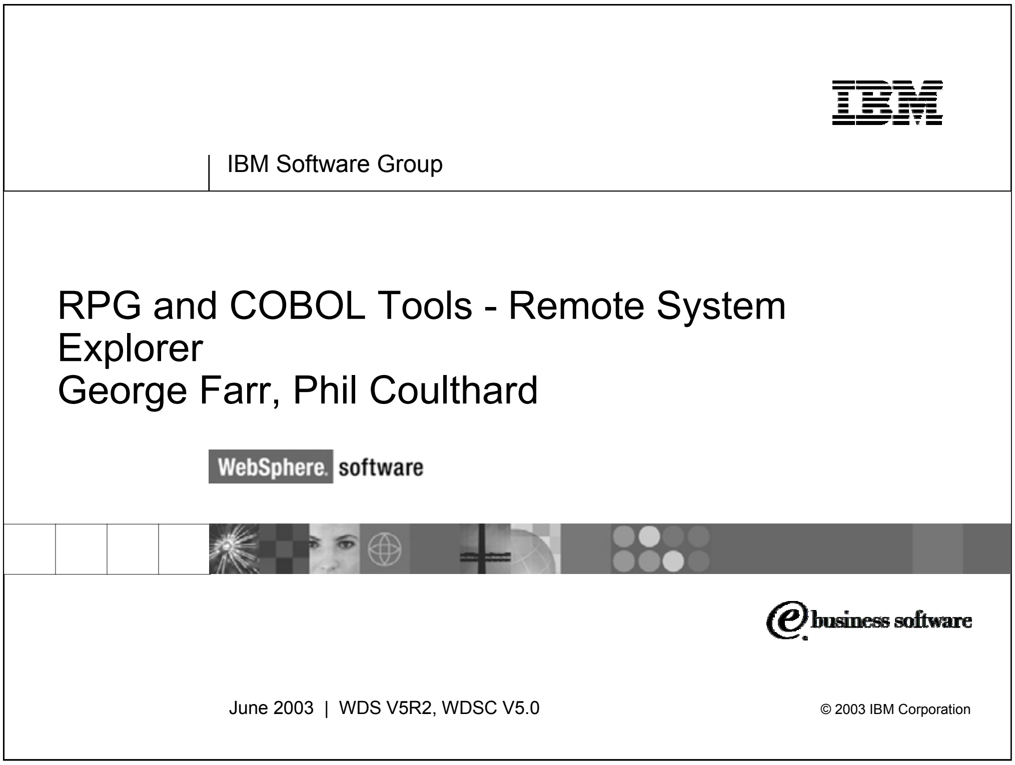


IBM Software Group

RPG and COBOL Tools - Remote System Explorer

George Farr, Phil Coulthard

WebSphere. software
e business software

June 2003 | WDS V5R2, WDSC V5.0

ADTS has been the traditional method for developing and maintaining server-side iSeries applications. WebSphere Development Studio Client for iSeries, V5.0, includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer programmers a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all.

These new generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with a Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

This presentation covers the Remote System Explorer.

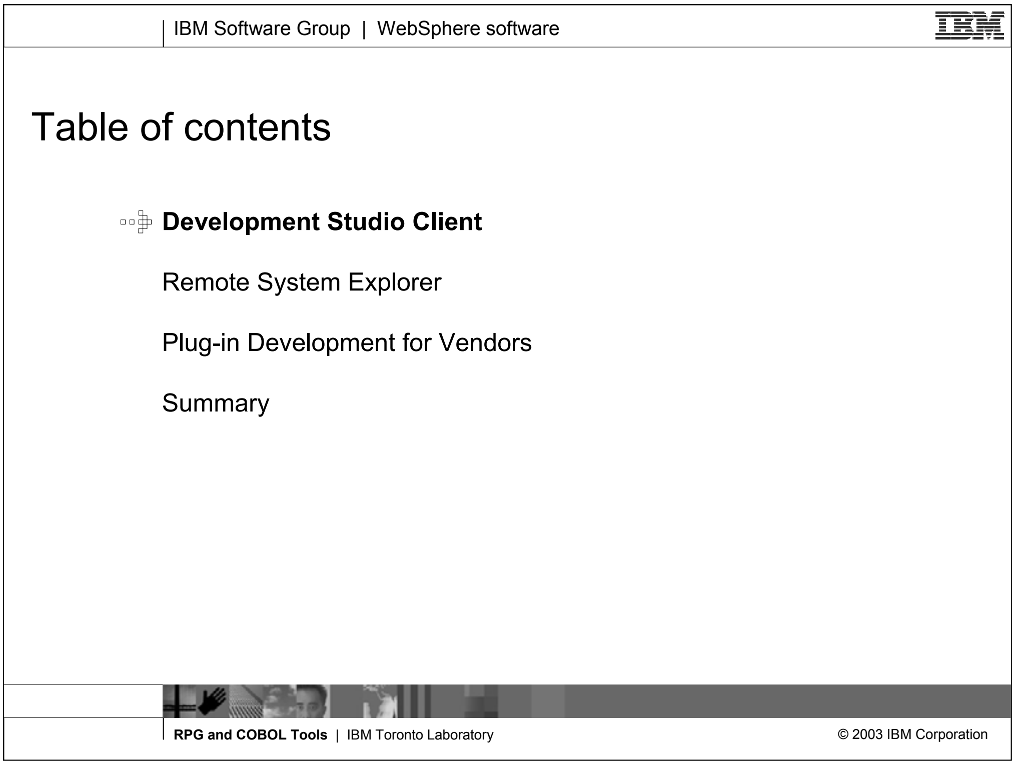


Table of contents

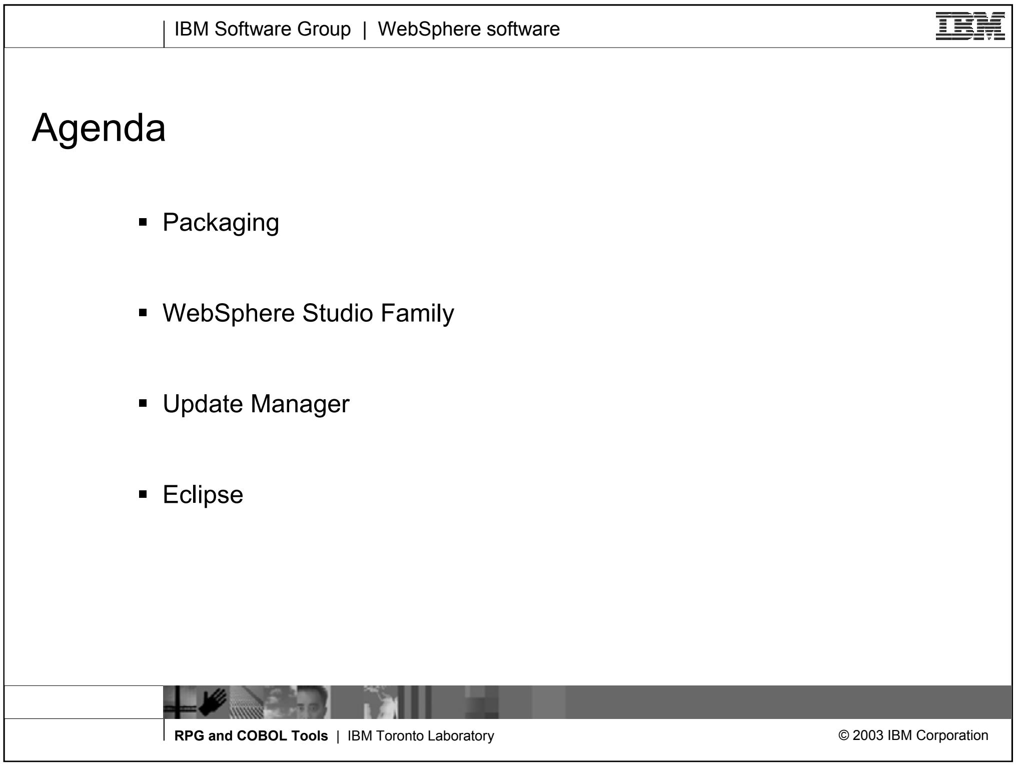
Development Studio Client

Remote System Explorer

Plug-in Development for Vendors

Summary

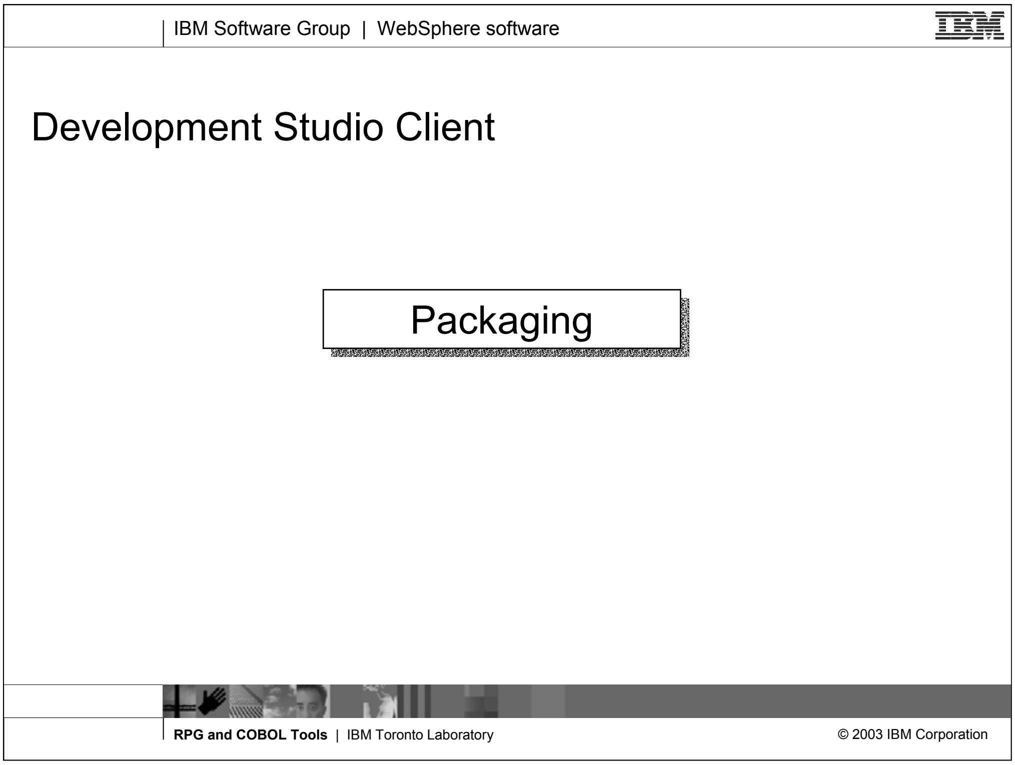
This presentation covers first the strategy behind Development Studio Client and is then followed by a review of the Remote System Explorer in Development Studio Client Version 5.0. The final section reviews how business partners can easily leverage and extend RSE by easily integrating their tools with the IBM base development environment.



Agenda

- Packaging
- WebSphere Studio Family
- Update Manager
- Eclipse

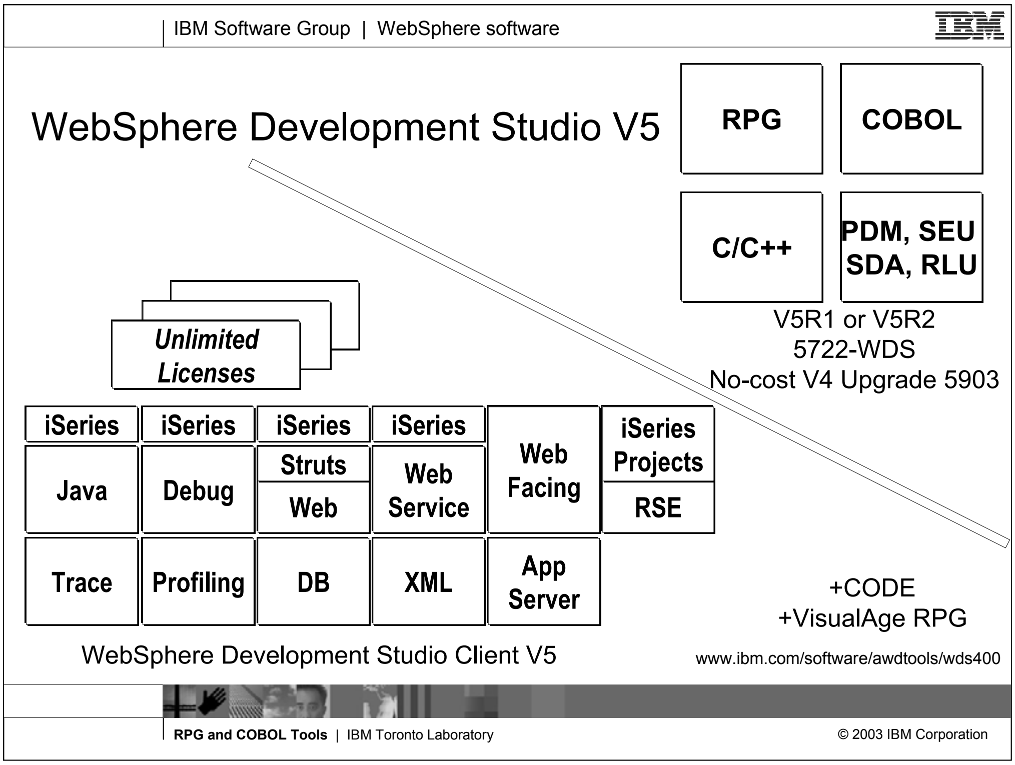
The first section of this presentation covers Development Studio Client packaging, how Development Studio Client fits into the WebSphere family of products, how Update Manager can easily be used to apply the latest product fixes and how the Eclipse technology forms the foundation of the Development Studio Client and all WebSphere products.



Development Studio Client

Packaging

The new packaging was actually done in May 2001, with the introduction of WebSphere Development Studio for iSeries, which as we will see offers a single product with all the host and client tools needed for all application development needs, from traditional to e-business.



IBM Software Group | WebSphere software

WebSphere Development Studio V5

RPG

COBOL

C/C++

PDM, SEU
SDA, RLU

V5R1 or V5R2
5722-WDS
No-cost V4 Upgrade 5903

Unlimited Licenses

iSeries Java

iSeries Debug

iSeries Struts Web

iSeries Web Service

Web Facing

iSeries Projects

RSE

Trace

Profiling

DB

XML

App Server

+CODE
+VisualAge RPG

WebSphere Development Studio Client V5

www.ibm.com/software/awdtools/wds400

RPG and COBOL Tools | IBM Toronto Laboratory

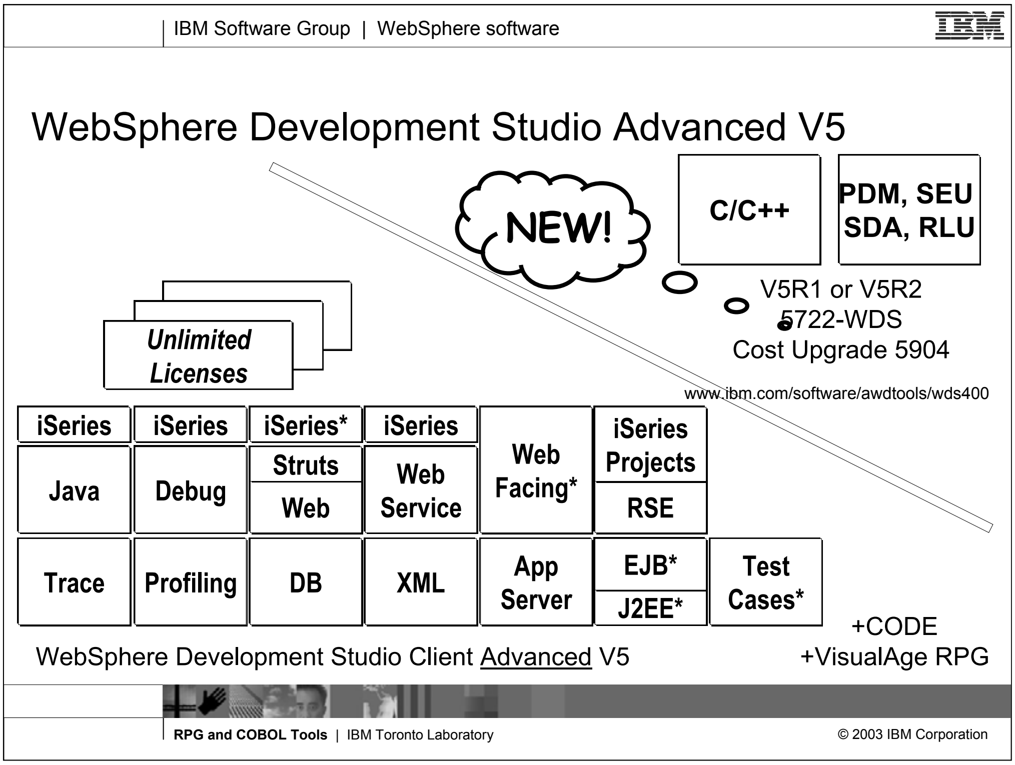
© 2003 IBM Corporation

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

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

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

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

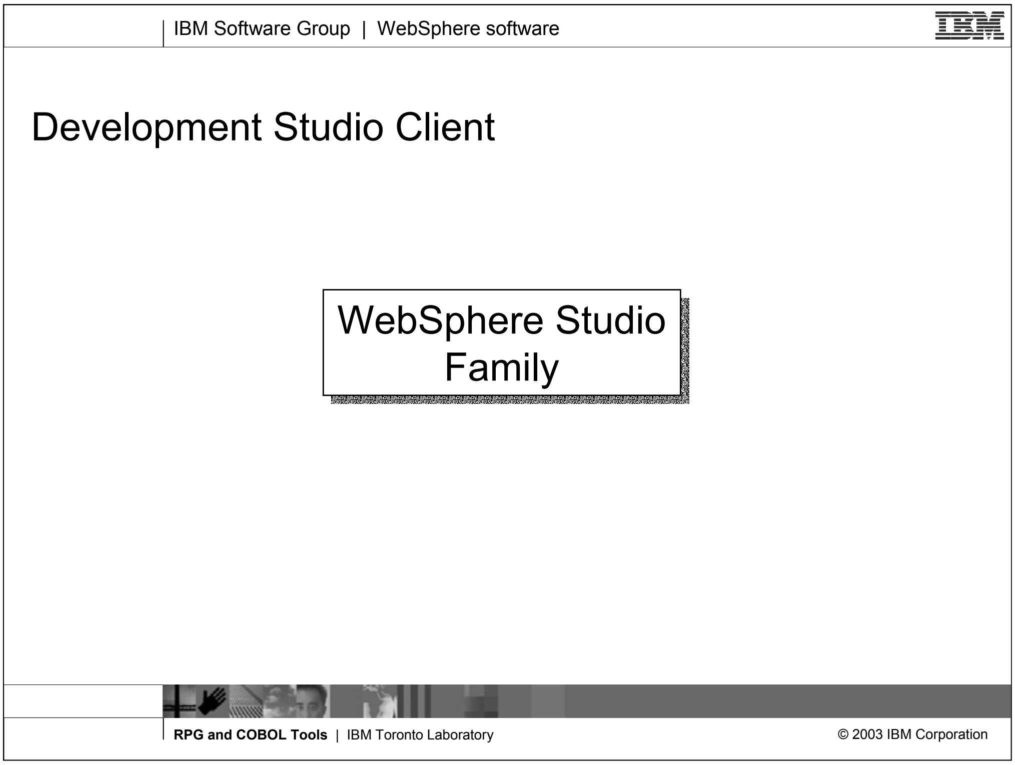


WebSphere Development Studio Advanced V5

WebSphere Development Studio Client Advanced V5

The Advanced edition of Development Studio Client, and Development Studio, is new as of April 25th 2003. Development Studio Advanced is currently the same as Development Studio, except the customer is entitled to unlimited licenses of Development Studio Client Advanced versus just Development Studio Client.

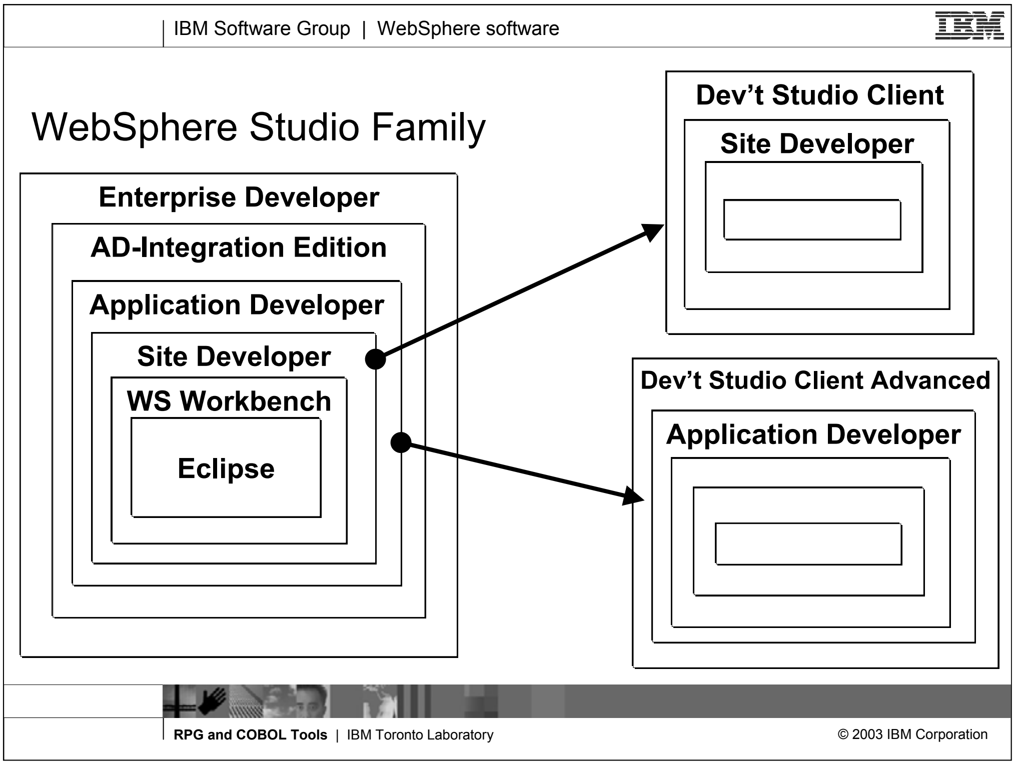
The difference of Development Studio Client Advanced over Development Studio Client is that it has additional tools (blue boxes with asterisks) and some enhancements to existing iSeries tools (green boxes with asterisks) which will be discussed later in this presentation. The majority of the new tools are related to Enterprise Java Bean (EJB) development. You'll see later that Development Studio Client Advanced is based on WebSphere Application Developer (Application Developer) versus WebSphere Studio Site Developer (Site Developer).



Development Studio Client

WebSphere Studio Family

Now you will see how Development Studio Client fits in the WebSphere Studio family of products.



WebSphere Studio Family

Enterprise Developer

AD-Integration Edition

Application Developer

Site Developer

WS Workbench

Eclipse

Dev’t Studio Client

Site Developer

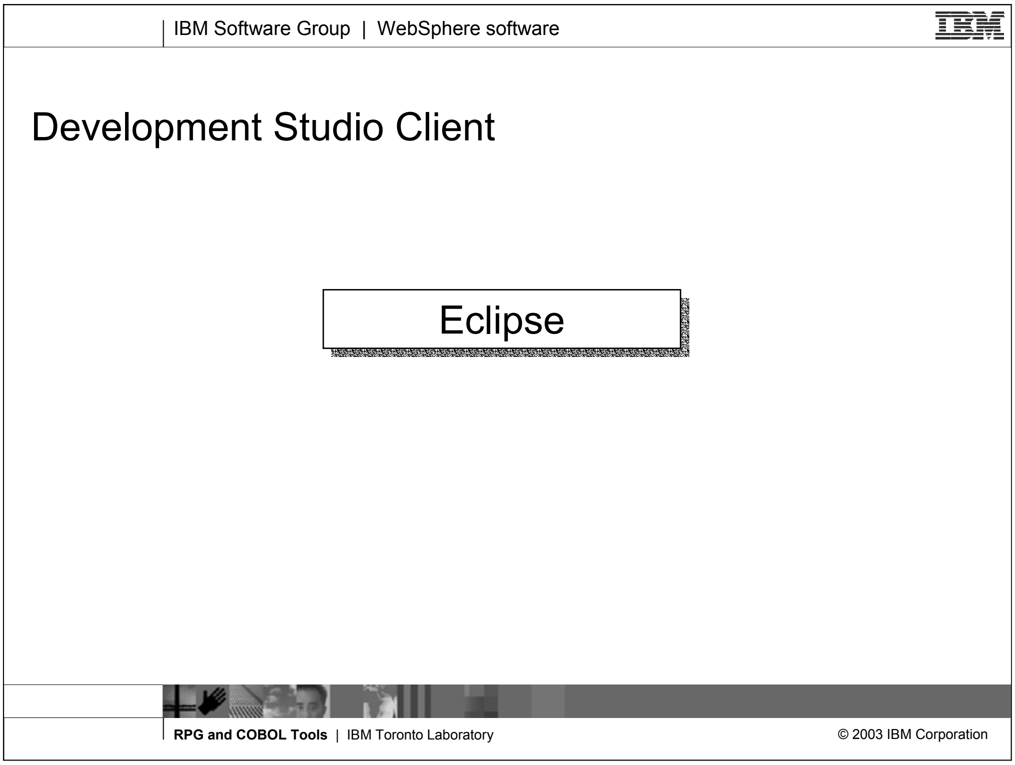
Dev’t Studio Client Advanced

Application Developer

Here you see that Development Studio Client is based on WebSphere Studio Site Developer, while Development Studio Client Advanced is based on WebSphere Studio Application Developer.

The Workbench is based on the open-source Eclipse technology about to be discussed. It is not for sale, but is the basis of all IBM WebSphere Studio products, and is available to business partners.

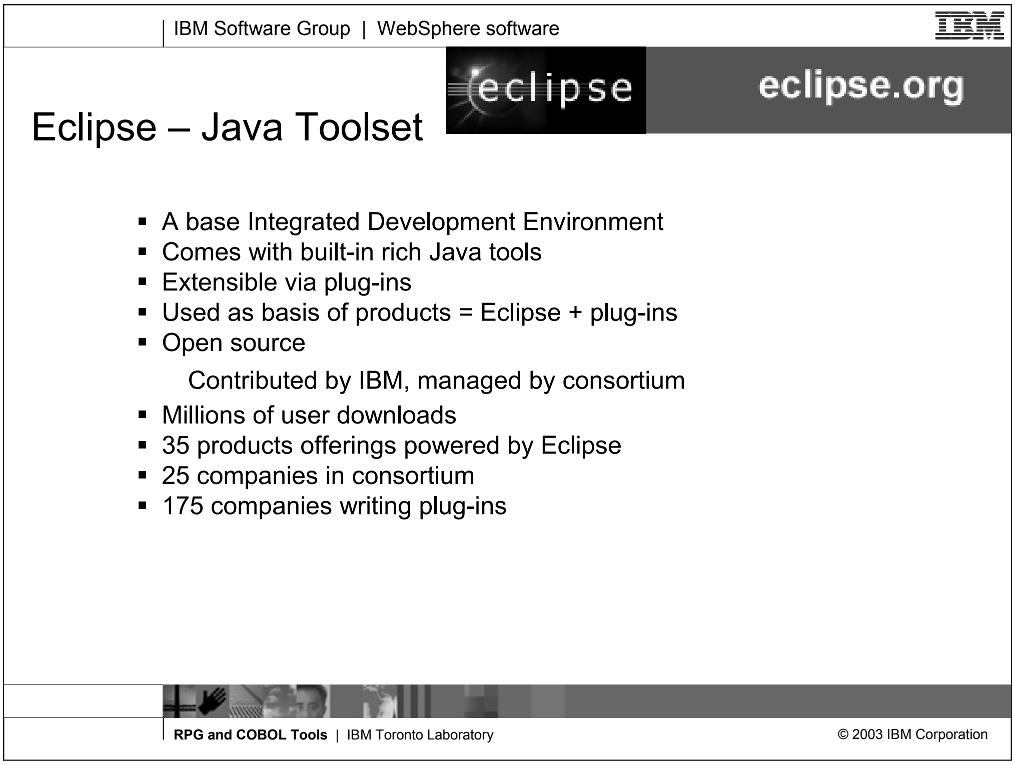
Site Developer is IBM’s entry level offering based on eclipse, and it is for building dynamic Web sites out of non-EJB Java. Application Developer extends Site Developer and adds support for EJBs. Application Developer-Integration Edition extends Application Developer and adds support for JCA Connectors and for Workflow. Enterprise Developer extends Application Developer-Integration Edition and adds support for S/390 and Enterprise Generation Language (EGL), the follow-on to VisualAge Generator.



Development Studio Client

Eclipse

With the 4.0 release of the client tools in June 2002, we introduced new Eclipse technology and totally new Eclipse-based tools. Further, for technology we also plan to keep up with technology both in the compilers and the tools, so iSeries programmers are very current. This was evident in the V5R2 release of the compilers in Sept 2002, and will be again in the V5R3 release of the compilers. With the April 2003 new 5.0 release of the client tools, we are in the second release of the eclipse-based technology and continue to improve the tools and introduce new technology.



eclipse
eclipse.org

Eclipse – Java Toolset

- A base Integrated Development Environment
- Comes with built-in rich Java tools
- Extensible via plug-ins
- Used as basis of products = Eclipse + plug-ins
- Open source

  Contributed by IBM, managed by consortium
- Millions of user downloads
- 35 products offerings powered by Eclipse
- 25 companies in consortium
- 175 companies writing plug-ins

Eclipse was developed by IBM and donated to the open source community. That donation is estimated to be worth $40 million. Anyone can download Eclipse for free, including the source code, from www.eclipse.org. Eclipse has generated extraordinary excitement in the development community and the tools community. It is written in Java, and can be extended by tools that are also written in Java. These tools are known as plug-ins. Out of the box, Eclipse offers an integrated development environment (IDE) that has built-in support for teams and projects and a robust and revolutionary user interface framework. It also has tools built-in to create Eclipse plug-ins. Further, there are extensive and very powerful tools built-in for developing Java applications with Eclipse. So, if all you want is the world's best Java toolset, then all you need is Eclipse. You can't beat the price!

IBM Software Group | WebSphere software

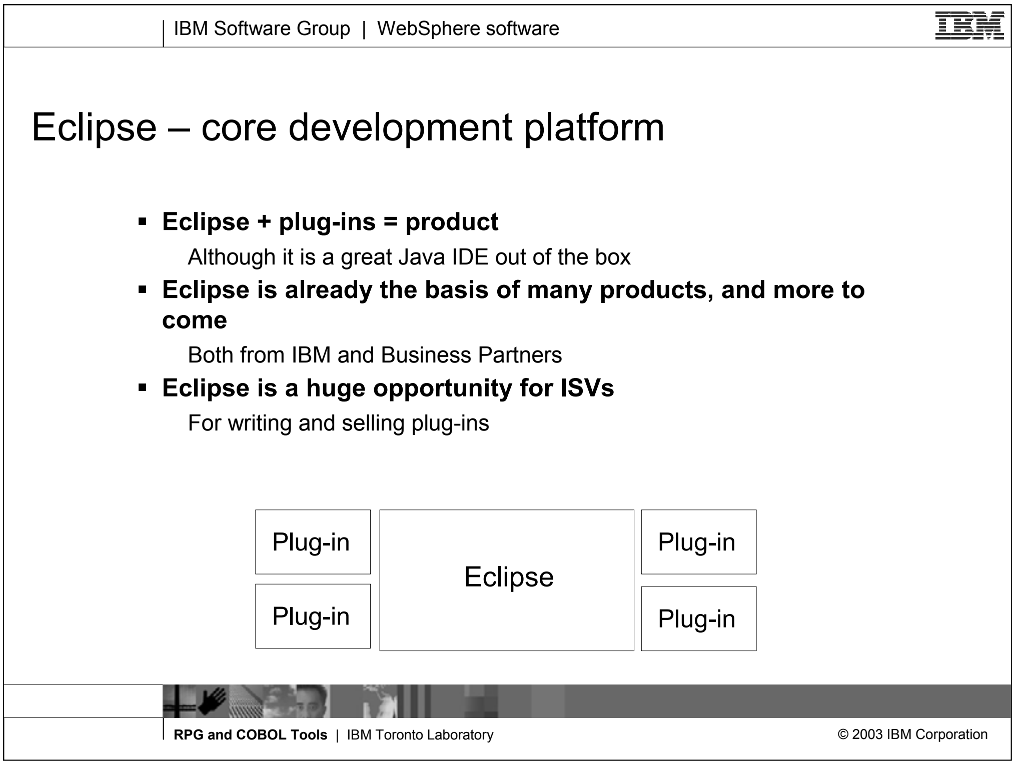
Eclipse Consortium

IBM
Rational the software development company
TimeSys™
Instantiations
Telelogic
PARASOFT
QNX Build a more reliable world
Borland
catalyst SYSTEMS CORPORATION
redhat
ETRI
FLASHLINE
FUJITSU THE POSSIBILITIES ARE INFINITE
SAP
AltoWeb
slickedit.
mks Build Better Software™
HITACHI Inspire the Next
teamstudio Strengthening the platforms business is built on.
Scapa Technologies
MERANT
MONTAVISTA SOFTWARE
TRANS-ENTERPRISE
serena™ Automating Change to Enterprise Code and Content
hp invent
OMG OBJECT MANAGEMENT GROUP
SuSE
SYBASE
TogetherSoft

RPG and COBOL Tools | IBM Toronto Laboratory

© 2003 IBM Corporation

IBM is not alone with Eclipse. The open-source consortium that oversees contributions made to it include a number of large companies, and that list is growing. Note that MKS is an iSeries tool vendor. See www.eclipse.org for the latest list.



Eclipse – core development platform

- **Eclipse + plug-ins = product**
  Although it is a great Java IDE out of the box
- **Eclipse is already the basis of many products, and more to come**
  Both from IBM and Business Partners
- **Eclipse is a huge opportunity for ISVs**
  For writing and selling plug-ins

Plug-in
Plug-in
Eclipse
Plug-in
Plug-in

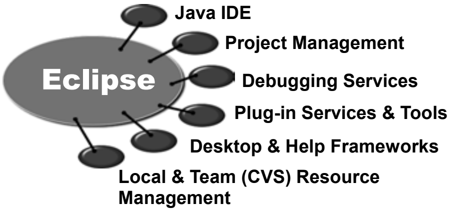
So what is Eclipse? With the exception of the built-in Java tools, Eclipse itself is not that interesting. It only gets interesting when you add to it some plug-ins (for example, tools) that do something interesting.

An Eclipse-based product is Eclipse plus a number of interesting plug-ins. IBM is building numerous such Eclipse-based products, including as we have seen Site Developer and Development Studio Client. Further, because Eclipse is free and business partners are free to include it in their products, there will be many other Eclipse-based products from other companies too, including Rational and TogetherSoft.

For business partners or software developers who write and sell application development tools, Eclipse is a fantastic opportunity. By writing plug-ins for Eclipse, those plug-ins can be sold to any developer using any product based on Eclipse or even just the raw Eclipse as downloaded from www.eclipse.org. This opportunity is not lost on iSeries tool vendors, who are all looking at offering Eclipse plug-ins for their tools. This will result in a rich offering of third party plug-ins for developers to choose from, all of which extend their core Development Studio Client development environment. One community, one core development platform, many IBM and 3rd party tools. This is community and excitement!

Eclipse Plug-in Development

Products built with Eclipse inherit these capabilities plus 'plug-ins' built by others



Eclipse

Java IDE

Project Management

Debugging Services

Plug-in Services & Tools

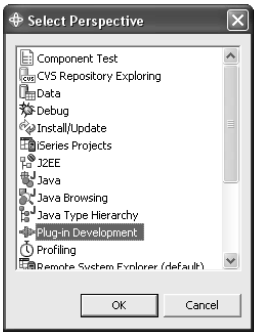
Desktop & Help Frameworks

Local & Team (CVS) Resource Management

Some of the common services Eclipse supplies include: the Java integrated development environment (IDE), project management, debug, plug-in frameworks, desktop and help frameworks, and resource managers.

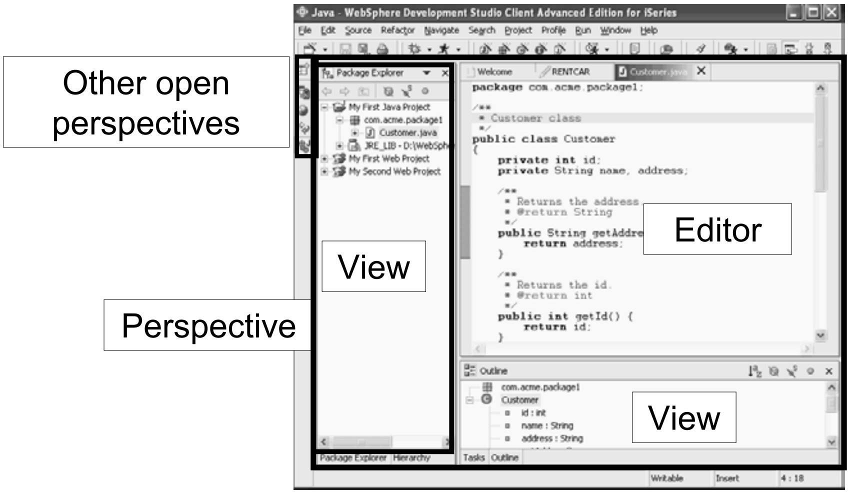
Eclipse User Interface – Select Perspective

- Users work with **perspectives**
  - –Collection of **views** and **editors**
    - –Tools for a particular task
    - –Allows for role-based development
  - –Many perspectives are pre-supplied for specific tasks like Java, Web, XML, RPG/CBL
  - –Users can create their own perspectives
- The user interface is very Windows-like
  - –Build from a Java wrapper of OS widgets
  - –Behaves and feels like any other native application
  - –Eg: views can be re-sized and re-positioned through drag ‘n drop



The core features of the Eclipse user interface include perspectives which is a collection of views and tools. Perspectives allow role based development. For example, if you are a Java developer you would use the Java perspective which includes tools and views for Java development. You can also create your own perspectives. Naturally, the Eclipse user interface applies to all Eclipse-based products, like Development Studio Client.

Eclipse User Interface – Open Perspective



```
package com.acme.package1;

/**
 * Customer class
 */
public class Customer
{
    private int id;
    private String name, address;

    /**
     * Returns the address.
     * @return String
     */
    public String getAddre
        return address;
    }

    /**
     * Returns the id.
     * @return int
     */
    public int getId() {
        return id;
    }
```

Other open perspectives

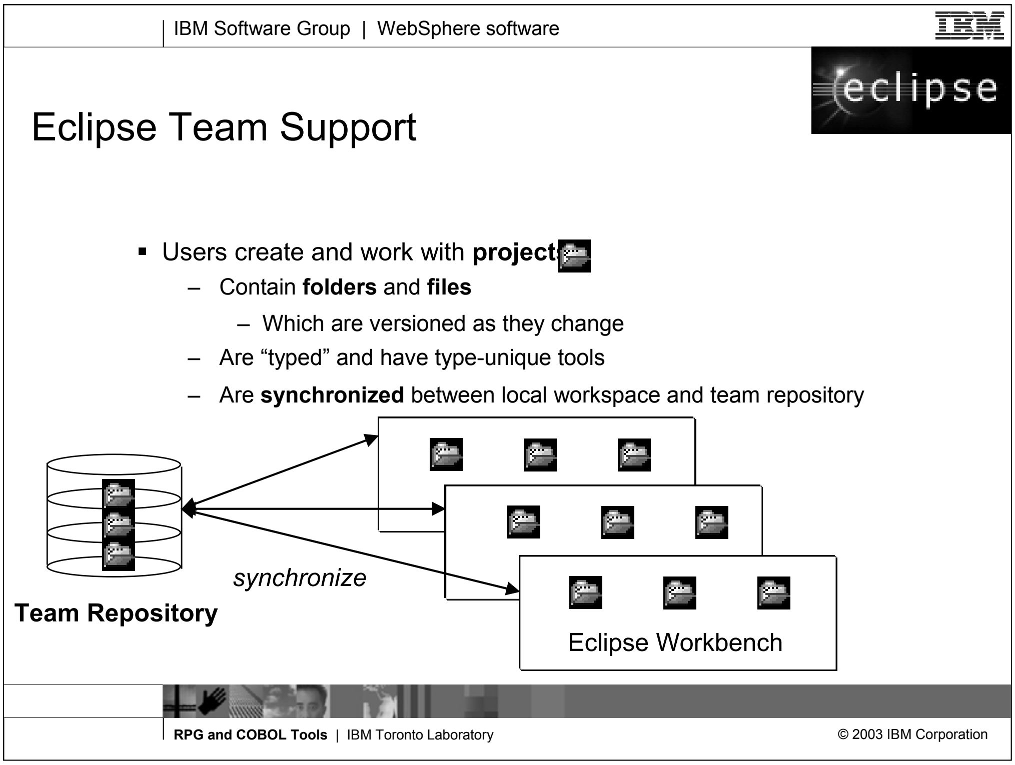
View

Editor

Perspective

View

The Eclipse workbench has four open perspectives. You can see them lined up on the left frame of the workbench. The current active perspective is the one whose icon is indented, which in this case is the Java perspective. To open a new perspective, you use Window->Open Perspective giving you the Select Perspective dialog that you saw previously. This particular perspective has a Package Explorer view which drives the Editor and the Outline view.



Eclipse Team Support

- Users create and work with **projects**
  - Contain **folders** and **files**
    - Which are versioned as they change
  - Are “typed” and have type-unique tools
  - Are **synchronized** between local workspace and team repository

synchronize

Team Repository

Eclipse Workbench

Central to the Eclipse integrated development environment is support for projects. Projects are simply a grouping of folders and files.

Plug-ins have types, such as Java or Web or XML. Tools that plug into Eclipse can define their own new project types. Tools that plug into Eclipse work against resources (folders or files) within projects. They can be scoped to projects of a particular type, if appropriate.

All projects, regardless of type, have common behavior and support. This includes the ability for a team to share a project by using an Eclipse-supporting software change management (SCM) product such as Concurrent Version System (CVS) or Rational Clearcase. This SCM acts as a central repository for one or more projects. Each team member can easily keep their local copy of any project "in synch" with the central repository copy. CVS is a free open-source SCM. It runs on Linux, UNIX and Windows. It comes on the Linux distribution CDs for iSeries Linux LPAR.

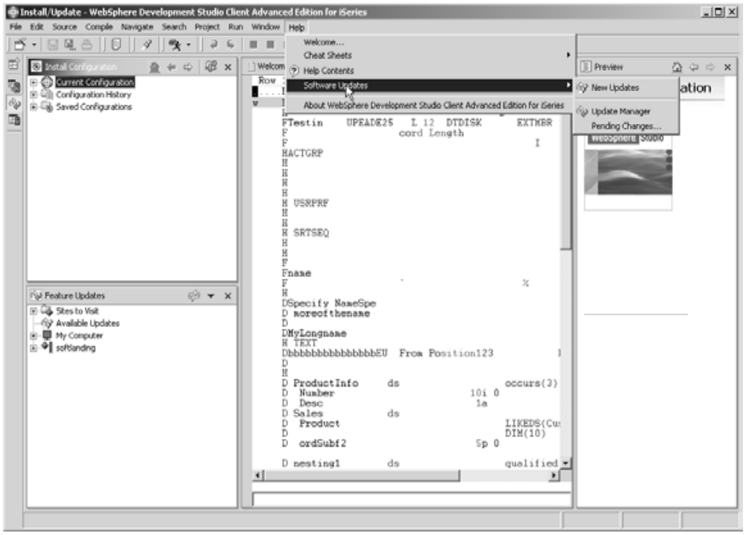
All IBM SCM vendors for iSeries are enabling, or have enabled, their products to Eclipse.

Development Studio Client

Update Manager

Update Manager is available from the product Help pull down menu so you can access the latest product fixes.

Update Manager – Software updates



```
FTestin    UPEADE25   L 12  DTDISK      EXTMBR
F                    cord Length
F                                        I
HACTGRP
H
H
H
H
H USRPRF
H
H
H SRTSEQ
H
H
F
Fname
F                                       %
H
DSpecify NameSpe
D moreofthename
D
DMyLongname
H TEXT
DbbbbbbbbbbbbbbbEU  From Position123
D
H
D ProductInfo     ds                   occurs(3)
D  Number                       10i 0
D  Desc                          1a
D Sales           ds
D  Product                             LIKEDS(Cu
D                                      DIM(10)
D  ordSubf2                      5p 0

D nesting1        ds                   qualified
```

You can quickly and easily update your product. Use Help > Software Updates > Update Manager

You will be presented with information on how to check for feature updates, how to install new features, how to work with installed features and how to review the history of installed updates.

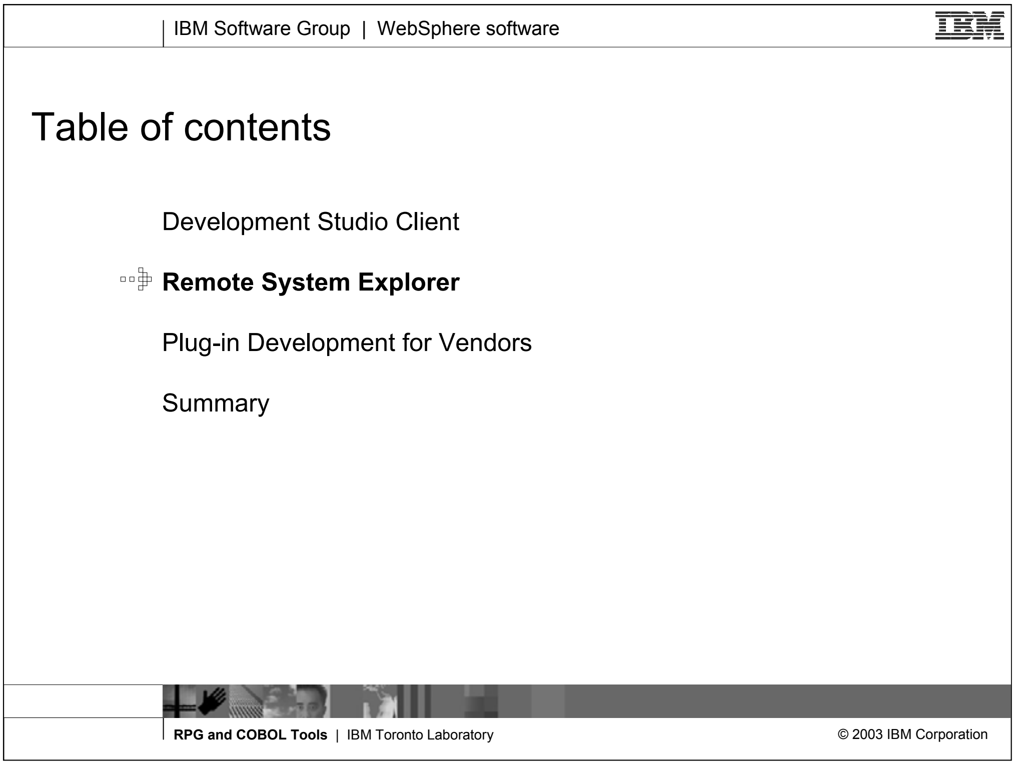


Table of contents

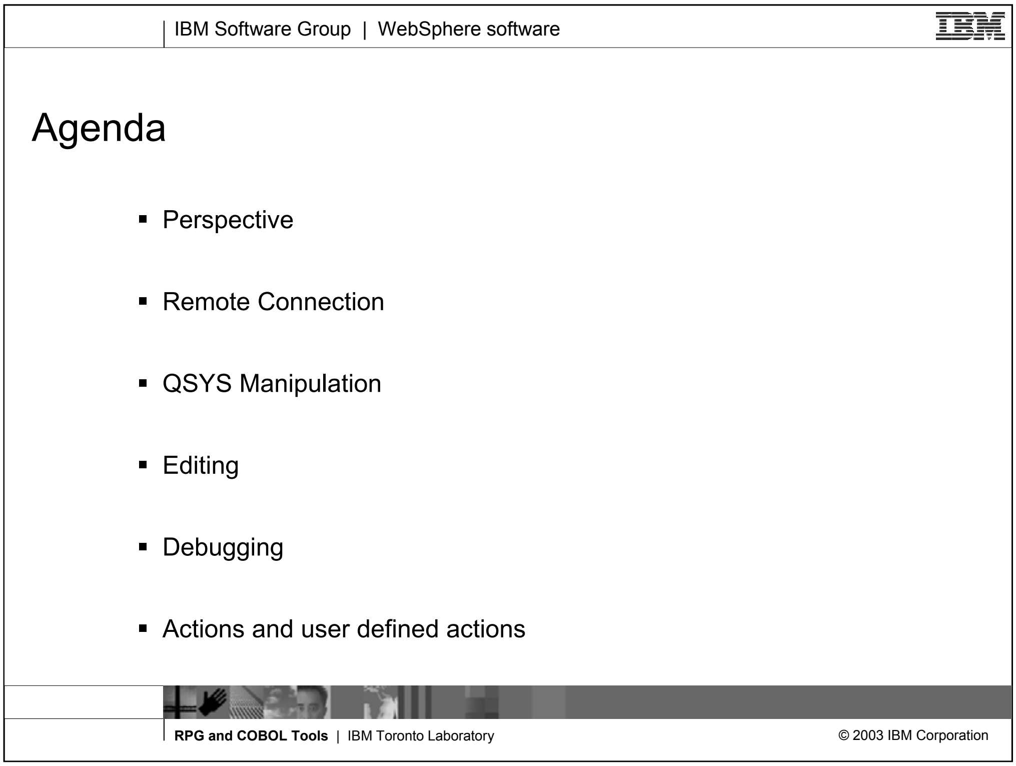
Development Studio Client

Remote System Explorer

Plug-in Development for Vendors

Summary

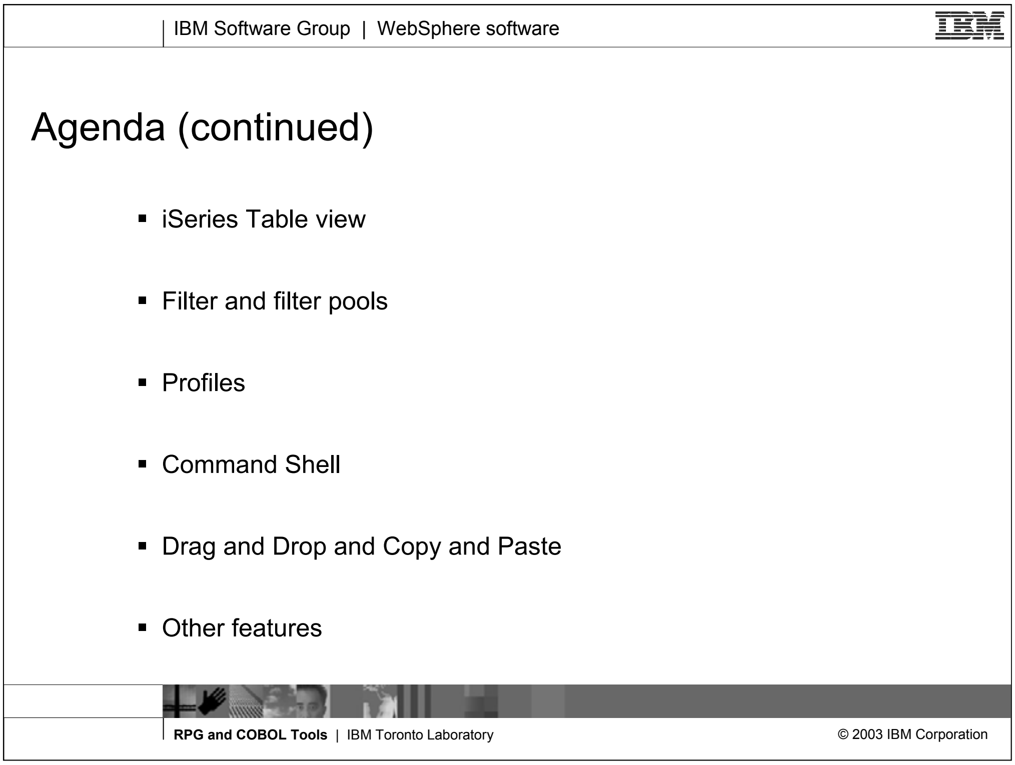
Now you know the story behind Development Studio Client. Lets look at the Remote System Explorer, the tool for iSeries application development and maintenance.



Agenda

- Perspective
- Remote Connection
- QSYS Manipulation
- Editing
- Debugging
- Actions and user defined actions

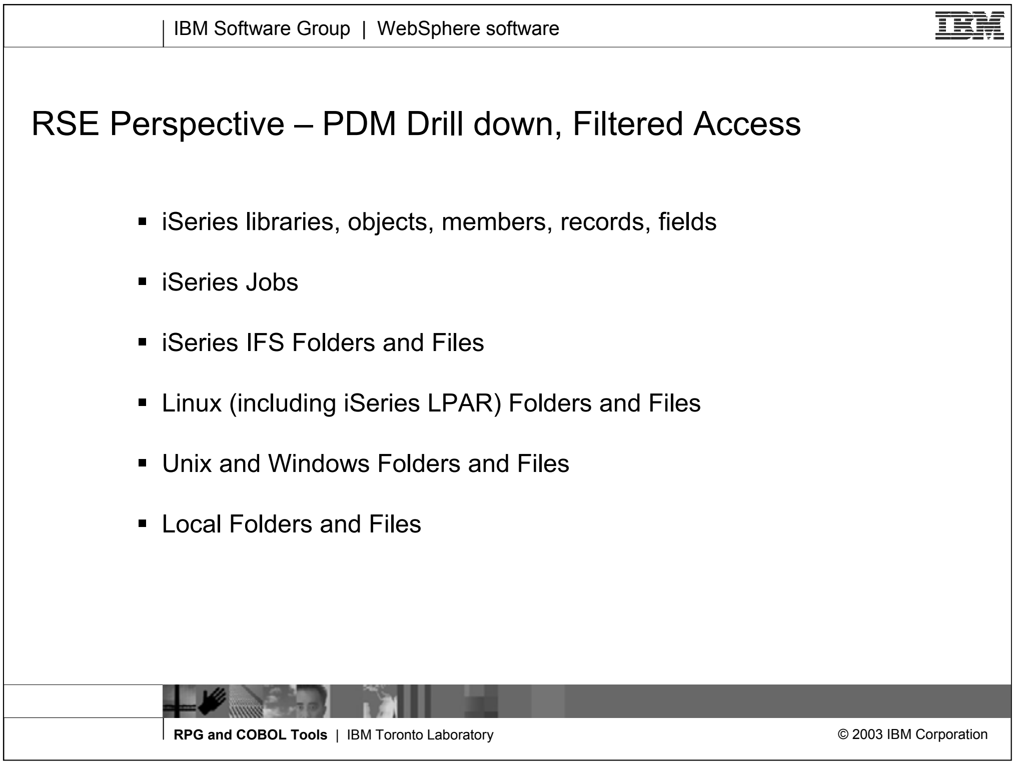
The second section of this presentation covers the Remote Systems Explorer(RSE), which has its own perspective and views. It is similar to PDM in that it allows you to drill down into the QSYS file system, or use filters to list specific objects within the QSYS file system. The Remote Systems Explorer goes well beyond PDM however! It also allows exploration of iSeries jobs and commands, and the IFS file system. Further, you can use RSE to explore the file system of remote Linux, UNIX and Windows systems. The Linux support works for any Linux, including Linux in an iSeries Logical Partition. You can also launch the built-in editor and debugger from RSE! You can create and manage your own user-defined actions just like in PDM and have them appear in the pop up menus.



Agenda (continued)

- iSeries Table view
- Filter and filter pools
- Profiles
- Command Shell
- Drag and Drop and Copy and Paste
- Other features

Instead of the Remote Systems view which is a tree view, PDM users are use to a table, so there is an iSeries Table view that shows what the tree shows but in table format. There are right-click actions that are the same as PDM's and a command line at the bottom of the table just like PDM. To get to the iSeries objects you want to work with you need to create your own filters and when you have many filters you will want to group them into filter pools. You give your filters a name because the Remote System Explorer saves them for future use, in opposition to PDM, which does not save filters. You can set up profiles to share connections, filters, user-defined actions, compile commands across a team of developers. The remote commands interface allows you to submit requests to the iSeries to invoke actions like compile, bind, or build objects on the host. Extensive support for dragging and dropping (Programmers can use drag and drop to copy source members between files on the same iSeries, or different iSeries. They can even copy members to and from their local Windows file system, or a remote UNIX, Windows, or Linux file system.)



RSE Perspective – PDM Drill down, Filtered Access

- iSeries libraries, objects, members, records, fields
- iSeries Jobs
- iSeries IFS Folders and Files
- Linux (including iSeries LPAR) Folders and Files
- Unix and Windows Folders and Files
- Local Folders and Files

There are two explicit areas of functionality, each with their own perspective, for iSeries programmers.

The first area of functionality is the Remote Systems Explorer, which has its own perspective and views.

This first area of functionality is similar to PDM in that it allows the developer to drill down into the QSYS file system, or use filters to list specific objects within the QSYS file system

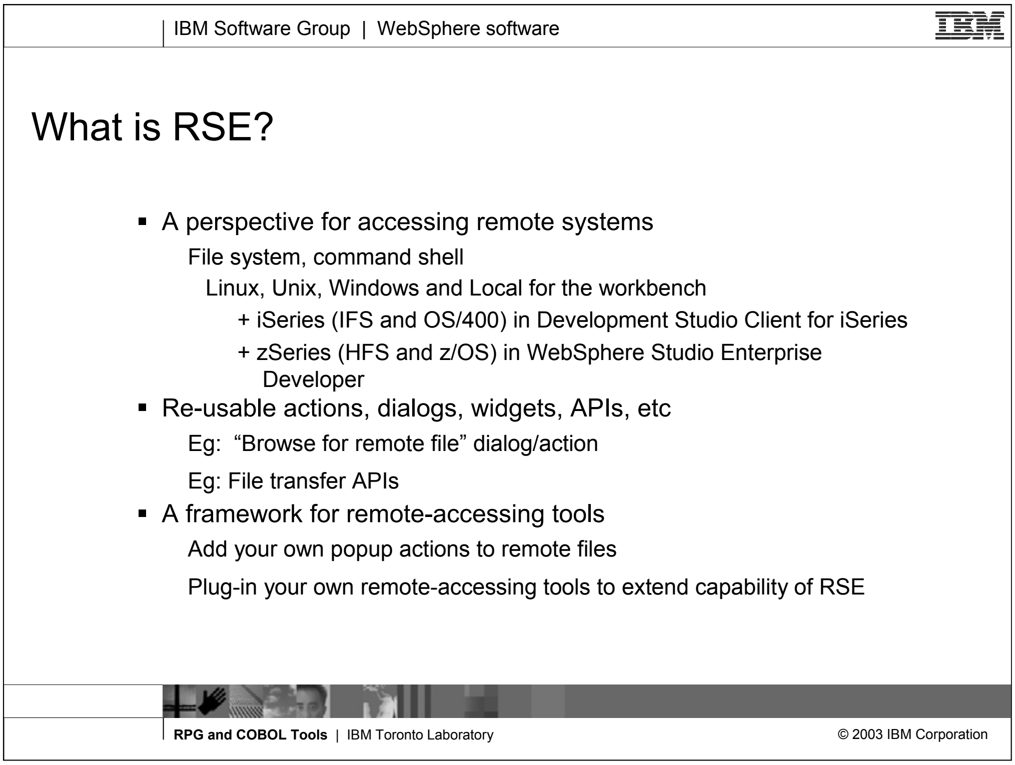
The Remote Systems Explorer goes well beyond PDM however! It also allows exploration of iSeries jobs and commands, and the IFS file system. Further, it can also be used to explore the file system of remote Linux, Unix and Windows systems. The Linux support works for any Linux, including Linux in an iSeries Logical Partition.

The second area of functionality is iSeries Projects, which also has its own perspective and views.

Unlike the Remote Systems Explorer, an iSeries project fully leverages the Eclipse support for resources. This means an iSeries project contains folders and files that exist in the local file system, yet can be shared and synchronized among a team, if an Eclipse-compliant repository is used. The options for this today include Rational's ClearCase or the free and open-source product CVS or Concurrent Versioning Systems, or the MKS Integrity. CVS is available on the distribution CDs for the Linux LPAR on iSeries. In the near future, the remaining iSeries change management vendors will also be available as repository options.

As you will see, an iSeries project allows RPG and COBOL developers to fully exploit the power of Eclipse, while developing applications targeted to run on iSeries.

We will explore each of these tools...



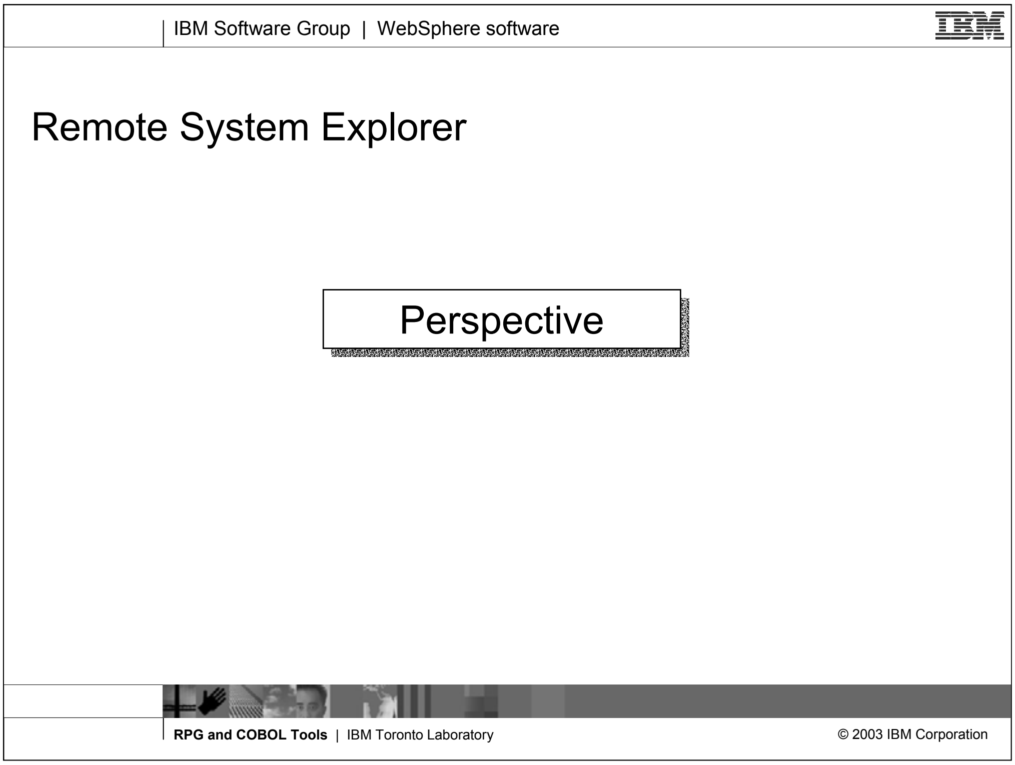
What is RSE?

- A perspective for accessing remote systems
  - File system, command shell
    - Linux, Unix, Windows and Local for the workbench
      - + iSeries (IFS and OS/400) in Development Studio Client for iSeries
      - + zSeries (HFS and z/OS) in WebSphere Studio Enterprise Developer
- Re-usable actions, dialogs, widgets, APIs, etc
  - Eg: “Browse for remote file” dialog/action
  - Eg: File transfer APIs
- A framework for remote-accessing tools
  - Add your own popup actions to remote files
  - Plug-in your own remote-accessing tools to extend capability of RSE

The Remote System Explorer is an Eclipse perspective with many views to help developers work with resources in a remote system. From here, you can manipulate, edit, create, delete and compile programs remotely.

Think of RSE As…

- Extending Eclipse to support direct access to remote resources:
  - Like an Eclipse Windows Explorer for remote files
  - Like an Eclipse MS-DOS view for remote commands
  - Using direct drill-down access. Not projects
- Extending java.io.File to work with remote files
  - Allows easy programmatic access to remote file system from Eclipse plug-ins

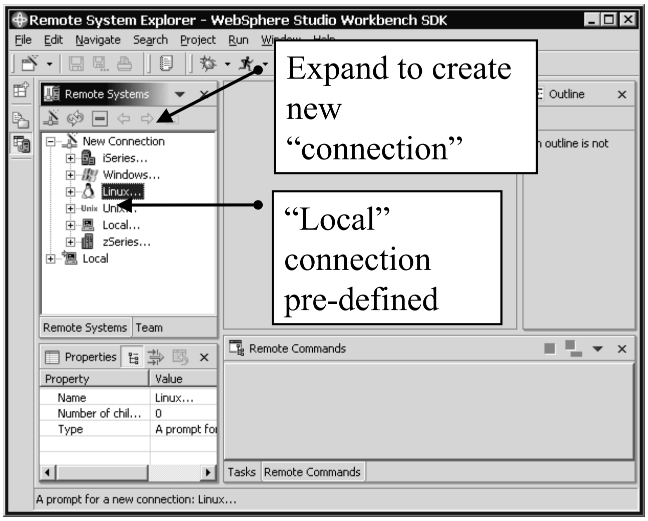


Remote System Explorer

Perspective

A perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors. Each perspective provides a set of capabilities aimed at accomplishing a specific type of task or working with specific types of resources. For example a Web developer will use the Web perspective, a Java developer will use the Java perspective, an iSeries developer will use the Remote Systems Explorer perspective.

RSE Perspective



Property	Value
Name	Linux...
Number of chil...	0
Type	A prompt fo

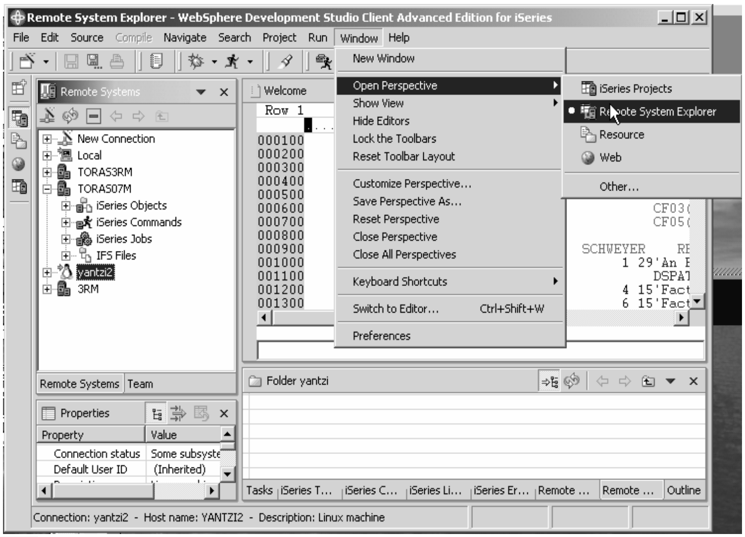
A prompt for a new connection: Linux...

Expand to create new "connection"

"Local" connection pre-defined

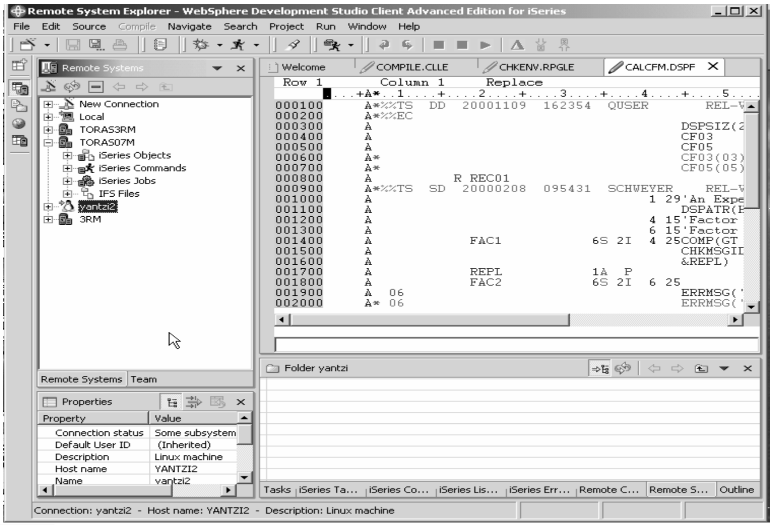
When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on our workstation. To connect to a remote iSeries host, you need to define a connection. In the Remote Systems view you use the New Connection wizard to configure a connection to your iSeries host. You can also use the Remote Systems view to explore the file system of remote Linux, UNIX and Windows systems. The Local connection is pre-defined.

How do I start it?



The Remote Systems Explorer perspective is the first perspective you see when starting Development Studio Client for the first time. If its not open you can open the perspective by selecting Window > Open Perspective > Remote System Explorer. Remember to check the left frame of the workbench for the RSE icon so that you don't open another Remote System Explorer perspective.

RSE Subsystems…



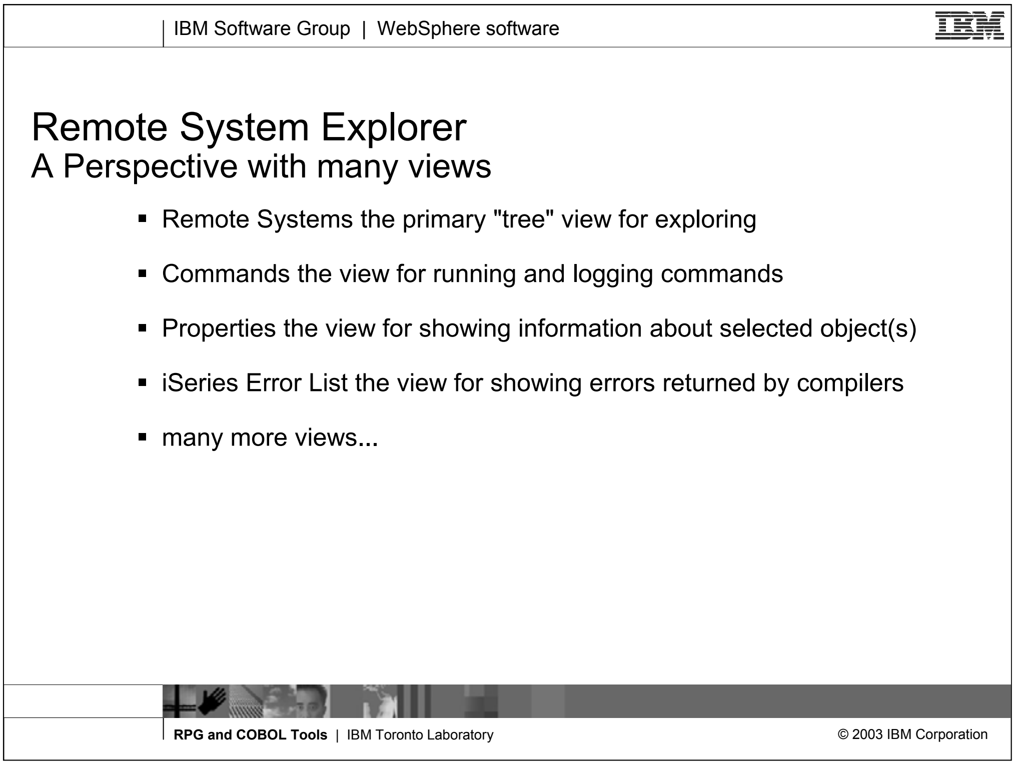
```
 Row 1          Column 1        Replace
           . . . .+A*. .1 . . . .+. . . .2 . . . .+. . . .3 . . . .+. . . .4 . . . .+. . . .5 . . . .
000100          A*%%TS   DD   20001109   162354   QUSER        REL-V
000200          A*%%EC
000300          A                                               DSPSIZ(2
000400          A                                               CF03
000500          A                                               CF05
000600          A*                                              CF03(03)
000700          A*                                              CF05(05)
000800          A                 R REC01
000900          A*%%TS   SD   20000208   095431   SCHWEYER     REL-V
001000          A                                           1 29'An Expe
001100          A                                               DSPATR(H
001200          A                                           4 15'Factor
001300          A                                           6 15'Factor
001400          A                   FAC1             6S 2I   4 25COMP(GT
001500          A                                               CHKMSGID
001600          A                                               &REPL)
001700          A                   REPL             1A   P
001800          A                   FAC2             6S 2I   6 25
001900          A  06                                           ERRMSG('
002000          A* 06                                           ERRMSG('
```

Property	Value
Connection status	Some subsystem
Default User ID	(Inherited)
Description	Linux machine
Host name	YANTZI2
Name	yantzi2

Connection: yantzi2 - Host name: YANTZI2 - Description: Linux machine

Once connections are defined they can be expanded within the Remote Systems Explorer. On expansion, you see subsystems, which are a functional grouping of the various types of remote resources that can be explored in the remote system. For iSeries connections, there are four subsystems:

1. iSeries Objects is the PDM-like grouping, allowing access to libraries, objects and members
2. iSeries Commands allows you to predefine command sets each of which contain one or more often used commands. When run, all commands in a command set are sent to the remote system and executed, and the results are logged in the Commands view
3. iSeries Jobs allows you to see various jobs, subset by job attributes, and to perform a limited number of operations on those jobs
4. IFS Files allows you to explore folders and files in the Integrated File System of the remote iSeries system



Remote System Explorer
A Perspective with many views

- Remote Systems the primary "tree" view for exploring
- Commands the view for running and logging commands
- Properties the view for showing information about selected object(s)
- iSeries Error List the view for showing errors returned by compilers
- many more views...

The Remote System Explorer is an Eclipse perspective with many views to help you work with resources in a remote system

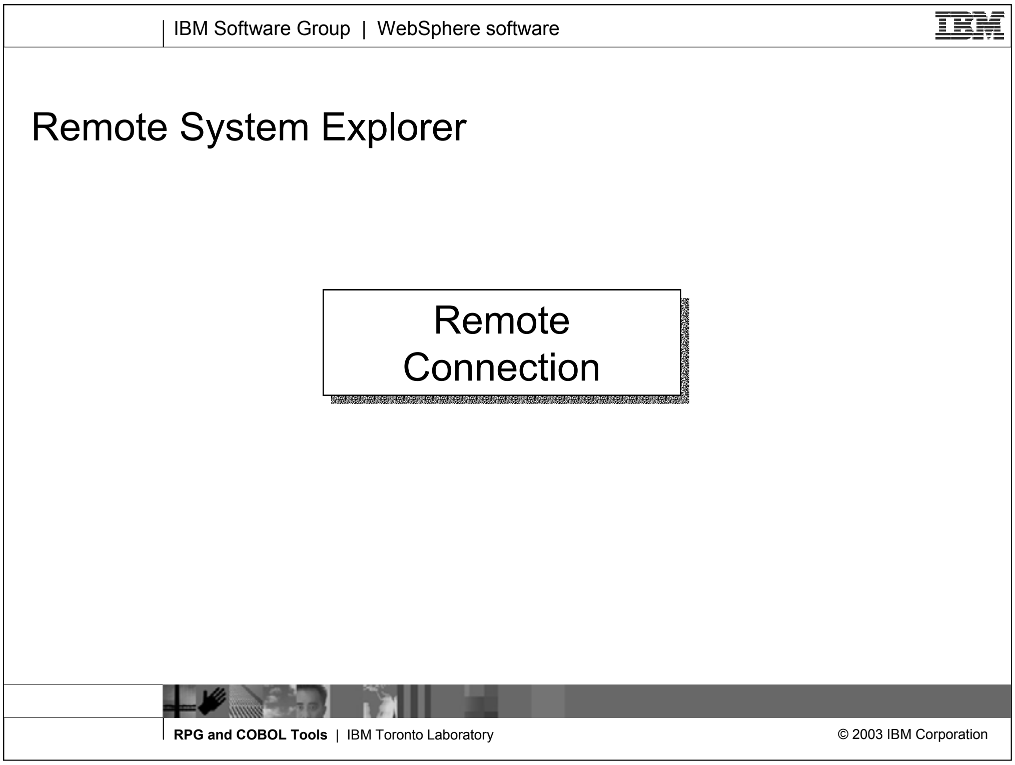
The primary view is the Remote Systems view, where remote system resources are explored, similar to the Windows Explorer.

There is a Commands view, or shell, for entering commands to be run remotely, and for logging the results of all commands.

There is a Properties view, which is common throughout Eclipse. It shows interesting information about the object currently selected in the primary view.

There is an iSeries Error List view where errors are shown after performing a remote compile.

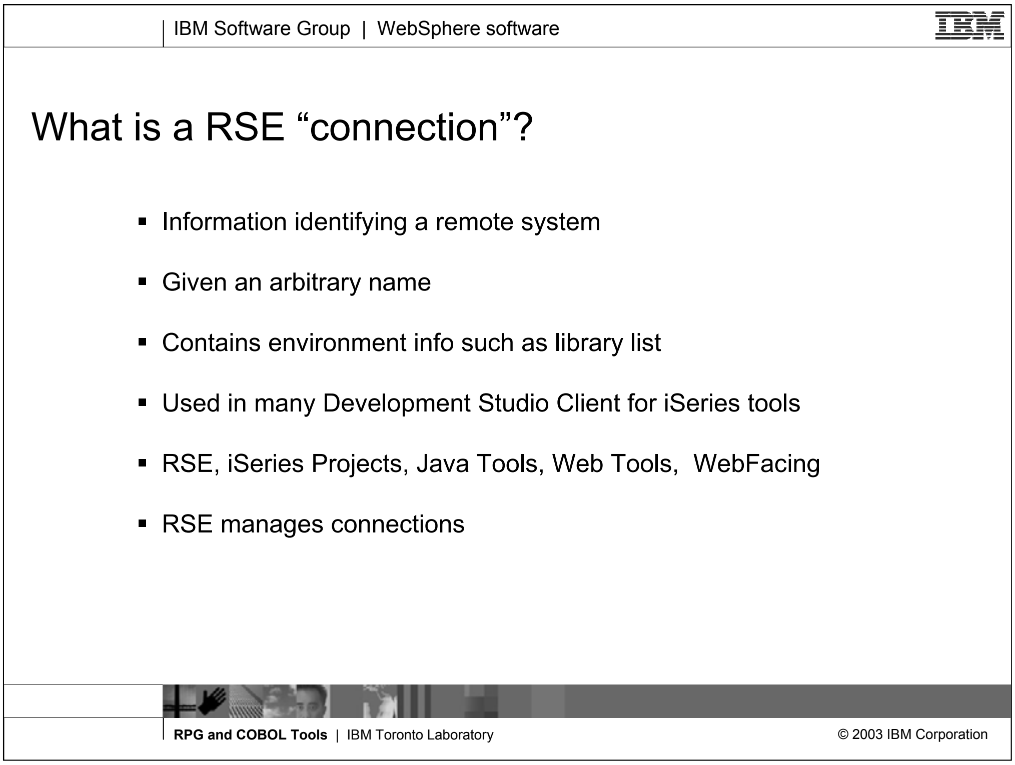
There are many other views which will appear when requested through pop-up menu actions.



Remote System Explorer

Remote Connection

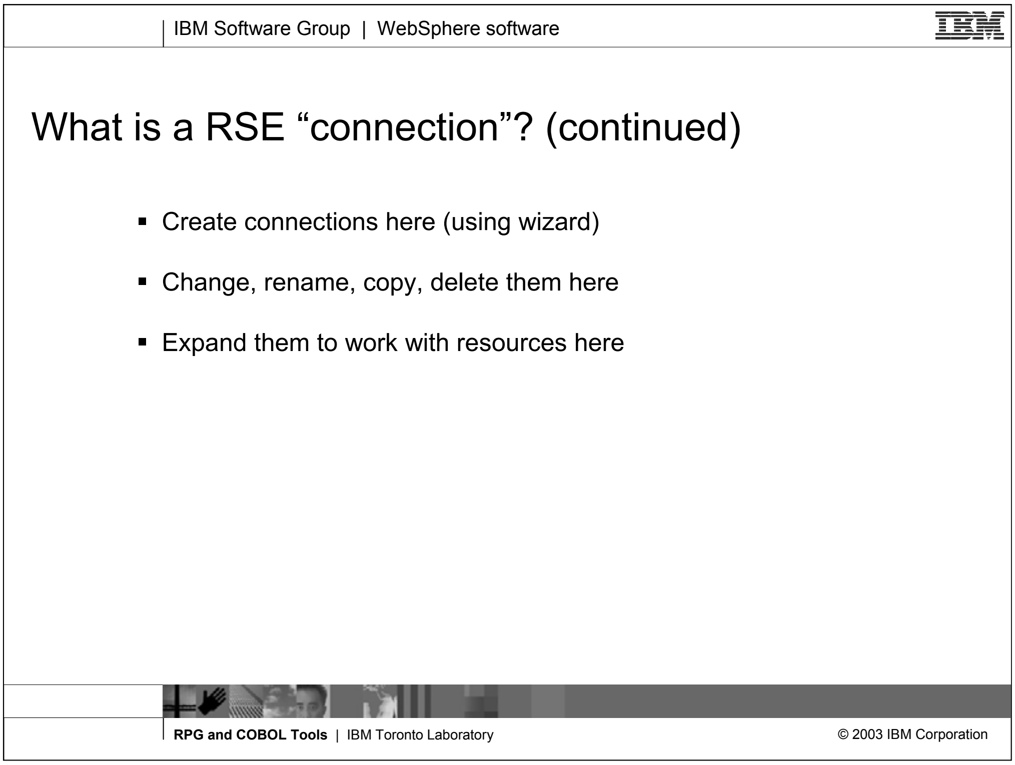
When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on our workstation. To connect to a remote iSeries host, you need to define a connection.



What is a RSE “connection”?

- Information identifying a remote system
- Given an arbitrary name
- Contains environment info such as library list
- Used in many Development Studio Client for iSeries tools
- RSE, iSeries Projects, Java Tools, Web Tools, WebFacing
- RSE manages connections

A very central concept to all of Development Studio Client for iSeries is that of connections. A connection defines information needed to access a remote system. Each connection is given an arbitrary name by you, and so multiple connections to the same system are permitted. Each connection also captures information that is applied when connecting to that remote system, such as the initial library list for iSeries connections. All iSeries tools within Development Studio Client for iSeries use connections to access a remote iSeries system. Connections are created and managed in the Remote Systems Explorer.

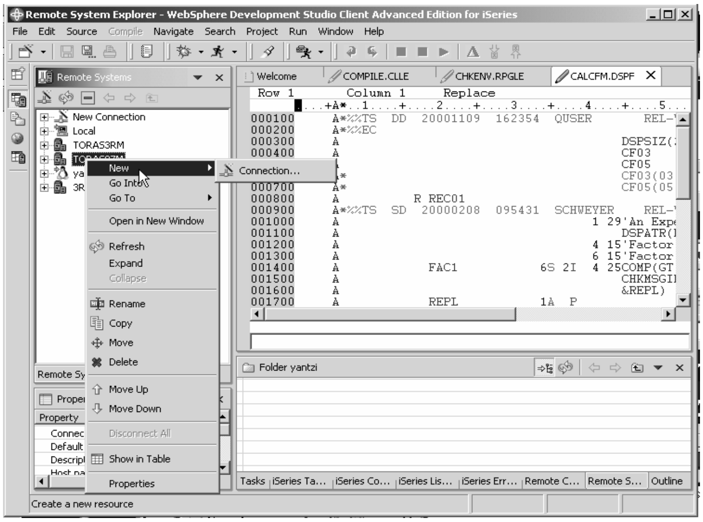


What is a RSE “connection”? (continued)

- Create connections here (using wizard)
- Change, rename, copy, delete them here
- Expand them to work with resources here

Further, the Remote System Explorer is also used to explore objects in a remote system, by expanding a connection.

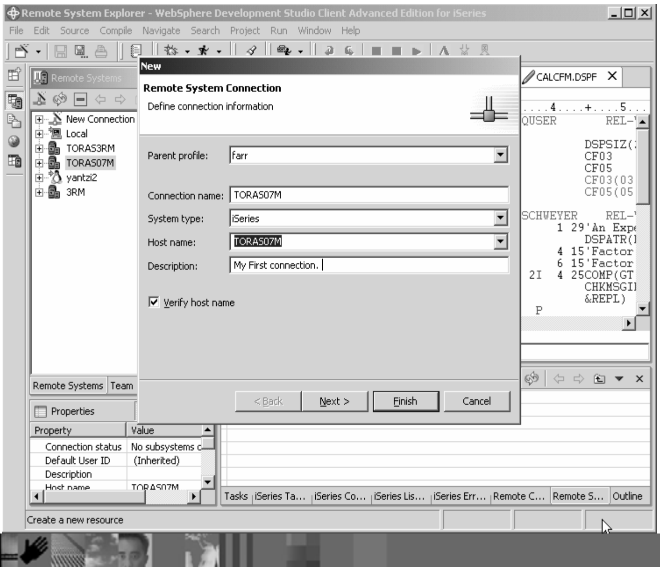
How do I create a connection?



```
 Row 1          Column 1       Replace
        ....+A*..1....+....2....+....3....+....4....+....5...
000100       A*%%TS  DD  20001109  162354  QUSER       REL-
000200       A*%%EC
000300       A                                        DSPSIZ(
000400       A                                        CF03
                                                      CF05
              *                                       CF03(03
000700       A*                                       CF05(05
000800       A          R REC01
000900       A*%%TS  SD  20000208  095431  SCHWEYER    REL-
001000       A                                   1 29'An Expe
001100       A                                        DSPATR(
001200       A                                   4 15'Factor
001300       A                                   6 15'Factor
001400       A            FAC1          6S 2I    4 25COMP(GT
001500       A                                        CHKMSGI
001600       A                                        &REPL)
001700       A            REPL          1A  P
```

The New Connection wizard prompts for information about the connection, including an arbitrary but unique name, the type of the remote system, the TCP/IP hostname for the system, the User ID to connect to the system with, and optionally a description of the connection.

Create your first connection !



Remote System Connection
Define connection information

Parent profile: farr

Connection name: TORAS07M
System type: iSeries
Host name: TORAS07M
Description: My First connection.

☑ Verify host name

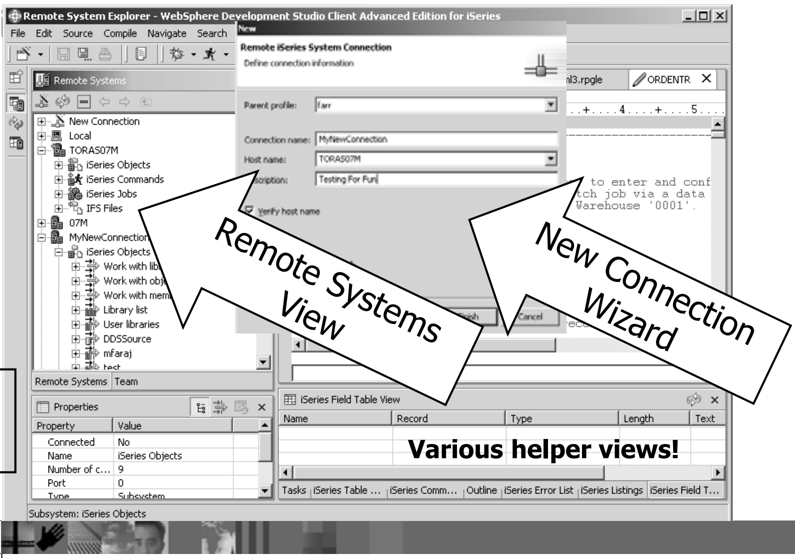
< Back | Next > | Finish | Cancel

Property	Value
Connection status	No subsystems c
Default User ID	(Inherited)
Description	
Host name	TORAS07M

Create a new resource

In the Remote Systems view you use the New Connection wizard to configure a connection to your iSeries host. If this was the first time you connect to an iSeries host, you would need to define a profile. All connections, filters, and filter pools belong to profiles. Profiles offer a way to group connections, share connections, or keep them private, and they help you partition data if you have a lot of connections or filter pools. Your first profile will be for your local workstation. This is your personal profile. You use this profile as you want to keep your connection private. If you wanted to share resources and information with other people you would set up a team profile.

Connections


Remote System Explorer - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Source Compile Navigate Search
Remote Systems
New Connection
Local
TORAS07M
iSeries Objects
iSeries Commands
iSeries Jobs
IFS Files
07M
MyNewConnection
iSeries Objects
Work with lib
Work with obj
Work with mem
Library list
User libraries
DDSSource
mfaraj
test
Remote Systems Team
Properties
iSeries Field Table View
Name Record Type Length Text
Tasks iSeries Table ... iSeries Comm... Outline iSeries Error List iSeries Listings iSeries Field T...
Subsystem: iSeries Objects
ORDENTR
Finish Cancel

New

Remote iSeries System Connection

Define connection information

Parent profile: farr

Connection name: MyNewConnection

Host name: TORAS07M

Description: Testing For Fun

☑ Verify host name

Property	Value
Connected	No
Name	iSeries Objects
Number of c...	9
Port	0
Type	Subsystem

```
...+....4....+....5....
 to enter and conf
tch job via a data
Warehouse '0001'.
```

Remote Systems View

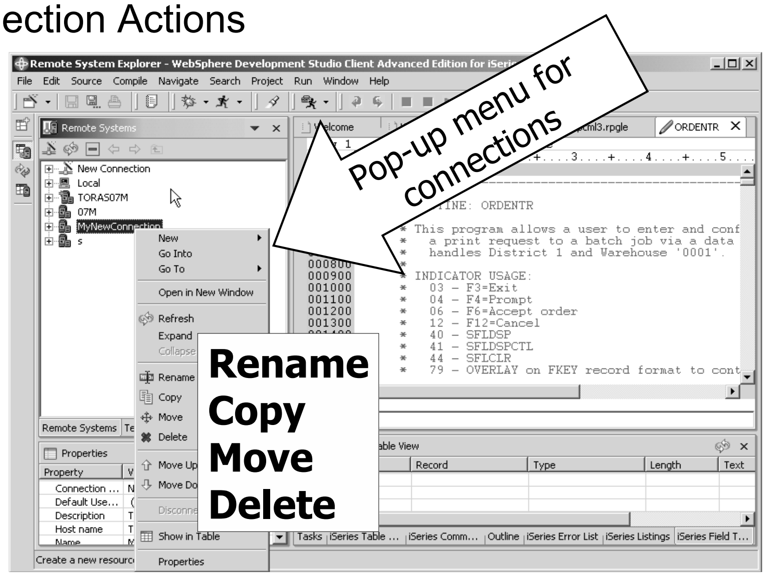
New Connection Wizard

Properties of selected object

Various helper views!

This is the Remote Systems Explorer perspective. This is the first perspective you see when starting Development Studio Client for the first time. The Remote Systems view lists your existing connections, and contains an item that when expanded launches the New Connection wizard for creating a new connection. The New Connection wizard prompts for information about the connection, including an arbitrary but unique name, the type of the remote system, the TCP/IP hostname for the system, the User ID to connect to the system with, and optionally a description of the connection. Once created, a connection is listed in the tree and can be expanded to explore the contents of that remote system, which we will describe soon. The Remote System Explorer contains other views of interest, including the Properties view in the lower left, and various other views in the lower right that are in a tabbed notebook. Like all views in Eclipse, these views can be moved to a different location by dragging and dropping them.

Connection Actions



Pop-up menu for connections

Rename
Copy
Move
Delete

```
        TINE: ORDENTR

        * This program allows a user to enter and conf
        *   a print request to a batch job via a data
        *   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
        *   40 - SFLDSP
        *   41 - SFLDSPCTL
        *   44 - SFLCLR
        *   79 - OVERLAY on FKEY record format to cont
```

Connections support actions in their pop-up menu for manipulating them. These include:

Rename Connection. For giving the connection a new name.

Copy Connection. For creating a new connection based on this connection. You will be prompted for a new name. You can also copy a connection to another profile. We will cover profiles later.

Move Connection. For moving a connection to another profile. We will cover profiles later.

Delete Connection. For deleting a connection.

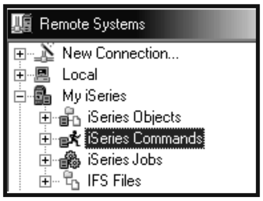
Move Up Connection. For moving the selected connections up in the connection list.

Move Down Connection. For moving the selected connections down in the connection list.

You can also change the attributes of a connection, such as the host name or user Id, either by selecting the Properties popup menu item, or by editing the attributes in the Properties view (be sure to press Enter!)

iSeries Connections

- Connections expand to
  - –“subsystems”
    - –Named grouping of functionality
- Subsystems for iSeries
  - –iSeries Objects
    - –For working with Libraries, Objects and Members
  - –iSeries Commands
    - –For pre-defining and running QSYS command sets
  - –iSeries Jobs
    - –For working with jobs
  - –IFS files
    - –For working with Integrated File System files



Once connections are defined they can be expanded within the Remote Systems Explorer. On expansion, the user sees subsystems, which are merely a functional grouping of the various types of remote resources that can be explored in the remote system

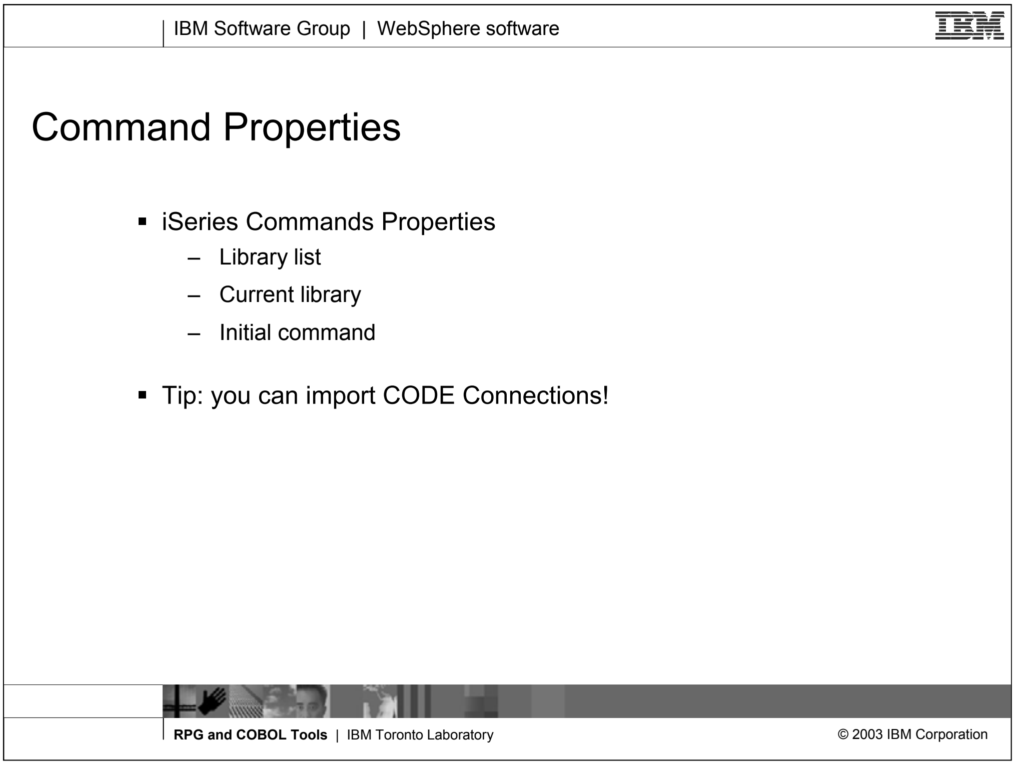
For iSeries connections, there are four subsystems:

•iSeries Objects is the PDM-like grouping, allowing access to libraries, objects and members

•iSeries Commands allows developers to predefine command sets each of which contain one or more often used commands. When run, all commands in a command set are sent to the remote system and executed, and the results are logged in the Commands view.

•iSeries Jobs allows developers to see various jobs, subsettable by job attributes, and to perform a limited number of operations on those jobs

•IFS Files allows developers to explore folders and files in the Integrated File System of the remote iSeries system

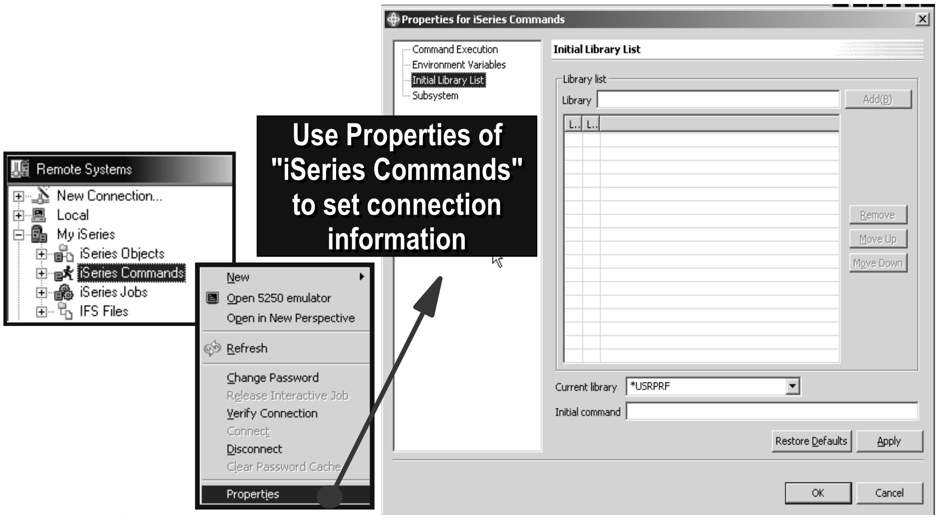


Command Properties

- iSeries Commands Properties
  - Library list
  - Current library
  - Initial command
- Tip: you can import CODE Connections!

When a connection is used to connect to a remote iSeries, your initial program as specified in your user ID is not honored. You can overcome this by using the Properties pop-up menu item from a selected iSeries Commands subsystem under a connection.

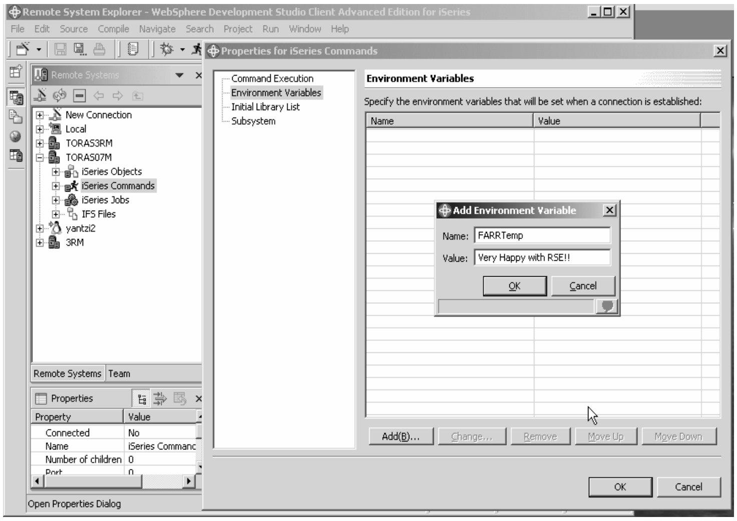
Command Properties (continued)



Use Properties of "iSeries Commands" to set connection information

On this Properties dialog you can specify libraries to add to the library list, specify a current library, and specify an initial command to run. The initial command must not be interactive! When the connection is used to connect to the iSeries, the RSE will execute the appropriate ADDLIBLE, CHGCURLIB commands, and call your initial command.

Set Environment Variables



Specify the environment variables that will be set when a connection is established:

Name	Value

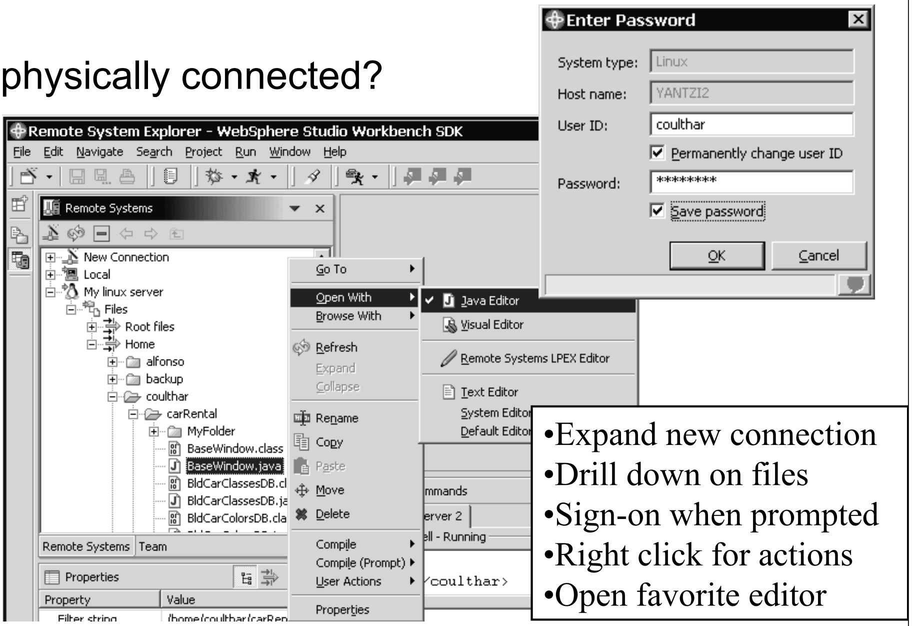
Name: FARRTemp

Value: Very Happy with RSE!!

Property	Value
Connected	No
Name	iSeries Commanc
Number of children	0
Port	0

You can define environment variables for all Remote System Explorer connection types (iSeries, Windows, Linux, and UNIX). If you change an environment variable while the Remote System Explorer is connected to a remote host, the change does not take affect until you disconnect and reconnect. Although environment variables are set in subsystem properties pages, the Remote System Explorer stores them globally by connection.

Is it physically connected?



Enter Password

System type: Linux

Host name: YANTZI2

User ID: coulthar

☑ Permanently change user ID

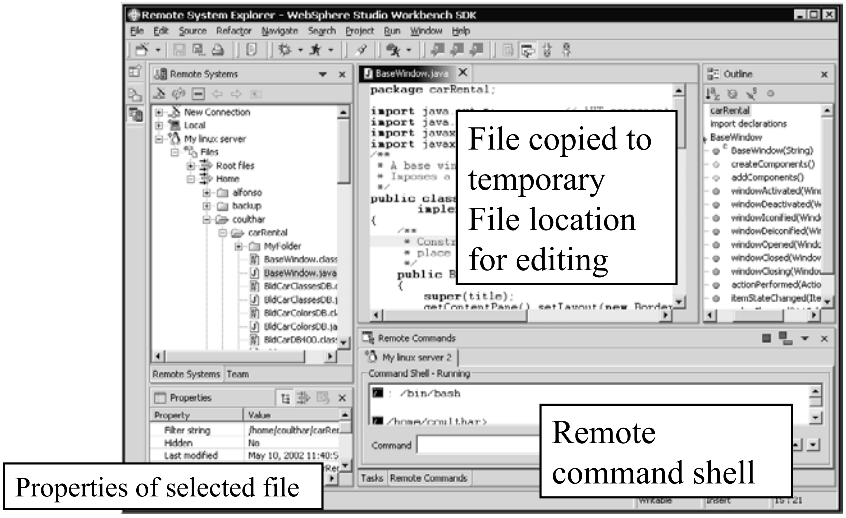
Password: ********

☑ Save password

- Expand new connection
- Drill down on files
- Sign-on when prompted
- Right click for actions
- Open favorite editor

Once you have access to your remote system, you expand the connection, drill down through the files until you locate the file that you want to work with. Next you are prompted to sign-on to the remote system. You can then right-click on the object and open your favorite editor.

RSE Editing Java source code



```
/bin/bash
/home/coulthar>
```

Property	Value
Filter string	/home/coulthar/carRer
Hidden	No
Last modified	May 10, 2002 11:40:5

File copied to temporary File location for editing

Remote command shell

Properties of selected file

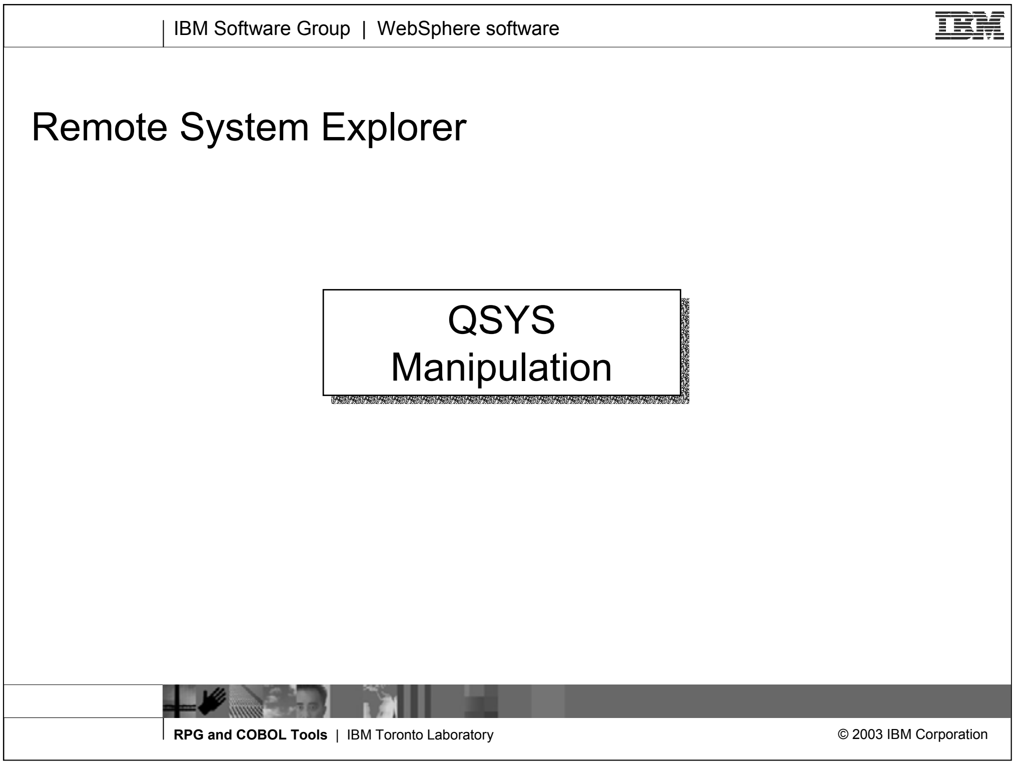
When you open a file for editing, the file is copied to a temporary location for editing. You can use the Remote Commands view to run and interact with commands and command shells on universal systems. A universal system includes Windows, Linux, and UNIX system types. Specifically, use the view to:

•Run commands in a command shell

•Display and interpret the output of a program

•Enter input to a program

•Display and manage different commands and shells from the same view. Multiple commands can be run in a single shell (one command at a time per shell), multiple shells may be run on a single system, and multiple systems may be running shells.



Remote System Explorer

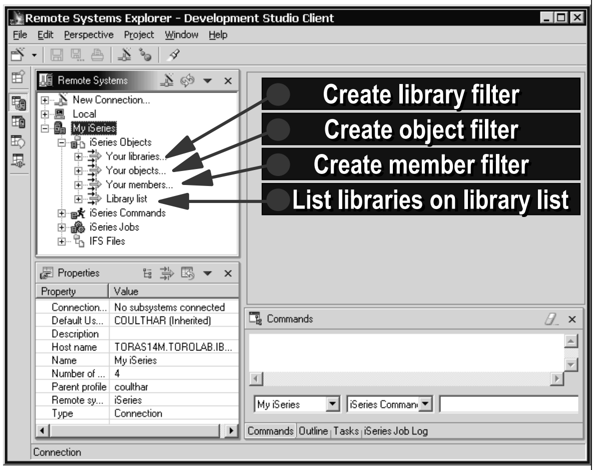
QSYS
Manipulation

The library (QSYS.LIB) file system supports the iSeries library structure. It provides access to database files and all of the other iSeries object types that are managed by the library support. This is a popular system you can access through the Remote System Explorer.

QSYS Explorer

- iSeries Objects SubSystem

  For drill-down or filtered access to QSYS



Property	Value
Connection...	No subsystems connected
Default Us...	COULTHAR (Inherited)
Description	
Host name	TORAS14M.TOROLAB.IB...
Name	My iSeries
Number of ...	4
Parent profile	coulthar
Remote sy...	iSeries
Type	Connection

Create library filter

Create object filter

Create member filter

List libraries on library list

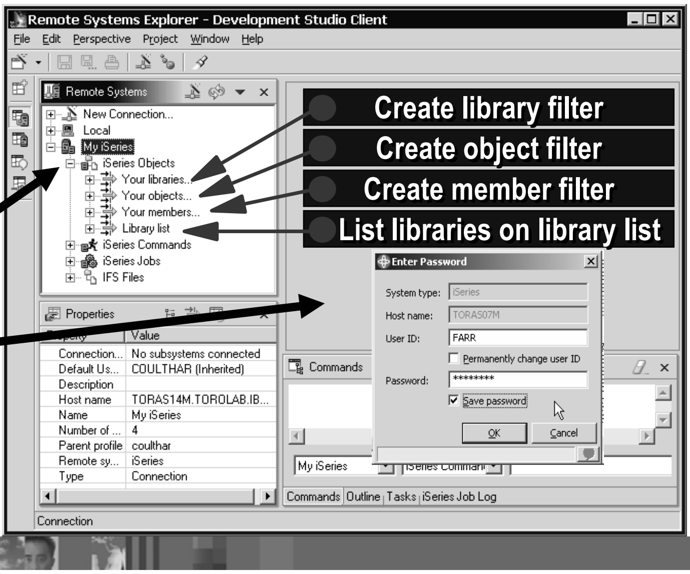
We now drill down into the iSeries Objects subsystem. This is the subsystem you will use most often! It is very similar to PDM, in that it allows you to access objects in the QSYS file system, and perform actions on those objects. The three child items at the top of the list are for creating filters, much like in PDM:

Your libraries... prompts you for a simple or generic library name, and lists all matching libraries. It is similar to WRKLIBPDM.

Your objects... prompts you for a simple or generic library name and simple or generic object name, as well one or more object type and attribute pairs. It lists all matching objects in all matching libraries. it is similar to WRKOBJPDM.

Your members... prompts you for a simple or generic library name, simple or generic file name, and simple or generic member name, as well as one or more member types which can also be generic. It lists all matching members in all matching files in all matching libraries. It is similar to WRKMBRPDM. Unlike PDM, the filters you create are permanently remembered and displayed in this list for easy re-use. We will have more to say about filters. To simulate STRPDM's option 12, you can start with the pre-defined Library list filter, that when expanded lists all libraries in your library list. With any filter, once it is expanded you can subsequently expand a library to see all objects in the library, and expand files to see all members in the file. When you expand your first filter, such as the pre-defined Library List filter, you are prompted for your password and then connected to the remote iSeries. Then, the results of resolving the filter are shown...

Connecting to the iSeries



Create library filter

Create object filter

Create member filter

List libraries on library list

Press "+"

Property	Value
Connection...	No subsystems connected
Default Us...	COULTHAR (Inherited)
Description	
Host name	TORAS14M.TOROLAB.IB...
Name	My iSeries
Number of ...	4
Parent profile	coulthar
Remote sy...	iSeries
Type	Connection

Enter Password

System type: iSeries

Host name: TORAS07M

User ID: FARR

☐ Permanently change user ID

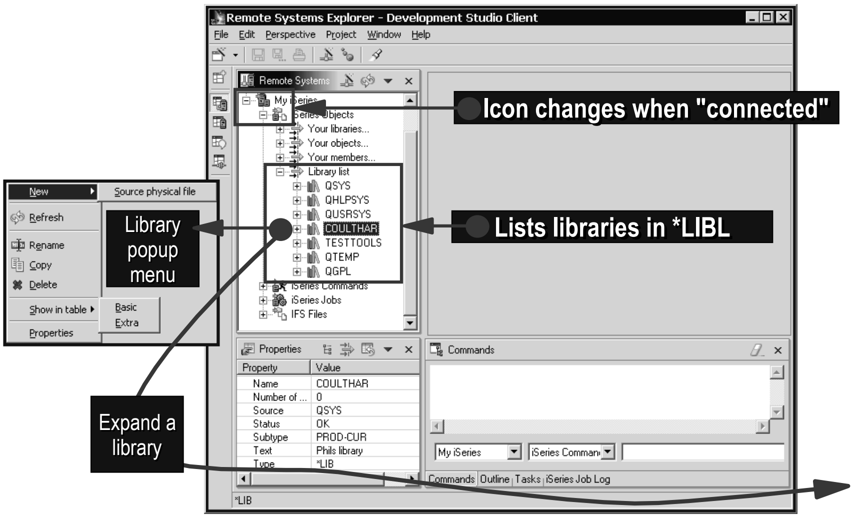
Password: ********

☑ Save password

OK Cancel

To connect to the iSeries, simply press the plus sign "+" next to the iSeries Objects, and a login dialog box will appear. You need to be connected to create the necessary filters or to display the libraries in the library list.

“Library list” is the only pre-defined filter



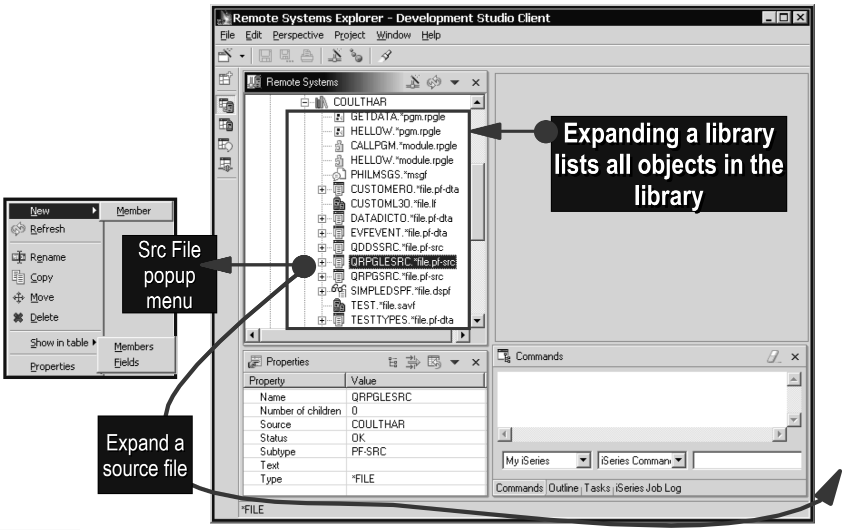
Property	Value
Name	COULTHAR
Number of ...	0
Source	QSYS
Status	OK
Subtype	PROD-CUR
Text	Phils library
Type	*LIB

*LIB

When you are connected to the remote iSeries, the icons for the connection and its subsystems change to include a small green arrow, so as to indicate the connection status. When the pre-defined library list filter is expanded, and the connection is successful, you will see the libraries on your library list. For each library, you can right-click and select from a number of useful actions. There is an action to create a new source file within the selected library, to refresh the contents of the library if it is expanded, to rename the library, copy the library or delete the library. These last three actions remotely run the appropriate iSeries command and you will see it logged in the Commands view. There are also actions to open a multiple-column table view showing the contents of the library, similar again to PDM. There are two levels of details to choose from for the table: Basic and Extra. Extra includes additional columns of information, but will take a bit longer to present.

If you expand a library, you will see all the objects in that library...

Expanding a library to list all objects



COULTHAR
- GETDATA.*pgm.rpgle
- HELLOW.*pgm.rpgle
- CALLPGM.*module.rpgle
- HELLOW.*module.rpgle
- PHILMSGS.*msgf
- CUSTOMERO.*file.pf-dta
- CUSTOML3O.*file.lf
- DATADICTO.*file.pf-dta
- EVFEVENT.*file.pf-dta
- QDDSSRC.*file.pf-src
- QRPGLESRC.*file.pf-src
- QRPGSRC.*file.pf-src
- SIMPLEDSPF.*file.dspf
- TEST.*file.savf
- TESTTYPES.*file.pf-dta

Property	Value
Name	QRPGLESRC
Number of children	0
Source	COULTHAR
Status	OK
Subtype	PF-SRC
Text	
Type	*FILE

Expanding a library lists all objects in the library

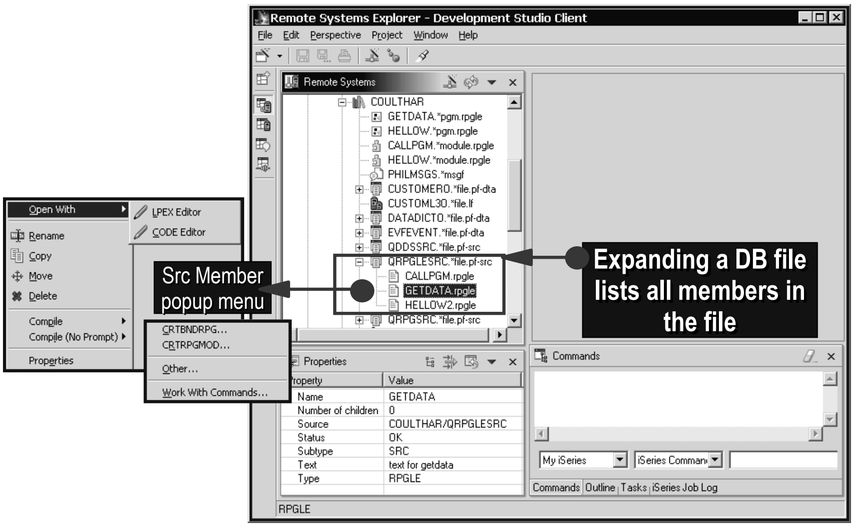
Src File popup menu

Expand a source file

When a library is expanded in the Remote Systems tree view, all the objects within that library are listed underneath the library.

For each object, you can right-click and select from a number of useful actions. The exact list of actions will depend on the type of object you select, and whether you selected one or multiple objects. For a source file, the popup menu has an action to create a new member within the selected file, to refresh the contents of the file if it is expanded, to rename the file, copy the file, move the file and delete the file. These actions remotely run the appropriate iSeries command and you will see it logged in the Commands view. For both data and source files, there are also actions to open a multiple-column table view showing the contents of the file. You can choose to list the members, similar to PDM, or list the fields. If you expand a file, you will see all the members in that file...

Expanding a SRC file to list all members



roperty	Value
Name	GETDATA
Number of children	0
Source	COULTHAR/QRPGLESRC
Status	OK
Subtype	SRC
Text	text for getdata
Type	RPGLE

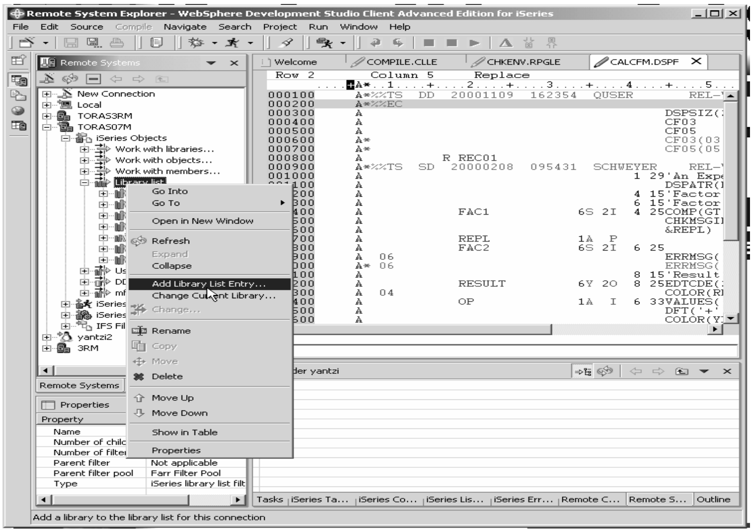
Src Member popup menu

Expanding a DB file lists all members in the file

When a data or source file is expanded in the Remote Systems tree view, all the members within that file are listed underneath the file.

For each member, you can right-click and select from a number of useful actions. The exact list of actions will depend on whether the member is a data file or a source, and whether you selected one or multiple members. For a source file, the popup menu hasctions for editing, renaming, copying, moving, deleting and compiling the source member. We will have more to say about the editing and compiling actions in the next slides...

Add a library to your library list - ADDLIBLE



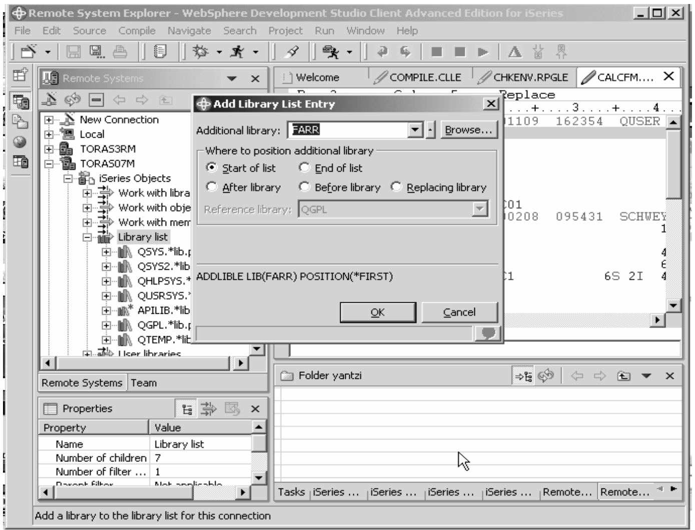
Property	
Name	
Number of chilc	
Number of filter	
Parent filter	Not applicable
Parent filter pool	Farr Filter Pool
Type	iSeries library list filt

```
 Row 2        Column 5      Replace
 .....A*. .1 . . . .+. . . .2. . . .+. . . .3. . . .+. . . .4. . . .+. . . .5. . .
000100     A*%%TS  DD  20001109  162354  QUSER          REL-
000200     A*%%EC
000300     A                                              DSPSIZ(
000400     A                                              CF03
000500     A                                              CF05
000600     A*                                             CF03(03
000700     A*                                             CF05(05
000800     A          R REC01
000900     A*%%TS  SD  20000208  095431  SCHWEYER        REL-
001000     A                                          1 29'An Expe
   100     A                                              DSPATR(
   200     A                                          4 15'Factor
   300     A                                          6 15'Factor
   400     A            FAC1           6S 2I  4 25COMP(GT
   500     A                                              CHKMSGI
   600     A                                              &REPL)
   700     A            REPL           1A  P
   800     A            FAC2           6S 2I  6 25
   900     A  06                                          ERRMSG(
   000     A* 06                                          ERRMSG(
   100     A                                          8 15'Result
   200     A            RESULT         6Y 2O  8 25EDTCDE(
   300     A  04                                          COLOR(R
   400     A            OP             1A  I  6 33VALUES(
   500     A                                              DFT('+'
   600     A                                              COLOR(Y
```

Add a library to the library list for this connection

You can modify your library list through the Remote System Explorer instead of going through a 5250 session. Right click on your library list and select Add Library List Entry… This will perform the ADDLIBLE command in the 5250 session.

Library List dialog



Add Library List Entry

Additional library: FARR Browse...

Where to position additional library

● Start of list ○ End of list

○ After library ○ Before library ○ Replacing library

Reference library: QGPL

ADDLIBLE LIB(FARR) POSITION(*FIRST)

OK Cancel

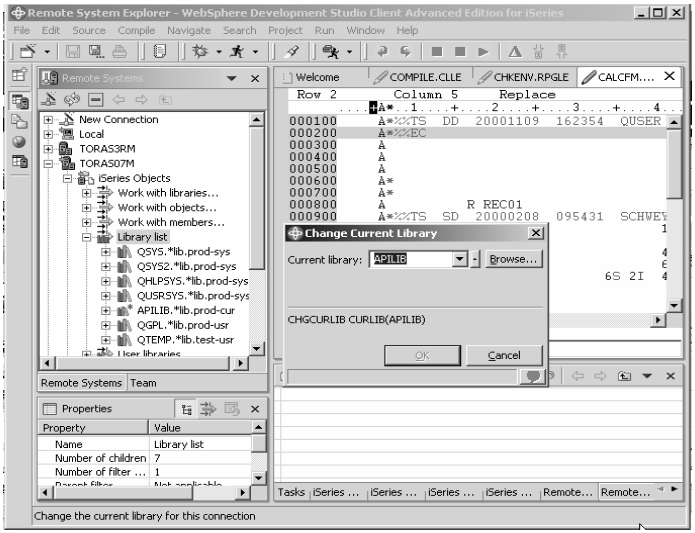
Properties

Property	Value
Name	Library list
Number of children	7
Number of filter ...	1

Add a library to the library list for this connection

Here is where you will fill in the name of your library and specify it’s relative position in your library list.

Change your current library



```
 Row 2        Column 5       Replace
        ....A*..1....+....2....+....3....+....4...
000100      A*%%TS   DD   20001109   162354   QUSER
000200      A*%%EC
000300      A
000400      A
000500      A
000600      A*
000700      A*
000800      A                R REC01
000900      A*%%TS   SD   20000208   095431   SCHWEY
```

Change Current Library

Current library: APILIB [Browse...]

CHGCURLIB CURLIB(APILIB)

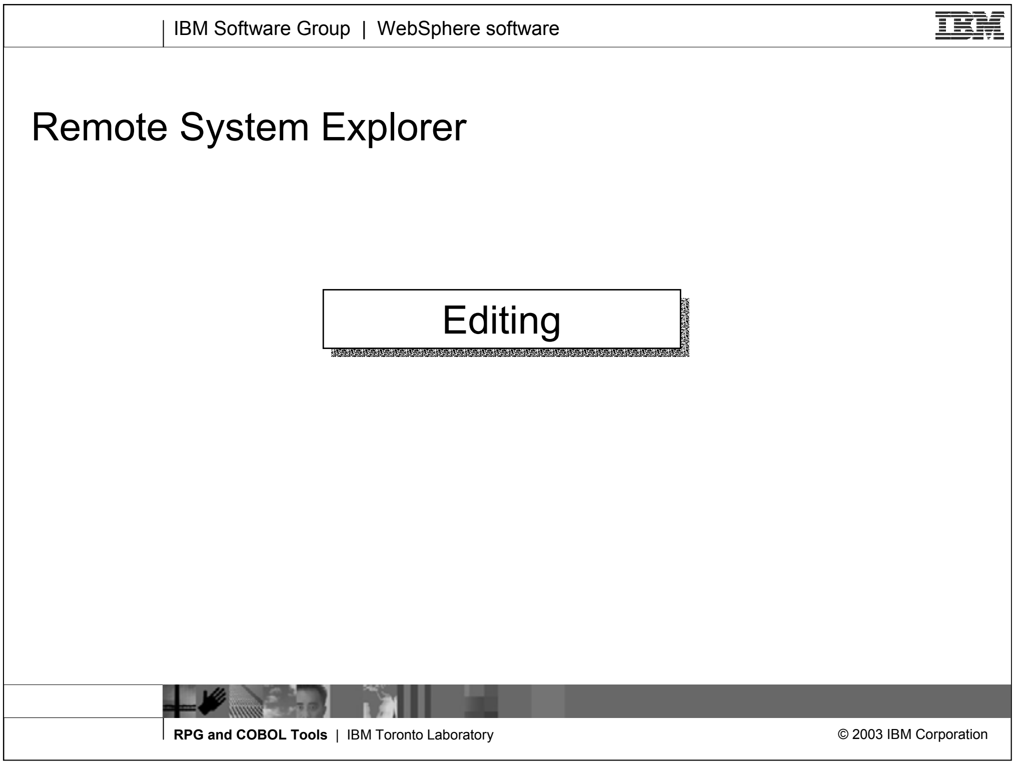
[OK] [Cancel]

Properties

Property	Value
Name	Library list
Number of children	7
Number of filter ...	1
Parent filter	Not applicable

Change the current library for this connection

This dialog shows how you can do a CHGCURLIB command in the Remote System Explorer. This dialog will allow you to change your current library. To get to this dialog, simply right click your library list and select Change Current Library… (its under Add Library List Entry…)

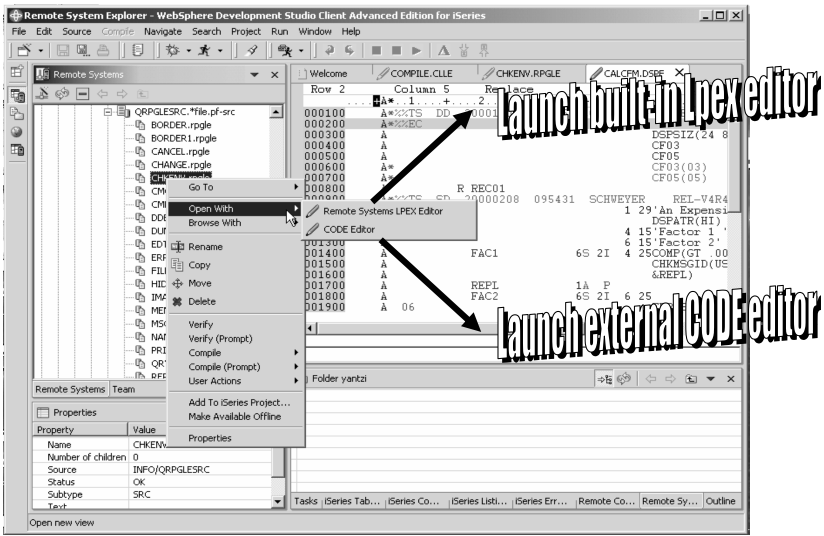


Remote System Explorer

Editing

Your program editing tasks are simplified with the LPEX editor. This is a powerful language-sensitive editor that is easy to customize. Token highlighting of source makes the various program elements stand out. It has SEU-like specification prompts for RPG and DDS to help enter column-sensitive fields. Local syntax checking and semantic verification for your RPG, COBOL and DDS source makes sure it will compile cleanly the first time on an iSeries. If there are verification errors, an Error List lets you locate and resolve problems quickly. On-line programming guides, language references, and context-sensitive help make finding the information you need just a keystroke away.

Editors



Launch built-in Lpex editor

Launch external CODE editor

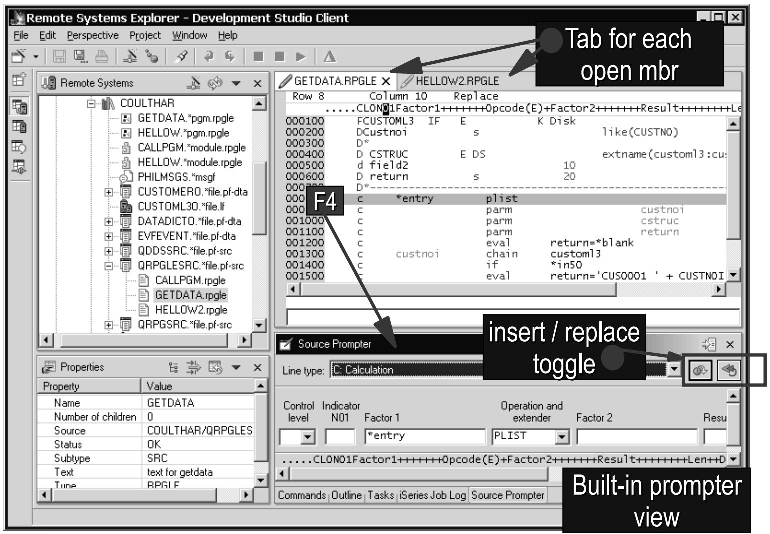
Property	Value
Name	CHKENV
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC
Text	

For a source member, there are two options for editing:

1. Remote Systems LPEX Editor. This is the new editor, written all in Java, that is built-in to the IDE. It is a re-write of the original CODE Editor, but as you will see has a subset of the functionality in CODE at this point.
2. CODE Editor. This is the classic full-functioned CODE editor, which is offered as an alternative until the Lpex editor catches up to the functionality of the CODE editor. This launches the CODE Editor in a separate window.

We will cover the Remote Systems LPEX editor next in more detail...

LPEX (“JLPEX”) built-in IDE editor



Property	Value
Name	GETDATA
Number of children	0
Source	COULTHAR/QRPGLES
Status	OK
Subtype	SRC
Text	text for getdata
Type	RPGLE

```
 Row 8        Column 10     Replace
      .....CLON01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Le
000100     FCUSTOML3   IF   E             K Disk
000200     DCustnoi             s                      like(CUSTNO)
000300     D*
000400     D CSTRUC           E DS                     extname(customl3:cu
000500     d field2                              10
000600     D return             s                20
000        D*-----------------------------------------------------------
000        c         *entry         plist
000        c                        parm                          custnoi
001000     c                        parm                          cstruc
001100     c                        parm                          return
001200     c                        eval      return=*blank
001300     c         custnoi        chain     customl3
001400     c                        if        *in50
001500     c                        eval      return='CUS0001 ' + CUSTNOI
```

Line type: C: Calculation

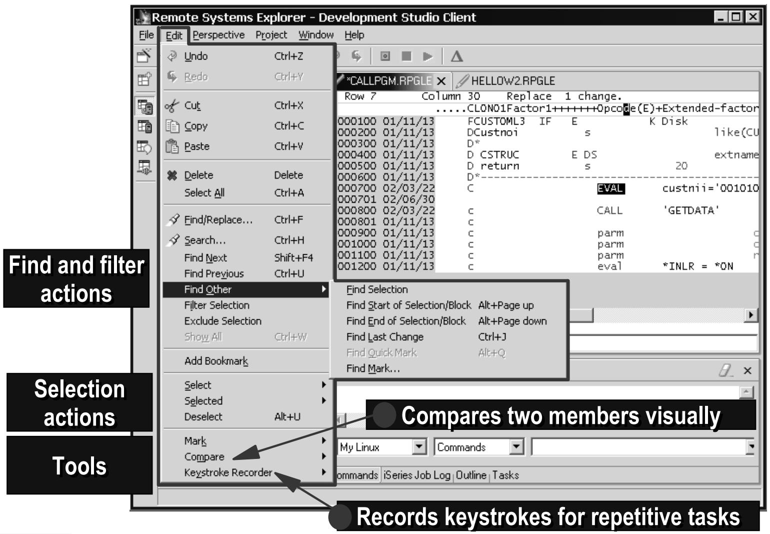
Control level	Indicator N01	Factor 1	Operation and extender	Factor 2	Resu
		*entry	PLIST		

```
.....CLON01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D
```

The Remote Systems LPEX editor is built-in, so it shows up in a pane within the IDE. You can open multiple members for editing, and each will be shown in the editor area with a tab that when selected brings that member to the foreground. You can double click on a tab to expand that member's edit window to full size. When a tab shows an asterisk in it, that indicates there are pending changes that should be saved. For RPG (both III and IV) you will notice there is color highlighting and familiar F4 support to prompt for the current line. The prompter sits is a view that doesn't overlap the editor. When done filling in the prompt, you can press one of two buttons to replace the current line or insert a new line.

LPEX Editor Functions

Edit Menu when Lpex editor has focus



```
Row 7        Column 30    Replace  1 change.
            .....CLON01Factor1+++++++Opcode(E)+Extended-factor
000100 01/11/13      FCUSTOML3  IF   E             K Disk
000200 01/11/13      DCustnoi          s                       like(CU
000300 01/11/13      D*
000400 01/11/13      D CSTRUC        E DS                      extname
000500 01/11/13      D return          s              20
000600 01/11/13      D*--------------------------------------------
000700 02/03/22      C                    EVAL      custnii='001010
000701 02/06/30
000800 02/03/22      c                    CALL      'GETDATA'
000801 01/11/13      c
000900 01/11/13      c                    parm
001000 01/11/13      c                    parm
001100 01/11/13      c                    parm
001200 01/11/13      c                    eval      *INLR = *ON
```

Find and filter actions

Selection actions

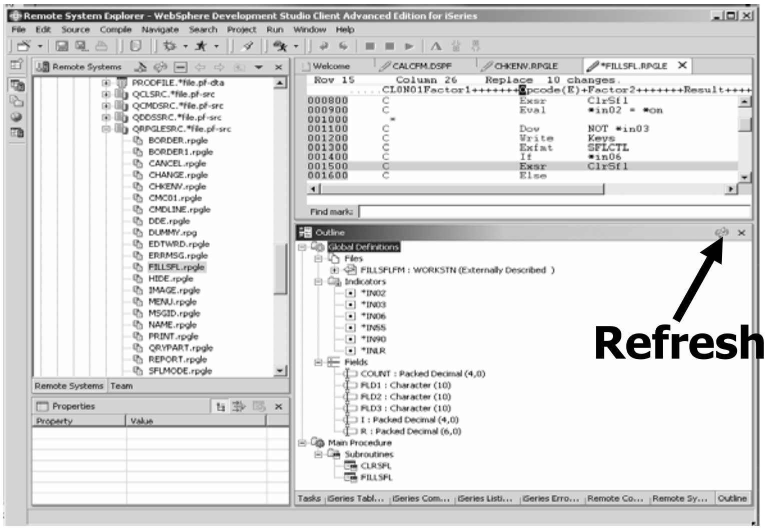
Tools

Compares two members visually

Records keystrokes for repetitive tasks

Here we see the pop-up menu within the LPEX editor. You can find strings in one or more files and you can filter files by type of line. You can also visually compare files for differences. Keystroke recording facilities let you *record* sets of keystrokes, which you can later replay or incorporate into your own macros.

LPEX Editor Views – Outline view



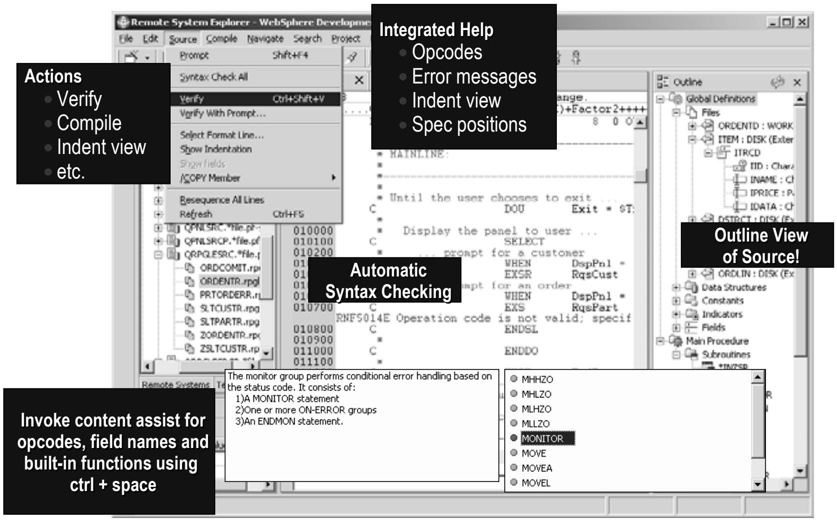
```
 Row 15      Column 26     Replace   10 changes.
      .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++
000800       C                         Exsr      ClrSfl
000900       C                         Eval      *in02 = *on
001000        *
001100       C                         Dow       NOT *in03
001200       C                         Write     Keys
001300       C                         Exfmt     SFLCTL
001400       C                         If        *in06
001500       C                         Exsr      ClrSfl
001600       C                         Else
```

```
Global Definitions
  Files
    FILLSFLFM : WORKSTN (Externally Described )
  Indicators
    *IN02
    *IN03
    *IN06
    *IN55
    *IN90
    *INLR
  Fields
    COUNT : Packed Decimal (4,0)
    FLD1 : Character (10)
    FLD2 : Character (10)
    FLD3 : Character (10)
    I : Packed Decimal (4,0)
    R : Packed Decimal (6,0)
Main Procedure
  Subroutines
    CLRSFL
    FILLSFL
```

Refresh!

The Outline view helps you visualize the member you are editing by displaying all the program objects and functions in a clear-cut view.

Editing Tools



```
                    )+Factor2++++
     * MAINLINE:
     *
     *------------------------------------
     *
     * Until the user chooses to exit ...
     C                   DOU       Exit = *ST
     *
     *   Display the panel to user ...
010000
010100 C                   SELECT
010200 *      ... prompt for a customer
01                         WHEN      DspPnl =
01                         EXSR      RqsCust
01          ...prompt for an order
01                         WHEN      DspPnl =
010700 C                   EXS       RqsPart
RNF5014E Operation code is not valid; specif
010800 C                   ENDSL
010900 *
011000 C                   ENDDO
011100 *
```

The monitor group performs conditional error handling based on the status code. It consists of:
1)A MONITOR statement
2)One or more ON-ERROR groups
3)An ENDMON statement.

Actions
- Verify
- Compile
- Indent view
- etc.

Integrated Help
- Opcodes
- Error messages
- Indent view
- Spec positions

Outline View of Source!

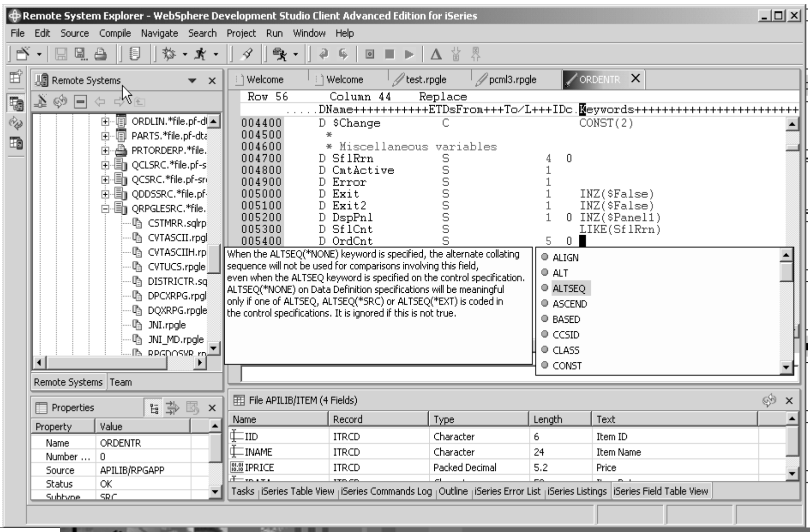
Automatic Syntax Checking

Invoke content assist for opcodes, field names and built-in functions using ctrl + space

The Remote Systems LPEX Editor opens, built right into the workbench, with rich editing functions and is iSeries aware! It is a superset of SEU! The syntax checker is ported down from SEU, the compilers embedded for verifying errors and the reference manuals are built-in and F1 cursor sensitive. The Outline view will show us the program hierarchy. There is explicit and rich iSeries support for verify, compile, run and debug of RPG, COBOL, C, C++, CL and DDS from the Remote Systems LPEX Editor.

Many of the editing features offered in the CODE Editor for RPG, COBOL, CL, and DDS source, such as syntax checking, automatic uppercasing, program verification, and so on, are now available in the Remote Systems LPEX Editor.

Content Assist



```
 Row 56     Column 44    Replace
     .....DName+++++++++++ETDsFrom+++To/L+++IDc.Keywords+++++++++++++++++++++++++
004400     D $Change         C                    CONST(2)
004500      *
004600      * Miscellaneous variables
004700     D SflRrn          S                4  0
004800     D CmtActive       S                1
004900     D Error           S                1
005000     D Exit            S                1   INZ($False)
005100     D Exit2           S                1   INZ($False)
005200     D DspPnl          S                1  0 INZ($Panel1)
005300     D SflCnt          S                    LIKE(SflRrn)
005400     D OrdCnt          S                5  0
```

When the ALTSEQ(*NONE) keyword is specified, the alternate collating sequence will not be used for comparisons involving this field, even when the ALTSEQ keyword is specified on the control specification. ALTSEQ(*NONE) on Data Definition specifications will be meaningful only if one of ALTSEQ, ALTSEQ(*SRC) or ALTSEQ(*EXT) is coded in the control specifications. It is ignored if this is not true.

Properties

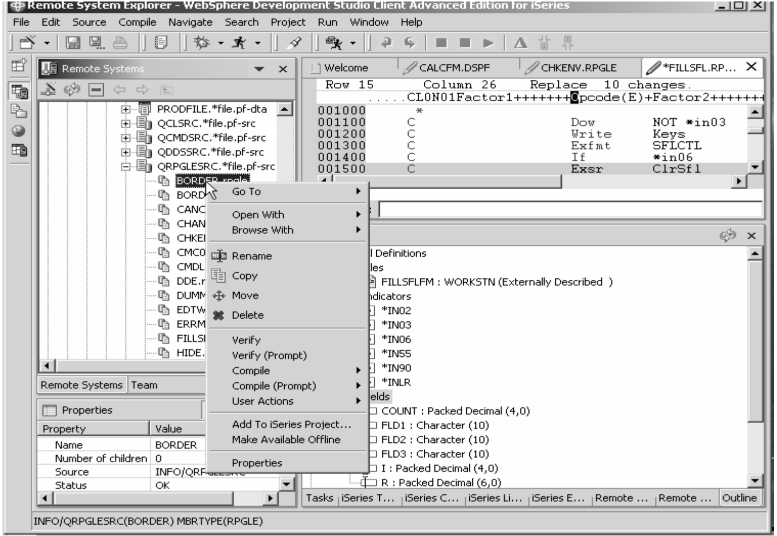
Property	Value
Name	ORDENTR
Number ...	0
Source	APILIB/RPGAPP
Status	OK
Subtype	SRC

File APILIB/ITEM (4 Fields)

Name	Record	Type	Length	Text
IID	ITRCD	Character	6	Item ID
INAME	ITRCD	Character	24	Item Name
IPRICE	ITRCD	Packed Decimal	5.2	Price

The Content Assist tool offers not only auto-complete functionality, by giving you a list of possible functions, objects or keywords to use, but also offers documentation on each of these to help you decide.

Compiling and Verifying



```
 Row 15      Column 26     Replace   10 changes.
 .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++
001000        *
001100       C                         Dow        NOT *in03
001200       C                         Write      Keys
001300       C                         Exfmt      SFLCTL
001400       C                         If         *in06
001500       C                         Exsr       ClrSfl
```

l Definitions
les
FILLSFLFM : WORKSTN (Externally Described)
ndicators
*IN02
*IN03
*IN06
*IN55
*IN90
*INLR
elds
COUNT : Packed Decimal (4,0)
FLD1 : Character (10)
FLD2 : Character (10)
FLD3 : Character (10)
I : Packed Decimal (4,0)
R : Packed Decimal (6,0)

Properties

Property	Value
Name	BORDER
Number of children	0
Source	INFO/QRPGLESRC
Status	OK

INFO/QRPGLESRC(BORDER) MBRTYPE(RPGLE)

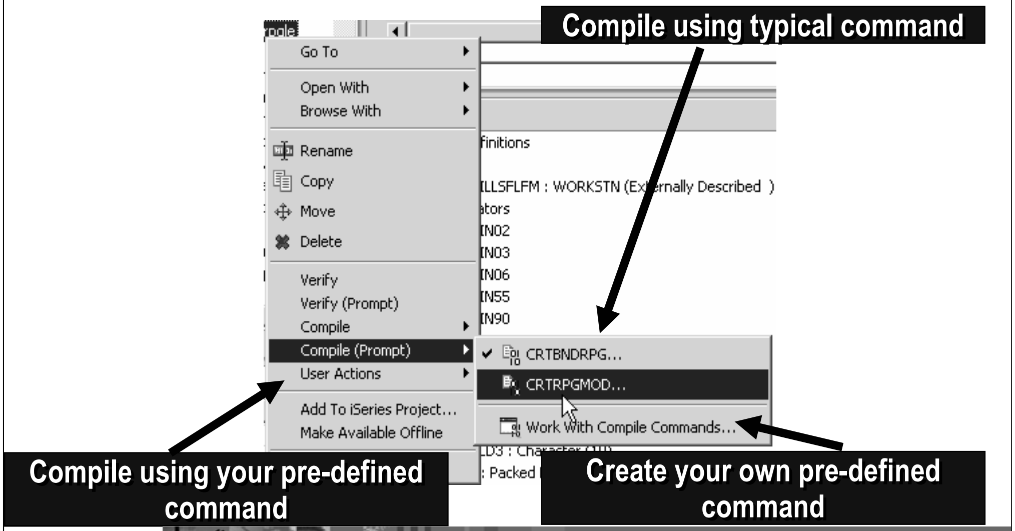
For a source member, there are two primary options for compiling:

1. Compile (prompt). This runs the selected compile command, and prompts you for parameters.

2. Compile (No Prompt). The runs the selected compile command, without prompting for the parameters.

IBM pre-supplies some compile commands specific for the member type, or you can identify your own commands to use see and use, per member type.

Compile options



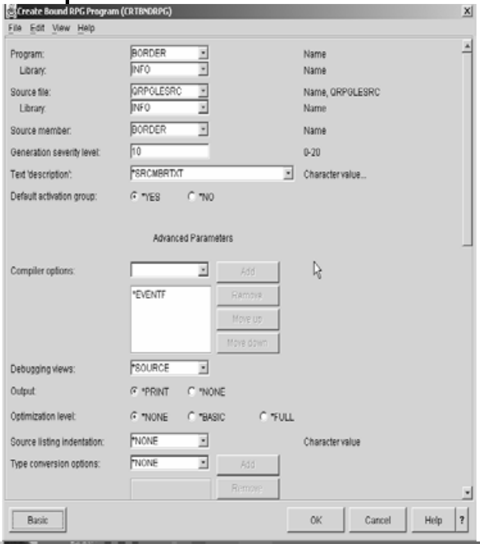
Compile using typical command

Compile using your pre-defined command

Create your own pre-defined command

Development Studio Client offers both ease and flexibility by allowing you to compile with a typical command or by letting you create and use your own compile commands.

Compile with prompt …



Create Bound RPG Program (CRTBNDRPG)

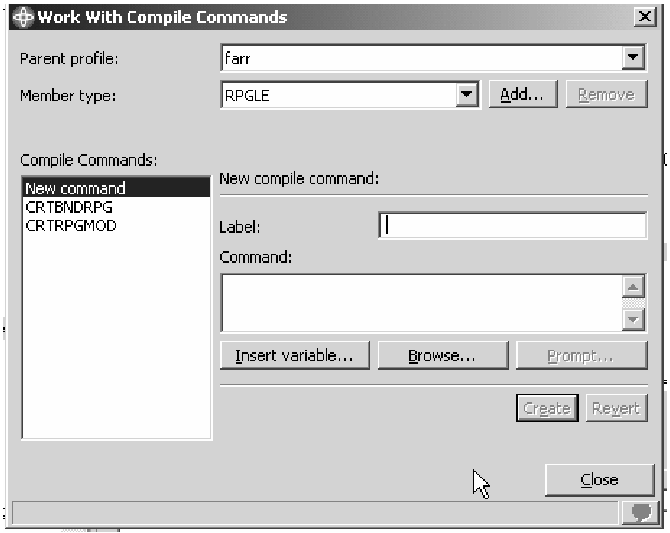
Program: BORDER Name
Library: INFO Name
Source file: QRPGLESRC Name, QRPGLESRC
Library: INFO Name
Source member: BORDER Name
Generation severity level: 10 0-20
Text 'description': *SRCMBRTXT Character value...
Default activation group: ● *YES ○ *NO

Advanced Parameters

Compiler options: *EVENTF
Debugging views: *SOURCE
Output: ● *PRINT ○ *NONE
Optimization level: ● *NONE ○ *BASIC ○ *FULL
Source listing indentation: *NONE Character value
Type conversion options: *NONE

When you select to prompt the compile command, the command prompt is converted to a GUI and displayed.

Work With Compile Commands



Work With Compile Commands

Parent profile: farr

Member type: RPGLE

Compile Commands:
- New command
- CRTBNDRPG
- CRTRPGMOD

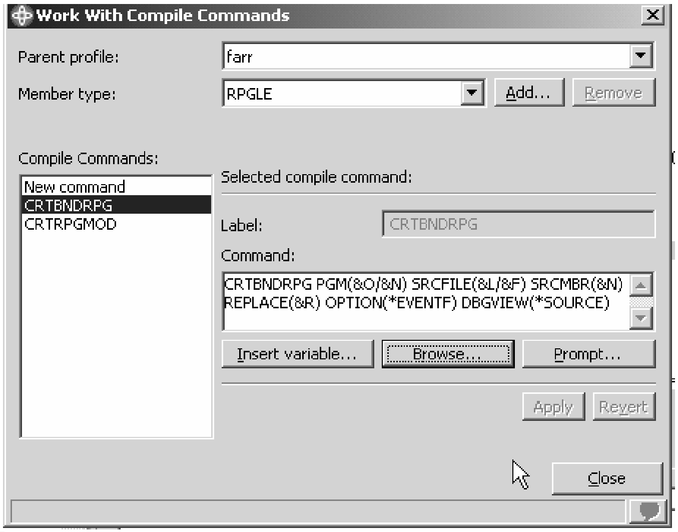
New compile command:

Label:

Command:

To create your own compile actions, select the Work With Commands menu item from the cascading compile menu in the popup menu for a source member. Here, use the Add or Duplicate buttons to create the commands you want. These may be totally different commands, or the same commands but with different parameters. When adding new commands, be sure to select the member type to scope it to. It will only show up in the compile menu for members of this type. You can elect to have these commands show up in the compile menus. If you choose not to, then you must use Other... to select the command at compile time.

RPG compile command



Work With Compile Commands

Parent profile: farr

Member type: RPGLE

Compile Commands:

- New command
- CRTBNDRPG
- CRTRPGMOD

Selected compile command:

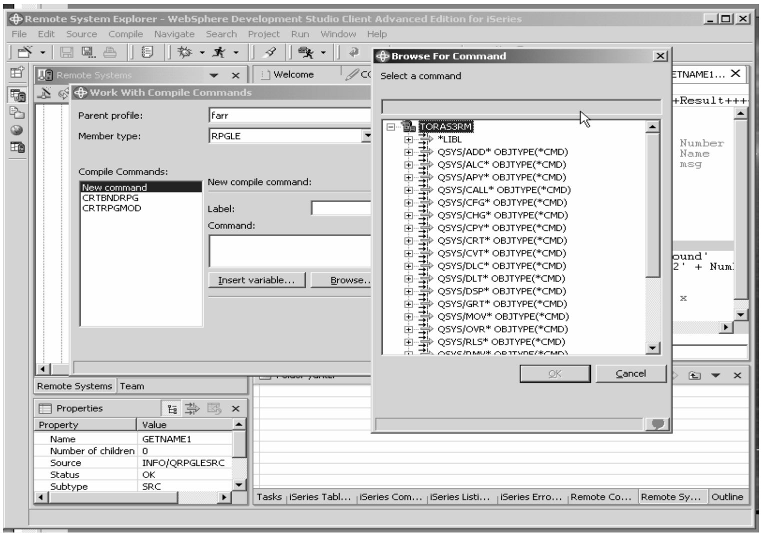
Label: CRTBNDRPG

Command:

```
CRTBNDRPG PGM(&O/&N) SRCFILE(&L/&F) SRCMBR(&N)
REPLACE(&R) OPTION(*EVENTF) DBGVIEW(*SOURCE)
```

Here you can edit the RPG compile command.

Browse For a Command



Browse For Command

Select a command

```
TORAS3RM
  *LIBL
  QSYS/ADD* OBJTYPE(*CMD)
  QSYS/ALC* OBJTYPE(*CMD)
  QSYS/APY* OBJTYPE(*CMD)
  QSYS/CALL* OBJTYPE(*CMD)
  QSYS/CFG* OBJTYPE(*CMD)
  QSYS/CHG* OBJTYPE(*CMD)
  QSYS/CPY* OBJTYPE(*CMD)
  QSYS/CRT* OBJTYPE(*CMD)
  QSYS/CVT* OBJTYPE(*CMD)
  QSYS/DLC* OBJTYPE(*CMD)
  QSYS/DLT* OBJTYPE(*CMD)
  QSYS/DSP* OBJTYPE(*CMD)
  QSYS/GRT* OBJTYPE(*CMD)
  QSYS/MOV* OBJTYPE(*CMD)
  QSYS/OVR* OBJTYPE(*CMD)
  QSYS/RLS* OBJTYPE(*CMD)
```

Work With Compile Commands

Parent profile: Farr

Member type: RPGLE

Compile Commands:

- New command
- CRTBNDRPG
- CRTRPGMOD

New compile command:

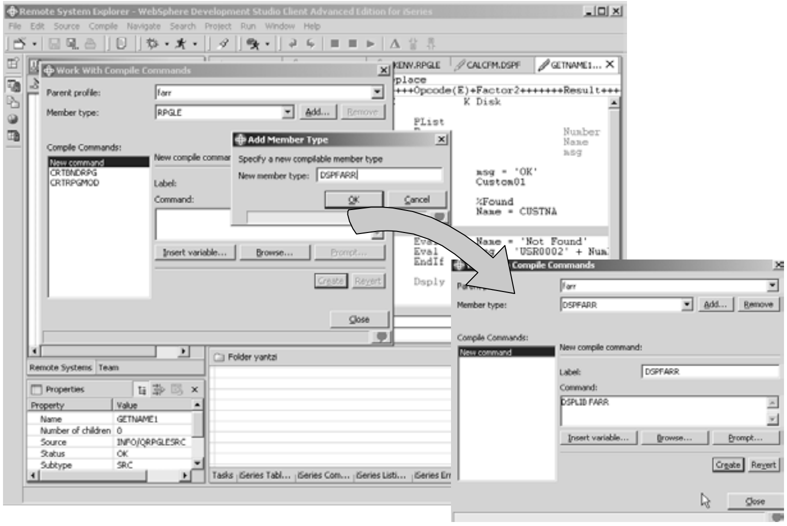
Label:

Command:

Property	Value
Name	GETNAME1
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC

You can also browse for a command instead of typing it in yourself.

Add Member type


Remote System Explorer - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Source Compile Navigate Search Project Run Window Help
KINV.RPGLE
CALCFM.DSPF
GETNAME1...
Remote Systems
Team
Folder yantzi
Tasks
iSeries Tabl...
iSeries Com...
iSeries Listi...
iSeries Err

Work With Compile Commands

Parent profile: farr

Member type: RPGLE Add... Remove

Compile Commands:

- New command
- CRTBNDRPG
- CRTRPGMOD

New compile comman

Label:

Command:

Insert variable... Browse... Prompt...

Create Revert

Close

Add Member Type

Specify a new compilable member type

New member type: DSPFARR

OK Cancel

```
place
+++Opcode(E)+Factor2+++++++Result+++
          K Disk
PList
                              Number
                              Name
                              msg
          msg = 'OK'
          Custom01
          %Found
          Name = CUSTNA
Eva       Name = 'Not Found'
Eval        g   'USR0002' + Num
EndIf
Dsply
```

Property	Value
Name	GETNAME1
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC

Compile Commands

Parent: farr

Member type: DSPFARR Add... Remove

Compile Commands:

- New command

New compile command:

Label: DSPFARR

Command: DSPLID FARR

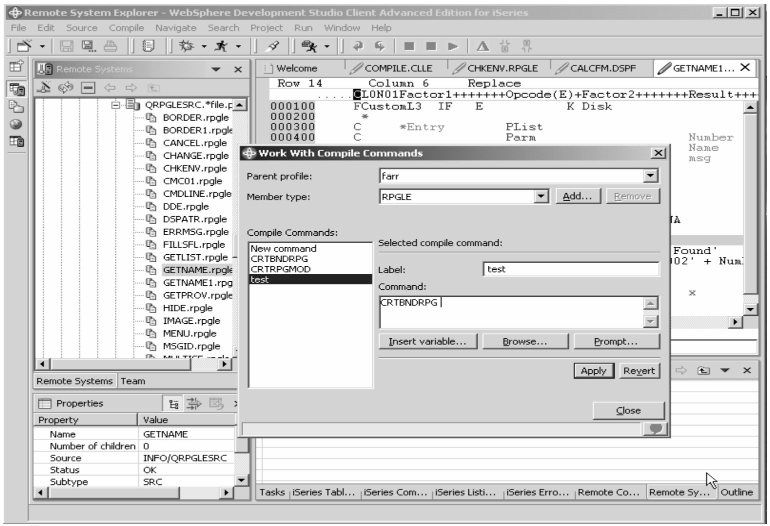
Insert variable... Browse... Prompt...

Create Revert

Close

If you want to work with something other than RPGLE member types, you can create your own member type.

Create your own compile command



QRPGLESRC.*file.p
BORDER.rpgle
BORDER1.rpgle
CANCEL.rpgle
CHANGE.rpgle
CHKENV.rpgle
CMC01.rpgle
CMDLINE.rpgle
DDE.rpgle
DSPATR.rpgle
ERRMSG.rpgle
FILLSFL.rpgle
GETLIST.rpgle
GETNAME.rpgle
GETNAME1.rpg
GETPROV.rpgle
HIDE.rpgle
IMAGE.rpgle
MENU.rpgle
MSGID.rpgle

```
 Row 14       Column 6      Replace
      .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result+++
000100      FCustomL3  IF   E           K Disk
000200        *
000300      C     *Entry        PList
000400      C                   Parm
```

Property	Value
Name	GETNAME
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC

Parent profile: farr

Member type: RPGLE

Compile Commands:
New command
CRTBNDRPG
CRTRPGMOD
test

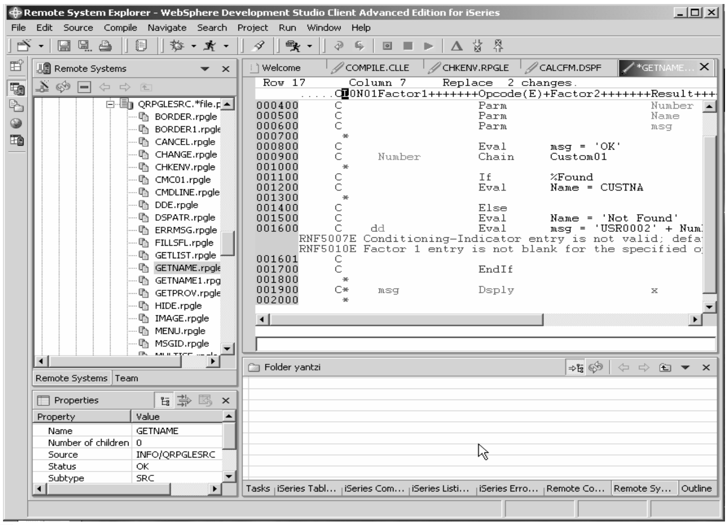
Selected compile command:

Label: test

Command: CRTBNDRPG

Here is how you can create your own compile command. You select the object you want to create a command for and then right-click. From the pop-up menu you select Work with Compile Commands.

Instant syntax checking



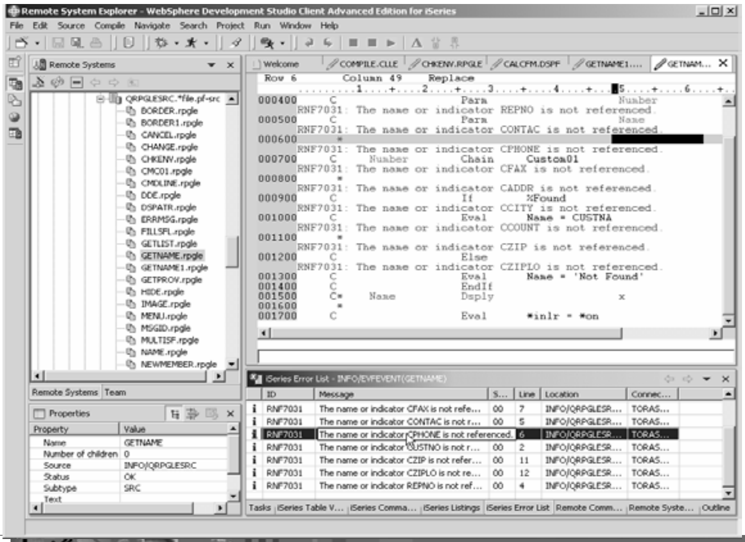
```
 Row 17        Column 7      Replace   2 changes.
        .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result+++
000400      C                        Parm                      Number
000500      C                        Parm                      Name
000600      C                        Parm                      msg
000700       *
000800      C                        Eval      msg = 'OK'
000900      C     Number             Chain     Custom01
001000       *
001100      C                        If        %Found
001200      C                        Eval      Name = CUSTNA
001300       *
001400      C                        Else
001500      C                        Eval      Name = 'Not Found'
001600      C    dd                  Eval      msg = 'USR0002' + Num
       RNF5007E Conditioning-Indicator entry is not valid; defa
       RNF5010E Factor 1 entry is not blank for the specified o
001601      C
001700      C                        EndIf
001800       *
001900      C*    msg                Dsply                     x
002000       *
```

Properties

Property	Value
Name	GETNAME
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC

The Remote Systems LPEX Editor gives you instant feedback for your code, at the location where a potential problem may occur.

Your own compile error list



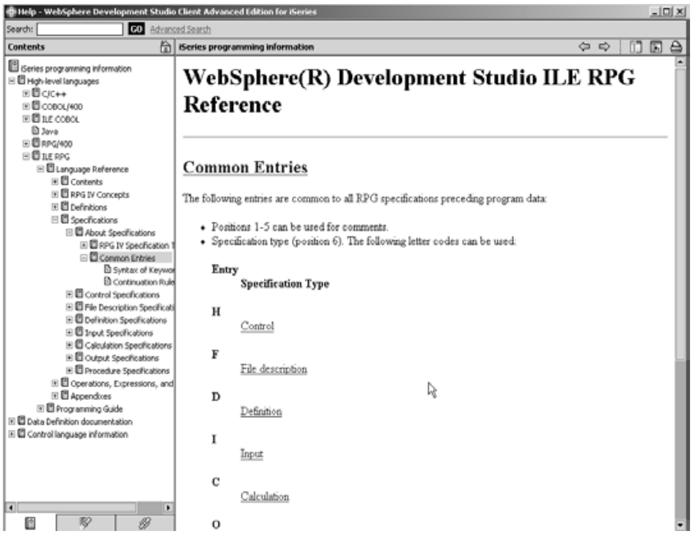
Property	Value
Name	GETNAME
Number of children	0
Source	INFO/QRPGLESRC
Status	OK
Subtype	SRC
Text	

```
Row 6        Column 49     Replace
 .........1....+....2....+....3....+....4....+...▮5....+....6....+.
000400     C                        Parm                       Number
      RNF7031: The name or indicator REPNO is not referenced.
000500     C                        Parm                       Name
      RNF7031: The name or indicator CONTAC is not referenced.
000600      *
      RNF7031: The name or indicator CPHONE is not referenced.
000700     C    Number              Chain      Custom01
      RNF7031: The name or indicator CFAX is not referenced.
000800      *
      RNF7031: The name or indicator CADDR is not referenced.
000900     C                        If         %Found
      RNF7031: The name or indicator CCITY is not referenced.
001000     C                        Eval       Name = CUSTNA
      RNF7031: The name or indicator CCOUNT is not referenced.
001100      *
      RNF7031: The name or indicator CZIP is not referenced.
001200     C                        Else
      RNF7031: The name or indicator CZIPLO is not referenced.
001300     C                        Eval       Name = 'Not Found'
001400     C                        EndIf
001500     C*   Name                Dsply                     x
001600      *
001700     C                        Eval       *inlr = *on
```

	ID	Message	S...	Line	Location	Connec...
i	RNF7031	The name or indicator CFAX is not refe...	00	7	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator CONTAC is not r...	00	5	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator CPHONE is not referenced.		6	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator CUSTNO is not r...	00	2	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator CZIP is not refer...	00	11	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator CZIPLO is not re...	00	12	INFO/QRPGLESR...	TORAS...
i	RNF7031	The name or indicator REPNO is not ref...	00	4	INFO/QRPGLESR...	TORAS...

Double clicking on an error will bring you directly to the place it occurred in the code.

F1 Help on Op-Codes…new in 5.0



WebSphere(R) Development Studio ILE RPG Reference

Common Entries

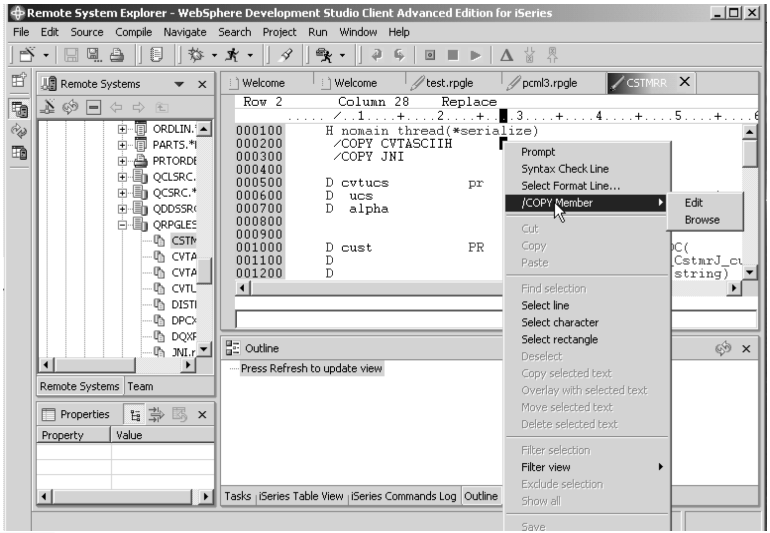
The following entries are common to all RPG specifications preceding program data:

- Positions 1-5 can be used for comments.
- Specification type (position 6). The following letter codes can be used:

Entry	Specification Type
H	Control
F	File description
D	Definition
I	Input
C	Calculation
O	

Now you can have a ILE RPG reference at your finger tips. Simply press F1

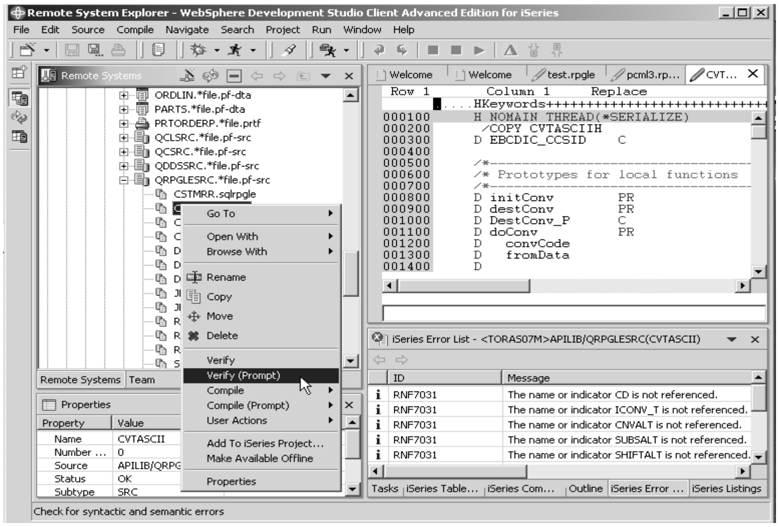
/Copy enhancement



```
H nomain thread(*serialize)
 /COPY CVTASCIIH
 /COPY JNI

D cvtucs          pr
D  ucs
D  alpha


D cust            PR
D
D
```

You can easily copy a member to edit or browse using the /COPY Member option.

Verify your code



Property	Value
Name	CVTASCII
Number ...	0
Source	APILIB/QRPG
Status	OK
Subtype	SRC

```
 Row 1        Column 1      Replace
      ....HKeywords++++++++++++++++++++++++++++
000100     H NOMAIN THREAD(*SERIALIZE)
000200      /COPY CVTASCIIH
000300     D EBCDIC_CCSID      C
000400
000500     /*-----------------------------------
000600     /* Prototypes for local functions
000700     /*-----------------------------------
000800     D initConv          PR
000900     D destConv          PR
001000     D DestConv_P        C
001100     D doConv            PR
001200     D   convCode
001300     D   fromData
001400     D
```

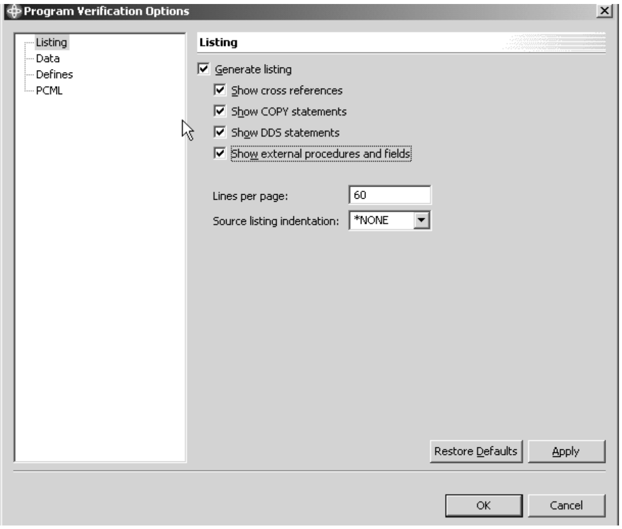
iSeries Error List - <TORAS07M>APILIB/QRPGLESRC(CVTASCII)

	ID	Message
i	RNF7031	The name or indicator CD is not referenced.
i	RNF7031	The name or indicator ICONV_T is not referenced.
i	RNF7031	The name or indicator CNVALT is not referenced.
i	RNF7031	The name or indicator SUBSALT is not referenced.
i	RNF7031	The name or indicator SHIFTALT is not referenced.

Check for syntactic and semantic errors

One of the most powerful and unique features of the Remote System Explorer is the Program Verifier. Before you compile your code on an iSeries, you can make certain that there are no errors by invoking the Program Verifier. The verifier checks for semantic (compile) errors on your workstation so that you can guarantee a clean compile on the iSeries. Think of the host cycles you'll save. It is especially handy when you are writing code but you are disconnected from an iSeries. You can do this because RSE ported the parsing and checking code from the iSeries host compilers to the workstation. The Error List window lists the errors that are found and their severity, inserts the error messages directly into the source and helps you to navigate between the errors.

Configuring the Verify tool



Program Verification Options

Listing

☑ Generate listing

☑ Show cross references

☑ Show COPY statements

☑ Show DDS statements

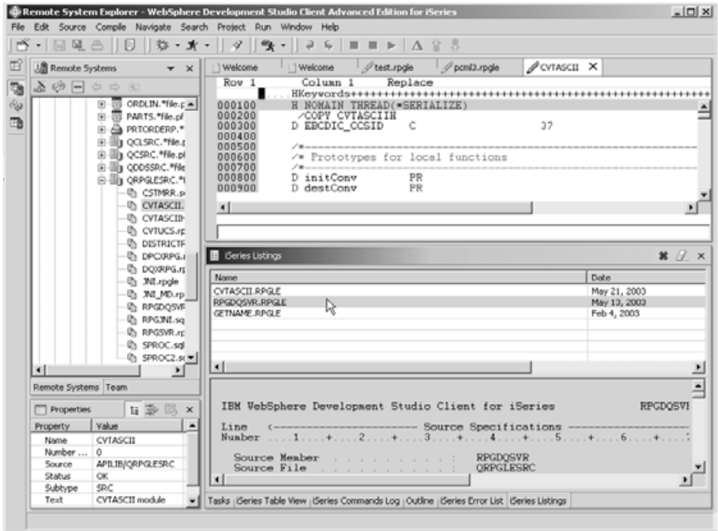
☑ Show external procedures and fields

Lines per page: 60

Source listing indentation: *NONE

You can easily configure the verify tool with the parameters you want to use. Here you see the Listing view parameters.

Listing View



```
 Row 1        Column 1     Replace
        ....HKeywords++++++++++++++++++++++++++++++++++++++++++++++++++++++++
            H NOMAIN THREAD(*SERIALIZE)
             /COPY CVTASCIIH
            D EBCDIC_CCSID     C                    37

            /*-----------------------------------------------------------------
            /* Prototypes for local functions
            /*-----------------------------------------------------------------
            D initConv         PR
            D destConv         PR
```

Name	Date
CVTASCII.RPGLE	May 21, 2003
RPGDQSVR.RPGLE	May 13, 2003
GETNAME.RPGLE	Feb 4, 2003

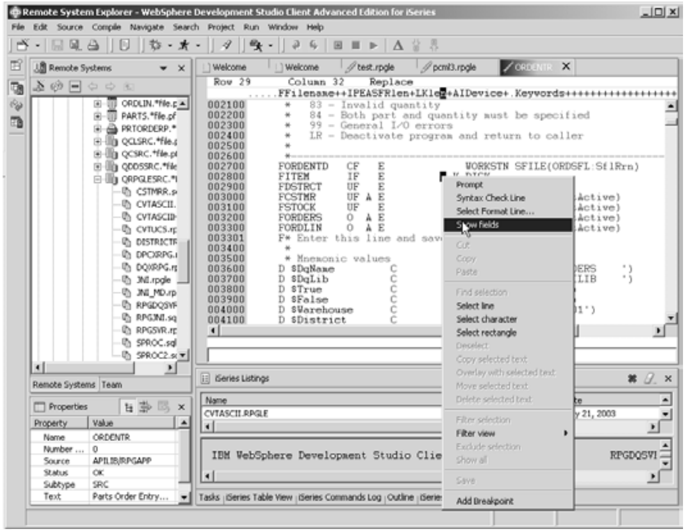
```
IBM WebSphere Development Studio Client for iSeries                RPGDQSVI

Line     <--------------------- Source Specifications ------------------
Number   ....1....+....2....+....3....+....4....+....5....+....6....+....:

  Source Member  . . . . . . . . . . :   RPGDQSVR
  Source File  . . . . . . . . . . . :   QRPGLESRC
```

Properties

Property	Value
Name	CVTASCII
Number ...	0
Source	APILIB/QRPGLESRC
Status	OK
Subtype	SRC
Text	CVTASCII module

Here is the Listing view from the Program Verifier.

F-Spec Show Fields!



Property	Value
Name	ORDENTR
Number ...	0
Source	APILIB/RPGAPP
Status	OK
Subtype	SRC
Text	Parts Order Entry...

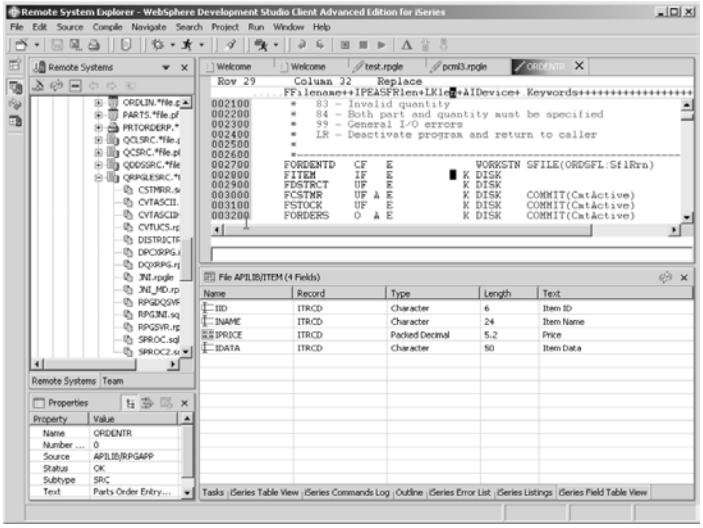
```
 Row 29      Column 32     Replace
      .....FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords++++++++++++++++++
002100        *   83 - Invalid quantity
002200        *   84 - Both part and quantity must be specified
002300        *   99 - General I/O errors
002400        *   LR - Deactivate program and return to caller
002500        *
002600        *-----------------------------------------------------------------
002700       FORDENTD   CF   E             WORKSTN SFILE(ORDSFL:SflRrn)
002800       FITEM      IF   E
002900       FDSTRCT    UF   E
003000       FCSTMR     UF A E                      Active)
003100       FSTOCK     UF   E                      Active)
003200       FORDERS    O  A E                      Active)
003300       FORDLIN    O  A E                      Active)
003301       F* Enter this line and sav
003400        *
003500        * Mnemonic values
003600       D $DqName         C                   ERS     ')
003700       D $DqLib          C                   LIB     ')
003800       D $True           C
003900       D $False          C
004000       D $Warehouse      C                   1')
004100       D $District       C
```

Name	te
CVTASCII.RPGLE	y 21, 2003

```
IBM WebSphere Development Studio Clie                RPGDQSVI
```

The Show fields option is available when the cursor is on an F-spec or COBOL Copy-DDS statement.

F-Spec Show Fields (continued)



```
 Row 29      Column 32    Replace
     .....FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++++++++++++++
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:SflRrn)
002800     FITEM      IF   E           K DISK
002900     FDSTRCT    UF   E           K DISK
003000     FCSTMR     UF A E           K DISK    COMMIT(CmtActive)
003100     FSTOCK     UF   E           K DISK    COMMIT(CmtActive)
003200     FORDERS    O  A E           K DISK    COMMIT(CmtActive)
```

File APILIB/ITEM (4 Fields)

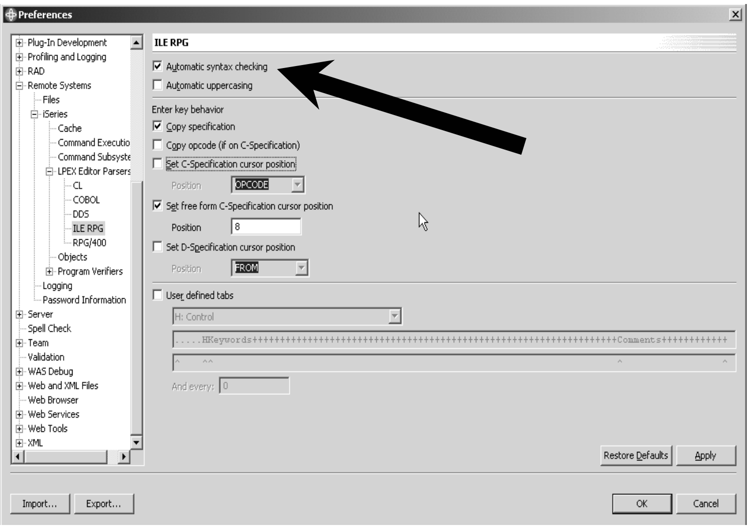
Name	Record	Type	Length	Text
IID	ITRCD	Character	6	Item ID
INAME	ITRCD	Character	24	Item Name
IPRICE	ITRCD	Packed Decimal	5.2	Price
IDATA	ITRCD	Character	50	Item Data

Properties

Property	Value
Name	ORDENTR
Number ...	0
Source	APILIB/RPGAPP
Status	OK
Subtype	SRC
Text	Parts Order Entry...

Here you see the results of the Show Fields option. This option opens a Window that shows all the fields in the referenced file.

Automatic Syntax Checking



ILE RPG

☑ Automatic syntax checking

□ Automatic uppercasing

Enter key behavior

☑ Copy specification

□ Copy opcode (if on C-Specification)

□ Set C-Specification cursor position

Position OPCODE

☑ Set free form C-Specification cursor position

Position 8

□ Set D-Specification cursor position

Position FROM

□ User defined tabs

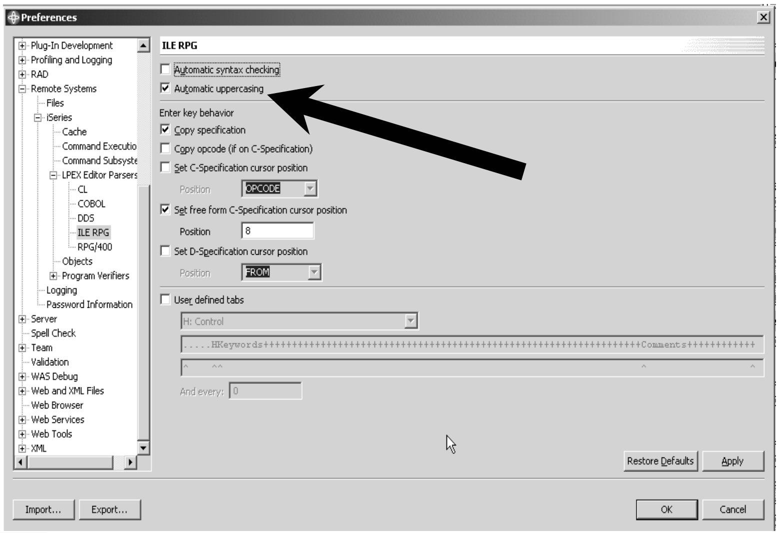
H: Control

```
.....HKeywords++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++Comments++++++++++++
^    ^^                                                                               ^                   ^
```

And every: 0

Using the Preferences window for the LPEX Editor Parsers you can set automatic syntax checking on for RPG, COBOL, DDS, CL. When automatic syntax checking is on, the parser automatically checks the syntax while you are editing.

Automatic Uppercasing



ILE RPG

□ Automatic syntax checking

☑ Automatic uppercasing

Enter key behavior

☑ Copy specification

□ Copy opcode (if on C-Specification)

□ Set C-Specification cursor position

Position OPCODE

☑ Set free form C-Specification cursor position

Position 8

□ Set D-Specification cursor position

Position FROM

□ User defined tabs

H: Control

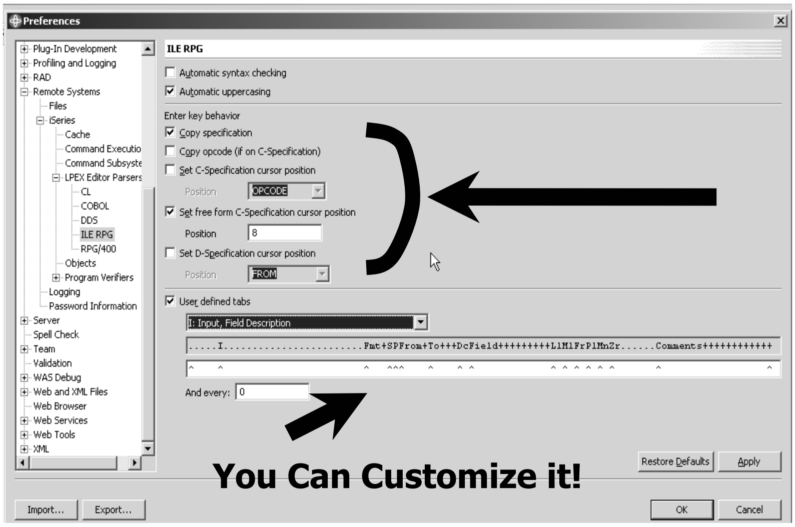
`.....HKeywords+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++Comments++++++++++++`

`^    ^^                                                                        ^                ^`

And every: 0

Through the Preferences window for LPEX Editor Parsers you can also set automatic uppercasing on for languages that expect uppercase.

Intelligent tabbing



ILE RPG

☐ Automatic syntax checking

☑ Automatic uppercasing

Enter key behavior

☑ Copy specification

☐ Copy opcode (if on C-Specification)

☐ Set C-Specification cursor position

Position: OPCODE

☑ Set free form C-Specification cursor position

Position: 8

☐ Set D-Specification cursor position

Position: FROM

☑ User defined tabs

I: Input, Field Description

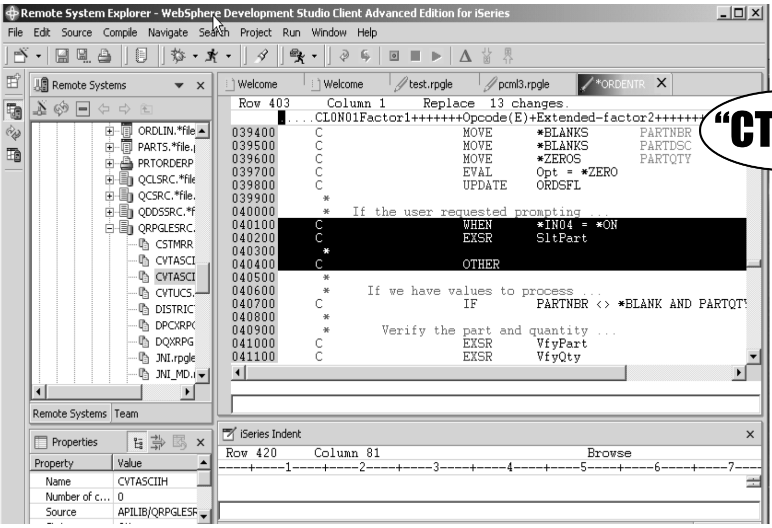
```
.....I........................Fmt+SPFrom+To+++DcField+++++++++L1M1FrP1MnZr......Comments++++++++++++
^    ^                        ^   ^^^    ^    ^ ^             ^ ^ ^ ^ ^ ^       ^                  ^
```

And every: 0

You Can Customize it!

Through the Preferences window for LPEX Editor Parsers you can also setup intelligent tabbing between columns for column-sensitive languages.

Match



Property	Value
Name	CVTASCIIH
Number of c...	0
Source	APILIB/QRPGLESR

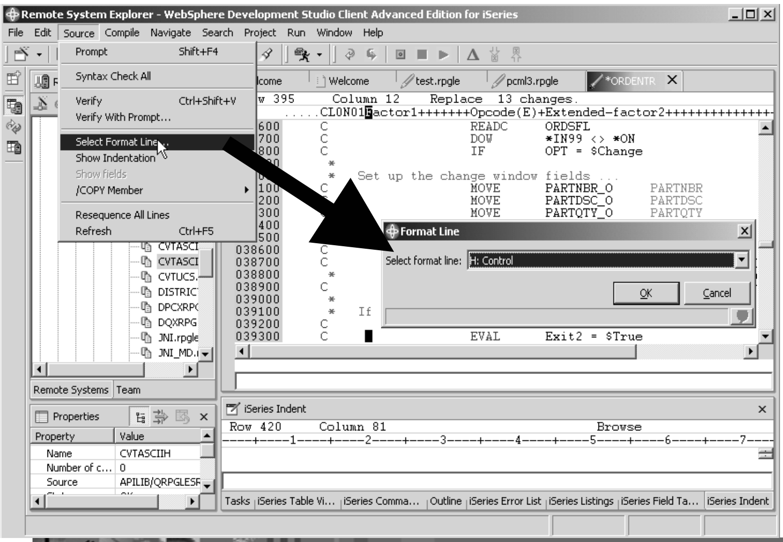
```
 Row 403     Column 1     Replace  13 changes.
       .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2+++++++
039400      C                   MOVE      *BLANKS       PARTNBR
039500      C                   MOVE      *BLANKS       PARTDSC
039600      C                   MOVE      *ZEROS        PARTQTY
039700      C                   EVAL      Opt = *ZERO
039800      C                   UPDATE    ORDSFL
039900       *
040000       *   If the user requested prompting ...
040100      C                   WHEN      *IN04 = *ON
040200      C                   EXSR      SltPart
040300       *
040400      C                   OTHER
040500       *
040600       *     If we have values to process ...
040700      C                   IF        PARTNBR <> *BLANK AND PARTQT
040800       *
040900       *       Verify the part and quantity ...
041000      C                   EXSR      VfyPart
041100      C                   EXSR      VfyQty
```

```
Row 420     Column 81                         Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

"CTRL M"

You can search for matching statements based on the cursor position!

Format Line Selection



```
w 395     Column 12    Replace  13 changes.
     .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2+++++++++++++
600       C                   READC     ORDSFL
700       C                   DOW       *IN99 <> *ON
800       C                   IF        OPT = $Change
           *
           *   Set up the change window fields ...
100       C                   MOVE      PARTNBR_O     PARTNBR
200                           MOVE      PARTDSC_O     PARTDSC
300                           MOVE      PARTQTY_O     PARTQTY
400
500
038600    C
038700    C
038800     *
038900    C
039000     *
039100     *   If
039200    C
039300    C                   EVAL      Exit2 = $True
```

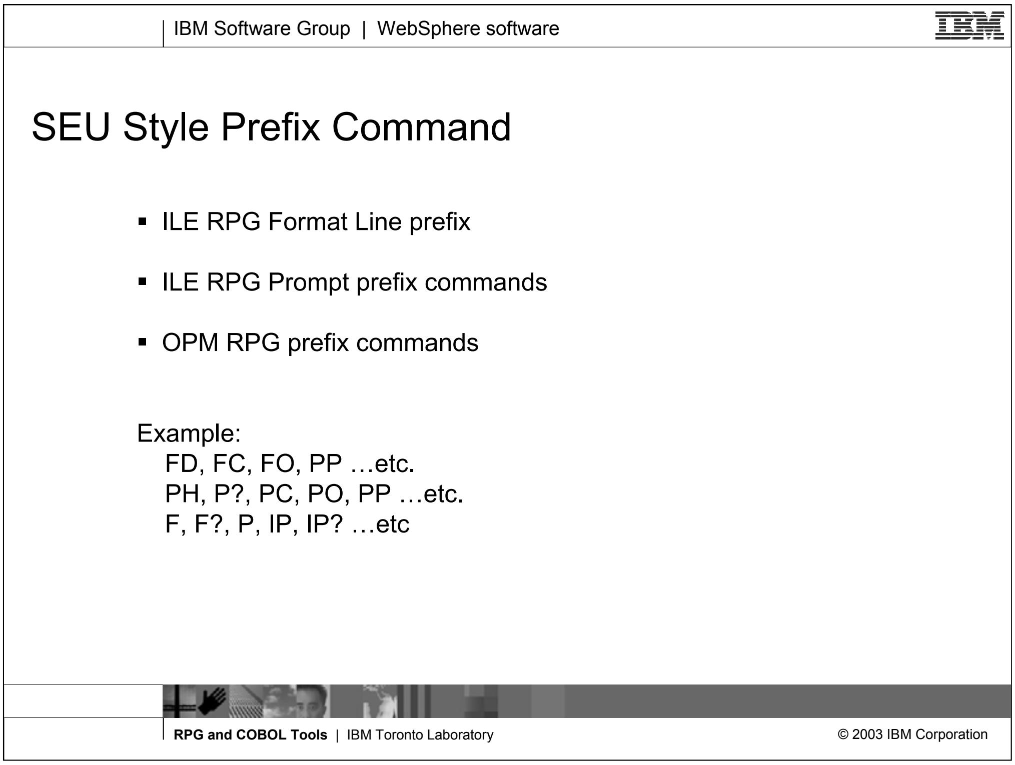
Format Line

Select format line: H: Control

Property	Value
Name	CVTASCIIH
Number of c...	0
Source	APILIB/QRPGLESF

```
Row 420       Column 81                             Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

The format line is at the top of the editor window, just above the first statement. A format line is used to help keep track of the columns in a particular specification line. The content of the format line can vary to reflect the particular type of specification being keyed such as **F** specs, **C** specs, **D** specs and so on.



SEU Style Prefix Command

- ILE RPG Format Line prefix
- ILE RPG Prompt prefix commands
- OPM RPG prefix commands

Example:
FD, FC, FO, PP …etc.
PH, P?, PC, PO, PP …etc.
F, F?, P, IP, IP? …etc

You can configure the LPEX editor to adopt the keyboard and command personalities of many popular editors. Most editor profiles differ only in the keys and commands used to perform various editor tasks. Some base editor profiles, listed below, also add a prefix information and command area at the start of each line:

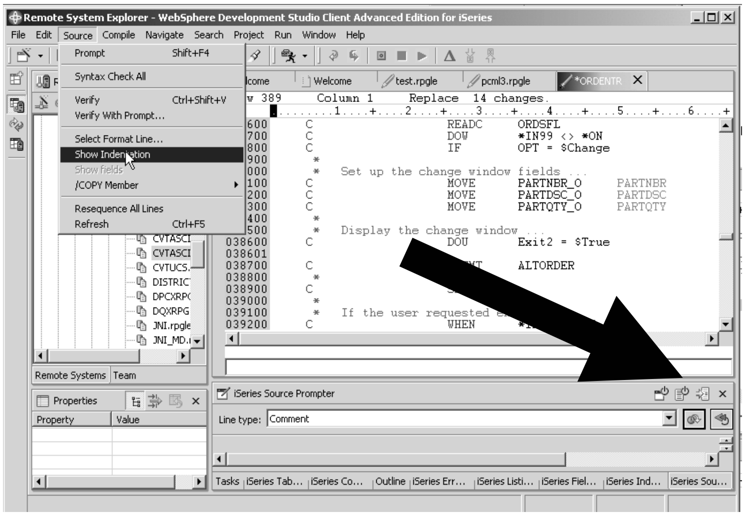
ispf

seu

xedit.

The editor recognizes prefix commands used by these editor profiles. Depending on which profile you are using, you can enter SEU, XEDIT, or ISPF commands when the prefix area is active.

Indent View

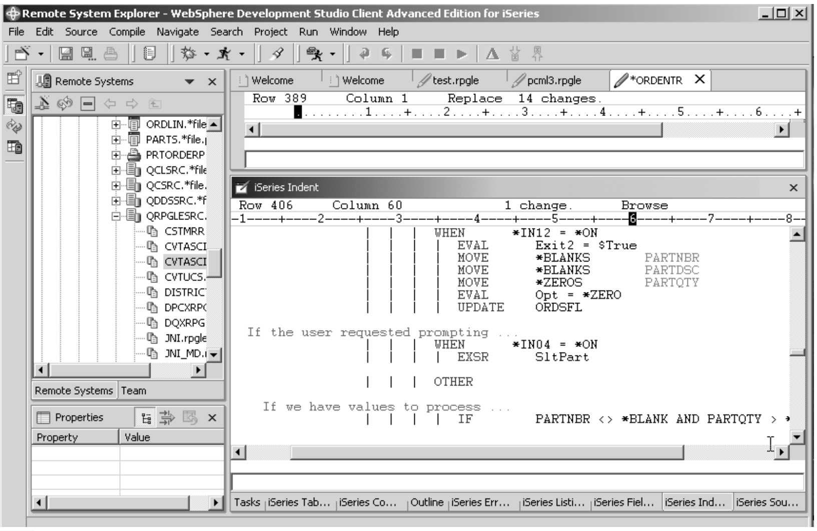


```
w 389      Column 1      Replace  14 changes.
 ........1....+....2....+....3....+....4....+....5....+....6....+
600      C                   READC     ORDSFL
700      C                   DOW       *IN99 <> *ON
800      C                   IF        OPT = $Change
900       *
000       *   Set up the change window fields ...
100      C                   MOVE      PARTNBR_O     PARTNBR
200      C                   MOVE      PARTDSC_O     PARTDSC
300      C                   MOVE      PARTQTY_O     PARTQTY
400       *
500       *   Display the change window ...
038600   C                   DOU       Exit2 = $True
038601
038700   C                     MT      ALTORDER
038800    *
038900   C
039000    *
039100    *   If the user requested e
039200   C                   WHEN      *I
```

Line type: Comment

When editing ILE RPG source, it can be difficult to determine the beginning and ending of constructs. The indent option allows you to view your source with constructs in an indented mode. By default, the indent option will split the screen horizontally and show the indented view in the bottom pane.

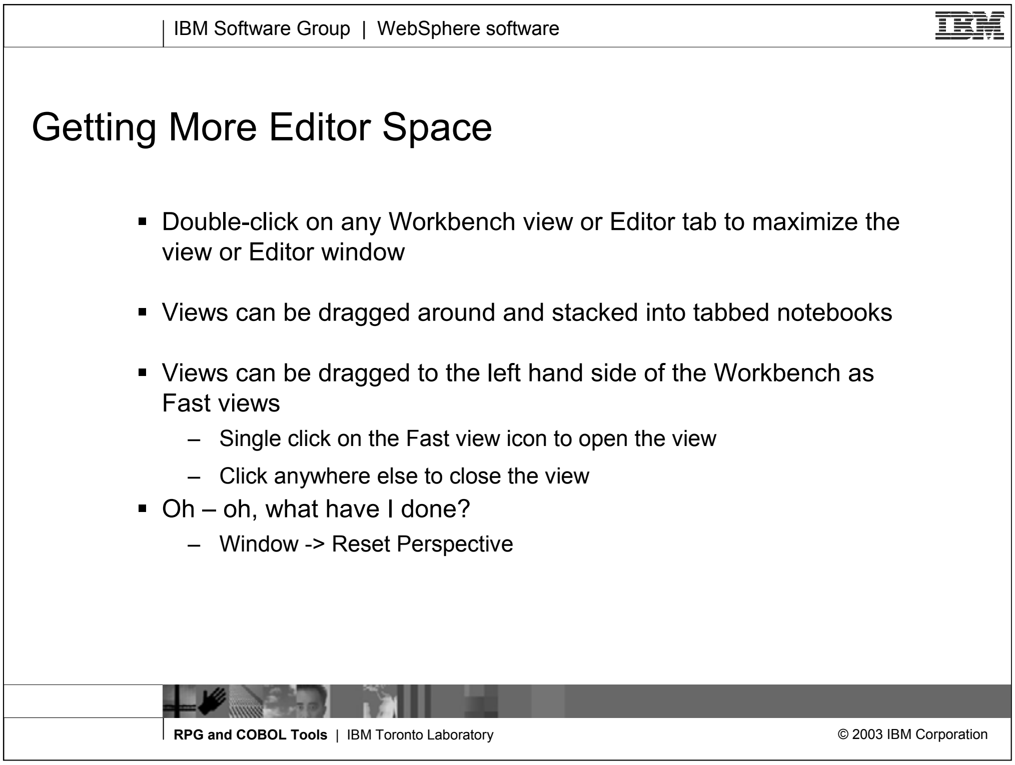
Indent View (continued)


Remote System Explorer - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Source Compile Navigate Search Project Run Window Help
Remote Systems
ORDLIN.*file
PARTS.*file.
PRTORDERP
QCLSRC.*file
QCSRC.*file.
QDDSSRC.*f
QRPGLESRC.
CSTMRR
CVTASCI
CVTASCI
CVTUCS.
DISTRIC
DPCXRPC
DQXRPG
JNI.rpgle
JNI_MD.
Remote Systems Team
Properties
Property Value
Welcome Welcome test.rpgle pcml3.rpgle *ORDENTR
iSeries Indent
Tasks iSeries Tab... iSeries Co... Outline iSeries Err... iSeries Listi... iSeries Fiel... iSeries Ind... iSeries Sou...

```
 Row 389      Column 1      Replace  14 changes.
```

```
Row 406      Column 60             1 change.       Browse
-1----+----2----+----3----+----4----+----5----+----6----+----7----+----8--
                 |  |  |  WHEN      *IN12 = *ON
                 |  |  |  |  EVAL      Exit2 = $True
                 |  |  |  |  MOVE      *BLANKS       PARTNBR
                 |  |  |  |  MOVE      *BLANKS       PARTDSC
                 |  |  |  |  MOVE      *ZEROS        PARTQTY
                 |  |  |  |  EVAL      Opt = *ZERO
                 |  |  |  |  UPDATE    ORDSFL

  If the user requested prompting ...
                 |  |  |  WHEN      *IN04 = *ON
                 |  |  |  |  EXSR      SltPart

                 |  |  |  OTHER

    If we have values to process ...
                 |  |  |  |  IF        PARTNBR <> *BLANK AND PARTQTY > *
```

Here you see the results of the indent option. The source is now showing indentation.

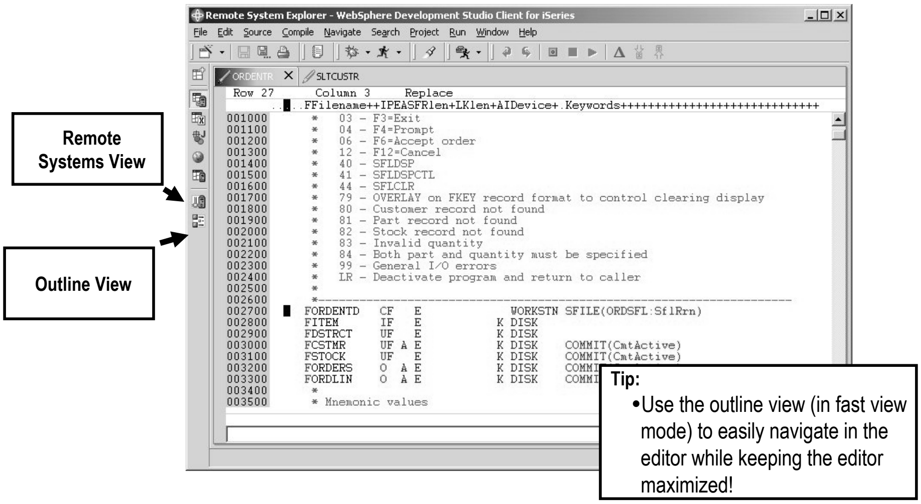


Getting More Editor Space

- Double-click on any Workbench view or Editor tab to maximize the view or Editor window
- Views can be dragged around and stacked into tabbed notebooks
- Views can be dragged to the left hand side of the Workbench as Fast views
  - Single click on the Fast view icon to open the view
  - Click anywhere else to close the view
- Oh – oh, what have I done?
  - Window -> Reset Perspective

You can easily double click on a view or editor tab to maximize the view or Editor window. If you double-click again you will return the view or Editor window to its original size. You can easily rearrange views by selecting hem and dragging them to another location in the workbench. If you are going to use a view frequently but don't ant to see it all the time in the workbench you can make it a Fast view. You select the view, right click and select ast view. The view will then appear in the left hand frame of the Workbench as a Fast view icon.
To reset your workbench window to its original layout, you can select Window -> Reset perspective.

Fast View


Remote Systems View
Outline View
Remote System Explorer - WebSphere Development Studio Client for iSeries
File Edit Source Compile Navigate Search Project Run Window Help
ORDENTR
SLTCUSTR

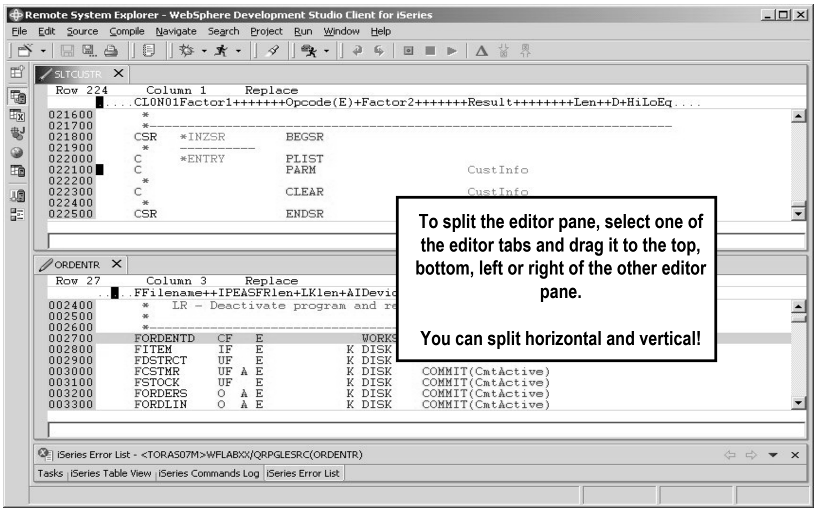
```
 Row 27      Column 3      Replace
 ...FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++++++++++++++++++++++++++
001000       *   03 - F3=Exit
001100       *   04 - F4=Prompt
001200       *   06 - F6=Accept order
001300       *   12 - F12=Cancel
001400       *   40 - SFLDSP
001500       *   41 - 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:SflRrn)
002800      FITEM      IF   E           K DISK
002900      FDSTRCT    UF   E           K DISK
003000      FCSTMR     UF A E           K DISK    COMMIT(CmtActive)
003100      FSTOCK     UF   E           K DISK    COMMIT(CmtActive)
003200      FORDERS    O  A E           K DISK    COMMI
003300      FORDLIN    O  A E           K DISK    COMMI
003400       *
003500       * Mnemonic values
```

Tip:

- Use the outline view (in fast view mode) to easily navigate in the editor while keeping the editor maximized!

Here you see ane example of the Remote Systems view and the Outline view as Fast views in the workbench left hand frame.

Split Screen Editing



Remote System Explorer - WebSphere Development Studio Client for iSeries

```
Row 224      Column 1      Replace
       ....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq....
021600        *
021700        *---------------------------------------------------------------------------
021800      CSR    *INZSR         BEGSR
021900        *    ----------
022000      C      *ENTRY         PLIST
022100      C                     PARM                          CustInfo
022200        *
022300      C                     CLEAR                         CustInfo
022400        *
022500      CSR                   ENDSR
```

```
Row 27       Column 3      Replace
     ....FFilename++IPEASFRlen+LKlen+AIDevic
002400        *    LR - Deactivate program and re
002500        *
002600        *-----------------------------------
002700      FORDENTD   CF   E              WORKS
002800      FITEM      IF   E            K DISK
002900      FDSTRCT    UF   E            K DISK
003000      FCSTMR     UF A E            K DISK     COMMIT(CmtActive)
003100      FSTOCK     UF   E            K DISK     COMMIT(CmtActive)
003200      FORDERS    O  A E            K DISK     COMMIT(CmtActive)
003300      FORDLIN    O  A E            K DISK     COMMIT(CmtActive)
```

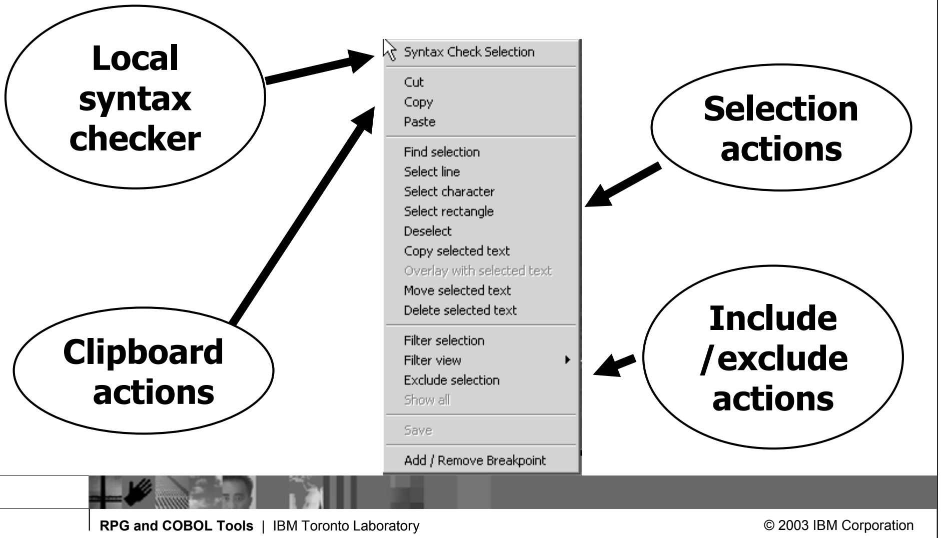
iSeries Error List - <TORAS07M>WFLABXX/QRPGLESRC(ORDENTR)

To split the editor pane, select one of the editor tabs and drag it to the top, bottom, left or right of the other editor pane.

You can split horizontal and vertical!

You can easily split the Editor window to see two files at the same time. You can also do this by dragging views or using an Editor menu option.

LPEX Editor Actions



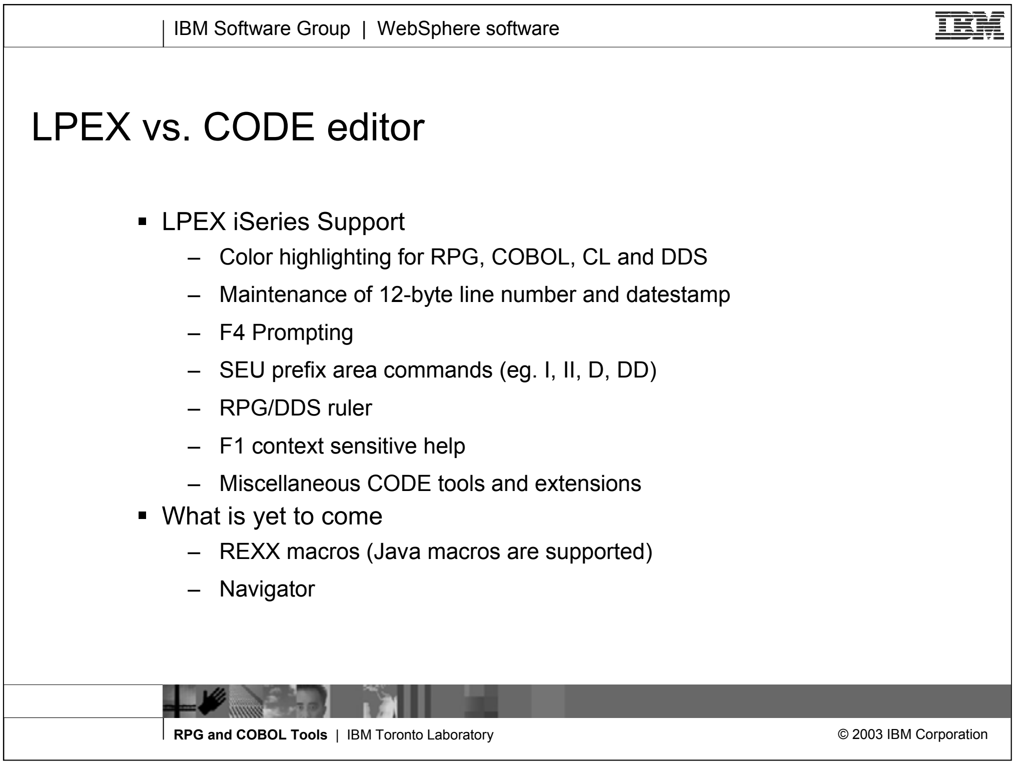
Local syntax checker

Selection actions

Clipboard actions

Include /exclude actions

Here you can see how easy it is to select actions against your source. Right-click and you see this pop-up with a rich set of actions.

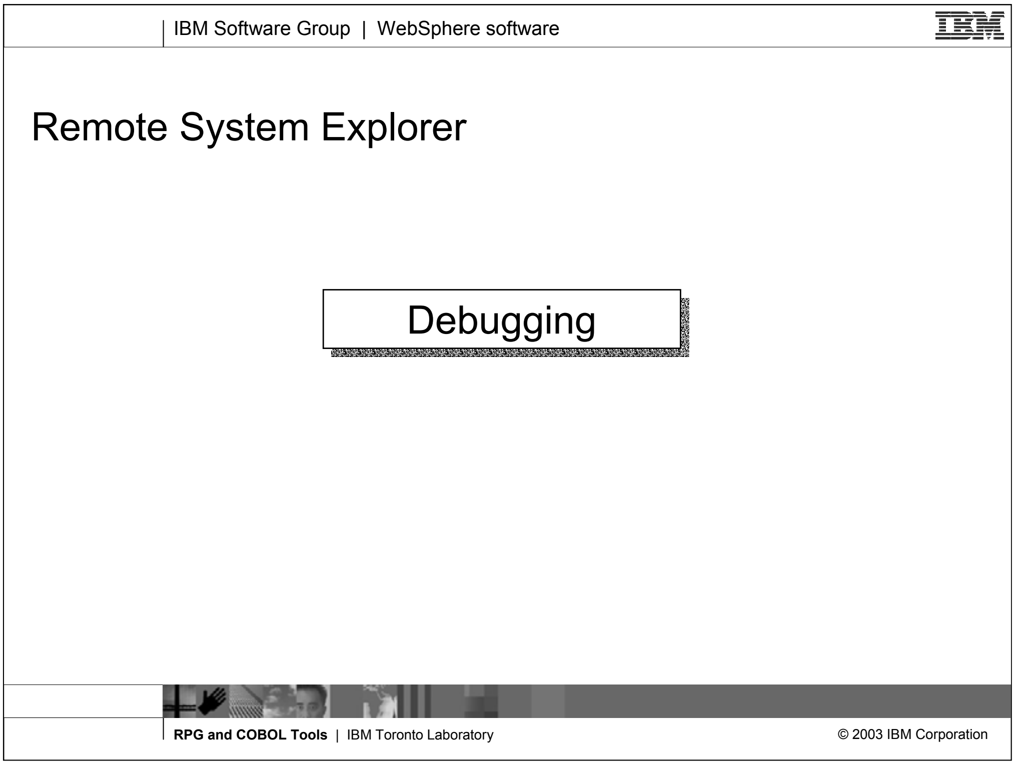


LPEX vs. CODE editor

- LPEX iSeries Support
  - Color highlighting for RPG, COBOL, CL and DDS
  - Maintenance of 12-byte line number and datestamp
  - F4 Prompting
  - SEU prefix area commands (eg. I, II, D, DD)
  - RPG/DDS ruler
  - F1 context sensitive help
  - Miscellaneous CODE tools and extensions
- What is yet to come
  - REXX macros (Java macros are supported)
  - Navigator

The LPEX editor in Development Studio Client is a Java rewrite of the classic CODE editor that is written in C++. The CODE editor has 10 years of evolution behind it, and not all of its functionality is available in LPEX today. However, the LPEX editor still does have a rich base of function, as shown here. In addition to the base editor function, there is additional support for each programming language supported by the editor.

The language-specific functionality currently available in LPEX is listed here. However, there is still a number of functions in the code editor that are yet to come in LPEX. These too are listed here. Many of these will come in the next release.

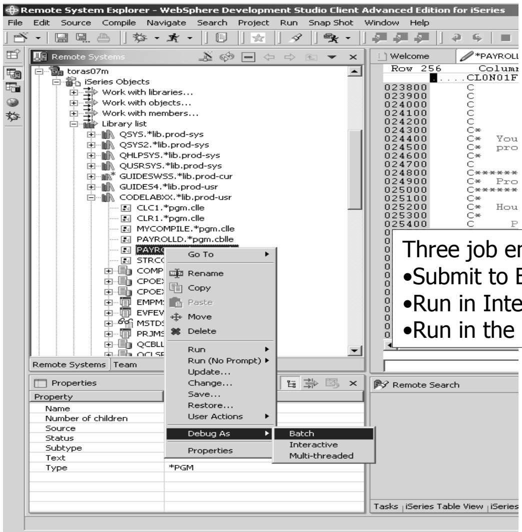


Remote System Explorer

Debugging

With the integrated iSeries debugger you can debug your program running on the iSeries host from a graphical user interface on your workstation. You can also set breakpoints before running the debugger, by inserting breakpoints directly in your source while editing. The integrated iSeries debugger client user interface also enables you to control program execution. For example, you can run your program, set line, watch, and service entry point breakpoints, step through program instructions, examine variables, and examine the call stack. You can also debug multiple applications, which may be written in different languages, from a single debugger window. Each session you debug is listed separately in the Debug view.

Debugging – Job environments



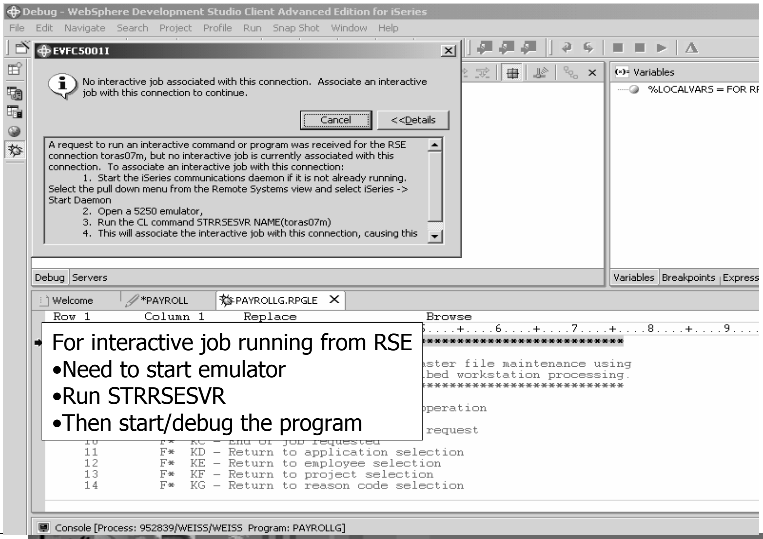
Three job environments to choose from:
- Submit to Batch
- Run in Interactive (need 5250)
- Run in the RSE job

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

- In the Remote System Explorer communications server job
- In a batch job
- In an interactive job

Using the first option lets you run the program in the same job as the communications server. With batch and interactive jobs, you cannot monitor the status as easily, however, you do not tie up your communications server and you are notified when the program command ends. Batch jobs work as you would expect and do not require any initial setup.

Debugging with the integrated Debugger


Debug - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Navigate Search Project Profile Run Snap Shot Window Help
EVFC5001I
Cancel
<<Details
Variables
%LOCALVARS = FOR RP
Debug
Servers
Variables
Breakpoints
Express
Welcome
*PAYROLL
PAYROLLG.RPGLE

No interactive job associated with this connection. Associate an interactive job with this connection to continue.

A request to run an interactive command or program was received for the RSE connection toras07m, but no interactive job is currently associated with this connection. To associate an interactive job with this connection:

1. Start the iSeries communications daemon if it is not already running. Select the pull down menu from the Remote Systems view and select iSeries -> Start Daemon
2. Open a 5250 emulator,
3. Run the CL command STRRSESVR NAME(toras07m)
4. This will associate the interactive job with this connection, causing this

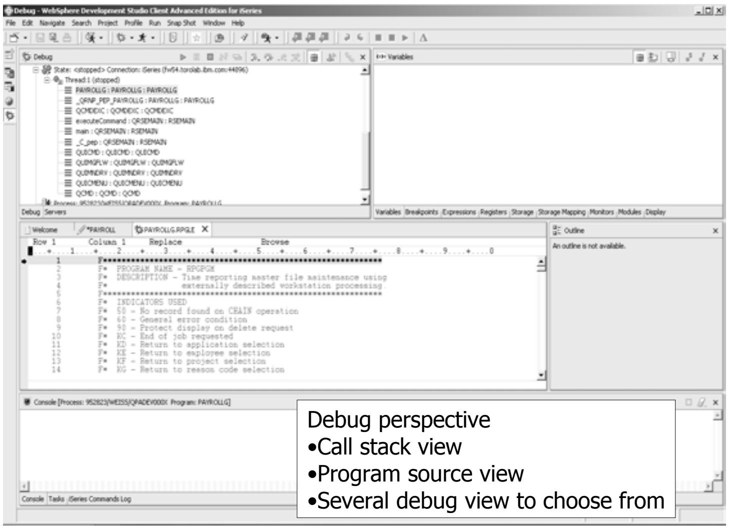
```
Row 1        Column 1     Replace                   Browse
                                                   5....+....6....+....7....+....8....+....9....
                                                   ***************************
                                                   aster file maintenance using
                                                   ibed workstation processing.
                                                   ***************************
                                                   operation
                                                   request
    11         F*  KD - Return to application selection
    12         F*  KE - Return to employee selection
    13         F*  KF - Return to project selection
    14         F*  KG - Return to reason code selection
```

Console [Process: 952839/WEISS/WEISS Program: PAYROLLG]

For interactive job running from RSE
- Need to start emulator
- Run STRRSESVR
- Then start/debug the program

Interactive programs require a 5250 emulator, so you need to first run a STRRSESVR <connectionName> command to associate the emulator with a particular connection in the Remote System Explorer communications server.

Debug perspective



Debug - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Variables
Debug Servers
Variables Breakpoints Expressions Registers Storage Storage Mapping Monitors Modules Display
Welcome
*PAYROLL
PAYROLLG.RPGLE
Outline
An outline is not available.
Console Tasks iSeries Commands Log

```
State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:44896)
  Thread:1 (stopped)
    PAYROLLG : PAYROLLG : PAYROLLG
    _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
    QCMDEXC : QCMDEXC : QCMDEXC
    executeCommand : QRSEMAIN : RSEMAIN
    main : QRSEMAIN : RSEMAIN
    _C_pep : QRSEMAIN : RSEMAIN
    QUICMD : QUICMD : QUICMD
    QUIMGFLW : QUIMGFLW : QUIMGFLW
    QUIMNDRV : QUIMNDRV : QUIMNDRV
    QUICMENU : QUICMENU : QUICMENU
    QCMD : QCMD : QCMD
```

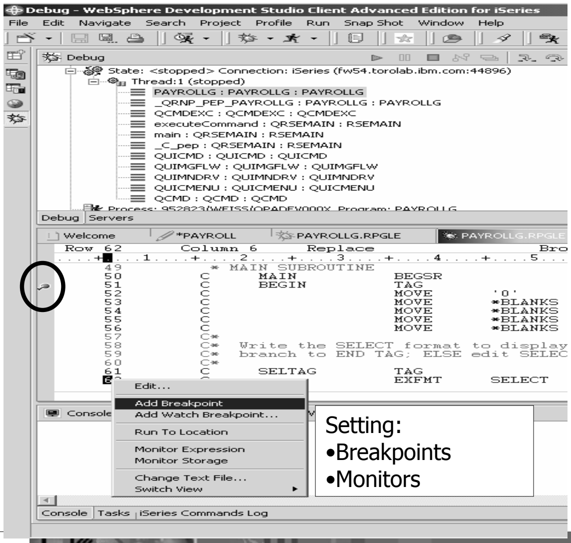
```
Row 1        Column 1     Replace                    Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
F**************************************************************
F*  PROGRAM NAME - RPGPGM
F*  DESCRIPTION - Time reporting master file maintenance using
F*                externally described workstation processing.
F**************************************************************
F*  INDICATORS USED
F*  50 - No record found on CHAIN operation
F*  60 - General error condition
F*  90 - Protect display on delete request
F*  KC - End of job requested
F*  KD - Return to application selection
F*  KE - Return to employee selection
F*  KF - Return to project selection
F*  KG - Return to reason code selection
```

Console [Process: 952823/WEISS/QPADEV000X Program: PAYROLLG]

Debug perspective
- Call stack view
- Program source view
- Several debug view to choose from

The Debug perspective contains the tools and views to debug a program. It opens when you run a program to debug. Here you see the call stack and source view. There are several other views you can choose. For example, variables, breakpoints, expressions.

Setting breakpoints



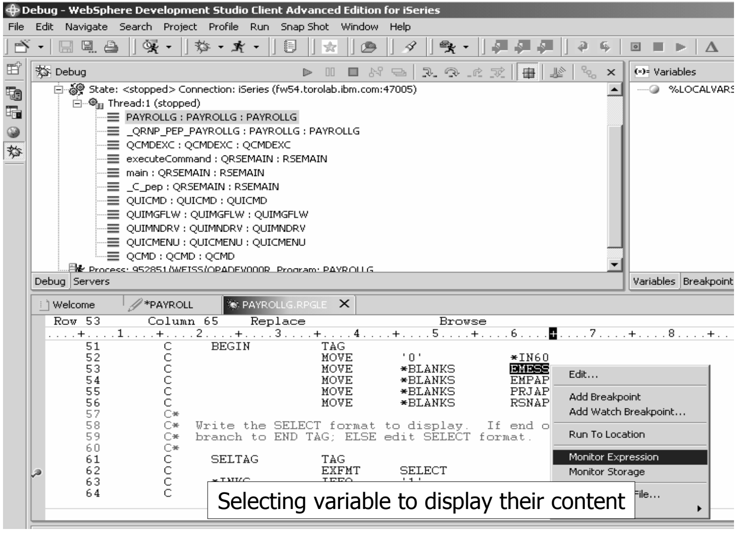
```
State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:44896)
   Thread:1 (stopped)
      PAYROLLG : PAYROLLG : PAYROLLG
      _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
      QCMDEXC : QCMDEXC : QCMDEXC
      executeCommand : QRSEMAIN : RSEMAIN
      main : QRSEMAIN : RSEMAIN
      _C_pep : QRSEMAIN : RSEMAIN
      QUICMD : QUICMD : QUICMD
      QUIMGFLW : QUIMGFLW : QUIMGFLW
      QUIMNDRV : QUIMNDRV : QUIMNDRV
      QUICMENU : QUICMENU : QUICMENU
      QCMD : QCMD : QCMD
Process: 952823/WEISS/QPADEV000X Program: PAYROLLG
```

```
Row 62         Column 6       Replace                      Bro
....+....1....+....2....+....3....+....4....+....5...
           * MAIN SUBROUTINE
          C     MAIN          BEGSR
          C     BEGIN         TAG
          C                   MOVE      '0'
          C                   MOVE      *BLANKS
          C                   MOVE      *BLANKS
          C                   MOVE      *BLANKS
          C                   MOVE      *BLANKS
          C*
          C*   Write the SELECT format to display
          C*   branch to END TAG; ELSE edit SELEC
          C*
          C     SELTAG        TAG
                              EXFMT     SELECT
```

Setting:
- Breakpoints
- Monitors

You can only set breakpoints at executable lines. All executables lines are displayed in blue.
You can also monitor variables in the Monitors view.

Display Variable Content



```
State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:47005)
  Thread:1 (stopped)
    PAYROLLG : PAYROLLG : PAYROLLG
    _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
    QCMDEXC : QCMDEXC : QCMDEXC
    executeCommand : QRSEMAIN : RSEMAIN
    main : QRSEMAIN : RSEMAIN
    _C_pep : QRSEMAIN : RSEMAIN
    QUICMD : QUICMD : QUICMD
    QUIMGFLW : QUIMGFLW : QUIMGFLW
    QUIMNDRV : QUIMNDRV : QUIMNDRV
    QUICMENU : QUICMENU : QUICMENU
    QCMD : QCMD : QCMD
Process: 952851/WEISS/QPADEV000R Program: PAYROLLG
```

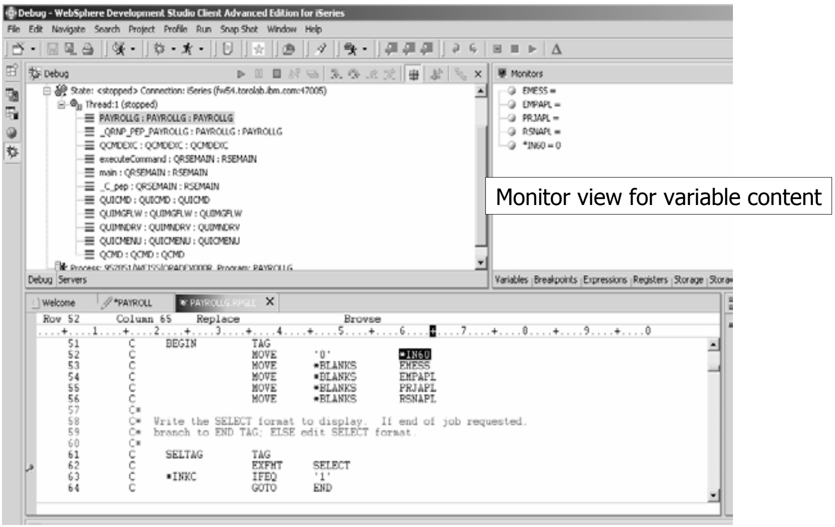
```
%LOCALVARS
```

```
Row 53      Column 65    Replace               Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+..
       C     BEGIN         TAG
       C                   MOVE      '0'        *IN60
       C                   MOVE      *BLANKS    EMESS
       C                   MOVE      *BLANKS    EMPAP
       C                   MOVE      *BLANKS    PRJAP
       C                   MOVE      *BLANKS    RSNAP
       C*
       C*  Write the SELECT format to display.  If end o
       C*  branch to END TAG; ELSE edit SELECT format.
       C*
       C     SELTAG        TAG
       C                   EXFMT     SELECT
       C     *INKC         IFEQ      '1'
       C
```

Selecting variable to display their content

To monitor a variable you use the Monitor Expression menu option.

Display Variable Content (continued)


Debug - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Monitors
Debug Servers
Variables Breakpoints Expressions Registers Storage Stora
Welcome
*PAYROLL
PAYROLLG.RPGLE

```
State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:47005)
  Thread:1 (stopped)
    PAYROLLG : PAYROLLG : PAYROLLG
    _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
    QCMDEXC : QCMDEXC : QCMDEXC
    executeCommand : QRSEMAIN : RSEMAIN
    main : QRSEMAIN : RSEMAIN
    _C_pep : QRSEMAIN : RSEMAIN
    QUICMD : QUICMD : QUICMD
    QUIMGFLW : QUIMGFLW : QUIMGFLW
    QUIMNDRV : QUIMNDRV : QUIMNDRV
    QUICMENU : QUICMENU : QUICMENU
    QCMD : QCMD : QCMD
Process: 952051/WEISS/QPADEV000R  Program: PAYROLLG
```

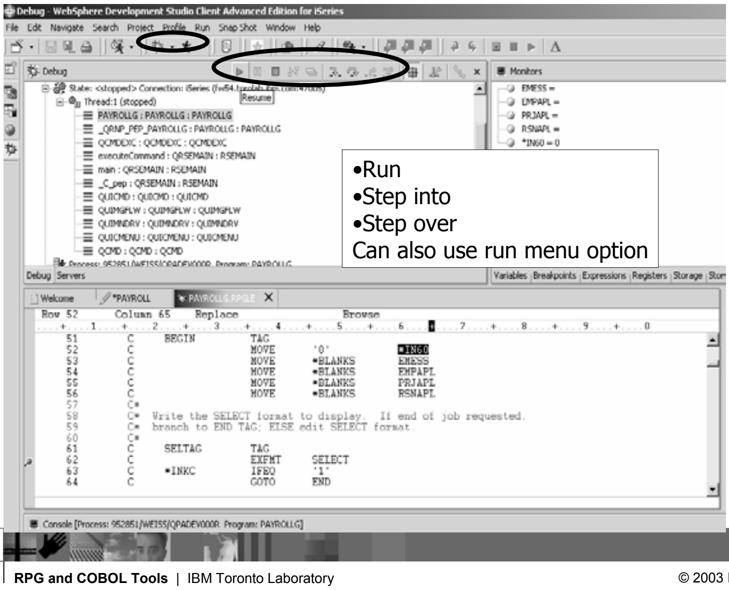
```
EMESS =
EMPAPL =
PRJAPL =
RSNAPL =
*IN60 = 0
```

Monitor view for variable content

```
 Row 52       Column 65    Replace                  Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
      51         C     BEGIN         TAG
      52         C                   MOVE      '0'          *IN60
      53         C                   MOVE      *BLANKS      EMESS
      54         C                   MOVE      *BLANKS      EMPAPL
      55         C                   MOVE      *BLANKS      PRJAPL
      56         C                   MOVE      *BLANKS      RSNAPL
      57         C*
      58         C*  Write the SELECT format to display.  If end of job requested.
      59         C*  branch to END TAG; ELSE edit SELECT format.
      60         C*
      61         C     SELTAG        TAG
      62         C                   EXFMT     SELECT
      63         C     *INKC         IFEQ      '1'
      64         C                   GOTO      END
```

The Monitors view shows the variable content.

Stepping through the program


Debug - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Resume
Monitors
Debug Servers
Variables Breakpoints Expressions Registers Storage Stor
Welcome
*PAYROLL
PAYROLLG.RPGLE

```
State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:4700)
  Thread:1 (stopped)
    PAYROLLG : PAYROLLG : PAYROLLG
    _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
    QCMDEXC : QCMDEXC : QCMDEXC
    executeCommand : QRSEMAIN : RSEMAIN
    main : QRSEMAIN : RSEMAIN
    _C_pep : QRSEMAIN : RSEMAIN
    QUICMD : QUICMD : QUICMD
    QUIMGFLW : QUIMGFLW : QUIMGFLW
    QUIMNDRV : QUIMNDRV : QUIMNDRV
    QUICMENU : QUICMENU : QUICMENU
    QCMD : QCMD : QCMD
Process: 952851/WEISS/QPADEV000R  Program: PAYROLLG
```

```
EMESS =
EMPAPL =
PRJAPL =
RSNAPL =
*IN60 = 0
```

```
Row 52      Column 65    Replace                Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
          C     BEGIN         TAG
          C                   MOVE      '0'           *IN60
          C                   MOVE      *BLANKS       EMESS
          C                   MOVE      *BLANKS       EMPAPL
          C                   MOVE      *BLANKS       PRJAPL
          C                   MOVE      *BLANKS       RSNAPL
          C*
          C*  Write the SELECT format to display.  If end of job requested.
          C*  branch to END TAG; ELSE edit SELECT format.
          C*
          C     SELTAG        TAG
          C                   EXFMT     SELECT
          C     *INKC         IFEQ      '1'
          C                   GOTO      END
```

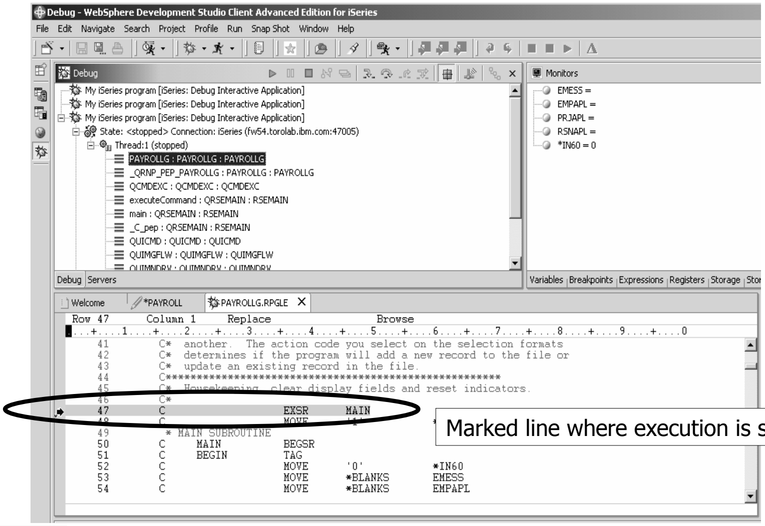
Console [Process: 952851/WEISS/QPADEV000R Program: PAYROLLG]

- •Run
- •Step into
- •Step over

Can also use run menu option

The Debugger allows you to step over a program call or step into it . When you step over a program call, the called program runs and the debugger stops at the next executable statement in the calling program. You can also use the Run option from the menu toolbar to step through the program.

Stopping at a Breakpoint



Debug - WebSphere Development Studio Client Advanced Edition for iSeries

File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Monitors

- My iSeries program [iSeries: Debug Interactive Application]
- My iSeries program [iSeries: Debug Interactive Application]
- My iSeries program [iSeries: Debug Interactive Application]
  - State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:47005)
    - Thread:1 (stopped)
      - PAYROLLG : PAYROLLG : PAYROLLG
      - _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
      - QCMDEXC : QCMDEXC : QCMDEXC
      - executeCommand : QRSEMAIN : RSEMAIN
      - main : QRSEMAIN : RSEMAIN
      - _C_pep : QRSEMAIN : RSEMAIN
      - QUICMD : QUICMD : QUICMD
      - QUIMGFLW : QUIMGFLW : QUIMGFLW
      - QUIMNDRV : QUIMNDRV : QUIMNDRV

```
EMESS =
EMPAPL =
PRJAPL =
RSNAPL =
*IN60 = 0
```

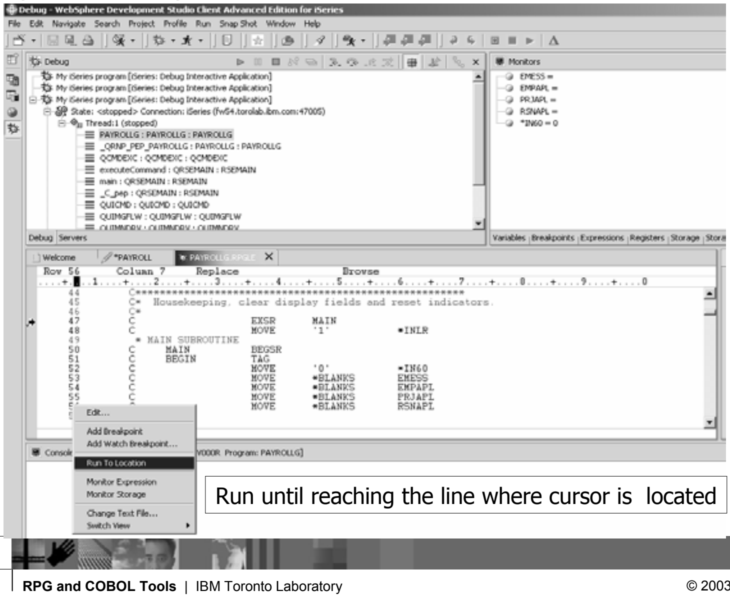
Debug Servers
Variables Breakpoints Expressions Registers Storage Stor
Welcome *PAYROLL PAYROLLG.RPGLE

```
Row 47       Column 1      Replace                   Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
         C*  another.  The action code you select on the selection formats
         C*  determines if the program will add a new record to the file or
         C*  update an existing record in the file.
         C*****************************************************
         C*  Housekeeping, clear display fields and reset indicators.
         C*
         C                       EXSR      MAIN
         C                       MOVE      '1'
          * MAIN SUBROUTINE
         C     MAIN              BEGSR
         C     BEGIN             TAG
         C                       MOVE      '0'           *IN60
         C                       MOVE      *BLANKS       EMESS
         C                       MOVE      *BLANKS       EMPAPL
```

Marked line where execution is stopped

The program starts running and stops when at the first breakpoint.

Stopping the program at the cursor



Debug - WebSphere Development Studio Client Advanced Edition for iSeries
File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Monitors
Debug Servers
Variables Breakpoints Expressions Registers Storage Stora
Welcome
*PAYROLL
PAYROLLG.RPGLE
Console
Edit...
Add Breakpoint
Add Watch Breakpoint...
Run To Location
Monitor Expression
Monitor Storage
Change Text File...
Switch View

```
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
  State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:47005)
    Thread:1 (stopped)
      PAYROLLG : PAYROLLG : PAYROLLG
      _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
      QCMDEXC : QCMDEXC : QCMDEXC
      executeCommand : QRSEMAIN : RSEMAIN
      main : QRSEMAIN : RSEMAIN
      _C_pep : QRSEMAIN : RSEMAIN
      QUICMD : QUICMD : QUICMD
      QUIMGFLW : QUIMGFLW : QUIMGFLW
      QUIMNDRV : QUIMNDRV : QUIMNDRV
```

```
EMESS =
EMPAPL =
PRJAPL =
RSNAPL =
*IN60 = 0
```

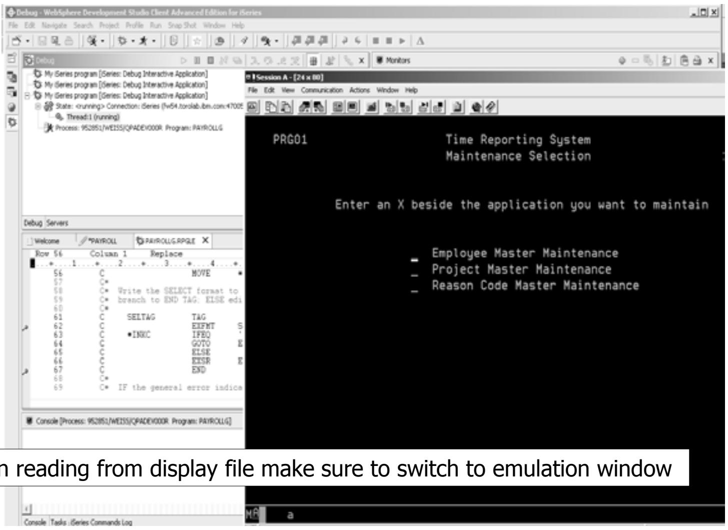
```
Row 56      Column 7     Replace                 Browse
....+.█..1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
         C****************************************************
         C*  Housekeeping, clear display fields and reset indicators.
         C*
         C                       EXSR      MAIN
         C                       MOVE      '1'          *INLR
          * MAIN SUBROUTINE
         C     MAIN              BEGSR
         C     BEGIN             TAG
         C                       MOVE      '0'          *IN60
         C                       MOVE      *BLANKS      EMESS
         C                       MOVE      *BLANKS      EMPAPL
         C                       MOVE      *BLANKS      PRJAPL
         C                       MOVE      *BLANKS      RSNAPL
```

```
V000R Program: PAYROLLG]
```

Run until reaching the line where cursor is located

You can also run the program to stop at a selected line in the source.

Program output to emulation window



Debug - WebSphere Development Studio Client Advanced Edition for iSeries

```
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
  State: <running> Connection: iSeries (fw54.torolab.ibm.com:4700
    Thread:1 (running)
  Process: 952851/WEISS/QPADEV000R Program: PAYROLLG
```

```
Row 56      Column 1      Replace
 ...+....1....+....2....+....3....+....4....+.
      56        C                     MOVE      *
      57        C*
      58        C*  Write the SELECT format to
      59        C*  branch to END TAG. ELSE edi
      60        C*
      61        C     SELTAG          TAG
      62        C                     EXFMT     S
      63        C     *INKC           IFEQ      '
      64        C                     GOTO      E
      65        C                     ELSE
      66        C                     EXSR      E
      67        C                     END
      68        C*
      69        C*  IF the general error indica
```

Console [Process: 952851/WEISS/QPADEV000R Program: PAYROLLG]

Session A - [24 x 80]

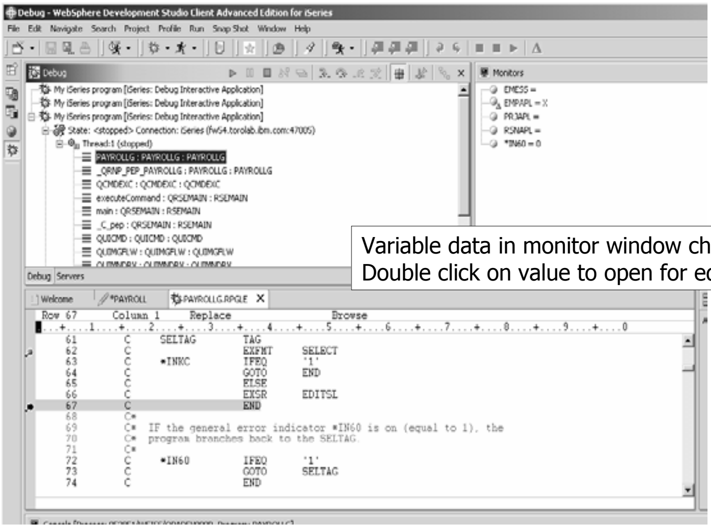
```
PRG01                      Time Reporting System
                           Maintenance Selection


          Enter an X beside the application you want to maintain


                       _  Employee Master Maintenance
                       _  Project Master Maintenance
                       _  Reason Code Master Maintenance
```

File Edit Navigate Search Project Profile Run Snap Shot Window Help
Debug
Monitors
Debug Servers
Welcome
*PAYROLL
PAYROLLG.RPGLE
File Edit View Communication Actions Window Help
MA a
Console Tasks iSeries Commands Log

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

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

Editing variable data



Debug - WebSphere Development Studio Client Advanced Edition for iSeries

```
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
My iSeries program [iSeries: Debug Interactive Application]
  State: <stopped> Connection: iSeries (fw54.torolab.ibm.com:47005)
    Thread:1 (stopped)
      PAYROLLG : PAYROLLG : PAYROLLG
      _QRNP_PEP_PAYROLLG : PAYROLLG : PAYROLLG
      QCMDEXC : QCMDEXC : QCMDEXC
      executeCommand : QRSEMAIN : RSEMAIN
      main : QRSEMAIN : RSEMAIN
      _C_pep : QRSEMAIN : RSEMAIN
      QUICMD : QUICMD : QUICMD
      QUIMGFLW : QUIMGFLW : QUIMGFLW
```

Monitors

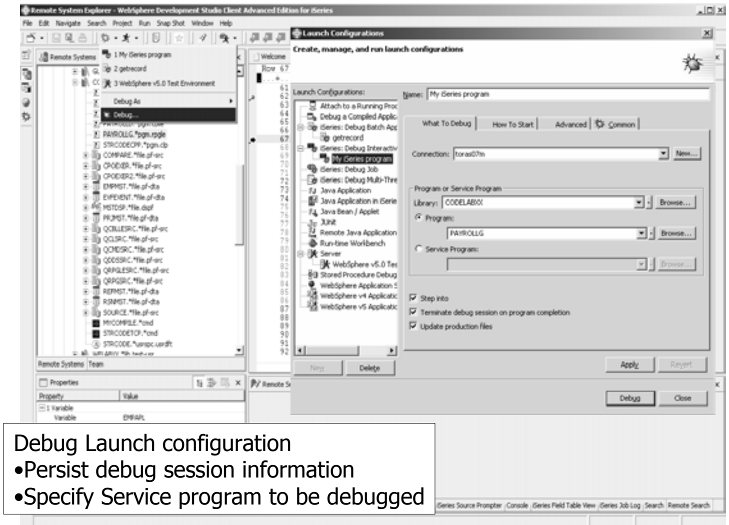
```
EMESS =
EMPAPL = X
PRJAPL =
RSNAPL =
*IN60 = 0
```

```
Row 67      Column 1      Replace                 Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
        C     SELTAG        TAG
        C                   EXFMT     SELECT
        C     *INKC         IFEQ      '1'
        C                   GOTO      END
        C                   ELSE
        C                   EXSR      EDITSL
        C                   END
        C*
        C*  IF the general error indicator *IN60 is on (equal to 1), the
        C*  program branches back to the SELTAG.
        C*
        C     *IN60         IFEQ      '1'
        C                   GOTO      SELTAG
        C                   END
```

Variable data in monitor window changed
Double click on value to open for edit

When the value of a variable changes, it will be highlighted in red. You can change the value of a variable while debugging by double clicking on the value in the Monitors view.

Configure the Debugger



Name: My iSeries program

Connection: toras07m

Program or Service Program

Library: CODELABXX

● Program: PAYROLLG

○ Service Program:

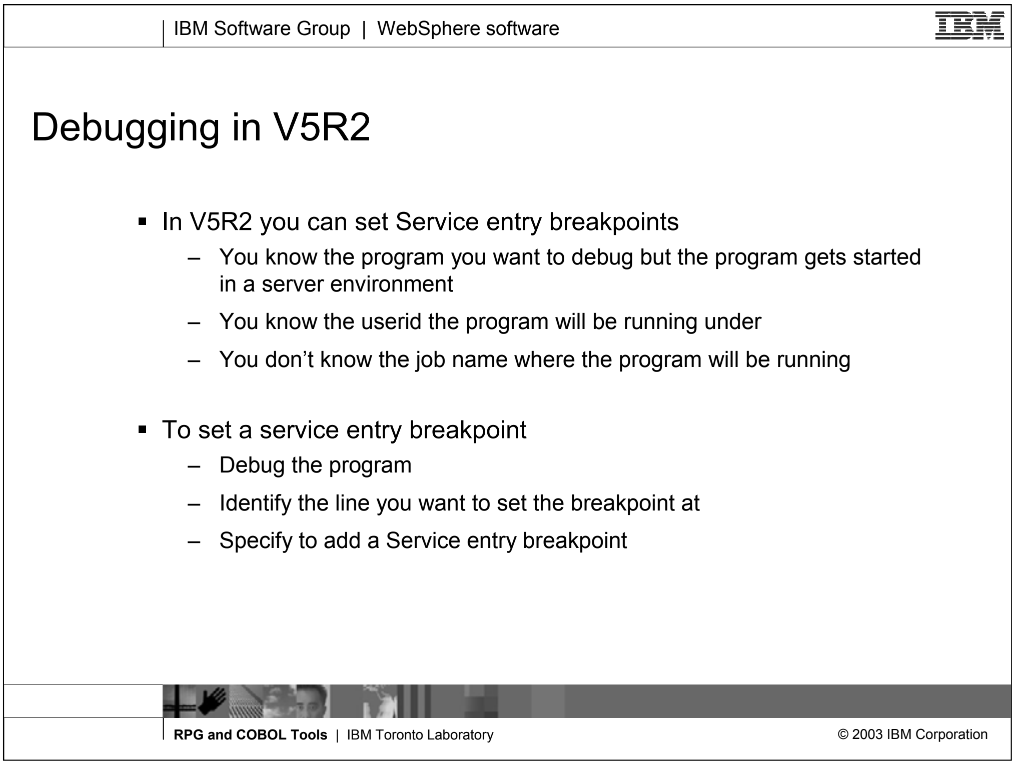
☑ Step into

☑ Terminate debug session on program completion

☑ Update production files

Debug Launch configuration
- •Persist debug session information
- •Specify Service program to be debugged

You can start the Debugger in several ways: direct from the Remote Systems view, or from the Launch Configuration dialog. Starting direct from the preview doesn't allow you to specify parameters to be passed to the program, the Launch Configuration dialog allows you to modify how the program is invoked including to specify parameters.



Debugging in V5R2

- In V5R2 you can set Service entry breakpoints
  - You know the program you want to debug but the program gets started in a server environment
  - You know the userid the program will be running under
  - You don't know the job name where the program will be running
- To set a service entry breakpoint
  - Debug the program
  - Identify the line you want to set the breakpoint at
  - Specify to add a Service entry breakpoint

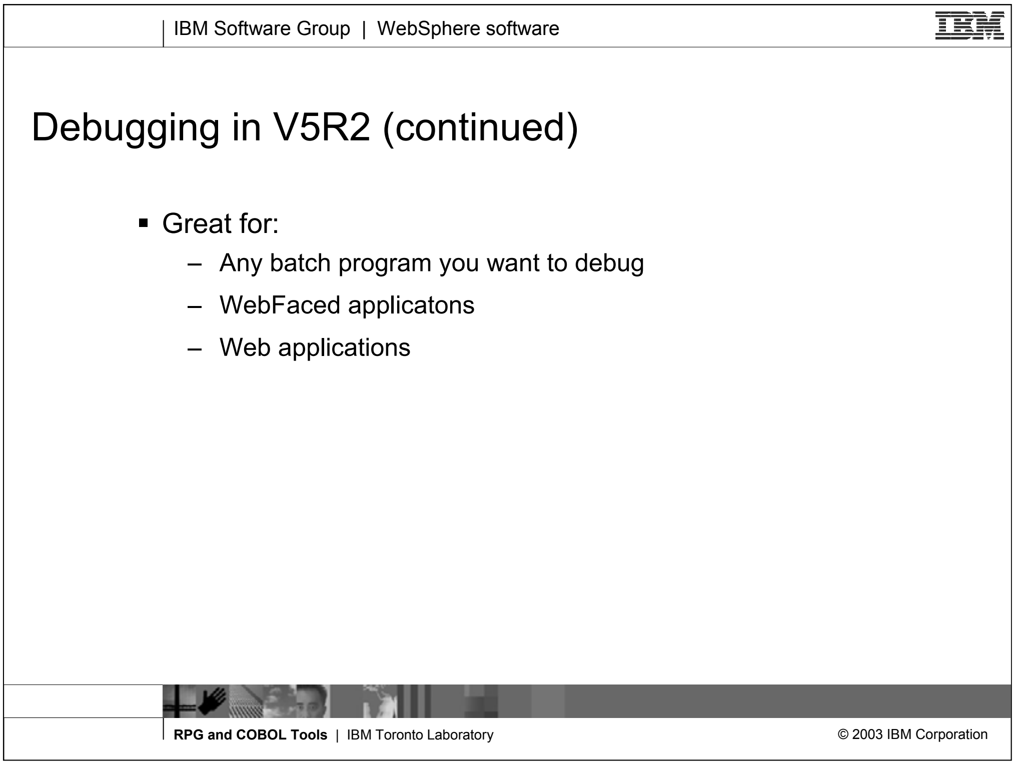
You use service entry points when you wish to debug an application that makes use of the Toolbox or multiple jobs. Examples of cases where you would want to use a service entry point include:

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

•Programs that are spawned by other programs. In this case, you would set a service entry point in the application that will be spawned. When the program is spawned and the line where the service entry point is set is about to execute, operation will be suspended and the debugger will be able to gain control of the program and stop at that line. When a service entry point is set, it is triggered when the application not currently under debug is called.

To set a service entry point, open the source of the application in the Debugger editor. Right-click in the editor or in the editor prefix area directly to the left of the line where you want to add the service entry point and select **Add Service Entry Point** from the pop-up menu. This will invoke the **Add Service Entry Point** dialog box, which displays the program, module, source file, and line number of the service entry point that will be created. In this dialog box, specify the user profile for which the service entry point will be activated. By default, the user profile is set to *CURRENT (the user profile for the current debug session).

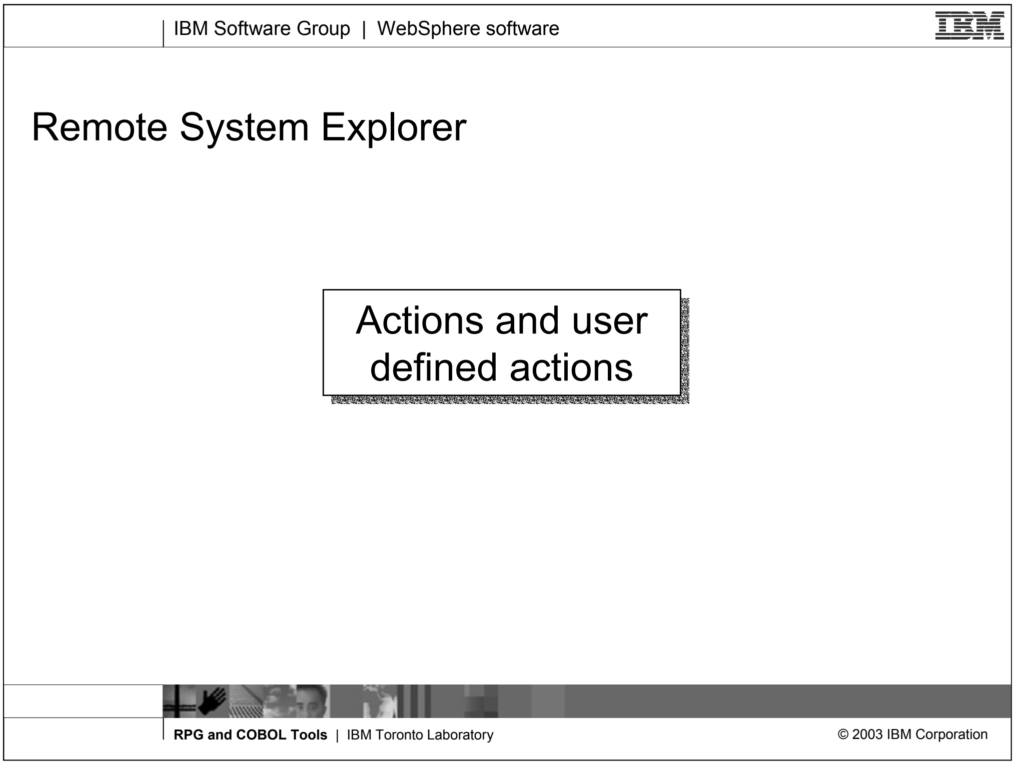
> Note: Whenever the program gets invoked with the specified userid regardless in which job it is running the program will be stopped at the line with the Service Entry breakpoint and the integrated debugger on your workstation will get active.



Debugging in V5R2 (continued)

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

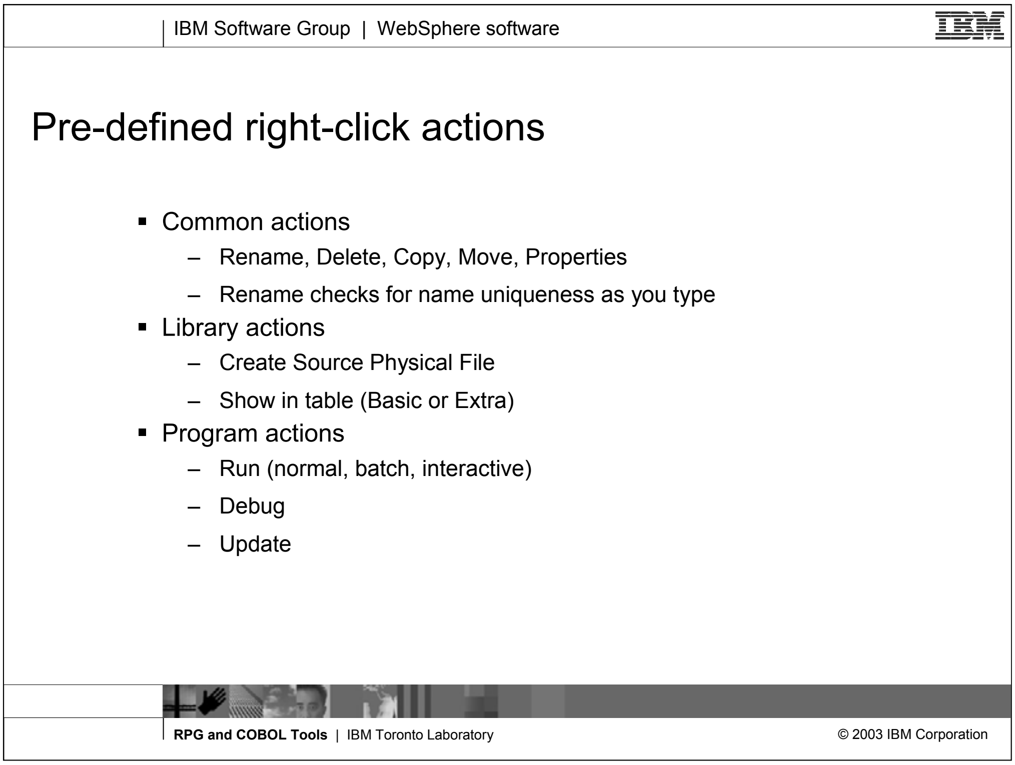
You can debug all sorts of applications such as any batch program, a WebFaced application or a Web application.



Remote System Explorer

Actions and user defined actions

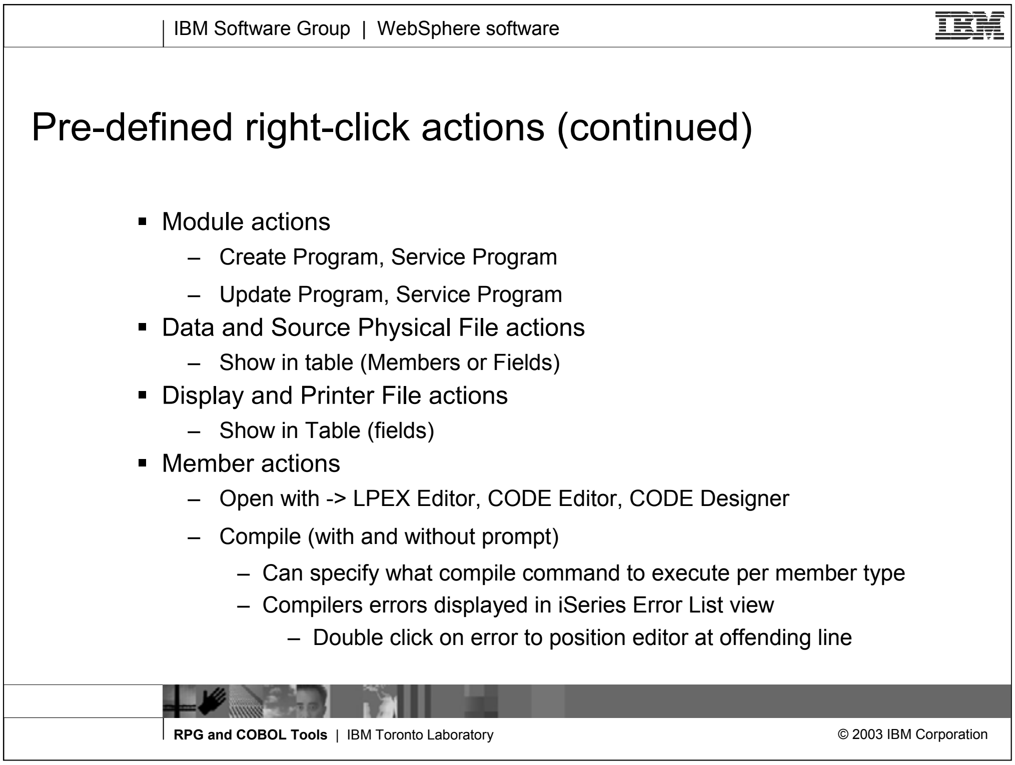
You can run predefined actions or you can create your own actions to run against iSeries libraries, objects, jobs, and members. They can also be defined for folders and files in any remote UNIX, Windows, Linux, Local, or IFS system.



Pre-defined right-click actions

- Common actions
  - Rename, Delete, Copy, Move, Properties
  - Rename checks for name uniqueness as you type
- Library actions
  - Create Source Physical File
  - Show in table (Basic or Extra)
- Program actions
  - Run (normal, batch, interactive)
  - Debug
  - Update

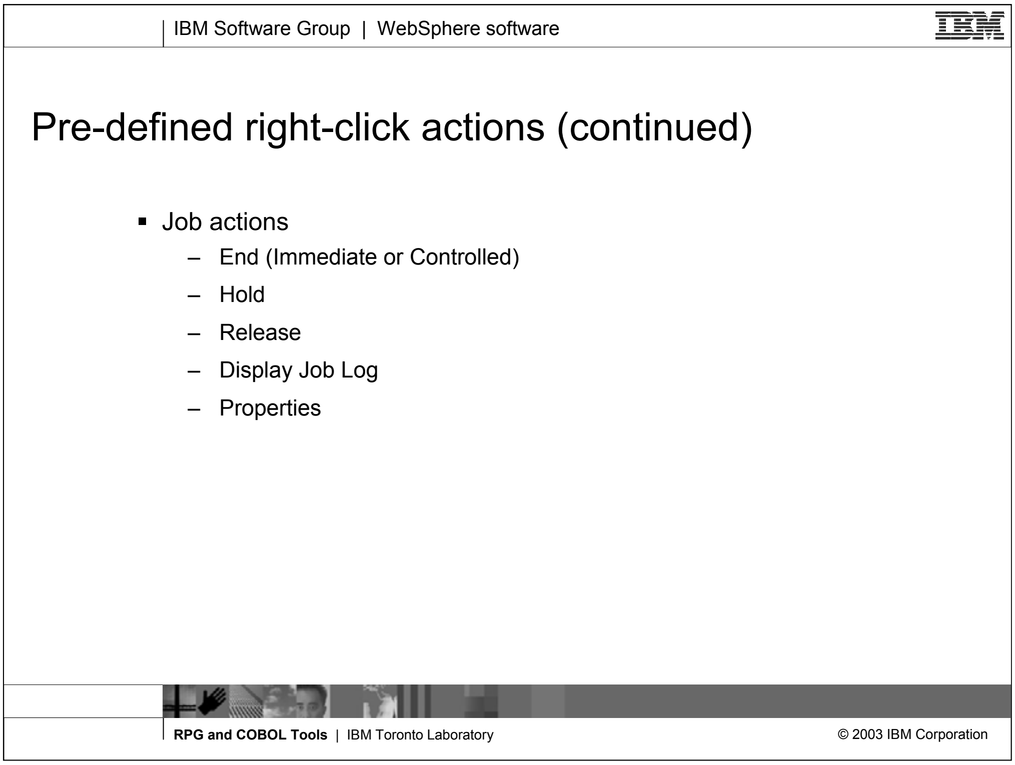
So far you have seen a number of the pop-up menu actions. Here you see a list of many of those pre-defined actions for remote libraries and objects. You simply select an object, right-click and choose from a list of actions that you want to run.



Pre-defined right-click actions (continued)

- Module actions
  - Create Program, Service Program
  - Update Program, Service Program
- Data and Source Physical File actions
  - Show in table (Members or Fields)
- Display and Printer File actions
  - Show in Table (fields)
- Member actions
  - Open with -> LPEX Editor, CODE Editor, CODE Designer
  - Compile (with and without prompt)
    - Can specify what compile command to execute per member type
    - Compilers errors displayed in iSeries Error List view
      - Double click on error to position editor at offending line

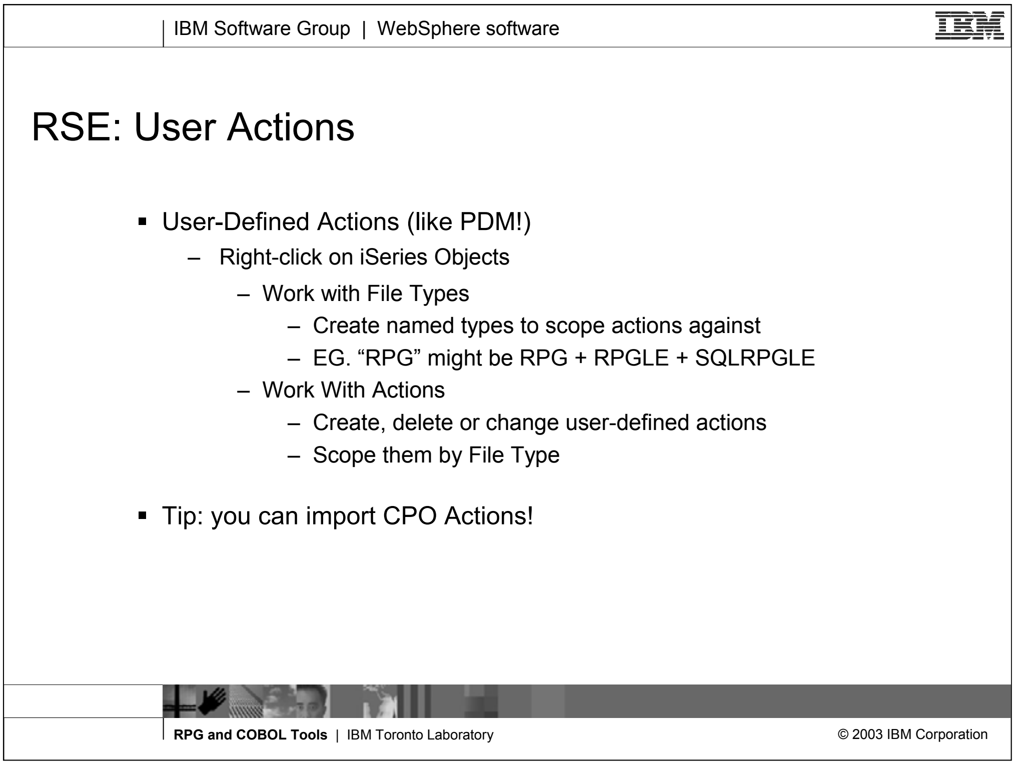
Here you see the pop-up menu actions for remote device files and members.



Pre-defined right-click actions (continued)

- Job actions
  - End (Immediate or Controlled)
  - Hold
  - Release
  - Display Job Log
  - Properties

Here you see the predefined pop-up menu actions forjobs (when using the iSeries Jobs subsystem).

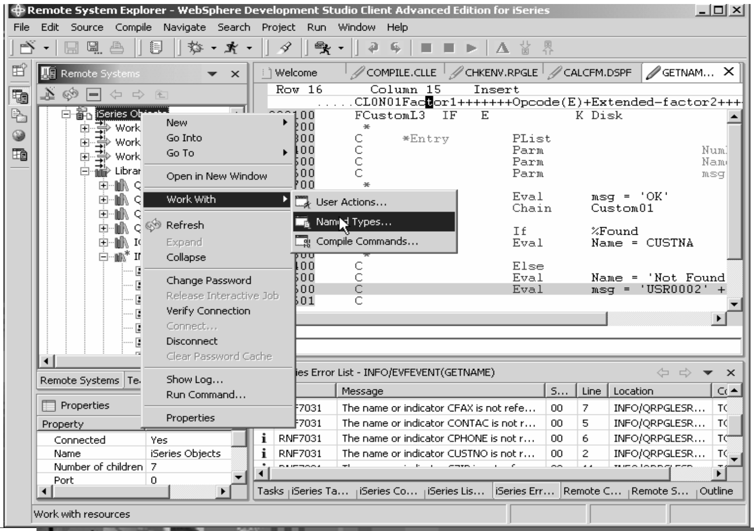


RSE: User Actions

- User-Defined Actions (like PDM!)
  - Right-click on iSeries Objects
    - Work with File Types
      - Create named types to scope actions against
      - EG. “RPG” might be RPG + RPGLE + SQLRPGLE
    - Work With Actions
      - Create, delete or change user-defined actions
      - Scope them by File Type
- Tip: you can import CPO Actions!

While IBM supplies a number of useful actions for remote iSeries objects, it is not possible to supply them all. Like PDM, you can easily define your own actions. To create your own actions, use the Work With actions in the popup menu for iSeries Objects. These user-defined actions will appear in the popup menus for remote resources. To avoid seeing all actions in all popup menus, you scope each action to a one or more object or member types. You first define named collections of object or member types, then you create your actions and scope them to one of these named collections of types. You actions will then only appear for object or members that match one of the types in the collection. If you are a CODE user, you can use File->Import to import existing actions from CODE Project Organizer.

File Types for User Actions



Property	
Connected	Yes
Name	iSeries Objects
Number of children	7
Port	0

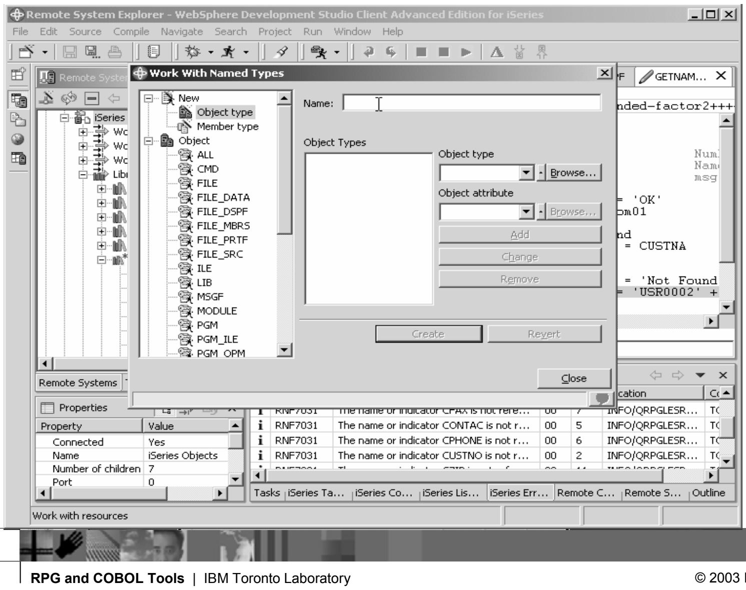
```
 Row 16       Column 15     Insert
      .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2+++
   100      FCustomL3  IF   E          K Disk
   200       *
   300      C      *Entry          PList
   400      C                      Parm                         Num
   500      C                      Parm                         Nam
   600      C                      Parm                         msg
   700       *
                                   Eval      msg = 'OK'
                                   Chain     Custom01

                                   If        %Found
                                   Eval      Name = CUSTNA
                                   
   400      C                      Else
   500      C                      Eval      Name = 'Not Found
   600      C                      Eval      msg = 'USR0002' +
   601      C
```

ies Error List - INFO/EVFEVENT(GETNAME)

		Message	S...	Line	Location	C
	F7031	The name or indicator CFAX is not refe...	00	7	INFO/QRPGLESR...	T(
	F7031	The name or indicator CONTAC is not r...	00	5	INFO/QRPGLESR...	T(
i	RNF7031	The name or indicator CPHONE is not r...	00	6	INFO/QRPGLESR...	T(
i	RNF7031	The name or indicator CUSTNO is not r...	00	2	INFO/QRPGLESR...	T(

To define named collections of types for action scoping, use the Work With -> File Types popup menu action for the iSeries Objects subsystem. Select New... to define a new collection: give the collection a name and type in one or more simple or generic types. There are two types of named collections: one for objects and one for members. Once you have defined a type, it shows up in the dialog, and can be edited by simply selecting it on the left, and changig the information on the right. There are many pre-defined types supplied by IBM

Object / Member Types

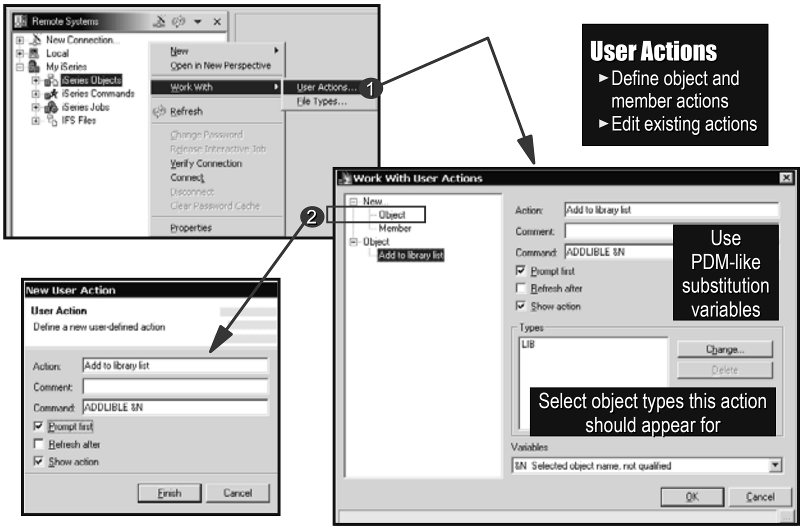


Property	Value
Connected	Yes
Name	iSeries Objects
Number of children	7
Port	0

RNF7031	The name or indicator CONTAC is not r...	00	5	INFO/QRPGLESR...	T(
RNF7031	The name or indicator CPHONE is not r...	00	6	INFO/QRPGLESR...	T(
RNF7031	The name or indicator CUSTNO is not r...	00	2	INFO/QRPGLESR...	T(

You can scope one or more file types to the resource types that apply to the action. For example, a command to start SDA would be scoped to members of type DSPF. The action is only shown for remote objects which is one of the specified types.

Define User Action



New User Action

User Action

Define a new user-defined action

Action: Add to library list

Comment:

Command: ADDLIBLE &N

☑ Prompt first

□ Refresh after

☑ Show action

Work With User Actions

Action: Add to library list

Comment:

Command: ADDLIBLE &N

☑ Prompt first

□ Refresh after

☑ Show action

Types: LIB

Variables: &N Selected object name, not qualified

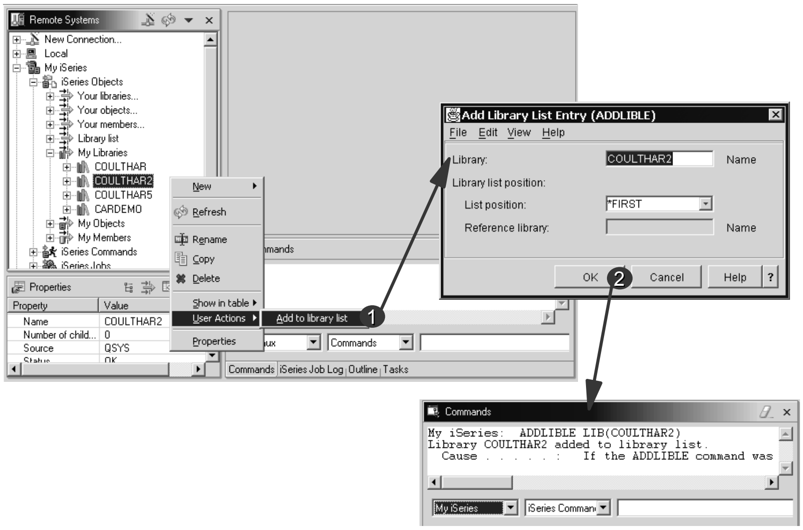
User Actions
- ►Define object and member actions
- ►Edit existing actions

Use PDM-like substitution variables

Select object types this action should appear for

Once you have defined a type to scope your action by, use Work With -> User Actions to define an object or member action. Object actions only appear in the popup menus for libraries or objects. Member actions only appear in the popup menus for members. When you define the action, you initially specify a label to show in the popup menu and a non-interactive iSeries command to run when that action is selected. This command can use substitution variables that are identical to those in PDM. After defining the command, select it in the list and select one or more types to scope it by.

Use your action you just created!



Property	Value
Name	COULTHAR2
Number of child...	0
Source	QSYS
Status	OK

Library: COULTHAR2 Name

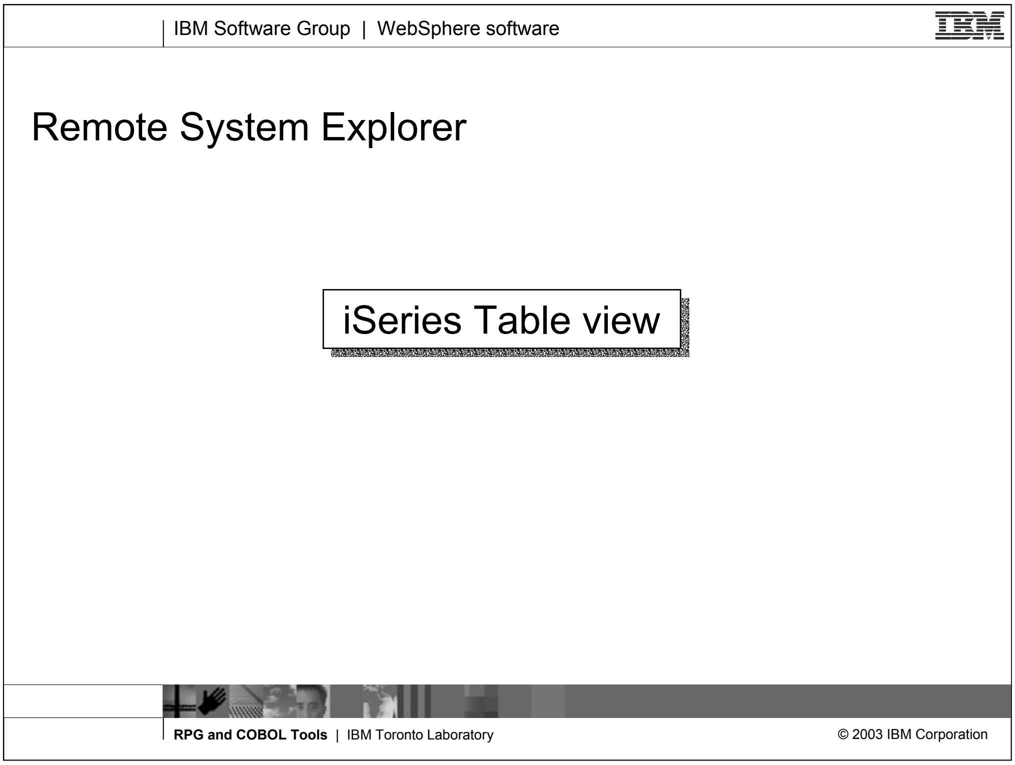
Library list position:

List position: *FIRST

Reference library: Name

```
My iSeries:  ADDLIBLE LIB(COULTHAR2)
Library COULTHAR2 added to library list.
  Cause . . . . . :   If the ADDLIBLE command was
```

Once your action is defined, you can use it. Right click on an object matching one of the types you specified, and expand the User Actions menu in the pop-up. Your action appears in the menu. Select it. If you chose to prompt the command, you will see the GUI prompt for the command. When the command has finished running, it results are logged in the Commands view.

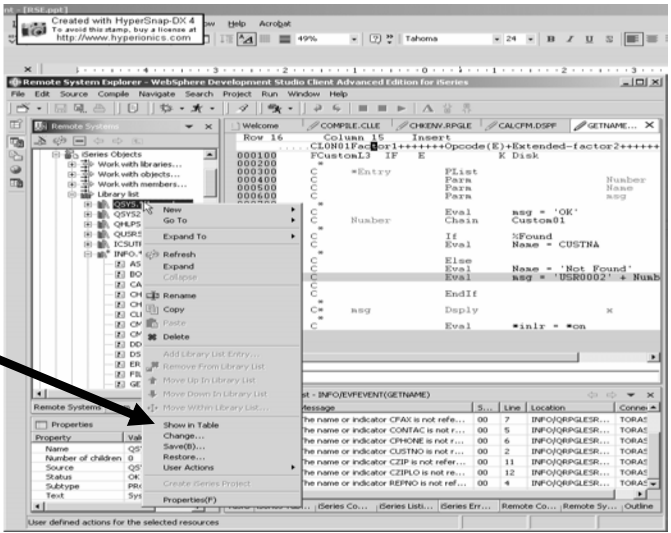


Remote System Explorer

iSeries Table view

The iSeries Table view is more similar to the view you are used to from PDM. You use this view to display a list of items, for example, members or objects, in a table format similar to PDM.

iSeries Table View example



```
Row 16        Column 15      Insert
      .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2++++++
000100     FCustomL3  IF   E             K Disk
000200      *
000300     C     *Entry        PList
000400     C                   Parm                    Number
000500     C                   Parm                    Name
000600     C                   Parm                    msg
            *
           C                   Eval      msg = 'OK'
           C     Number        Chain     Custom01
            *
           C                   If        %Found
           C                   Eval      Name = CUSTNA
            *
           C                   Else
           C                   Eval      Name = 'Not Found'
           C                   Eval      msg = 'USR0002' + Numb
           C
           C                   EndIf
            *
           C*    msg           Dsply                   *
            *
           C                   Eval      *inlr = *on
```

Property	Val
Name	QS
Number of children	0
Source	QS
Status	OK
Subtype	PR
Text	Sys

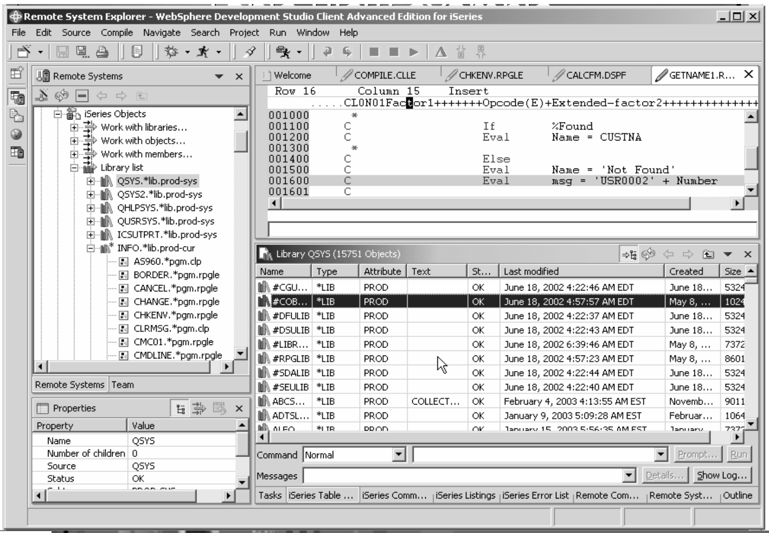
st - INFO/EVFEVENT(GETNAME)

Message	S...	Line	Location	Conne
The name or indicator CFAX is not refe...	00	7	INFO/QRPGLESR...	TORAS
The name or indicator CONTAC is not r...	00	5	INFO/QRPGLESR...	TORAS
The name or indicator CPHONE is not r...	00	6	INFO/QRPGLESR...	TORAS
The name or indicator CUSTNO is not r...	00	2	INFO/QRPGLESR...	TORAS
The name or indicator CZIP is not refer...	00	11	INFO/QRPGLESR...	TORAS
The name or indicator CZIPLO is not re...	00	12	INFO/QRPGLESR...	TORAS
The name or indicator REPNO is not ref...	00	4	INFO/QRPGLESR...	TORAS

User defined actions for the selected resources

For libraries and files we can use a pop-up menu action to open table views to see the contents of the library or file. Here we see what those views look like. In all these tables, the columns are attributes for the object or member in each row. The table can be sorted by an attribute by simply clicking on the column heading. The first two tables are listing library contents, first in Basic mode and then in Extra mode. The third table is listing members in a database file, and the fourth table is listing fields in a device file or record format.

iSeries Table View example (continued)



iSeries Objects
- Work with libraries...
- Work with objects...
- Work with members...
- Library list
  - QSYS.*lib.prod-sys
  - QSYS2.*lib.prod-sys
  - QHLPSYS.*lib.prod-sys
  - QUSRSYS.*lib.prod-sys
  - ICSUTPRT.*lib.prod-sys
  - INFO.*lib.prod-cur
    - AS960.*pgm.clp
    - BORDER.*pgm.rpgle
    - CANCEL.*pgm.rpgle
    - CHANGE.*pgm.rpgle
    - CHKENV.*pgm.rpgle
    - CLRMSG.*pgm.clp
    - CMC01.*pgm.rpgle
    - CMDLINE.*pgm.rpgle

Property	Value
Name	QSYS
Number of children	0
Source	QSYS
Status	OK

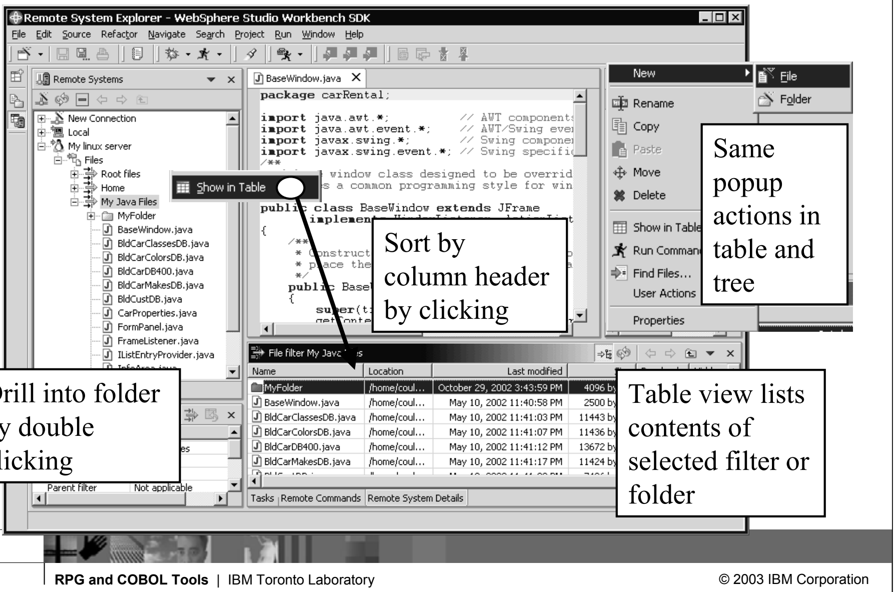
```
 Row 16      Column 15    Insert
      .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2+++++++++++++
001000       *
001100      C                        If        %Found
001200      C                        Eval      Name = CUSTNA
001300       *
001400      C                        Else
001500      C                        Eval      Name = 'Not Found'
001600      C                        Eval      msg = 'USR0002' + Number
001601      C
```

Library QSYS (15751 Objects)

Name	Type	Attribute	Text	St...	Last modified	Created	Size
#CGU...	*LIB	PROD		OK	June 18, 2002 4:22:46 AM EDT	June 18...	5324
#COB...	*LIB	PROD		OK	June 18, 2002 4:57:57 AM EDT	May 8, ...	1024
#DFULIB	*LIB	PROD		OK	June 18, 2002 4:22:37 AM EDT	June 18...	5324
#DSULIB	*LIB	PROD		OK	June 18, 2002 4:22:43 AM EDT	June 18...	5324
#LIBR...	*LIB	PROD		OK	June 18, 2002 6:39:46 AM EDT	May 8, ...	7372
#RPGLIB	*LIB	PROD		OK	June 18, 2002 4:57:23 AM EDT	May 8, ...	8601
#SDALIB	*LIB	PROD		OK	June 18, 2002 4:22:44 AM EDT	June 18...	5324
#SEULIB	*LIB	PROD		OK	June 18, 2002 4:22:40 AM EDT	June 18...	5324
ABCS...	*LIB	PROD	COLLECT...	OK	February 4, 2003 4:13:55 AM EST	Novemb...	9011
ADTSL...	*LIB	PROD		OK	January 9, 2003 5:09:28 AM EST	Februar...	1064
ALFO...	*LIB	PROD		OK	January 15, 2003 5:56:35 AM EST	January...	7372

Here is the result of selecting a library and requesting to show its contents in a table view.

Remote Systems Table View



```
package carRental;

import java.awt.*;          // AWT components
import java.awt.event.*;    // AWT/Swing even
import javax.swing.*;       // Swing componen
import javax.swing.event.*; // Swing specifi
/**
   window class designed to be overrid
   s a common programming style for win

public class BaseWindow extends JFrame
        implements
{
    /**
     * Construct
     * place the
     */
    public Base
    {
        super(ti
```

Name	Location	Last modified	
MyFolder	/home/coul...	October 29, 2002 3:43:59 PM	4096 by
BaseWindow.java	/home/coul...	May 10, 2002 11:40:58 PM	2500 by
BldCarClassesDB.java	/home/coul...	May 10, 2002 11:41:03 PM	11443 by
BldCarColorsDB.java	/home/coul...	May 10, 2002 11:41:07 PM	11436 by
BldCarDB400.java	/home/coul...	May 10, 2002 11:41:12 PM	13672 by
BldCarMakesDB.java	/home/coul...	May 10, 2002 11:41:17 PM	11424 by

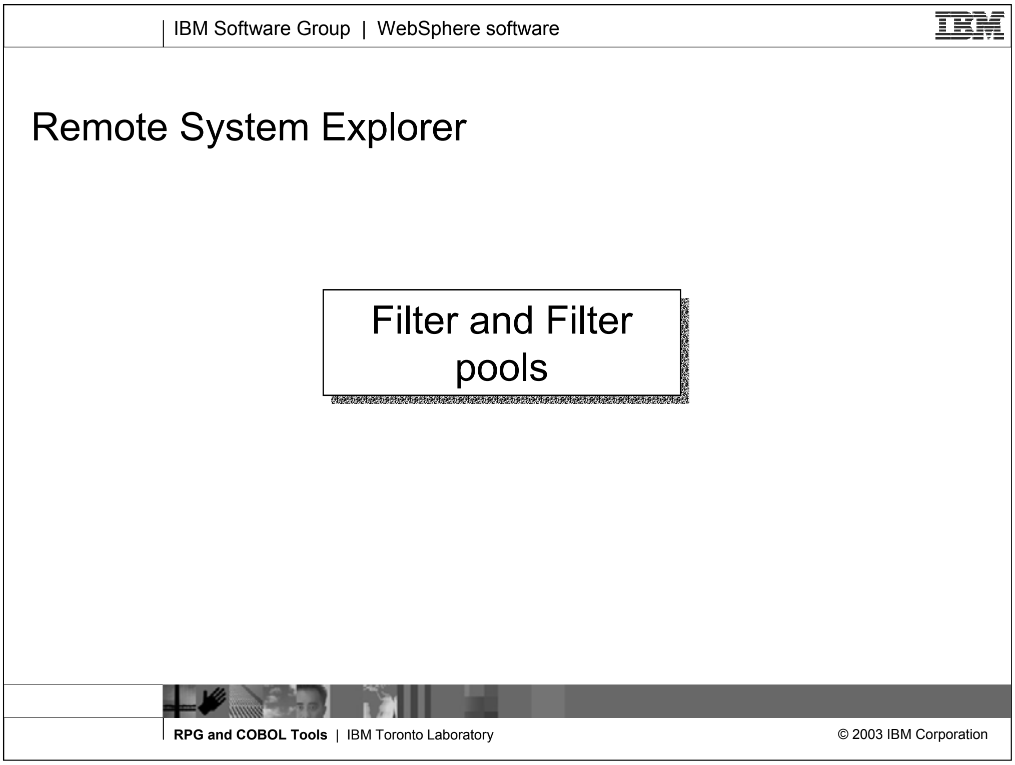
Sort by column header by clicking

Same popup actions in table and tree

Drill into folder by double clicking

Table view lists contents of selected filter or folder

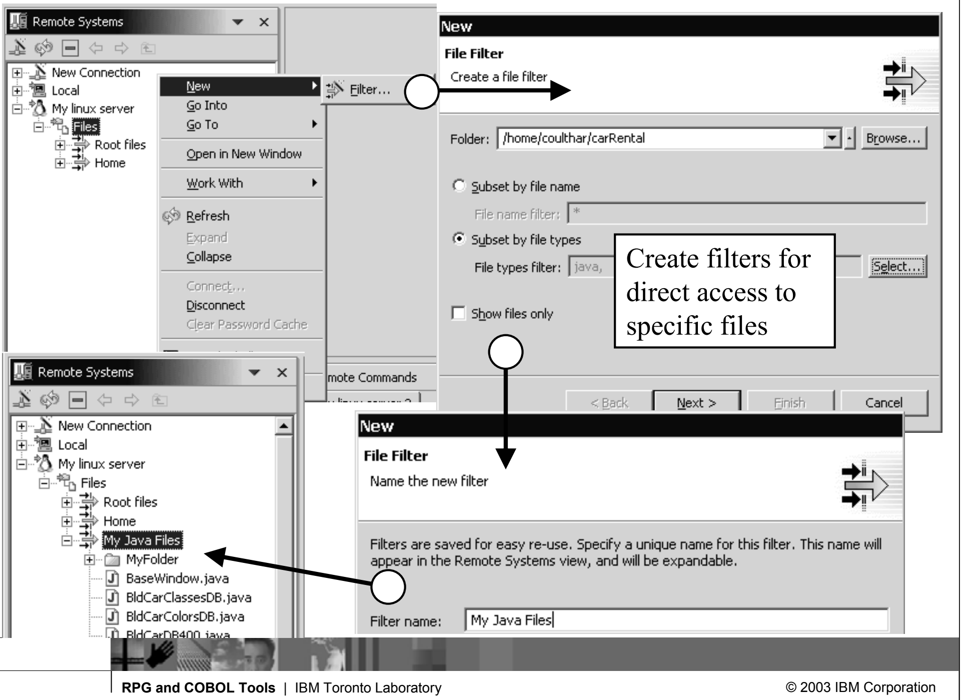
You can perform actions against these items such as editing and compiling. The iSeries Table view takes the selected object in the Remote Systems view as input, and displays the contents in the table. For source physical files, this step displays the members inside, their names, types, attributes, text descriptions, and status. You can click any of the column titles to sort by that attribute, for example, click Size to sort the items by size, which sorts them from smallest to largest. Click the same attribute again to reverse the order that the items are sorted, for example, click Size again to sort from largest to smallest.



Remote System Explorer

Filter and Filter pools

IBM supplies predefined filters, but it won’t be long before you want to create your own filter to subset the list for performance and productivity reasons. When you have many filters you will want to group them into filter pools.



New

File Filter

Create a file filter

Folder: /home/coulthar/carRental

○ Subset by file name

File name filter: *

● Subset by file types

File types filter: java,

□ Show files only

Create filters for direct access to specific files

New

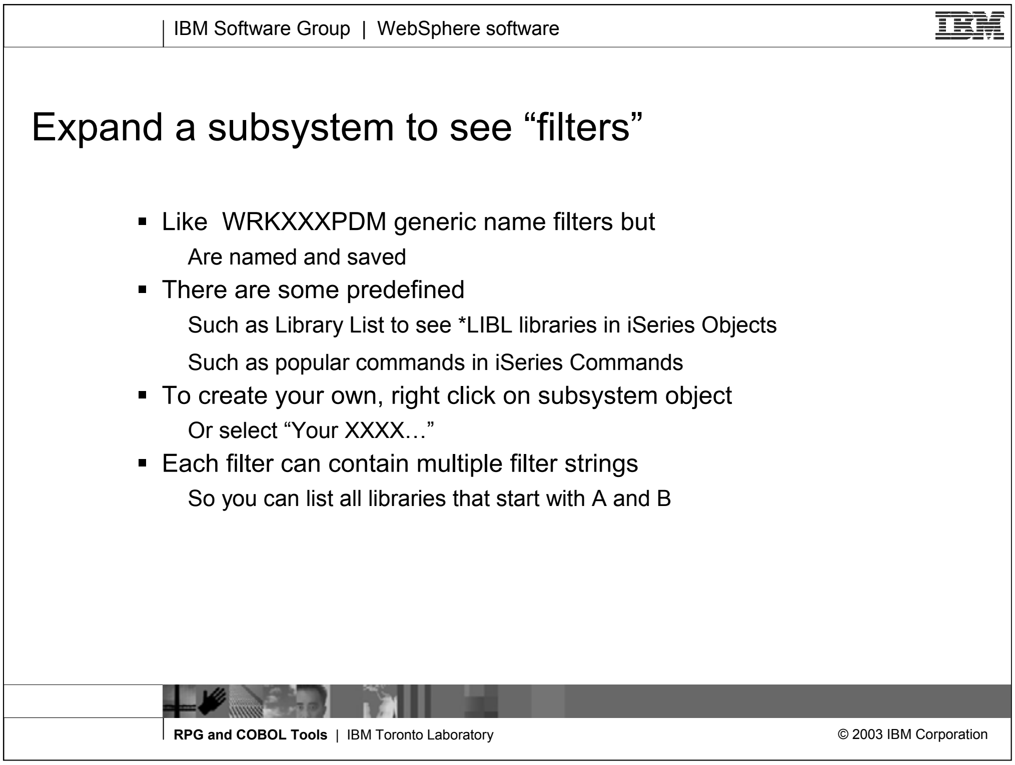
File Filter

Name the new filter

Filters are saved for easy re-use. Specify a unique name for this filter. This name will appear in the Remote Systems view, and will be expandable.

Filter name: My Java Files

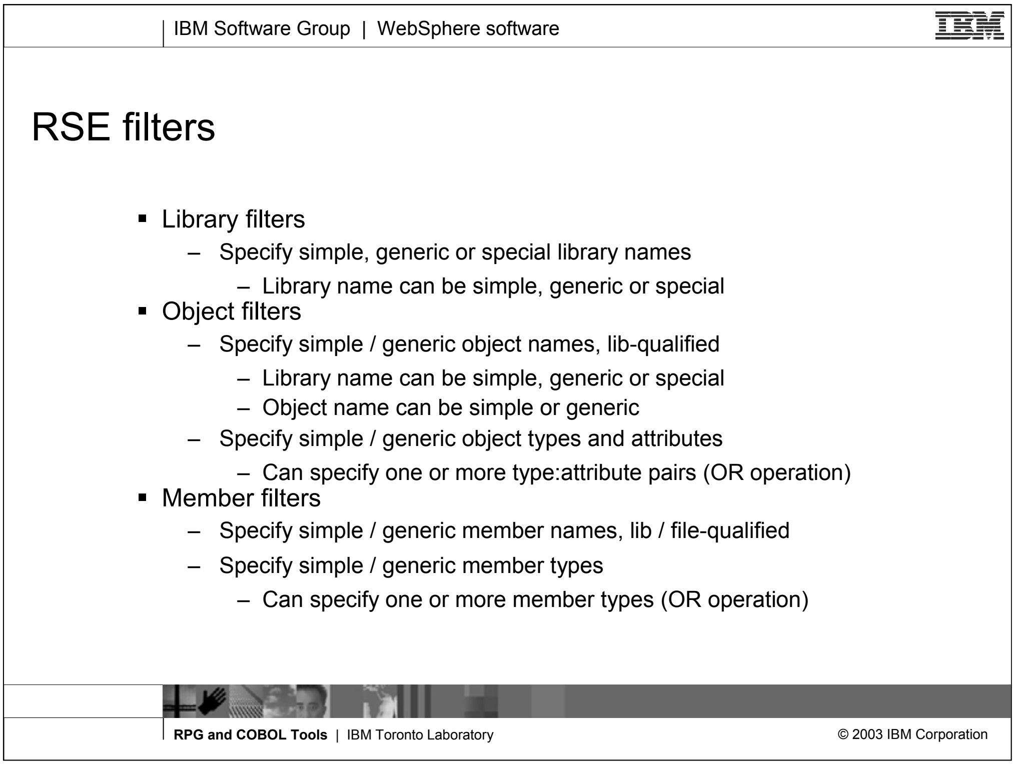
You can easily create your own object filter to subset files for quick access.



Expand a subsystem to see “filters”

- Like WRKXXXPDM generic name filters but
  - Are named and saved
- There are some predefined
  - Such as Library List to see *LIBL libraries in iSeries Objects
  - Such as popular commands in iSeries Commands
- To create your own, right click on subsystem object
  - Or select “Your XXXX…”
- Each filter can contain multiple filter strings
  - So you can list all libraries that start with A and B

The Library List filter IBM supplies is useful, but it won't be long before you want to create your own filter to subset the list for performance and productivity reasons. Filters in RSE are like WRKXXXPDM commands in that you specify simple or generic names, and object or member type criteria, to generate a subset list. Unlike PDM however, the filters in RSE are saved between sessions, so you need only specify the filtering criteria once. All filters in RSE are named collections of filter strings. It is the filter strings themselves that hold the filtering criteria. By allowing multiple filter strings for each filter, you can complete flexibility in what objects are shown when that filter is expanded. For example, sometimes you cannot use a single generic name to capture all the objects or members for a particular task.



RSE filters

- Library filters
  - Specify simple, generic or special library names
    - Library name can be simple, generic or special
- Object filters
  - Specify simple / generic object names, lib-qualified
    - Library name can be simple, generic or special
    - Object name can be simple or generic
  - Specify simple / generic object types and attributes
    - Can specify one or more type:attribute pairs (OR operation)
- Member filters
  - Specify simple / generic member names, lib / file-qualified
  - Specify simple / generic member types
    - Can specify one or more member types (OR operation)

There are three types of filters you can create in the iSeries Objects subsystem.

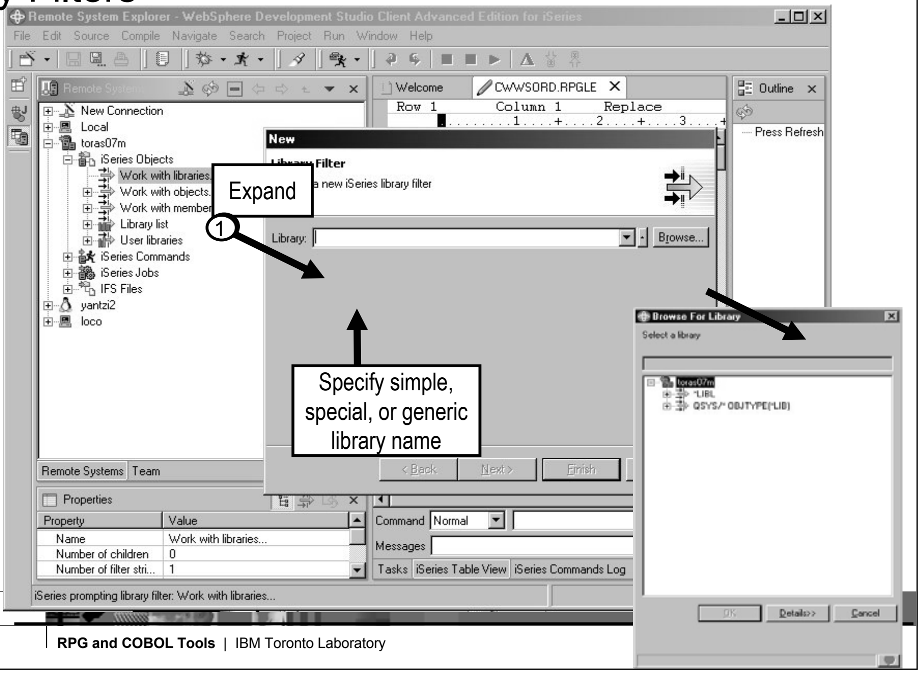
1. Library filters. These list libraries when expanded.

2. Object filters. These list objects when expanded.

3. Member filters. These list members when expanded.

We will see an example of creating each of these. If you are an existing CODE user, you can use File->Import to import filters from CODE Project Organizer.

Library Filters



Row 1 Column 1 Replace

Library:

Command: Normal

Property	Value
Name	Work with libraries...
Number of children	0
Number of filter stri...	1

iSeries prompting library filter: Work with libraries...

Expand

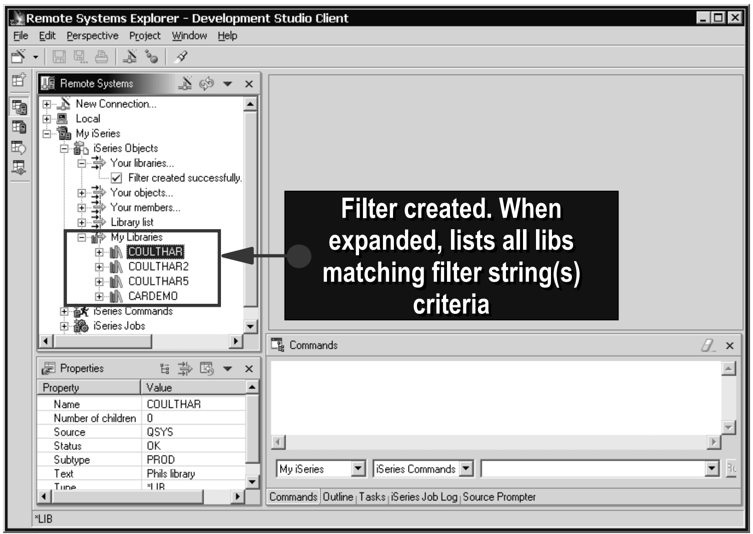
1

Specify simple, special, or generic library name

Here we use the **Your libraries...** prompt to create a library filter. This prompt creates a filter, and then immediately expands it for you. Alternatively, you can also right click on the iSeries Objects subsystem and use a popup menu action to create a new library filter. To create a library filter, type in a name to give the filter. This is what will appear in the tree under iSeries Objects.

Press **Add...** once or more to add library filter strings. Each is a simple or generic library name.

Expanding Library Filters

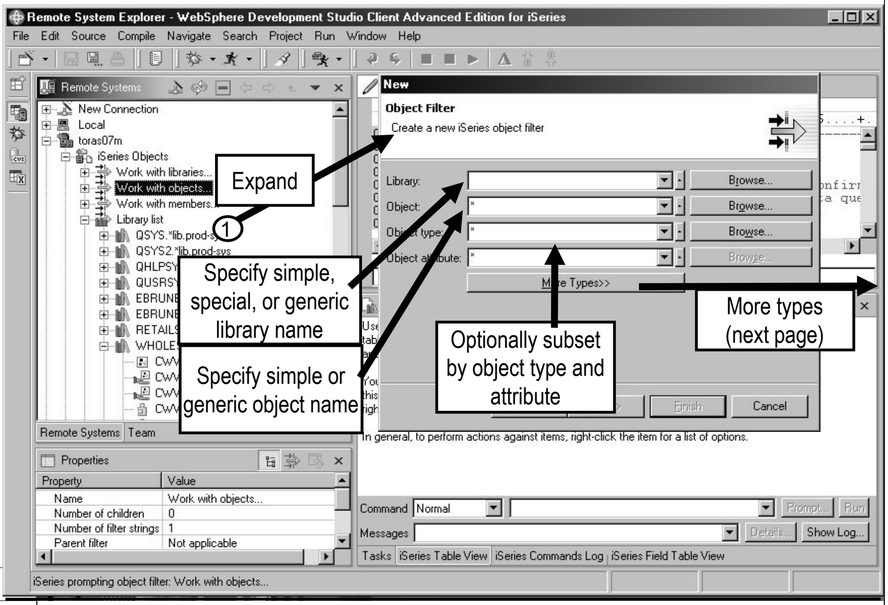


Property	Value
Name	COULTHAR
Number of children	0
Source	QSYS
Status	OK
Subtype	PROD
Text	Phils library
Type	*LIB

Filter created. When expanded, lists all libs matching filter string(s) criteria

When a library filter is created, it shows up in the list of existing filters for iSeries Objects. You can edit the filter by right clicking on it and selecting the Change... action from its popup menu. When a library filter is expanded, all libraries matching one or more of the filter's filter strings are listed underneath the filter. Note that these libraries can subsequently be expanded, just as they can for the Library List filter.

Object Filters



Object Filter

Create a new iSeries object filter

Library:

Object: *

Object type: *

Object attribute: *

Property	Value
Name	Work with objects...
Number of children	0
Number of filter strings	1
Parent filter	Not applicable

In general, to perform actions against items, right-click the item for a list of options.

iSeries prompting object filter: Work with objects...

Expand

1

Specify simple, special, or generic library name

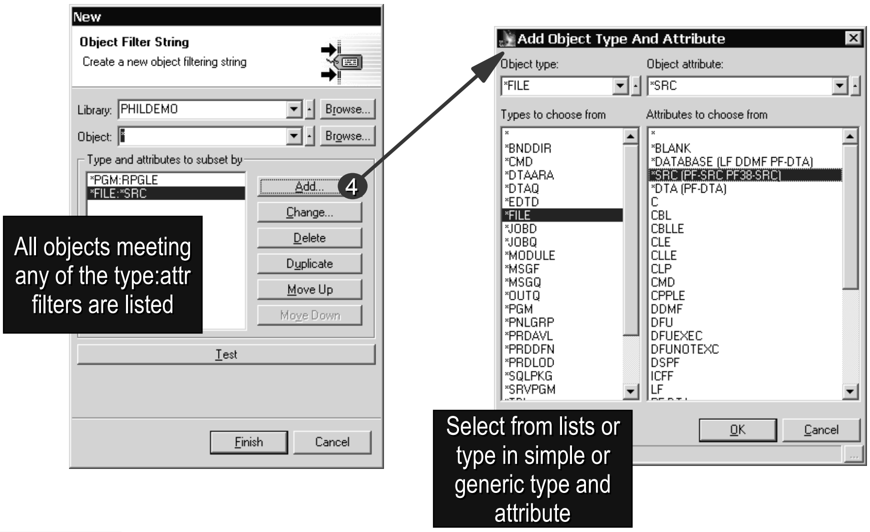
Specify simple or generic object name

Optionally subset by object type and attribute

More types (next page)

Here you use the Your Objects... prompt to create a library filter. This prompt creates a filter, and then immediately expands it for you. Alternatively, you can also right click on the iSeries Objects subsystem and use a popup menu action to create a new object filter. To create an object filter, type in a name to give the filter. This is what will appear in the tree under iSeries Objects. Press Add... once or more to add object filter strings. Each object filter string identifies objects to list. The library and object name can be simple or generic. To further subset the list to only objects of particular types or attributes, press Add... from the filter string wizard...

Filtering objects by type + attribute



New

Object Filter String

Create a new object filtering string

Library: PHILDEMO

Object: *

Type and attributes to subset by

*PGM:RPGLE
*FILE:*SRC

All objects meeting any of the type:attr filters are listed

Add Object Type And Attribute

Object type: *FILE

Object attribute: *SRC

Types to choose from

*
*BNDDIR
*CMD
*DTAARA
*DTAQ
*EDTD
*FILE
*JOBD
*JOBQ
*MODULE
*MSGF
*MSGQ
*OUTQ
*PGM
*PNLGRP
*PRDAVL
*PRDDFN
*PRDLOD
*SQLPKG
*SRVPGM

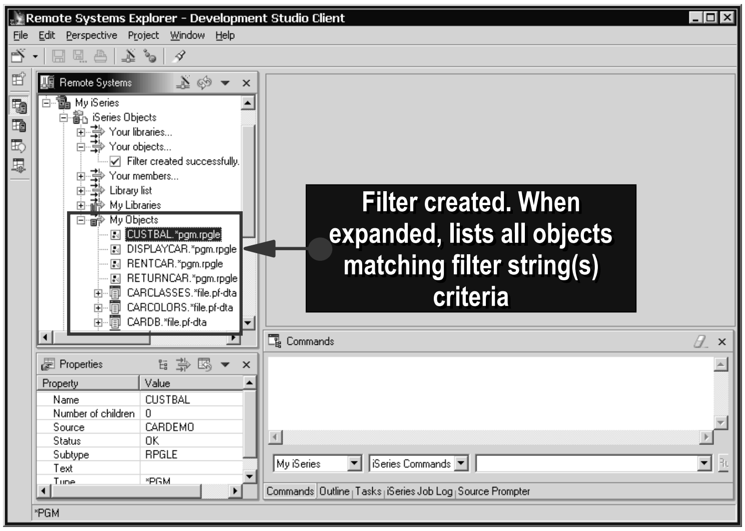
Attributes to choose from

*
*BLANK
*DATABASE (LF DDMF PF-DTA)
*SRC (PF-SRC PF38-SRC)
*DTA (PF-DTA)
C
CBL
CBLLE
CLE
CLLE
CLP
CMD
CPPLE
DDMF
DFU
DFUEXEC
DFUNOTEXC
DSPF
ICFF
LF

Select from lists or type in simple or generic type and attribute

The Add button in the Object Filter String wizard allows you to specify one or more object type and object attribute pairs. The object type can be any valid iSeries object type. The object attribute can be * to match on all objects of that type, or a simple or generic attribute to restrict to only objects of that type that match the attribute. You can add as many object type and attribute pairs as you need to refine your list. All objects that match on any of these type and attribute pairs will be listed, when the filter is expanded.

Expanding Object Filters



Remote Systems Explorer - Development Studio Client
File Edit Perspective Project Window Help
Remote Systems
Properties
Commands
My iSeries
iSeries Commands
Commands Outline Tasks iSeries Job Log Source Prompter

My iSeries
iSeries Objects
Your libraries...
Your objects...
☑ Filter created successfully.
Your members...
Library list
My Libraries
My Objects
CUSTBAL.*pgm.rpgle
DISPLAYCAR.*pgm.rpgle
RENTCAR.*pgm.rpgle
RETURNCAR.*pgm.rpgle
CARCLASSES.*file.pf-dta
CARCOLORS.*file.pf-dta
CARDB.*file.pf-dta

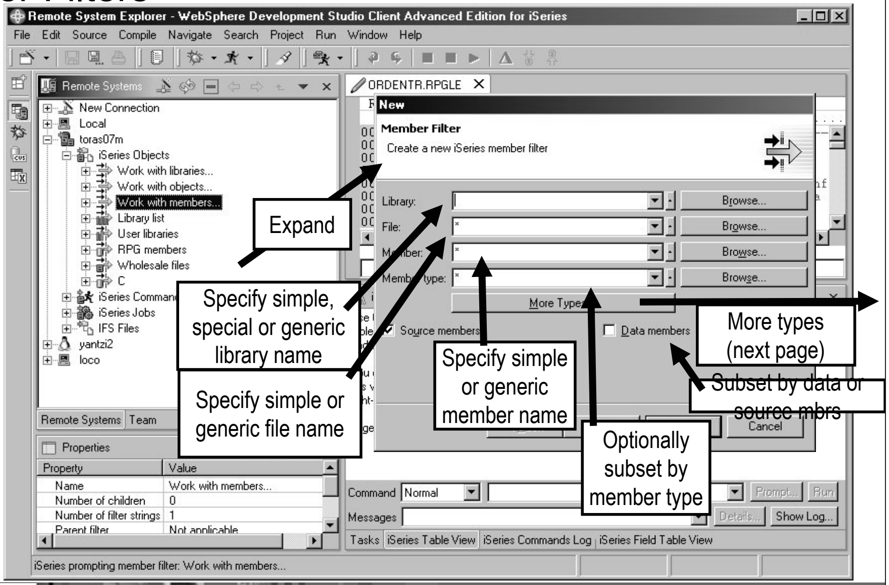
Property	Value
Name	CUSTBAL
Number of children	0
Source	CARDEMO
Status	OK
Subtype	RPGLE
Text	
Type	*PGM

*PGM

Filter created. When expanded, lists all objects matching filter string(s) criteria

When an object filter is created, it shows up in the list of existing filters for iSeries Objects. You can edit the filter by right clicking on it and selecting the Change... action from its popup menu. When an object filter is expanded, all objects matching one or more of the filter's filter strings are listed underneath the filter. Note that these objects can subsequently be expanded, if they are files, just as they can from the Library List filter.

Member Filters



Remote System Explorer - WebSphere Development Studio Client Advanced Edition for iSeries

New

Member Filter

Create a new iSeries member filter

Library:

File: *

Member: *

Member type: *

☑ Source members ☐ Data members

Property	Value
Name	Work with members...
Number of children	0
Number of filter strings	1
Parent filter	Not applicable

Command: Normal

iSeries prompting member filter: Work with members...

Expand

Specify simple, special or generic library name

Specify simple or generic file name

Specify simple or generic member name

Optionally subset by member type

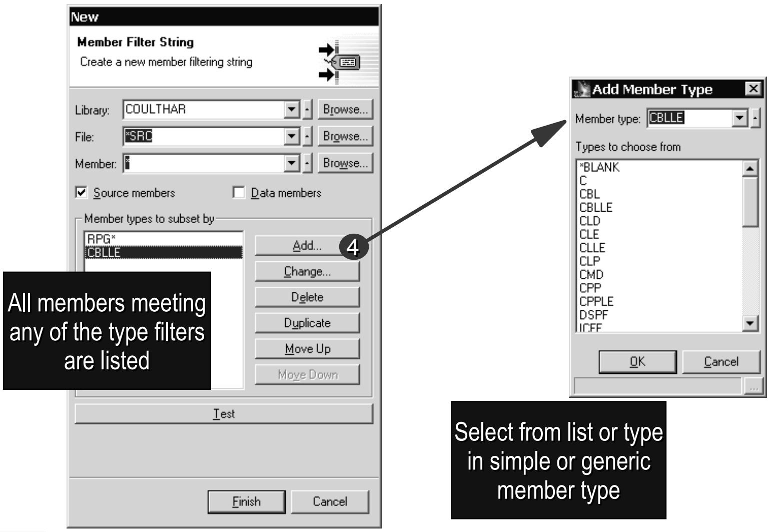
More types (next page)

Subset by data or source mbrs

Here you use the Your Members... prompt to create a member filter. This prompt creates a filter, and then immediately expands it for you. Alternatively, you can also right click on the iSeries Objects subsystem and use a popup menu action to create a new member filter.

To create a member filter, type in a name to give the filter. This is what will appear in the tree under iSeries Objects. Press Add... once or more to add member filter strings. Each member filter string identifies members to list. The library, file and member name can be simple or generic. To further subset the list to only members of particular types press Add... from the filter string wizard...

Filtering by member type



New

Member Filter String

Create a new member filtering string

Library: COULTHAR

File: *SRC

Member: *

☑ Source members

□ Data members

Member types to subset by

RPG*
CBLLE

All members meeting any of the type filters are listed

Add Member Type

Member type: CBLLE

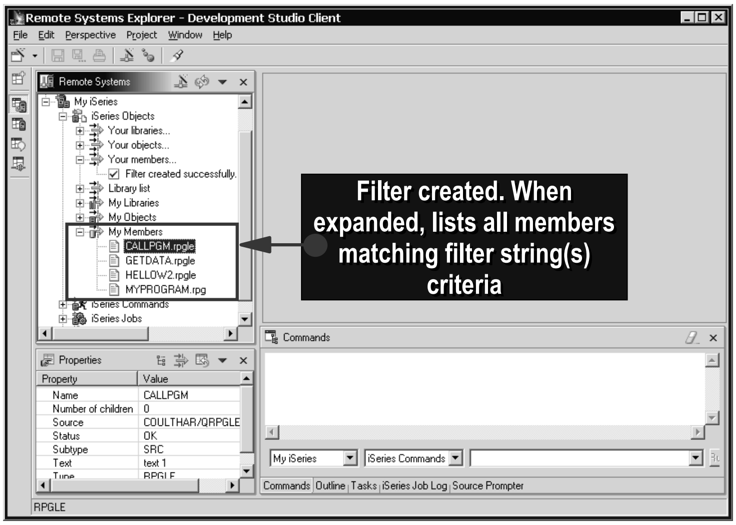
Types to choose from

*BLANK
C
CBL
CBLLE
CLD
CLE
CLLE
CLP
CMD
CPP
CPPLE
DSPF
ICFF

Select from list or type in simple or generic member type

The Add button in the Member Filter String wizard allows you to specify one or more member types. Each member type can be a simple member type, or a generic member type as in RPG*. You can add as many member types as you need to refine your list. All members that match on any of these type will be listed, when the filter is expanded.

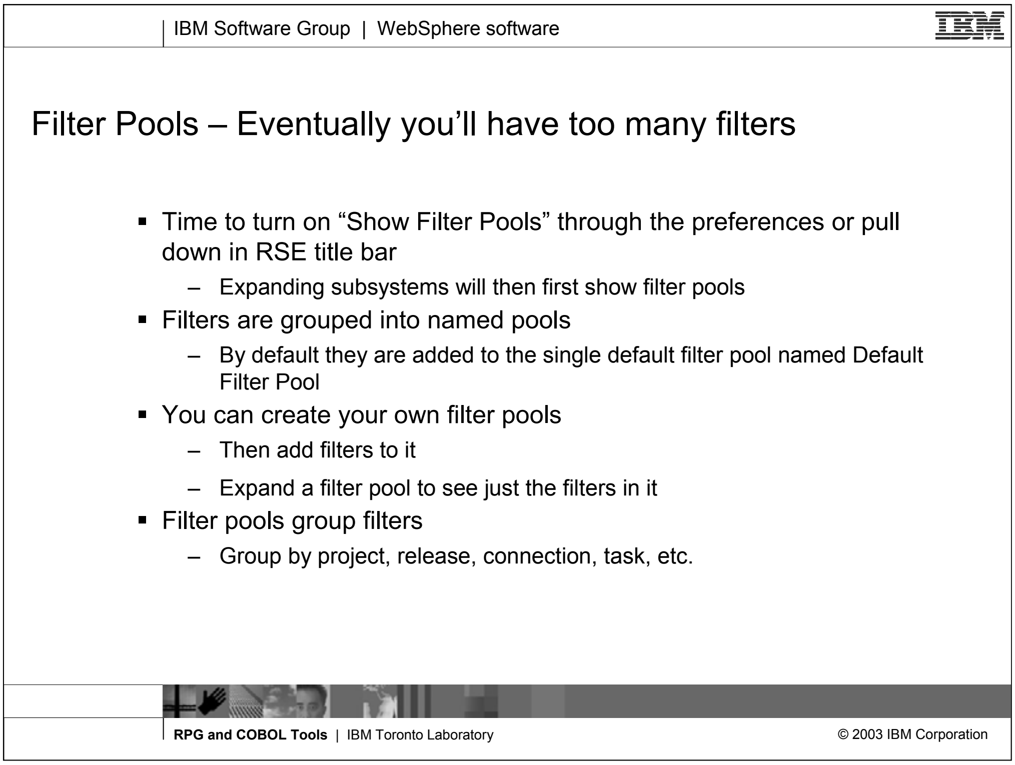
Expanding Member Filters



Property	Value
Name	CALLPGM
Number of children	0
Source	COULTHAR/QRPGLE
Status	OK
Subtype	SRC
Text	text 1
Type	RPGLE

Filter created. When expanded, lists all members matching filter string(s) criteria

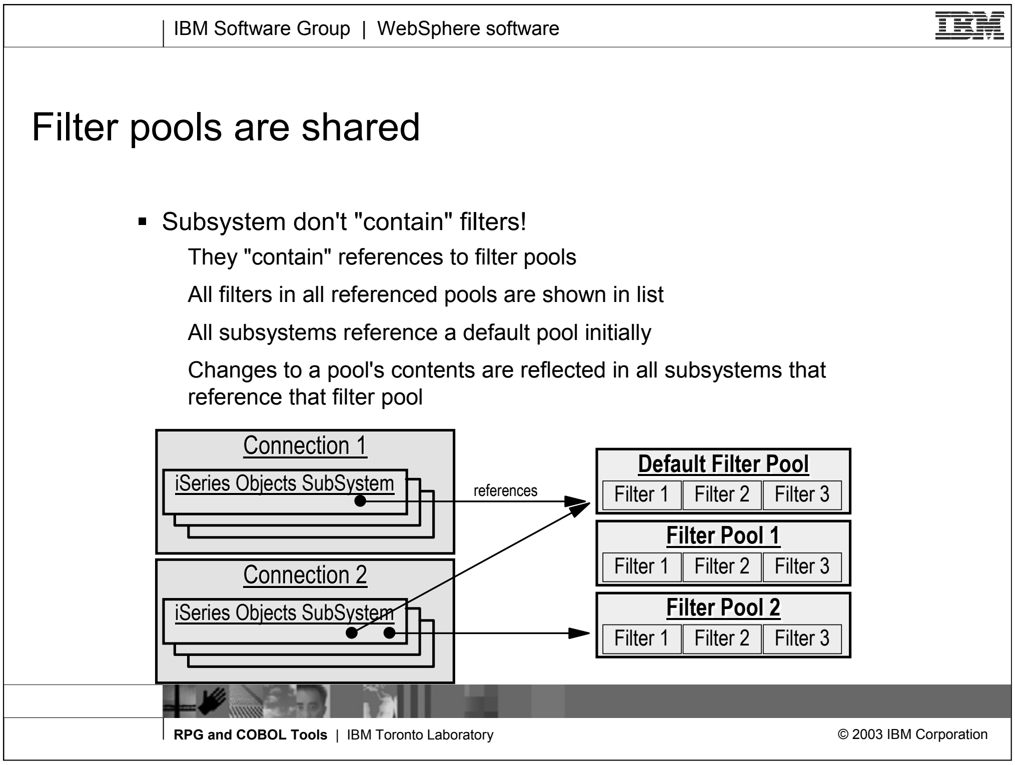
When a member filter is created, it shows up in the list of existing filters for iSeries Objects. You can edit the filter by right clicking on it and selecting the Change... action from its popup menu. When a member filter is expanded, all members matching one or more of the filter's filter strings are listed underneath the filter.



Filter Pools – Eventually you'll have too many filters

- Time to turn on "Show Filter Pools" through the preferences or pull down in RSE title bar
  - Expanding subsystems will then first show filter pools
- Filters are grouped into named pools
  - By default they are added to the single default filter pool named Default Filter Pool
- You can create your own filter pools
  - Then add filters to it
  - Expand a filter pool to see just the filters in it
- Filter pools group filters
  - Group by project, release, connection, task, etc.

There is an advanced feature in the RSE for partitioning filters into named collections, called filter pools. To enable this, you must select the "Show Filter Pools" preference. Actually, all filters are contained in a filter pool. By default, they all go into one IBM supplied filter pool, named Default Filter Pool. By turning on the Show Filter Pools preference, you will see these filter pools when you expand a subsystem. When you expand a filter pool, you will then see the filters. Filter pools can be used to effectively group filters by task, or project, or release, or developer, or whatever you like.



Filter pools are shared

- Subsystem don't "contain" filters!

  They "contain" references to filter pools

  All filters in all referenced pools are shown in list

  All subsystems reference a default pool initially

  Changes to a pool's contents are reflected in all subsystems that reference that filter pool

Connection 1

iSeries Objects SubSystem

references

Connection 2

iSeries Objects SubSystem

Default Filter Pool

Filter 1 | Filter 2 | Filter 3

Filter Pool 1

Filter 1 | Filter 2 | Filter 3

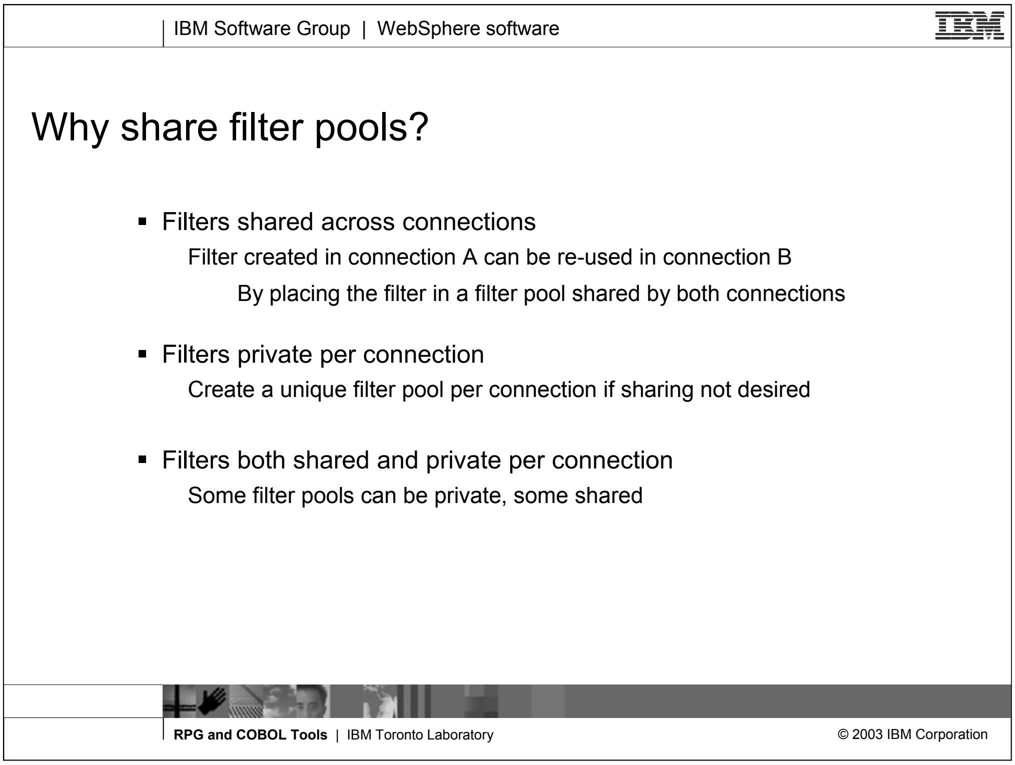
Filter Pool 2

Filter 1 | Filter 2 | Filter 3

Subsystems within each connection do not actually contain filter pools per se. They contain references to filter pools. Every connection you create is supplied with a single reference to the default filter pool. You can easily add references to other filter pools for any connection.

When filters are added, changed or deleted within a filter pool, these changes are reflected in all connections that reference the filter pool. Or more accurately, these changes are reflected in the subsystem that references the filter pool within all connections. It is important to know that each connection has a unique subsystem instance.

The diagram shows two connections and three filter pools. The first connection contains only the IBM supplied reference to the default filter pool. The second connection also contains a reference to another filter pool. Filters in Filter Pool 2 only appear for Connection 2 in this example.



Why share filter pools?

- Filters shared across connections

  Filter created in connection A can be re-used in connection B

  By placing the filter in a filter pool shared by both connections

- Filters private per connection

  Create a unique filter pool per connection if sharing not desired

- Filters both shared and private per connection

  Some filter pools can be private, some shared

There are good reasons to store filter pool references, versus actual filter pools, within each connection. Consider creating a connection, then adding a number of filters to it:

If you delete the connection, would you expect the filters to be deleted too?

If you create a second connection, would you expect to be able to see the filters from the first connection?

If you do see the filters in two connections, if you change a filter in one connection, do you expect it to be changed in the other connection?

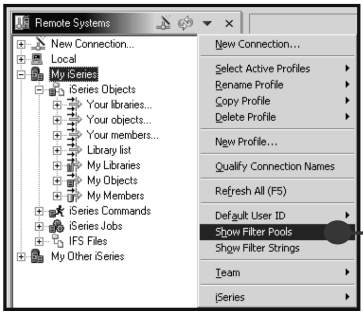
By using filter pools and filter pool references, we get the most flexibility:

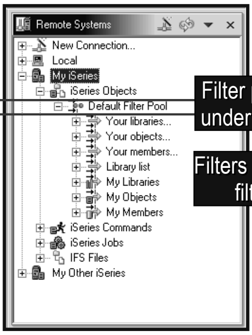
1. You can share filters across connections by referencing the same pool in multiple connections.
2. You can have filters unique to a connection by having a unique filter pool only referenced by that connection.
3. You can choose to reference multiple filter pools in one connection, so some filters can be shared between connections while others are unique to a connection. This is most powerful when used with team support that we describe later.

In all cases, there is only a single copy of each filter, and when changed all connections referencing its parent filter pool see that change.

Filters are never deleted implicitly, for example when a connection is deleted. You must explicitly delete the filter, or its filter pool.

Show Filter Pools





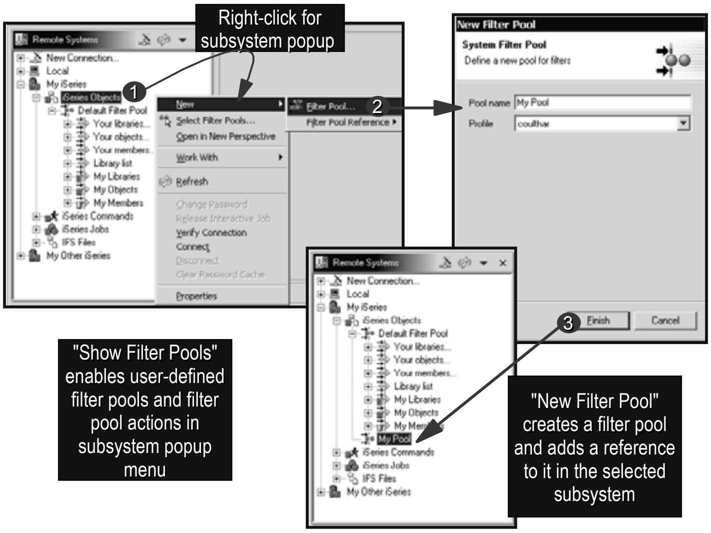
Filter pools shown under subsystems

Filters shown under filter pools

"Show Filter Pools" changes RSE tree-view to display filter pools underneath subsystems

Here we see how to turn on the Show Filter Pools preference. Once enabled, when a subsystem is expanded you see not the filters, but first the filter pools referenced by this subsystem in this connection. When a filter pool is expanded, then you see the filters within that filter pool.

Create a Filter Pool



Pool name: My Pool

Profile: coulthar

Right-click for subsystem popup

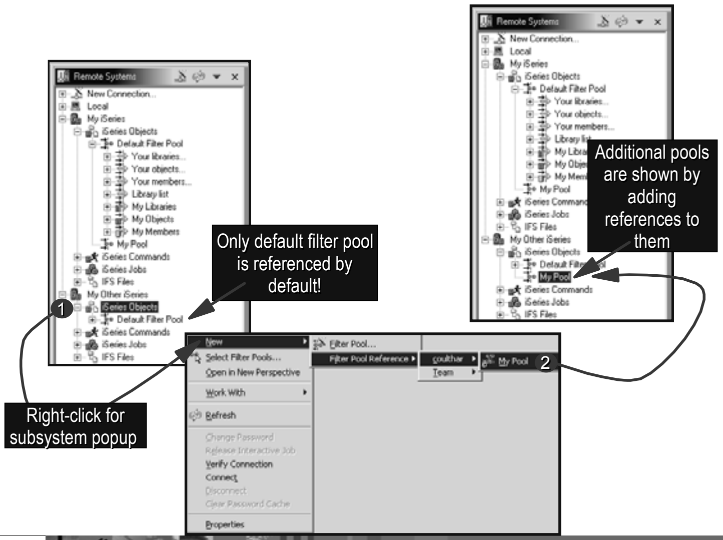
"Show Filter Pools" enables user-defined filter pools and filter pool actions in subsystem popup menu

"New Filter Pool" creates a filter pool and adds a reference to it in the selected subsystem

To create a new filter pool, right click on the subsystem, then select New -> Filter Pool...

All you need to specify for your new filter pool is a unique name. This is what you see when the subsystem is expanded. Never mind the Profile prompt for now, we will explain that later. It is defaulted so you need not think about it. When Finish is pressed, the filter pool is created, AND a reference to it is added to this subsystem in this connection. No other connection has a reference to this new filter pool yet!

Filter Pool References



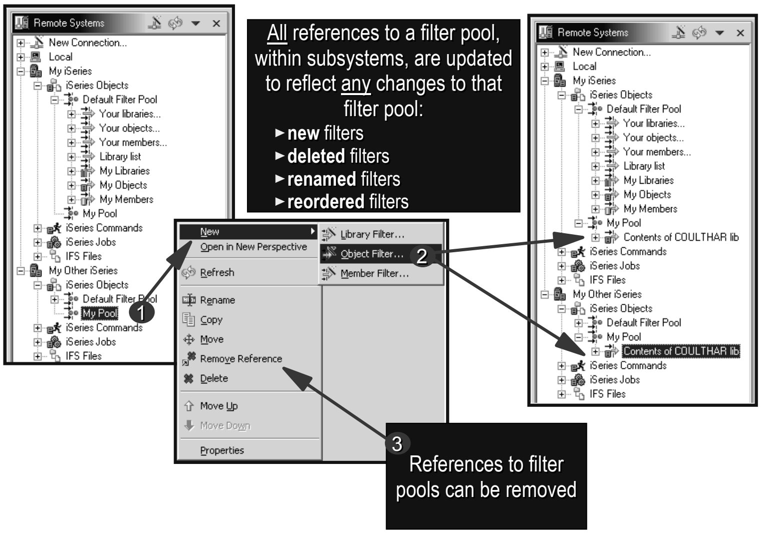
Only default filter pool is referenced by default!

Right-click for subsystem popup

Additional pools are shown by adding references to them

To enable another connection to see the filters we will add to this new filter pool, you must explicitly add a reference to this filter pool in that other connection. Right click on the iSeries Objects subsystem for the other connection, and select the New -> Filter Pool Reference menu. Select the filter pool, within the profile it was created in (by default it will be the profile with your workstation's hostname). Once a reference is added, this pool will now show up under the subsystem for this other connection too. Now let's add some filters to this new filter pool...

Filter Pool References



All references to a filter pool, within subsystems, are updated to reflect any changes to that filter pool:

- **new** filters
- **deleted** filters
- **renamed** filters
- **reordered** filters

References to filter pools can be removed

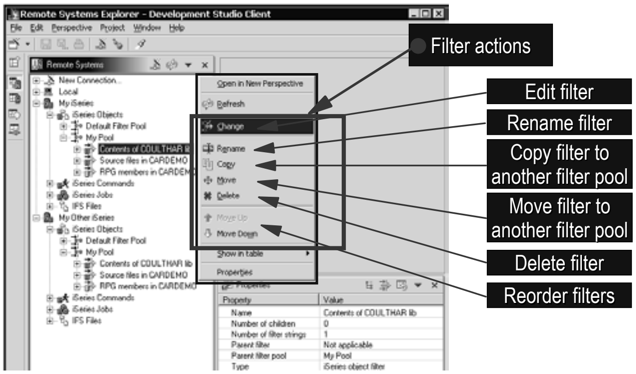
To populate a filter pool with filters, right click on it and select New->XXX Filter..., choosing the type of filter you want to create. The New Filter wizard you get is the same as we already described for the Your Libraries, Your Objects and Your Members prompts. Those are simply shortcuts to these menu items. As you create filters in the filter pool, they appear in the Remote Systems view underneath each reference to this filter pool.

Filter Actions

- Popup menu actions for filters:

  Update parent filter pool that contains the filter

  Reflected in all subsystems that reference pool



Property	Value
Name	Contents of COULTHAR lib
Number of children	0
Number of filter strings	1
Parent filter	Not applicable
Parent filter pool	My Pool
Type	iSeries object filter

Filter actions

Edit filter

Rename filter

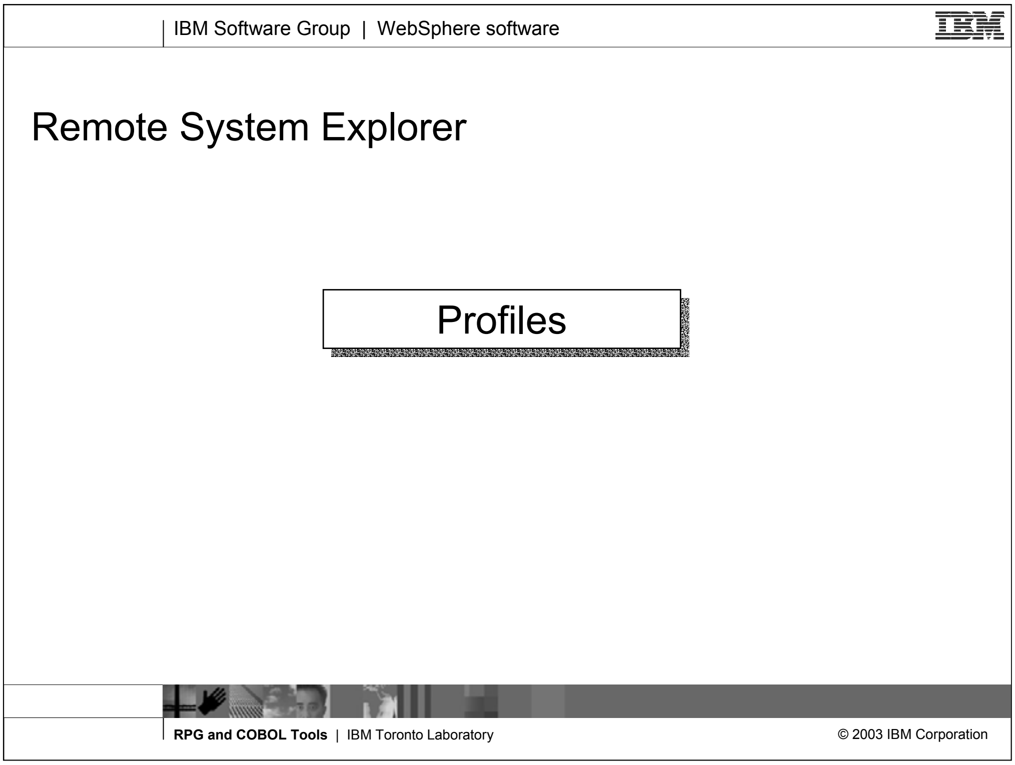
Copy filter to another filter pool

Move filter to another filter pool

Delete filter

Reorder filters

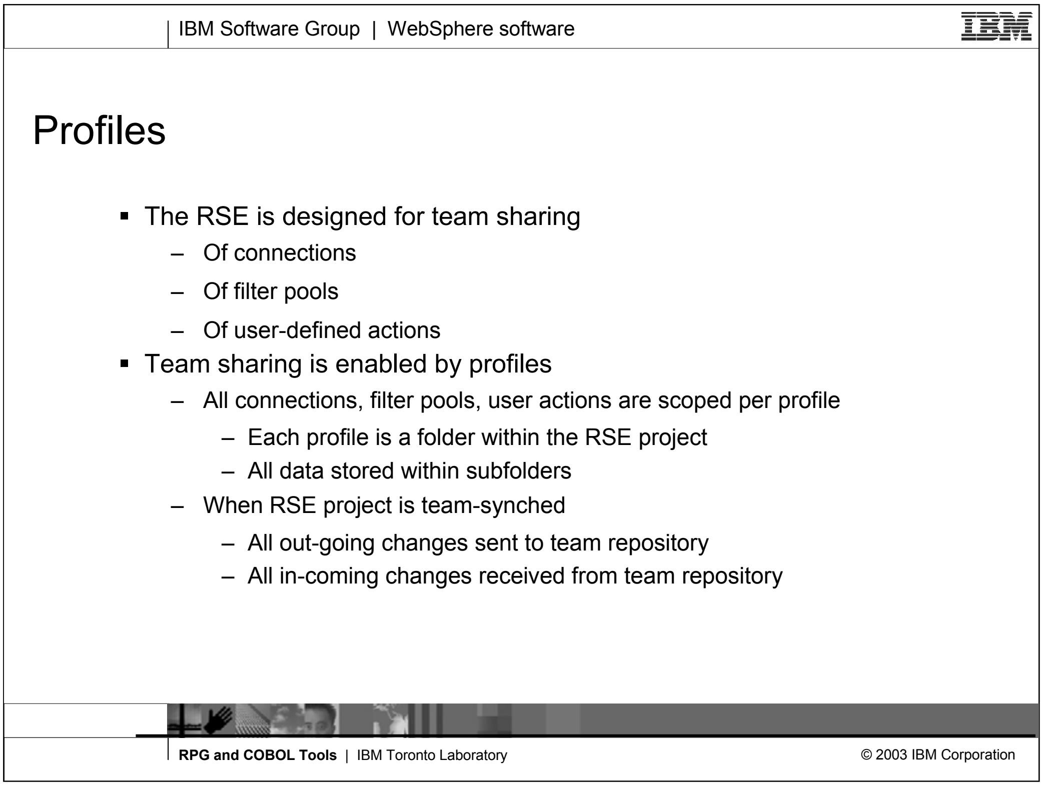
You can do much with filters by using the right click popup menu. For example, you can edit, rename, copy, move, delete and reorder filters. All these actions affect the parent filter pool for the selected filters, and all references to that filter pool will be updated automatically to reflect the changes.



Remote System Explorer

Profiles

You can use profiles to share connections, filter pools, user defined actions and compile commands for team application development.



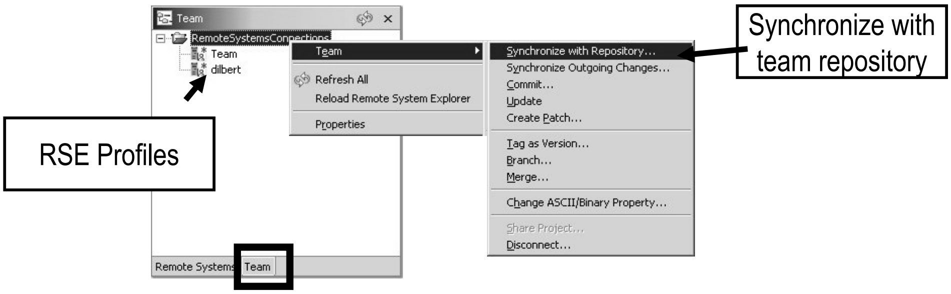
Profiles

- The RSE is designed for team sharing
  - Of connections
  - Of filter pools
  - Of user-defined actions
- Team sharing is enabled by profiles
  - All connections, filter pools, user actions are scoped per profile
    - Each profile is a folder within the RSE project
    - All data stored within subfolders
  - When RSE project is team-synched
    - All out-going changes sent to team repository
    - All in-coming changes received from team repository

We see that with the RSE you will be creating connections, filter pools, user defined actions and compile commands. In a team environment, we might wish to share some or all of this information to save the effort of each team member having to redundantly create them. This is especially for team members working on a shared task, such as mainenance of an application. The RSE leverages the Eclipse team support to enable this.

The first order of business for effectively enable team support is to allow delineation between "team" information and "private" information per developer. The RSE enables this by using profiles. Every bit of information you create in the RSE is owned by a specific profile, and each developer decides which profiles they wish to see the information of. Team sharing is done using the Eclipse team support, which uses a central repository, and a "synchronize" action to synchronize individual developer's information with the repository.

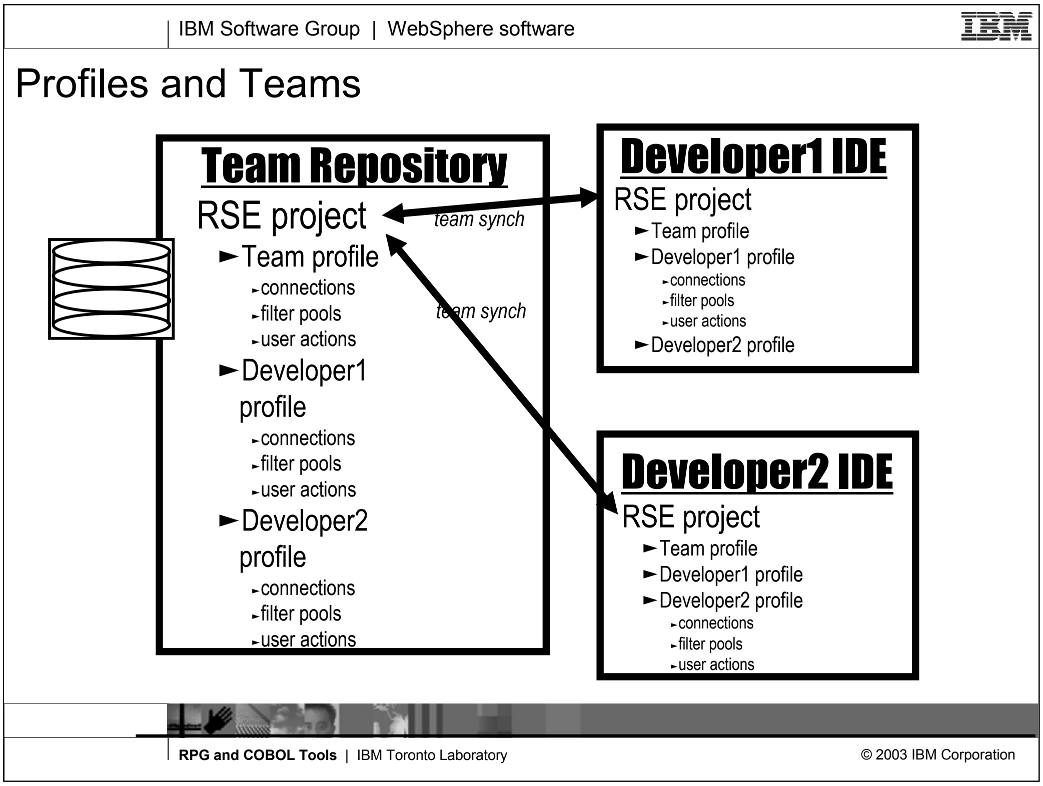
Persistent data stored in Eclipse Project



Synchronize with team repository

RSE Profiles

The RSE uses a single Eclipse project to hold all of the data created within it by a user. The project is named RemoteSystemsConnections. It is visible in Eclipse-supplied Navigator view. This project inherits the team support that all Eclipse projects have, which includes the ability to be associated with a central repository (Team->Properties) and to synchronized with that repository (Team->Synchronize with Stream).



Profiles and Teams

Team Repository

RSE project

- ►Team profile
  - ▸connections
  - ▸filter pools
  - ▸user actions
- ►Developer1 profile
  - ▸connections
  - ▸filter pools
  - ▸user actions
- ►Developer2 profile
  - ▸connections
  - ▸filter pools
  - ▸user actions

team synch

team synch

Developer1 IDE

RSE project

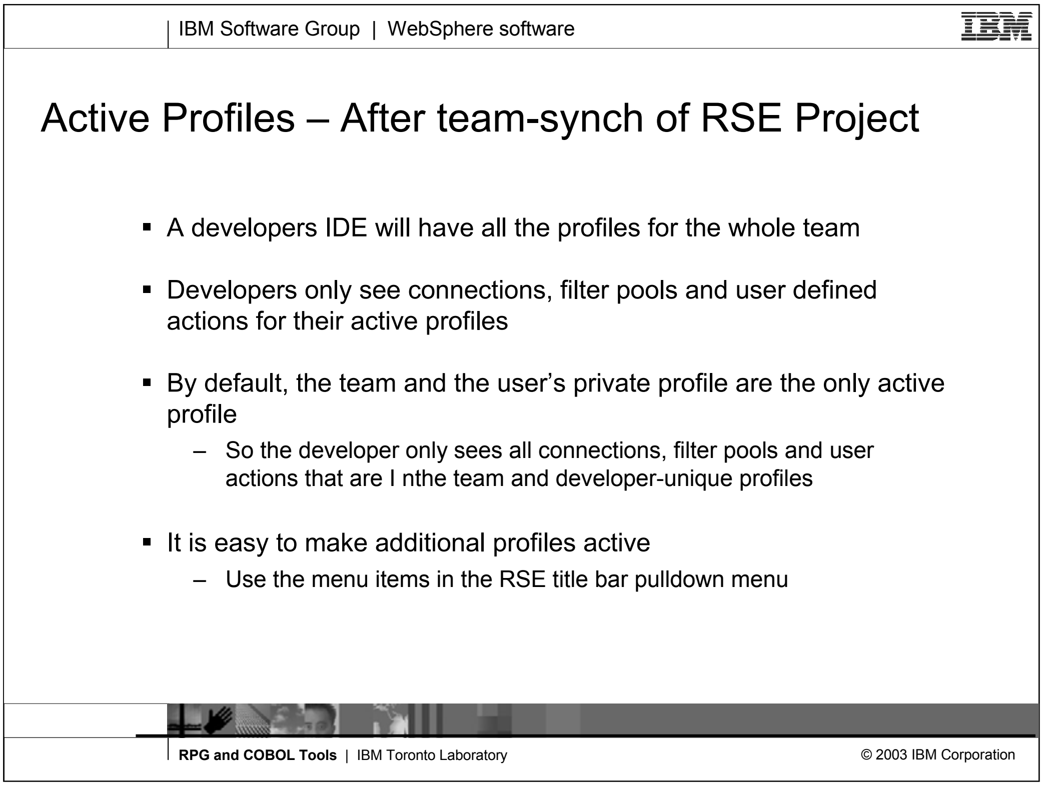
- ►Team profile
- ►Developer1 profile
  - ▸connections
  - ▸filter pools
  - ▸user actions
- ►Developer2 profile

Developer2 IDE

RSE project

- ►Team profile
- ►Developer1 profile
- ►Developer2 profile
  - ▸connections
  - ▸filter pools
  - ▸user actions

Here we see how a central repository relates to each developer's IDE, with respect to the RSE information the team will share. You should notice that every developer actually has all the information for every profile within the team, when team support is used. We will see why this is and how it is handled...

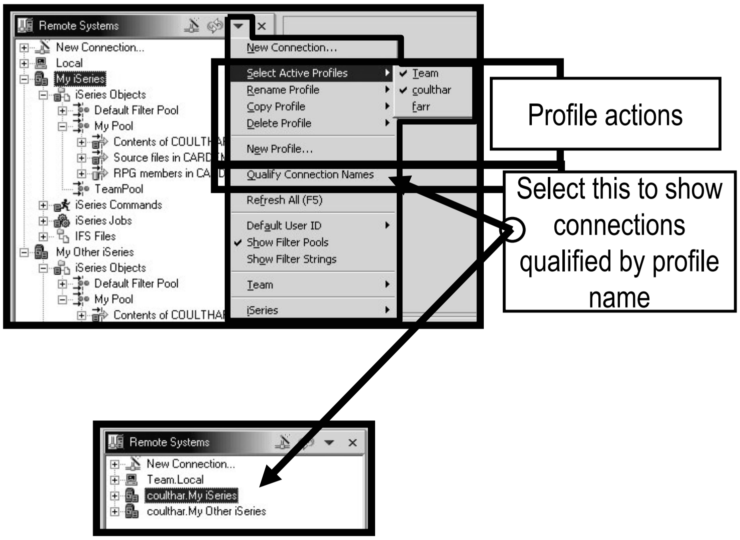


Active Profiles – After team-synch of RSE Project

- A developers IDE will have all the profiles for the whole team
- Developers only see connections, filter pools and user defined actions for their active profiles
- By default, the team and the user's private profile are the only active profile
  - So the developer only sees all connections, filter pools and user actions that are I nthe team and developer-unique profiles
- It is easy to make additional profiles active
  - Use the menu items in the RSE title bar pulldown menu

Because a team-synchronize action copies all the contents of the project from the central repository to the target developer's IDE (and vice versa), after such an action the developer will have all the profiles for all his team members in his/her IDE. This could be overwhelming to see connections and filter pools and user actions and compile commands for the whole team. So, by default the RSE only shows a subset of these. It shows only the information for those profiles which are "active", which by default is the "team" profile and the unique profile for that developer. Should there be a desire to see information from other profiles, this is easily enabled by making those profiles active too. We will see how to do this...

Profile Actions

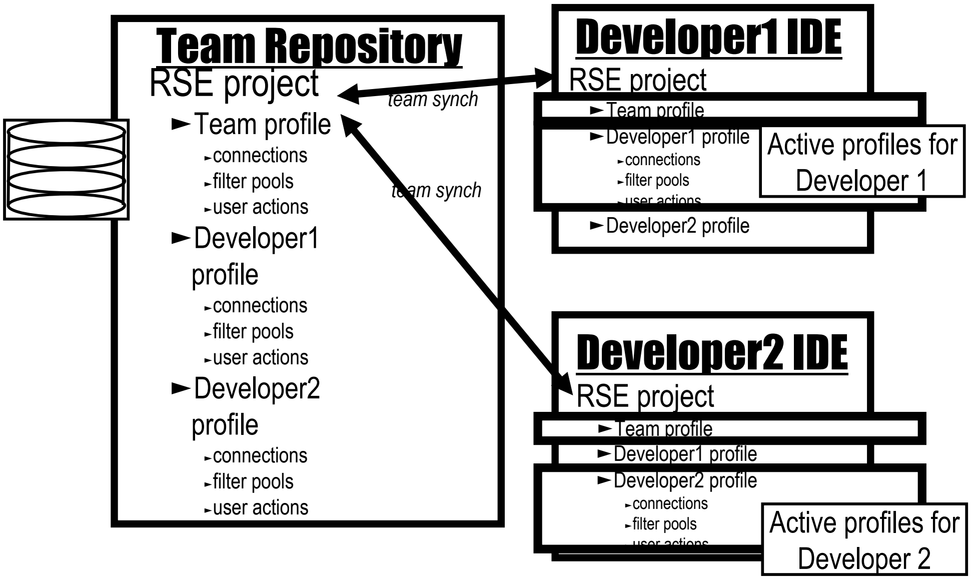


Profile actions

Select this to show connections qualified by profile name

The pulldown menu from the title bar of the Remote Systems view contains all the profile menu actions. From here, you can choose which profiles are active, and you can create, rename, copy and delete profiles.A very important preference setting in this menu is "Qualify Connection Names". When toggled on, this changes the Remote Systems view to show each connection prefixed by the profile it comes from. This can be handy when working with both shared and private connections.

Profiles and Teams



Team Repository

RSE project

- Team profile
  - connections
  - filter pools
  - user actions
- Developer1 profile
  - connections
  - filter pools
  - user actions
- Developer2 profile
  - connections
  - filter pools
  - user actions

team synch

team synch

Developer1 IDE

RSE project

- Team profile
- Developer1 profile
  - connections
  - filter pools
  - user actions
- Developer2 profile

Active profiles for Developer 1

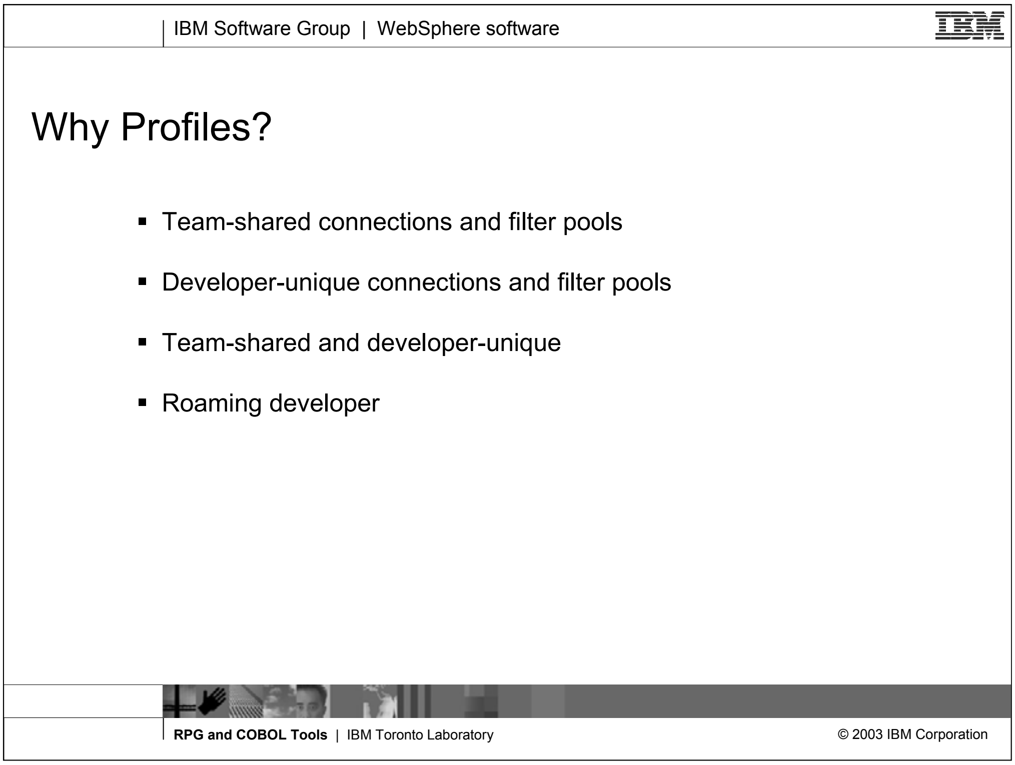
Developer2 IDE

RSE project

- Team profile
- Developer1 profile
- Developer2 profile
  - connections
  - filter pools
  - user actions

Active profiles for Developer 2

Here we see how active profiles keep team members from being flooded with the information from all other team members. While each workstation has all the connections, filter pools, user actions and compile commands for all team members, by default each developer only sees their own and those that were created in the team profile and hence intended to be seen and shared by all.



IBM Software Group | WebSphere software

Why Profiles?

- Team-shared connections and filter pools
- Developer-unique connections and filter pools
- Team-shared and developer-unique
- Roaming developer

RPG and COBOL Tools | IBM Toronto Laboratory

© 2003 IBM Corporation

There are many good reasons for the use of profiles within the Remote System Explorer:

Team-shared connections and filter pools

•So everyone on the team sees common connections for the common library list, environment settings and filter pools

•For common development work, for example, define profiles such as “team”, “fix team”, “project1 team”, “release 1”, “emergency fix”, and so on

Developer-unique connections and filter pools

•So developers can have connections with unique library list, environment settings and filter pools

•For developer unique work through their unique profiles

•Each such private profile is only made active by one developer

Team-shared and developer-unique

•Each developer sees connections and filter pools from both shared and private profiles: whatever is “active”

Roaming developer

•On PC1: Developer synchronizes with team repository

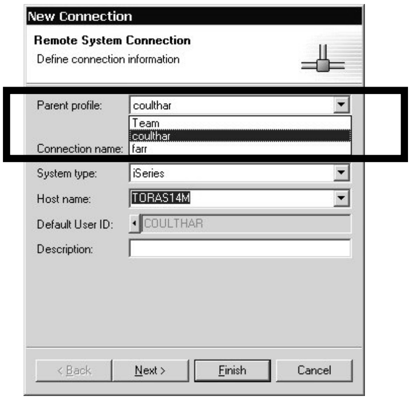
•Sends all connections, filter pools, actions to repository

•On PC2: Developer synchronizes RSE with team repository

•Gets all connections, filter pools, actions from repository

•Makes his profile “active”

Active profiles



New Connection

Remote System Connection

Define connection information

Parent profile: coulthar

Team
coulthar
farr

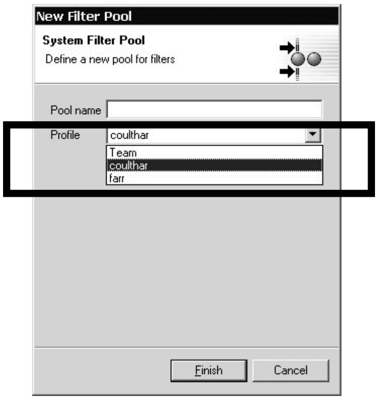
Connection name:

System type: iSeries

Host name: TORAS14M

Default User ID: COULTHAR

Description:



New Filter Pool

System Filter Pool

Define a new pool for filters

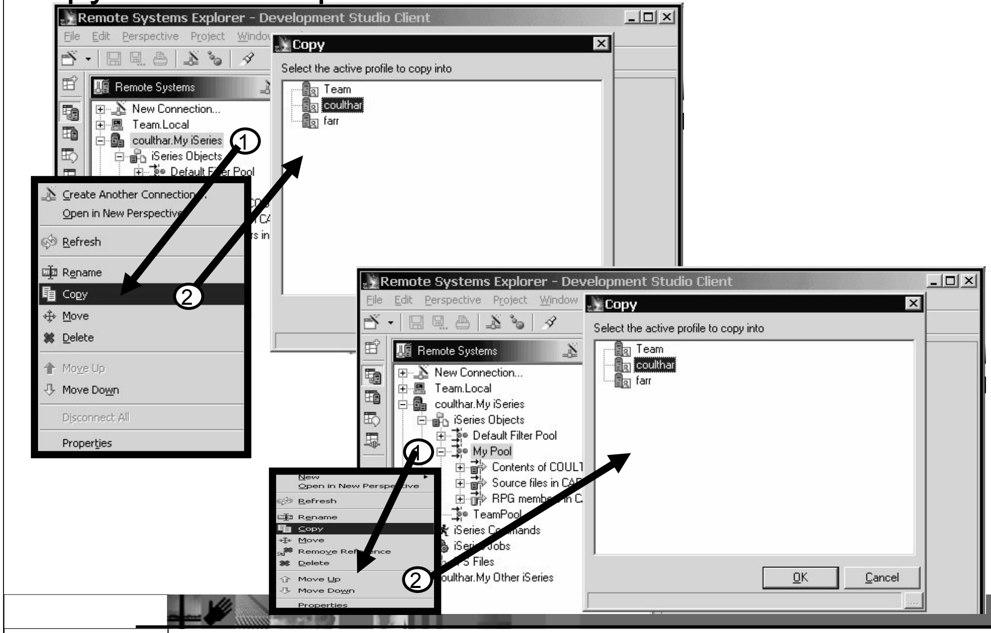
Pool name

Profile coulthar

Team
coulthar
farr

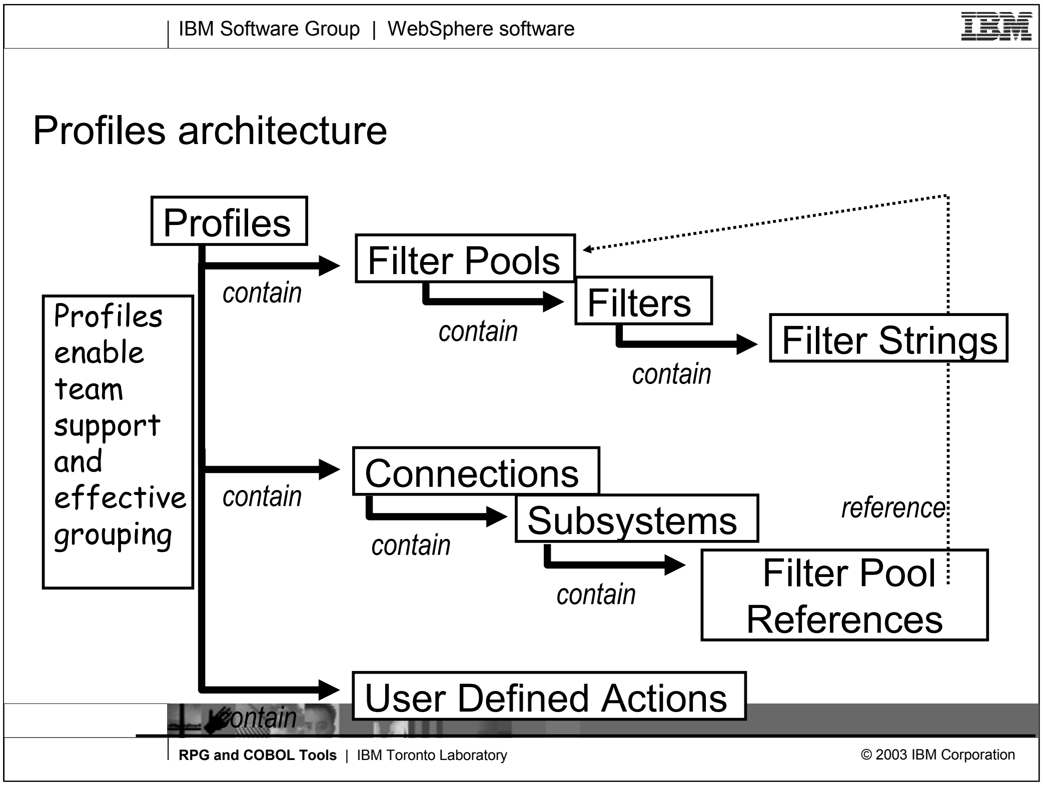
We see now why both the connection and the filter pool wizards contain a dropdown for selecting the target profile. These both default to the first non-team active profile. When you create connections and filter pools you must specify the active profile to contain the new connection or profile.

Copy/move filter pools or connections



Select the active profile to copy into

Here we seen how we can use the copy and move actions to move information between profiles.



Profiles architecture

Profiles

contain

Filter Pools

contain

Filters

contain

Filter Strings

Profiles enable team support and effective grouping

contain

Connections

contain

Subsystems

contain

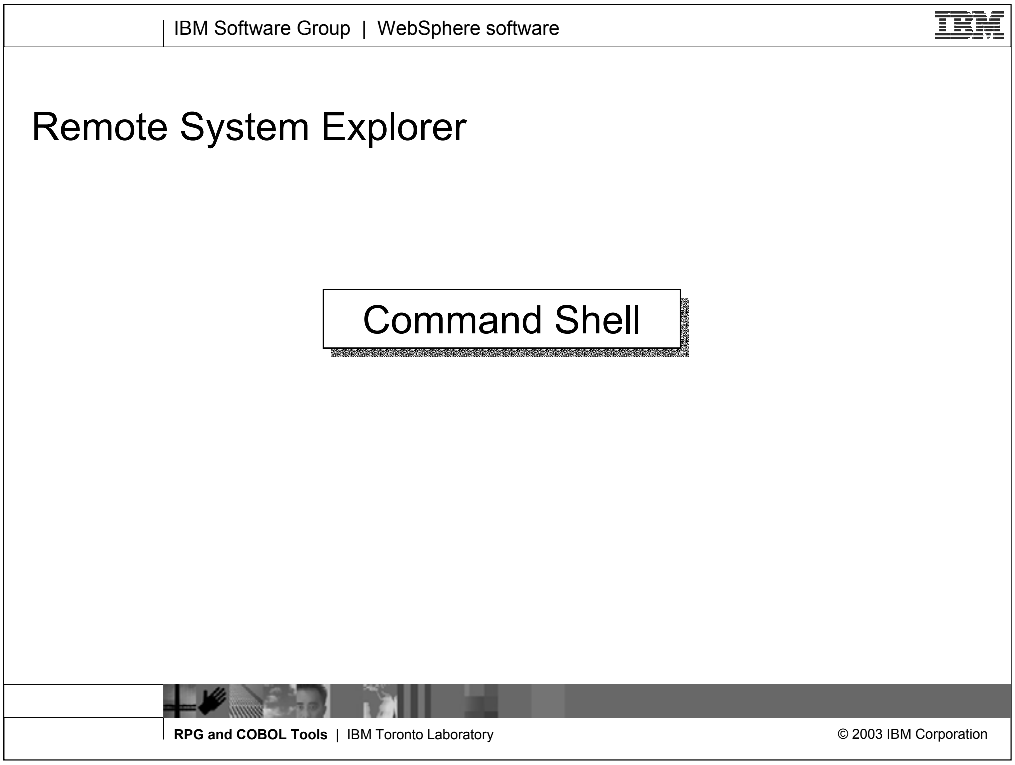
Filter Pool References

reference

contain

User Defined Actions

This is an architecture chart showing the underlying model of all the information that the RSE maintains. For those with an architectural bent, this may help you to visualize all the pieces we have been talking about.

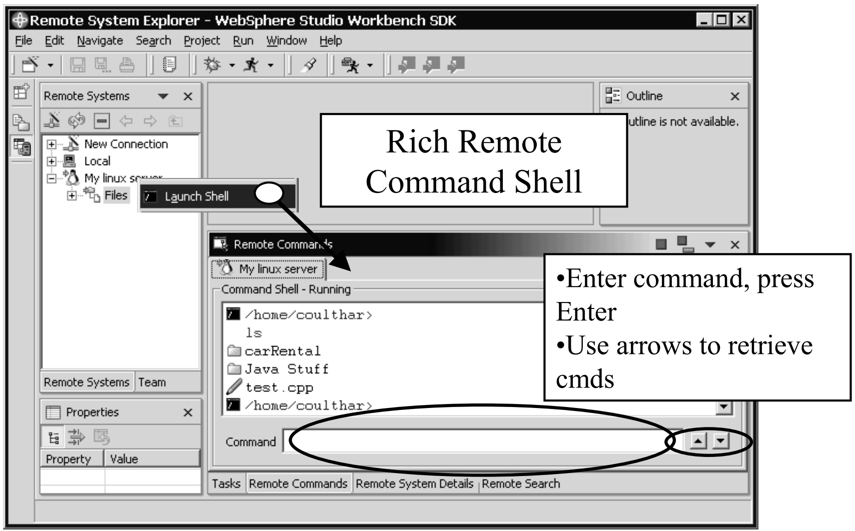


Remote System Explorer

Command Shell

You can use the Remote Commands view to run and interact with commands and command shells on universal systems. A universal system includes Windows, Linux, and UNIX system types.

RSE Command Shell


Remote System Explorer - WebSphere Studio Workbench SDK
File Edit Navigate Search Project Run Window Help
Remote Systems
New Connection
Local
My linux server
Files
Launch Shell
Outline
Outline is not available.
Remote Commands
My linux server
Command Shell - Running
Command
Remote Systems Team
Properties
Property Value
Tasks Remote Commands Remote System Details Remote Search

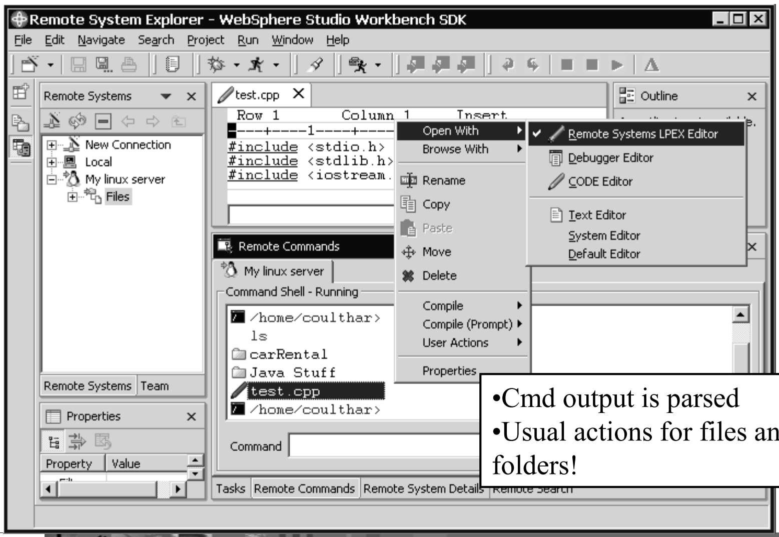
```
/home/coulthar>
 ls
carRental
Java Stuff
test.cpp
/home/coulthar>
```

Rich Remote Command Shell

- Enter command, press Enter
- Use arrows to retrieve cmds

Here you enter input to the program.

RSE Command Shell



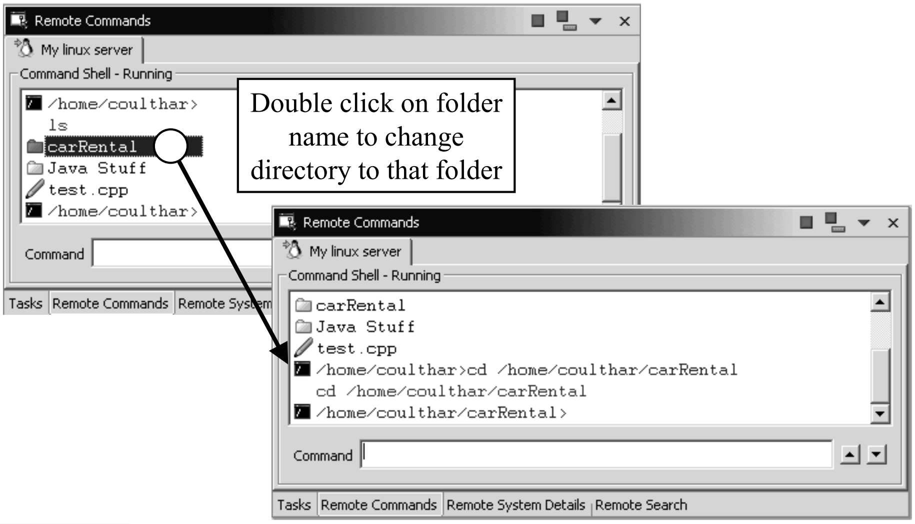
```
Row 1        Column 1       Insert
#include <stdio.h>
#include <stdlib.h>
#include <iostream.
```

```
/home/coulthar>
  ls
carRental
Java Stuff
test.cpp
/home/coulthar>
```

- Cmd output is parsed
- Usual actions for files and folders!

Next you see the output for the program. You can invoke actions on the output in the Command Shell.

RSE Command Shell



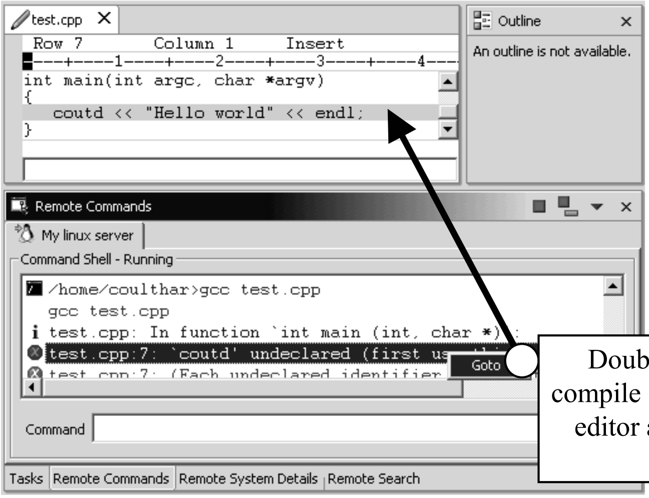
```
/home/coulthar>
ls
carRental
Java Stuff
test.cpp
/home/coulthar>
```

Double click on folder name to change directory to that folder

```
carRental
Java Stuff
test.cpp
/home/coulthar>cd /home/coulthar/carRental
cd /home/coulthar/carRental
/home/coulthar/carRental>
```

Its easy to change the directory you want to work with in the Command Shell. Just double click on the folder.

RSE Command Shell



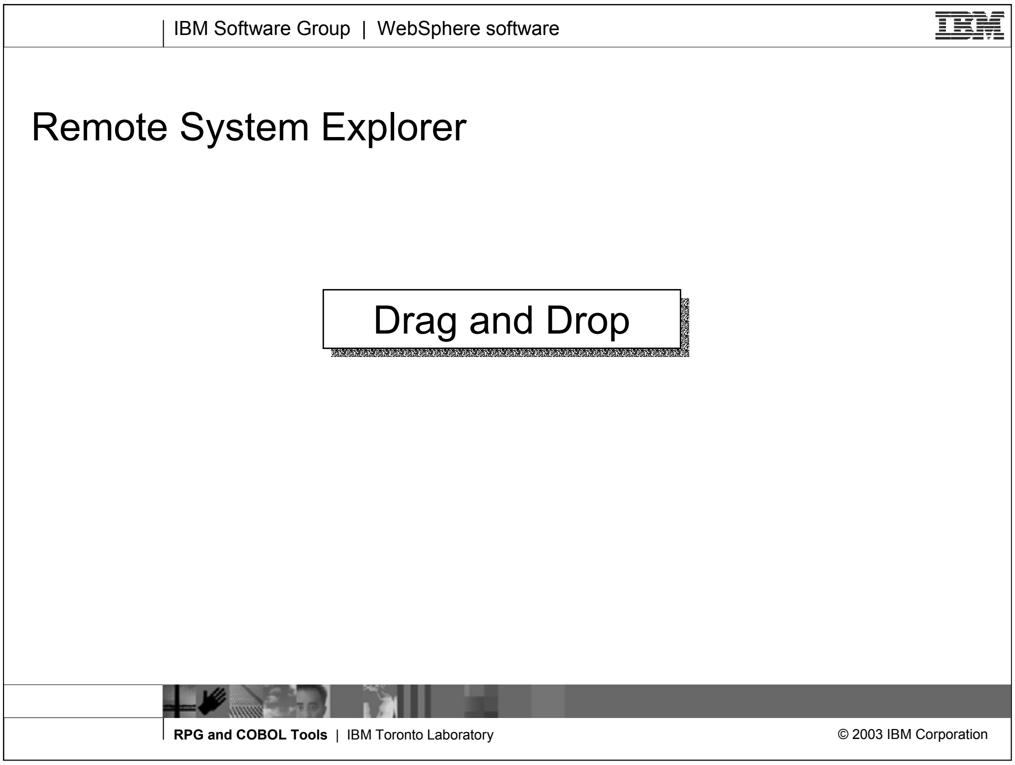
```
Row 7        Column 1      Insert
----+----1----+----2----+----3----+----4---
int main(int argc, char *argv)
{
   coutd << "Hello world" << endl;
}
```

An outline is not available.

```
/home/coulthar>gcc test.cpp
gcc test.cpp
test.cpp: In function `int main (int, char *)':
test.cpp:7: `coutd' undeclared (first us
test.cpp:7: (Each undeclared identifier
```

Double click on compile errors to open editor and jump to error

Double-click on compile errors to open the editor and jump to the line in the source where the error occurred.

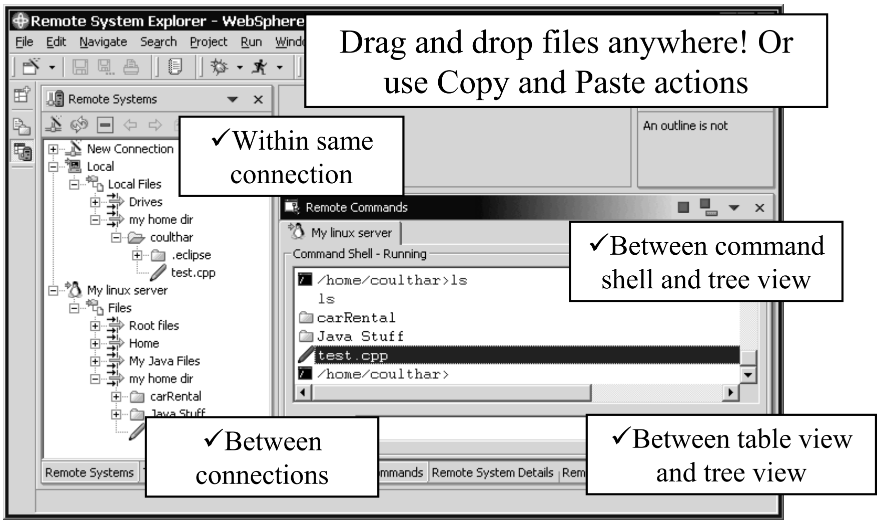


Remote System Explorer

Drag and Drop

▪Extensive support for dragging and dropping (Programmers can use drag and drop to copy source members between files on the same iSeries, or different iSeries. They can even copy members to and from their local Windows file system, or a remote UNIX, Windows, or Linux file system.)

RSE Drag and Drop



Drag and drop files anywhere! Or use Copy and Paste actions

✓Within same connection

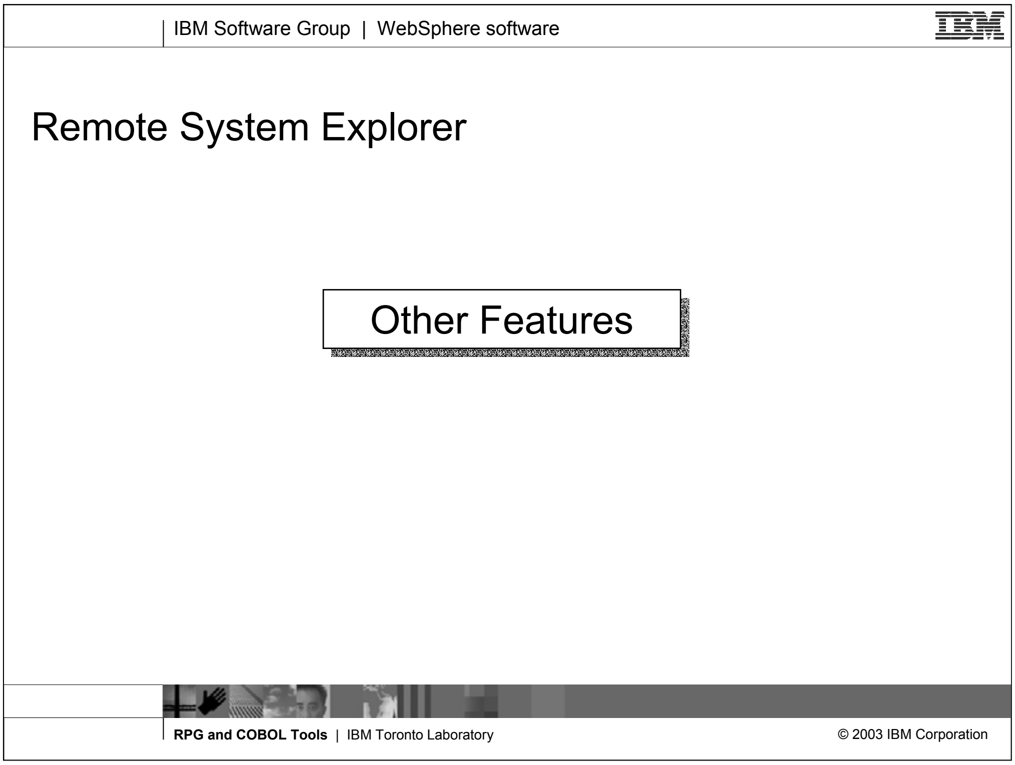
✓Between command shell and tree view

✓Between connections

✓Between table view and tree view

```
/home/coulthar>ls
 ls
carRental
Java Stuff
test.cpp
/home/coulthar>
```

Here you can see that you can drag and drop files or copy and paste files across multiple file systems, within the same connection and using the Remote Systems view, iSeries Table view or the Command Shell.

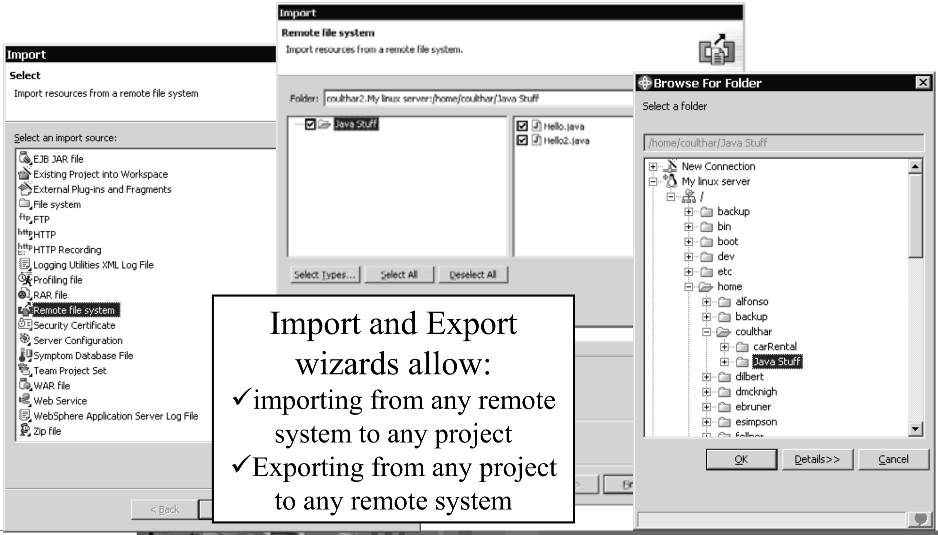


Remote System Explorer

Other Features

Other features include import and export file options, reusable parts, adding your own tools to RSE, and finding remote files.

File Import and Export



Import

Select

Import resources from a remote file system

Select an import source:

Import

Remote file system

Import resources from a remote file system.

Folder: coulthar2.My linux server:/home/coulthar/Java Stuff

Browse For Folder

Select a folder

/home/coulthar/Java Stuff

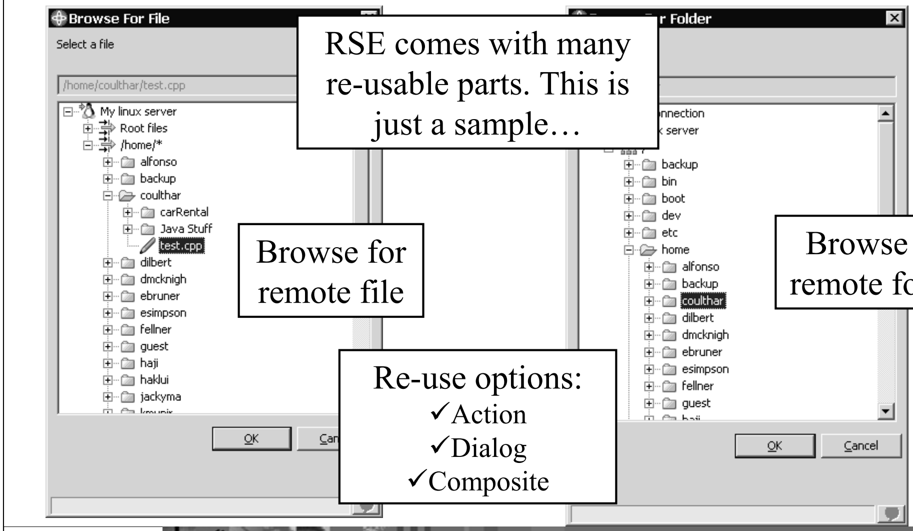
Import and Export wizards allow:

✓importing from any remote system to any project

✓Exporting from any project to any remote system

You can easily import and export files into the workbench.

RSE Re-Usable Parts



RSE comes with many re-usable parts. This is just a sample…

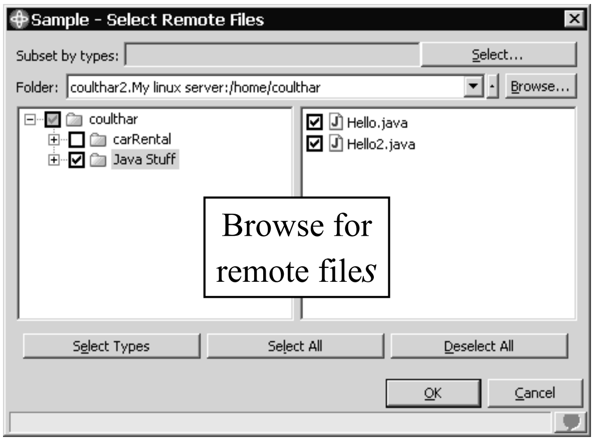
Browse for remote file

Browse for remote folder

Re-use options:
✓Action
✓Dialog
✓Composite

You can reuse RSE actions and dialogs.

RSE More Re-Usable Parts



Sample - Select Remote Files

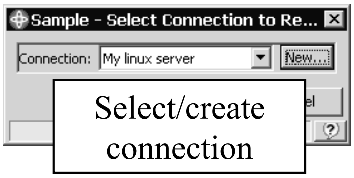
Subset by types:

Folder: coulthar2.My linux server:/home/coulthar

- ⊟ ☑ coulthar
  - ⊞ □ carRental
  - ⊞ ☑ Java Stuff

☑ Hello.java
☑ Hello2.java

Browse for remote files



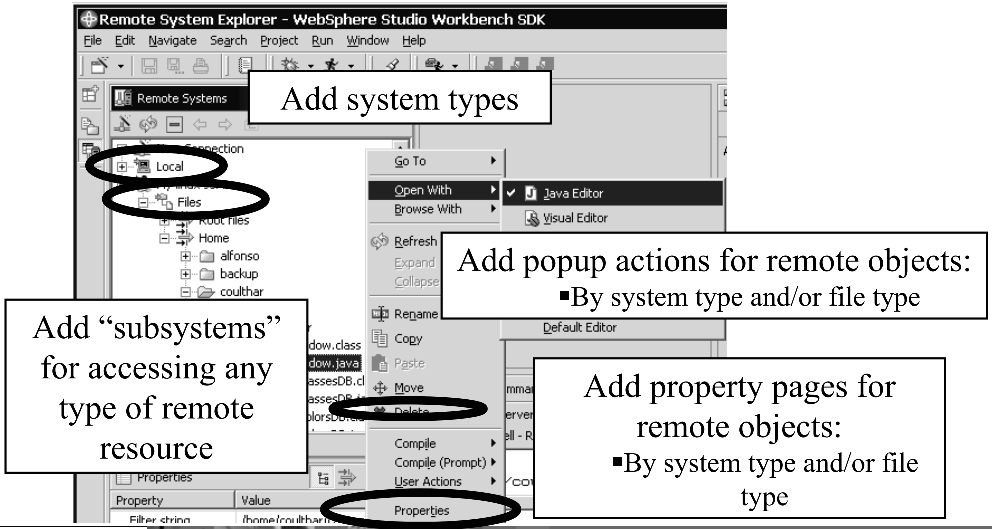
Sample - Select Connection to Re...

Connection: My linux server

Select/create connection

Here are some more reusable parts.

RSE Extension Points



Add system types

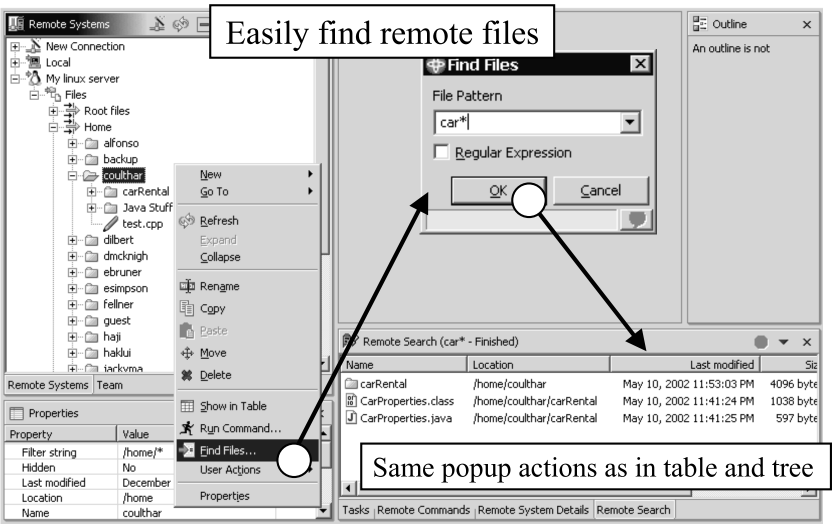
Add "subsystems" for accessing any type of remote resource

Add popup actions for remote objects:
- By system type and/or file type

Add property pages for remote objects:
- By system type and/or file type

There are tool extensions into the Remote Systems Explorer for vendors to interface and exploit remote systems.

RSE Find Files



Easily find remote files

Same popup actions as in table and tree

Property	Value
Filter string	/home/*
Hidden	No
Last modified	December
Location	/home
Name	coulthar

Name	Location	Last modified	Siz
carRental	/home/coulthar	May 10, 2002 11:53:03 PM	4096 byte
CarProperties.class	/home/coulthar/carRental	May 10, 2002 11:41:24 PM	1038 byte
CarProperties.java	/home/coulthar/carRental	May 10, 2002 11:41:25 PM	597 byte

You can easily find remote files using the Find Files option.

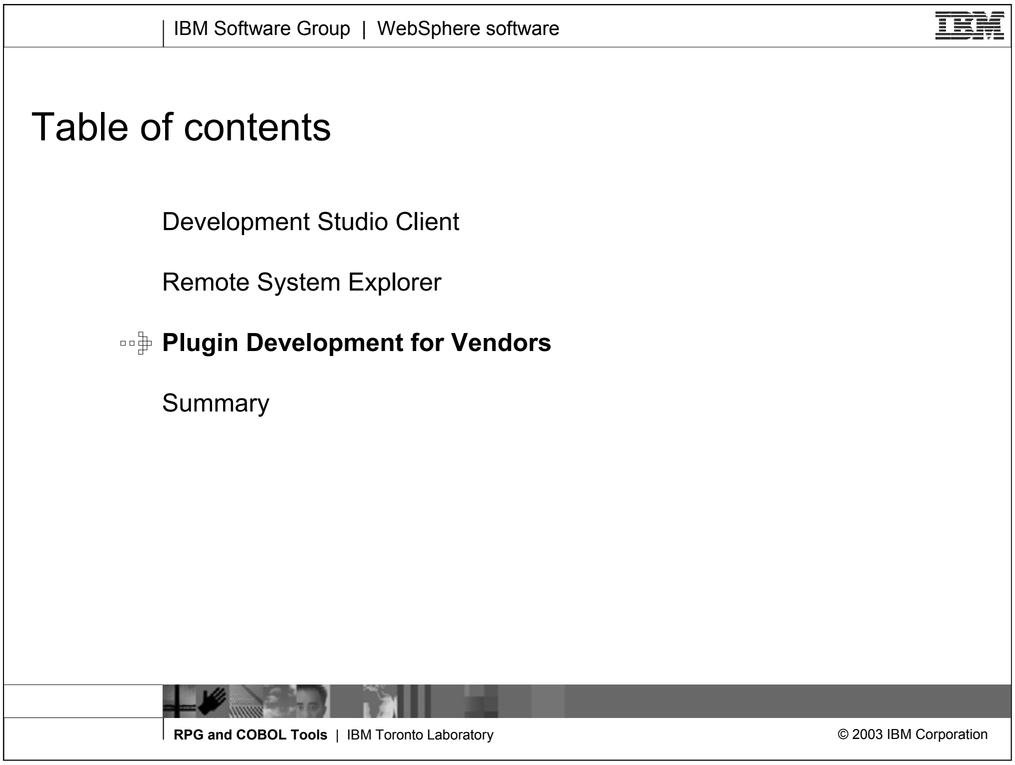


Table of contents

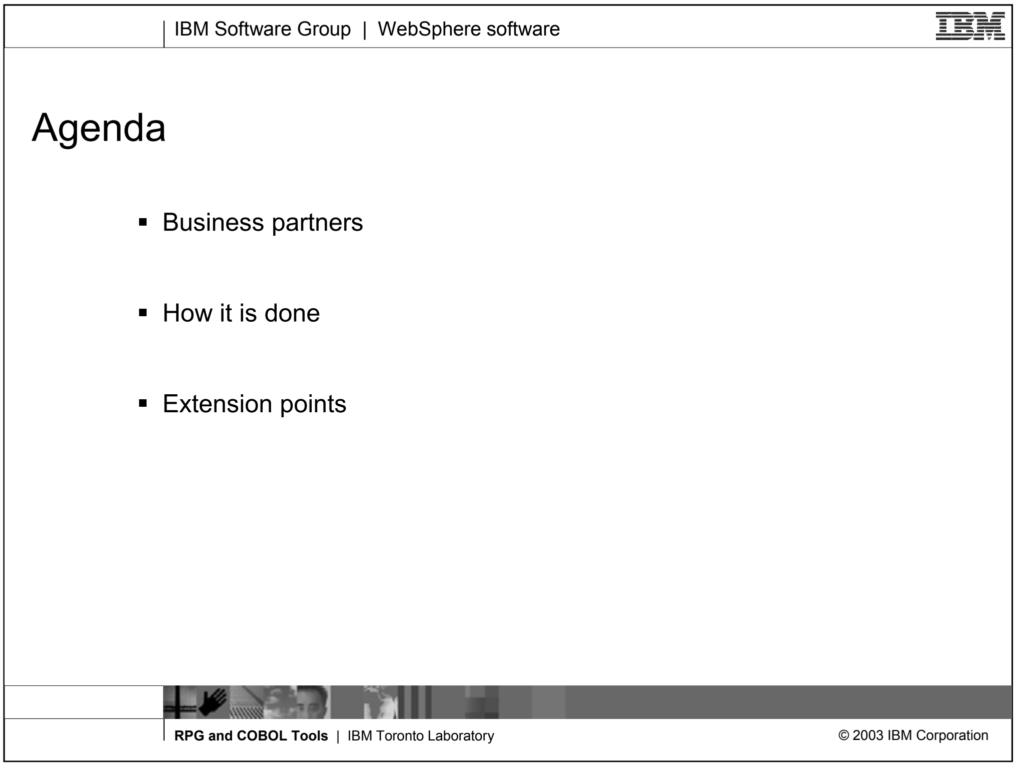
Development Studio Client

Remote System Explorer

Plugin Development for Vendors

Summary

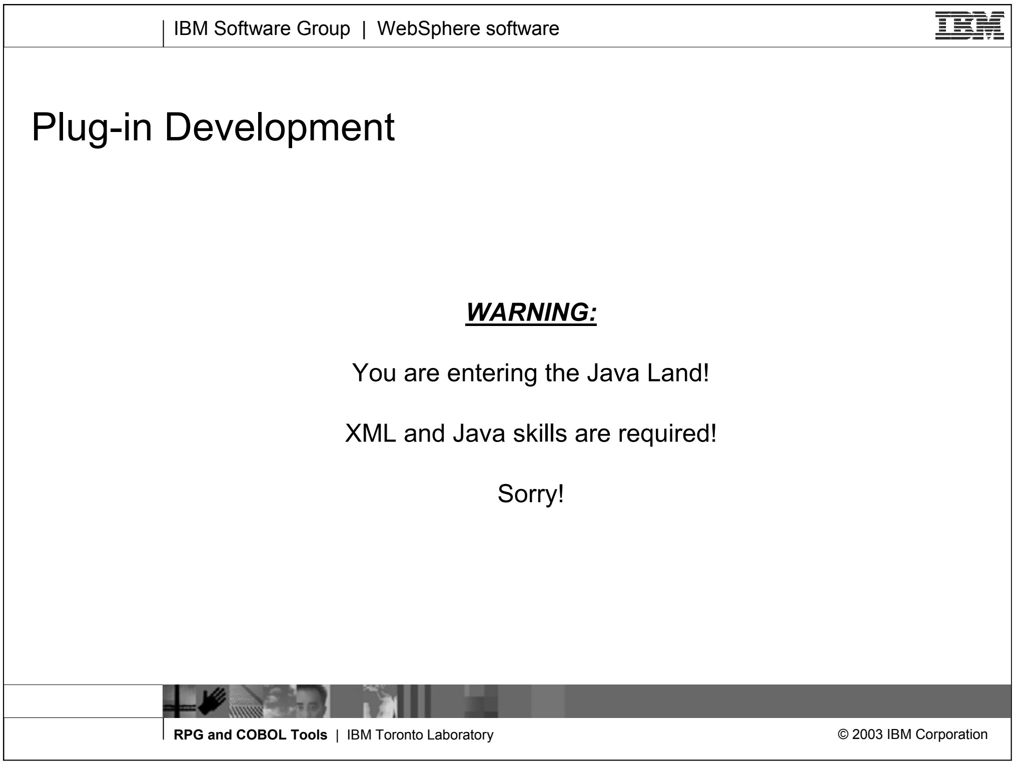
Now we have reviewed all the features of the RSE. Business partners can easily leverage and extend RSE.



Agenda

- Business partners
- How it is done
- Extension points

This section describes business partners role in extending RSE, how to extend and what the extension points are.



Plug-in Development

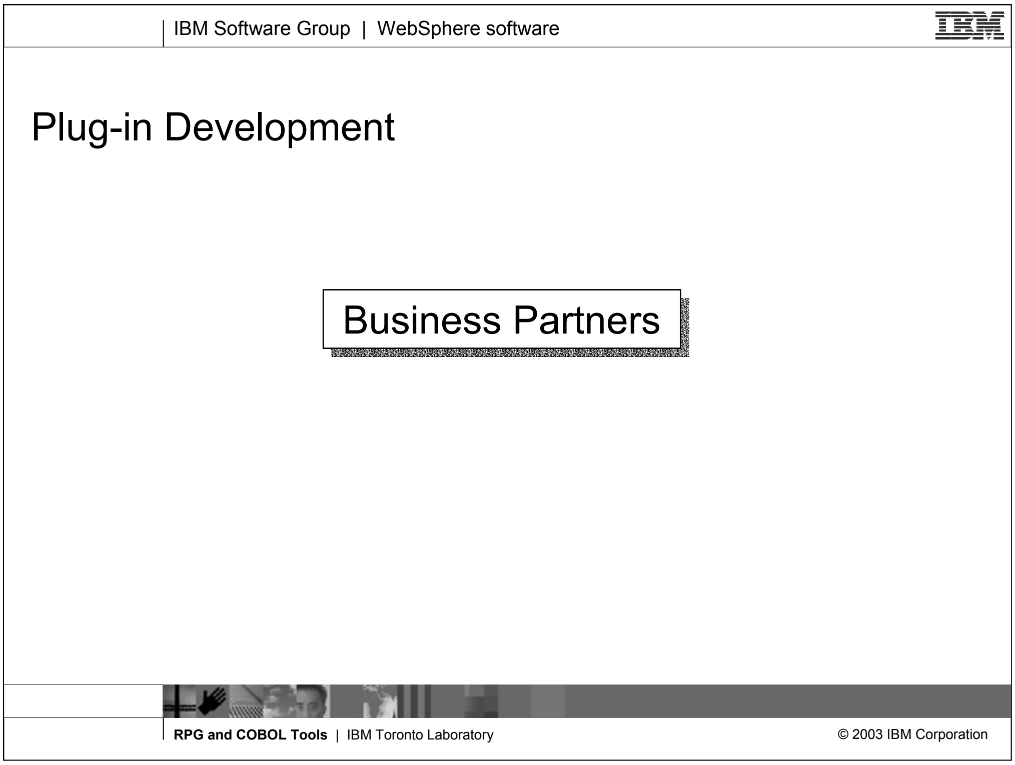
WARNING:

You are entering the Java Land!

XML and Java skills are required!

Sorry!

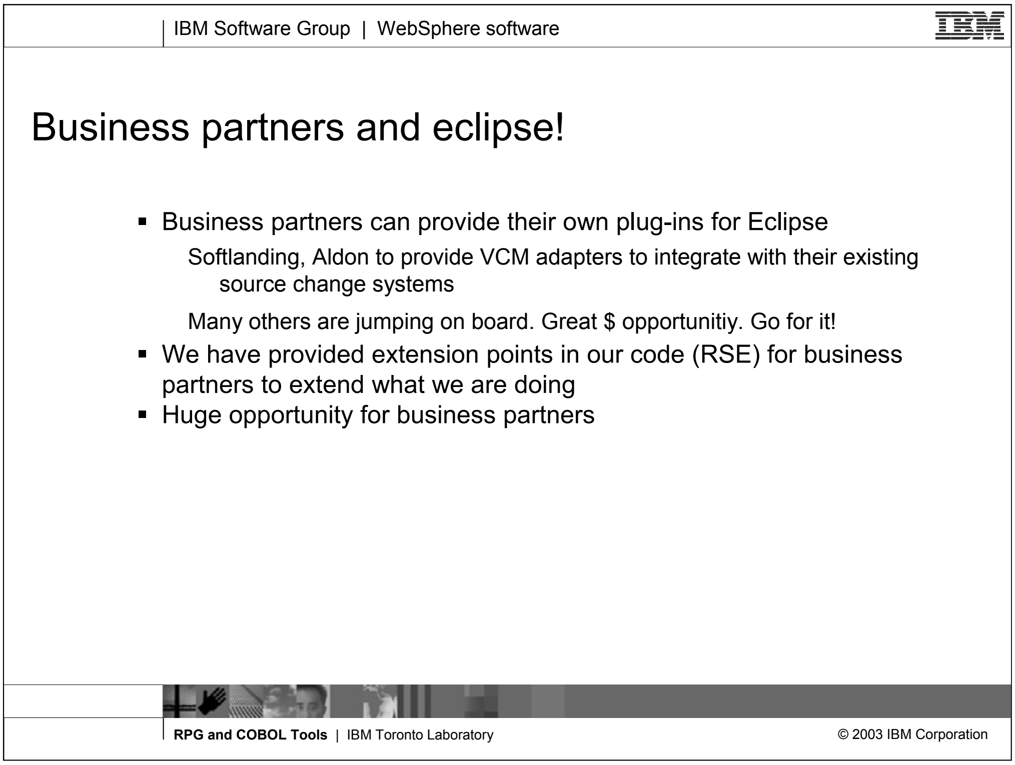
To do plug-in development you must have Java and XML skills.



Plug-in Development

Business Partners

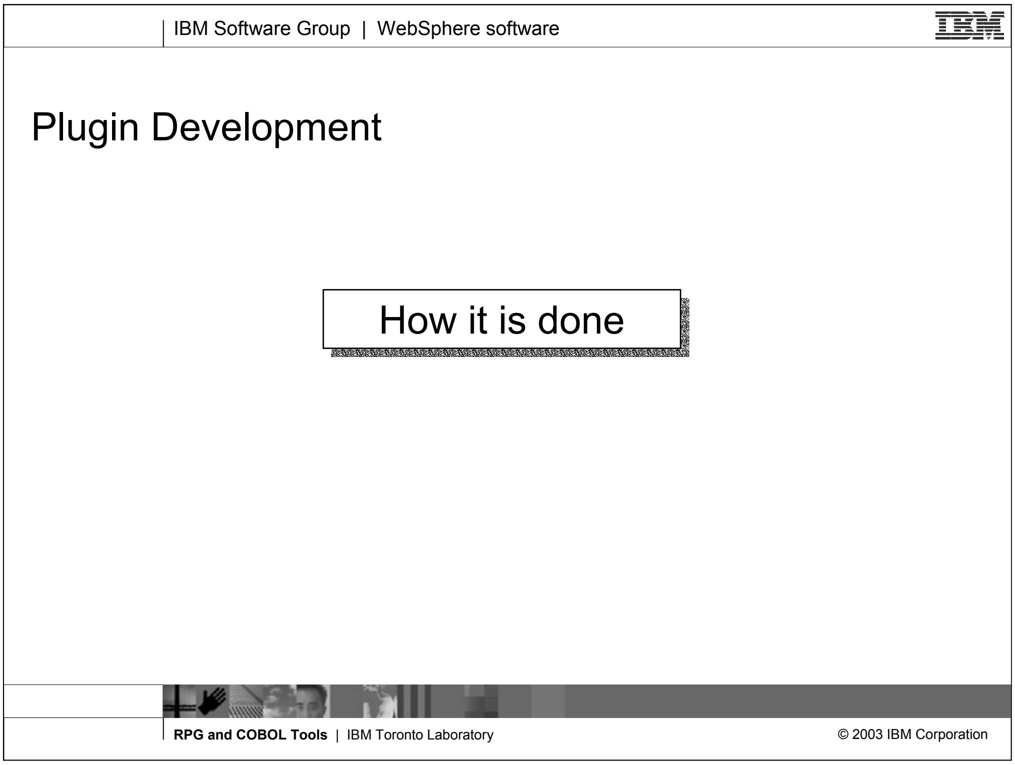
Many leading iSeries tool vendors are actively integrating their tools into this new release. These vendors include, but are not limited to change management and impact analysis tool suppliers.



Business partners and eclipse!

- Business partners can provide their own plug-ins for Eclipse
  - Softlanding, Aldon to provide VCM adapters to integrate with their existing source change systems
  - Many others are jumping on board. Great $ opportunitiy. Go for it!
- We have provided extension points in our code (RSE) for business partners to extend what we are doing
- Huge opportunity for business partners

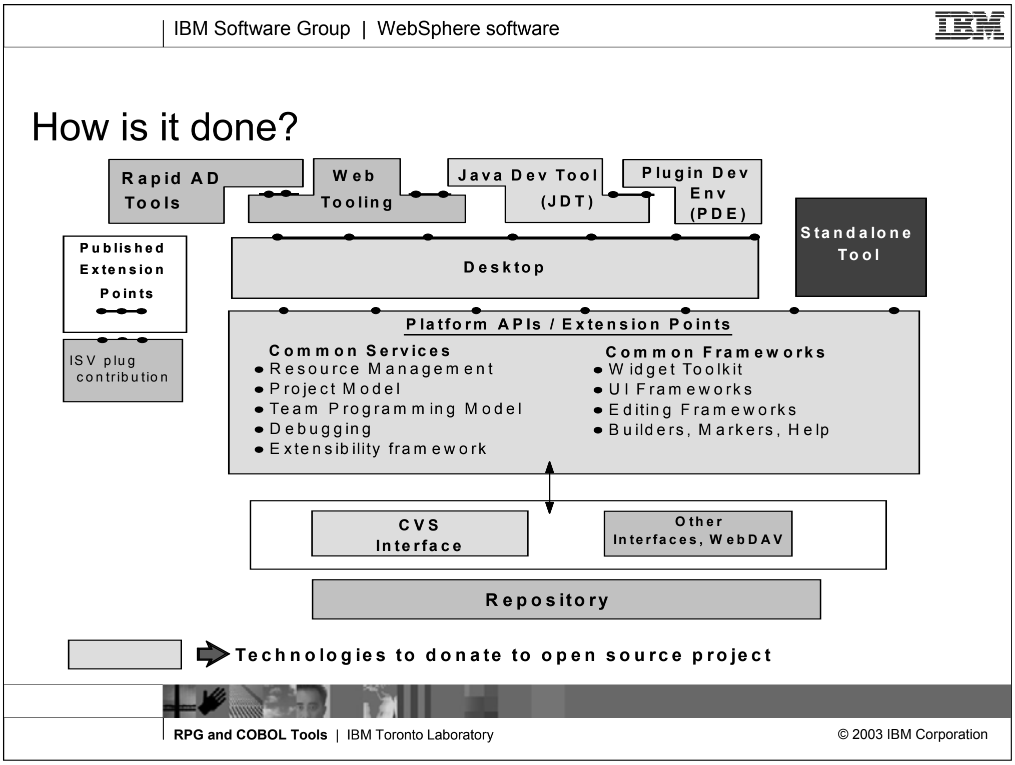
Business partners can provide their own tools and plug them into the Remote System Explorer. This is a tremendous opportunity for business partners.



Plugin Development

How it is done

Here is how plug-in your tools.



IBM Software Group | WebSphere software

How is it done?

Rapid AD Tools

Web Tooling

Java Dev Tool (JDT)

Plugin Dev Env (PDE)

Standalone Tool

Published Extension Points

ISV plug contribution

Desktop

Platform APIs / Extension Points

Common Services
- Resource Management
- Project Model
- Team Programming Model
- Debugging
- Extensibility framework

Common Frameworks
- Widget Toolkit
- UI Frameworks
- Editing Frameworks
- Builders, Markers, Help

CVS Interface

Other Interfaces, WebDAV

Repository

Technologies to donate to open source project

RPG and COBOL Tools | IBM Toronto Laboratory

© 2003 IBM Corporation

You can extend the Remote System Explorer by authoring Java code that adds additional items to the perspective, such as pop-up menu actions, property pages, and even subsystems. This feature is valuable if you are a business partner or independent software vendor, and you want to tightly integrate your tools with the Remote System Explorer.

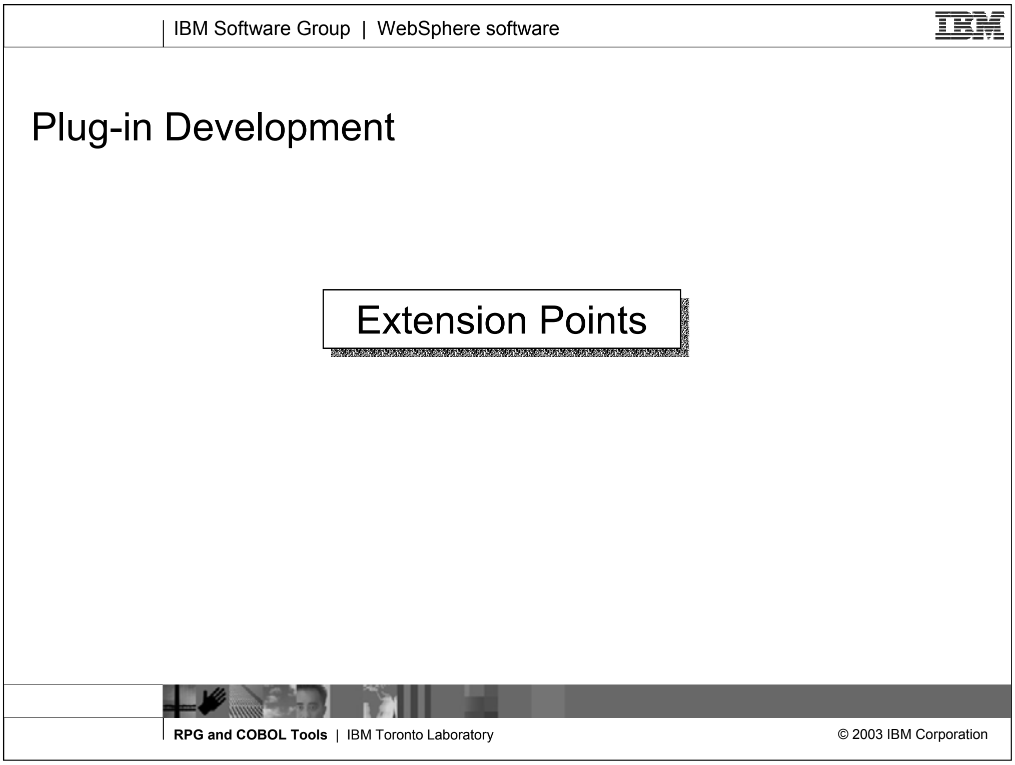
Programmer documentation for extending the Remote System Explorer is currently available in English only. To retrieve the documentation:

Navigate to the plug-in folder com.ibm.etools.systems.doc.isv, in the iseries\plugins directory where you installed this product.

Rename the file toc.xml.off to toc.xml.

Close and re-start the workbench.

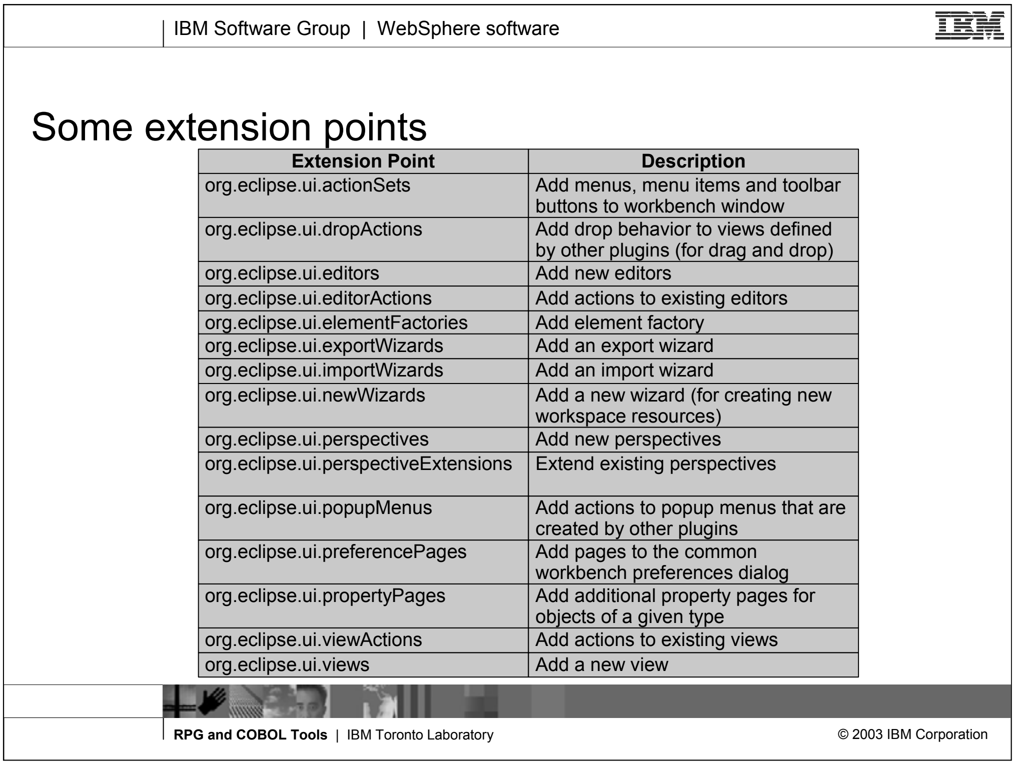
Select Help Contents from the Help menu, and click Extending the Remote System Explorer (RSE) in the Contents frame to read the new programmer documentation.



Plug-in Development

Extension Points

Extension points allow you to plug-in your tools into RSE.



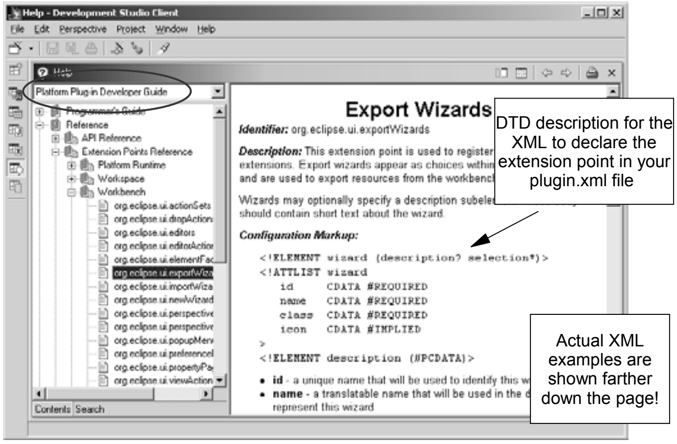
Some extension points

Extension Point	Description
org.eclipse.ui.actionSets	Add menus, menu items and toolbar buttons to workbench window
org.eclipse.ui.dropActions	Add drop behavior to views defined by other plugins (for drag and drop)
org.eclipse.ui.editors	Add new editors
org.eclipse.ui.editorActions	Add actions to existing editors
org.eclipse.ui.elementFactories	Add element factory
org.eclipse.ui.exportWizards	Add an export wizard
org.eclipse.ui.importWizards	Add an import wizard
org.eclipse.ui.newWizards	Add a new wizard (for creating new workspace resources)
org.eclipse.ui.perspectives	Add new perspectives
org.eclipse.ui.perspectiveExtensions	Extend existing perspectives
org.eclipse.ui.popupMenus	Add actions to popup menus that are created by other plugins
org.eclipse.ui.preferencePages	Add pages to the common workbench preferences dialog
org.eclipse.ui.propertyPages	Add additional property pages for objects of a given type
org.eclipse.ui.viewActions	Add actions to existing views
org.eclipse.ui.views	Add a new view

Here are a list of extension points and what they do.

So where do you find all this information?

- The online help!



Help - Development Studio Client
File Edit Perspective Project Window Help
Help
Platform Plug-in Developer Guide
Programmer's Guide
Reference
API Reference
Extension Points Reference
Platform Runtime
Workspace
Workbench
org.eclipse.ui.actionSets
org.eclipse.ui.dropAction
org.eclipse.ui.editors
org.eclipse.ui.editorAction
org.eclipse.ui.elementFac
org.eclipse.ui.exportWiza
org.eclipse.ui.importWiza
org.eclipse.ui.newWizard
org.eclipse.ui.perspective
org.eclipse.ui.perspective
org.eclipse.ui.popupMen
org.eclipse.ui.preferencel
org.eclipse.ui.propertyPa
org.eclipse.ui.viewAction
Contents Search

Export Wizards

Identifier: org.eclipse.ui.exportWizards

Description: This extension point is used to register extensions. Export wizards appear as choices within and are used to export resources from the workbench

Wizards may optionally specify a description subele should contain short text about the wizard.

Configuration Markup:

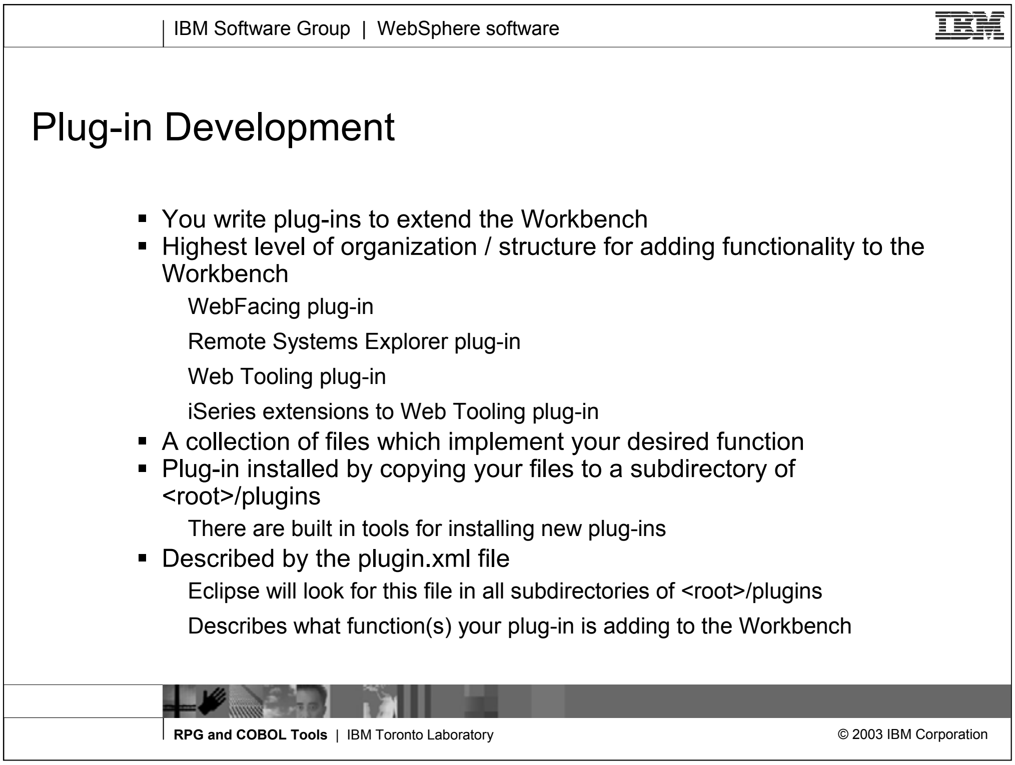
```
<!ELEMENT wizard (description? selection*)>
<!ATTLIST wizard
   id      CDATA #REQUIRED
   name    CDATA #REQUIRED
   class   CDATA #REQUIRED
   icon    CDATA #IMPLIED
>
<!ELEMENT description (#PCDATA)>
```

- **id** - a unique name that will be used to identify this w
- **name** - a translatable name that will be used in the d represent this wizard

DTD description for the XML to declare the extension point in your plugin.xml file

Actual XML examples are shown farther down the page!

Use the online help to view more information on how to declare the extension point in your plugin.xml file.

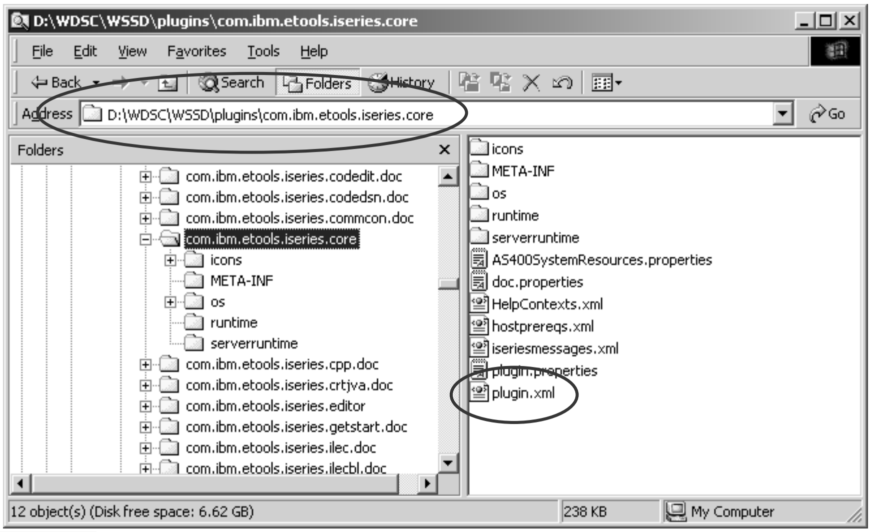


Plug-in Development

- You write plug-ins to extend the Workbench
- Highest level of organization / structure for adding functionality to the Workbench
  - WebFacing plug-in
  - Remote Systems Explorer plug-in
  - Web Tooling plug-in
  - iSeries extensions to Web Tooling plug-in
- A collection of files which implement your desired function
- Plug-in installed by copying your files to a subdirectory of <root>/plugins
  - There are built in tools for installing new plug-ins
- Described by the plugin.xml file
  - Eclipse will look for this file in all subdirectories of <root>/plugins
  - Describes what function(s) your plug-in is adding to the Workbench

Here are the steps to develop plug-in’s.

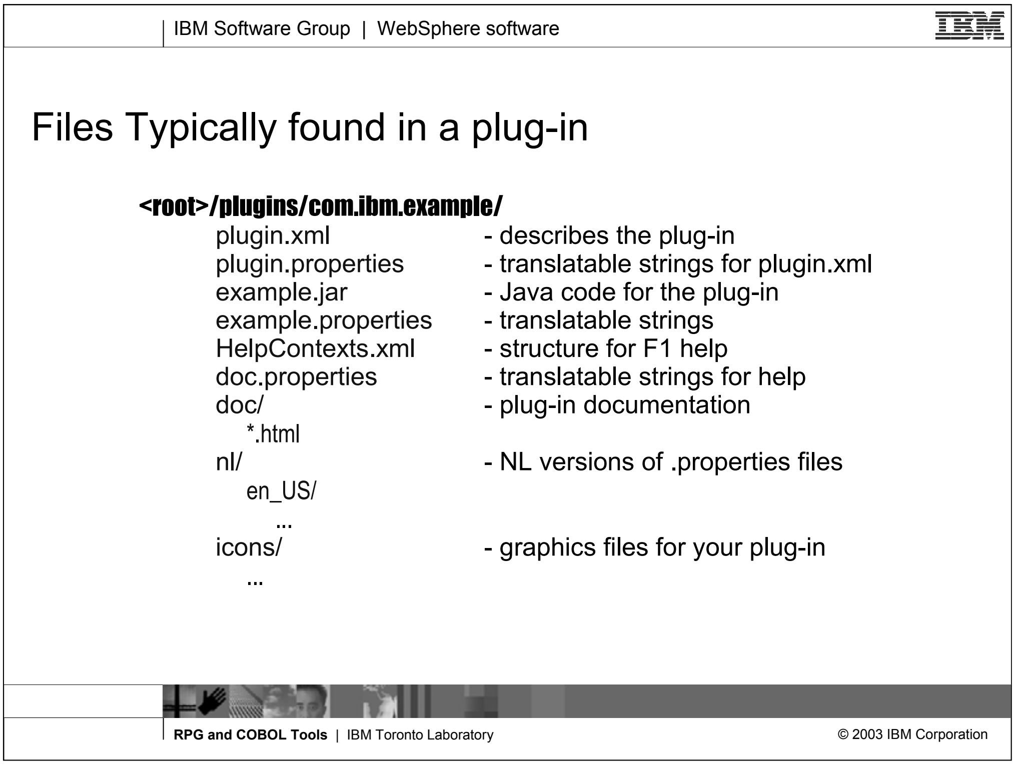
Plug-in Information



D:\WDSC\WSSD\plugins\com.ibm.etools.iseries.core

File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
History
Address
Go
Folders
com.ibm.etools.iseries.codedit.doc
com.ibm.etools.iseries.codedsn.doc
com.ibm.etools.iseries.commcon.doc
com.ibm.etools.iseries.core
icons
META-INF
os
runtime
serverruntime
com.ibm.etools.iseries.cpp.doc
com.ibm.etools.iseries.crtjva.doc
com.ibm.etools.iseries.editor
com.ibm.etools.iseries.getstart.doc
com.ibm.etools.iseries.ilec.doc
com.ibm.etools.iseries.ilecbl.doc
icons
META-INF
os
runtime
serverruntime
AS400SystemResources.properties
doc.properties
HelpContexts.xml
hostprereqs.xml
iseriesmessages.xml
plugin.properties
plugin.xml
12 object(s) (Disk free space: 6.62 GB)
238 KB
My Computer

Here is the location of the plug-in information.

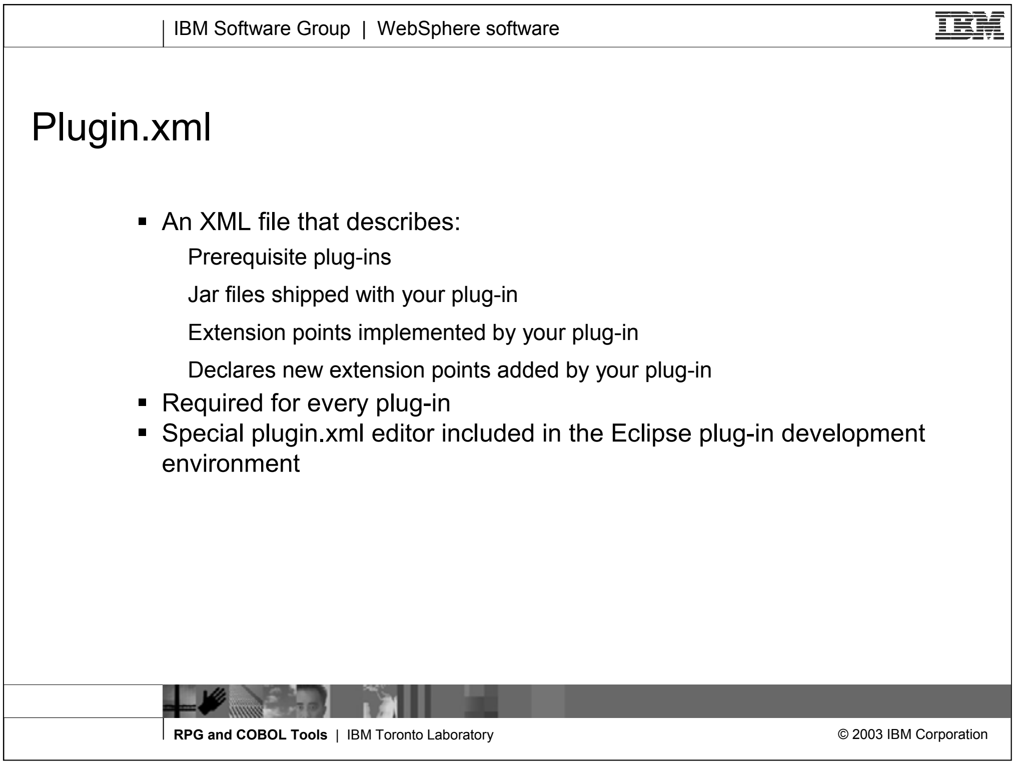


Files Typically found in a plug-in

<root>/plugins/com.ibm.example/

plugin.xml	- describes the plug-in
plugin.properties	- translatable strings for plugin.xml
example.jar	- Java code for the plug-in
example.properties	- translatable strings
HelpContexts.xml	- structure for F1 help
doc.properties	- translatable strings for help
doc/	- plug-in documentation
*.html	
nl/	- NL versions of .properties files
en_US/	
...	
icons/	- graphics files for your plug-in
...	

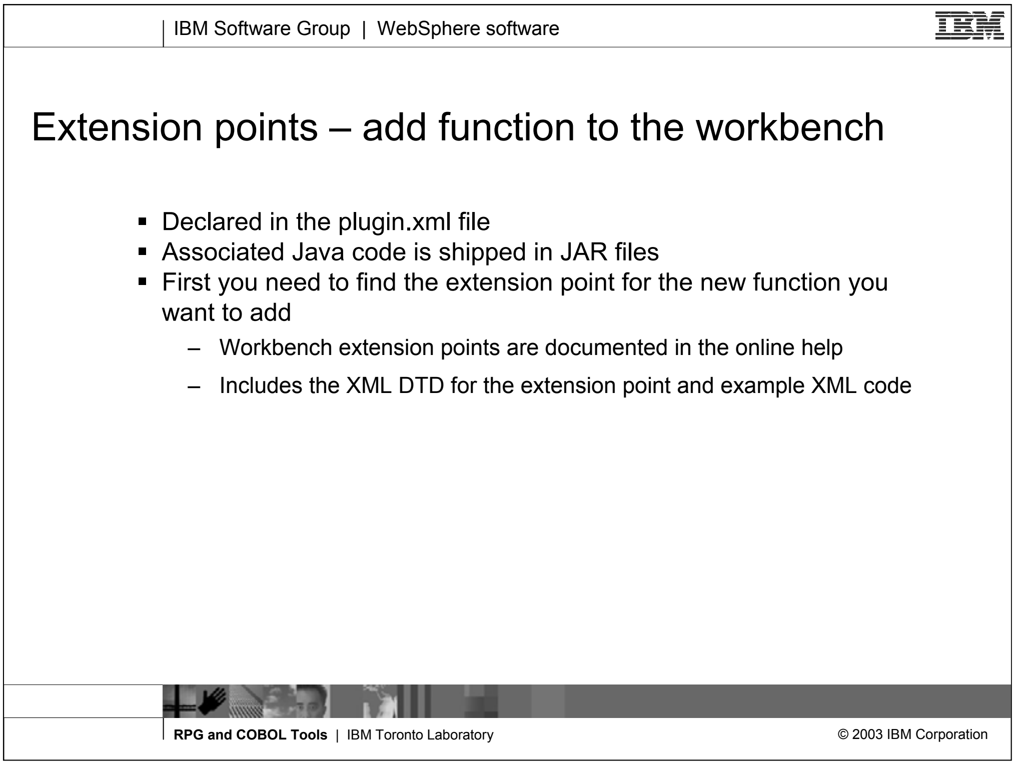
These are the files you typically find in the plug-in.



Plugin.xml

- An XML file that describes:
  - Prerequisite plug-ins
  - Jar files shipped with your plug-in
  - Extension points implemented by your plug-in
  - Declares new extension points added by your plug-in
- Required for every plug-in
- Special plugin.xml editor included in the Eclipse plug-in development environment

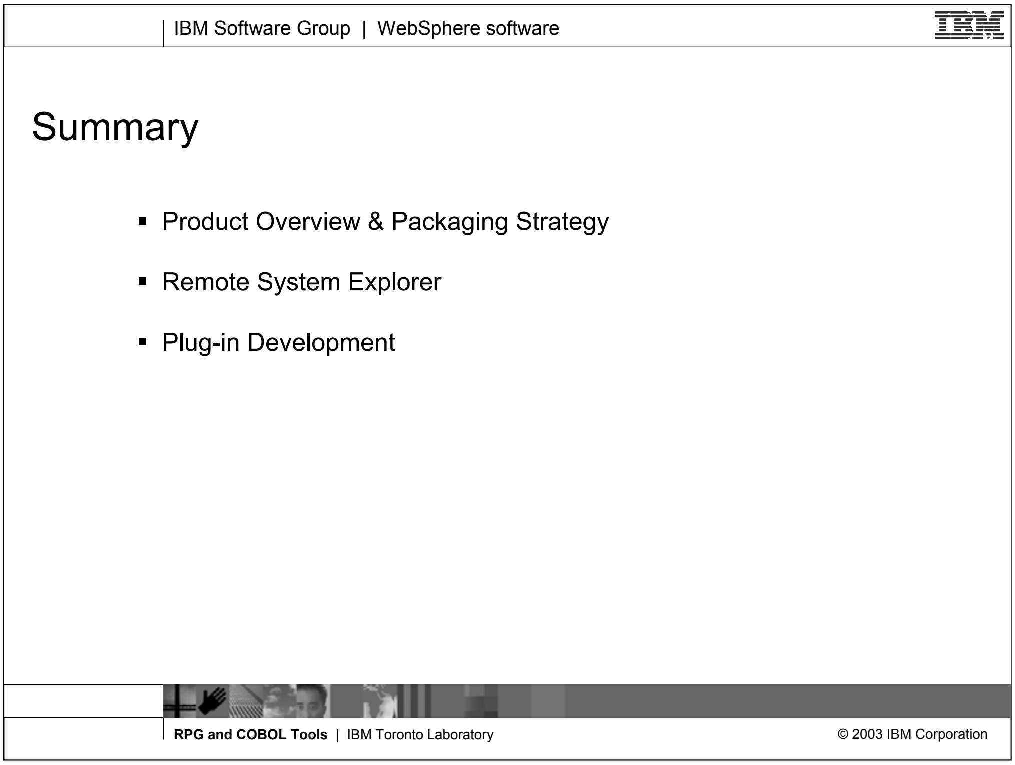
The plugin.xml file describes the extension points added by your plug-in. You need this file for your plug-in.



Extension points – add function to the workbench

- Declared in the plugin.xml file
- Associated Java code is shipped in JAR files
- First you need to find the extension point for the new function you want to add
  - Workbench extension points are documented in the online help
  - Includes the XML DTD for the extension point and example XML code

Extension points add function to the workbench and allow you to integrate your tools into the RSE. Check out the online help for information on the extension you want to use.



Summary

- Product Overview & Packaging Strategy
- Remote System Explorer
- Plug-in Development

We hope this presentation helped you understand more about Development Studio Client iSeries application development tools. We started with an overview of our strategy for iSeries application development tools, went onto review the Remote System Explorer, the perspective for iSeries programmers to maintain and develop iSeries applications and explained how to extend RSE to include your own iSeries application development tools.

Legal information

Acknowledgement:

- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
  - Phil Coulthard, George Farr, Claus Weiss, Don Yantzi, and David Slater

Disclaimer:

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

Reproduction:

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

Thank you for coming.
Enjoy WebSphere Development Studio Client !