. Saving to a text file allows the information from the diagram to also be saved for later reference, easily copied, or searched using a text search.

Save as a workspace file

To save the diagram to a workspace file, select **File** → **Save As** in the menu. This prompts you for a folder inside the local workspace and the file name to save the diagram as. The term “folder” here can be a project or a folder within a project. In our example in Figure 10-48 on page 350 we first created a general project called ApplicationDiagrams. However, you can also create or use any other type of project.

To create a general project, select **File** → **New** → **Project** from the main menu, then select **General** → **Project** from the New project dialog.

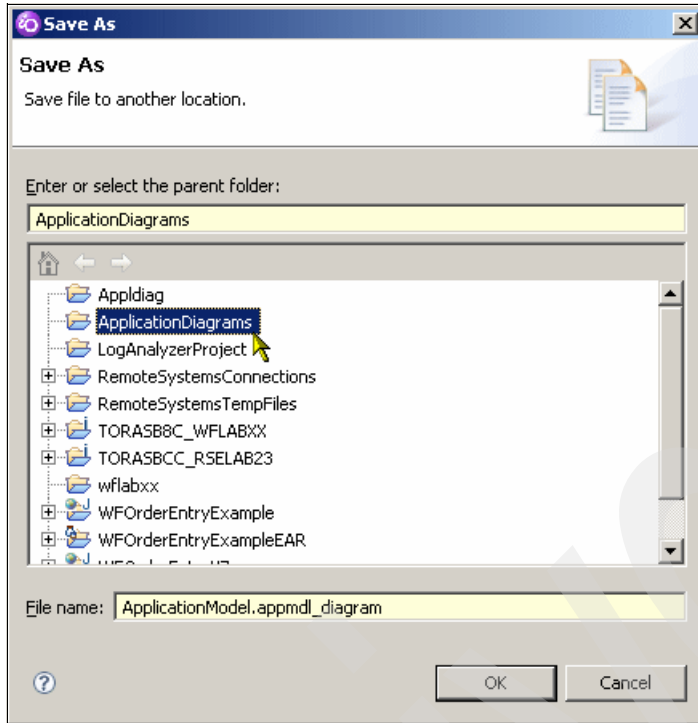


Figure 10-48 Saving a diagram to the workspace

Saving the diagram creates two files: an `ApplicationModel.appmdl` file and an `ApplicationModel.appmdl_diagram` file. The first file contains the model generated by the Visualize Application Diagram action. The second file contains the layout information and any notes you have added to the diagram. Double-click the `ApplicationModel.appmdl_diagram` file to reopen the saved diagram, as shown in Figure 10-49.

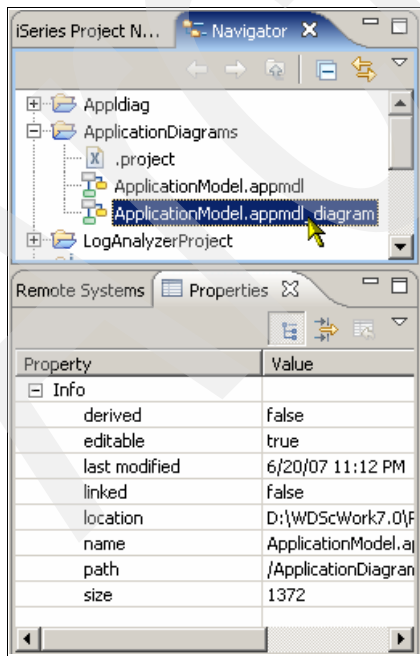


Figure 10-49 Restoring a previously saved diagram from a project

Save as a text file

If you want the diagram to be in a format that you can easily use to share, duplicate, and search in, you may also save it as a text file. To do so, right-click on a blank section of the canvas and select **File** → **Save As Text File** from the context menu, as in Figure 10-50.

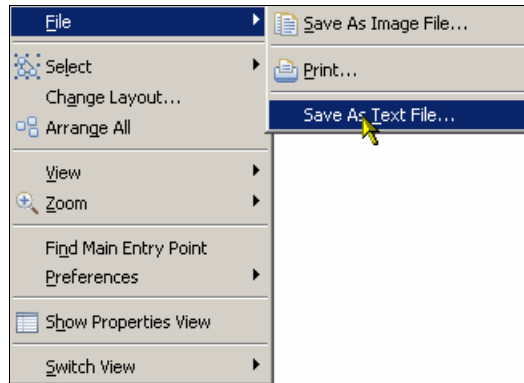


Figure 10-50 Save the application diagram as a text file

Besides information on the name of the source member, source file, library and the RSE connection, the text file contains a `Called by:` and `Calls:` list of procedures and subroutines. Figure 10-51 on page 352 shows a simple example for a diagram saved to a text file.

Saving as text file will only save the application model. It will not save any diagram layout and notes.

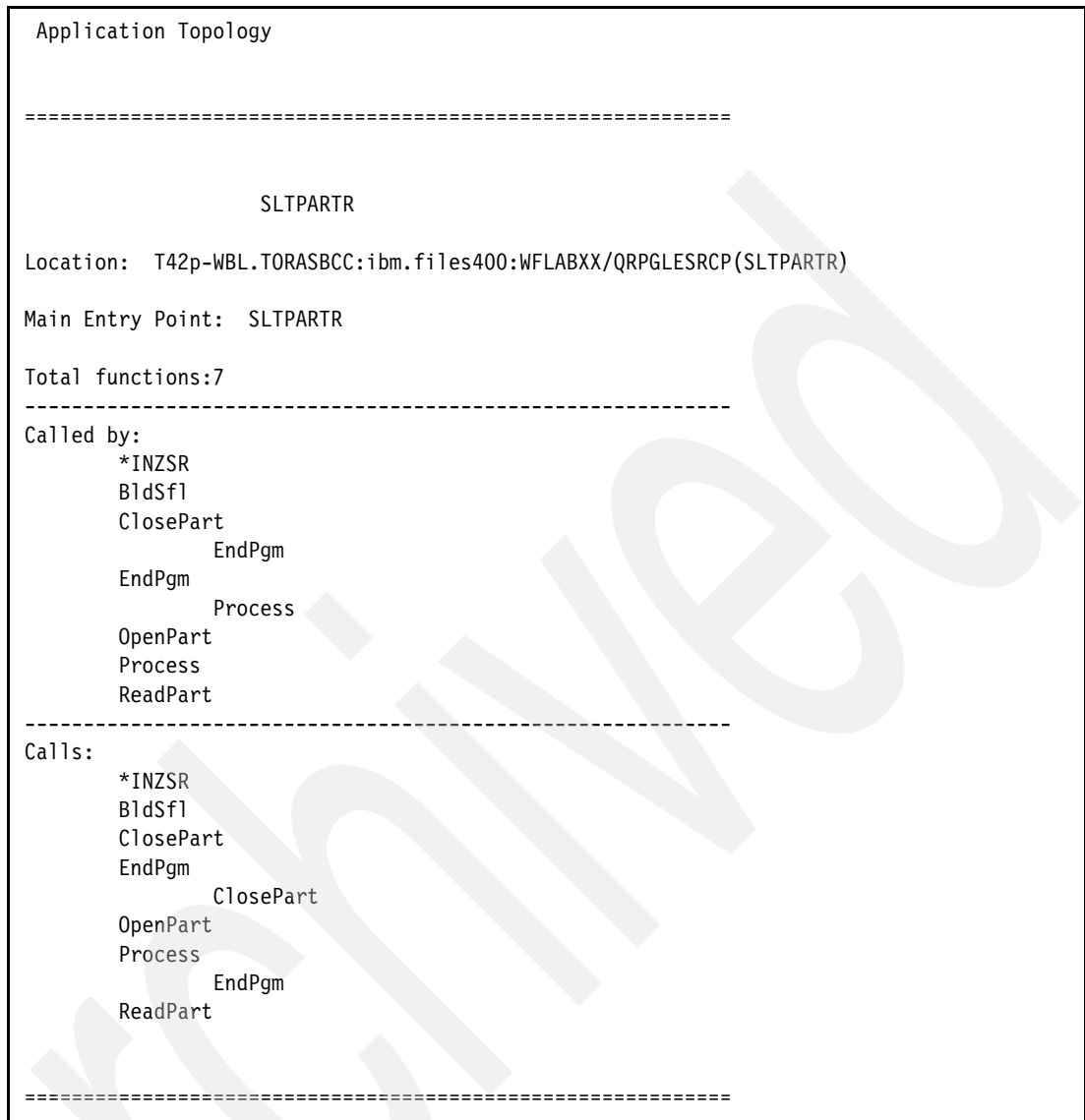


Figure 10-51 Example for a diagram saved to a text file

Save as an image file

To save the diagram to an image file, right-click on any blank section within the diagram and select **File** → **Save As Image File** from the context menu. It allows you to save the file outside of the current workspace.

You can also save a part of the diagram to an image file by first selecting all nodes and connections you would like saved and then selecting **File** → **Save As Image File** from the context menu of any of the selected nodes or connections.

10.3.13 Printing an Application Diagram

You can print the diagram from the workbench menu or from the context menu of the diagram or by using the shortcut Ctrl-p. To print an application diagram from the workbench menu, select **File** → **Print** from the main menu. To print a diagram from the context menu, right-click on any blank section within the diagram and select **File** → **Print** from the context menu.

In either case, the dialog shown in Figure 10-52 appears.

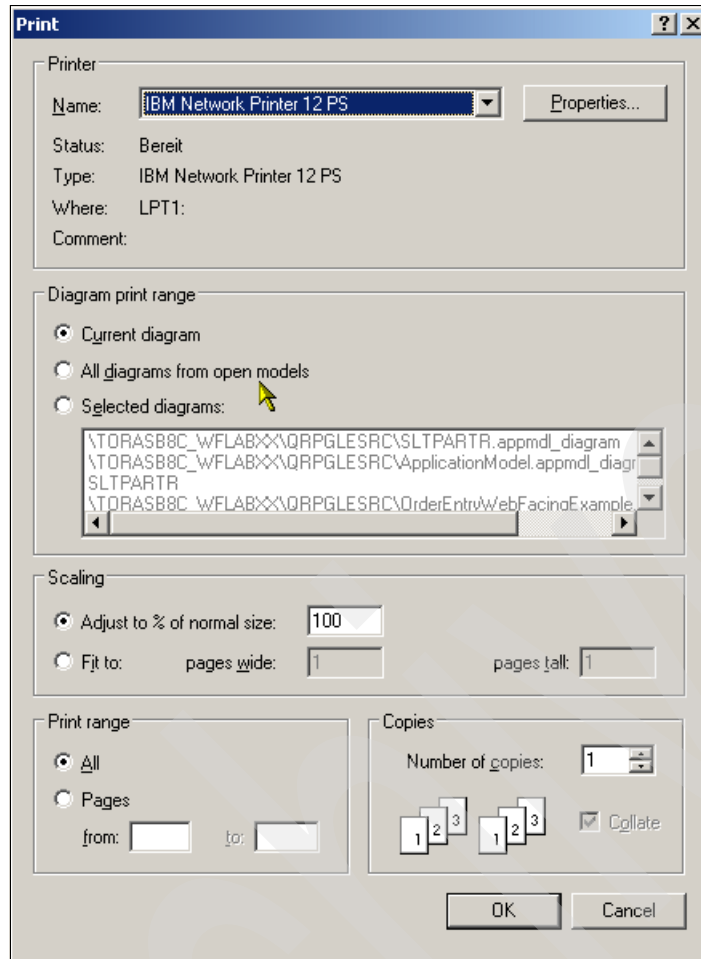


Figure 10-52 Print Application Diagram dialog

You may choose to print the currently active, all open, or some of the currently open diagrams.

The print preview is also available. To see the preview before actually printing, select **File** → **Print Preview** from the main menu.

10.3.14 Setting your preferences

There are two places where you can specify preferences for the Application Diagram:

- ▶ Application Diagram, to specify default view, line style, and orientation
- ▶ Colors and Fonts, for descriptive text and lines

Both can be edited either through the Workbench preferences page or from the context menu when you right-click an empty space of the diagram canvas.

Application Diagram preferences

To change the Application Diagram preferences, perform one of the following steps (Figure 10-53 on page 354):

- ▶ Select **Window** → **Preferences** → **Application Diagram** from the main menu.

- ▶ Right-click an empty space of the diagram canvas and select **Preferences** → **Application Diagram** from the context menu.
- ▶ Click the Remote System view drop-down button and select **Preferences** → **Application Diagram** from the context menu.

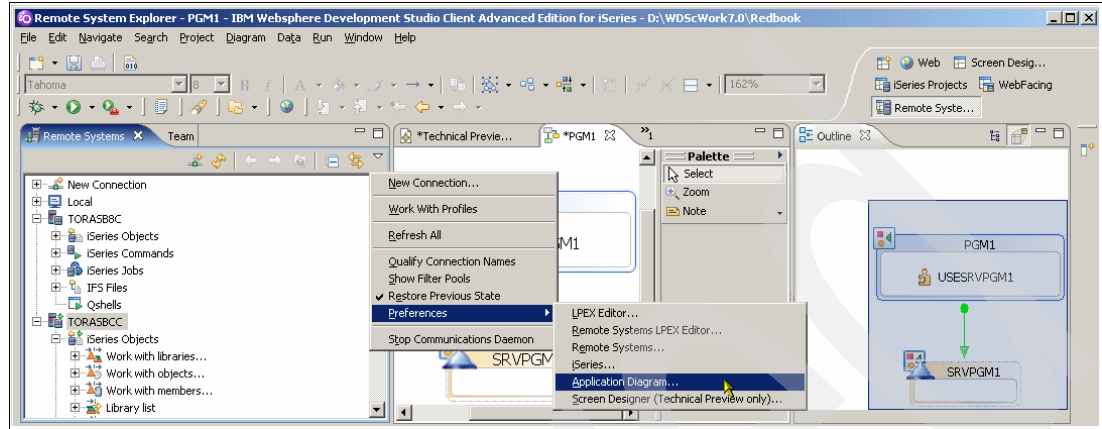


Figure 10-53 Application Diagram preferences from Remote System view drop-down button

In either case, the Application Diagram preferences are shown, as in Figure 10-54. However, when called from the main menu, the left pane would allow you to navigate to other preferences.

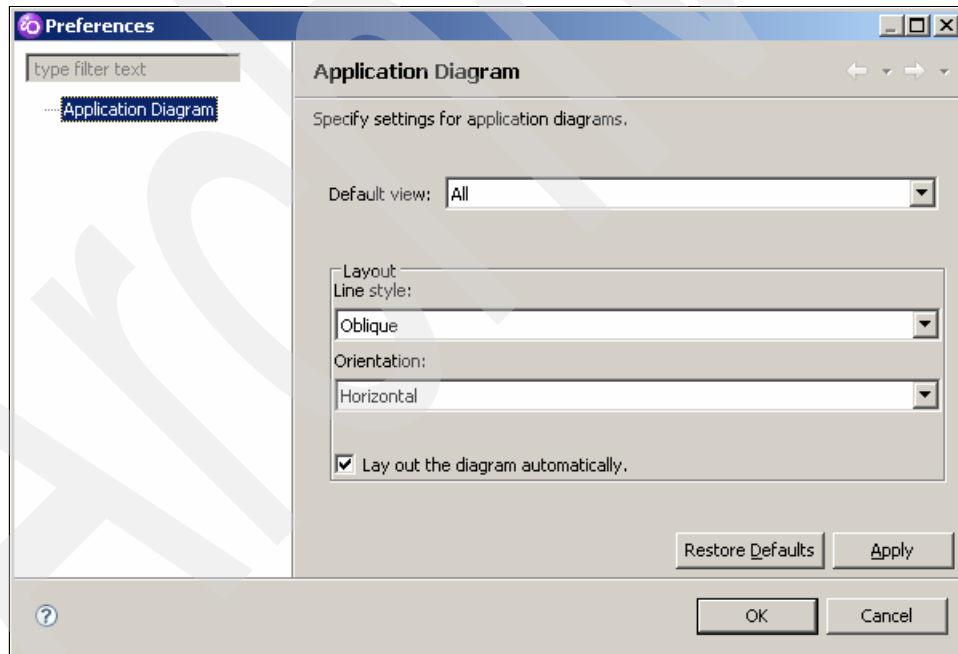


Figure 10-54 Application Diagram preferences

Colors and Fonts preferences

You can find the Colors and Fonts preferences sections in the Workbench preferences page: **General** → **Appearance** → **Colors and Fonts** or by right-clicking an empty space of the diagram canvas and selecting **Preferences** → **Colors and Fonts** from the context menu.

In either case, expand the tree under Application Diagram (see mouse pointer in Figure 10-55 on page 355) and select one of the following:

- ▶ Application Diagram font
- ▶ Incoming connection highlight color
- ▶ Outgoing connection highlight color

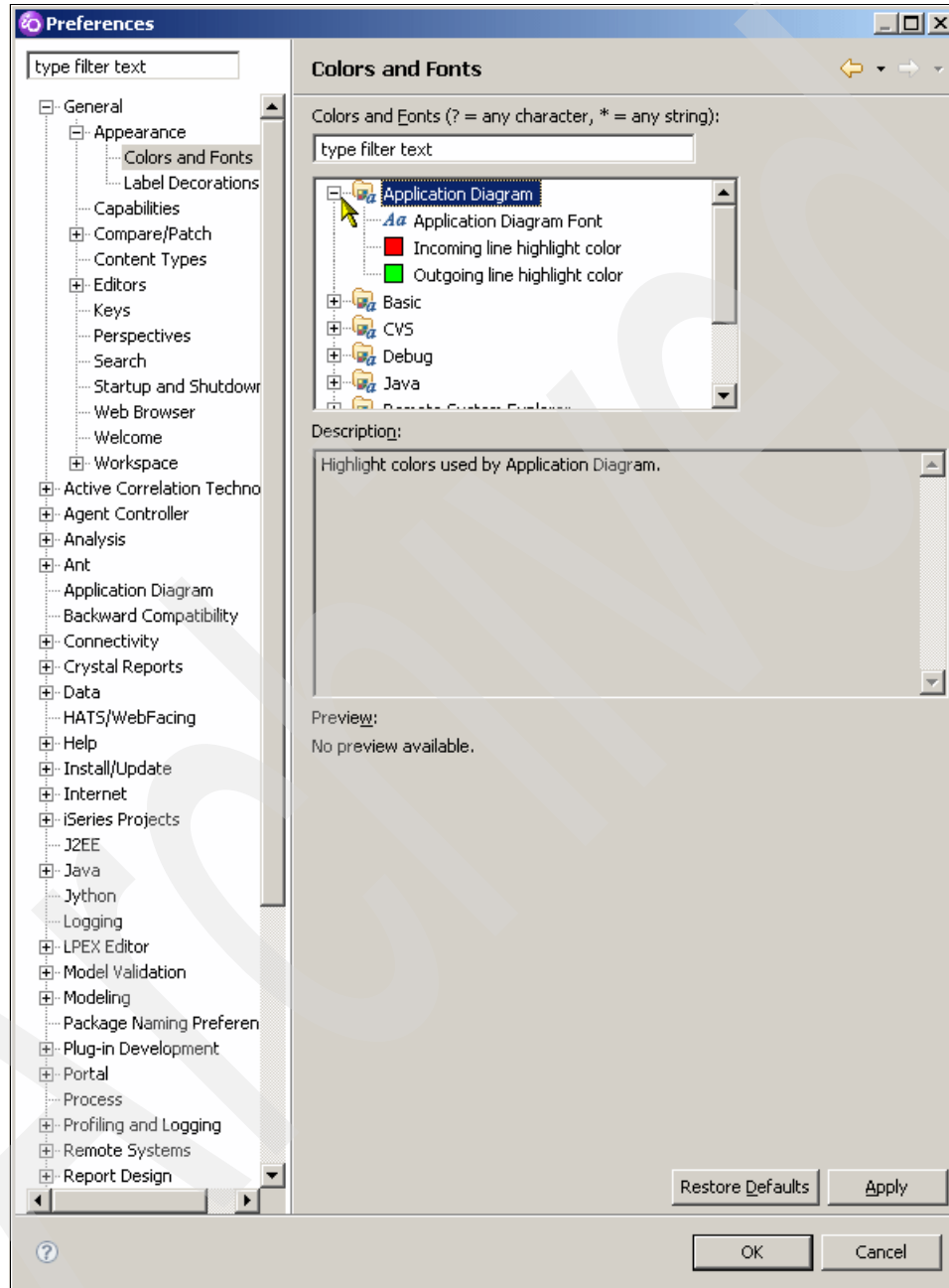


Figure 10-55 Application diagram color and fonts preferences

Archived

System i Projects

This chapter provides an overview of the System i Projects perspective. It introduces the concept of organizing your System i development resources as off-line projects. In this mode of development, no connection to the System i host is required for a majority of your development tasks. System i Projects can also be used for more structured System i application development.

We cover the following topics:

- ▶ The System i Projects perspective
 - Disconnected mode
 - System i connection and associated libraries
- ▶ Opening the System i Projects perspective
 - System i Project Navigator
- ▶ Creating a System i Project
 - Project containers, local source physical files and members
- ▶ Importing and viewing remote host objects
- ▶ Working with local source members
 - Offline source verify
 - Cached file descriptions
- ▶ Remote actions
 - Push operations, remote conflict detection
 - Compile actions
 - Build operations, build styles
- ▶ Working disconnected actions
- ▶ System i projects enhancements in V7
- ▶ Team support and version control
 - Using Rational ClearCase

References to more Information

You can also find information about the Application Diagram in the online help. From the menu select, **Help** → **Help Contents**, then expand

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Developing System i server applications using Remote System Explorer**
- ▶ → **Developing System i project-based applications.**

There is also a tutorial under **Help** → **Tutorials Gallery** and then expand

- ▶ **Do and Learn**
- ▶ → **System i**
- ▶ → **Develop System i applications with the Remote System Explorer**
- ▶ → **Work offline with System i projects.**

You may either follow the online tutorial or download and print a PDF attached to the first page. If you do the latter, make sure to read and perform at least the steps described under **Introduction** → **Prerequisites** → **Restore the WFLABXX savefile (WFLABXX.savf) on a System i system** → **Import the WFLABXX project to your workspace**, since it provides the sample application which you can install onto your own System i.

11.1 The System i Projects perspective

A project is a container that enables top-level organization of your resources in the workbench. A project contains files and folders. Projects are used for:

- ▶ Building applications
- ▶ Version management
- ▶ Sharing and organizing resources

A project can contain session and persistent properties, settings for environmental variables, and references to other projects.

The primary focus of the System i Projects perspective is on disconnected System i development on a Windows platform. The intent is that you edit your System i resource locally. When you finish updating your files and members locally, you upload these resources to the System i host and submit a build of your application. You can add a save file from the Remote Systems view, and also push a save file from a project to a System i server.

11.1.1 Working in disconnected mode

The System i Project perspective allows for disconnected development. A live connection is required only when code updates to the remote host or application builds are needed, or when you need to view remote resources for a project. If you are working on a remote laptop or mobile computer, or are at home with a slower connection, you may choose to work in disconnected mode. With this option, you explicitly copy files to your PC, laptop, or mobile computer while you have a live connection, close the connection, work on the files at your leisure and then upload them back to the host after you finish.

Another advantage to working in disconnected mode is that you can verify source code for syntax and semantic errors without being connected. And you can submit a compile only when you are connected.

This System i Projects development life cycle requires two workbench entities:

- ▶ A System i connection
- ▶ An associated library

System i connection

You *must* associate a System i connection to a System i project. The connection allows you to remotely access the System i host. It also manages user ID and password authentication to the remote host. For more information about System i connections, see 7.2.2, “Connections” on page 158.

Associated library

The associated library is a library on the remote System i host to which a System i project maps. This remote host library is the target of code uploads and application builds submitted from the System i project. It is expected that this library be treated as a “personal” developer library. Ultimately, there should be no need for any source in this library to be edited directly on the host.

This concept is in line with most change management procedures. That is, “production” code is normally “checked out” into a developer’s own sandbox area. And this sandbox is where a developer makes changes to the code.

11.2 Opening the System i Projects perspective

Before you can start working with System i projects, you *must* open the System i Projects perspective in the workbench area.

To open the System i Projects perspective, follow these steps:

1. From the workbench menu, click **Window** → **Open Perspective**. You can also click the **Open Perspective** () button on the toolbar on the left-hand side of the workbench.
2. Select **System i Projects**. If this option does not appear on this submenu, select **Other...** The Select Perspective window opens. Select **System i Projects** from this window and click **OK**.

The System i Project perspective has four main views (see Figure 11-1):

- ▶ **System i Project Navigator:** The System i Projects Navigator shows all System i projects that currently reside in your workspace.
- ▶ **Remote Systems view:** The Remote Systems view enables browsing of various remote systems, including System i servers. This view is paramount to the operation of the Remote System Explorer (RSE) perspective. For more information about the RSE and how the Remote Systems view is used, see Chapter 7, “Working with i5/OS-based development resources using the Remote System Explorer” on page 139, and Chapter 8, “Remote Systems LPEX Editor: The modern SEU and much more” on page 249.
- ▶ **Properties view:** Displays various properties about the object in the current perspective. By default, the Properties view appears in the lower left pane together with the Remote Systems view. By clicking the tab at the bottom of that pane you can bring each view into the foreground.
- ▶ **The Tasks view:** The Tasks view displays feedback on recent tasks performed.

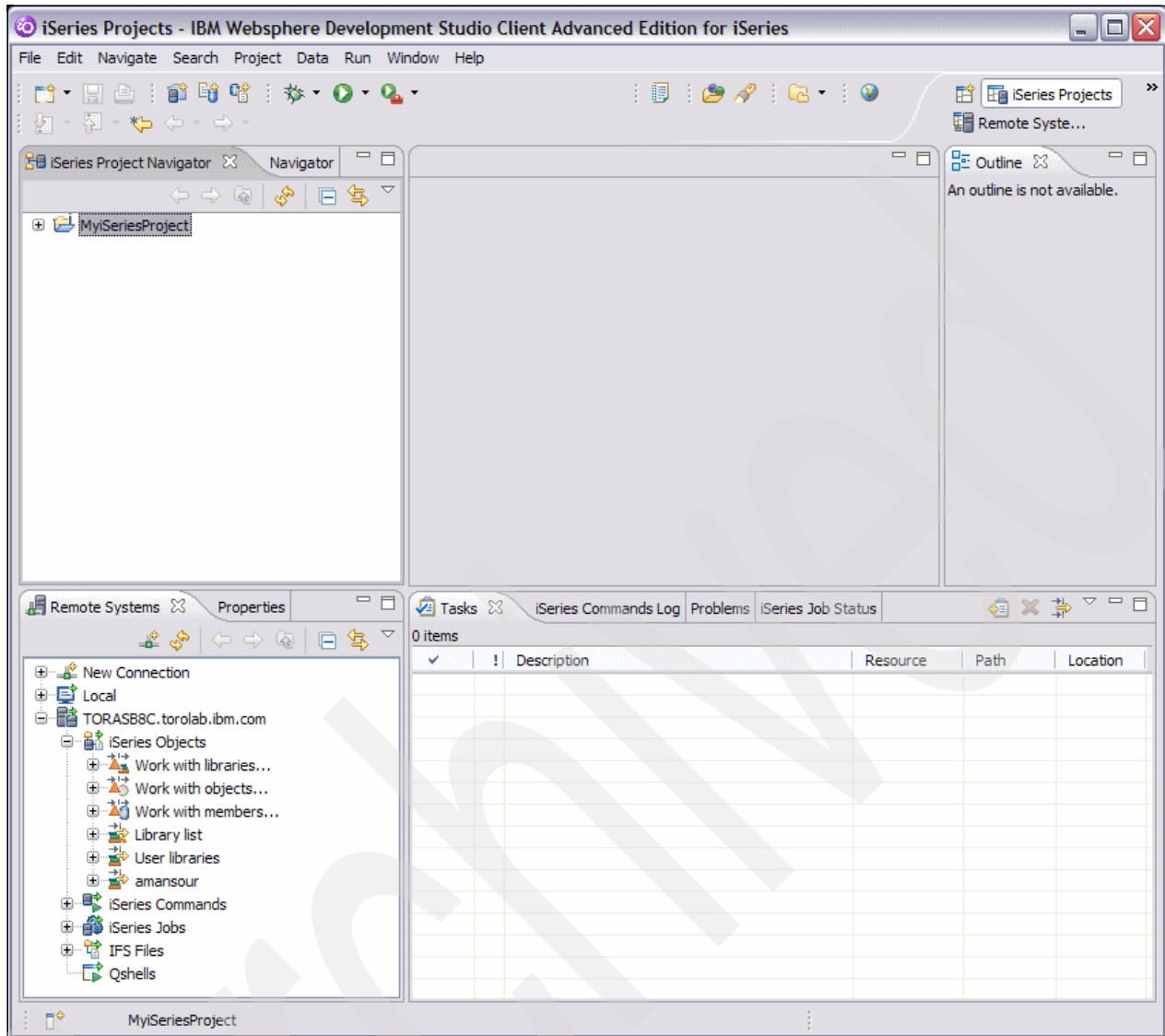


Figure 11-1 The System i Projects perspective

Note: Each time that you exit, the workbench is automatically saved, including all open perspectives and windows. The next time you open the workbench, it will appear exactly as it was when you closed it.

For more information about perspectives and views and how to work with them, see 2.2.3, “Perspectives” on page 21 and 2.2.4, “Views” on page 23.

11.3 Creating a System i project


Creating a System i project provides you with a workspace on your local workstation that contains the resources for the project. The System i Project Navigator view allows you to work while disconnected. It uses occasional synchronization with a host to do pushes and builds.

11.3.1 Creating the project container

You can create a System i project from the workbench menu option or from files and members context menu in RSE library.

From the workbench menu

To create a System i project container from the workbench menu, follow these steps:

1. Switch to the System i Projects perspective. From the workbench menu, click **Window** → **Open Perspective** → **Other** → **System i Projects**.
2. Perform one of the following actions:
 - From the workbench menu, select **File** → **New** → **System i Project**.
 - Click the arrow button to the right of the Open the New Wizard button  and select **System i Project**.
 - Right-click anywhere in the System i Project Navigator view and select **New** → **System i Project**.
3. In the System i Project window (Figure 11-2), complete the following tasks:
 - a. In the Project name field, enter the name of your System i Project.
 - b. For Project contents, deselect the **Use default location** box. This option stores the contents of your project under the workspace directory to which the workbench currently points.
 - c. Click **Next**.

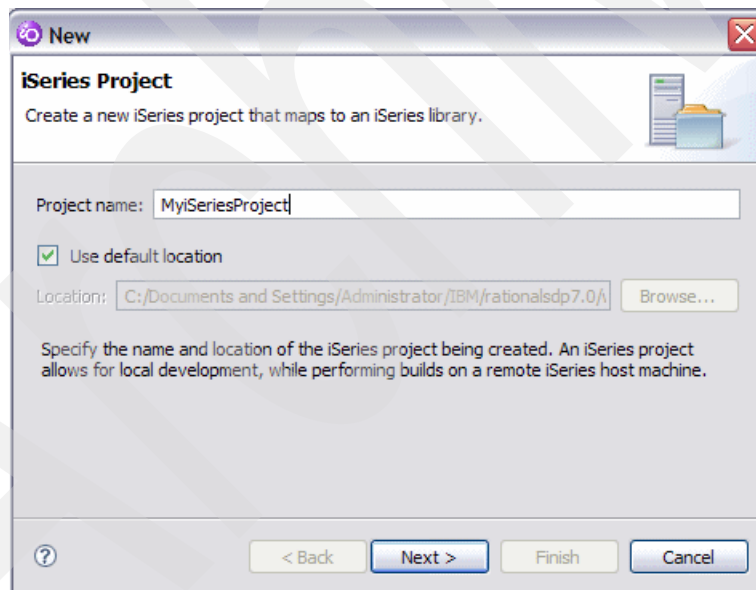


Figure 11-2 New - System i Project window: Entering the project name

4. In the System i Project Properties window (Figure 11-3), complete the following tasks:
 - a. From the Connection drop-down list, select an existing System i connection, or click **New** to create a new connection. For more information about creating connections, see 7.2.2, “Connections” on page 158.
 - b. In the Associated library field, enter the name of the remote library to which this project maps. If this library does not exist on the remote server, it will be created the first time that you push your code to the System i server. Or click **Browse** to select the associated library from a list of libraries on the System i host.

- c. Click **Next**.

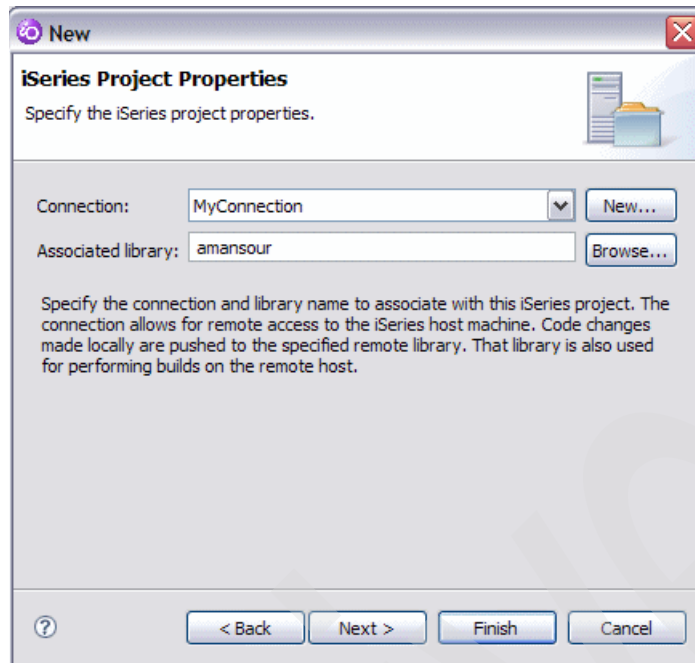


Figure 11-3 New - System i Project Properties window: Specifying the connection and associated library

5. There are three *active build styles* to choose from. Build styles specify the set of actions performed when a build of the application is submitted to the System i host. For more information about build styles, see “Build styles” on page 386, and “Specifying build styles” on page 386. For now, select **CL Program** as shown in Figure 11-4.
6. Click **Finish**. The newly created project appears as a node in the System i Project Navigator view.

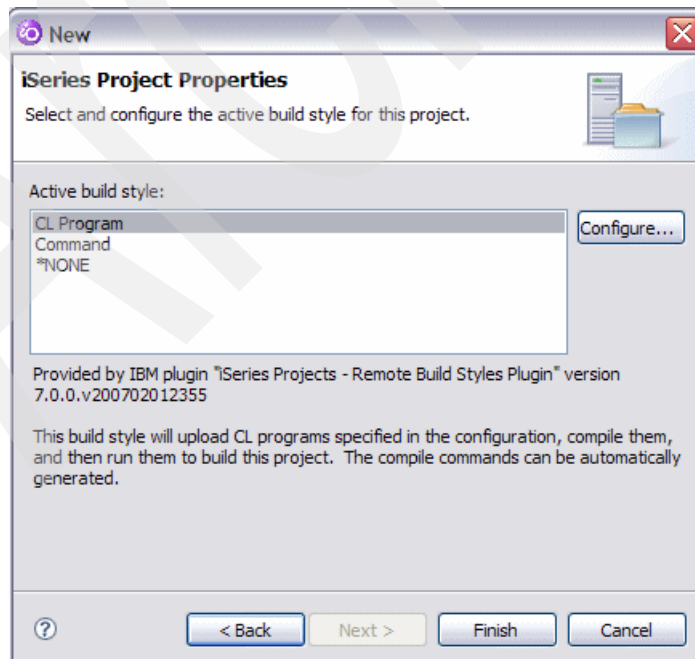


Figure 11-4 New - System i Project Properties window: Specifying the active build style

To start using System i Projects, you will need to populate them with source physical files and members. You can do this by creating new source files and members locally, or by importing remote objects from the System i server. Refer to 11.3.2, “Creating a local source physical file” on page 366, and 11.3.3, “Creating a local source member” on page 368 for more information on these tasks.

From the RSE library

To create a System i project container from the RSE library, follow these steps:

1. In the Remote Systems view browse to the library which you want to associate with the System i Project.
2. Right-click the library, and from the context menu select **Create System i Project** as shown in Figure 11-5.

A new project, associated with your library, will be created in the System i Project Navigator view.

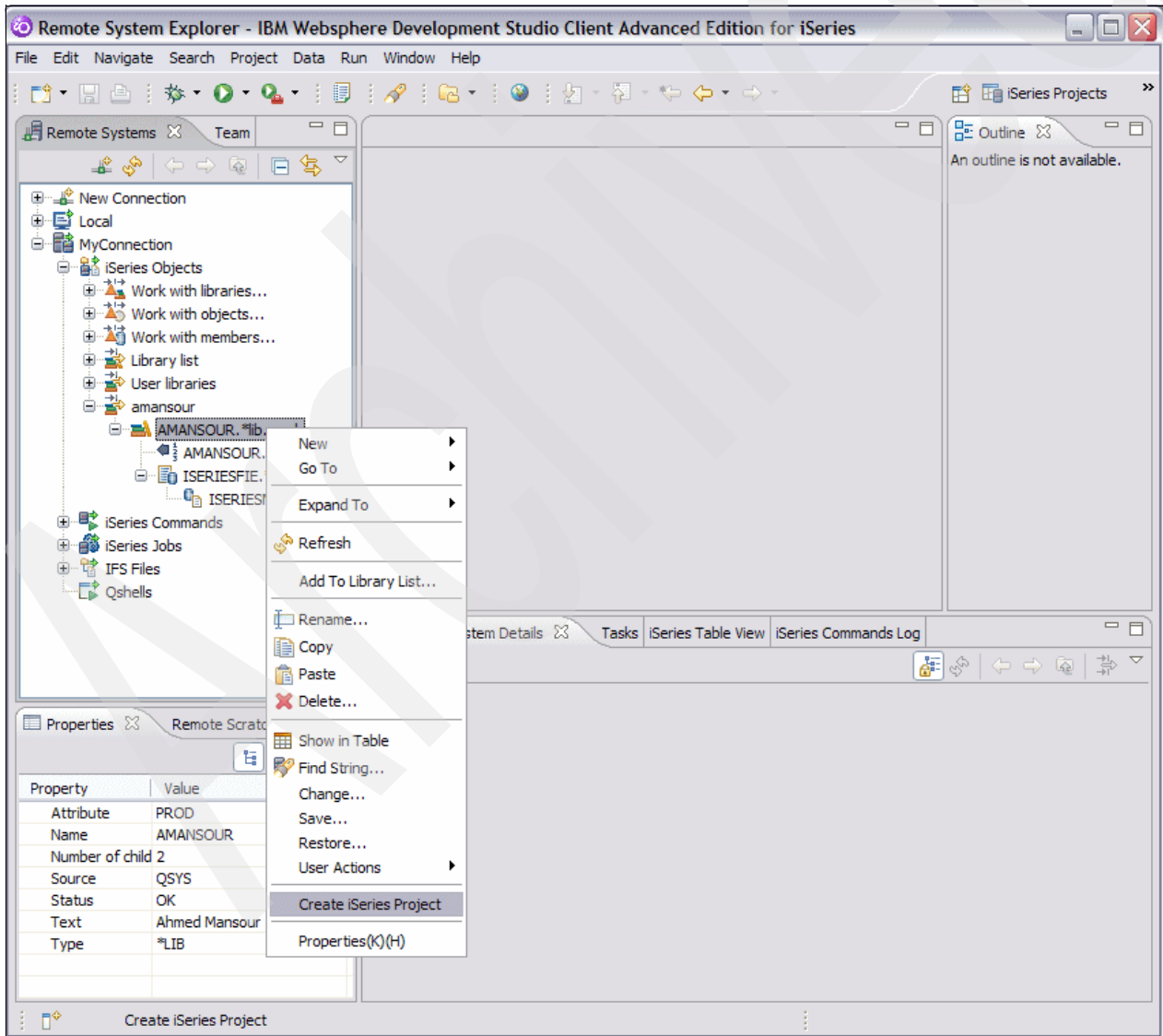


Figure 11-5 New - System i Project from the RSE library

To add files and members to a System i project from the RSE library, follow these steps:

1. In the Remote Systems view, expand your library.
2. Select the file or the member which you want to add to the project.
3. Right-click the selected item and from the context menu click **Add To System i Project...** as shown in Figure 11-6.
4. Select a System i project where objects will be downloaded and click **OK** as shown in Figure 11-7.

Note: If you have only one System i Project in System i Project Navigator, the selected file or member will be downloaded to this project.

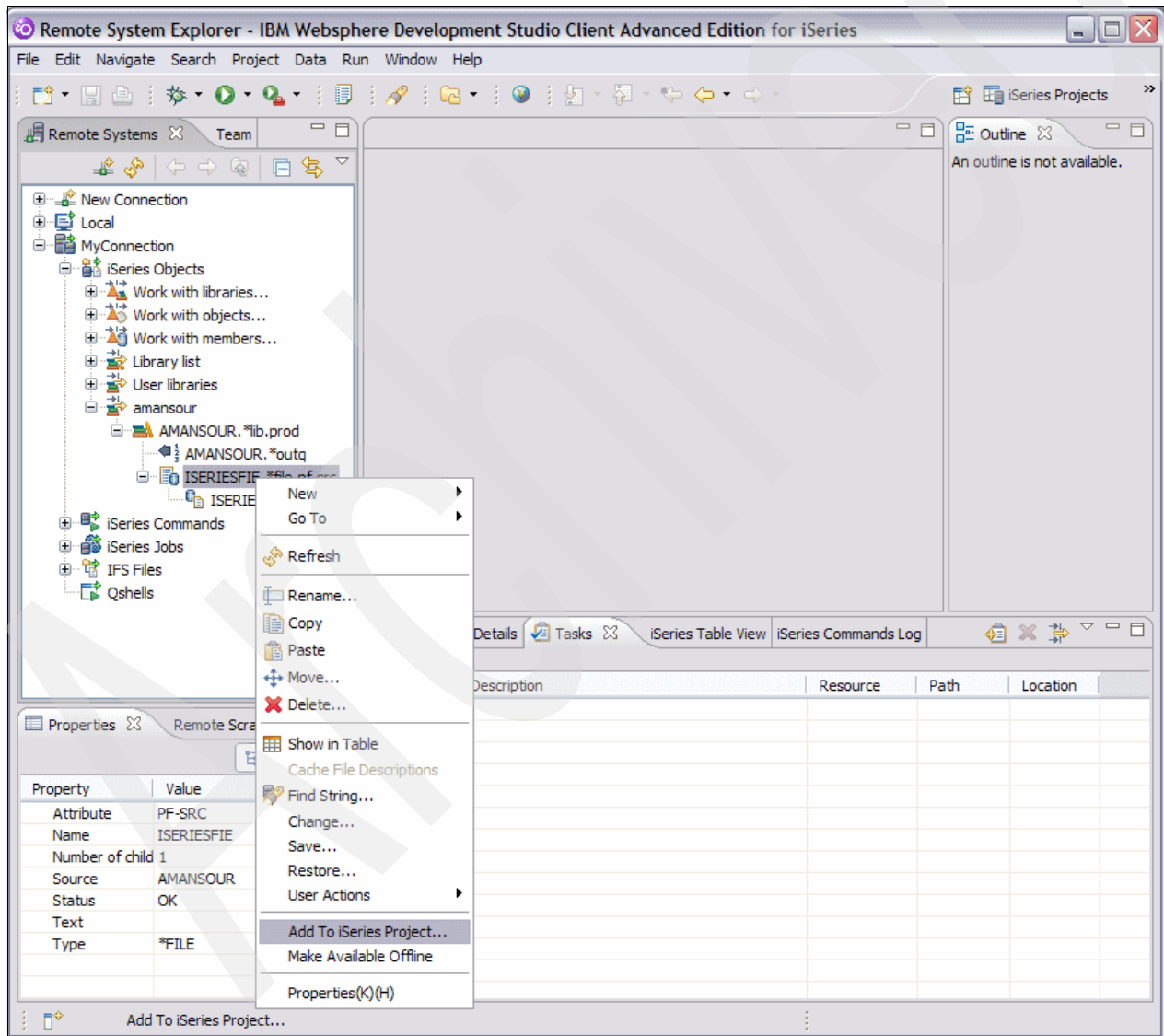


Figure 11-6 Add file to System i project from the RSE library

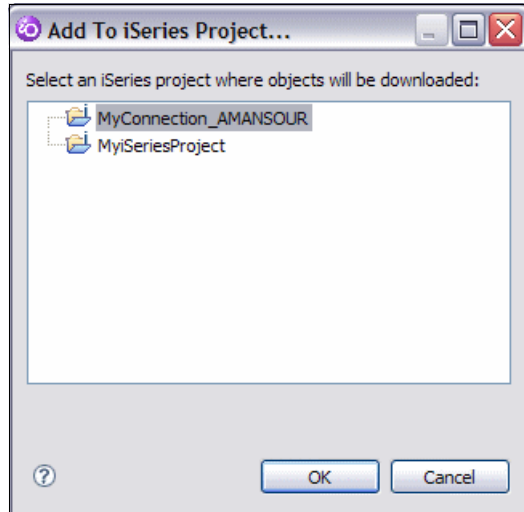


Figure 11-7 Add file to a System i project - select System i project

11.3.2 Creating a local source physical file

To start using System i projects, you need to populate your project with source physical files and members. In 11.3.1, “Creating the project container” on page 362, we created a local project container to store the contents of the project. We can now create a local source physical file to hold source members. Creating a local source file creates a windows folder of the same name on your local system in the workspace.

If you created new source files and members in a local project, you can select **Push Selected** or **Push Changes** to upload them to the server.

To create a local source physical file, follow these steps:

1. Right-click the System i project node in the System i Project Navigator view and select **New** → **System i Source Physical File**. The New - System i Source Physical File window (Figure 11-8) opens.
2. In the New - System i Source Physical File window, complete the following tasks:
 - a. Enter or select the project created in 11.3.1, “Creating the project container” on page 362. Valid System i Projects that can hold a source physical file are distinguished by a blue **i** on the top right hand corner of the project icon.
 - b. In the File name field, enter the name of the source physical file. If this source file does not exist on the host, it is created when any changes are pushed to the host or when a build of the project is submitted.
 - c. Click **Next**.

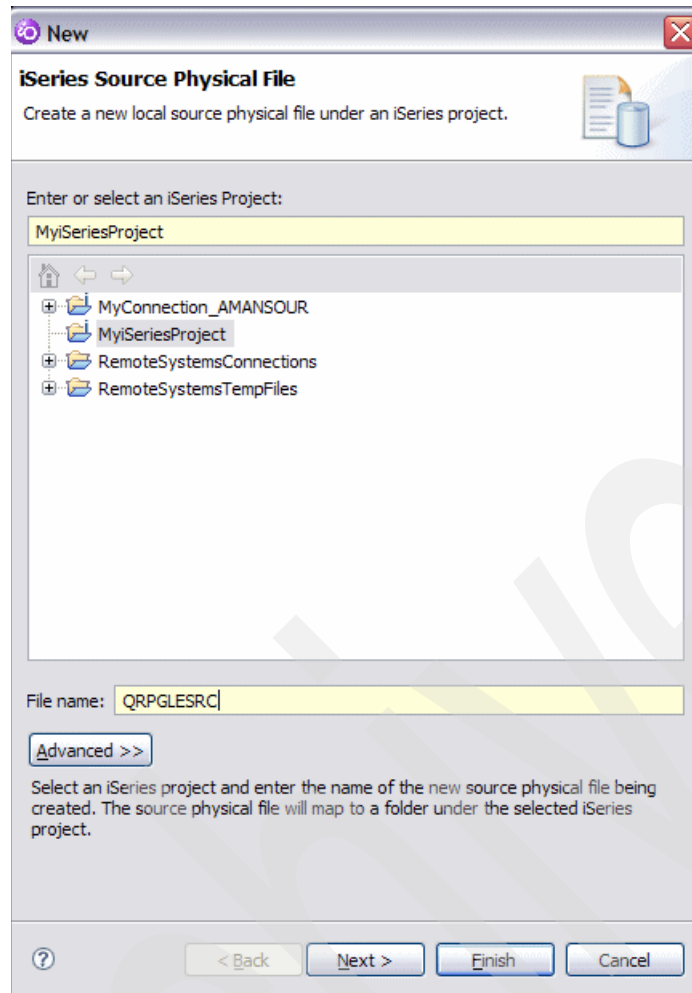


Figure 11-8 New - System i Source Physical File window

3. In the New - System i Source Physical File Parameters window, specify any additional parameters. We specify a value of 112 in the Record length field and enter some text in the Description field.
4. Click **Finish**. The source file is created and appears as a child node to the System i project in the System i Project Navigator view (Figure 11-9). If Show Remote Objects is enabled, the blue arrow pointing to the right indicates that a System i resource is local, and that the same resource does not exist in the remote library on the server. If Show Remote Objects is disabled, then the blue arrow means that the resource is a System i resource. To show remote objects, right-click the project in System i Project Navigator and select **Show Remote Objects** ().

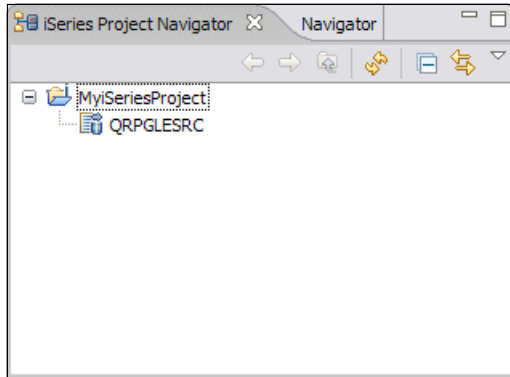


Figure 11-9 The local source physical file as a child node to the System i project

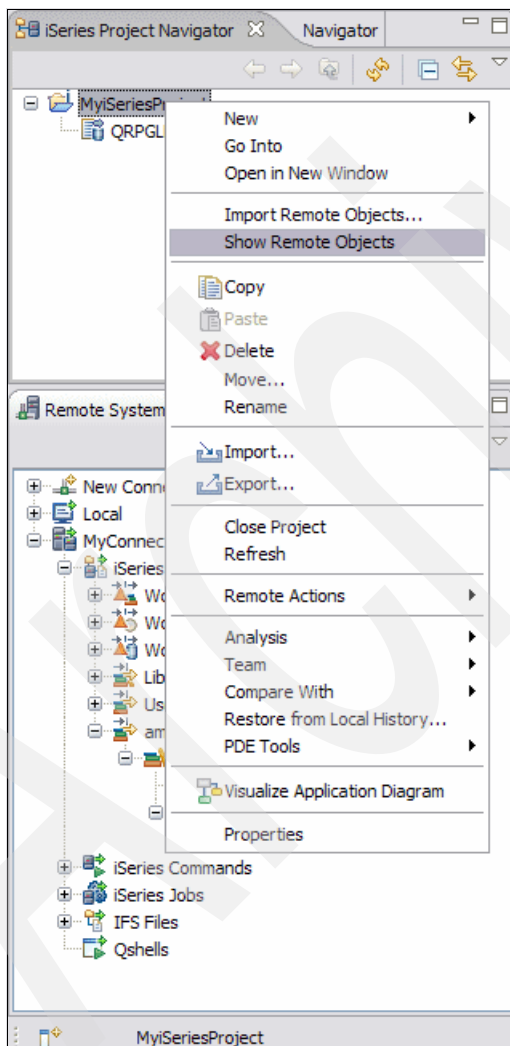


Figure 11-10 Show Remote Objects in System i Project Navigator

11.3.3 Creating a local source member

In 11.3.2, “Creating a local source physical file” on page 366, we created a local source physical file. We can now create a local source member in the local source physical file.

Creating a local source member creates a file that resides in the Windows folder that maps to the source physical file. The member will be created on the server the next time that you select **Push Selected** or **Push Changes**.

To create a local source member, follow these steps:

1. Right-click the System i local source physical file in the System i Project Navigator view. Select **New** → **System i Source Member**. The New - System i Member window (Figure 11-11) opens.
2. In the New - System i Source Member window, complete the following tasks:
 - a. In the Member name field, enter a name for the member. If this source member does not exist on the host, it is created when any changes are pushed to the host or when a build of the project is submitted.
 - b. In the Source type field, enter or select the source type from the drop-down list.
 - c. Click **Next**.

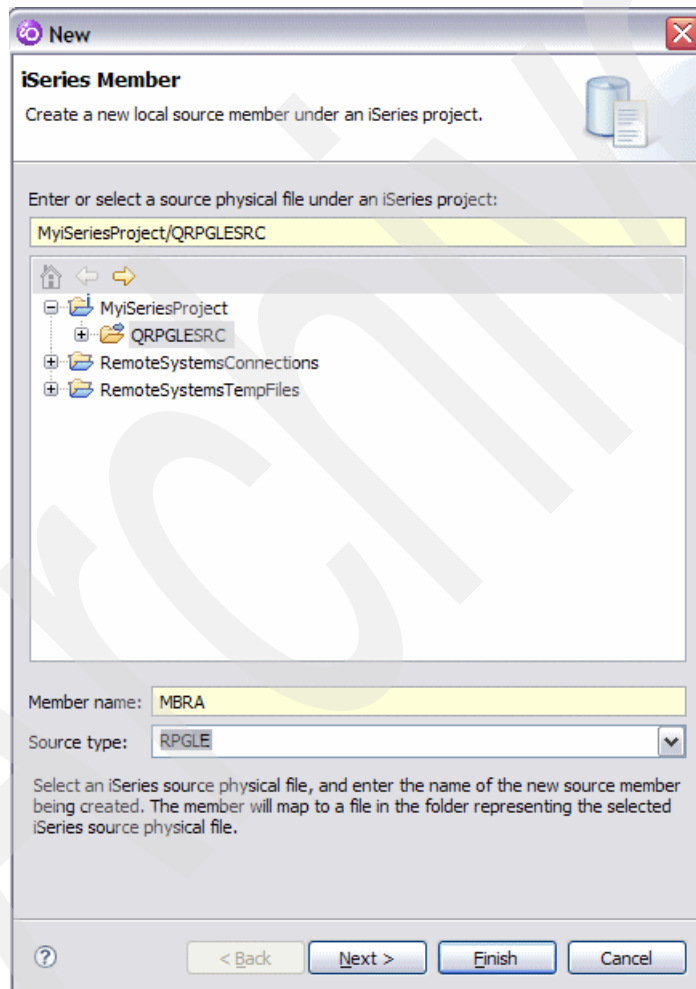


Figure 11-11 The New - System i Member window

3. In the Description field, enter a description for the source member.

Note: To modify the description after it has been specified, right-click the member and select **Properties**. The properties will be updated on the server the next time you select **Push Selected** or **Push Changes**.

4. Click **Finish**. The source member is created and appears as a child node to the local source physical file in the System i Project Navigator view (Figure 11-12). The source type of the member maps to the file extension of the local file created.

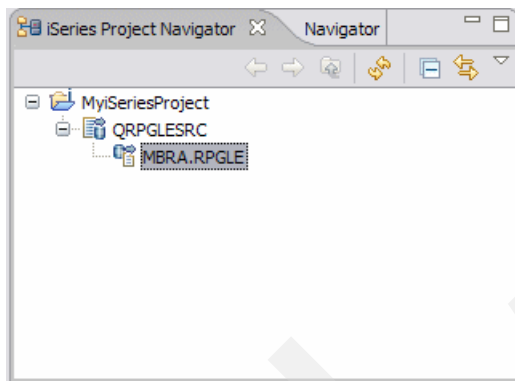


Figure 11-12 The local source member as a child node to the local source physical file

5. After the source member is created, the Remote Systems LPEX Editor launches and loads the member.

Note: If this is the first push of this member to this server and the associated library and the member already exist, it is assumed to be in conflict and a warning is issued.

11.4 Importing and viewing remote objects

In the previous section, we created local source files and members in the System i project container from scratch. Existing System i host objects can also be imported into a System i project. You may want to do this for a “maintenance” type of project for example.

Tip: You do not need to create an empty source physical file or member if you want to create a local copy of an existing source physical file or member. Simply perform the “import” action as explained in the following section. The local copy of all selected files or members is created in the System i project.

11.4.1 Importing remote host objects

You can import System i host source files and members into existing System i projects.

Important: If you import a source file or member from a library different than the library associated with the project, then, if you make any changes to the copy in the project and push changes the changes are *not* pushed back to the original member. Instead, they are pushed to the associated library.

For example: If you import member abc from library A into a System i project that is associated with library B. If you push changes to the member from the project they will be written to member abc in library B not in library A. In order to push changes back to library A you would need to create a System i project associated with that library.

To import remote host objects, follow these steps:

1. Right-click the project in the System i Project Navigator view and select **Import Remote Objects**.
2. In the Import Remote Objects window (Figure 11-13), expand the appropriate library filters to drill down to the source files and members you want to import. For more information about filters and how to create and use them, see 7.4.1, “Types of filters in the System i Objects subsystem” on page 183.
3. Select the source files or members you want to import. You can select source members by first expanding the appropriate source file node. If you select specific source members to import, a local folder is created that maps to the source physical file in which the member or members reside on the host. You can make multiple selections by using standard Windows keyboard/mouse combinations such as Ctrl+Click and Shift+Click. Right-click the object and select **Add To Project**.
4. Click **OK**. The source files or members are downloaded to the local system. The tree in the System i Project Navigator view is updated with the new information.

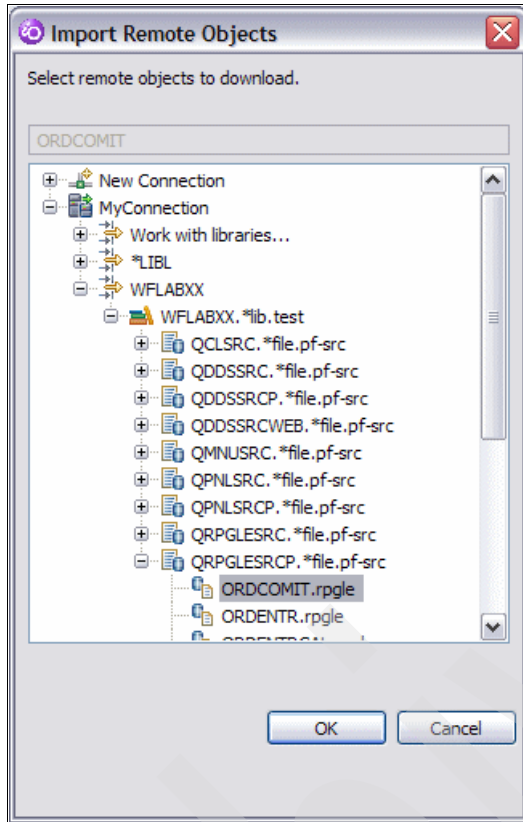


Figure 11-13 The Import Remote Objects window

Tip: You can also add remote source physical files and members to System i projects from the Remote Systems view. For more information, refer to “From the RSE library” on page 364.

11.4.2 Viewing remote objects

In the System i Project Navigator view, you can display both local and remote objects at the same time.

To view remote objects, follow these steps:

1. Right-click the project node in the System i Project Navigator view.
2. Select **Show Remote Objects**. This action displays all the remote objects that reside in the remote library specified as the associated library of the project.

The distinction between local and remote objects is made clear through the use of *decorators*. Decorators are small arrows on the icons for each source file and member (see Figure 11-14). For more information about decorators, refer to 11.8.1, “Label decorations” on page 390.

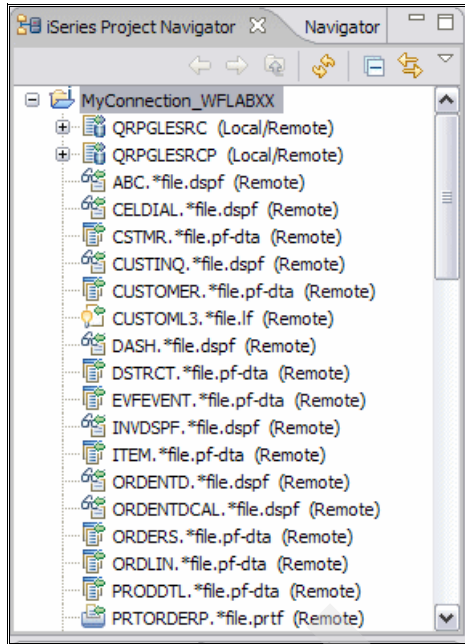


Figure 11-14 Showing local and remote resources

11.5 Working with local source members

This section discusses a cross-section of tasks that can be performed on local source members.

11.5.1 General management

You can copy, delete, move, and rename local source members. To gain access to these tasks, right-click the source member and select the appropriate action from the pop-up menu.

11.5.2 Editing

All that has been said about editing remote source members in the RSE also applies to local source members. For more information, see 8.1, “Editing source members in the RSE” on page 250.

11.5.3 Source verify (offline)

As stated in 11.1, “The System i Projects perspective” on page 359, the primary focus of the System i Projects perspective is on disconnected (or offline) development. With this in mind, you can also perform source verifications offline as well. Information pertaining to your resources is cached locally. This information is used by the source verifier to perform validations on your source code. A live connection is required only when code updates or builds are needed.

To work offline and use the local cache, follow these steps:

1. In the Remote Systems view, expand your connection and right-click the source physical file or member that you want to work with offline.

2. Select **Make Available Offline**; see Figure 11-15. If a System i project does not exist for the library and connection that contains the source physical file or member, a new System i project is automatically created. By default, the project name will be System iConnection_LibraryName. Since the project has been created using the library and connection values, the next time that the code is pushed to the server from the project, it will be pushed to the original library that it came from. If one or more System i projects are found, you will be asked to use an existing project or create a new one.

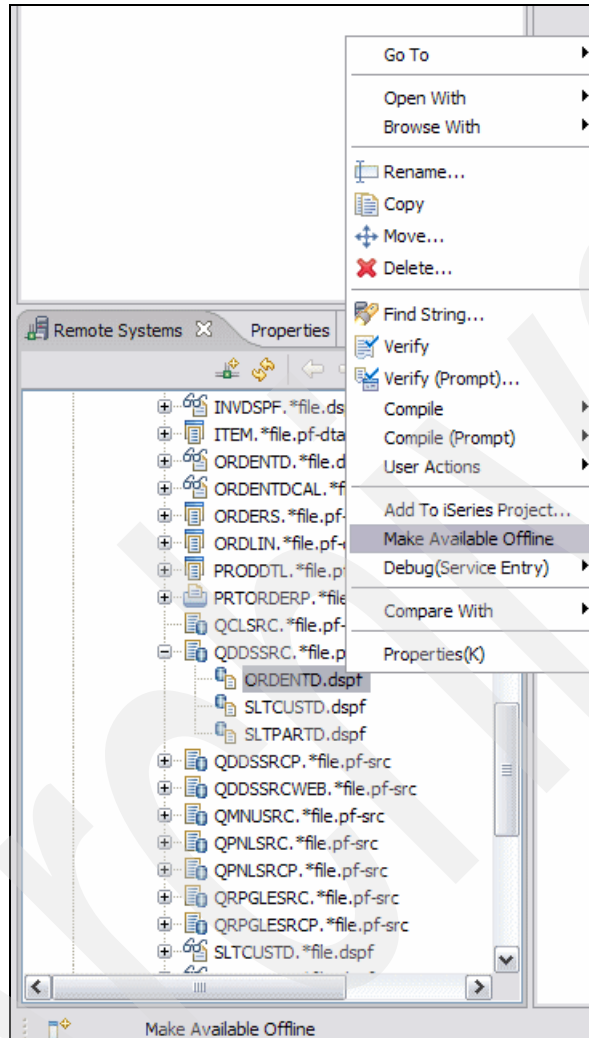


Figure 11-15 Working offline - Make Available Offline

3. To see the System i project, select **Window** → **Open Perspective** → **Other**. In the Select Perspective list, select **System i Projects**.

To verify your local source members, follow these steps:

1. Double-click the source member to open it in the LPEX editor.
2. From the workbench menu, select **Source** → **Verify** or **Source** → **Verify with Prompt**. You can also right-click the source member and select **Verify** in the pop-up menu; see Figure 11-16.

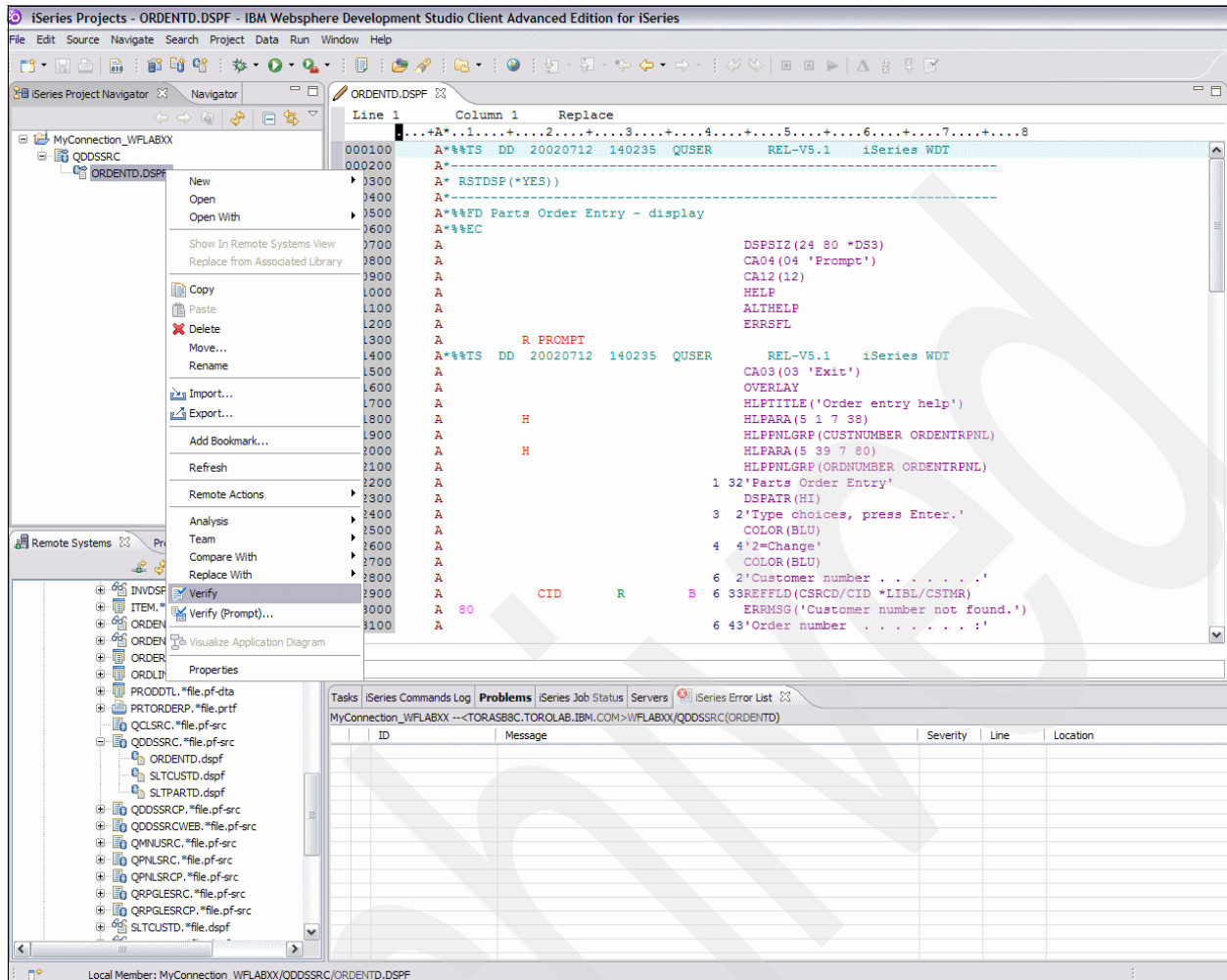


Figure 11-16 Working offline - Verify local source members

11.5.4 Putting your connection into offline mode

You can also put an entire System i connection, and all the contents defined with that connection, into offline mode. This method of working offline prevents the Remote System Explorer from attempting to connect to the System i server if a request is made for remote information.

For example, if you are verifying a local source file or member in a System i project, the project would attempt to connect to the server if the required information was not available in the cache. This can be prevented by switching the connection to offline mode. There are two ways to do this:

1. Right-click your System i connection and select Work Offline, see Figure 11-17.
2. Right-click the System i connection and select Properties. Then select the Work Offline check box in the properties dialog box; see Figure 11-18.

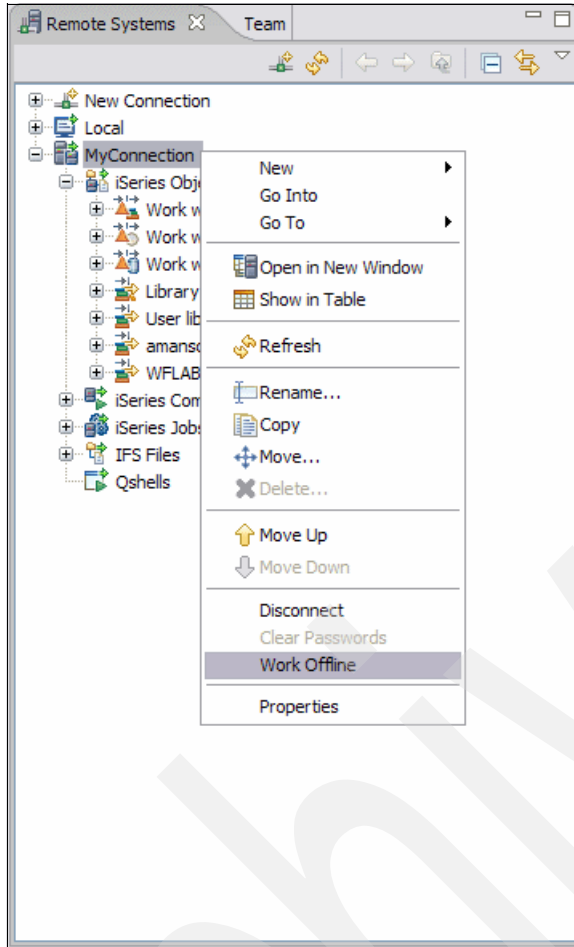


Figure 11-17 Putting your connection into offline mode from the connection pop-up menu

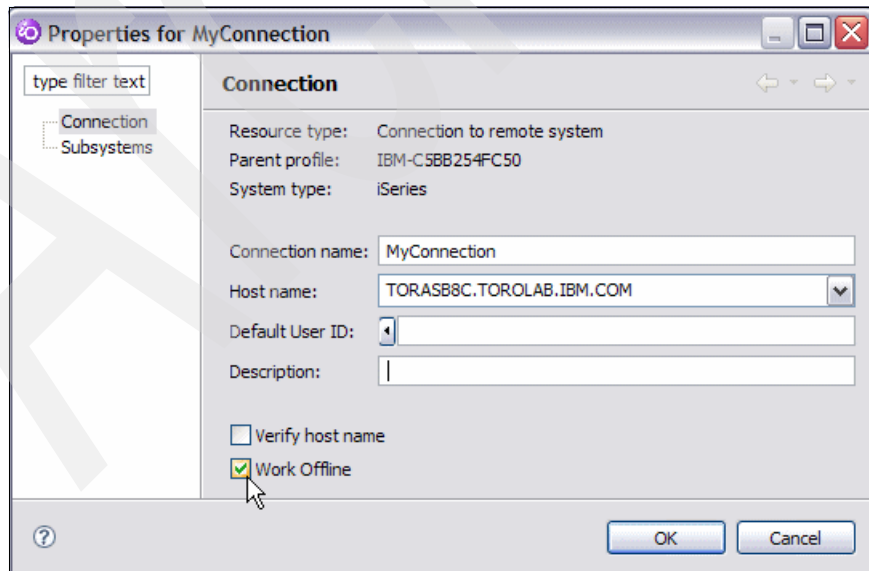


Figure 11-18 Putting your connection into offline mode from the connection properties dialog

Caching file descriptions

If your source members reference remote files, whether they be data, display, or printer files, you *must* cache information for these remote files for offline source verification to work.

If you verify once while connected then all required external descriptions are automatically cached. If you want to update the cache or cache file descriptions for files you think you might need (but have not used recently) then you can perform the following steps:

1. In the Remote Systems view of the RSE, drill down through your filters and find the file (*FILE) object or object that you want to cache.
2. Right-click the file object and select **Cache File Descriptions**. You must perform this action on the actual *compiled file object*, and *not* the associated source member. See Figure 11-19.
3. A Progress Information window opens. It indicates that information about the file object is being cached to the local system.

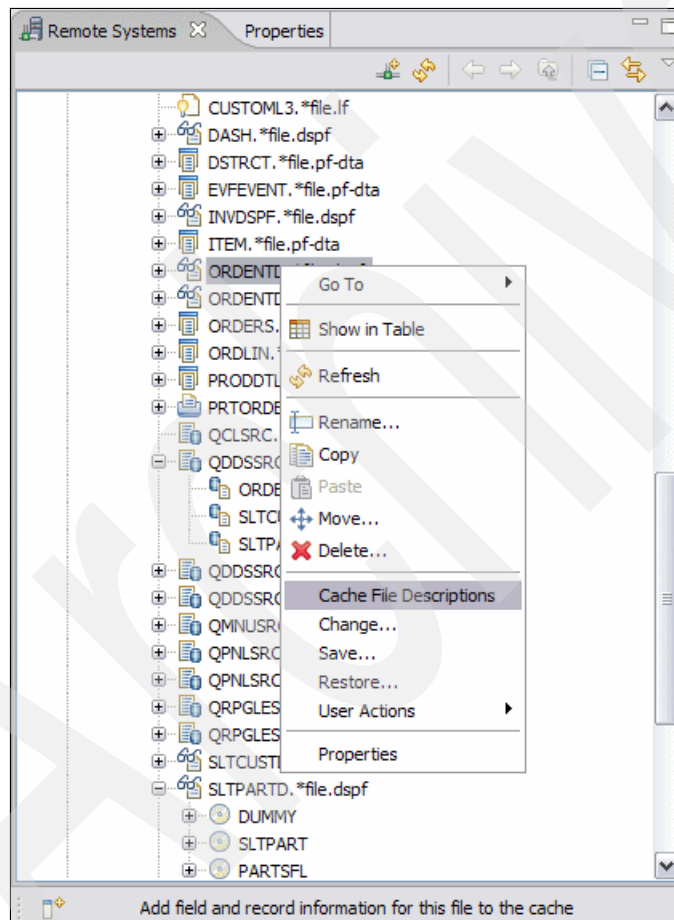


Figure 11-19 Choosing Cache File Descriptions for compiled file object

11.6 Remote actions

This section explains the processes for uploading (or pushing) a System i project to the remote host and building your project on the host. The actions associated with these processes are invoked from the Remote Actions menu option.

To access this menu option, follow these steps:

1. Right-click the project, any project resource, or anywhere in the System i Project Navigator view.
2. Select **Remote Actions**.

11.6.1 Push operations

Push operations enable you to upload your project resources to the remote System i host. This ensures that the content of your local and remote source members is synchronized. Push operations are activated when either one of the Command or CL Program build styles are specified for the project. See “Specifying build styles” on page 386.

There are two types of push operations:

- ▶ Push Changes
- ▶ Push Selected

Push Changes

This operation pushes any *new* or *changed* resources found in the selected project. An item is changed if it was updated locally since the last time it was pushed by either a Push Changes or a Push Selected action. The first time it is done, all resources are uploaded to the associated library. Each subsequent upload is incremental.

To invoke the Push Changes action, follow these steps:

1. Right-click either the System i project, any project resource, or anywhere in the System i Project Navigator view.
2. Select **Remote Actions** → **Push Changes**. The Progress Information window opens to indicate the status of the upload of the changed project resources to the System i host, as shown in Figure 11-20.

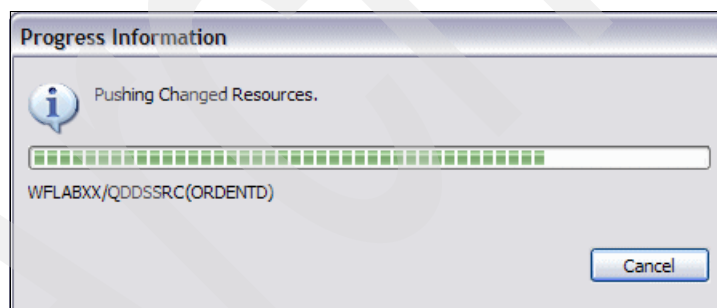


Figure 11-20 Pushing System i project changes to the host

Push Selected

This action uploads all *selected* resources and their children to their corresponding System i source files and members in the associated library. It creates any resources that do not exist. It overwrites any changes that were made on the host system only if confirmed by the user:

- ▶ Any changes that have been made locally since the last push operation are pushed to the associated library.
- ▶ The Commands view will also show these source files and members as their properties are modified on the server.

Use this action to force the contents of the associated library to match the workspace.

To invoke the Push Selected action, follow these steps:

1. Right-click the project or project resources you want to select for this action.
2. Select **Remote Actions** → **Push Selected**. The Progress Information window opens to indicate the status of the upload of the selected project resources to the System i host as shown in Figure 11-21.

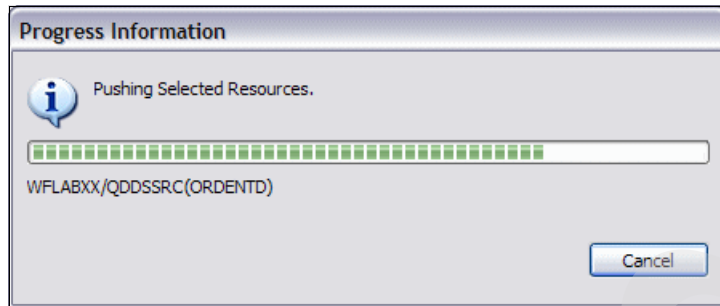


Figure 11-21 Pushing System i project selections to the host

Note: You can use Show Remote Objects to compare the resources that are in the workspace with those that are in the associated library. To do this, right-click your System i project and select **Show Remote Objects**. This will show that all members are now both local and remote.

Remote conflict detection

When a remote source physical file or member is imported into a System i project, its last modified timestamp is cached internally for later comparisons. When you select Show Remote Objects at the System i Project level or select the Push Selected or Push Changes action, you are warned if that object was modified on the remote host.

Conflicts on push operations can happen in two situations:

- ▶ There is no locally stored member timestamp for a member and the member exists on the host. This happens if any of the following scenarios occur:
 - The source member was created in the workspace without retrieving it from the host.
 - The member is renamed.
 - The member is copied or moved.
 - The host system is changed in the connection definition.
 - The connection is changed, causing the host system to change.
 - The associated library is changed.
 - The parent source file is renamed, copied, or moved.
- ▶ There is a locally stored member timestamp, the member exists on the host, and its timestamp is different than the stored one. This happens if any of the following scenarios occur:
 - The member on the host system is updated after the local source member is pushed.
 - The member on the host system is restored from a backup copy.

If there is a conflict, you are prompted with a Conflict Detected warning window (Figure 11-22):

- ▶ If you click **Yes**, you overwrite the source member on the host.
- ▶ If you click **Yes to All**, all source members that conflict for the duration of this push operation are overwritten without further warning.

- ▶ If you click **No**, this source member is *not* pushed to the host.
- ▶ If you click **No to All**, all source members that conflict during this particular push operation are skipped without further warning.
- ▶ If you click **Cancel**, the push operation is canceled.

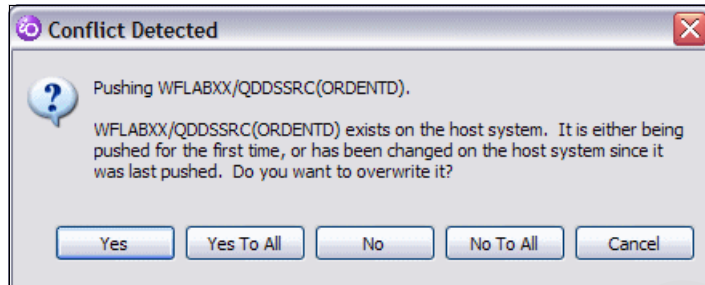


Figure 11-22 Remote conflict detected

11.6.2 Compile actions

Compiling System i project members

You can compile members in a System i project, one or several at a time, either submitted to a batch job or compiled immediately. When you compile, the default is that the workbench first pushes any changes that have been made in the project to the System i server, and then compiles the members you have selected. You can change the build and compile options in System i Project preferences. The two options that are available are “Push all changed resources in the project” (the default option), and “Push selected resource if there are any changes.”

To compile a single member:

1. Expand your System i project until you see the member that you want to compile.
2. Right-click, select **Remote Actions** (see Figure 11-23), and then select one of the following:
 - **Compile** to compile the member without prompt. The member is compiled using the current compile command for its source type.
 - **Compile (Prompt)** to compile the member with prompt, that is, to specify extra parameters.
 - **Select Compile Command** to make a certain command the current one for this source type.
 - **Work With Compile Commands** to add, change, or delete associated compile commands.

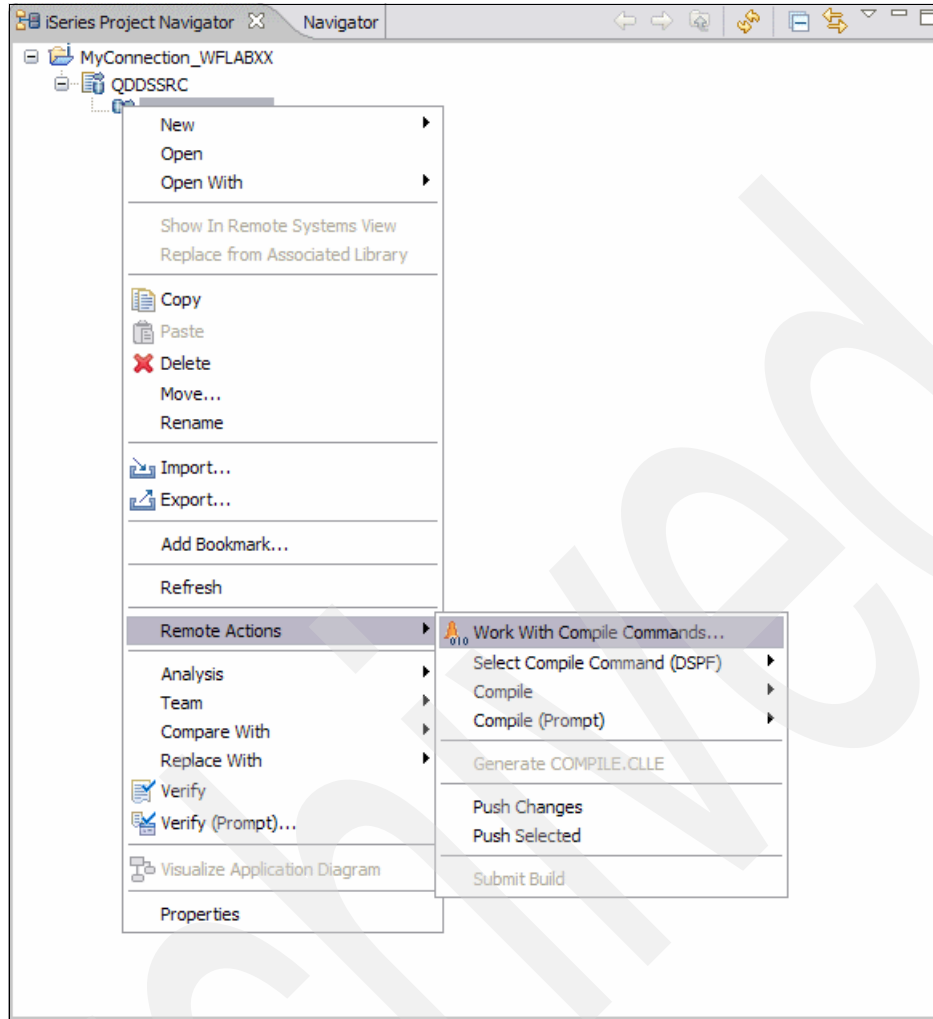


Figure 11-23 Compiling a single System i project member

To compile multiple members:

1. Expand your System i project until you see the members that you want to compile.
2. Select the members by holding down the Ctrl key and clicking each member.
3. Right-click, select **Remote Actions**, and then select one of the following (Figure 11-24 on page 382):
 - **Compile** to submit a compile command for each selected member according to the order based on their type. (This order can be changed through the preferences as mentioned below. See “Modifying compile order preferences:” on page 383.) The members are then compiled using the current compile command for their source type.
 - **Work With Compile Commands** to add, change, or delete associated compile commands.

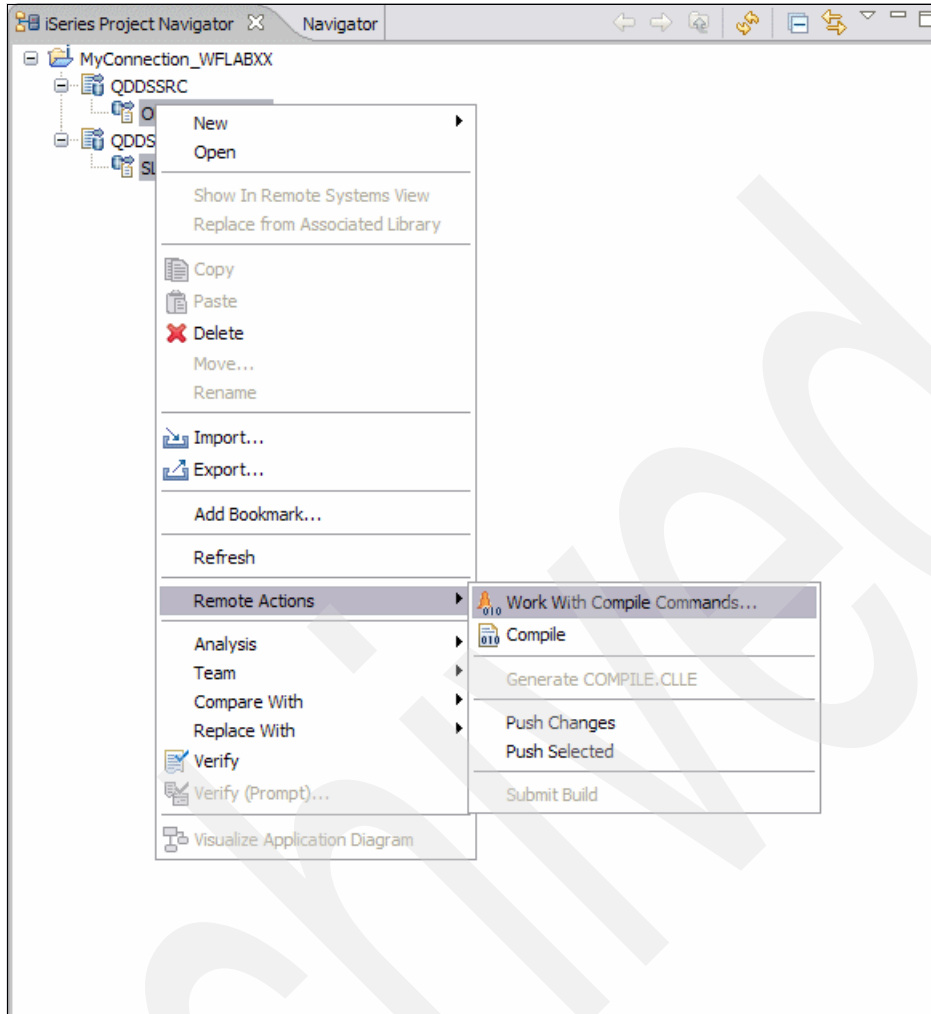


Figure 11-24 Compiling multiple System i project members

Batch versus immediate mode:

- ▶ If you compile in batch mode, all selected members are compiled in a single batch job. The progress of the submitted job displays in the System i Job Status view.
- ▶ If you compile in batch mode and use OPTION(*EVENTF) in your compile commands, the compile errors are gathered in a combined event file that you can retrieve after the action is performed.
- ▶ If you compile in immediate mode, then the compile commands are submitted serially. Any compile errors are shown in a separate tab for each member in the error list.

You can modify the batch versus immediate preference using the Build Style preference page. Click **Window** → **Preferences** from the workbench menu. Then expand **System i Projects** and select **Build and Compile**. Note the check box for **Run compiles as batch jobs**. If selected, compiles from System i projects are submitted in a batch job. If deselected, compiles are run in immediate mode; see Figure 11-25.

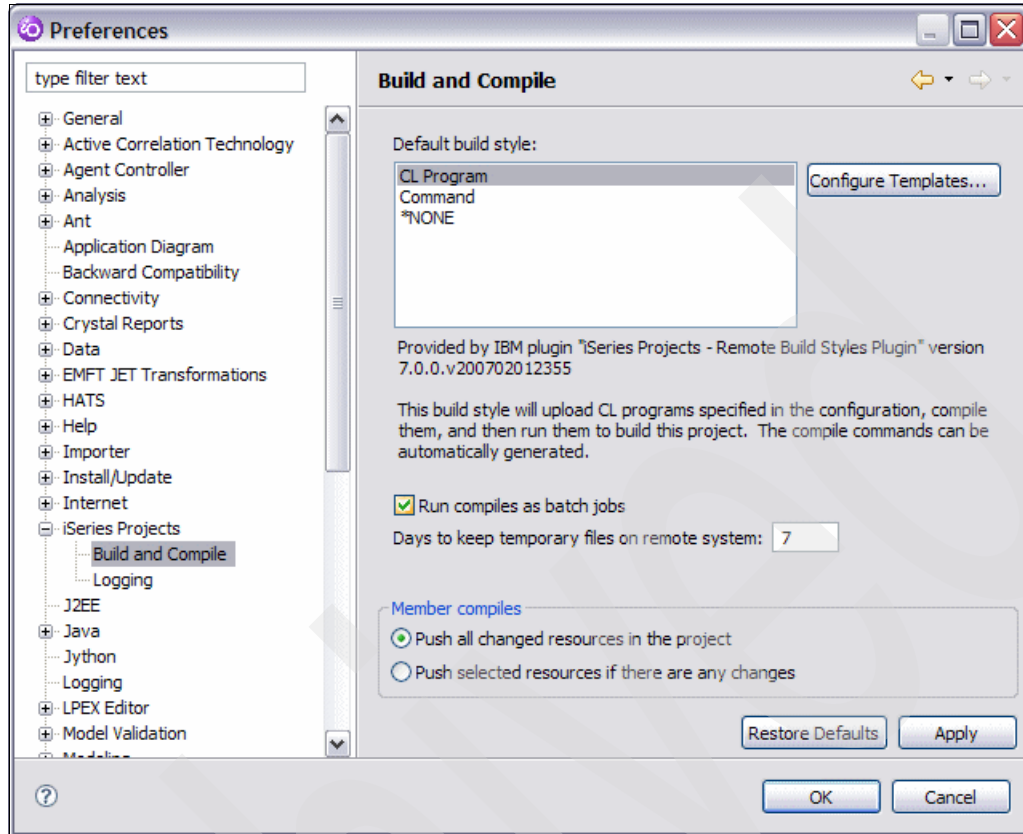


Figure 11-25 Modifying the batch versus immediate preference

Modifying compile order preferences:

When you compile more than one source member at a time, the order in which the source members are compiled follows the member type ordering in the System i Command Execution preference page. To access this preference page, click **Window** → **Preferences** from the workbench menu. Then expand **Remote Systems** → **System i** and click **Command Execution**. In the **Compile member types in this order** area, you can select various member types and move them up or down in the hierarchy list; see Figure 11-26.

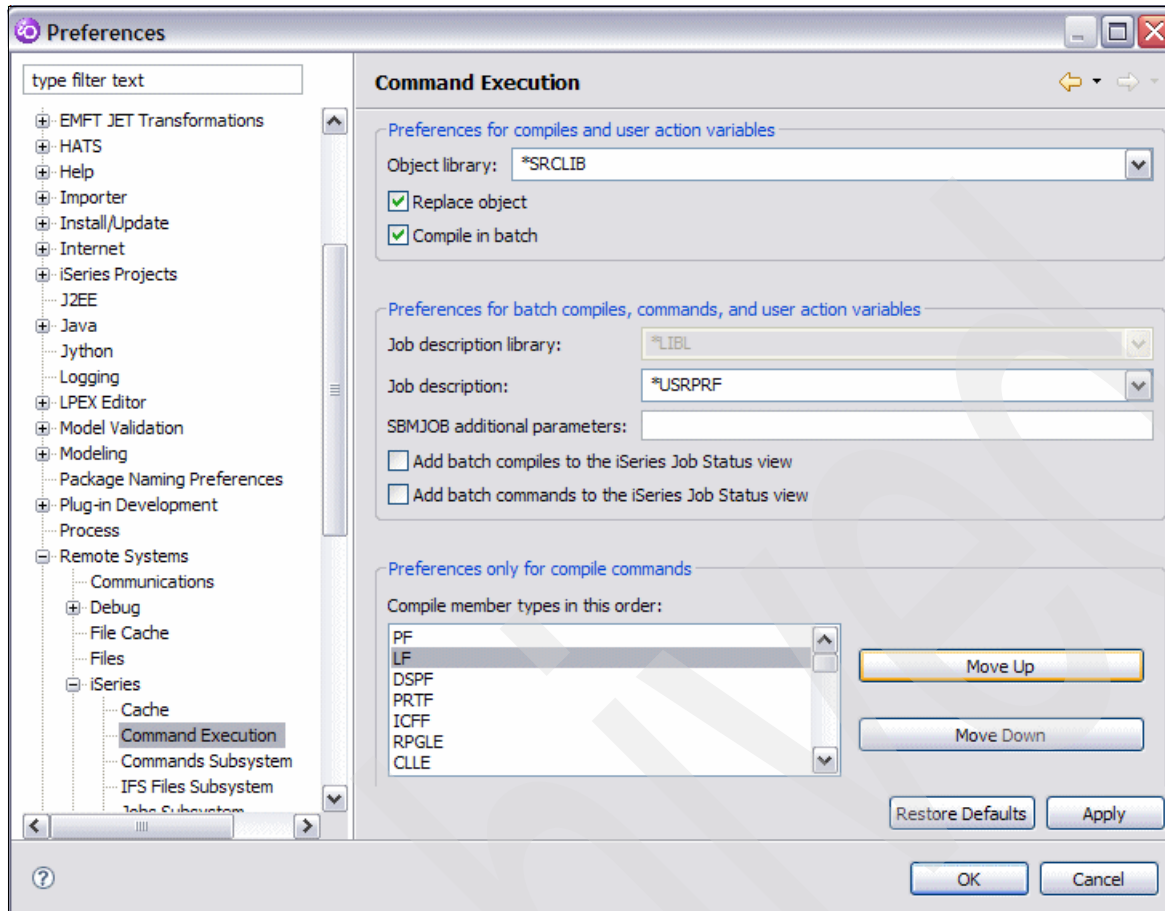


Figure 11-26 Modifying compile order preferences

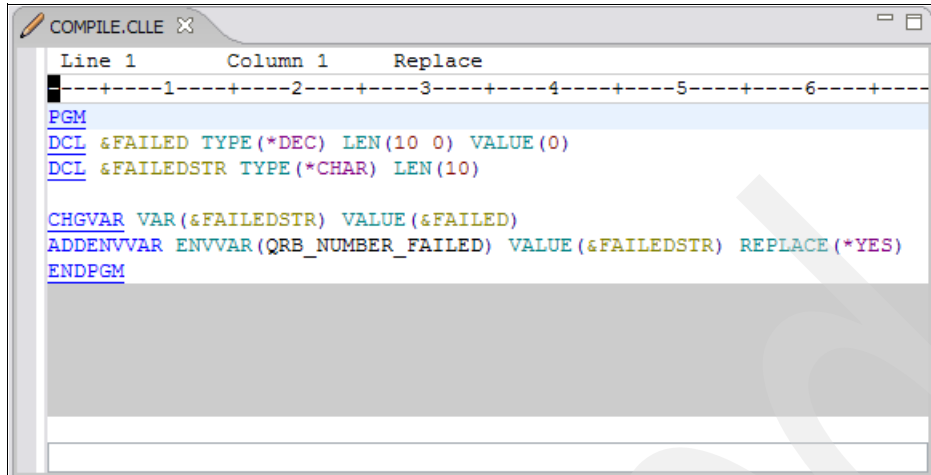
Generating compile code

The Generate Compile Code action is applicable to System i projects that have a CL Program build style configured. For more information about build styles, see “Build styles” on page 386.

Invoking this action generates (or overwrites, as the case may be) a source member named **COMPILE.CLLE** in the source file specified in the build style configuration. This source member contains the commands that compile your project source members.

To invoke the Generate Compile Code action, follow these steps:

1. Right-click either the project, any project resource, or anywhere in the System i Project Navigator view.
2. Click **Remote Actions** → **Generate Compile Code**. The Progress Information window opens to indicate the current status of the action.
3. After the action is completed, the **COMPILE.CLLE** member is created (or re-generated) in the source file specified in the build style configuration. The source file and member appear as child nodes to your project in the System i Project Navigator view.
4. Double-click the **COMPILE.CLLE** member to view its contents in the default editor. See Figure 11-27.



```
COMPILE.CLLE X
Line 1      Column 1      Replace
-----1-----2-----3-----4-----5-----6-----
PGM
DCL &FAILED TYPE(*DEC) LEN(10 0) VALUE(0)
DCL &FAILEDSTR TYPE(*CHAR) LEN(10)

CHGVAR VAR(&FAILEDSTR) VALUE(&FAILED)
ADDENVVAR ENVVAR(QRB_NUMBER_FAILED) VALUE(&FAILEDSTR) REPLACE(*YES)
ENDPGM
```

Figure 11-27 COMPILE.CLLE source member

Managing COMPILE.CLLE

In many situations, the choice of compile commands used in the COMPILE.CLLE source member may *not* be suitable in the context of your project. This is due to the method that the Generate Compile Code action applies in deciding which compile command to use. The command chosen is based on the *member type* of the project source member only. The last command run to compile a member of that type in the RSE is the command it uses in the COMPILE.CLLE source member. For example, in the case of project source members with a type of RPGLE, if the last compile command invoked on a member of type RPGLE in the RSE was Create Bound RPG Program (CRTBNDRPG), then this command is used for all RPGLE project source members in the COMPILE.CLLE member.

In many cases, this is *not* an acceptable scenario. You will, without doubt, have times when you require both bound programs and modules in your project and these objects require two different compile commands. This is when you need to manage the content of the COMPILE.CLLE source member. We recommend that you refrain from using the Generate Compile Code action. We also recommend that you *deselect* the Automatically generate COMPILE.CLLE prior to each build option in the build style configuration at the project level in this situation. See “Specifying build styles” on page 386.

The functionality of the Generate Compile Code action was *not* fully developed at the time of writing this book. It is expected that more flexibility will be implemented for this action in a future release of WebSphere Development Studio Client for iSeries.

ILE-based projects

For Integrated Language Environment® (ILE)-based projects, where modules are bound together to create programs and service programs, you *must* create a local source member named BIND.CLLE. This source member must be stored in the source file specified in the build style configuration. There is no action to create this member for you automatically. The BIND.CLLE member contains the commands that bind your project’s module objects together to build the required programs or service programs.

11.6.3 Build operations

Build styles

A build style is a set of actions that provide *push* and *build* operations for your System i projects from the Remote Actions menu. You select one of the styles while creating a new System i Project. There are three IBM-supplied build styles:

- ▶ ***NONE:** This build style contributes no actions. It is useful if you want to disable the menu items associated with a build style. This style is always available.
- ▶ **Command:** The command build style contributes the following actions to the Remote Actions menu:
 - Push Changes
 - Push Selected
 - Submit Build Command
- ▶ **CL program:** This build style contributes the following actions to the Remote Actions menu:
 - Generate Compile Code
 - Push Changes
 - Push Selected
 - Submit Build

Specifying build styles

Each System i project has its own build style. Two types of active build styles are available:

- ▶ ***NONE:** This build style contributes no actions and is useful if you want to disable the menu items associated with a build style. This style will always be available.
- ▶ **Command:** This style provides operations to push (upload) resources of the project to the associated library. It also provides an operation to run a command you supply in a batch job. The command you supply can be any command. However, it is expected to take the source in the associated library and create programs, service programs, and files from it.
- ▶ **CL program:** This style also provides operations to push (upload) resources to the project's associated library. It compiles and runs two CL programs:
 - **COMPILE:** This program compiles source members into programs or modules. It is created from the COMPILE.CLLE source member in the source file specified in the configuration of the build style. This source member can be generated automatically or created and maintained manually. See "Specifying a build style at the project level".
 - **BIND:** This program creates programs and service programs after the COMPILE program has run successfully. It is created from the BIND.CLLE source member in the source file specified in the configuration of the build style. If you require this source member, then you *must* create and maintain it manually.

Specifying a build style at the project level

To specify a build style for a System i project, follow these steps:

1. Right-click the System i project and select **Properties**.
2. In the left side pane, select **System i Build**.
3. Select **Command** or **CL Program** from the list and click **Configure**.
 - If you select Command, complete these tasks:
 - i. In the Build Command field, specify the command that is used to build the project.
 - ii. Click **OK**.
 - iii. Click **OK**.

- If you select CL Program (see Figure 11-28), complete these tasks:
 - i. In the Build Source File field, specify the source file that contains the members that will be uploaded to build the project.
 - ii. Select **Automatically generate COMPILE.CLE prior to each build** to automatically generate the COMPILE.CLE file in the designated source file before you submit the build to the host. Deselect this option if you want to manually control the content of the COMPILE.CLE source member.
 - iii. Select **Automatically push all changed members prior to build** so any changed members are automatically created or updated in the associated remote library prior to the project build. In general, we recommend that you select this option.
 - iv. Click **OK**.
 - v. Click **OK**.

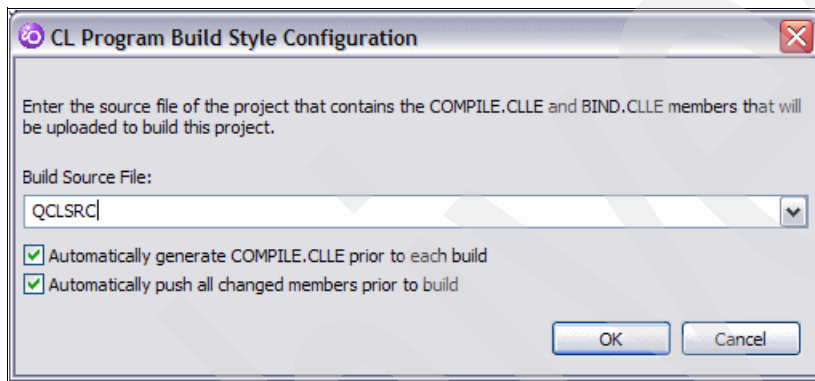


Figure 11-28 CL Program Build Style Configuration window

Specifying a default build style

You can also specify a default build style that applies to all *new* System i projects:

1. From the workbench drop-down menu, select **Window** → **Preferences**.
2. In the left side pane, expand **System i Projects** and select **Build and compile**.
3. Select your default build style from the list and click **Configure Templates**.
4. Follow the steps specified in “Specifying a build style at the project level” on page 386 for the default build style selected.
5. Click **OK**.

Important: Changes made to the default build style do *not* apply to existing System i projects. Only new projects inherit the default build style. As discussed in “Specifying a build style at the project level” on page 386, the default build style *can* be overridden at the project level.

Building your System i project

This section explains how to build your System i project on the remote System i host.

The build process

The high-level tasks performed when you request a build of your System i project associated with a CL Program build style can be summarized as follows:

1. Any project resources that changed since the project was last uploaded are pushed to the System i host. This only occurs if you selected the **Automatically push all changed**

members prior to build option in the build style configuration. We recommend that you keep this option selected.

2. The build job is submitted to batch on the remote System i host. The associated library for the project is made the *current library* for the submitted job. The rest of the library list is as specified for the connection that is associated with the project.
3. The COMPILE.CLLE and BIND.CLLE source members are compiled to create the COMPILE and BIND programs in the QTEMP library. For more information about these source members, see “Generating compile code” on page 384.
4. The COMPILE program is run. This program compiles your project’s resources.
5. The BIND program is run. This program binds your compiled objects together to create the ILE programs and service programs for your project.

Submitting project builds

To build your System i project on the remote System i host, follow these steps:

1. Right-click the project, any project resource, or anywhere in the System i Project Navigator view.
2. Select **Remote Actions** → **Submit Build**. The Progress Information window opens to indicate the current status of the action.
3. After the build job is submitted on the remote host, the status of the job can be monitored from the System i Build Status view. See “The System i Job Status view” on page 388.

The System i Job Status view

This view displays all build jobs that are submitted to the remote host. When you submit a build job for a project using one of the Command or the CL Program build styles, the job is added to the *Job Status view* as shown in Figure 11-29.

Date/Time Submitted	Job	Connection	Status	Task
Jun 21, 2007 5:14:00 PM	480342/AMANSOUR/AMAN...	MyConnection	*OUTQ	MyConnection_AMANS...
Jun 21, 2007 5:20:00 PM	480355/AMANSOUR/AMAN...	MyConnection	*OUTQ	MyConnection_AMANS...
Jun 21, 2007 5:22:00 PM	480358/AMANSOUR/AMAN...	MyConnection	*OUTQ	MyConnection_WFLABX...

Refresh Interval: 10 seconds

Figure 11-29 The System i Build Status view

This view can be set up to update at regular intervals and monitor all System i Project build jobs that you submitted. The update interval is set via the refresh slider situated in the bottom left corner of the view. You can specify the refresh time in seconds. It takes some time to gather the information and it can only be gathered if a connection can be established to the System i host. Setting too small an interval may have an impact on performance. To set the refresh interval to “never refresh”, move the slider all the way to the right. To update the view, when the refresh interval is inactive, click the **Refresh** (🔄) button in the top right corner of the view.

When the job finishes, and compile errors are found during the build, right-click the job and select **Task Actions** → **Retrieve Errors** to populate the System i Errors List view with the compile errors found. From the System i Errors List view, you can double-click an error and

load the source member with the error or errors into the default editor and have the cursor positioned at the error. You can also remove jobs that finished from the System i Build Status View list. Simply right-click the job and select **Remove**. To end jobs that are queued or running, right-click the job and select **Task Actions** → **Cancel**. See Figure 11-30.

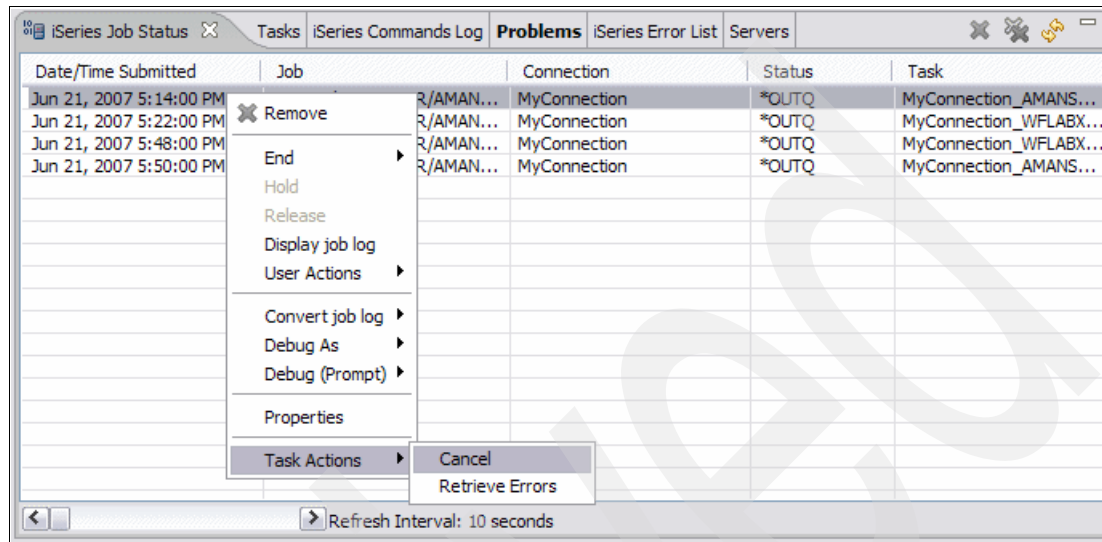


Figure 11-30 Invoking actions from the System i Build Status view

11.7 Working disconnected actions

This section summarizes the actions you can take and the actions you cannot take while working disconnected.

11.7.1 Actions applicable

The following list of activities can be performed off-line:

- ▶ Edit local and offline source files or members
- ▶ Syntax check RPG, COBOL, DDS
- ▶ Syntax check CL for cached commands
- ▶ Prompt RPG and DDS
- ▶ Prompt CL for cached commands
- ▶ Access language help for RPG, COBOL, DDS
- ▶ Access CL help for cached commands
- ▶ Verify RPG, COBOL, DDS for those members that are cached

11.7.2 Actions not applicable

You can perform the following actions only if you are connected to your System i server:

- ▶ Edit, design or compile host source members
- ▶ Run or debug host programs
- ▶ Run host commands
- ▶ Syntax check, prompt or F1 on CL commands not cached
- ▶ Verify members not previously verified

11.8 System i projects enhancements in V7

11.8.1 Label decorations

Label decorations are used to show extra information about an item by modifying its label or icon. They can be used to obtain information about the state of an item without having to look at its properties in the Properties view or open its Properties dialog box.

Decorators are usually found on the corners of the resource icons, suffixed or prefixed for text decorations. Decorators could be either an image or icon decoration, or both.

System i projects decorations are now visible in views other than System i Project Navigator view.

To enable or disable System i Project icons and decorations:

1. Go to **Window** → **Preferences** → **General** → **Appearance** → **Label Decorations**.
2. Select or clear the check box for **System i Projects Local/Remote Decorator**.
3. Click **OK** or **Apply**; see Figure 11-31.

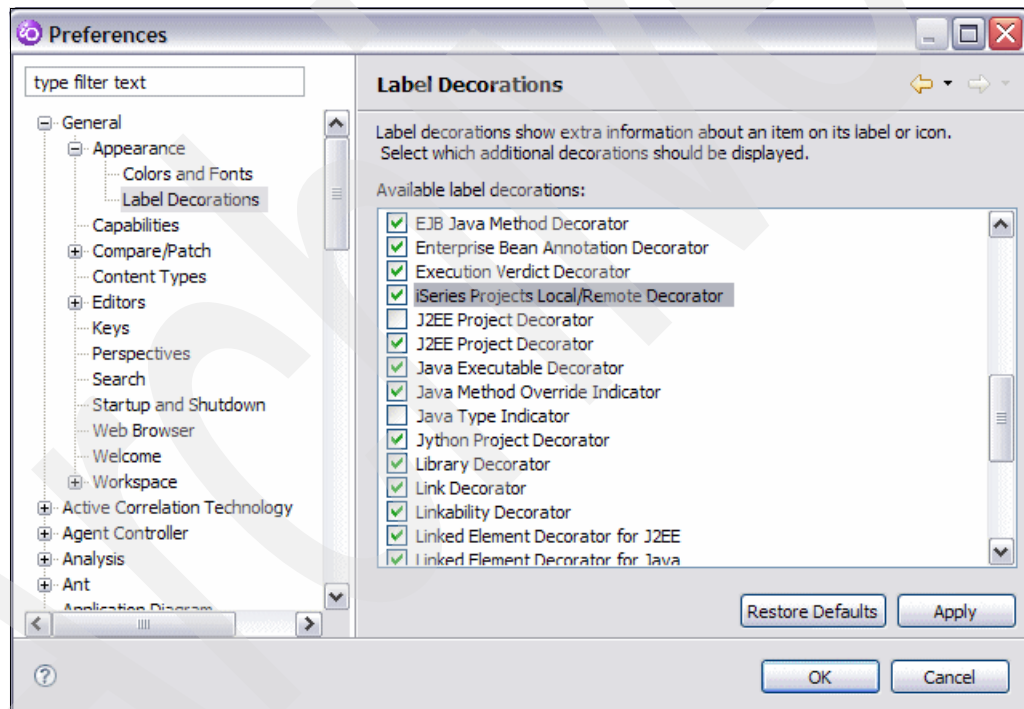


Figure 11-31 Enable System i Projects Local or Remote Decorator

To show the decoration, select the **Show Local/Remote Decorator** option on the System i Projects pull-down menu; see Figure 11-32.

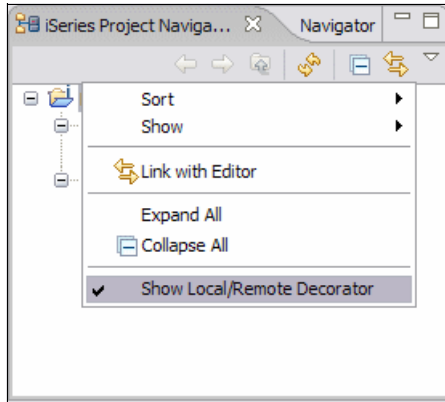


Figure 11-32 Show System i Projects Local and Remote Decorator

Following are descriptions of the System i Projects label decorations.

Table 11-1 Decorator descriptions

Decorator	Description
	Indicates that a project is a System i Project.
	Indicates the resource is a local object. If Show Remote Objects is enabled, the label decoration will change if there is a remote counterpart for the object on the host system.
	Indicates the resource is a remote object.
	Indicates the resource is both a local and remote object.
	Indicates that the object is missing a timestamp property, and appears only if the object is both local and remote. This is not applicable for objects that are remote only.
	Indicates a conflict. The object has been modified on the remote server.
	Indicates an error on the resource. Select the project and view the error information displayed in the taskbar. For example, a System i Project may need certain properties specified, or there may be invalid names for the source files or members.

11.8.2 Sequence numbers

A new preference is added to have sequence numbers and timestamp information automatically removed from members when they are downloaded to a System i project and added when they are uploaded back to the remote library. In addition, actions are added to add/remove this information from members within System i project.

Source control management (SCM) tools are used to share projects in a team environment and to manage source code. For more information about team support, refer to 11.9, “Team support and version control with System i Projects” on page 401. You can use System i Projects support to version source using certain SCMs, such as IBM Rational clearCase.

Removing sequence numbers

Some SCMs tools such as ClearCase do not have the concept of sequence numbers. If you change records in a member and resequence, when you compare the changes to what is in the SCM tool, it will show that all the records have been changed, although you didn't change

all the records, but the sequence numbers have changed. WDS 7.0 introduces new function for removing sequence numbers on download.

To enable the remove sequence numbers feature:

1. From the workbench menu select **Window** → **Preferences** → **System i Projects**.
2. Check **Removing sequence number and date fields on download**; see Figure 11-33.
3. Check **Generate sequence number warnings** to generate warnings for those members in System i Projects that do have sequence numbers, when the option to remove sequence numbers is checked.

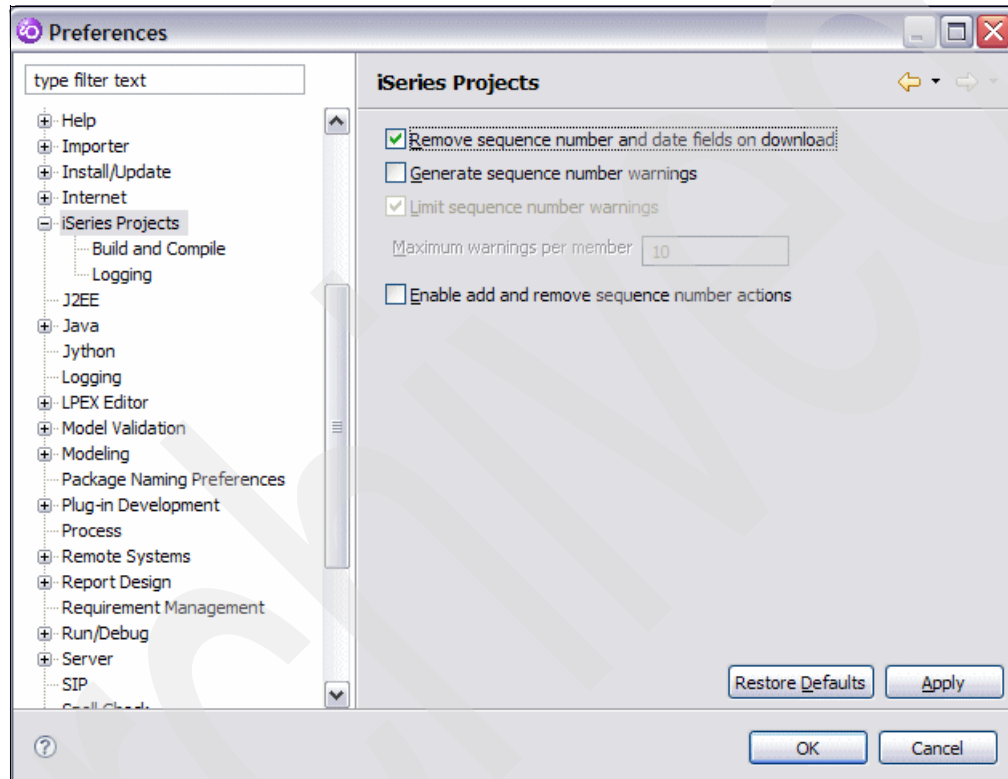


Figure 11-33 System i Projects preferences - removing sequence numbers

Additional functions in the System i Projects preferences page:

- ▶ Option to limit the number of warnings (only available if option to generate warnings is on).
- ▶ A field for specifying the maximum warnings per member (only available if the option to limit sequence number warnings is on). The default value is 10.
- ▶ Enable right-click actions to add and remove sequence numbers of members in System i Projects.

Members without sequence numbers

To remove sequence numbers from a member:

1. Select the option to remove sequence numbers and date area on download in the preference page, see “Removing sequence numbers” on page 391.
2. In RSE, add a member to the System i project. Right-click the member and select **Add to System i Project** or **Make Available Offline**.

The member should be added to the System i Project with the sequence number and date area stripped off.

Note: This option only applies to members being downloaded. If a System i Project already had members with sequence numbers they would not be affected in any way.

3. When the member is opened in the editor, sequence numbers are not shown. See Figure 11-34 for a member with sequence numbers, see Figure 11-35 for a member without sequence numbers.

```
Line 1      Column 1      Replace
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8
000100      *-----
000200      *
000300      * ROUTINE: ORDENTR
000400      *
000500      * This program allows a user to enter and confirm an order. It sends
000600      *   a print request to a batch job via a data queue. The program only
000700      *   handles District 1 and Warehouse '0001'.
000800      *
000900      * INDICATOR USAGE:
001000      *   03 - F3=Exit
001100      *   04 - F4=Prompt
001200      *   06 - F6=Accept order
001300      *   12 - F12=Cancel
001400      *   40 - SFLDSP
001500      *   41 - SFLDSPCTL
001600      *   44 - SFLCLR
001700      *   79 - OVERLAY on FKEY record format to control clearing display
001800      *   80 - Customer record not found
001900      *   81 - Part record not found
002000      *   82 - Stock record not found
002100      *   83 - Invalid quantity
002200      *   84 - Both part and quantity must be specified
002300      *   99 - General I/O errors
002400      *   LR - Deactivate program and return to caller
002500      *
002600      *-----
002700      FORDENTD  CF  E           WORKSTN SFILE(ORDSFL:Sf1Rrn)
002800      FITEM    IF  E           K DISK
002900      FDSTRCT  UF  E           K DISK
```

Figure 11-34 System i project member with sequence numbers

```

Line 1      Column 1      Replace
.....1....+...2....+...3....+...4....+...5....+...6....+...
*
*
* ROUTINE: ORDENTR
*
* This program allows a user to enter and confirm an order. It sends
* a print request to a batch job via a data queue. The program only
* handles District 1 and Warehouse '0001'.
*
*
* INDICATOR USAGE:
* 03 - F3=Exit
* 04 - F4=Prompt
* 06 - F6=Accept order
* 12 - F12=Cancel
* 40 - SFLDSP
* 41 - SFLDSPCTL
* 44 - SFLCLR
* 79 - OVERLAY on FKEY record format to control clearing display
* 80 - Customer record not found
* 81 - Part record not found
* 82 - Stock record not found
* 83 - Invalid quantity
* 84 - Both part and quantity must be specified
* 99 - General I/O errors
* LR - Deactivate program and return to caller
*
*-----
FORDENTD  CF  E          WORKSTN  SFILE(ORDSFL:Sf1Rrn)
FITEM    IF  E          K  DISK
FDSTRCT  UF  E          K  DISK

```

Figure 11-35 System i project member without sequence numbers

Warnings for sequence numbers

To enable warnings to be generated:

1. Select the option to generate number warnings in the preference page; see “Removing sequence numbers” on page 391.
2. If the option to remove sequence numbers is on, for a member in a System i Project that has sequence numbers warnings will be generated.
3. The warnings are shown in the Problems view.
4. You can double-click a warning in the Problems view and the associated line will be highlighted in the editor.

Note: The number of warnings per member can be changed in the preference page. The default value is 10 warnings.

Add and remove sequence numbers of a member

The following steps show how to add sequence numbers to a member:

1. Select the “Enable add and remove sequence number actions” option in the preference page; see Figure 11-36.

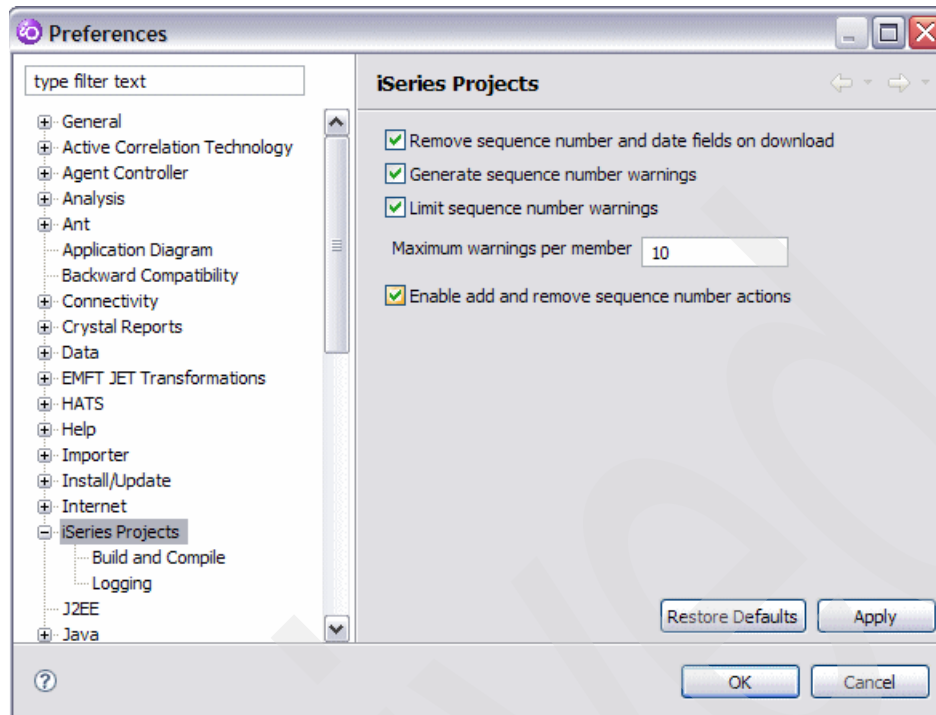


Figure 11-36 System i Projects - Enable add and remove sequence number actions

2. Right-click a member; you can see two new actions in the pop-up menu:
 - Add Sequence Numbers and Dates
 - Remove Sequence Numbers and Dates
3. Select **Add Sequence Numbers and Dates**, from the pop-up menu.
4. In the dialog, select the member for which you want to add sequence numbers and dates. There's a preview pane to show the contents of a selected member; see Figure 11-37.

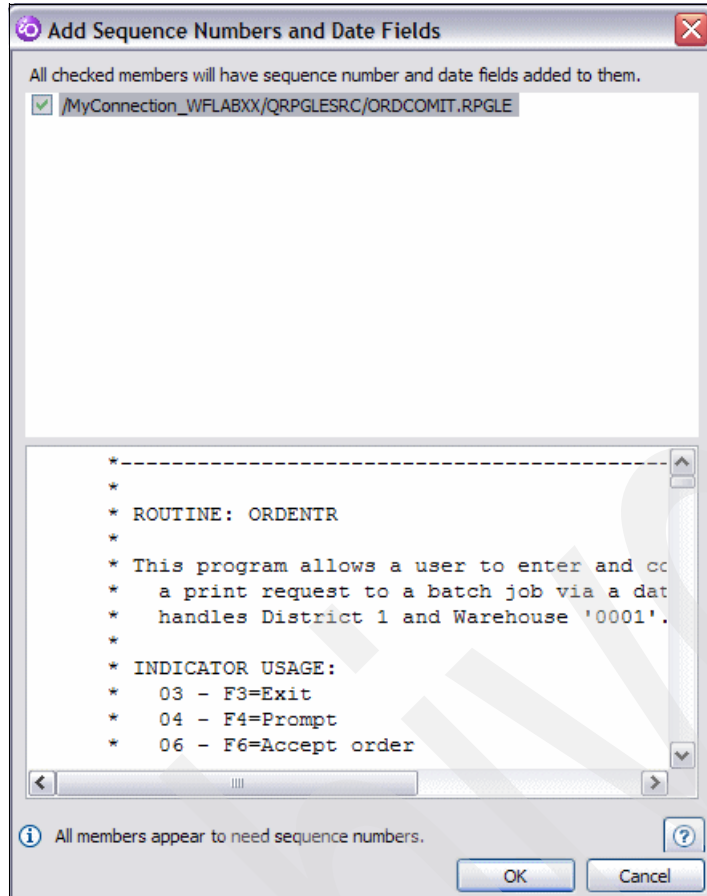


Figure 11-37 System i Projects - add sequence numbers and date dialog

If the member already has sequence numbers and date fields, then a warning will be shown in the dialog and the member will not be selected.

Note: This Add and remove sequence numbers of a member action is available for:

- ▶ Multiple member selections
- ▶ Source files - applies to all members in the source file
- ▶ Project - applies to all members in all source files in the project

To Remove Sequence Numbers and Dates, repeat the above steps and select **Remove Sequence Numbers and Dates** from the pop-up menu in step 3.

11.8.3 Save files support for System i projects (WDSC AE 7.0)

Advanced As of Version 7.0, this capability is only provided with WebSphere Development Studio Client Advanced Edition for iSeries (WDSC AE 7.0).

Save files are a way of creating back-ups of System i resources. You can package the contents of your source and objects in a save file.

Creating and restoring System i save files

You can store libraries and objects in save files by the following steps:

1. Create a save file with the CRTSAVF command or with a CRTSAVF command prompt; see Figure 11-38.

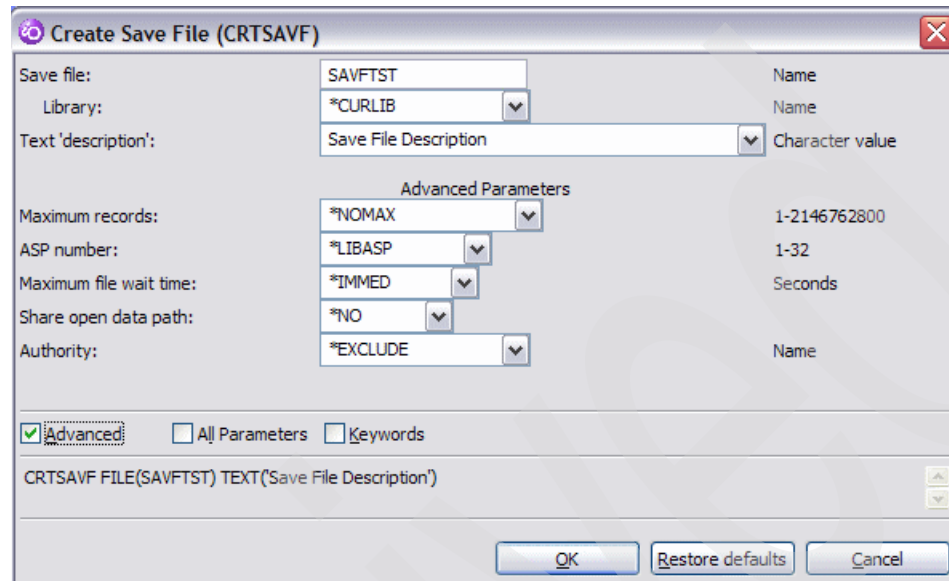


Figure 11-38 Create save file

2. You can store either objects or libraries in the save file by right-clicking the library or the object and selecting **Save**.

You can restore the contents of save files by the following steps:

1. Right-Click your save file.
2. Depending on your save file contents, select one of the following:
 - a. Restore the library to bring up the RSTLIB command prompt.
 - b. Restore Object to bring up the RSTOBJ command prompt.

You can use the **Restore on System i** action to upload and restore save files (in workspace navigator and local Connection filter in RSE); see Figure 11-39 on page 398.

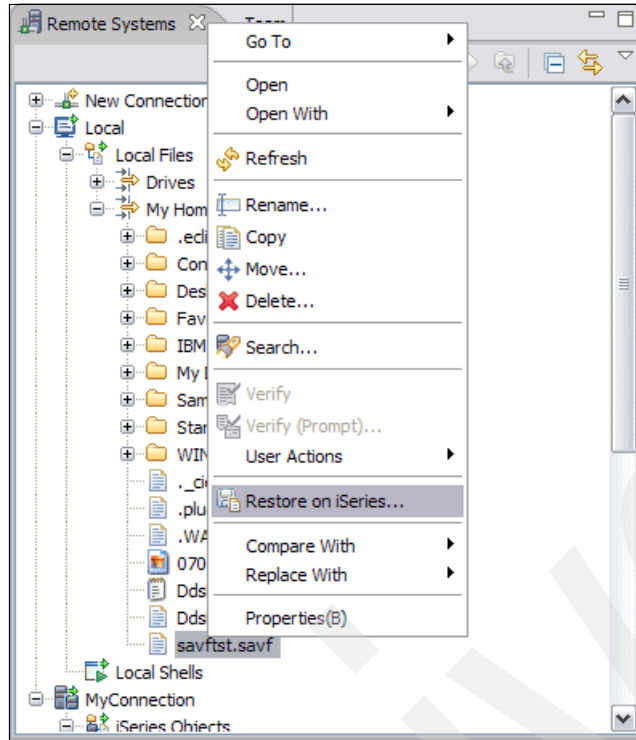


Figure 11-39 Restore save files on System i

System i save file support in WDCS 7.0

You can download save files to System i projects from RSE connections, and push save files from local System i projects to the host.

One of the benefits of adding save files in System i Projects is that you can package the contents of your source, binaries and other objects in save files, which can then be checked and versioned using a Source Control Management (SCM) tool, such as ClearCase or CVS. This provides an indirect way of allowing non-source resources to be versioned in an SCM that does not run on System i, such as ClearCase.

Downloading System i save files

To download a System i save file to your System i host, right-click the save file in the Remote Systems view, and select **Add to System i Project** or **Make Available offline**; see Figure 11-40 on page 399.

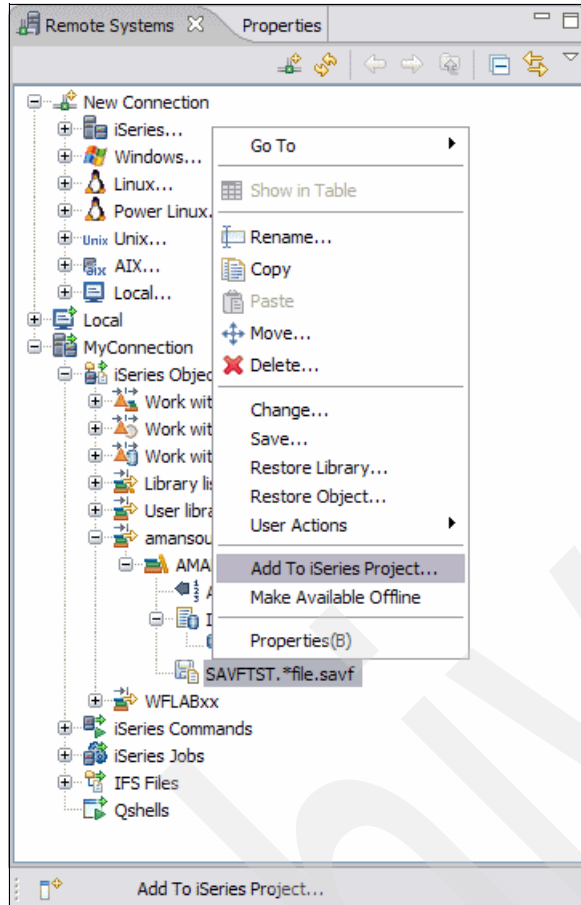


Figure 11-40 Add save file to System i project

Uploading System i save files

To upload a System i save file from your System i project to your System i host:

1. Right-click the save file in System i Project View; see Figure 11-41 on page 400.
2. Select **Remote Actions**.
3. You have two options to choose from. For more information about push operations, refer to 11.6.1, “Push operations” on page 378:
 - Push Changes
 - Push Selected

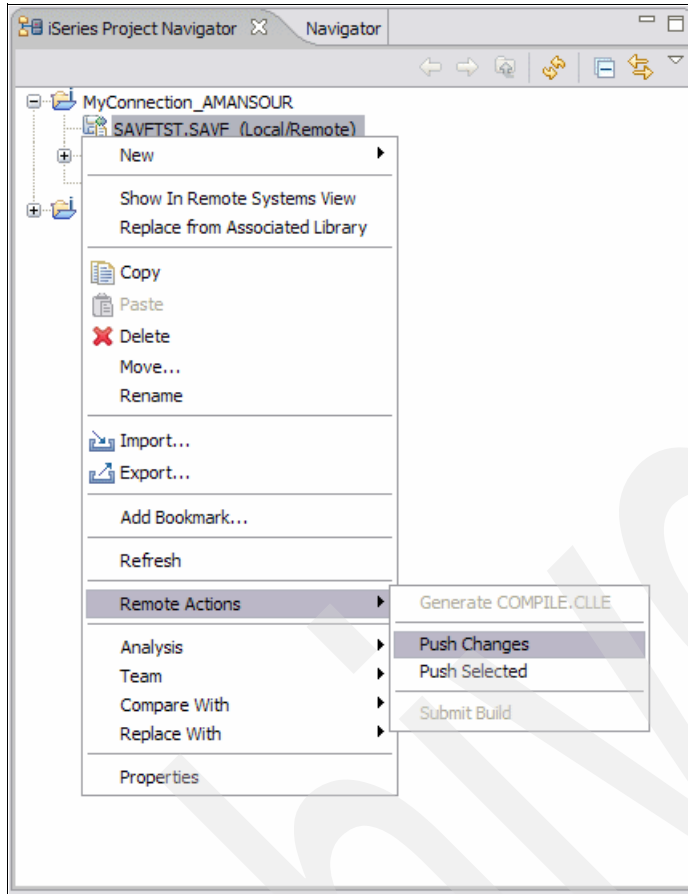


Figure 11-41 Upload save file from System i Project

System i save file properties page

You can use the System i save file properties page to edit the description of the save file. The properties are saved in a .SAVF_properties file under the System i project.

To change the description of the save file:

1. Go to the System i Project Navigator view.
2. Right-click the save file.
3. Select **Properties** from the pop-up menu. This opens the Properties dialog for the selected save file.
4. In the Properties dialog, select **System i Save File**, see Figure 11-42 on page 401.
5. In the Description field type your description.
6. Click **Apply**, then **OK**.

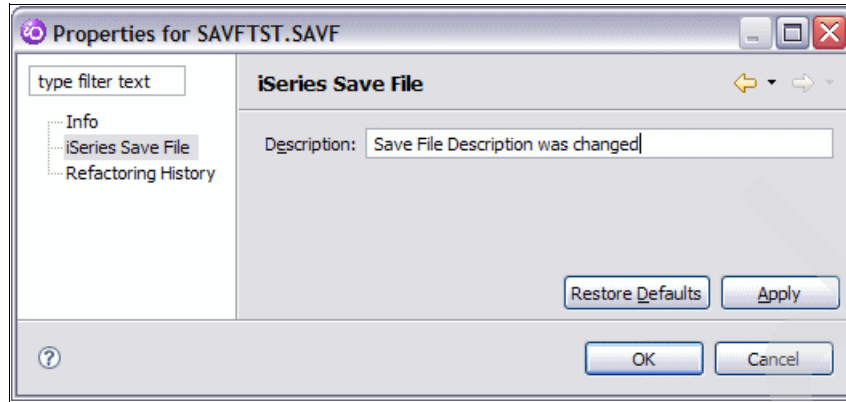


Figure 11-42 System i save file properties page

11.8.4 Push method selection during compile

In WDSC 7.0 you have the option to set the type of push operation performed, when compiling a member in System i project. The old behavior was to **Push all changed resources in the project**. This is set as the default action now, but you can change this option to **Push selected resources if there are any changes**. You can change this action from the System i Preferences page. For more information about compile actions, refer to 11.6.2, “Compile actions” on page 380.

11.9 Team support and version control with System i Projects

If you want to share your System i project with your team by specifying a common repository, right-click your project and select **Team** → **Share Project**. Enter your repository location information in the dialog box and click **Finish**.

11.9.1 Rational ClearCase

Rational ClearCase refers to both Rational ClearCase and Rational ClearCase LT. Rational ClearCase features two configuration management interfaces:

- ▶ Unified Change Management (UCM), an activity-based change management process included in Rational ClearCase.
- ▶ Base ClearCase, which is a set of tools that can be used to create a configuration management solution tailored to a particular development environment.

Benefits of using Rational ClearCase

Rational ClearCase can be used to manage and track all software assets, whether they are mission-critical software artifacts, documentation, Web pages, or source code. Like other source code control tools, Rational ClearCase can be used to:

- ▶ Create new versions of a software artifact
- ▶ Compare versions of a software artifact
- ▶ Merge changes from one version of an artifact into another version
- ▶ Control simultaneous changes to a software artifact
- ▶ Mark certain versions of artifacts as stable sources to be used in builds
- ▶ Determine the who, when, and why of a particular change to an artifact

Rational ClearCase also provides advantages to different development team members:

- ▶ Team leaders can coordinate the activities of people developing products together.
- ▶ Project managers have control over the extent and frequency with which team members synchronize their work.
- ▶ Developers can work in parallel, developing their artifacts in private areas so that the work of other team members is not affected by their changes.
- ▶ Integrators can combine the efforts of the team in a controlled manner.

Advanced Rational ClearCase LT

Rational ClearCase LT, part of the Rational ClearCase product family, is included with WDSC AE and provides reliable, entry-level version control for small and medium project workgroups.

Rational ClearCase LT supports parallel development with automatic branching and snapshot views, and enables developers to efficiently design, code, test and enhance software from a common code base. It supports sharing resources from multiple projects among team members in environments including:

- ▶ Local projects where assets for all projects are stored in the same repository
- ▶ Distributed projects where project assets are stored in different repositories and where those repositories may be on the same server or distributed between remote servers

11.9.2 Using Rational ClearCase for i5/OS application development

To enable WebSphere Development Studio Client to work with ClearCase LT, you must install the Rational ClearCase SCM Adapter feature. By default, this feature is selected when you install WebSphere Development Studio Client; however, if you did not include it, you can install it later using the Modify Packages wizard in IBM Installation Manager. For details, refer to Modifying installations.

For complete instructions on installing Rational ClearCase LT, refer to the installation documentation provided with the ClearCase LT installation media. It is also strongly recommended that you read the ClearCase LT release notes before you install the product.

Important: You must enable the Rational ClearCase SCM adapter before you can work with it. Refer to the online help for details on enabling and working with the adapter.

Work with ClearCase from System i Projects perspective:

1. Switch to the System i perspective **Window** → **Open Perspective** → **System i Projects**.
2. You can add the ClearCase commands menu to the System i Projects perspective:
 - a. Select **Window** → **Customize Perspective**.
 - b. Select the **Commands** tab.
 - c. Select **ClearCase** from the **Available command groups** list; Figure 11-43.

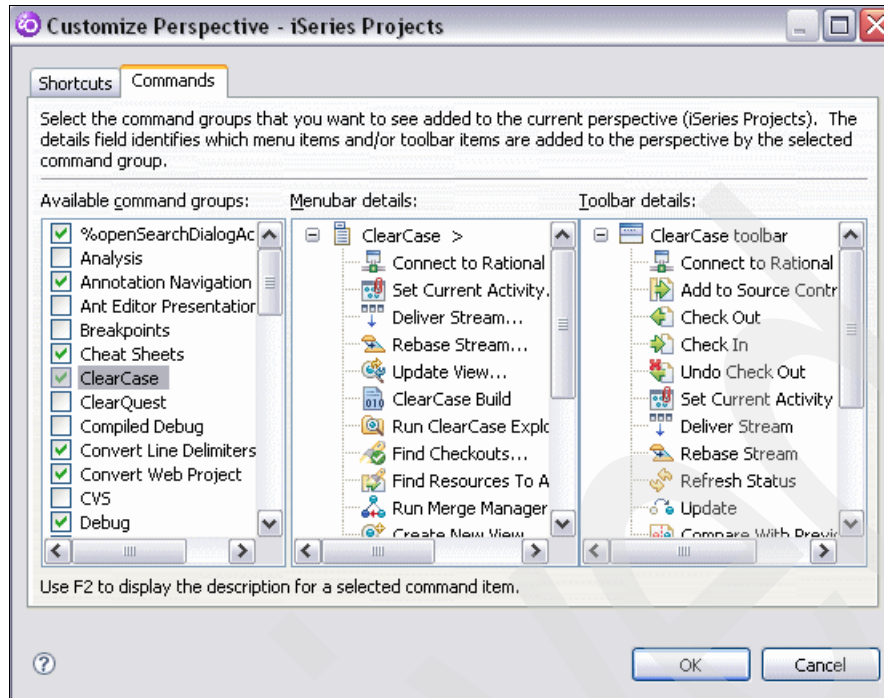


Figure 11-43 Add ClearCase commands to perspective

- From the workbench menu, select **ClearCase** → **Connect to Rational ClearCase**; Figure 11-44.



Figure 11-44 Connect to Rational ClearCase

- Select the project that you created in section 11.3, “Creating a System i project” on page 361.
- Right-click the project, and select **Team** → **Share Project** in the pop-up menu.
- In the Share Project dialog, select **ClearCase SCM Adapter** (Figure 11-45) and click **Next**.

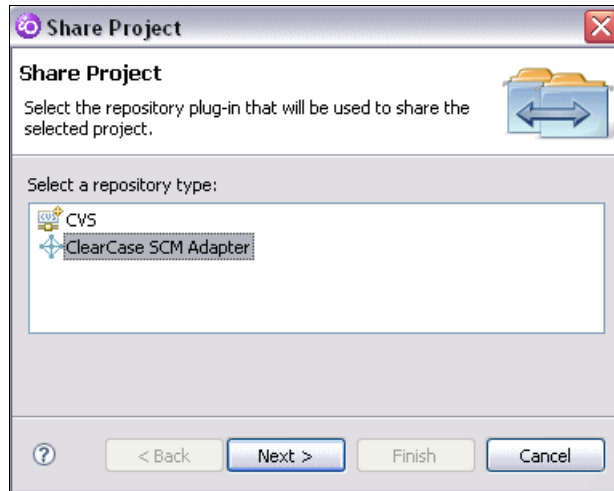


Figure 11-45 Select repository plug-in

7. Select a directory in ClearCase VOB for your project, and click **Finish**; Figure 11-46.

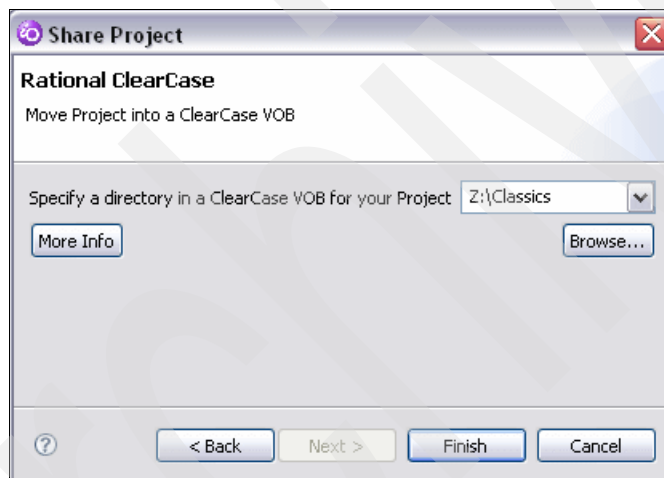


Figure 11-46 Select ClearCase VOB directory

8. In the Add Element(s) to Source Control window, select the elements that you want to add and click **OK**.
9. You will be prompted to select a ClearCase activity. You can either select an existing activity from the drop-down list, or create a new activity; see Figure 11-47.

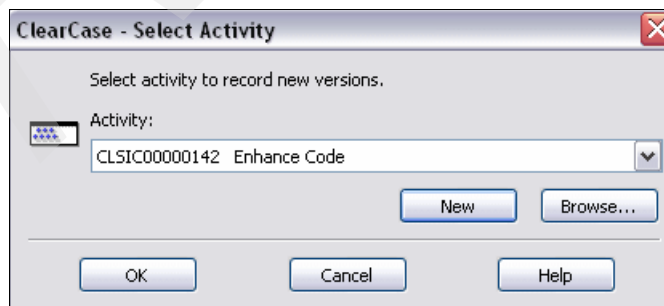


Figure 11-47 Select ClearCase activity

10. Click **OK**. The project will be added to ClearCase.

11. To check out a member in the project:

- Right-click the member, and select **Team** → **Check out**; see Figure 11-48.

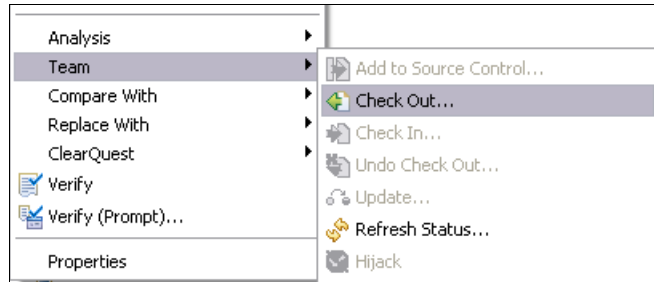


Figure 11-48 Check out a member

12. The icon of the checked-out member will have a green check mark decorator on it; see Figure 11-49.

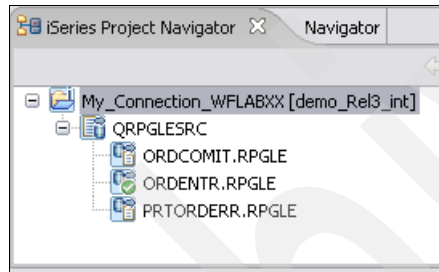


Figure 11-49 Checked out members

13. To check the member in, after making your changes: Right-click the member, and from the pop-up menu select **Team** → **Check In**.

14. You can perform additional ClearCase actions such as Undo Check out, Update, Hijack, and others. For a list of the available actions, right-click a member and select **Team**.

Archived

Integrated System i Debugger

This chapter discusses how to debug and test your applications using the various debugging options and profiling tools in WebSphere Development Studio Client for System i.

For more information about debugging, select **Help** → **Help Contents** from the main menu, then expand:

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Debugging System i Applications**
- ▶ → **Integrated System i Debugger**

See also **Help** → **Help Contents**, then expand:

- ▶ **WebSphere Development Studio Client for System i**
- ▶ → **Reference**
- ▶ → **Debuggers**
- ▶ → **Integrated System i Debugger**

There is also a tutorial which you can find by selecting **Help** → **Tutorials Gallery** in the main menu of the WDSC 7.0 workbench, and then expand.

- ▶ **Do and Learn**
- ▶ → **System i**
- ▶ → **Developing System i server applications using Remote System Explorer**
- ▶ and then one of the following:
 - → **Maintain an ILE RPG application using Remote System Explorer**
 - → **Maintain an ILE COBOL application using Remote System Explorer**

The ILE RPG and the ILE COBOL-based tutorials both contain one chapter called *Debugging a program*, providing a step-by-step example. Although a tutorial did exist in previous versions, it has been enhanced for WDSC 7.0 and now includes a sample library you can install onto your own System i.

To install the sample library for the tutorials, as well as for this chapter, follow the steps described under **Introduction** → **Prerequisites** → **Import the RSELABXX project into your workspace** in one of the two tutorials.

12.1 Available debuggers for i5/OS-based applications

IBM WebSphere Development Studio Client for System i offers the following debuggers:

- ▶ The Integrated System i debugger
- ▶ The Java debugger for System i
- ▶ The IBM Distributed Debugger

The Integrated System i debugger

The Integrated System i debugger helps you to find errors in code that is developed using OS/400 or i5/OS *Integrated Language Environment (ILE)* or *Original Program Model (OPM)* languages. This debugger enables you to debug applications that are running on the System i host while working in the workbench at your workstation.

The Integrated System i debugger supports any program written in the following languages:

- ▶ ILE RPG
- ▶ ILE COBOL
- ▶ ILE CL
- ▶ C
- ▶ C++
- ▶ RPG
- ▶ COBOL
- ▶ CL
- ▶ DB2 and SQL stored procedures

You can use this debugger, in most cases, during System i application development.

The Java debugger for System i

The Java debugger for System i enables you to detect and diagnose errors in Java applications that are running either locally on your workstation or remotely on System i.

The IBM Distributed Debugger

You can use the IBM Distributed Debugger for problem determination when you develop applications with the CoOperative Development Environment (CODE) tools, such as CODE Editor, CODE Designer, CODE Project Organizer, and CODE Program Generator.

The CODE tools are deprecated in WDCS V7.0 and WDCS AE V7.0. This means there will be no future enhancements made to these tools, they will not be updated to work with new releases of i5/OS, and they will not be ported to run on future versions of the Windows operating system. Customers using these tools should migrate to use the equivalent function in the Remote System Explorer and System i Integrated Debugger.

However, even if you decide to continue developing applications with CODE, you can still use the Integrated System i debugger to debug those applications.

For more planning information, see:

<http://ibm.com/software/awdtools/wdt400/news/planning700.html>

12.2 Prerequisites for debugging System i applications

Before you start debugging a System i application, you need to perform some or all of the following tasks:

1. Compile a program with debug data.
2. Change the debugger settings.
3. Ensure that your System i user profile has the appropriate authorities.
4. Start a debug session using one of the debugging methods:
 - a. Setting Service Entry Point (SEP) for a program and whenever the program is invoked the debug session starts
 - b. Attaching the debugger to a running application
 - c. One-step startup of the debug session
 - d. Start up a debug session with full prompt

For methods a.) and b.) the Integrated Debugger does not start any program for you; it rather attaches to an already running job.

12.2.1 Compile options

Programs must be compiled with appropriate debug information before compiled program objects can be used with the debugger. For example, ILE RPG, COBOL, and CL programs must be compiled with the *SOURCE, *LIST, *STMT, or *ALL option for the Debugging views (DBGVIEW) parameter. The default is *STMT (which, however, is not very useful for debugging) when compiling from the System i display. The default is *SOURCE when compiling in WDSC.

For more information about compiling each language, see your compiler documentation by selecting **Help** → **Help Contents**, then:

- WebSphere Development Studio Client for System i**
- **Reference**
- **Language Reference**
- **System i programming information**
- **High.level Languages**

12.2.2 Authority required to debug i5/OS applications

The user profile that you use to sign on to a System i system in the workbench System i logon dialog must have the following authorities:

- ▶ *USE authority to the Start Debug (STRDBG) command
- ▶ *USE authority to the End Debug (ENDDBG) command
- ▶ *USE authority to the Start Service Job (STRSRVJOB) command
- ▶ *USE authority to the End Service Job (ENDSRVJOB) command
- ▶ Either *CHANGE authority to the program being debugged, or *USE authority to the program being debugged and *SERVICE special authority.
- ▶ *JOBCTL special authority if you do not explicitly use fully qualified program names (library/program). In other words, if you use *CURLIB or *LIBL or you do not specify a library name, you need to have *JOBCTL special authority.

If the job that you are debugging is running under a different user profile than the user profile you use to sign on to the System i system from the debugger, the user profile that you use to sign on to the System i system from the debugger must have *USE authority to the user profile that the job you are debugging is running under.

The group profile QPGMR gives you the correct authority to the STRDBG, ENDDBG, STRSRVJOB, and ENDSRVJOB commands, and *JOBCTL special authority.

12.2.3 Debug considerations when writing programs

You can make your programs easier to debug by following these simple guidelines:

- ▶ Where possible, do not put multiple statements on a single line, because some debugger features operate on a line basis. For example, you cannot step over or set line breakpoints on more than one statement on the same line.
- ▶ Assign intermediate expression values to temporary variables to make it easier to verify intermediate results by monitoring the temporary variables.
- ▶ If you have a bug in a complex expression, you can debug the expression more easily by breaking it down into intermediate values and assigning those values to temporary variables. This allows you to verify each expression and variable.

12.2.4 Starting up the debug router

Before you can use the Integrated System i Debugger, you have to start the Debug Server. You need to do this only once for the entire i5/OS system. You can start the server one of two ways:

- ▶ In the Remote System view, right-click the System i Objects subsystem and select **Remote Servers** → **Debug** → **Start**.
- ▶ Use the Start Debug Server (STRDBGSVR) CL command.

To change the port that the Debug Server listens on, use the Work with Service Table Entry (WRKSRVTBLE) CL command and add or change the entry QDBGSVR2. If this entry is not found in the table, the debug server will listen on the default port 3825.

12.2.5 Verify the preference for updating production files

By default, whenever you place a job in debug mode, updating of production files is prevented by i5/OS. While some of the methods to start a debug session allow a parameter to be set to enable updating production files, others, such as using the service entry point, rely on a setting in your preferences.

To use a service entry point to start a debug session for your application and to allow updating of files in production libraries while debugging, the Update Production Files check box in the preferences of System i Debug must be checked, by selecting **Window** → **Preferences**, then expanding **Run/Debug** and selecting **System i Debug**.

See “Setting debug preferences” on page 448 for a more complete discussion of setting your preferences.

12.2.6 Setting source line breakpoints prior to debugging

Breakpoints are markers you place in your program to tell the debugger to stop your program whenever execution reaches that point. For example, if you suspect that a particular

statement in your program is causing problems, you could set a line breakpoint on the line containing the statement, then run your program.

The Execution stops at the breakpoint before the statement is executed. You can then check the contents of variables and view the call stack, and execute the statement to see how the problem arises.

The debugger supports the following types of breakpoints:

- ▶ *Line breakpoints* are triggered before the code at a particular line in a program is executed.
- ▶ *Watch breakpoints* are triggered when the contents of a specified storage location, for example a program variable, change.

You can set line breakpoints even before you start debugging by setting a source breakpoint in the Remote Systems LPEX editor. To do so, perform the following steps:

1. Open the source code in the Remote Systems LPEX editor.
2. Right-click the line number area on the left side of the editor and select **Add Breakpoint** as shown by the mouse pointer in Figure 12-1, or, much easier, double-click on the left of the prefix area on the left of any executable line as indicated by the mouse pointer in Figure 12-2. In other words, do *not* double-click *in the prefix area*. (Double-clicking a second time removes the breakpoint).




Figure 12-1 Adding a breakpoint to an ILE RPG source member

After the breakpoint is set, a new line breakpoint marker appears in the prefix area of the editor and in the Breakpoints view (in the Debug perspective). This marker appears as a filled dot as indicated by the mouse pointer in Figure 12-2.

005500	C	EVAL	EMESS = *BLANK
005600	C	EVAL	EMPAPL = *BLANK
005700	C	EVAL	PRJAPL = *BLANK
005800	C	EVAL	RSNAPL = *BLANK
005900	C*		

Figure 12-2 Breakpoint marker in an ILE RPG source member

Later, when a debug session for this program begins, the breakpoint appears in the Breakpoints view and in the editor with a check mark  at the lower left to indicate an

installed breakpoint. An installed breakpoint has already been set successfully in the program during a debug session.

When the debug session terminates, the breakpoint becomes a filled dot, like the one shown in Figure 12-2 again, indicating an *enabled breakpoint*.

The breakpoint will be removed only if you delete the marker (by double-clicking or selecting **Remove Breakpoint** from the context menu) or deleting the resource file in your WDSC workspace that contains the marker.

Note: A breakpoint in your source can only be used by yourself and only as long as you do not switch to a different workspace.

12.3 How to start a debug session

There are several methods to launch a debug session. You can do one of the following:

- ▶ Use *service entry points* to trigger the debugging session when the first line of a specified program, module, or procedure is executed in a job that is not yet under debug.
- ▶ Attaching the debug session to a running application by using *Debug as System i job*.
- ▶ Invoke a program object from the WDSC workbench and start debugging it.

For the latter two methods, you have two choices:

- ▶ To select *one-step debugging*, which involves selecting the program object or the job that you wish to debug,
- ▶ To *debug with full prompt*, which means that you use a launch configuration dialog to launch the debug session. You can use full-prompt debugging with one or more programs or service programs.

Note: When you use a service entry point, you also have the choice of supplying some necessary parameters or using defaults (as we describe in “Service entry point for ILE applications” on page 413), but it is not called *one-step debugging* and *debug with full prompt* because it uses different parameters and a different technique to define and manage them (instead of using the concept of a *debug configuration* described in “Debugging a running application with full prompt” on page 441).

The basic difference between the methods mentioned above is the fact when and how the debug session is actually invoked. In early versions of WDSC you could only use the last two methods by selecting a program object.

When you invoke the debug session using a service entry point or by attaching to a running application, the debugger will not start the application for you. You need to start it or the application is already running.

Over time the *service entry points* proved to be the most versatile of those methods, although there are situations when you would want to choose one of the other methods.

Tip: One of the reasons why you might not be able to use *service entry points* could be the fact that it only works for ILE programs. If you cannot convert all your applications to ILE (although highly recommended), there is a workaround at least in some case: Just create an ILE CL program which calls the program you want to debug. Then define a service entry point for the ILE CL program. When the debug session is invoked by the SEP in the ILE CL program, you can step into the Original Program Model (OPM) program it calls.

In this section we first describe the most flexible method using service entry points, followed by a discussion of the other techniques.

12.3.1 Service entry point for ILE applications

The service entry point (SEP) feature is designed to allow easy debugging of applications that use programs written in ILE RPG, ILE COBOL, or ILE CL, as well C or C++. It is a special kind of entry breakpoint that can be set directly from the Remote System Explorer before you even start the debugging session.

The debug session will be automatically started when the first line of a specified procedure, module, or program is executed. That way you can debug a program even without knowing the job name it is running in or having to start the program manually.

While the original service entry point design was meant to assist with debugging Web applications, using the feature is also beneficial when debugging applications in an environment where the job name is not known ahead of time. Some sample scenarios include:

- ▶ Debugging WebFacing applications
Typically you would want to debug your RPG or COBOL code that is driven by the JavaServer Page (JSP) that is executing in the application server. You do not know the name of the job in which your RPG or COBOL program will run, ahead of time, so the use of service entry points is an ideal solution for this scenario.
- ▶ Debugging any application that uses the Toolbox program or service program calls
When a program or service 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. With this technique, you can put the program or service program invoked by the Toolbox under debug although you do not know which job it will be running in.
- ▶ Debugging applications running in a multi-job environment
If your application runs in a multi-job environment, service entry points will significantly simplify your life while debugging. You would set a service entry point in the program that is about to be spawned in a new job. When the program is spawned and the line where the service entry point is set is about to execute, the debugger will be able to gain control of the program and stop at that line.
- ▶ Debugging SQL stored procedures
Debugging SQL procedures is very similar to debugging *PGM objects, because generated SQL procedures are *PGM objects. When debugging SQL procedures, you can have the Debugger editor display a source view after the source-level debug view is created.

Note that you can only use this capability with ILE programs running under OS/400 V5R2 or later.

12.3.2 Start debugging an application using a service entry point

There are two ways to set an SEP. In the Remote Systems view, you start off either with the program or service program object, or you begin with a source member.

To set a service entry point on a program object you can set a service entry point on the entire program or service program, or you can set it on one of the program's modules or procedures by performing these steps:

1. Expand the System i Objects node to the desired level (program, module, or procedure).
2. Right-click that item and choose **Debug(Service Entry) → Set Service Entry Point** from the pop-up menu.

For example, Figure 12-3 on page 414 shows service program SRVPGM2 with containing modules RETURN10 and RETURN20, each containing one procedure, return10() and return20(), respectively. The context menu opened for SRVPGM2 in that figure could well have been opened for one of the four items below it. However, in that case, the SEP would have been set only for one of the modules or procedures.

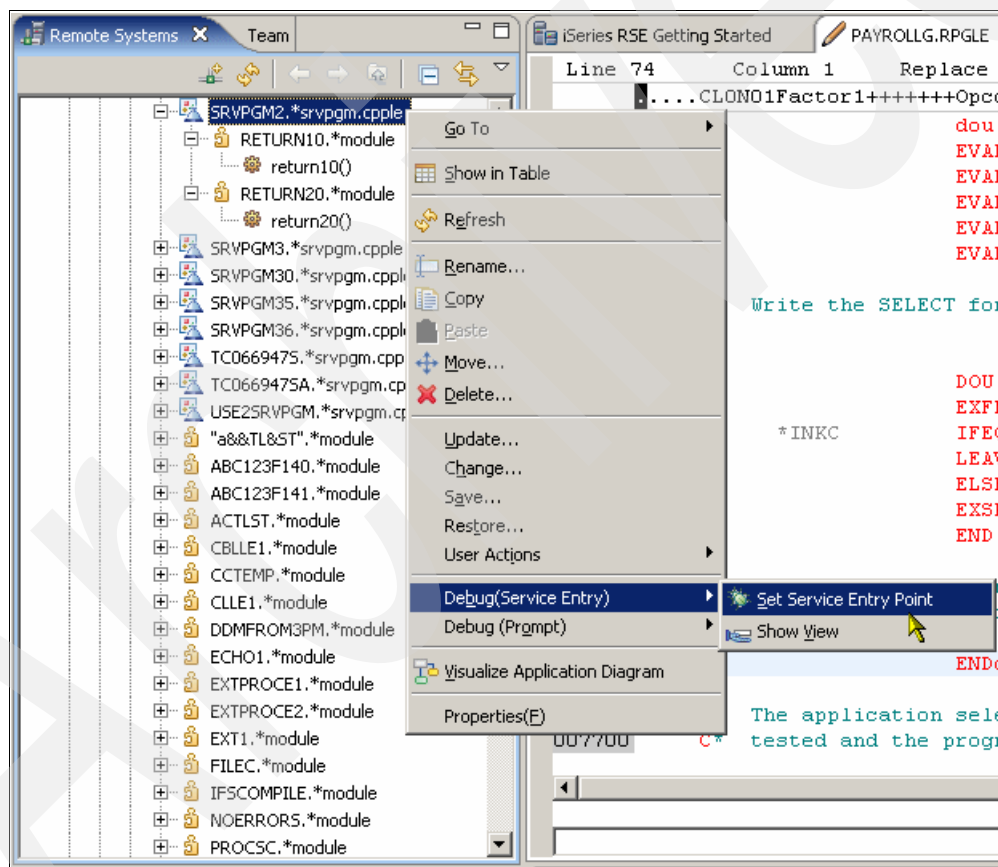


Figure 12-3 Set a service entry point (SEP) for a service program

To set a service entry point from a source member or from some specific file types in IFS (files with extensions .c, .cpp, .rpg, .rpgle, .cbl, .cbll), right-click the source file and select **Debug(Service Entry) → Set Service Entry Point** from the pop-up menu. A dialog opens allowing you to specify the Connection, Library, Program or Service Program, Module, Procedure, and User ID as shown in Figure 12-4 on page 415.

Note: You are only prompted for the parameters shown in Figure 12-4 when you set the SEP through a source member. If you set it through a program object (as in Figure 12-3), defaults are used for each of the parameters, which, however, can be changed later through the Service Entry Point view as we explain in “Using the System i Service Entry Points view” on page 417.

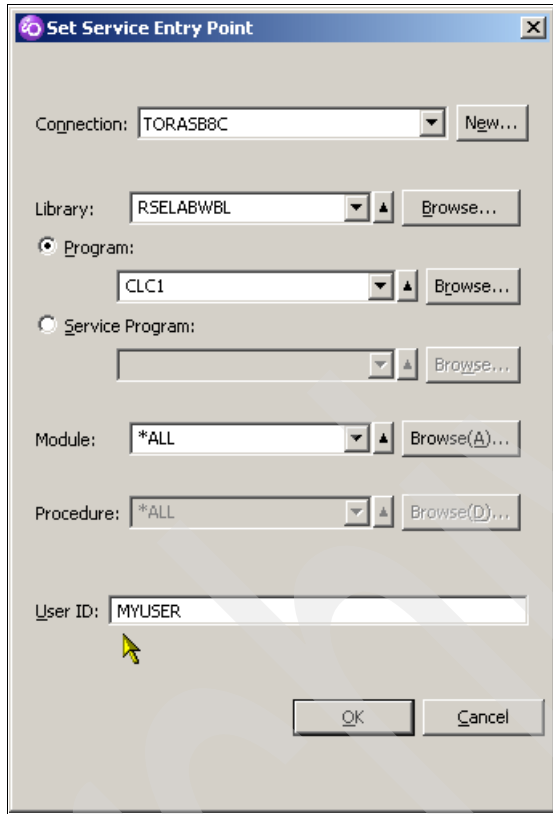


Figure 12-4 Set Service Entry Point dialog

Once you clicked **OK** in Figure 12-4, the SEP is set for the program, module, or procedure specified there. Whenever the SEP is encountered when the application is running *under the user profile specified in the User ID field*, a debug session is opened for your workbench.

Important:

- ▶ If the program starts under a user profile different from the one specified in the User ID field in Figure 12-4, the debug session will not start.
- ▶ If the user profile specified in the User ID field in Figure 12-4 is different from your own user profile (that is, the one you specified in your RSE connection), you need to have at least *USE authority for the other *USRPRF object.

If the debugging server has not been started before you get to this point, the SEP cannot be set and an error message is shown in Figure 12-5. You need to start the debugging sever as described in “Starting up the debug router” on page 410 and set the SEP again.

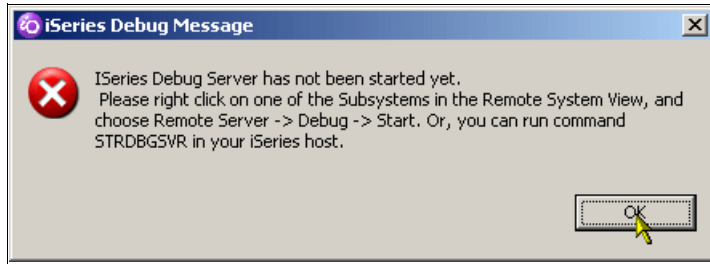


Figure 12-5 System i Debug Server has not been started yet

After you clicked **OK** in Figure 12-4 or selected **Debug(Service Entry) → Set Service Entry Point** from the pop-up menu in Figure 12-3 on page 414, a confirmation message as in Figure 12-6 is shown. You can avoid showing the message in the future by selecting the check mark for the *Don't show this message again* field.

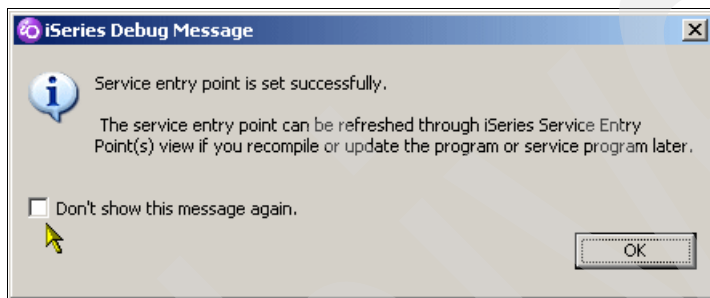


Figure 12-6 Service entry point is set successfully

From this point on till you either close your workbench, lose connection to your System i, remove, disable or invalidate the SEP, i5/OS monitors your program. Whenever it starts under the specified user profile and an SE is encountered, a debug session is started in your workbench. Depending on your preferences, you may see a warning like the one shown in Figure 12-7, before the session actually starts.

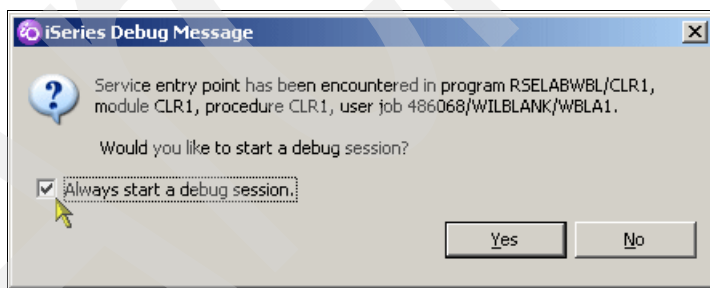


Figure 12-7 Service entry point has been encountered

Important: Whenever you recompile your program, all the service entry points set on that program become invalid. Therefore, you should refresh the SEP after each compile as described in “Refreshing service entry points in the System i Service Entry Points view” on page 418. Note that the System i Service Entry Point view shows all the SEPs still as enabled.

All Service Entry Points are removed, if the communication between your workbench and the i5/OS is interrupted.

If the service entry point was set successfully, the *System i Service Entry Points* view as shown in Figure 12-8 will automatically open, if it is not already open. This view allows you to manage your service entry points, as we explain in the following section.

Library	Program	Program Type	Module	Procedure	User ID	Connection	Enabled
BCOLLECTOR	PGM2	*PGM	*ALL	*ALL	WILBLANK	TORASBCC	Yes
BCOLLECTOR	SRVPGM2	*SRVPGM	RETURN10	*ALL	WILBLANK	TORASBCC	Yes
BCOLLECTOR	SRVPGM2	*SRVPGM	RETURN20	return20()	WILBLANK	TORASBCC	Yes
RSELABWBL	PAYROLLG	*PGM	*ALL	*ALL	YESSONG	TORASB8C	Yes
RSELABWBL	CLC1	*PGM	*ALL	*ALL	WILBLANK	TORASB8C	Yes
RSELABWBL	PAYROLL	*PGM	*ALL	*ALL	QPGMR	TORASB8C	Yes

Figure 12-8 System i Service Entry Points view

Using the System i Service Entry Points view

The System i Service Entry Points view, like the one shown in Figure 18-7, allows you to view, set, modify, remove, enable, disable, and refresh service entry points.

When you set a service entry point from the Remote Systems view, the System i Service Entry Points view automatically opens for you and the service entry point appears in it. If the System i Service Entry Points view is not open, you can do one of the following to open it:

- ▶ Choose **Window** → **Show View** → **Other** from the main menu bar. In the resulting dialog, choose **System i** → **System i Service Entry Points** and then click **OK**.
- ▶ Right-click a program, service program, module, or procedure in the Remote Systems view and select **Service Entry Point** → **Show View** from the context menu.

In the view, service entry points are listed in a table format with the following headings:

- Library:** Indicates the library that contains the program in which the service entry point is set.
- Program:** Indicates the program name in which the service entry point is set.
- Program Type:** Indicates whether the program is a program or service program.
- Module:** If the service entry point was set on a program module, the module name is listed here.
- Procedure:** If the service entry point was set on a procedure, the procedure is listed here.
- User ID:** Indicates the user profile whose job will trigger the service entry point.
- Connection:** Indicates the System i connection from which the service entry point was set.
- Enabled:** Indicates whether the service entry point is currently enabled or disabled.

You can sort the list of service entry points by clicking the table column heading that you want to sort by.

You may also move the mouse pointer over the border between two headings. When the shape of the pointer changes to two arrows with a vertical line (see Figure 12-8 on page 417 between Program Type and Module), you can double-click and the width of the column is changed to the maximum content width.


The list of service entry points in the view can be streamlined for easier viewing. You do this by filtering the list of service entry points. For further information about this, see the related topic in the online Help.

Refreshing service entry points in the System i Service Entry Points view

Refreshing a service entry point is useful if you set a service entry point in a program or service program and then recompile or update the program or service program.

After the program is recompiled, all the service entry points set on that program are now invalid. Then, when you run the program, a debug session will not start. To avoid this, refresh at least one of the service entry points set on this program. This verifies that all the original service entry points set on this program are still valid. Any invalid service entry points are removed, and those that are still valid are refreshed.

You can refresh one or more service entry points from the System i Service Entry Points view by performing the following steps:

1. In the System i Service Entry Points view, select the service entry points that you want to refresh.
2. Do one of the following:
 - Right-click your selection and click **Refresh** from the context menu.
 - In the toolbar, click the Refresh service entry points icon .

12.3.3 How to use the power of a debugging session

Once the debugging session has been started, either automatically through a service entry point as described above in “Service entry point for ILE applications” on page 413, or manually as described later in this chapter, the Debug perspective opens, providing you with many capabilities to analyze why your application is not showing the results you expect.

The following sections describe all those capabilities and tools, beginning with an overview of the Debug perspective. Later, in “Attach to a running application by debugging a System i job” on page 440, “One-step debugging a running application” on page 441, “Debugging a running application with full prompt” on page 441, we describe other methods to initiate the debug session.

Note, we use the same example in library RSELABxx as the tutorial mentioned in the introduction to this chapter on page 407 does. However, the following sections are not meant to be a step-by-step tutorial.

Using the Debug perspective

The Debug perspective as shown in Figure 12-9 on page 419 contains several views that allow you to manage your debug session; view information about your programs, modules, procedures or functions; monitor variables, work with breakpoints; and view your source.

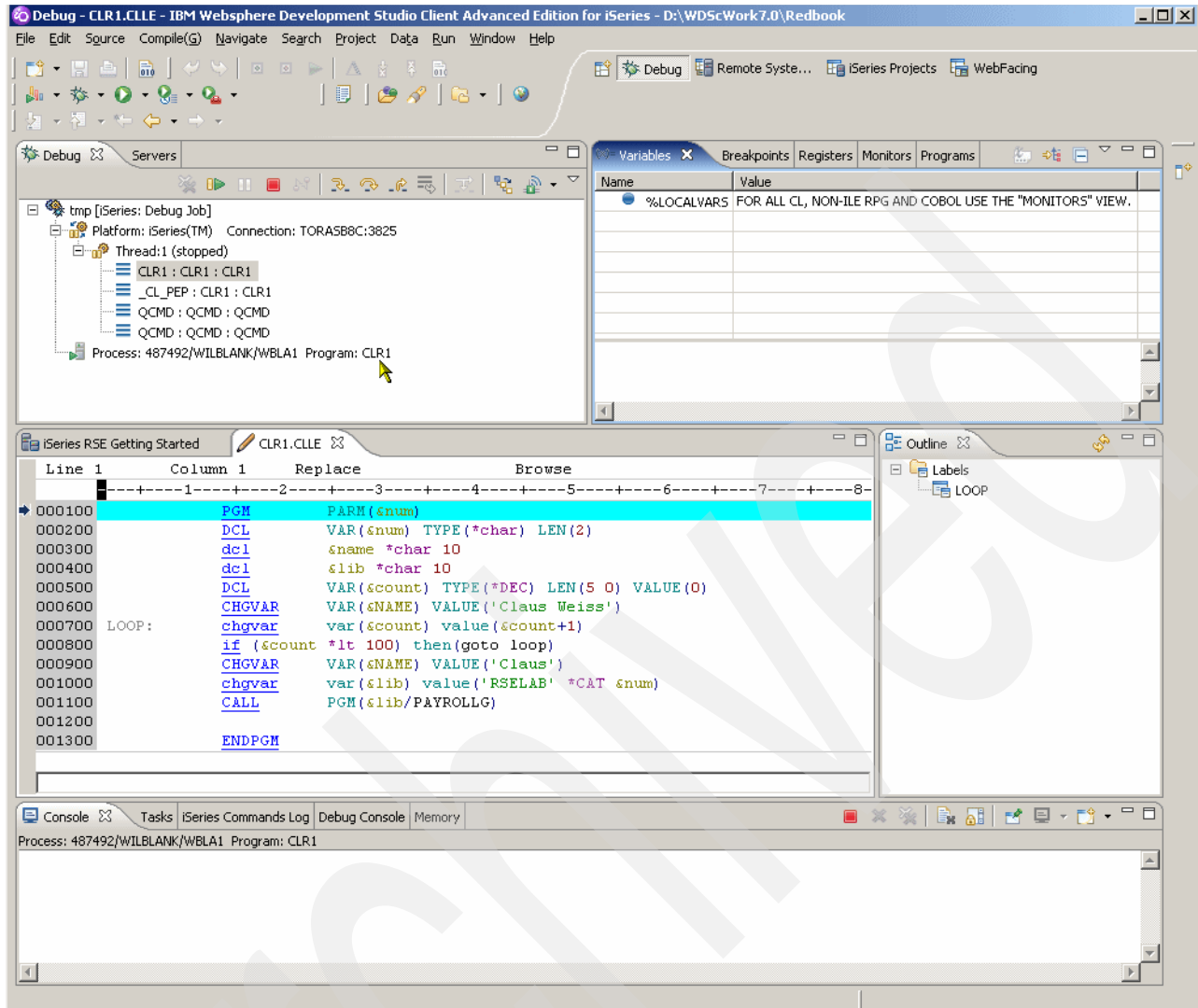


Figure 12-9 Debug perspective showing CL program CLR1 of the tutorial

Views to use when debugging System i applications

When you are debugging a System i application, you will want to make use of the following views:

- ▶ *Debug view*: This view shows by default in the upper left pane of the Debug perspective (mouse pointer in Figure 12-9) and allows you to manage the debugging of a program in the workbench. It displays the call stack entry for all the threads for each target you are debugging.
- ▶ *Breakpoints view*: This view in the upper right (covered by the Variables view in Figure 12-9) lists all the breakpoints you have set in the workbench projects. You can double-click a breakpoint to display its location in the editor. In this view, you can also enable or disable breakpoints, delete them, or add new ones.
- ▶ *Variables view*: This view in the upper right of Figure 12-9 displays information about the local variables in the currently-selected call stack entry. We describe this in more detail in “Variables view” on page 432.

- ▶ *Programs/Modules view*: This view in the upper right (covered by the Variables view in Figure 12-9) displays and allows you to navigate programs, service programs, and ILE modules.
- ▶ *Monitors view*: The Monitors view in the upper right (covered by the Variables view in Figure 12-9) shows variables and expressions that you have selected to monitor. You can select the variables or expressions from the editor to monitor, or you can add them to monitor via a dialog box in the Monitors view. Use the Monitors view to monitor global variables or variables you want to see at all times during your debug session. From the Monitors view, you can also modify the content of variables or change the representation of values.
- ▶ *Memory view*: The Memory view by default in the bottom pane of the Debug perspective, as in Figure 12-9 on page 419, allows you to look at and change the contents of storage areas used by your program. You can also modify the address range to view and change the contents of storage. The content can only be changed for programs using the *TERASPACE storage model.
- ▶ *Memory Rendering view*: The Memory Rendering view, also by default in the upper right of the Debug perspective (but not visible in Figure 12-9 on page 419), allows you to monitor memory for variables in data formats such as ASCII, EBCDIC, signed integer, and unsigned integer. The Memory Rendering view is synchronized with the Memory view.
- ▶ *Debugger editor view*: The pane in the middle left of Figure 12-9 contains all editor pages that were open when you started the debug session. In addition, you see the source of the program you are currently debugging. We describe this in more detail in “Debugger editor view” on page 421.
- ▶ The *Outline view* shows on the right of the Source view and has the same function as using it with the Remote Systems LPEX editor. The outline view is empty when you choose viewing either *LISTING or *STATEMENT.

We explain some of the views in the sections below.

The Debug view

The Debug view, by default in the upper left pane of the Debug perspective indicated by the mouse pointer in Figure 12-9, allows you to manage the debugging of a program in the workbench and view information for the active programs, modules, procedures, or functions on a thread's call stack.

The Debug view displays the call stack entry for all the threads for each target you are debugging. Each thread in your program appears as a node in the tree. The thread at the current execution point automatically expands. If the thread is stopped, its node is expanded to show its call stack entry.

If the resource containing a selected thread is not open and/or active, the file opens in the editor and becomes active, focusing on the source with which the thread is associated and highlighting the source line that the program is about to execute.

In the Debug view, you can perform various resume, step, and terminate debug actions.

To view stack information, click the Debug view to bring it in focus, and expand the appropriate thread entry. Right-click the call stack entry that you want to view and select **Properties** from the context menu, as shown in Figure 12-10.

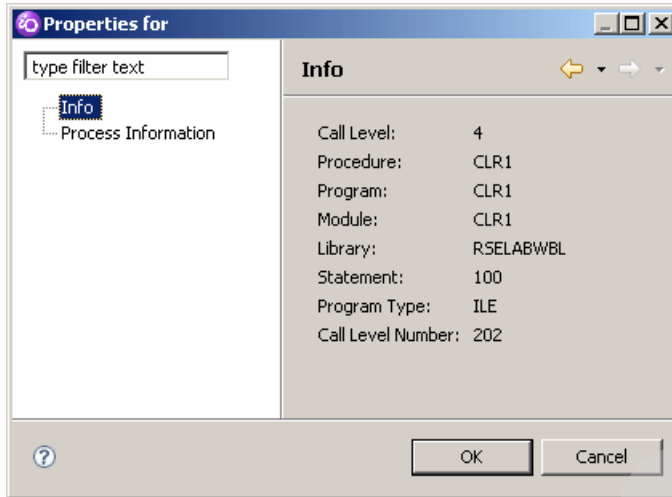


Figure 12-10 Call stack information for CLR1

Alternatively, you can have detailed properties of stacks displayed in the Properties view, as follows:

1. Open the Properties view by choosing **Window** → **Show View** → **Other** and then selecting the Properties view from the set of General views.
2. In the Debug view, go to the call stack entry whose properties you wish to view. If necessary, expand the thread nodes and use the scroll bars, Up and Down keys, or PageUp and PageDown keys to scroll the Debug view until the call stack entry is visible.
3. Select the call stack entry to have its properties appear in the Properties view.

Nodes in the Debug view are visible if they are at the top level of the tree structure, or if the nodes a level higher have been expanded. For all nodes that are visible in this view, you can jump to a specific node by typing the first letter of the node. The view will jump to the next node that begins with that letter. If this node is not the node that you wanted to jump to, type the first letter again to jump to the next node that begins with the letter, and so on.

Debugger editor view

The pane in the middle left of Figure 12-9 on page 419 contains all editor pages that were open when you started the debug session. In addition, you see some of the programs you are currently debugging. If it was compiled with the appropriate debugging information, you have up to three choices for viewing it here:

- | | |
|------------------------|---|
| Source (*SOURCE) | To see (and possibly edit) the source code for your program as shown in Figure 12-9 on page 419. |
| Listing (*LISTING) | To see a compiler-generated listing of your program as shown in Figure 12-11 on page 422. |
| Statement (*STATEMENT) | To see the statement number for each executable statement, as well as the name of the procedure to which the statement belongs; see Figure 12-12 on page 422. |

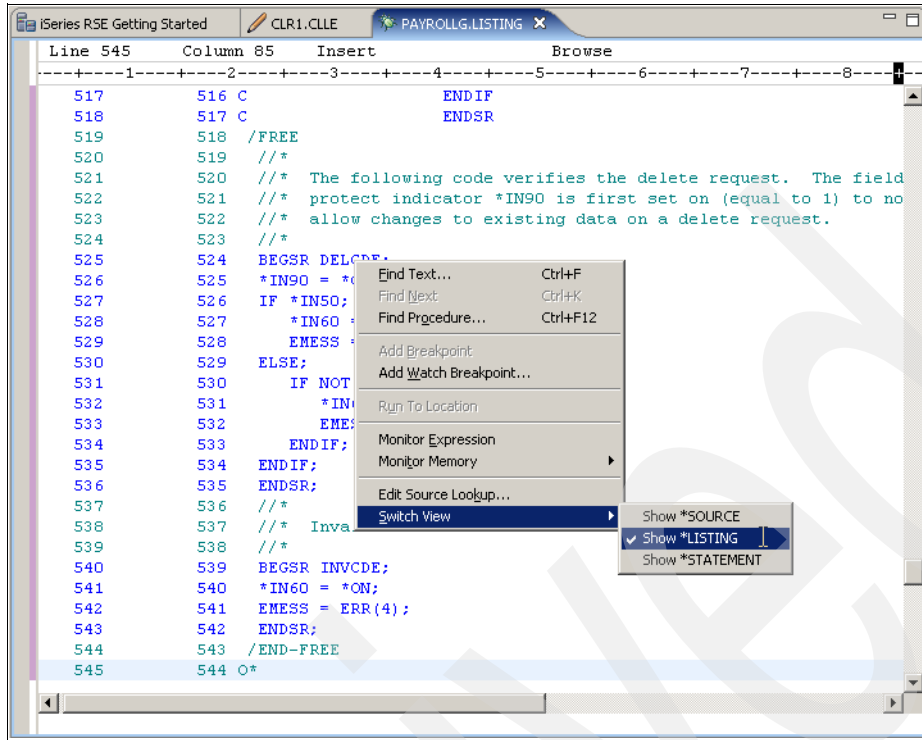


Figure 12-11 Debugger editor showing the Listing view

Unlike the source view, the Listing view does not populate the Outline view. It also shows only two colors: by default green for non-executable (such as comments) and blue for executable source statements.

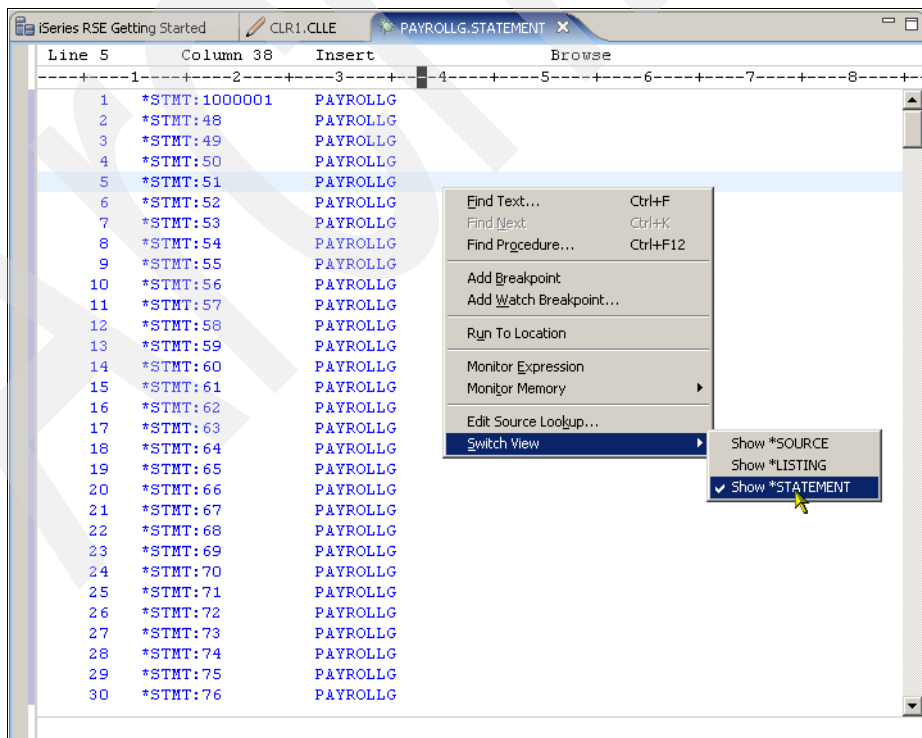


Figure 12-12 Debugger editor showing the Statement view

Right click anywhere in the editor view and select one of the following:

- ▶ **Switch View** → **Show *LISTING** to switch to the Listing view
- ▶ **Switch View** → **Show *STATEMENT** to switch to the Statement view
- ▶ **Switch View** → **Show *SOURCE** to switch to the Source view

Source file not found

If you compile the program with a compile option for Source debug view (*SOURCE) and then move either the source file or the program object to a different library (or even a different System i), the source file cannot be located by the debugger. In this case, an empty editor with three buttons is displayed, as shown in Figure 12-13.

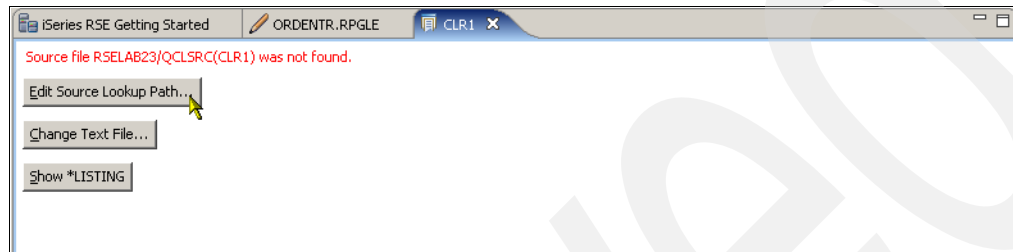


Figure 12-13 Source file not found

Since the debugger could not open the source, you have three possibilities to solve this problem:

- ▶ Select **Edit Source Lookup Path** to set up the System i Source File or Remote Folder source containers for your debug session.
- ▶ Select **Show *LISTING** to show the Listing view (as in Figure 12-11 on page 422) instead of the Source view.
- ▶ Select **Change Text File** to browse your local file system for the source file.

In most cases, you may want to try to edit the Source Lookup Path first. Only in cases where you do not have access to the original source, you might want to use the listing, if the program was compiled with `DBGVIEW(*LIST)`. Simply click **Show *LISTING** to open the *LISTING view. Note, however, that the *LISTING view has fewer capabilities than the *Source view.

For the rare situation where the source code for your program is stored in a text file on your PC, you can use the **Change Text File** button to access it instead of an i5/OS source member.

Edit Source Lookup Path

If you click **Edit Source Lookup Path**, the Edit Source Lookup Path dialog shown in Figure 12-14 opens.

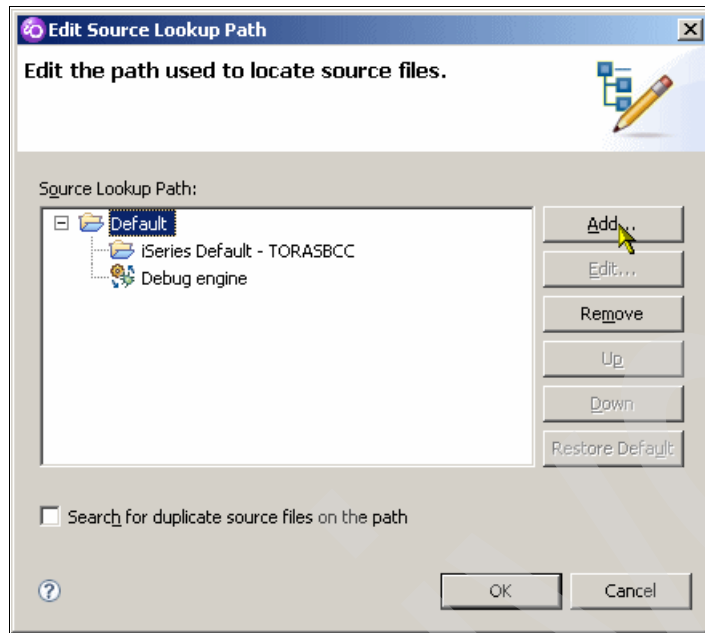


Figure 12-14 Edit Source Lookup Path dialog

Whenever a debuggable program is started while you are in a debug session, the debugger searches for a corresponding source in order to display it in the Source view. To do so, the debugger uses *source lookup paths* and, by default, it looks for the source in the location from which the program object is compiled.

If the source cannot be found, the Edit Source Lookup Path dialog shown in Figure 12-13 allows you to modify the source lookup paths. To do so, click **Add** and the Add source dialog as shown in Figure 12-15 appears.

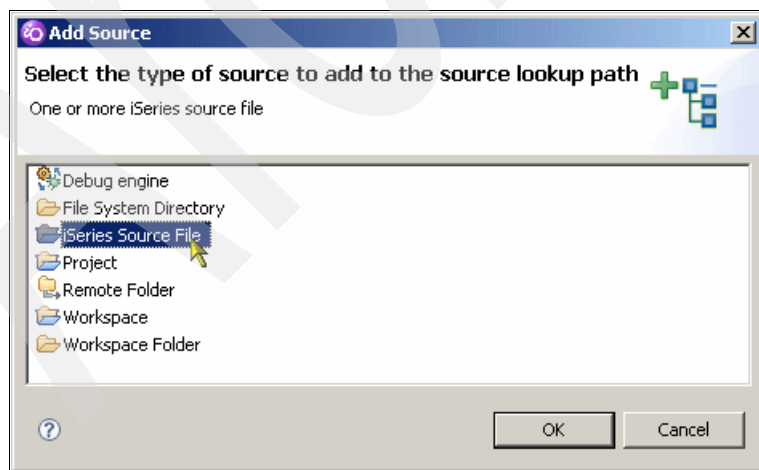


Figure 12-15 Add source to the source lookup path

In the Add Source dialog, do one of the following:

- ▶ Select **System i Source File** if your source resides in source physical files and click **OK**.
- ▶ Select **Remote Folder** if your source resides in the IFS and click **OK**. In the “Browse for a folder” dialog, navigate to the remote folder containing your source.

If you used the first (and most common) option, the *Select one or more source physical files* dialog, as shown in Figure 12-16, appears.

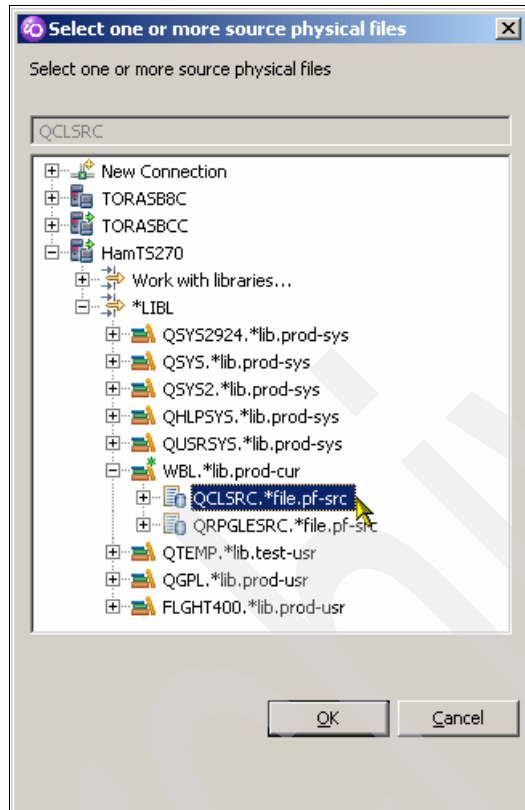


Figure 12-16 *Select one or more source physical files*

You can now navigate to your source physical file. Select one and click **OK**.

The Source file can be on a different i5/OS system!

Tip: As shown in our example in Figure 12-16, the source file may even reside on an i5/OS system different from the one where the program object is debugged, as long as your workstation has access to that system.

After you selected your source file in Figure 12-16 and clicked **OK**, you are returned to the Edit Source Lookup Path dialog, now showing your new source file location on top of the list.

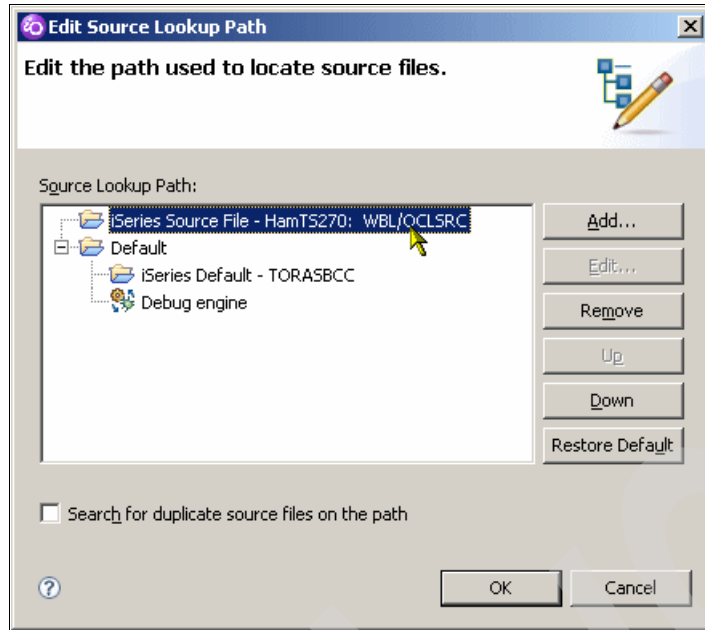


Figure 12-17 New source file in source lookup path

If the *Search for duplicate source files on the path* check box in the lower left corner of Figure 12-17 is not selected, the debugger searches the specified paths in sequence and uses the first occurrence. If this check box is selected, the debugger searches the specified paths and lists all occurrences, allowing you to select one. This is useful if the possibility exists that there are multiple source members with this name.

After you clicked **OK** in Figure 12-17, the source for the debugged program appears in the Source view, provided a member with the same name as your program exists in the source file or IFS directory you selected.

Breakpoints


As described earlier in “Setting source line breakpoints prior to debugging” on page 410, there are several types of breakpoints you can place in your program to tell the debugger to stop your program whenever execution reaches that point.




You can set the frequency and conditions on line breakpoints and the frequency of watch breakpoints. The term *frequency* here means that you can specify how many times the code must be executed before it stops at the breakpoint (we describe this in step 4 on page 430). When you run a program with a line breakpoint, execution stops at the breakpoint before the statement is executed, if the breakpoint condition is met.

When you start debugging a program for the first time, no breakpoints are set unless you created them in the editor prior to debugging (also described on page 410). During the debug session, you may set breakpoints. When you exit the debugger, breakpoints are saved. The next time you start debugging the same program in the same WDSC workspace, all line and watch breakpoints will be restored. However, watch breakpoints will restore as disabled.

Adding a line breakpoint

By setting line breakpoints, you can stop program execution at the line number you select. You can only set breakpoints on lines containing executable statements. If there is more than one executable statement on a line, the program will only stop once before executing the first statement.

The easiest way to add a line breakpoint is to do so from the Source view of the Debug perspective, regardless of whether you currently see the *SOURCE, *LISTING or *STATEMENT view there. You can do so by double-clicking directly to the left of the prefix area of that line where you want to add the breakpoint and an icon  shows that a breakpoint was installed here.

Note: As we explain earlier in Figure 12-2, the breakpoint appears in the Breakpoints view and in the editor with a check mark  at the lower left to indicate an *installed breakpoint*. When the debug session terminates, the breakpoint becomes a filled dot  again. Also, if you see any source member in the Source view of your Debug perspective showing a filled dot , it is *not* installed. The reason is most likely because the source member cannot be correlated to any program object currently being debugged.

When you double-click the icon representing the installed breakpoint at the left of the prefix area again, it is removed.

There are several alternatives to add a breakpoint:

- ▶ As we describe in 12.2.6, “Setting source line breakpoints prior to debugging” on page 410, you can set a line breakpoint before starting the debug session in the Remote Systems LPEX editor.
- ▶ You may also right-click anywhere on that line and select **Add Breakpoint** from the context menu.
- ▶ The *Breakpoints view* of the Debug perspective also allows setting breakpoints. This method is described in the online help under **WebSphere Development Studio Client for System i** → **Debugging System i applications** → **Controlling program execution and working with breakpoints** → **Setting a line breakpoint**.

Adding a conditional breakpoint

Technically speaking, any line breakpoint is also a conditional breakpoint. However, using the method described in the previous section, a condition cannot be specified immediately, but you may edit the breakpoint at any time to add a condition or specify a frequency:

1. Click the desired source statement within the Source view to highlight the line.

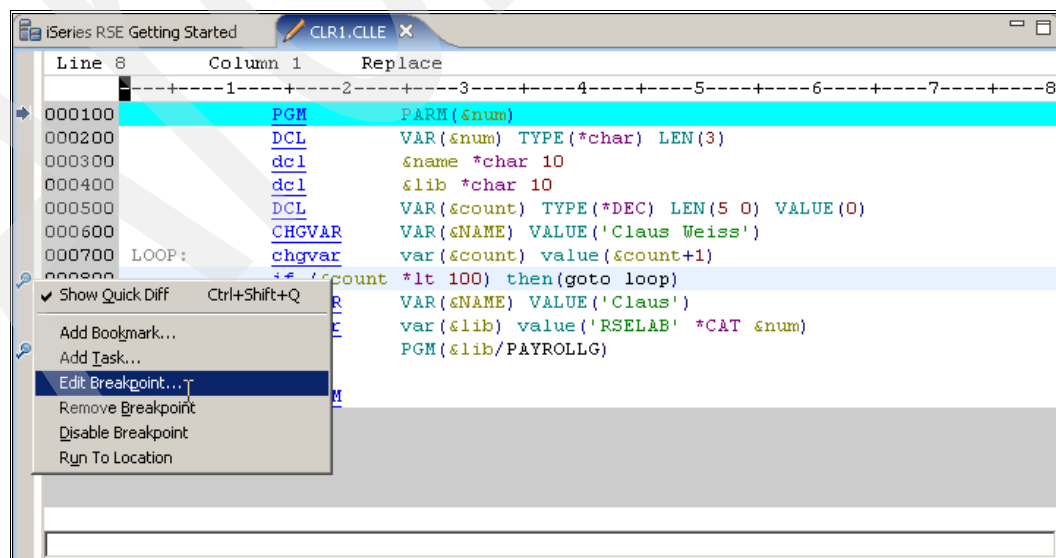


Figure 12-18 Edit a Line Breakpoint

2. In the upper right pane of the debug perspective click the **Breakpoints** tab. You see a list of all currently defined breakpoints.
3. Right-click anywhere within the Breakpoints view.
4. Click **Edit Breakpoint** → **Line** in the context menu. The Edit a Line Breakpoint window opens, as shown in Figure 12-19. The content of the fields in this dialog are set depending on which source line you selected.

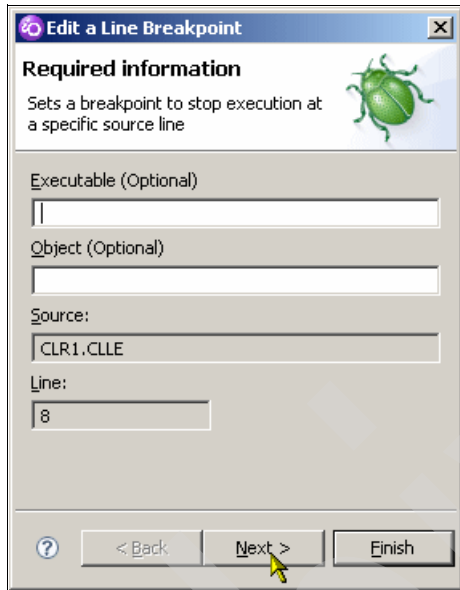


Figure 12-19 Edit a Line Breakpoint - required information

5. Click **Next** to see the Optional Parameters part of the dialog, as shown in Figure 12-20.

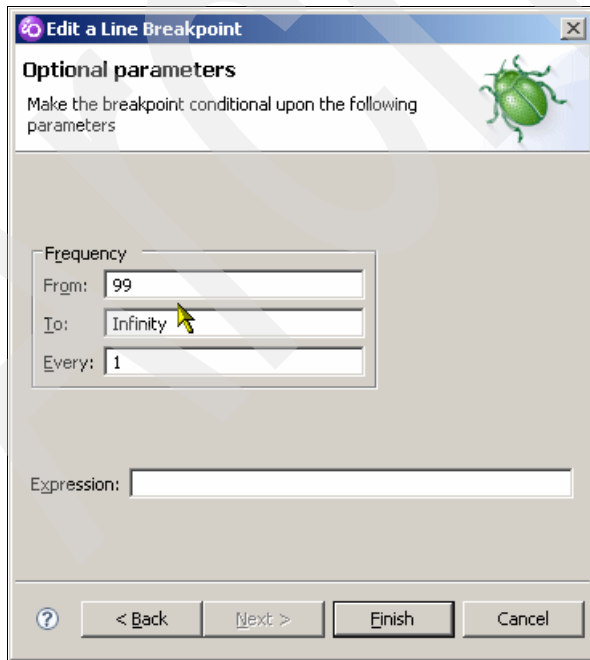


Figure 12-20 Edit a Line Breakpoint - optional parameters

The Optional Parameters part of the dialog shown in Figure 12-20 allows you to specify a frequency, that is, how often the breakpoint should stop program execution, as well as a logical expression which must be met for the breakpoint to be triggered.

The debugger keeps track of how many times each breakpoint is encountered. The fields in this section tell the debugger on which encounter of a breakpoint the debugger should first stop, how often it should stop, and on which encounter the debugger should no longer stop. The following parameters are used to set the breakpoint frequency:

- ▶ **From:** Enter the first breakpoint encounter you want the debugger to stop on. For example, if you want the debugger to skip over the breakpoint the first five times it is encountered, enter “6”.
- ▶ **To:** Enter the last breakpoint encounter you want the debugger to stop on. For example, if you want it to start ignoring the breakpoint after the 20th encounter, enter “20”. To have it always stop on the breakpoint, enter “Infinity”.
- ▶ **Every:** Enter the frequency with which you want the debugger to stop on this breakpoint. For example, if you want it to stop on only one out of every four it encounters, enter “4”.

In our example, we specified to ignore the breakpoint 99 times and after that suspend the program every time when line 8 of program CLR1 is executed.

Adding a watch breakpoint

A *watch breakpoint* is a type of breakpoint that is not tied to a specific line in your program, It is rather triggered whenever execution changes data at a specific address. You use this type of breakpoint when you want to see where and how a variable is being changed in your program.

Note: Unlike a line breakpoint, which can be defined any time in your source, the debug session must be already started in order to set a watch breakpoint.

You can set a watch breakpoint directly from the LPEX editor in the Source view by performing the following steps:

1. In the Editor go to a line containing the variable name you would like to watch.
2. Double-click the variable (*IN60 in our example in Figure 12-21) to highlight it.

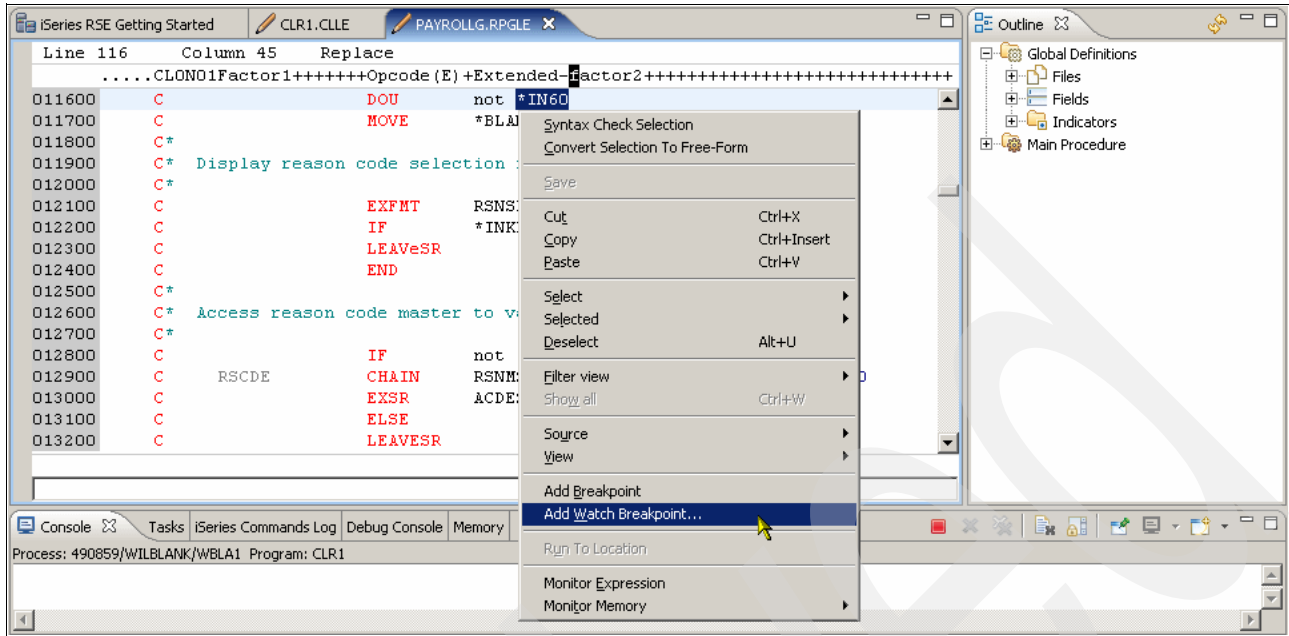


Figure 12-21 Add a watch breakpoint

3. Right-click, select **Add Watch Breakpoint** from the context menu and the Add a Watch Breakpoint dialog opens, as shown in Figure 12-22.

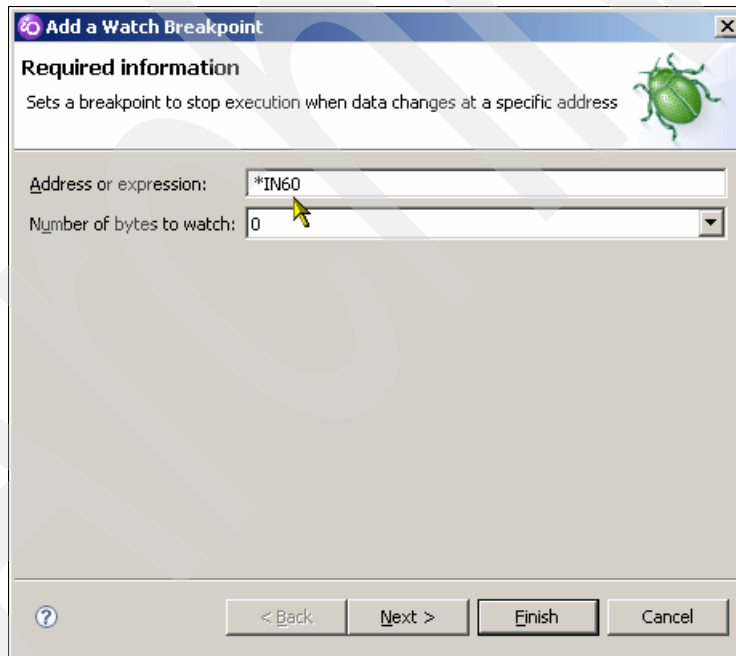


Figure 12-22 Add a Watch Breakpoint dialog

As you can see in Figure 12-22, the Expression field is pre-filled with the highlighted variable **IN60*. By default the *Number of bytes to watch* field is set to zero, which means the variable will be watched in its defined length.

4. If you click **Next**, the Add Watch Breakpoint - Optional parameters appears. It allows you to set a frequency of how often to perform the watch breakpoint.

The debugger keeps track of how many times each breakpoint is encountered. The fields in this section tell the debugger on which encounter of a breakpoint the debugger should first stop, how often it should stop, and on which encounter the debugger should no longer stop. The following parameters are used to set the breakpoint frequency:

- **From:** Enter the first breakpoint encounter you want the debugger to stop on. For example, if you want the debugger to skip over the breakpoint the first five times it is encountered, enter “6”.
- **To:** Enter the last breakpoint encounter you want the debugger to stop on. For example, if you want it to start ignoring the breakpoint after the 20th encounter, enter “20”. To have it always stop on the breakpoint, enter “Infinity”.
- **Every:** Enter the frequency with which you want the debugger to stop on this breakpoint. For example, if you want it to stop on only one out of every four it encounters, enter “4”.

5. When you click **Finish**, the Watch Breakpoint is set and you can see a new icon in the Breakpoint view, as shown in Figure 12-23.

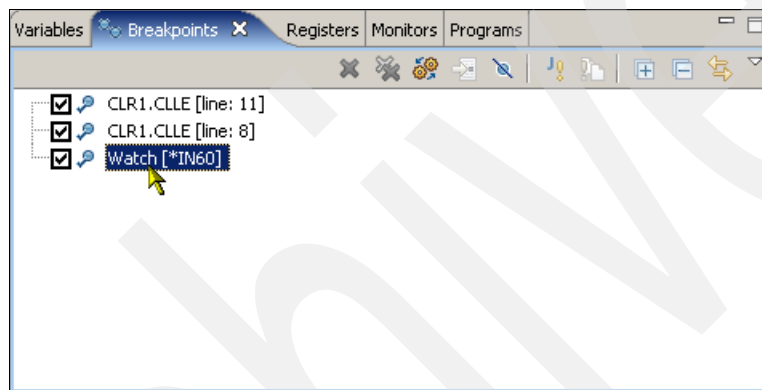


Figure 12-23 Watch Breakpoint shown in the Breakpoint view

When you run the program in debug, from now on, whenever the variable changes, the program stops and message DBG0009I is displayed indicating that the variable has changed.

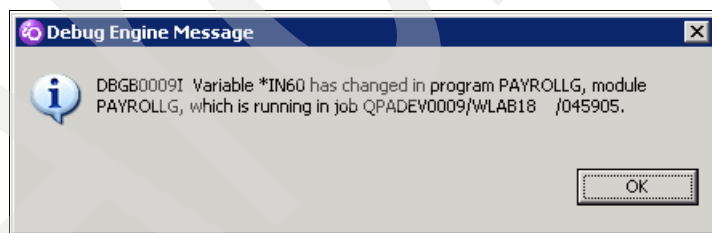



Figure 12-24 Message DBG0009I variable has changed

At this point you can either look at the contents of some variables as we describe in “Variables view” on page 432 or resume the suspended job as we explain in “Resume a suspended job” on page 432.

Halting a running application

If the application that you are debugging is running, you can halt its execution. When you are debugging an application and choose to halt, the application suspends until you issue a resume or step action.





To halt application execution, do the following:

1. In the Debug view, select one of the threads or the node that represents your application.
2. Do one of the following:
 - Click **Suspend**  in the Debug view toolbar.
 - Right-click and select **Suspend** from the context menu.

This will halt the application. The current line of execution will be highlighted in the Debugger editor.

Resume a suspended job

To resume a suspended job, do the following:

1. Select the thread, one of its stack frames, or the connection node associated with it.
2. Do one of the following:
 - Click Resume  in the Debug view toolbar.
 - Right-click and select Resume from the context menu.
 - Press F8.
 - Issue a *step over* command by clicking the  button.
 - Issue a *step into* command by clicking the  button.
 - Issue a *step return* command by clicking the  button.

The application will resume execution and, if it runs to completion, call stack entries will no longer be displayed for the thread.

Variables view

You can inspect variables in the Variables view by default in the upper right pane of the debug perspective. You can do so only if you are debugging ILE COBOL or ILE RPG for programs compiled with i5/OS V5R3 or higher level, or if you are debugging C, C++.

Important: If you want to inspect the contents of variables for other languages (such as CL or non-ILE) or for an older release of OS/400, you need to monitor the variables as described in “Monitors view” on page 434.

The Variables view displays all variables for a selected call stack entry. To view and inspect one or multiple variables during the whole debug session, use the Monitors view, as we describe in “Monitors view” on page 434. You can also add variables from the Variable view to the Monitor view. Right-click a variable in the Variables view, then select **Monitor local variable**.

The Variables view always shows the variables of the currently-selected call stack entry. When a thread suspends, the top call stack entry of the thread is automatically selected. When a call stack entry is selected, the visible variables in that call stack entry are displayed in the Variables view. Complex variables, such as arrays or the RPG indicator (by clicking the plus sign on the left of *IN in Figure 12-25) for example, can be expanded to show the elements that make up the variable.

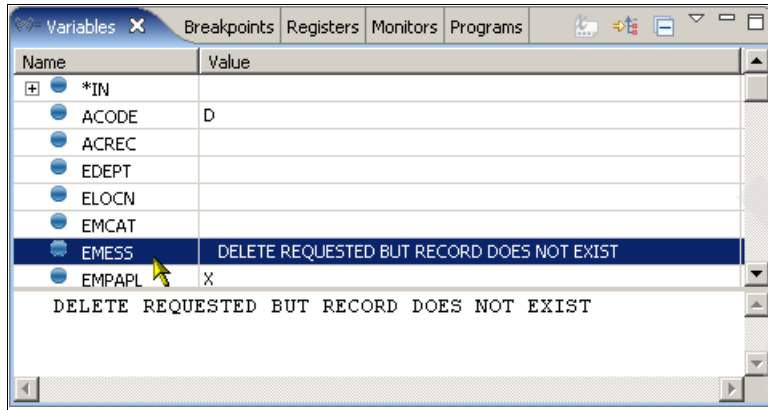


Figure 12-25 Variables view

If you click one of the variables, as shown in Figure 12-25, the content is also shown in the pane below the list.

If desired, you can change the representation of the variable by right-clicking the variable and selecting **Change Representation** from the context menu. Then select the desired representation from the context menu, for example **2 Hexadecimal** as in Figure 12-26 on page 433.

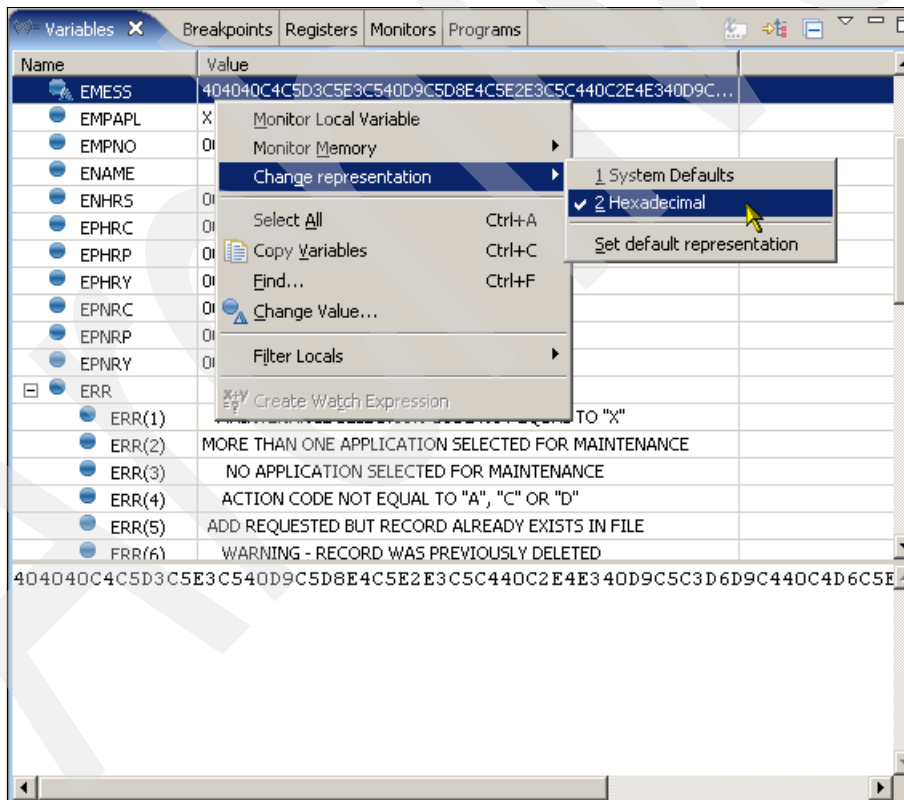


Figure 12-26 Change Representation in the Variables view to hexadecimal

Using filters to reduce the number of variables shown

You can turn filters on for either predefined or user-defined variables for ILE COBOL or ILE RPG programs. This is useful if you have performance problems when debugging ILE COBOL or ILE RPG, because there are so many variables displayed in the Variables view.

To set filters for local variables:

1. Right-click any variable shown in the view.
2. Select one of the following:
 - **Filter locals** → **User-defined variables** if you want the view to show only those variables you have defined in your program.
 - or **Filter locals** → **Predefined variables** if you want the view to show only those variables designated as reserved expressions by the language (such as RPG indicators) you are debugging.

A check mark appears next to the selection. By default, both options are selected. If you clear both options, no filtering takes place and both predefined and user-defined variables are displayed.

Monitors view

The Monitors view shows variables or expressions that you have selected and wish to monitor at all times. As long as the value of the variable or expression can be determined, its value is shown.

To add an expression from the editor to the Monitors view:

1. In the editor, first double-click (highlight) and then right-click the expression that you want to monitor, as shown in Figure 12-27.
2. Select **Monitor Expression** from the pop-up menu. If not already open, the Monitors view is shown in the upper right pane of the Debug perspective with a new icon representing the variable to be monitored as shown in Figure 12-27.

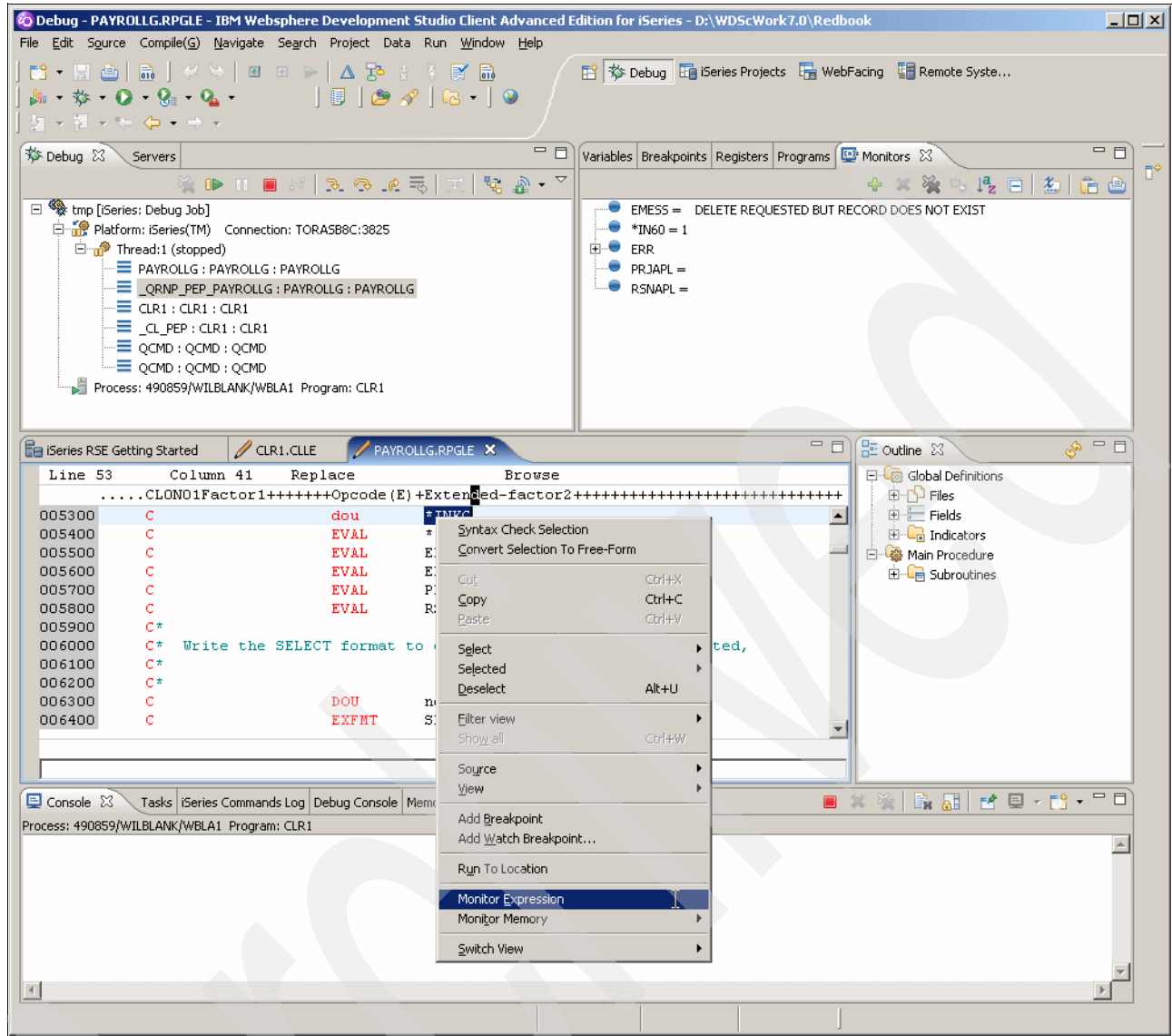


Figure 12-27 Select Monitor Expression in editor

You can also add a variable or expression to the Monitors view, by selecting the thread from the Debug view. Since the method through the editor described above and in Figure 12-27 is much easier, we do not describe the other way here.

Viewing the contents of a variable or expression in the Monitors view

To view the contents of a variable or expression you have already added to the Monitors view:

1. Use the scroll bars to scroll the view until the variable is visible.
2. If your variable is a structure or array, it can be expanded to show its individual elements.
3. If desired, change the representation of the variable: right-click on the variable and select **Change Representation** from the pop-up menu. Then select the desired representation from the resulting pop-up selections. A check mark will appear beside the current representation.

Nodes in the Monitors view are visible if they are at the top level of the tree structure, or if the nodes a level higher have been expanded. For all nodes that are visible in this view, you can

jump to a specific node by typing the first letter of the node. The view will jump to the next node that begins with that letter. If this node is not the node that you wanted to jump to, type the first letter again to jump to the next node that begins with the letter, and so on.

Changing the contents of a variable

To change the contents of a variable in the Monitors view, perform the following steps:

1. Select the variable whose value you want to modify.
2. If your variable is a structure or array, expand it to show its individual elements.
3. Scroll down to the variable you want to change and do one of the following:
 - Double-click the variable or variable element.
 - Right-click the variable or variable element and choose **Edit Value** from the pop-up menu.

Note: If you double-click on a variable and its value field cannot be edited, the variable is a type that cannot be modified (for example, constants and arrays).

4. Enter a new value for the variable or variable element and press Enter or click away from the variable value field to submit the new value.
5. To indicate that the variable value has changed, its indicator will have a delta symbol next to it. All variables affected by the change will also have a delta symbol next to their indicators. In Figure 12-28 you can see variable EMESS, which has been changed, and *IN60, which can be changed at this moment.

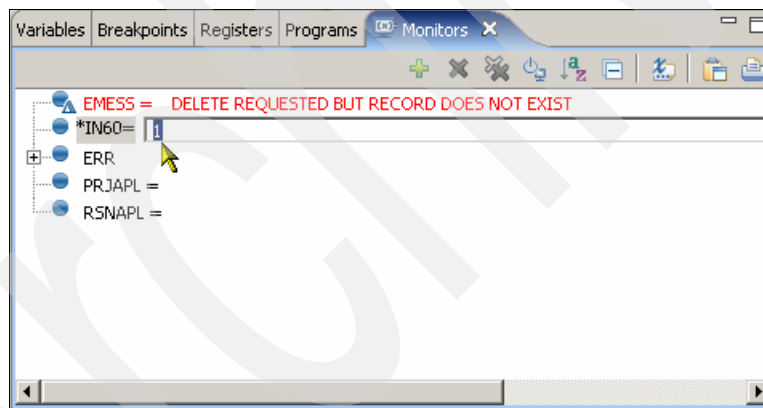


Figure 12-28 Changing the contents of a variable and indicating changed contents

Note that while some variables can be edited in place, as shown in Figure 12-28, you will see a pop-up window to change the contents of other variables.

If the same variable is displayed in more than one monitor, the changes you make are reflected in all monitors. However, if two variables with the same name qualify for different programs or threads, they are considered to be two different variables. Therefore, a change to variable A in Program 1 or Thread 1 does not affect the value of variable A in Program 2 or Thread 2.

Enabling and disabling a monitored variable or expression

You can disable the monitoring of a variable or expression in the Monitors view. The advantage of disabling a monitored expression instead of deleting it is that it is easier to enable a monitored expression than to recreate it.

To enable or disable a monitored expression:

1. Locate the variable or expression you want to disable or enable in the Monitors view.
2. Right-click the variable or expression you want to enable or disable.
3. Select or clear **Disable Monitored Variable** from the pop-up menu.

Removing a monitored expression

To remove a monitored expression that you have added to the Monitors view:

1. Select the expression that you want to remove.
2. Do one of the following:
 - Press the Delete key.
 - Right-click in the expression and select **Remove Monitored Expression** from the context menu.
 - Click **Remove Monitored Expression** in the Monitors view.

Programs view

The Programs view displays and allows you to navigate programs, service programs, ILE modules, and Original Program Model (OPM) programs.

The programs or service programs in the list can be expanded to show modules. When you are viewing programs or service programs, clicking the **Show programs in debug** icon (see the mouse pointer in Figure 12-29) will filter out modules without debug data, leaving only the programs with debug information. To display non-debuggable programs, click this icon again.

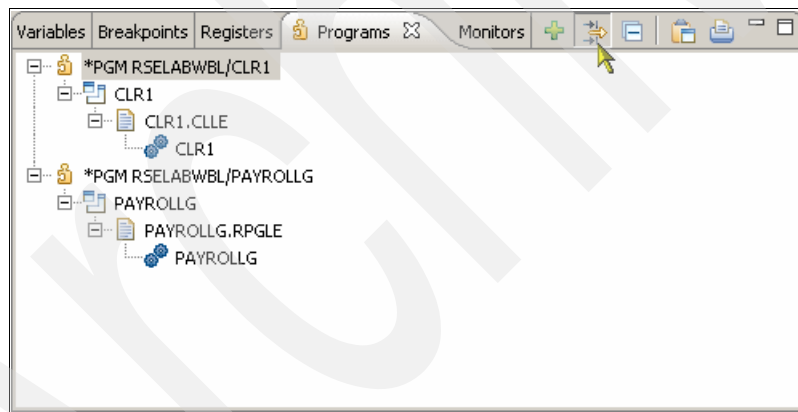


Figure 12-29 Programs view

You can have the properties of compile units, modules, procedures, and methods displayed in the Properties view, as follows:

1. Open the Properties view by choosing **Window** → **Show View** → **Other** and then selecting the Properties view from the set of Basic views.
2. In the Programs view, go to the item whose properties you wish to view. If necessary, expand the nodes and use the scroll bars, Up and Down keys, or PageUp and PageDown keys to scroll the view until the item is visible.
3. Select the module to have its properties appear in the Properties view.

Entries in the Programs view are visible if they are at the top level of the tree structure, or if the nodes at a level higher have been expanded. For all nodes that are visible in this view, you can jump to a specific node by typing the first letter of the node. The view will jump to the

next node that begins with that letter. If this node is not the node that you wanted to jump to, type the first letter again to jump to the next node that begins with the letter, and so on.

Memory view

The Memory view in the bottom pane of the debug perspective, by default, allows you to look at and change the contents of storage areas used by your program. You can also modify the address range to view and change the content of storage. The content can only be changed for programs using the *TERASPACE storage model.

Adding an expression to the Memory view

To add an expression from the Memory view, perform the following steps:

1. In the editor, double click and then right-click the expression for which you would like to monitor storage.
2. Select **Monitor Memory** from the context menu. Then select how you want to render the view. For example, HEX and Character or EBCDIC.

Viewing a location in storage

To view the contents of storage from the Memory view, perform the following steps:

1. Select the storage monitor that contains the storage location that you would like to view.
2. If necessary, use the scroll bar in the monitor to view storage locations above or below the starting address of the Memory view. Alternatively, right-click in the monitor and choose the **Go to Address** pop-up menu item. This will open the Go to Address dialog, in which you can enter an address that you would like to jump to.
3. Click **OK** to exit the dialog and go to the address. If the address was not already visible in the view, the address will become the top entry in the view. If the address was already visible in the view, the cursor will move to the entered address.
4. If desired, change the size (in bytes) of the memory content columns by right-clicking in the storage monitor and selecting **Format** from the pop-up menu. This opens the Format dialog, in which you can set the units per row and units per column.

Note: When you set a default column size in the Default Column Size dialog, it is not used by any storage monitors that you currently have open. It will be used for the next storage monitor that you create.

If you are in a storage monitor and move away from the address that you originally set to monitor, selecting **Reset to Base Address** (or choosing the **Reset to Base Address** pop-up menu item) will position the cursor back to the base address of the memory monitor.

Terminating a debug session

First users of the Integrated Debugger may likely expect that they are in a debug session whenever they see the Debug perspective. This is not correct! If you just open the Debug perspective it does not start a debug session. Likewise (and more important to understand), closing the Debug perspective does not terminate any debug session.

We talked earlier, in 12.3.2, “Start debugging an application using a service entry point” on page 414 about starting a debug session. There are a number of ways to end a debug session. This section describes various methods.

Note: If the program was launched via a launch configuration and the launch configuration “Terminate debug session on program completion” check box was not selected, you can suspend program execution and then perform a terminate action or you can terminate the application as it is running.

Terminate

Terminating results in an immediate end to a debug session, and the program that you are debugging continues to run. You can terminate a debug session when the application is running and when program execution has suspended.

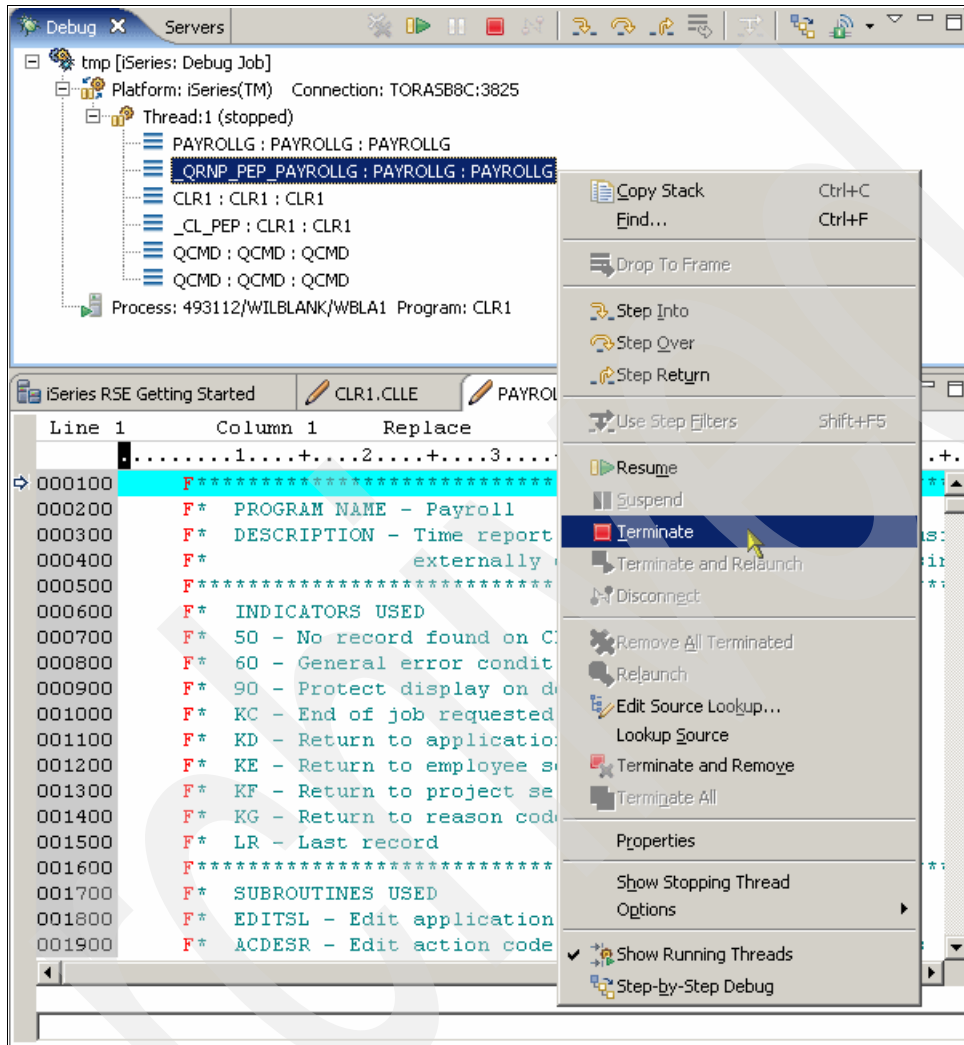



Figure 12-30 Terminate a debug session

To terminate a debug session, perform the following steps:

1. In the Debug view, the upper part of Figure 12-30 on page 439, select the launched debug session that you want to terminate.
2. Do one of the following:
 - Select **Run** → **Terminate** from the main menu in the Debug perspective.
 - Right-click and select **Terminate** from the context menu. Alternatively, you can select **Terminate and Remove** to terminate the session and remove it from the Debug view.
 - Click the **Terminate** button  in the bar above the Debug view.

To terminate all debug sessions, right-click in the Debug view and choose **Terminate All** from the pop-up menu.

If you have terminated one or more sessions and would like to remove them from the Debug view, right-click in the Debug view and choose **Remove All Terminated** from the pop-up menu.

When you are debugging an interactive job, terminating the debug session will not terminate the running program. Switch to the 5250 session and terminate the program before starting a new debug session for the interactive job.

Running an application to termination

If you are debugging an application that runs to termination (that is, it ends normally, for example by setting *on* the LR indicator in RPG or executing Stop Run in Cobol), and which contains no active breakpoints or unmonitored exceptions, choosing to **Resume** suspended threads will result in an end of the debug session (that is, the program will run to completion and the debug session will terminate).

12.3.4 Attach to a running application by debugging a System i job

As we mentioned earlier, we believe using a service entry point to be the most versatile method to start a debug session. However, there may be situations where you might prefer to attach the debug session to a running application. For example, if you want to debug a batch job on hold, or you already pass the entry point of the program, module, or procedure, you should choose to debug the job. Other reasons for debugging a specific job include the following:

- ▶ You anticipate a problem at a particular point in your program, and do not want to step through the program or set breakpoints. In this situation, you can run your program, and during a program pause shortly before the anticipated failure (for example, while the program is waiting for keyboard input), you start debugging the job. You can then provide the input, and debug from that point on.
- ▶ You are developing or maintaining a program that hangs sporadically, and want to find out why it is hanging. In this situation, you can debug the job, and look for infinite loops or other problems that might be causing your program to hang.
- ▶ An enhancement introduced with WSDC V7.0 and WSDC AE 7.0 is the possibility to attach to a suspended job, for example a job waiting for an error message to be responded to. For example, if a program tries to call another program that does not exist and the calling program did not monitor that error condition, a message is sent to the QSYSOPR message queue and waits for a reply.

The job still appears as being active. However, prior to WSDC V7.0 you could not debug such a job. Using the method described in this section, you can attach a debug session to this job as long as the error message has been applied. Although you cannot execute any program statements, you can still use the Variables view to see the content of the variables as described in “Variables view” on page 432, or examine the call stack to find the one that caused the exception.

To launch a debug session other than using a service entry point, you can do one of the following:

- ▶ You can choose *one-step debugging*, which involves selecting the job or program object that you wish to debug.
- ▶ You can choose to *debug with full prompt*, which means that you use a launch configuration dialog to launch the debug session. You can use full-prompt debugging with one or more programs or service programs.

In the following sections, we describe how to debug a System i job using either one-step debugging or with full prompt. Later, in 12.3.5, “Launching a program object for debugging” on

page 445, we describe the somewhat older methods to launch a program for being debugged.

One-step debugging a running application

If you are debugging a job and do not need to specify any parameters, you can start a *one-step debug session*, which is a session that does not use a launch configuration you created before, but rather uses an implicit configuration with default values.

The filters in the Remote Systems view allow you to easily find any job running under i5/OS. For example, if you submitted the job under the same user profile as currently used for the System i connection in RSE, you just select **System i Jobs** → **My active jobs**. If somebody else started the job and you know the name of the user profile, you can create a new filter for that. For more information, see 7.3.4, “System i jobs subsystem” on page 172.

Once you locate the job that you want to debug in Remote Systems view, right-click it and select **Debug As** → **System i Job**, as shown in Figure 12-31.

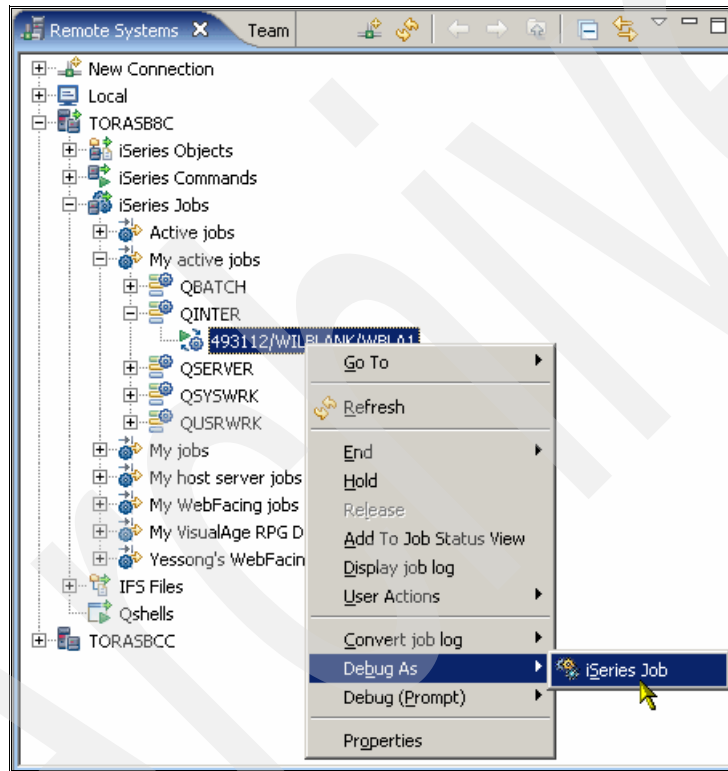



Figure 12-31 Single-step debug System i job

A debug session launches and you see the Debug perspective. Refer to 12.3.3, “How to use the power of a debugging session” on page 418 to understand how to use the tools to actually debug your program.

Debugging a running application with full prompt

To debug a job with a full prompt of launch parameters, you first need create a *launch configuration* (also referred to as *debug configuration*) for the debug session. To do so, perform the following steps:

1. Although not required, the best point to start is by selecting the job to be debugged in the Remote Systems view as described in the previous section. However, if you do not do so, you need to enter the connection and the job number manually into the next dialog.

2. Perform one of the following steps:
 - Click the down arrow of the debug button () and then select **Debug**.
 - Select **Run** → **Debug** to open the Launch Configuration window.
 - Right-click that job and select **Debug(Prompt)** → **System i Job**.

The Launch Configuration dialog as shown in Figure 12-32 on page 442 opens.

Note that depending on whether and how you used the Debug configuration before, you may see different content in the right pane compared to that in Figure 12-32 on page 442. The reasons for this are:

- The Launch Configuration dialog tries to open the last used configuration again.
- If you did debug a job as described in “One-step debugging a running application” on page 441, a configuration has been created for you.

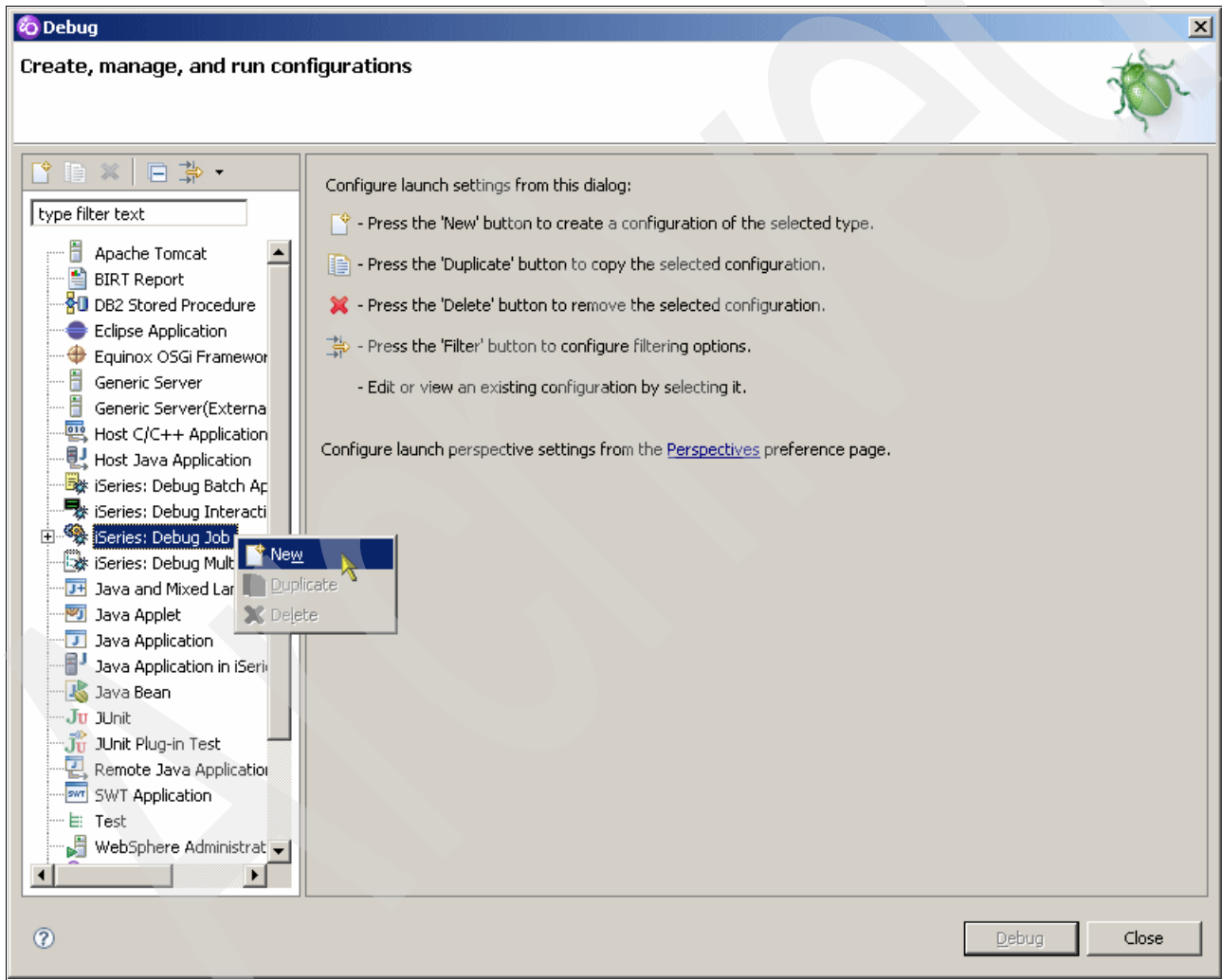


Figure 12-32 Debug Launch Configuration window

In the left pane of the Debug launch configurations dialog box, right-click the **System i: Debug Job** entry and select **New** from the context menu. A New_configuration dialog opens in the right pane.

Tip: You may click the Help (question mark) button in the lower left of the dialog shown in Figure 12-32 (as well as many others), to see context-sensitive online help. This capability did not exist in WDS 6.01 and before.

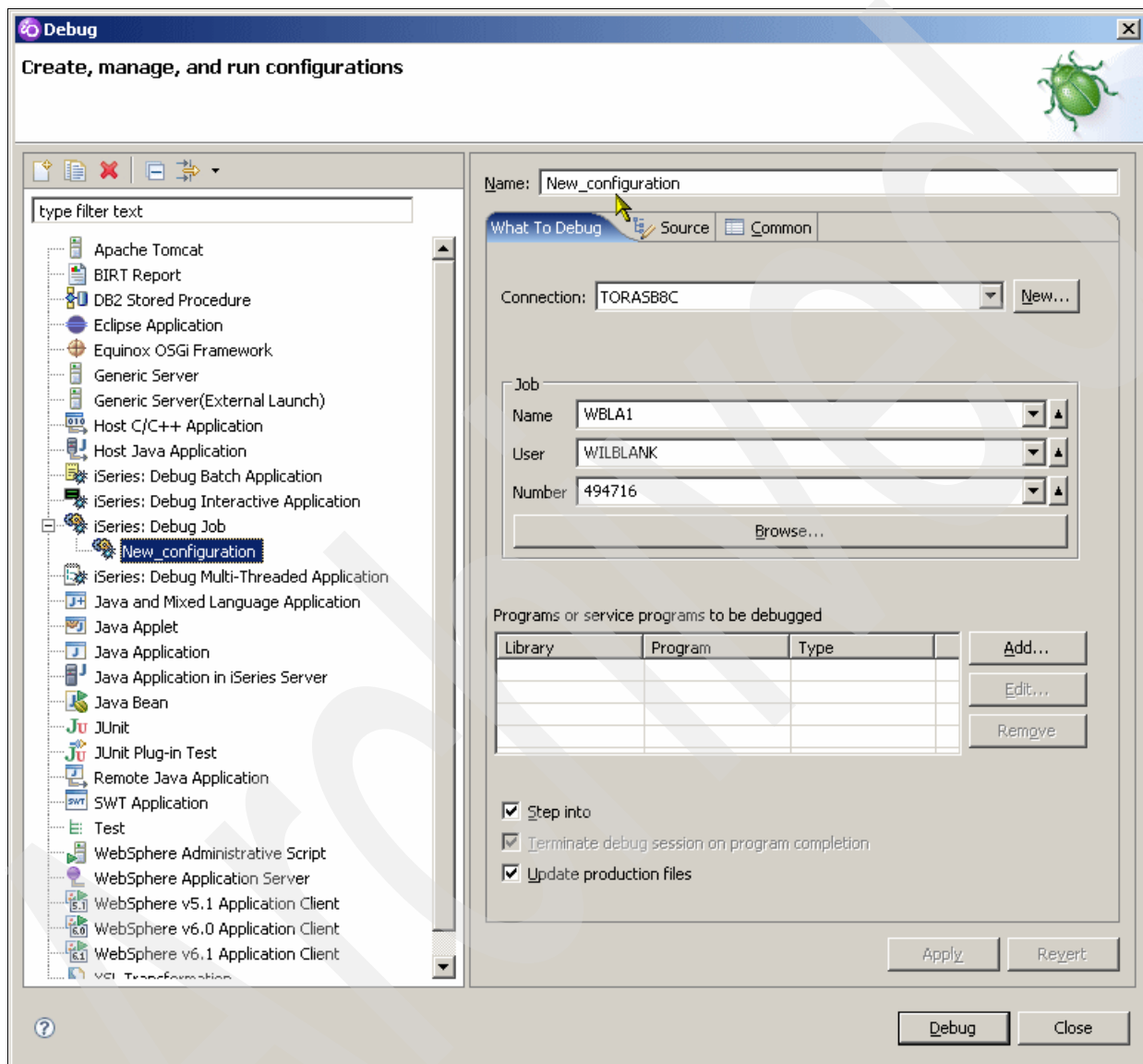


Figure 12-33 *New_configuration dialog for debugging a System i Job*

3. In the launch configuration Name field, above the mouse pointer in Figure 12-33, you may replace `New_configuration` with a unique name of your choice for the configuration that you are creating. This name will show up in the Debug menu later, so you can select to launch a debug session again without entering all parameters again.
4. Select any of the three launch configuration tabs, **What to Debug**, **Source**, or **Common**, and complete them according to the description in the sections below.
5. When all information mandatory to the launch configuration has been provided, the **Debug** and **Apply** push buttons are enabled. This should be the case if you started by selecting an active job in the Remote Systems view as suggested in Step 1.

Click **Apply** to save the launch configuration to run later, or click **Debug** to launch a debug session with the new settings now.

If there is no debuggable program currently active in your job at this point, you are prompted to invoke your application in the job that you specified. Once the program is called, the debug session launches.

What To Debug page

Click the **What To Debug** tab to bring the page to the foreground if it is not already there.

The two mandatory fields in the What To Debug page, Connection and Job, already contain the correct values, if you selected the job to be debugged in Step 1.

In the *Programs or service programs to be debugged* list, you can enter the name and library name for one or more programs or service programs you want to debug.

If the *Step into* check box is *not selected*, the debugger will stop at the first encountered breakpoint. This could be a *profile breakpoint* (one that you set in a previous debug session) or a *source breakpoint* (one that you set in the Debugger editor before starting to debug, as described in 12.2.6, “Setting source line breakpoints prior to debugging” on page 410).

If *Step into* is selected and you specified an initial program or service program to debug in the “Programs or service programs to be debugged” table, the debugger steps into the specified program or service program object.

If *Step into* is selected and you did not specify any program or service program in the “Programs or service programs to be debugged” table, the debugger stops at the next debuggable statement in the application.

If you select *Terminate debug session on program completion* and a valid program (but not a service program) is specified as the initial program in the “Programs or service programs to be debugged” table, the debug session ends when the program runs to completion.

If the *Update production files* check box is selected, the application under debug is allowed to update database records in production libraries.

Source page

By default, the System i Default and Debug engine source containers are listed. The System i Default source container allows the debugger to look for the source in the location from which the program object is compiled. The Debug engine source container provides the text for views other than *SOURCE view (for example, the *LISTING view or *STATEMENT view).

Here, you can also specify your own source lookup path in a similar way as what we described for service entry points in “Source file not found” on page 423.

If the *Search for duplicate source files on the path* check box is not selected, the debugger searches the specified paths in sequence and uses the first occurrence. If this check box is selected, the debugger searches the specified paths and lists all occurrences, allowing you to select one.

Common page

If you want the launch configuration to be stored locally and available for a single user, ensure that the Local radio button is selected. If the launch configuration is to be stored as a file in the workspace and shared in a repository for team use, select the Shared radio button and enter the project location where you want the launch configuration to be stored in the “Location of shared configuration” field. Alternatively, you can browse for the project.

If you want the launch configuration to appear in the Debug favorites menu or the Debug menu Debug History list, select the Debug check box in the “Display in favorites menu” area.

12.3.5 Launching a program object for debugging

Instead of using a service entry point (described in 12.3.1, “Service entry point for ILE applications” on page 413) or attaching a debug session to a running job (12.3.4, “Attach to a running application by debugging a System i job” on page 440), you can also invoke a program directly out of your Remote Systems view and start a debug session for it.

Meanwhile, you should be able to handle most situations much easier with an SEP or by attaching the debug session to a running job. Therefore, in the following sections we cover how to use the older methods only very briefly.

Debugging a batch application

This option offers a quick way of debugging batch System i applications. When you choose to debug a batch application, the debugger submits a job for you. You are required to specify the program to debug, as well as the SBMJOB command that starts your application.

One-step batch application debugging

If you are debugging a program and do not need to specify any parameters for your program, you can start a one-step batch debug session.

In the Remote Systems view, right-click a program object and select **Debug As → Batch**.

This creates a debugging session with the following launch settings:

- ▶ The debugger will stop at the first debuggable entry of the selected program.
- ▶ The debug session will end when the program runs to completion.
- ▶ The application will be launched for the debug session assuming the following command:

```
SBMJOB CMD(CALL PGM(ProgramName)) HOLD(*YES)
```

The debugger will release the batch job and the debug session will begin.

If you want debug session launch settings other than the above, you need to launch the debug session with full prompt. For information about this, see “Debugging a batch application with full prompt” on page 445.

The debugging session starts by calling the selected object and steps into the first debuggable line. Now you can start debugging your program. For more information about debugging, see 12.3.3, “How to use the power of a debugging session” on page 418.

Debugging a batch application with full prompt

To debug a batch application with a full prompt of launch parameters, you create a launch configuration for the debug session similar to what we described in “Debugging a running application with full prompt” on page 441.

To create a launch configuration, do the following:

1. Open the “Debug launch configurations” dialog box. Instructions for doing this can be found in “Starting to debug.”
2. In the left side of the “Debug launch configurations” dialog box, right-click System i: Debug Batch Application and select **New** from the pop-up menu.

In the launch configuration Name field, enter a unique name of your choice for the configuration that you are creating. Later, this name will show up in the Debug menu for relaunching.

Select the launch configuration tabs and complete them according to the tables in the sections below:

- ▶ What to Debug tab
- ▶ How To Start tab
- ▶ Source tab
- ▶ Common tab

For detailed information about each page, select **Help** → **Help Contents**, from the main menu, then expand:

- WebSphere Development Studio Client for System i**
 - **Debugging System i Applications**
 - **Starting to debug**
 - **Debugging a batch application**
 - **WebSphere Development Studio Client for System i**

When all information mandatory to the launch configuration has been provided, the Debug and Apply buttons are enabled. Click **Apply** to save the launch configuration to run later, or click **Debug** to launch a debug session with the new settings now. Your application will run in a batch job that is submitted by the debugger.

After you have saved a debug launch configuration, you may edit it. For further information about editing debug launch configurations, see the related task below.

To dismiss the dialog without starting a debug session, click **Close**.

Debugging an interactive 5250 application

This option offers a quick way to debug interactive System i applications while you develop and test a single program and need to start and run through it several times. Note that now after introduction of service entry points, as well as the capability to debug running applications and to set breakpoints in the source, as described earlier in this chapter, the method described here may not always be the best choice.

When you choose to debug an interactive application using this method, the debugger will use an interactive job that is associated with a specific connection in order to run your application. You need to specify the program that you would like to debug and the program or command that will start your interactive application.

One-step interactive application debugging

If you are debugging a program and do not need to specify any parameters, you can start a one-step interactive debug session.

In the Remote Systems view, right-click a program and select **Debug As** → **Interactive**.

This creates a debugging session with the following launch settings:

- ▶ The debugger will stop at the first debuggable entry of the selected program.
- ▶ The debug session will end when the program runs to completion.
- ▶ The application will be launched for the debug session assuming the following command:
CALL PGM(*Program_Name*), where *Program_Name* is the name of the selected program.

If you want debug session launch settings other than the above, you need to launch the debug session with full prompt. For information about this, see the related topic below.

When you launch an interactive application debug session without prompt, the application will run in the interactive job associated with the System i connection where the selected program object came from. If this job does not exist, you will be prompted to associate an interactive job with the connection.

In other words: When you start debugging interactive applications, you must start a 5250 emulator session. Within that session, start the IBM RSE Communications Server with the following command:

```
STRRSESVR NAME(your-connection-name)
```

The parameter NAME is your connection name from the RSE. Note that it is case-sensitive and needs to be specified exactly as in your RSE perspective. You can also specify a working library, remote location name, and a TCP/IP port number, if you press F4=Prompt.

Note: If you start debugging without starting the RSE communication server first, you are informed to start it, as shown in the message in Figure 12-34.

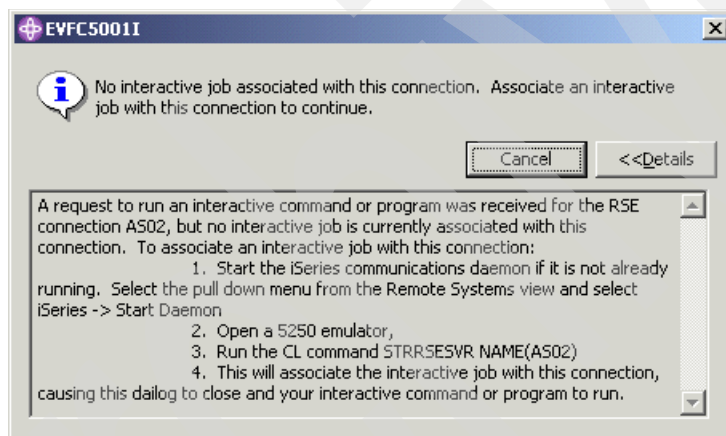



Figure 12-34 RSE Communication Server not active

Start a debugging session in Development Studio Client. Select and right-click a program object from the RSE perspective. Select **Debug As** → **Interactive** (Figure 12-35). Alternatively, you can select a program object, click the down arrow of the debug button , and select **Debug As** → **System i Interactive Application**.

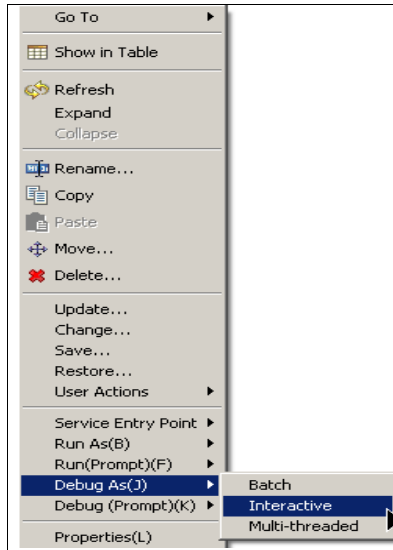


Figure 12-35 *Debug as Interactive*

The debugging session starts by calling the selected object and steps into the first debuggable line. Now you can start debugging your program. For more information about debugging, see 12.3.3, “How to use the power of a debugging session” on page 418.

12.3.6 Setting debug preferences

There are several preferences you can set for your debug sessions. To do so, select **Window** → **Preferences** from the workbench menu bar to open the Preferences page, as shown in Figure 12-36 on page 449.

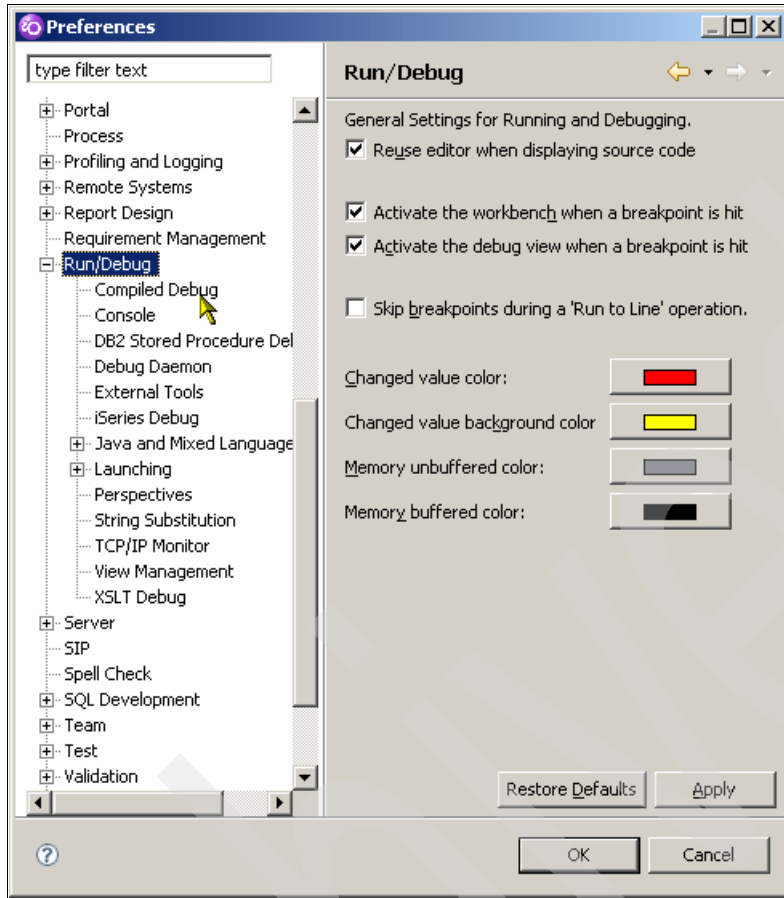


Figure 12-36 Preferences Run/Debug

In the Preferences Run/Debug page you can set some general parameters related to all kinds of debug sessions (not only System i related).

To reach the settings related the Integrated System i Debugger, expand the tree underneath Run/Debug in the left pane and select either Compiled Debug or System i Debug. The following sections describe the possibilities of the preferences you may set there.

Compiled Debug preferences

The Compiled Debug page shown in Figure 12-37 on page 450 allows you to control some behavior of the debug session.

Tip: Also in Figure 12-37 on page 450, you can see how the context help (new in WDSC 7.0) can assist you while working with dialogs by clicking the help button at the lower left of the panel.

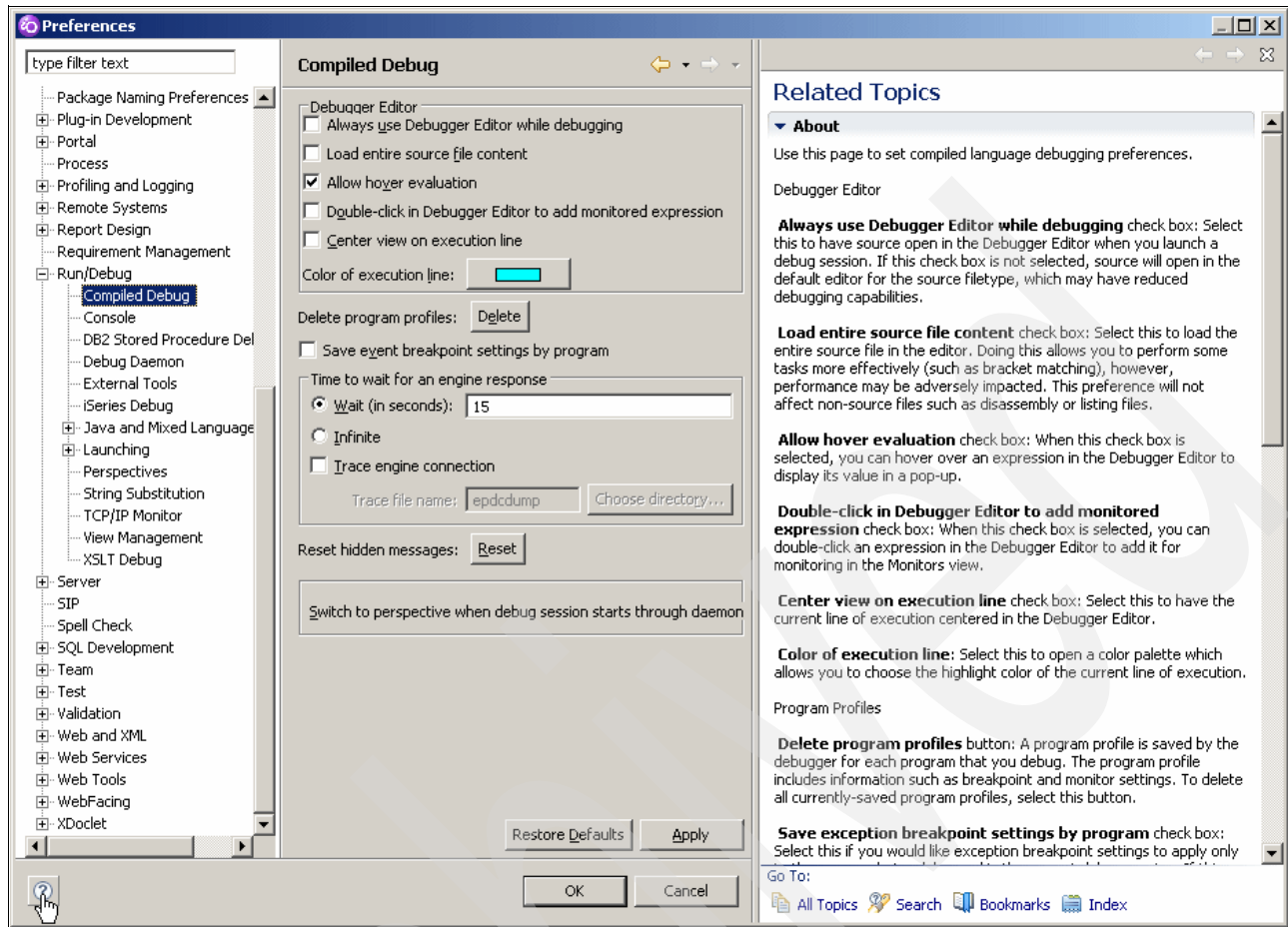


Figure 12-37 Compiled Debug preferences

When you launch a debug session, the editor used by the debugger depends on the settings in the Debugger Editor section of this Preference page.

- ▶ If **Always use Debugger Editor while debugging** is selected, then it uses the Debugger Editor. This check box is not selected by default.

Note: The Remote Systems LPEX editor provides many more functions than the Debugger Editor does. Therefore, we highly recommend using LPEX.

- ▶ Otherwise, the debugger checks whether the source is currently open in any editor.
 - If it is currently open in an editor, the debugger uses that editor.
 - Otherwise it uses the default editor associated with the source file type. In most cases this will be the Remote Systems LPEX Editor.

Select **Allow hover evaluation** if you want the value of a variable to be displayed when you hover the cursor over the variable in the editor. This option is selected by default. For the System i Integrated Debugger, this feature is available for ILE RPG, ILE COBOL, and OPM RPG.

Select **Double-click in Debugger Editor to add monitored expression** if you want to enable this option. If it is enabled, when you double-click on an expression in the editor, the

expression is added to the Monitors view. This option is disabled by default so that double-clicking simply highlights the expression.

In addition to editor preferences, you can choose to delete program profiles. A program profile is saved by the debugger for each program that you debug. The program profile includes information such as breakpoint and monitor settings. To delete all currently-saved program profiles, click **Delete**.

Select the **Center view on execution line** check box if you want to have the current line of execution centered in the Debugger editor for all debug sessions. In this section, you can also choose the color of the line of execution.

All other settings on this page do not apply to the System i Debugger.

System i debug preferences

Use this page to set preferences that will be applied when launching System i debug sessions. The settings of Update product files on this page are reflected (or prefilled) in launch configurations and used for one-step debugging.

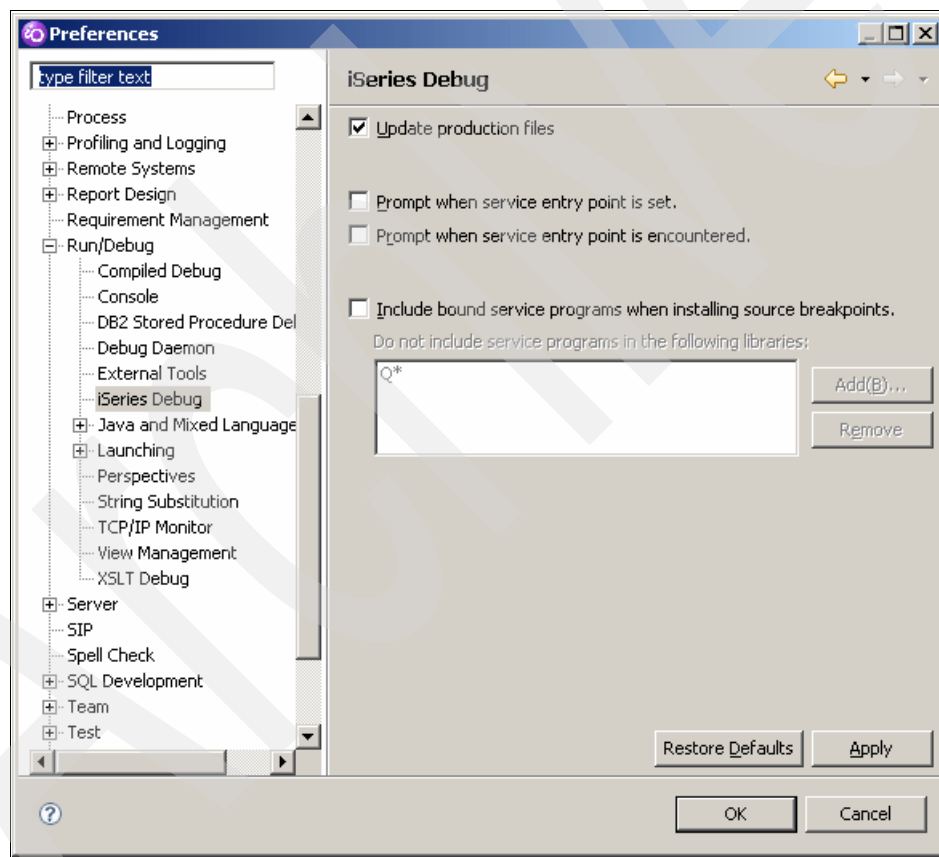


Figure 12-38 System i Debug preferences page

The **Update production files** setting causes applications under debug to be allowed to update database records in production libraries. If this check box is selected, the launch configuration What to Debug tab will have this setting selected by default.

If the **Prompt when service entry point is encountered** check box is selected, you will be notified via a dialog whenever a service entry point is triggered, and you can choose in the dialog whether you want a debug session to start or not.

If you select **Yes** in the message, a debug session for the program will be launched in the Debug perspective.

If you select **No**, you will resume the program without debugging it. If you do not want to be prompted to start a debug session right away when a service entry point is hit, select the check box in the prompt dialog or use the debugger preferences settings.

If the **Prompt when service entry point is set** check box is selected, you will be notified via a dialog whenever a service entry point is set from the Remote System Explorer.

The “Include bound service programs when installing source breakpoints” check box (a new function with WDSC 7.0) enables the source breakpoints you set for the source of your program's bound service programs to be installed and the bound service programs to be added to the debug session.

If this check box is selected, you can also specify the libraries that you don't want to include for this option. Service programs in these libraries will not be added to the debug session and their source breakpoints will not be installed.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 456. Note that some of the documents referenced here may be available in softcopy (PDF) only at

<http://ibm.com/redbooks>

- ▶ *Maximum Performance with WebSphere Application Server V5.1 on System i*, SG24-6383
- ▶ *Mastering the IBM WebFacing Tool*, SG24-6331
- ▶ *WebSphere Studio V5 Architecture and Overview*, REDP-3742
- ▶ *Patterns: Self-Service Application Solutions Using WebSphere V5.0 for System i*, REDP-3670
- ▶ *System i Access for Web and HATS Limited Edition: V5R2 Hot Topics for IBM @server System i Browser Users*, SG24-7005
- ▶ *WebSphere for the IBM @server System i Server Buying and Selling Guide*, REDP-3646

Note: This paper is frequently updated to match the current software releases. Therefore, make sure to check

<http://ibm.com/redbooks>

for the latest edition.

- ▶ *Student Edition: WebSphere Development Studio Client for System i V5.0*, SG24-7086

Note: This book is targeted to university and college students who are studying computer and information technology. It is based on the original edition of *WebSphere Development Studio Client for System i V5.0*, SG24-6961, published August 2003. The Student Edition contains additional exercises and quizzes as well as an on-going study.

- ▶ *WebSphere Portal Express and Express Plus V5 for the IBM @server System i Server*, SG24-6096
- ▶ *AS/400 Programming with VisualAge for RPG*, SG24-2222
- ▶ *Moving to Integrated Language Environment for RPG IV*, GG24-4358
- ▶ *Cool Title About the AS/400 and Internet*, SG24-4815
- ▶ *AS/400 Applications: A Fast and Easy Way to Install, Set Up and Work with VRPG and CODE/400 (ADTS CS)*, SG24-4841
- ▶ *Unleashing AS/400 Applications on the Internet*, SG24-4935
- ▶ *IBM Web-to-Host Integration Solutions*, SG24-5237
- ▶ *Who Knew You Could Do That with RPG IV? A Sorcerer's Guide to System Access and More*, SG24-5402
- ▶ *IBM Host Access Client Package Update*, SG24-6182
- ▶ *WebSphere Studio Application Developer Programming Guide*, SG24-6585

- ▶ *WebSphere Application Server V5 for System i: Installation, Configuration, and Administration*, SG24-6588
- ▶ *HTTP Server (powered by Apache): An Integrated Solution for IBM @server System i Servers*, SG24-6716
- ▶ *EJB 2.0 Development with WebSphere Studio Application Developer*, SG24-6819
- ▶ *WebSphere Studio Application Developer Version 5 Programming Guide*, SG24-6957
- ▶ *WebSphere Application Server - Express V5.0 for System i*, REDP-3624

Other publications

These publications are also relevant as further information sources:

- ▶ *HTTP Server for AS/400 Programming*, GC41-5435
- ▶ *Programming with VisualAge RPG*, SC09-2449
- ▶ *VisualAge RPG Parts Reference*, SC09-2450
- ▶ *VisualAge RPG Language Reference*, SC09-2451
- ▶ *System i Performance Capabilities Reference*, SC41-0607
- ▶ *IBM @server i5 and System i System Handbook*, GA19-5486

Note: This book is frequently updated to match the current hardware and software releases. Therefore, make sure to check

<http://ibm.com/redbooks>

for the latest edition.

- ▶ *Understanding the IBM WebFacing Tool* by Claus Weiss and Emily Bruner. Mc Press, September 2003, ISBN 1931182094
- ▶ *Java for RPG Programmers, Second Edition* by George Farr and Phil Coulthard. Mc Press, April 2002, ISBN 193118206X
- ▶ *Experience RPG IV Tutorial* by Heather Rogers and M. Masri. Advice Press, April 1999, ISBN 1889671223
- ▶ *VisualAge for RPG by Example*, by Bryan Meyers and Jef Sutherland. 29th Street Press, April 1998, ISBN 1882419839

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ If you do not have WDSi installed on your workstation, you can also access the online help at:
<http://publib.boulder.ibm.com/infocenter/iadthelp/index.jsp>
- ▶ WebSphere Development Studio for System i
<http://www.ibm.com/software/awdtools/wds400/>
- ▶ The IBM WebFacing Tool
<http://www.ibm.com/software/awdtools/wdt400/about/webfacing.html>
- ▶ Technical resources for the IBM WebFacing Tool:
<http://ibm.com/servers/enable/site/ebiz/webfacing/start.html>
- ▶ IBM Web application development tools for System i

- ▶ [http://www.ibm.com/software/awdtools/System i/](http://www.ibm.com/software/awdtools/System%20i/)
- ▶ Technical support and resources for System i and AS/400
<http://www-912.ibm.com/>
- ▶ WebSphere Studio Site Developer
<http://www.ibm.com/software/ad/studiositedev>
- ▶ WebSphere Studio Application Developer
<http://www.ibm.com/software/awdtools/studioappdev>
- ▶ The eclipse.org Web site
<http://www.eclipse.org>
- ▶ The Concurrent Versions System home page
<http://www.cvshome.org>
- ▶ IBM WebSphere Application Server Version 4.0 Advanced Edition for System i
<http://publib.boulder.ibm.com/was400/40/AE/english/docs>
- ▶ The IBM @server Workload Estimator
<http://www-912.ibm.com/wle/EstimatorServlet>
- ▶ The Struts home page
<http://jakarta.apache.org/struts>
- ▶ System i Information Center
[http://www.ibm.com/eserver/System i/infocenter](http://www.ibm.com/eserver/System%20i/infocenter)
- ▶ WebSphere Host On-Demand
<http://ibm.com/software/webservers/hostondemand>
- ▶ System i Access for Web
[http://ibm.com/servers/eserver/System i/clientaccess/webaccess.htm](http://ibm.com/servers/eserver/System%20i/clientaccess/webaccess.htm)
[http://ibm.com/servers/eserver/System i/clientaccess/wabook.htm](http://ibm.com/servers/eserver/System%20i/clientaccess/wabook.htm)
- ▶ WebSphere Host Publisher
<http://www.ibm.com/software/webservers/hostpublisher>
- ▶ WebSphere Transcoding Publisher
<http://www.ibm.com/software/webservers/transcoding>

Online tutorials

Several self-study tutorials covering WDSi are available on the Internet:

- ▶ You can register for the Web-based Training Course SW738 *Introducing IBM WebSphere Development Studio Client for System i* at:
<http://ibm.com/developerworks/websphere/library/tutorials/d1/sw738/>
- ▶ Another Web-based Training Course SW712 *IBM WebSphere Development Studio Client V5.1 - Application Development with Struts* is located at:
<http://ibm.com/developerworks/websphere/library/tutorials/d1/sw712/>
- ▶ You can find *An A – Z Hands-on Guide to IBM WebFacing Tool V5.1.2 Advanced Edition Lab* at:
http://ibm.com/servers/enable/site/education/abstracts/3e06_abs.html
- ▶ The IBM WebFacing Tool V5.1 - Advanced Customization:

http://www-1.ibm.com/servers/enable/site/education/abstracts/webfacing_abs.html

- ▶ There are also some tutorials provided with the online Help under *Tutorials and samples*. If you do not have WDSi installed on your workstation, you can also access the online help at:

<http://publib.boulder.ibm.com/infocenter/iadthelp/index.jsp>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols

- &L 227
- &L - Object or Member Library 227
- &N 227
- *CHANGE 409
- *JOBCTL 409
- *LIST 409, 423
- *LISTING 421
- *SAVF 200
- *SERVICE 409
- *SOURCE 409, 421, 423
- *STATEMENT 421
- *STMT 409
- *USE 409–410
- .metadata 86
- .sav 201
- .savf 201
- .zip 204
- /COPY members 266
 - in Application Diagram 322

Numerics

- 127.0.0.1 165
- 5722-WDS
 - option 60 78

A

- active build styles 363
- ad hoc commands 244
- Add Bookmark 32
- Add Physical File Member (ADDPFM) 167
- Add to Library List 260, 276
- ADDPFM 167
- ADTS (Application Development ToolSet) 5
- Advanced Printer Function (APF) 6
- All records screen 306
- Allow hover evaluation 450
- Always use Debugger Editor while debugging 450
- APF (Advanced Printer Function) 6
- application client project 21
- Application Component Provider 18
- Application Diagram 319
- Application Diagram Viewer 321
- Application Diagram viewer 323
- Auto-indent 286
- Automatic formatting check box 287

B

- batch application 445
- batch mode 241
- BEGSR 285–286
- BIND.CLLE 385

- bookmark 32
- breakpoint
 - frequency 429, 431
 - prior to debugging 410
 - source 411
- Breakpoints view 419, 427
- Browse local 283
- Browse project 283
- Browse remote 283
- build style
 - *NONE 386
 - CL program 386
 - command 386
 - specifying at project level 386
- builder 35
- By Date 259

C

- C/C++ 5
- CGU (Character Generator Utility) 6
- Change Command Defaults (CHGCMDDFT) command 238
- Change Object Owner (CHGOBJOWN) 228
- Change Password 164
- Change Password (CHGPWD) 161, 163–164
- Change System Library List (CHGSYSLIBL) 208
- Change Text File 423
- Change(B) 223
- Character Generator Utility (CGU) 6
- CHGCMDDFT command 238
- CHGOBJOWN 227–228
- CHGPWD 161, 163–164
- CHGSYSLIBL 208
- Clear Password 161
- ClearCase 156
- COBOL 5
- CODE 7, 408
 - Host Access Programs 6
- Code
 - filter view 259
- CODE Designer 12, 408
- Code Designer 293
- CODE Editor 408
- CODE Navigator 321
- CODE Program Generator 408
- CODE Project Organizer 12, 408
- command line interface 101
- Command position field 287
- command sets 167
 - creating 168
 - working with 171
- command shells 180
- Comments
 - filter view 259

- Compare editor 36
- Compile (Prompt) 152
- compile commands 236
- compile options 409
- Compiled Debug
 - preferences 449
- compiling source members 229
 - CHGCMDDFT command 238
 - creating programs 234
 - creating service programs 234
 - CRTBNDRPG command 232, 236
 - CRTPGM command 234
 - CRTRPGMOD command 232
 - CRTSRVPGM command 234
 - RPGLE 232
 - run-time compile options 234
 - working with compile commands 236
- conditional breakpoint 427
- conflict detection 379
- connections
 - Connection name 159
 - creating 158
 - System type 159
 - Verify host name 159
 - verifying 160
- connection-specific run-time options 236
- content assist 69, 261
- context-sensitive help 261
- Continued line indent field 287
- Control
 - filter view 259
- CoOperative Development Environment (CODE) 408
- copy 30
- copy books 266
- crash recovery 28
- Create Bound RPG Program (CRTBNDRPG) command 236
- Create source member 167
- creating objects
 - ADDPFM command 197
 - CRTDTAARA command 192
 - CRTDTAQ command 193
 - CRTMSGF command 194
 - CRTSRCPF command 195
- creating your own job filters 174
- CRTBNDRPG 237
- CRTBNDRPG command 236
- Ctrl+F5 153
- Ctrl+F8 156

D

- Data File Utility (DFU) 6
- Data File Utility bean 11
- Database Field List bean 11
- DATFMT 280
- DBG0009I 431
- DBGVIEW 409
- DBGVIEW(*LIST) 423
- deactivate profiles 158
- Debug As

- Batch 445
 - Interactive 446
- Debug as iSeries job 412
- debug configuration 441
- debug interactive 5250 application with full prompt 446
- debug perspective 418
- debug preference 448
- Debug view 419
- debug with full prompt 412, 440
- debugger 408
- Debugger editor view 420
- debugging batch applications 441
- debugging iSeries applications 409
- debugging programs in the RSE 245
- Debugging views (DBGVIEW) 409
- debugging with full prompt 441
- deleting profiles 158
- Design area 295–296, 298
- Design Area Toolbar 311
- Design page 296, 298
- Development Studio Client
 - features 29
- DFU (Data File Utility) 6
- DHCP 243
- Display File (DSPF) 295
- Display Job (DSPJOB) 172
- Distributed appropriately 284
- DO 286
- DOU 285–286
- double-click in Debugger Editor 450
- DOUxx 286
- DOW 285–286
- DOWxx 286
- drawer 303
- D-Specification wizard 280
- DSPF 295
- DSPJOB 172

E

- Eclipse consortium 17
- Eclipse project 4, 17
- Edit Library List (EDTLIBL) 145, 204
- Edit Source Lookup Path 423–424
- Editor information area 26
- editors 25, 32
- EDTLIBL 145, 204
- EJB project 21
- ELSE 285
- ELSEIF 285
- enabled breakpoint 412
- End Debug (ENDDBG) 409
- End Service Job (ENDSRVJOB) 409
- ENDDBG 409
- ENDSRVJOB 409
- enterprise application project 21
- Error List 153
- Errors
 - filter view 259
- Every parameter 429, 431
- EVFR0026E 305

Expand to 185
Extending an existing Eclipse IDE 85
External description 275
Externally-described data structure 282
EXTPGM 279
EXTPROC 279

F

F17 in PDM 149
Fast View 276
FCMU (File Compare and Merge Utility) 6
Field Table View 305
file 179
 filter 180
 importing and exporting by dragging and dropping 30
File Compare and Merge Utility (FCMU) 6
File Export/Import wizards 11
File information data structure 282
File Transfer Protocol (FTP) 200
filter
 library 183
 member 188
 object 186
 pool 190
Find String Using PDM (FNDSTRPDM) 198
FNDSTRPDM 198
folders 179
folders and files 179
FOR 285–286
free form ILE RPG code 284
frequency 427, 429, 431
frequency of a breakpoint 426
From 429, 431
FTP 200
full prompt 441

G

Generate Compile Code 384
global run-time options 234

H

host integration
 IBM WebFacing Tool 10
 WebSphere Development Studio 4
host name 163
host servers
 starting from RSE 171
How To Start page 444

I

i5/OS objects 186
IBM Distributed Debugger 408
IBM RSE Communications Server 447
IBM Toolbox for Java 11
IBM WebFacing Tool 10
IF 285–286
IFS files subsystem
 launching command shells 180

 predefined filters 177
 search for files and folders 179
 working with folders and files 179
IFxx 286
ILE 385
 debugging 408
ILE RPG 5
INAVL1005Q 327
indentation 265
Insert Variable 227
Install Packages wizard 77
Installation Guide 82
Installation Manager 75–76
installed breakpoint 412, 427
Installing WDSC 81
Integrated iSeries debugger 408
Integrated Language Environment (ILE) 385
 debugging 408
Integrated Palette 298
interactive mode 242
Interactive Source Debugger (ISDB) 6
ISDB (Interactive Source Debugger) 6
iSeries application in RSE job 153
iSeries applications 409
iSeries Build Status view 388
iSeries Commands Log 203
 running ad hoc commands 244
 view 167
iSeries Commands subsystem
 command sets 167
iSeries communications daemon 170
iSeries Debug
 preferences 449
iSeries Error List 153
iSeries extension 11
iSeries Field Table View 305
iSeries job
 working with 173
iSeries Jobs subsystem
 creating your own job filters 174
 predefined filters 172
 viewing jobs 174
 working with job filters 177
 WRKACTJOB command 172
iSeries Objects subsystem
 creating data areas 192
 creating data queues 193
 creating message files 194
 creating source physical files 195
 filter pools 190
 predefined filters 166
 WRKLIBPDM command 166
 WRKMBRPDM command 166
 WRKOBJPDM command 166
iSeries Projects
 associated library 359
 build process 387
 caching file descriptions 377
 ILE based 385
iSeries Build Status view 388

- iSeries connection 359
- managing COMPILE.CLLE 385
- push changes 378
- submitting project builds 388
- iSeries Service Entry Points view 417
- iSeries sign-on information 160
- iSeries Table View 146, 212
- iSeries table views
 - direct editing of table items 215
 - locking and unlocking 216
 - navigate views 213
 - print item list 213
 - sorting items 213
 - subsetting lists 214

J

- Java builder 35
- Java debugger for iSeries 408
- Java development tools 11
- Java Method Call wizard 283
- Java project 21
- Java Tools for iSeries examples 11
- JavaServer Page (JSP) 413
- job filter 174
- Job Status view 388
- JSP 413

L

- Label above command 287
- Label position 287
- launch configuration 441
- launching command shells 180
 - terminate 182
 - terminate and remove 182
- launchpad 76
- library filters 183
- library list 205
- Line breakpoints 411
- line number
 - not the same 275
- Live Parsing Extensible Editor (LPEX) 250
- local 200
- local history 35
- localhost 165
- LPEX 295, 321
- LPEX (Live Parsing Extensible Editor) 250
- LPEX editor
 - /COPY members 266
 - closing 273
 - copy books 266
 - filter views 258
 - online help 260
 - saving changes 266
 - source line prompting 256
 - source verify 262

M

- Main Entry Point 336

- Manage Licenses wizard 77
- Marquee tool
 - Screen Designer palette 304
- Maximum file wait time 223, 225
- Maximum record wait time 223, 225
- member filter 188
 - WRKMBRPDM command 188
- Memory Rendering view 420
- Memory view 420, 438
- message descriptions 195
- message files 194
 - working with message descriptions 195
- MNUDDS) 295
- modes of execution 241
- Modify Packages wizard 77
- Modules view 437
- MONITOR 285–286
- Monitors view 420, 434
- move workbench resources 30
- Mozilla 80
- Multi-Threaded 153
- Multi-threaded 244
- My jobs 172

N

- Named Constant 281
- named type 221
 - creating 220
 - removing 222
- Number of bytes to watch field 430

O

- Object Action 227
- object filter 186
 - WRKOBJPDM command 187
- Object filters 186
- Object Info 187
- Object List bean 11
- Objects filters 190
- Oblique 337
- One parameter per line 287
- ON-ERROR 285
- one-step debug session 441
- one-step debugging 412, 440
- online help 37, 165
- Open Member (Ctrl+Shift+A) 152
- OPM 329, 437, 450
 - debugging 408
 - using SEP 413
- Original Program Model (OPM) 329, 437
 - debugging 408
- OTHER 285
- Other jobs 173
- outline mode 334
- Outline view 24, 322, 420
 - Application Diagram 323
- overview mode 333

P

- package group 84
- Palette
 - Screen Designer 296, 298, 301
- Passwords
 - Remote Systems 162
- PDM 141
- PDM (Programming Development Manager) 6
- PDM actions 212
- PEP (program entry procedure) module 234
- personal profile 156
- perspective 21
- platform roles 18
- pool, filter 190
- PORT 243
- Port
 - Debug Server 410
- ports
 - TCP/IP 164
- pre-defined filters 177
- Preferences
 - Remote Systems 161
- pre-supplied JavaBeans 11
- Preview 296, 298
- profile
 - creating 156
 - deactivate 158
 - deleting 158
 - personal 156
 - shared 156
 - team 156
- profile breakpoint 444
- program 234
- Program Call wizard 11
- program entry procedure (PEP) module 234
- Program status data structure 282
- Program Structure Diagram 319, 322
 - Edit or browse source 346
- Program-described data structure 281
- Programming Development Manager (PDM) 6, 141
- Programs view 437
- Programs/Modules view 420
- Prompt first 228
- prompting filters 166
- Properties view 24
- proxy server 91
- Push Changes 378
- Push Selected 378–379

Q

- QDBGSVR2 410
- QPGMR 410
- QSYS 186

R

- RAD 76
- Rational Application Developer (RAD) 76
- Rational Software Delivery Platform (RSDP) 76
- rearranging views and editors 32

- record format 294
- Rectilinear line style 337
- Redbooks Web site 456
 - Contact us xiii
- Reference fields 305
- reference fields 305
- Remote Java Compile and Run 11
- Remote Servers 171
- Remote System Details 141
- Remote System Explorer (RSE)
 - ADDBNDDIRE command 168
 - compiling source members 229
 - debugging programs 245
 - perspective, opening 141
 - running programs 239
 - views, default 141
- Remote Systems Details 212
- Remote Systems LPEX Editor 250
- Remote Systems LPEX editor 295, 321
- Remote Systems Passwords 162
- Remote Systems view 24, 160
- Remove Breakpoint 412
- rename 30
- Repeat previous specification type 286
- Report Layout Utility (RLU) 6
- repository
 - for install packages 81
- Resequence lines at save 275
- Reset Perspective 274
- response file 100
- Restore to iSeries 201
- Resume debug 432
- RLU (Report Layout Utility) 6
- RMTLOCNAME 242
- RPG 5
- RPG procedure
 - creating 278
- RSDP 76
- RSE
 - communications daemon port number 243
- RSE communication server 204
- RSE subsystem 160
- RSE2001 243
- RSELABXX 407
- Run (no prompt) option 239
- Run (with prompt) option 239
- running ad hoc commands 244
- running modes 243
 - CALL command 241
 - SBMJOB command 241
- running programs
 - CALL command 241
 - from RSE 239
 - in batch mode 241
 - in interactive mode 242
 - modes of execution 241
- run-time compile options 234

S

- Save Library (SAVLIB) 201

- SAVF 200
- SAVLIB 201
- SBMJOB 153
- Screen 306
- Screen Controls 296, 298
- Screen Design Aid (SDA) 6, 295
- SDA 295
- SDA (Screen Design Aid) 6
- Search for duplicate source files on the path check box 426, 444
- search for text strings and files from the workbench 31
- searching for files and folders 179
- SELECT 285–286
- Selection tool
 - Screen Designer palette 304
- SEP 409, 413
- server project 21
- service entry point 412–413, 440, 451
- Service Entry Point (SEP) 409
- Service entry point (SEP) 413
- Service Entry Points view 417
- service program 234, 319
- setting breakpoints prior to debugging 410
- SEU (Source Entry Utility) 6
- shared profile 156
- shared resources directory 85
- sharing
 - command set 169
- Show *Listing 423
- Show Columns Customized 149
- Show in table 223
- silent installation 99
- simple project 20
- source breakpoint 411, 444
- Source Call Diagram 319, 321
- Source Entry Utility (SEU) 6
- source line breakpoint 410
- Source Lookup Path 423–424
- source lookup paths 424
- source physical file 195
- Source Refresh 153
- Source Remove Messages 153
- Start Debug (STRDBG) 409
- Start Debug Server (STRDBGSVR) 410
- Start Service Job (STRSRVJOB) 409
- Step into 444
- STRDBG 409
- STRDBSVR 410
- STRRESVR 242, 447
 - TCP/IP port 165
- STRRESVR command 170, 243
- STRSRVJOB 409
- Submit Job (SMBJOB) 153
- subsystems
 - filter pools 190
 - prompting filters 166
- Suspend Debug 432
- Swing JFormatted bean 11

T

- Table View 305
- Tasks view 24
- team environment 36
- team profile 156
- TIMFMT 280
- To parameter 429, 431
- Toggler 329
- Toolbar
 - Design Area 311
- Toolbox for Java 11

U

- Uninstall Packages wizard 77
- Update Packages wizard 77
- Update production 444
- Update production files 451
- User Actions 226
- user actions 226
- User Id 162
- user-defined tokens 262

V

- Variables view 419, 432
- Verifying source members 229
- view 23, 32
 - iSeries table 141
- Views
 - iSeries Build Status 388
 - iSeries Commands Log 141
 - Outline 141
 - Property Sheet 141
 - Remote Systems 141
 - Tasks 141
- VisualAge RPG (VARPG)
 - Host Access Programs 6
- Visualize Application Diagram 326

W

- WAITFILE 225
- WAITRCD 225
- watch breakpoint 411, 429
- WDSi xi
- WDSC Lite 82
- Web perspective 68
- Web project 21
- Web Site Designer 68
- Web tools, iSeries extensions 11
- web.xml 69
- WebFacing
 - TCP/IP port 165
- WebSphere Development Studio 4
- WebSphere Development Studio Client for iSeries 4
 - debuggers 408
- WebSphere Development Tools for iSeries 4
- WebSphere Studio Application Developer 4
- WebSphere Studio Site Developer Advanced 7
- WebSphere Studio Workbench 15

WebSphere Workbench 11
WHEN 285
wizards 68
Work with Libraries Using PDM (WRKLIBPDM) 156
Work with Members Using PDM (WRKMBRPDM) 156
Work with Objects Using PDM (WRKOBJPDM) 156, 187
Work with Service Table Entry (WRKSRVTBLE) 410
Work with user actions 226
Work with User Jobs (WRKUSRJOB) 172
workbench 20
working with file filters 180
working with folders and files 179
workspace
 location 20
WRKLIB 242
WRKLIBPDM 156
WRKMBRPDM 156
WRKOBJPDM 156
WRKSRVTBLE) 410
WRKUSRJOB 172
WYSIWYG 295

Archived



Redbooks

Mastering WDSC V7.0 and WDSC V7.0 Advanced Edition for i5/OS V5R4

(1.0" spine)

0.875" x 1.498"

460 x 788 pages



Mastering WDSC V7.0 and WDSC V7.0 Advanced Edition for i5/OS V5R4



**Learn a new way to
build applications for
i5/OS**

**Learn how to install
and manage WDSC
V7.0**

**Learn new features of
RSE**

This IBM Redbooks publication discusses new features of WDSC V7.0, focusing on Remote System Explorer (RSE) and new ways of installing and managing the products. For features of the previous version, WDSC 5.1.2, refer to *WebSphere Development Studio Client for System i Version 5.1.2*, SG24-6961-01.

Specific topics discussed in detail include:

- ▶ Installing WDSC V7.0 using a new installation management tool of Installation Manager
- ▶ Managing a WDSC V7.0 instance
- ▶ Working with i5/OS-based development resources using the Remote System Explorer
- ▶ Remote Systems LPEX Editor - The modern SEU and much more
- ▶ Screen Designer Technical Preview - The modern Screen Design Aid (SDA) in WDSC AE 7.0
- ▶ Application Diagram in WDSC AE - Visualizing program structures
- ▶ System i Projects
- ▶ Integrated System i Debugger

WebSphere Development Studio Client for System i (WDSC) is a collection of many features and functions, such as:

- ▶ Integrated Development Environment: Seamlessly integrate all your tools, teams, assets, and other workbench-based projects.
- ▶ IBM WebFacing Tool: Convert your existing 5250 interfaces to Web interfaces with minimal changes to the server application.
- ▶ Remote System Explorer and System i Development Tools: Develop and maintain System i applications while working with remote systems.
- ▶ Integrated System i Debugger: Debug code that is running on a System i machine.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**

SG24-7578-00

ISBN 0738431419