



IBM Software Group

Remote System Explorer in IBM WebSphere Development Studio Client for iSeries: **Advanced**

WebSphere software iSeries AD Team. IBM Toronto Lab
www.ibm.com/software/awdtools/wds400



1H 2004 | WDS V5.1 and WDS V5.R2

© 2004 IBM Corporation

This presentation reviews the Remote System Explorer advanced features. These include User Actions and Compile Commands, Running Interactive Commands in RSE, RSE Filters and Filter pools, Working Disconnected, Team Support and iSeries Projects.

IBM Software Group | WebSphere software

WebSphere Development Studio 5.1
www.ibm.com/software/awdtools/wds400

Unlimited Licenses

iSeries	iSeries	iSeries	iSeries	Web Facing	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	App Server	

V5R1 or V5R2: 5722-WDS
 No-cost Upgrade 5905 (V5R1) or 5913 (V5R2)

+CODE
 +VisualAge RPG

WebSphere Development Studio Client V5.1

2 | Remote System Explorer | Advanced | © 2004 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.

Table of contents

User Actions and Compile Commands

Running Interactive Commands in RSE

RSE Filters and Filter pools

Working Disconnected

Team Support

iSeries Projects

Agenda

RSE: User Actions

■ User-Defined Actions ... like PDM!

- Right-click on iSeries Objects -> Work With
 - User Actions
 - Create, delete or change user-defined actions
 - Scope them so you only see them when appropriate
 - Named Types
 - Create named types to scope actions against
 - EG. "RPG" might be RPG + RPGLE + SQLRPGLE

■ ... and even beyond PDM!

- Create user actions for libs/objects, members ... and even jobs and IFS folders and files!

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 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. Your actions will then only appear for object or members that match one of the types in the collection. These actions can be created for objects, members, jobs and IFS folders and files.

Defining User Actions

Define your own actions for object and member popup menus

Work With User Actions

Parent profile: PhilCoulthard

Action name: Add to Library List

Comment: Add selected library to lib list

Command: Normal command

ADDLIBLE &N

PDM-like subst'n vars

Insert variable... Edit... Browse...

Prompt... Prompt first Refresh after Show action

Single selection only

Lots of options

Resource types for which this action will appear

Defined Types	Selected Types
ALL	LIB
CMD	
FILE	
FILE_DATA	
FILE_DSPF	
FILE_MBRFS	
FILE_PRTF	

LIB: *LIB:*

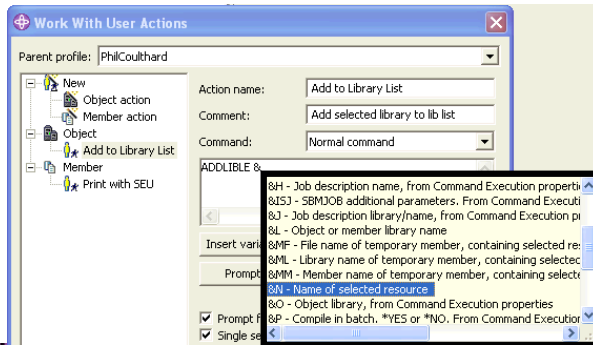
Apply Revert

Close

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 an iSeries command to run when that action is selected. This command can use substitution variables that are identical to those in PDM. There are also numerous options you can specify to affect the action, and you can scope it precisely: for example, object actions can be scoped one or more type and attribute pairs, and member actions by one or more simple or generic member types.

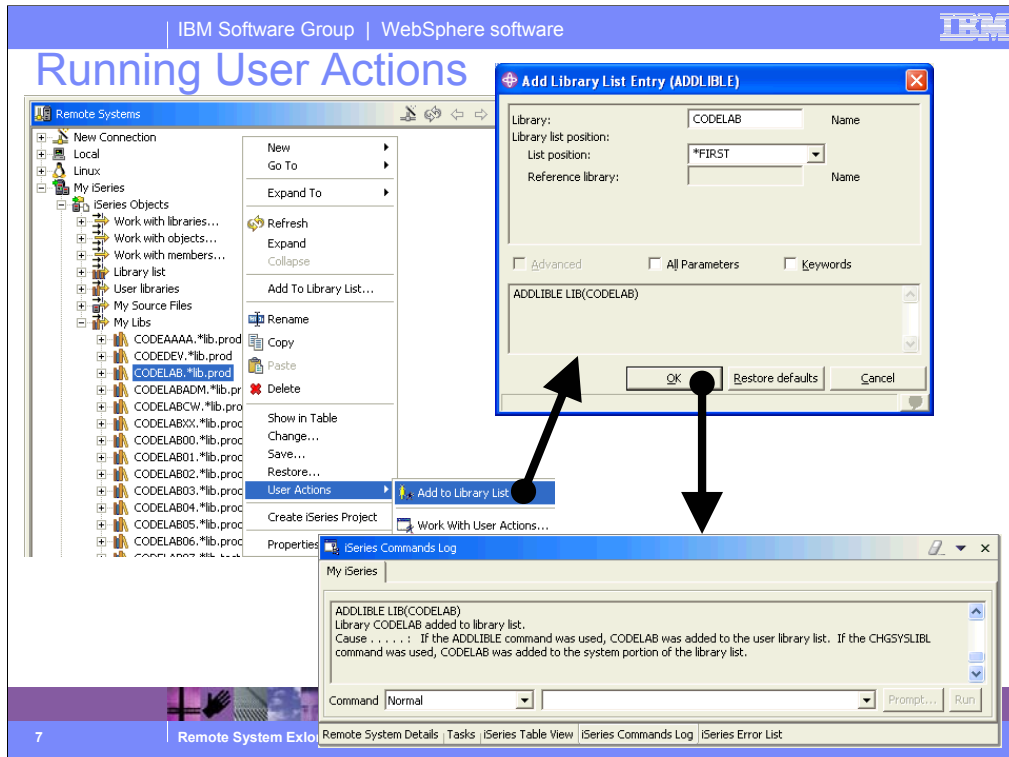
Substitution Variables

- Many substitution variables
 - Superset of PDM
- Different sets for objects vs members vs jobs vs IFS
- Use Insert Variable to see list
 - or Ctrl+Space to see list: or type '&' to get prompted



All PDM substitution variables are supported for the command, and some new ones that PDM didn't support.

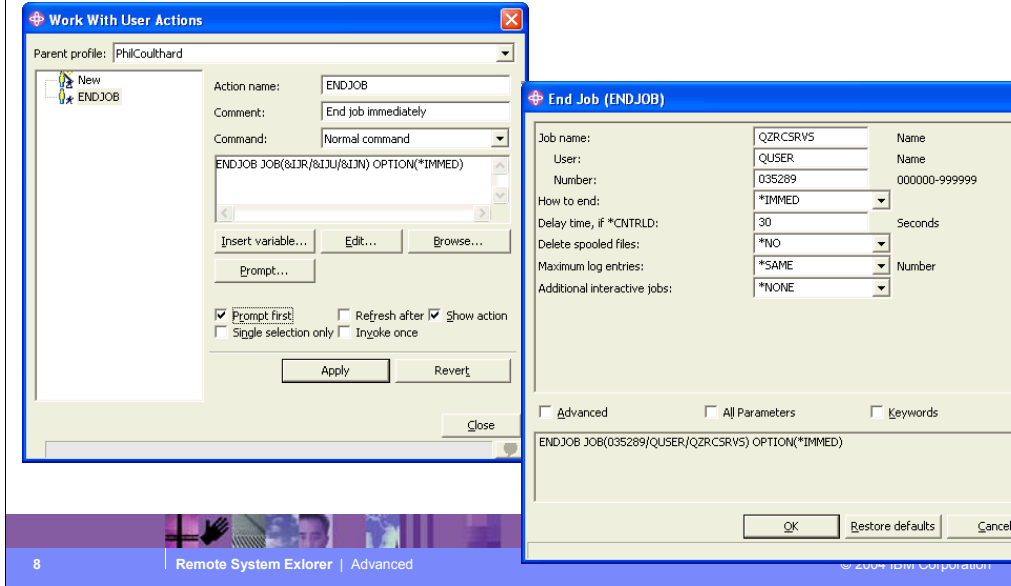
When you type &, you see a popup selection list, or you can type Ctrl+Space or press the Insert Variables button. From the list, double click to insert the selected variable, at the cursor position.



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 (via the Prompt option), you will see the GUI prompt for the command. When the command has finished running, it results are logged in the Commands view.

User Actions for Jobs

- Yes, you can create user actions for Jobs too!



You can also create user actions for Jobs, which will appear in the User Actions popup menu for jobs in the iSeries Jobs subsystem under a connection. The substitution variables include variables for the selected job's number, user and name.

User Actions for IFS

- And even IFS folder and files

You can choose between QShell commands or QSYS commands!

The substitution variables change, depending on command type

For QSYS cmds

For QShell cmds

Similar for Local, Unix, Windows and Linux user actions

Parent profile: PhilCoulthard

Action name: javadoc

Comment:

Command: javadoc \${resource_name}

Command type: QShell command

Insert variable... Edit...

Prompt first Refresh after Show action
 Single selection only Invoke once

Resource types for which this action will appear

Defined Types	Selected Types
ALL	JAVA
C	
C_COMPILEABL	
CPP	
CPP_COMPILEA	
CLASS	

JAVA: java

Apply Revert

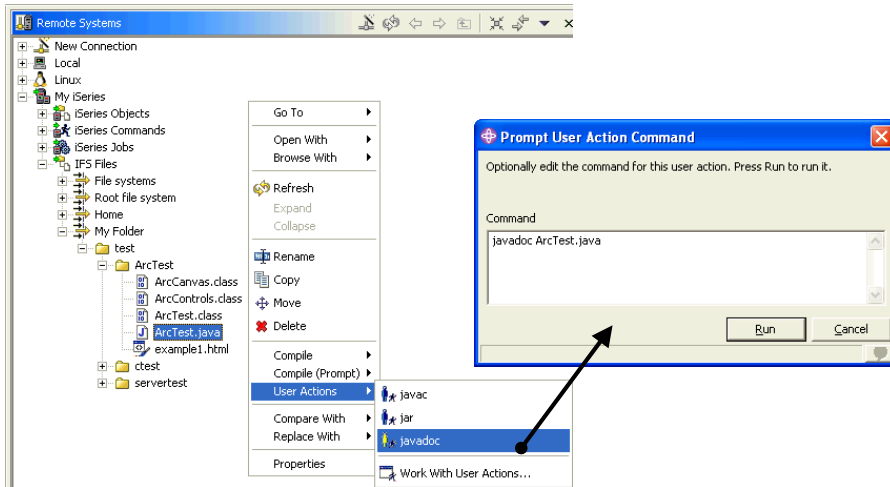
© 2004 IBM Corporation

You can also create user actions for IFS folders and files, which appear in the User Actions popup menu for folders and files in the IFS Files subsystem under a connection.

You can create user actions that are QShell commands or *CMD objects as usual. The substitution variables change depending on the type of command: for qsys commands you get the PDM substitution variables, slightly different to deal with folder/file names versus object/member names. For qshell commands, you get a different list using eclipse-style `${...}` variables, with many variables for all aspects of the selected folder/file name, but a number of other environment variables.

Note the qshell-support is the same as when creating local, unix, windows or linux user actions, which are also supported.

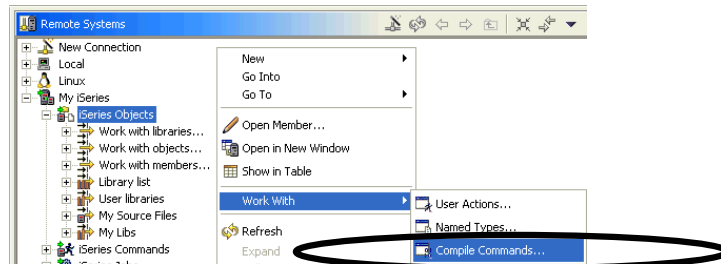
Running User Actions for IFS



IFS user actions show up in the right click of the IFS files or folders, as shown here. If the action is defined to prompt, then you get a prompt dialog where you can edit the command before submitting it.

Work With Compile Commands

- You can change IBM or vendor supplied compile commands, and add your own



In addition to user actions, there is specific support for creating compile commands too. Use the Work With -> Compile Commands option from the iSeries Objects subsystem under an iSeries connection. IBM pre-supplies compile commands, and enables vendors to pre-supply commands as well, some of which do.

Customizing Compile Commands

Create new or edit existing

Commands per member type

Brings up larger window for editing

Right click for more actions

Browse iSeries for *CMD object

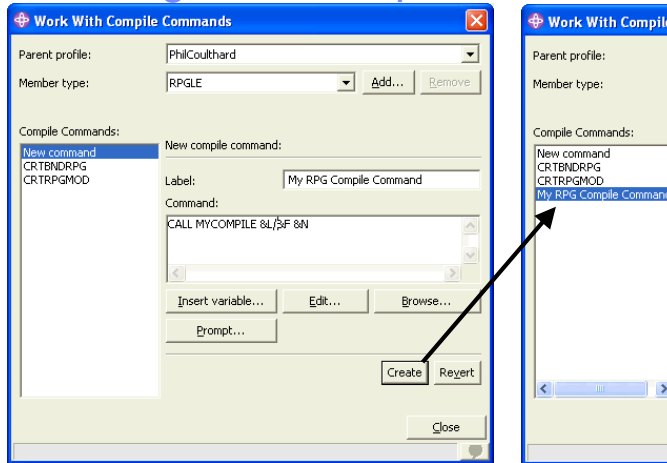
Subset of PDM substitution variables allowed

&F - Name of file containing selected member
 &L - Object or member library name
 &N - Name of selected resource
 &O - Object library, from Command Execution properties
 &R - Replace object when compiling, *YES or *NO; From Command
 &X - Object or member text, in single quotes

12 Remote System Explorer | Advanced © 2004 IBM Corporation

In the Work With Compile Commands dialog, you can create new compile commands and edit existing ones. To edit an existing one, first find it by selecting the member type it applies to (or add a new member type if necessary) at the top of the dialog, and selecting the command in the list of commands on the left. Then edit the command and press Apply. You can also right-click on a command to delete it, copy/paste it, or re-order it. Note you cannot delete IBM-supplied commands, but after editing them, you can restore them to their shipped value.

Creating New Compile Command



To create a new compile command, select “New command” on the left, and then enter the label to show, and command to run, and press Create. The command will show up on the list on the left, and will also now be available at runtime, as we see on the next slide...

Running New Compile Command

The screenshot shows the 'Library list' pane with a context menu open over the member 'PAYROLL.rpgle'. The menu includes options like 'Go To', 'Open With', 'Browse With', 'Rename', 'Copy', 'Move', 'Delete', 'Verify', 'Verify (Prompt)', 'Compile', 'Compile (Prompt)', 'User Actions', 'Add To iSeries Project...', and 'Make Available Offline'. The 'Compile' option is selected, and a sub-menu is visible with 'My RPG Compile Command' highlighted. A yellow callout box with arrows pointing to the 'My RPG Compile Command' option and the 'iSeries Error List' window contains the text: 'You will see the Error List view if the command generates an event file'.

The 'iSeries Error List - CODELAB/EVEVENT(PAYROLLG)' window displays the following table of errors:

ID	Message	Severity	Line	Location	Connection
RNS9308	Compilation stopped. Severity 40 errors found in program.	50	0	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF2120	External descriptions for file MSTDSP not found; file is ignored.	40	27	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF2120	External descriptions for file PRJMST not found; file is ignored.	40	29	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF2120	External descriptions for file RSNMST not found; file is ignored.	40	30	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF7030	The name or indicator ACODE is not defined.	30	103	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF7030	The name or indicator EMPAPL is not defined.	30	54	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF7030	The name or indicator PRCE is not defined.	30	257	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries
RNF7030	The name or indicator PRCMP is not defined.	30	264	CODELAB/QRPGLESRC(PAYROLLG)	My iSeries

Remote System Details | Tasks | iSeries Table View | iSeries Commands Log | iSeries Error List | Remote Commands

14 | Remote System Explorer | Advanced | © 2004 IBM Corporation

Here we see our new compile command appearing when we right click on a member of the type we created the command for. Note the checkmark appears beside the last-used compile command for the selected member's type. When the compile command runs, if it generates an event file that event file is automatically retrieved and the errors produced by the compile are displayed in the Error List window, where you can double click to open the editor and position it at the error.

Defining IFS Compile Commands

- Yes, you can even create compile commands for IFS files ... and even local, Unix, Windows and Linux!

Work With Compile Commands

Parent profile: PhilCoulthard

Source type: java

Compile Commands:

New command: Selected compile command:

Label: javac

Command: QShell command

javac -deprecation -classpath . \${resource_name}

Insert variable... Edit...

Apply Revert

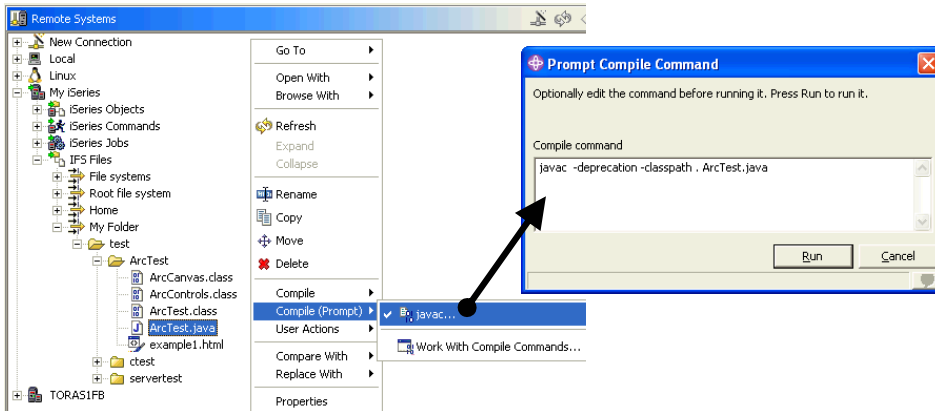
Close

QShell command
OS/400 command
QShell command

&FCN - Name of parent folder containing selected file or folder
&FCP - Fully qualified path of parent folder, including name of selected file
&FNE - Extension part of the name of the selected file
&FNFR - Name of selected file without the extension
&FP - Fully qualified path of selected file or folder, including name
&N - Name of selected file or folder
&O - Object library, from Command Execution properties
&R - Replace object when compiling. *YES or *NO. From Command Execution properties

\$(container_name) - Name of folder containing selected resource, unqualified
\$(container_path) - Path of folder containing selected resource, including name
\$(resource_name_root) - Name of selected resource without the extension
\$(resource_name) - Name of selected resource, unqualified
\$(resource_path_drive) - Drive letter on Windows, empty string on others
\$(resource_path_root) - Root of selected file's path. "c:" on Windows, "/" on others
\$(resource_path) - Path of selected resource, including name
\$(system_filesep) - File separator. "/" on Windows, "/" on Unix and Linux
\$(system_homedir) - Home directory on the target system, for the connection's user ID
\$(system_pathsep) - Path separator. ";" on Windows, ":" on Unix and Linux
\$(system_tmpdir) - Temporary directory on the target system

Running IFS Compile Commands



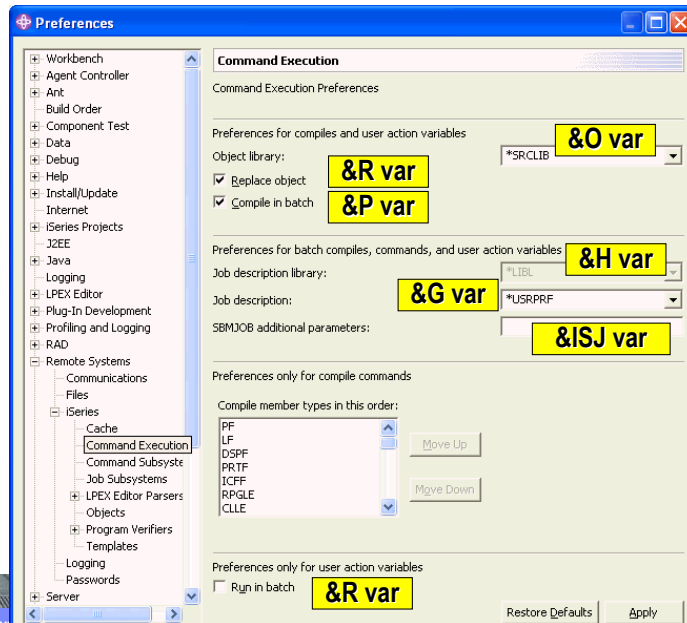
Here we see our new compile command appearing when we right click on an IFS file of the type we created the command for. Note the checkmark appears beside the last-used compile command for the selected file's type. When an OS/400 compile command runs, if it generates an event file that event file is automatically retrieved and the errors produced by the compile are displayed in the Error List window, where you can double click to open the editor and position it at the error. When a QShell compile command runs, the errors are shown in the Command view, and are double clickable.

Preferences For User Actions and Compile Cmds

Window->
Preferences

Specify
preferences
here, for all
connections

Override per
connection via
Properties
dialog for
iSeries Objects



There is a preferences dialog that has many preferences which affect substitution variables for user actions and compile commands.

Table of contents

User Actions and Compile Commands

➤ **Running Interactive Commands in RSE**

RSE Filters and Filter pools

Working Disconnected

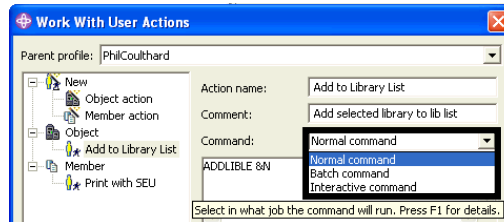
Team Support

iSeries Projects

Agenda

Running Interactive Commands in RSE

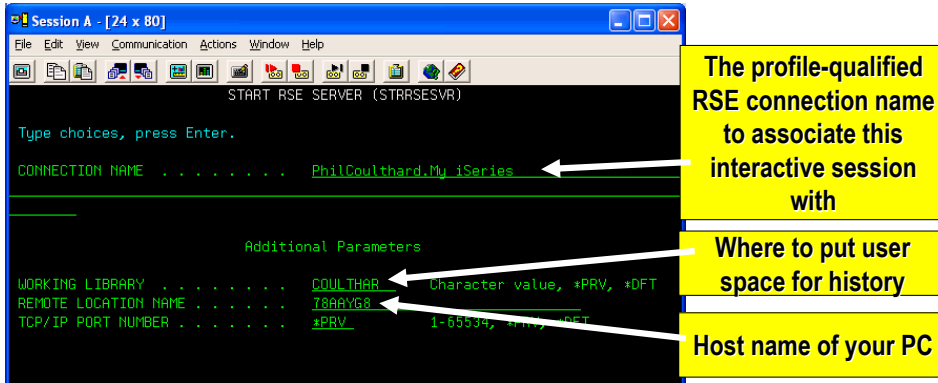
- There are three types of iSeries commands in RSE
 - Normal -> runs in RSE batch job (with adopted user ID)
 - Batch -> runs via SBMJOB (using preferences for JobD, etc)
 - Interactive -> runs in interactive job (you must do STRRSESVR)
- You will see this prompt in many places. Eg:
 - When defining User Actions
 - When defining Compile Commands
 - When running commands in iSeries Command log view



Within the Remote System Explorer, there are 3 ways to run commands on the iSeries: normal, batch or interactive, as described here. You will see the prompt for the three types of commands throughout the RSE, so its important to know the differences between them.

Running Interactive Commands in RSE

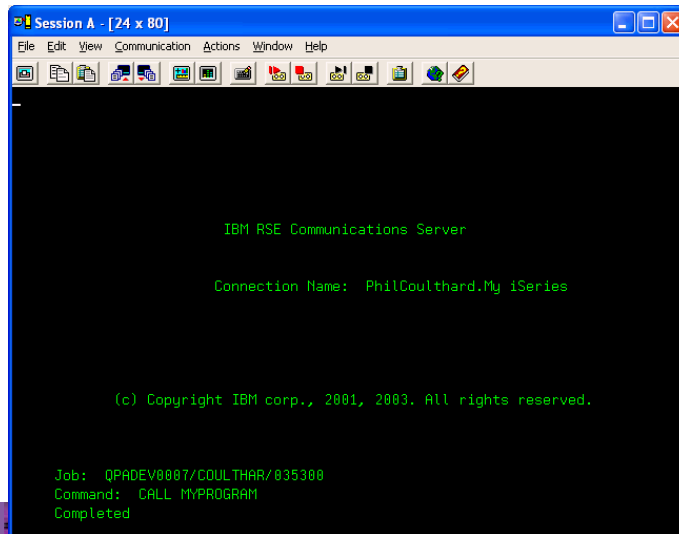
- Step 1: Run STRRSESVR in any emulator



To run interactive commands, you must run the STRRSESVR command from within an emulator. Specify the name of the connection within the RSE to associate this interactive session with ... qualified by profile. It may help to select the "Qualify connection names" option in the dropdown of the title bar of the Remote Systems view in the RSE, so you can see what profile your connection is in. Also specify a library unique to you, to hold *PRV values for this command, and of course your PC's hostname or IP address.

Running Interactive Commands in RSE

- Step 1: Emulator is now reserved...



The screenshot shows a window titled "Session A - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main area is a black terminal window with green text. The text displayed is:

```
IBM RSE Communications Server

Connection Name: PhilCoulthard.My iSeries

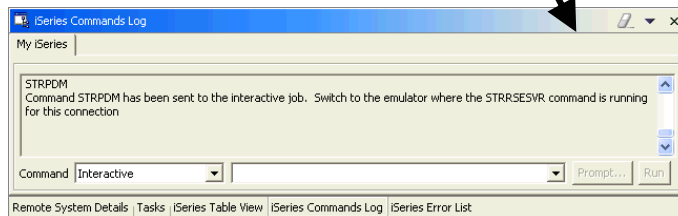
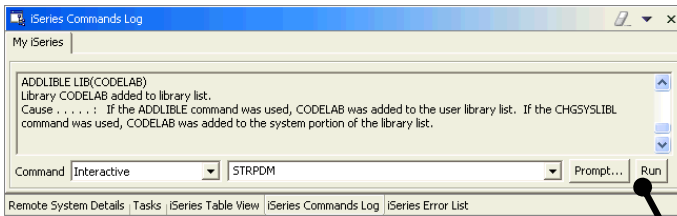
(c) Copyright IBM corp., 2001, 2003. All rights reserved.

Job: QPADEV0007/COULTHAR/035300
Command: CALL MYPROGRAM
Completed
```

When the STRRSESVR command is successful, it will reserve the emulator by putting up the screen shown here.

Running Interactive Commands in RSE

- Step 2: Run interactive command in RSE



Now, within the connection associated with the STRRSESVR command, you can run interactive commands from within the RSE, and debug interactive programs. The example here shows running STRPDM from the iSeries command log view. You need to give focus to your reserved emulator to see the results...

Running Interactive Commands in RSE

- Step 2: ... and your command runs in emulator:

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
AS/400 Programming Development Manager (PDM)

Select one of the following:

 1. Work with libraries
 2. Work with objects
 3. Work with members
 4. Work with projects
 5. Work with groups
 6. Work with parts

 9. Work with user-defined options

Selection or command
==> _

F3=Exit    F4=Prompt  F9=Retrieve  F10=Command entry
F12=Cancel F18=Change defaults

(C) COPYRIGHT IBM CORP. 1981. 2000.
```

... and here we see that the reserved emulator is now running the interactive command we entered in the RSE.

Table of contents

- User Actions and Compile Commands
- Running Interactive Commands in RSE
- **RSE Filters and Filter pools**
- Working Disconnected
- Team Support
- iSeries Projects

Agenda

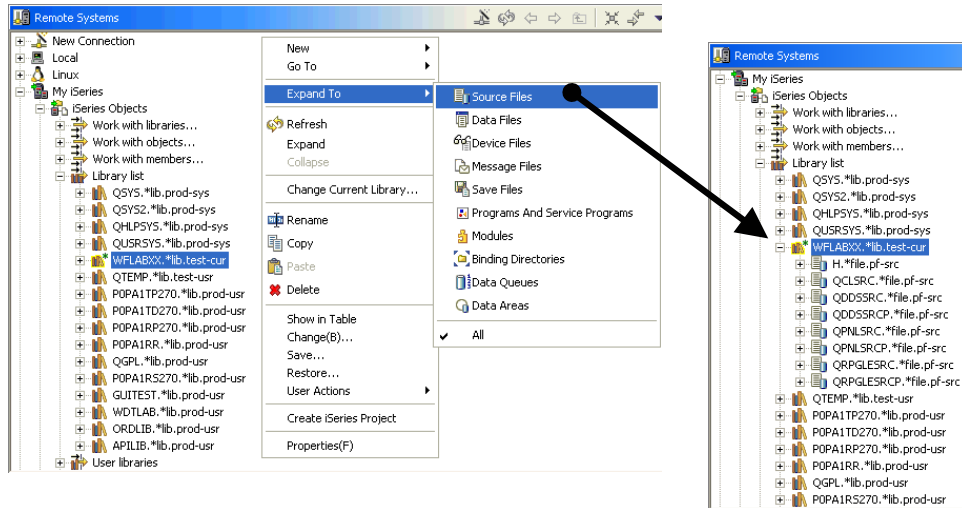
Drilling Down

- Typically we start using the RSE by just expanding:
 - Expand Library list to see libraries on lib list
 - Expand a library to see all objects in it
 - Expand a source file to see members in it
 - Expand Home directory to see folders in \home in IFS
 - Expand a folder to see all folders and files in it
 - And so on

- But sometimes this produces lists that are too big
 - Its unwieldy to scroll through thousands of things
 - You really want to keep lists small, to a few hundred at most

The Remote System Explorer is all about random access to remote things ... like stuff in QSYS, or jobs, or IFS files. While it is designed to be immediately useful by merely expanding nodes to drill down to what we want, sometimes that can produce unsatisfactory results if this results in vast numbers of items.

*LIB Expand To → For Quick Filtering



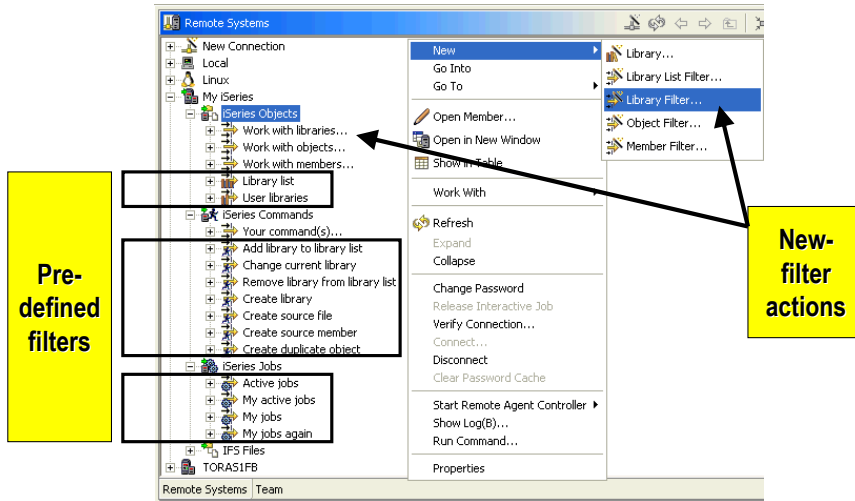
One very quick way to reduce the amount of things is to use the Expand To object for libraries. It allows you to expand a library to see only objects of a particular type. This subsetting remains in effect forever, even when you re-expand with the plus sign, until you subsequently choose All or any other expand-to criteria.

Introducing Filters

- **Eventually you will need to see a subsetted list**
 - Using criteria like generic names, types and attributes
 - All subsystems (nodes under a connection) support “**filters**”
 - These allow fine-grained control over what is shown in the RSE.
- **To create a filter, right click on any subsystem and select New -> xxx**
 - iSeries Objects -> New -> Library Filter, Object Filter, Member Filter
 - iSeries Commands -> New -> Command Set
 - iSeries Jobs -> New -> Job Filter
 - IFS Files -> New -> Filter
- **Or use the fastpath Work With XXX... prompts under iSeries Objects**

Eventually, you will find the need to see a subsetted list. That is what filters offer, and the RSE has extensive filter support.

Introducing Filters



RSE filters for iSeries Objects

- **Library filters**
 - Specify simple, generic or special library names
- **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

Remote Systems

New Connection

Local

Linux

My iSeries

iSeries Objects

Work with libraries...

Work with objects...

Work with members...

Library list

User libraries

iSeries Commands

iSeries Jobs

IFS Files

TORAS1FB

Expand

Specify simple, special, or generic library name

Library: *LAB*

Library Filter

Create a new iSeries library filter

Library: *LAB*

Browse...

< Back

Next >

Library 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 Libraries

Specify a filter name

30 Remote System Explorer | Advanced © 2004 IBM Corporation

Here we use the Work With Libraries... prompt to create a library filter. Specify the library name, simple or generic, to list when this filter is expanded. Note the generic names in RSE support two asterisks, not just one as is the case with PDM. On the second page, specify the name for this filter.

Expanding Library Filters

The screenshot shows the Remote System Explorer interface. The left pane displays a tree view under 'My iSeries Objects' with 'My Libraries' expanded. A filter is selected, and its contents are listed below it. A context menu is open over the filter, showing options like 'Change...', 'Rename', 'Copy', 'Move', 'Delete', 'Move Up', and 'Move Down'. Three yellow callout boxes provide additional information:

- Filter is created, and expanded. It lists all libraries matching the given criteria.**
- Filter exists until you delete it**
- Filter exists in all connections by default**

The context menu also includes options like 'New', 'Go Info', 'Go To', 'Open in New Window', 'Show in Table', 'Refresh', 'Expand', 'Collapse', 'Rename', 'Copy', 'Move', 'Delete', 'Move Up', 'Move Down', 'Show in Table', and 'Properties'.

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

The screenshot shows the 'New Object Filter' wizard in the Remote System Explorer. The first page, 'Create a new iSeries object filter', has the following fields:

- Library:** ADTSLAB (with a 'Browse...' button)
- Object:** (with a 'Browse...' button)
- Object type:** *FILE (with a 'Browse...' button)
- Object attribute:** * (with a 'Browse...' button)
- More Types >>** (button)

The second page, 'Name the new filter', has the following field:

- Filter name:** My Objects

Yellow callout boxes with arrows point to the following fields:

- Specify simple, special, or generic library name** (points to Library)
- Specify simple or generic object name** (points to Object)
- Specify object type** (points to Object type)
- Specify simple or generic object attribute** (points to Object attribute)
- More types (next page)** (points to More Types >>)
- Expand** (points to the 'Work with objects...' prompt in the left tree view)

The bottom of the screenshot shows the page number 32 and the text 'Remote System Explorer | Advanced'.

Here you use the Work With Objects... prompt to create an object filter to list objects in one or more libraries. Specify the library, the object, the type and attribute filtering criteria. All but the object type can be generic.

On the 2nd page of the wizard, you specify the name for this filter.

Filtering objects by type + attribute

When More Types>>> pressed

Prompt for object type

Prompt for object attribute

Specify multiple object type + attribute pairs

Will list all objects with any of the specified types+attributes

Select Object Type

Select an object type:

- *ALRTBL
- *AUTL
- *BNDDIR
- *CFGL
- *CHTFMT
- *CLD
- *CLS
- *CMD
- *CNIL
- *COSD
- *CSI
- *CSMAP
- *CSTBL
- *CTLD
- *CRG
- *CROD
- *DEVD
- *DOC
- *DTAARA
- *DTAQ
- *EDIT
- *EXTRG
- *FCT
- *FTI F

Select Object Attribute

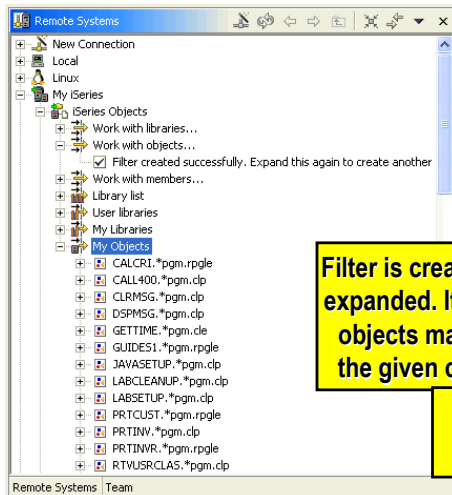
Select an object attribute:

- ASM38
- BA5
- BAS38
- CBL
- CBLE
- CBL36
- CBL38
- CLE
- CLL
- CLP
- CLP38
- CPPL
- CSP
- DFU
- DFUEXC
- DFUNOTEXC
- FTN
- PAS
- PAS38
- PLI
- PLI38
- RMC
- RPG
- RPG F

33 Remote System Explorer | Advanced © 2004 IBM Corporation

By using More Types>>>, you can specify more than one object type and attribute combination. Note the attribute can be generic. This multiple-type support is beyond what even PDM supports, and allows for extremely specific lists based on filters.

Expanding Object Filters



Use Change to edit filter later

Filter is created, and expanded. It lists all objects matching the given criteria.

Filter exists until you delete it

Filter exists in all connections by default

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 Systems

New Connection

Local

Linux

My iSeries

iSeries Objects

Work with libraries...

Work with objects...

Work with members...

Library list

User libraries

My Libraries

My Objects

iSeries Commands

iSeries Jobs

IFS Files

TORAS1FB

Expand

Specify simple, special, or generic library name

Specify simple or generic file name

Specify simple or generic member name

Specify simple or generic member type

More types >> (next page)

New

Member Filter

Create a new iSeries member filter

Library: ADTSLAB Browse

File: QRPGLESRC Browse...

Member: * Browse...

Member type: RPG* Browse...

More Types >>

Object members Data members

< Back Next >

New

Member 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 Members

35 Remote System Explorer | Advanced © 2004 IBM Corporation

Here you use the Work With Members... prompt to create an object filter to list objects in one or more libraries. Specify the library, the object, the type and attribute filtering criteria. All but the object type can be generic.

On the 2nd page of the wizard, you specify the name for this filter.

Filtering by member type

When More Types>>> pressed

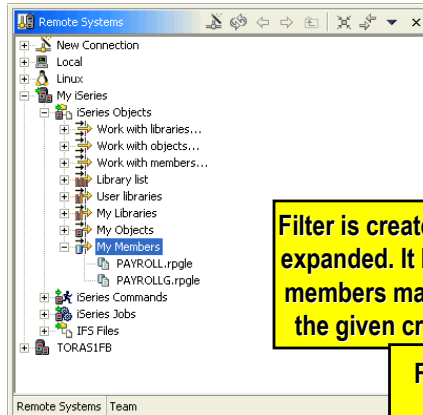
Specify multiple member types. Handy when generic name won't work

Will list all member with any of the specified types

36 Remote System Explorer | Advanced © 2004 IBM Corporation

By using More Types>>>, you can specify more than one member type. Note that each type can be generic. This multiple-type support is beyond what even PDM supports, and allows for extremely specific lists based on filters.

Expanding Member Filters



Filter is created, and expanded. It lists all members matching the given criteria.

Filter exists until you delete it

Filter exists in all connections by default

Use Change to edit filter later

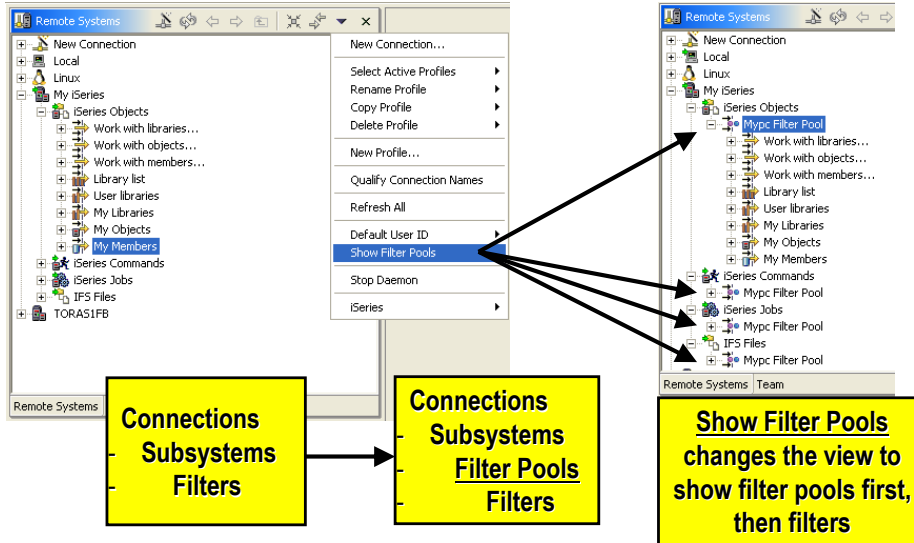
When a member filter is created, it shows up in the list of existing filters for iSeries Objects. You can edit the filter by right clicking on it and selecting the Change... action from its popup menu. When a member filter is expanded, all members matching one or more of the filter's filter strings are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

Filter Pools

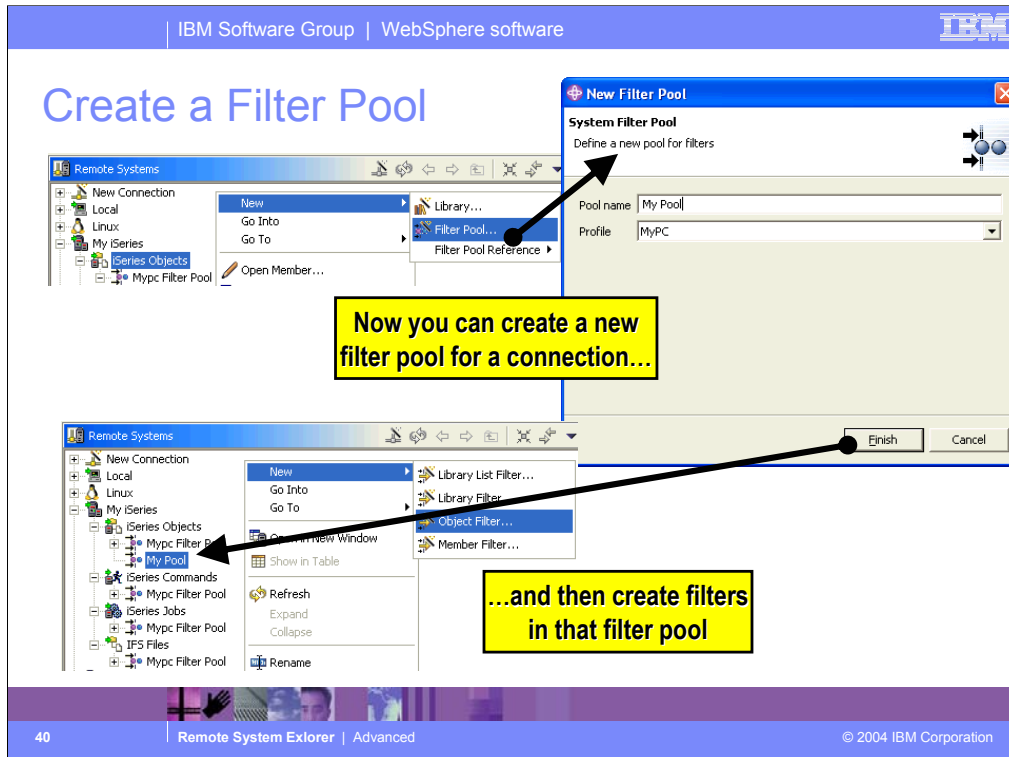
- **Eventually you will have too many filters!**
 - Or you will want a filter not to show in all connections
- **Time to turn on “Show Filter Pools”**
 - Expanding subsystems will then first show filter pools
- **Filters are grouped into named pools**
 - By default they are added to the “default filter pool”
 - You can create your own filter pools
 - Then add filters to it
 - Expand a filter pool to see just the filters in it
 - Connections reference filter pools ... only see those you reference
 - The default pool is always referenced
- **Filter pools group filters**
 - Nothing more than a named collection of filters
 - Allow you to group filters 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.

Show Filter Pools



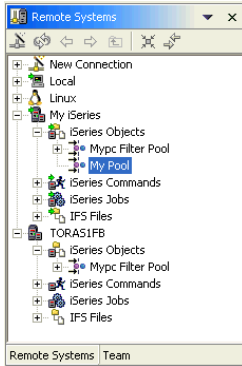
Here we see how to turn on the Show Filter Pools preference. Once enabled, when a subsystem is expanded you see not the filters, but first the filter pools referenced by this subsystem in this connection. When a filter pool is expanded, then you see the filters within that filter pool.



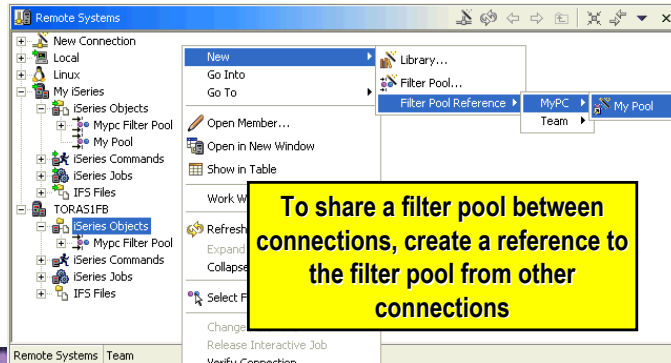
To create a new filter pool, right click on the subsystem, then select New -> Filter Pool...

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!

Reference a Filter Pool



By default, only the connection where you created the filter pool has a reference to it, so filters created in it only appear for that connection



To share a filter pool between connections, create a reference to the filter pool from other connections

By default, filter pools you create are not shared by multiple connections. To affect that, create a reference to the same pool from multiple connections. Note any changes made to the filters in the pool are now immediate across all shared connections.

Work With Filter Pools

In **Show Filter Pools** mode, there are two new actions on subsystems like iSeries Objects:

1. Work With Filter Pools
2. Select Filter Pools

The screenshot displays the Remote System Explorer interface. The left pane shows a tree view of subsystems, including 'MySeries Objects' and 'iSeries Objects'. The right pane shows a context menu with actions like 'Filter Pools...' and 'Select Filter Pools...'. Two callout boxes provide details:

- 1. WW Filter Pools: One stop shopping for filter pool management**: This callout points to the 'Work With Filter Pools' dialog box, which shows a tree view of filter pools (Team, My Filters, Team Filter Pool, MyPC, Mypc Filter Pool, My Pool) and a 'Move selected filter pool to a different profile' button.
- 2. Select Filter Pools: Easily manage which pools this connection references**: This callout points to the 'Select Filter Pools' dialog box, which shows a tree view of filter pools with checkboxes for selection.

When you are in Show Filter Pools mode, you will see a couple actions on subsystems, for managing filter pools and managing references to filter pools.

Table of contents

- User Actions and Compile Commands
- Running Interactive Commands in RSE
- RSE Filters and Filter pools
- **Working Disconnected**
- Team Support
- iSeries Projects

Agenda

Working disconnected

What you *can* do while disconnected

- ✓ Edit local and offline source files / mbrs
- ✓ Syntax check RPG, COBOL, DDS
- ✓ Syntax check CL for cached commands
 - ✓ Prompt RPG and DDS
 - ✓ Prompt CL for cached commands
- ✓ Access language help for RPG, COBOL, DDS
- ✓ Access CL help for cached commands
- ✓ Verify RPG, COBOL, DDS for those members that are cached

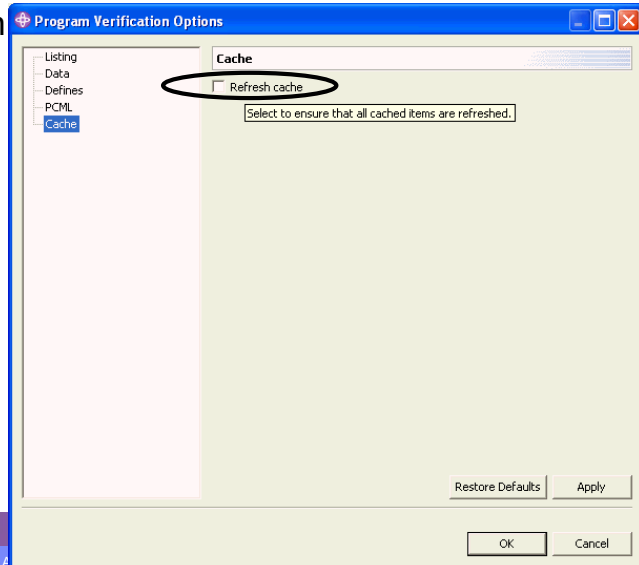
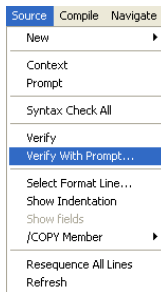
What you *cannot* do while disconnected

- ✓ Edit, design or compile host source members
 - ✓ Run or debug host programs
 - ✓ Run host commands
- ✓ Syntax check, prompt or F1 on CL commands not cached
- ✓ Verify members not previously verified

This summarizes what can and cannot be done when working in disconnected mode ... that is, without a live connection to an iSeries host.

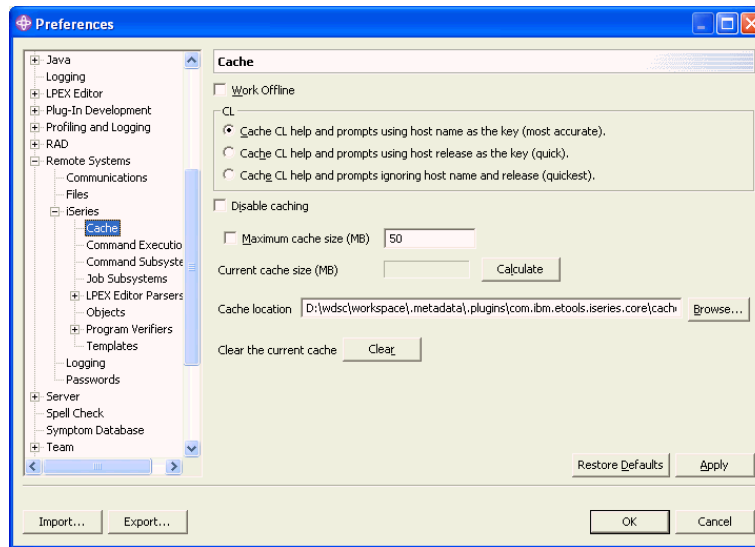
Caching

- Verify source member
- Caches information
- To update cache:
 - Select Verify (Prompt)
 - Select Refresh cache
 - Click OK



When you verify a member, the external references are automatically cached locally. On subsequent verifies, the local cache is used. To force a refresh of the cache, prompt the verify action and select Refresh Cache.

Cache Settings



You can affect cache settings via the Cache preferences page.

Make Available Offline

In order to work offline, you have to make a local copy of your member first. Use Make Available Offline

This creates an iSeries project, and imports the member(s) to it. This is where you work offline

What are iSeries Projects? Stay tuned...

47 Remote System Explorer | Advanced © 2004 IBM Corporation

To prepare for working offline, use the Make Available Offline action on the members you intend to edit. Then, while offline, you edit your source from the iSeries Projects perspective. More on iSeries Projects later.

Table of contents

- User Actions and Compile Commands
- Running Interactive Commands in RSE
- RSE Filters and Filter pools
- Working Disconnected
- **Team Support**
- iSeries Projects

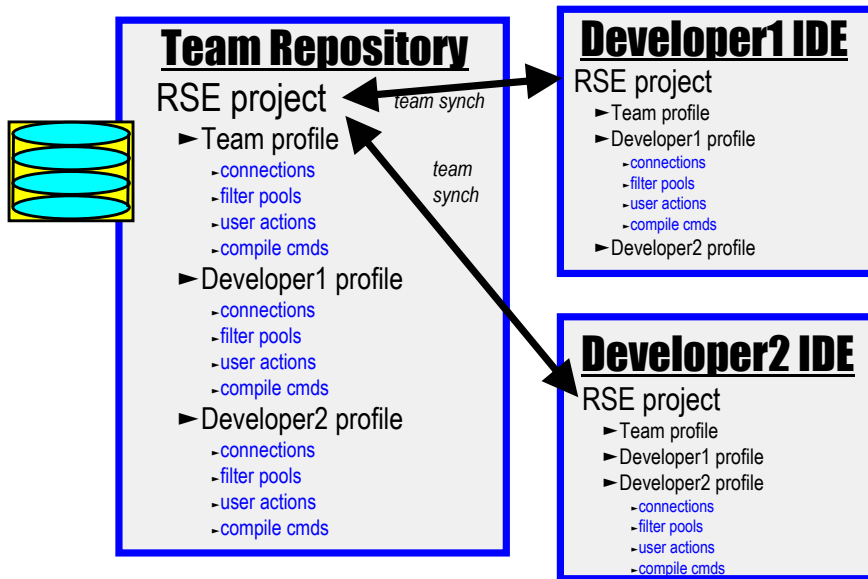
Agenda

Profiles

- The RSE is designed for team sharing
 - Of connections
 - Of filter pools
 - Of user-defined actions
 - Of compile commands
- 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 maintenance 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.

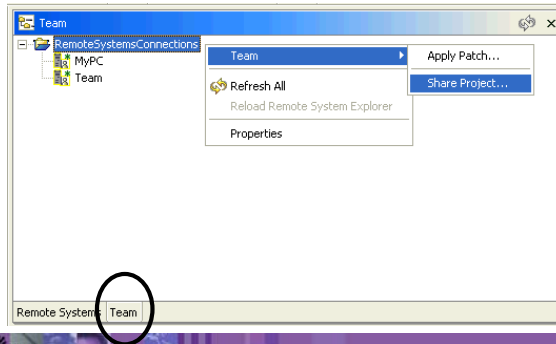
Profiles and Teams



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...

Team View

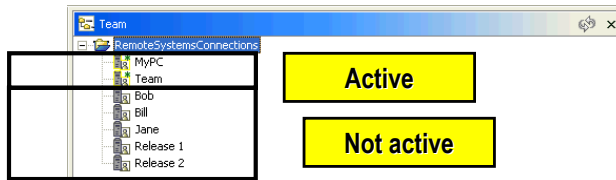
- Use the Team view to share the RSE project
 - You'll need to have an Eclipse-enabled change management repository: EG: CVS, ClearCase, etc
- Once associated with a repository, new Team actions will appear for synchronizing with the repository



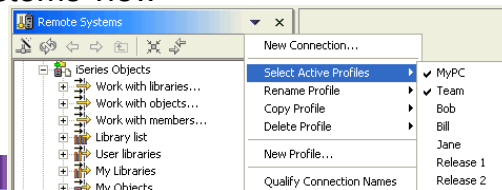
The Team View, which is stacked behind the Remote Systems view by default, is where users go to team-share their RSE artifacts.

Team View

- After synchronizing, you will have all profiles of all team members, but only yours and the Team profile are “active” by default



- To change which profiles are active, use local toolbar of Remote Systems view



After sharing the RemoteSystemsConnections project, and synchronizing it, you will see all the profiles of all team members. By default only yours and the Team profile are active, however.

Active versus Inactive Profiles

- The RSE shows a concatenated list of the following, from all active profiles:
 - Connections
 - Filters (actually filter pools)
 - User actions
 - Compile commands
 - All of these are scoped/owned by profiles.
- To create a shared connection/filter/action/compile-command:
 - Place it in the Team profile, which is the only profile one that is active for everyone, by default.
 - Or place it in a profile that all teams members agree to make active
 - After synchronizing with repository, everyone on team will see it

This discusses how to create shared artifacts in the RSE.

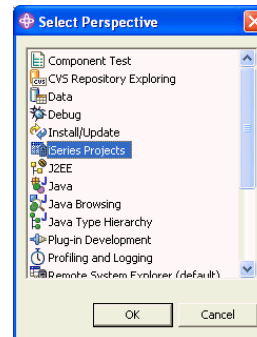
Table of contents

- User Actions and Compile Commands
- Running Interactive Commands in RSE
- RSE Filters and Filter pools
- Working Disconnected
- Team Support
- **iSeries Projects**

Agenda

iSeries Projects

- A special project type
 - For holding source destined to be compiled on iSeries
- A dedicated perspective
 - For working with all and only iSeries projects



There is a special project type designed to hold source that is destined to be compiled and run on an iSeries. There is a wizard for creating a new iSeries Project. There is a special perspective for working with iSeries projects.

What is an iSeries project?

- **Typical Eclipse project**
 - Contains folders and files, can be shared by a team
 - Can have its own tools and perspectives
- **But also**
 - Holds copies of some source members from an iSeries library
 - Those copies are “pushed” up occasionally and the project re-built

An iSeries Project is a typical Eclipse project. This means it contains folders and files that are in the local file system of the IDE, but these can be synchronized with a central repository for team support.

Since the files are local, there is actions to support "pushing" those files up to an iSeries library, where they become source members. This is done prior to compiling or "building" the source for the purpose of testing.

iSeries Project Library, files, members

- **iSeries Project**

Each iSeries project is associated with a single iSeries library
1 to 1 association

Use **File -> New -> iSeries Project** wizard to create

- **iSeries Source Physical File**

Source physical files represented as folders in the project
CCSID, record length, IGC data and description stored as
“metadata” for folder

Use **New -> iSeries Source Physical File** wizard to create

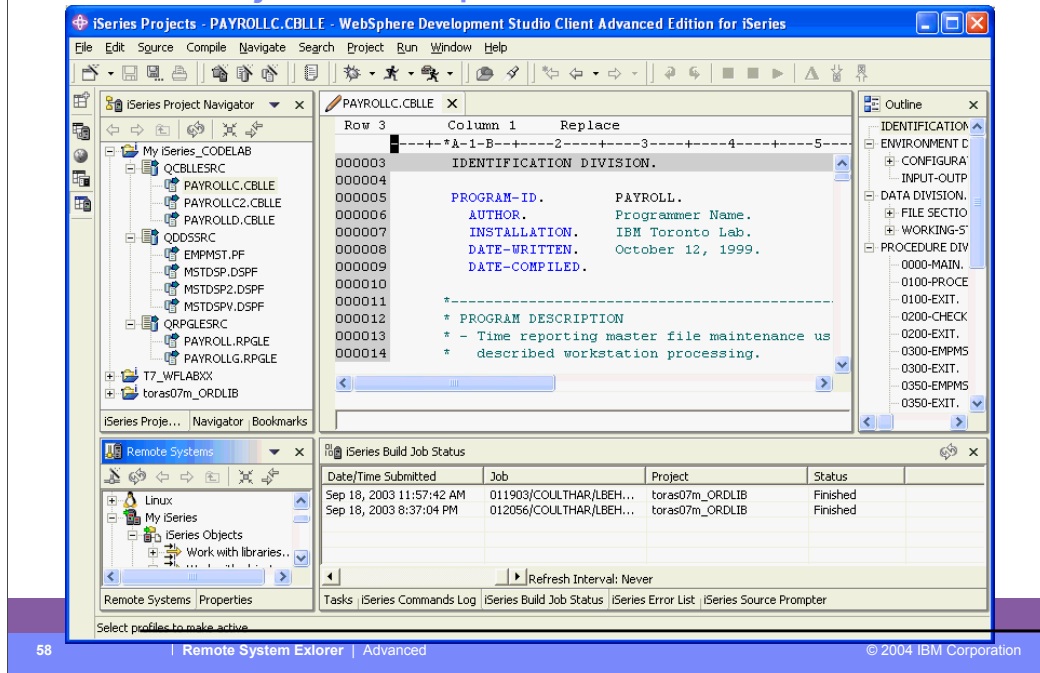
- **iSeries Member**

Members stored as files within “source” folders

Uses format: `memberName.memberType` `ORDRENT. RPGLE`

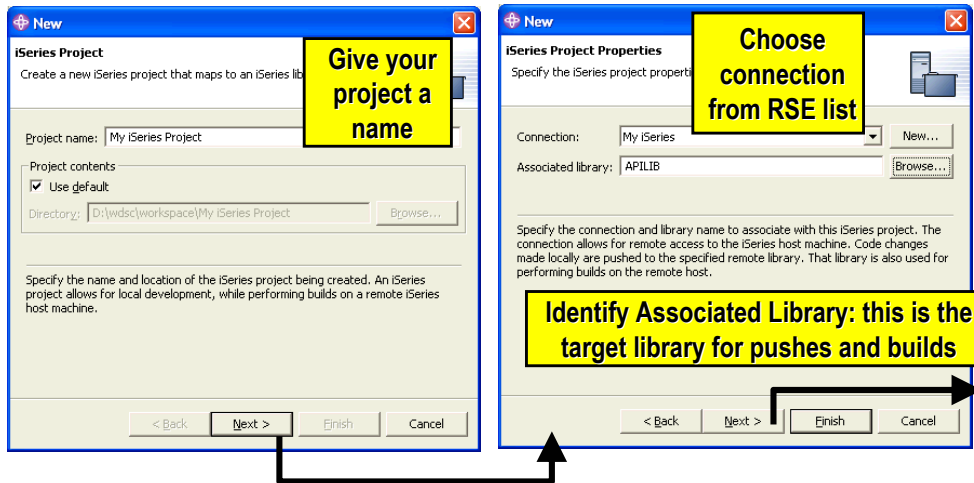
Use **New -> iSeries Source Member** wizard to create

iSeries Projects Perspective



Here we see the iSeries Projects perspective, which you can explicitly open. The primary view is the iSeries Project Navigator that allows exploration of all existing iSeries Projects. There are wizards launchable from the toolbar for creating a new iSeries Project, or creating source physical files and source members within an existing project. The iSeries Projects perspective includes a full copy of the Remote Systems Explorer, as a view, for your convenience. This allows you to easily work with the contents of the associated library while simultaneously working with the local copies of the source within the project.

iSeries Project wizard



Here we see the iSeries Project wizard. First, give the project a name. Any name you want! Then select or create a connection (from the RSE!) that identifies the iSeries with which this project is associated. Also select the library on that iSeries where the contents of this project will be pushed to.

iSeries Project Wizard (next page)

Build style controls how the iSeries project is built on the remote system

iSeries Project Properties
Select and configure the active build style for this project.

Active build style:
CL Program
Command
*NONE

Configure...

This build style will upload CL programs specified in the configuration, compile them, and then run them to build this project. The compile commands can be automatically generated.

Provided by IBM plugin "iSeries Projects - Remote Build Styles Plugin" version 5.1.0

< Back Next > Finish Cancel

CL Program Build Style Configuration
Enter the project's source file that contains the COMPILE.CLLE and BIND.CLLE members that will be uploaded to build this project.

Build Source File:
QCLSRC

Automatically generate COMPILE.CLLE prior to each build
 Automatically push all changed members prior to build

OK Cancel

Specify Build Style and style configuration parameters for iSeries Project more information on next slide ...

Command Build Style Configuration
Enter the command to be submitted to build this project.

Build Command:
|

Automatically push all changed members prior to build

OK Cancel

Here we see the rest of the iSeries Project wizard. Here you specify what to do for the Build action on a project. There are 3 IBM-supplied choices, and ISVs can add more choices. The default is to generate a CL source member containing a compile command for each member that is changed.

Build Styles

- Specify how to build the iSeries project on the remote system
- Three IBM supplied build styles
 - CL program
 - Command
 - *NONE
- ISVs can plug-in additional build styles

CL program

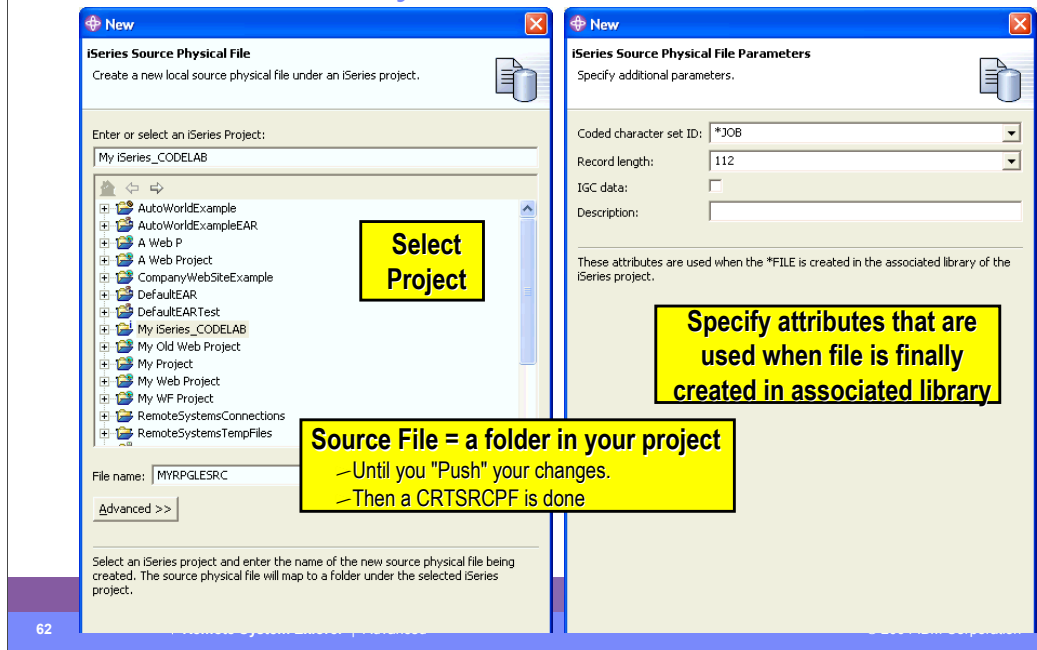
- Automatically generates COMPILE.CLLE and BIND. CLLE
 - Adds compile command for each source file in project
 - Uses last compile command from RSE for specified source type
 - For example, CRTBNDRPG versus CRTRPGMOD
 - At build time these two members are pushed to iSeries, compiled and run

Command

- You specify single command to run build
- Good if you have an existing build script

ISVs can plug-in additional build styles to work with their existing iSeries systems

iSeries Source Physical File Wizard



Here we see the wizard for creating a new iSeries source physical file. Within the project, this is actually a folder. When the project is pushed to its associated library, this results in a CRTSRCPF command being run to create a file with the attributes specified in this wizard.

iSeries Member wizard

Select the parent folder (src file)

Member = a file in your folder

- Until you "Push" your changes.
- Then an ADDPFM is done, or
- Contents of remote member are replaced

Enter the member name and select source type

Member name: MYMBR
Source type: RPGLE

Select an iSeries source physical file, and enter the name of the new source member being created. The member will map to a file in the folder representing the selected iSeries source physical file.

63 Remote System Explorer | Advanced © 2004 IBM Corporation

Here we see the wizard for creating a new iSeries source member within an iSeries project. Actually, it is within a source physical file (aka folder) within an iSeries project. Locally, a member is really a file on disk. The file's extension is the member's type, as in ABC.RPGLE. When this project is pushed to its associated library, this will result in an ADDPFM command being run to create the file with the name and type, in its parent file.

What about existing members?

- Use Import or Add to Project action to copy them
- Files added become folders locally
- Members added are copied to that folder
- Developer works on local copy
- Full power of Eclipse at disposal

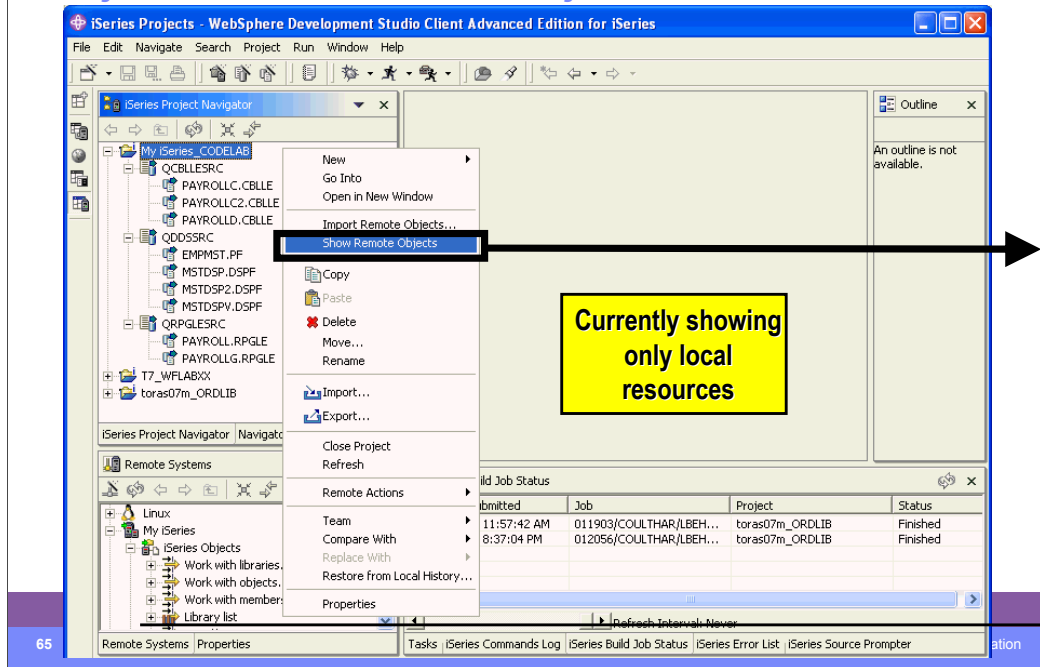
Add project action

- Right click on project, select “Show Remote Objects”
- Drill down to source file, right-click, select “Add to Project”
- Drill down to members, right-click, select “Add to Project”

Developer works on local copy

- Edits file
- Pushes changes using “Push” action
- Does a build using “Build” action
- Shares with team using “Synchronize” action

Project "Show Remote Objects" mode



Here we see an iSeries project in the iSeries Navigator, within the iSeries Projects perspective, that has a number of local files and members. Right-clicking on the project or anything in the project, gives a popup menu with the all-important "Show Remote Objects" menu item....

Project "Show Remote Objects" mode

Now showing both local and resources

To add a remote member to the project, select **Add To Project**

To do actions against remote objects, use **Show In Remote Systems View**

(Remote) =>
 - only exists in associated library

(Local/Remote) =>
 - exists locally and in associated lib

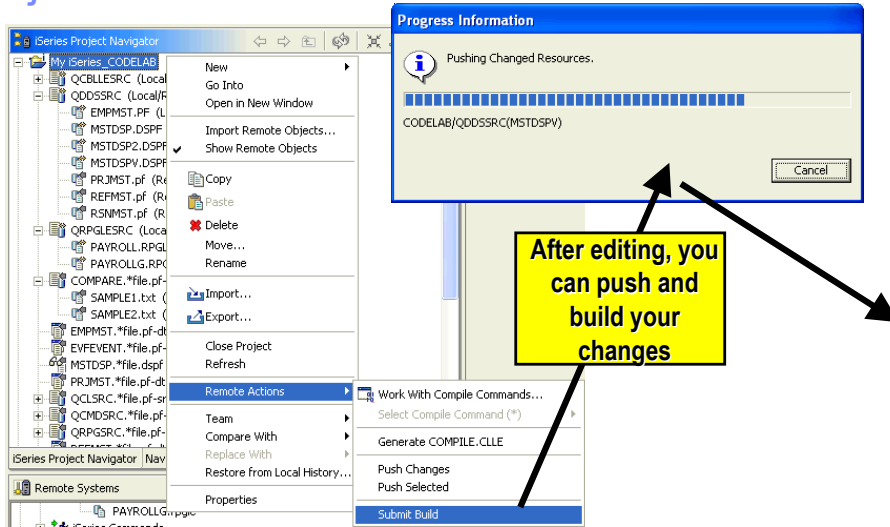
(Local/Remote) =>
 - conflict between local and remote

otherwise =>
 - only exists locally

66 Remote System Explorer | Advanced © 2004 IBM Corporation

Once "Show Remote Objects" is enabled (it is a toggle) all the objects in the associated library for this project are shown in the iSeries Project Navigator. This navigator is called a "bleed through" view because it lists both local files/members and remote objects, files and members. However, if a file or member exists both locally and remotely, it is not shown twice. Rather, it is only shown once, and its icon and bracketed text identify it as existing in both places. The icons and text enable you to see at a glance which files/members exist only locally, only remotely, or both locally and remotely. In the latter case, the color of the icon indicates if the two are in synch or not. If not, it is an indication you need to "push" your local source to the associated library to get it in synch.

Project Build



After editing, you will want to send the changed members back to the associated library, and then compile them there. This can be done in one step with Submit Build, or in two steps with Push Selected and then Generate COMPILE.CLLE.

iSeries Build Job Status

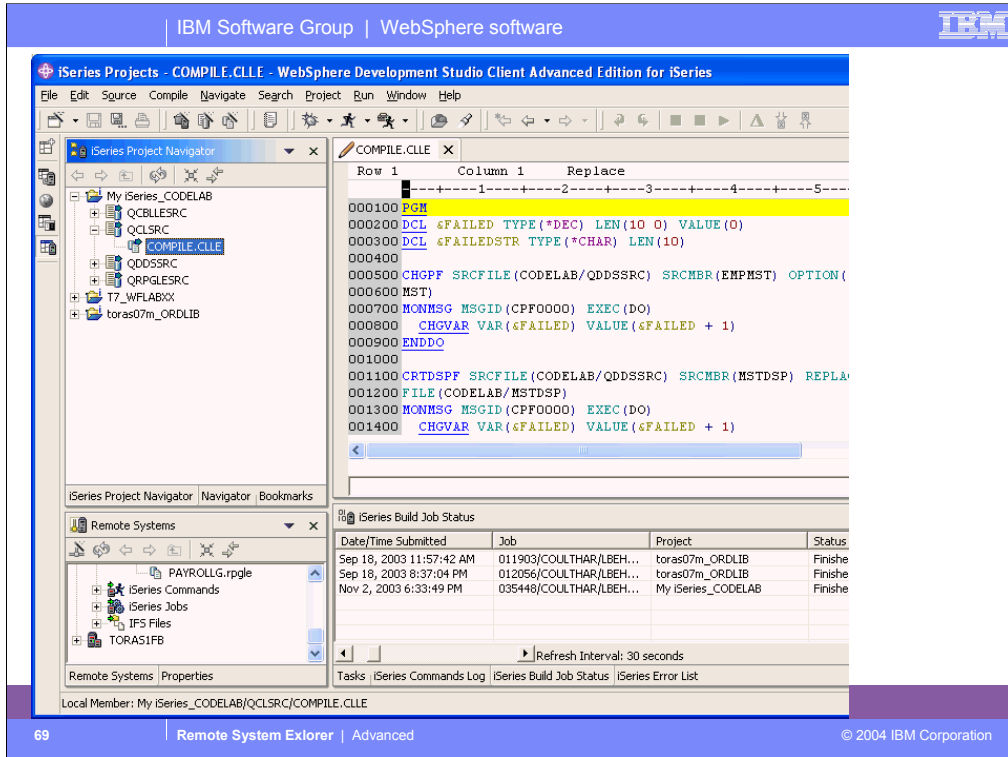
The screenshot shows the 'iSeries Build Job Status' window. It contains a table with columns for Date/Time Submitted, Job, Project, and Status. The table lists three jobs: two 'Finished' and one 'Queued'. A context menu is open over the 'Queued' job, showing 'Cancel', 'Retrieve Errors', and 'Remove' options. A 'Refresh Interval: 30 seconds' label is at the bottom of the window. Annotations with arrows point to the 'Job where build is running' (the 'Queued' job), 'Status of the build' (the 'Status' column), 'Refresh interval determines how frequently status is updated' (the refresh interval label), and 'Retrieve errors from build to the iSeries Error List' (the 'Retrieve Errors' menu option).

Date/Time Submitted	Job	Project	Status
Sep 18, 2003 11:57:42 AM	011903/COULTHAR/LBEH...	toras07m_ORDLIB	Finished
Sep 18, 2003 8:37:04 PM	012056/COULTHAR/LBEH...	toras07m_ORDLIB	Finished
Nov 2, 2003 6:33:49 PM	035448/COULTHAR/LBEH...	My iSeries_CODELAB	Queued

Refresh Interval: 30 seconds

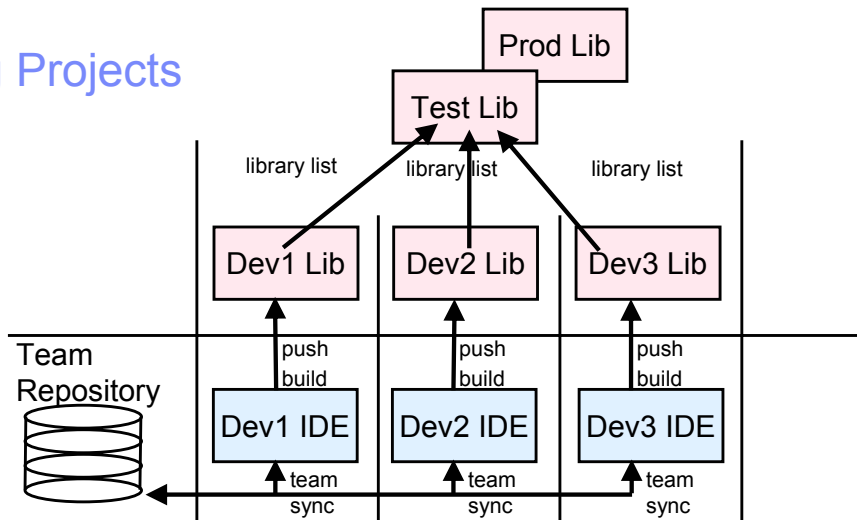
Retrieve Errors

This shows the Build Job Status view where you can monitor your submitted build jobs.



An example of what is generated for the build. This is a CL member. You can affect what compile command is used for each member type.

Using Projects



The typical usage scenario for iSeries projects is to use them for tasks, such as adding a feature to an existing application. Rarely will you use an iSeries project to hold all the source for an application... use an iSeries change management vendor for that. Consider a change that involves 3 developers changing a number of files...

The lead developer will create the iSeries project and synchronize it with the repository. The other developers will use the Eclipse team support to add that project to their workspace. Each will change the properties of the project to associate it with their own private library. The master library containing all the source will be on their library list, so compiles will work, but their private library will only hold the members they work on. If they use an SCM product, then they will use it to check source members out to their private library. These members and files will then be imported into their project, where they will edit them.

After editing, they will "push" their changes to their library and do a "Build" or a compile, and run and test their unique changes.

Every so often, they will do a team synchronization to give their changes to their fellow team members, and pick up the changes of those team members. After a synch, they will push their colleague's files to their own associated library and test the changes all work together.

At the end of the cycle, someone will pick one of the libraries, and use the SCM product to promote their changes back into the test and production stages.

Additional Information

- **homepage:**

<http://ibm.com/software/awdtools/iseries>

Select Library link for Labs, Tutorials, Presentations

- **newsgroup:**

<news://news.software.ibm.com/ibm.software.websphere.code400>

- **email distribution list:**

“This is the Websphere Development Studio Client for iSeries (WDSCI-L) mailing list

To post a message email: WDSCI-L@midrange.com

To subscribe, unsubscribe, or change list options,

visit: <http://lists.midrange.com/mailman/listinfo/wdsci-l>

or email: WDSCI-L-request@midrange.com

Before posting, please take a moment to review the archives at <http://archive.midrange.com/wdsci-l>.”

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 iSeries Application Development presentation team, including work done by: Phil Coulthard, George Farr, Inge Weiss, Claus Weiss, and Don Yantzi

Disclaimer:

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

Reproduction:

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



IBM Software Group

Remote System Explorer in IBM WebSphere Development Studio Client for iSeries: **Advanced**

WebSphere. software



1H 2004 | WDSC V5.1 and WDS V5.R2

© 2004 IBM Corporation