IBM WebSphere Development Studio Client for iSeries

Work offline with iSeries Projects

Version 6.0.1

IBM WebSphere Development Studio Client for iSeries

Work offline with iSeries Projects

Version 6.0.1

Contents

Introduction

When you do iSeries™ project based development you will want to work in the iSeries Projects perspective. This perspective serves two main purposes, namely for performing iSeries offline development on Windows® clients and for enabling team development.

Offline development allows for a disconnected mode of development where a live connection to a host is only required when code updates or builds are needed.

Team development is needed when code changes need to be shared between a team of developers and synchronization of these changes is needed as part of the development cycle. Team development is not covered in this tutorial.

Typically, you create local source physical files and members and edit them locally. You can also download existing members from any remote library into your project. Once done with the local editing, you can submit a build that will push all your changes to the associated library of the iSeries Project and perform a build on the remote members. The code pushed to this scratch library is only pushed there for the purpose of performing that build. If source code management is needed, the iSeries Project could be associated with any Team provider, for example, Rational® ClearCase® LT version control, that is installed with Development Studio Client.

This tutorial teaches you about disconnected development. A live connection is required only when code updates or builds are needed, or when you need to view remote resources for a project. If you are working on a remote laptop, or are at home with a slower connection, you might choose to work in disconnected mode. With this option, you explicitly copy files locally while you have a live connection, close the connection, work on the files at your leisure, and upload them back to the server once you have finished. Then you can run another verify to locally check your source code for syntax and semantic errors and submit a compile only when your code is error free.

Prerequisites

In order to complete this tutorial end to end, you should already have completed the tutorial named Maintain an ILE application using Remote System Explorer and have working knowledge of the following:

- Basic Microsoft® Windows operations such as working with the desktop and basic mouse operations such as opening folders and performing drag-and-drop operations.

It is also useful, but not necessary, for you to have basic knowledge of ILE RPG.

Time required

To complete the modules in this tutorial, you will need approximately **1 hour**.

Learning objectives

This tutorial is divided into 3 modules, each with their own specific learning objectives. You can choose to complete one or all of the modules. Each module contains several exercises that must be completed in order for the tutorial to work properly.

Chapter 1, "Module 1. Starting the product and connecting to an iSeries," on page 1 teaches you how to start the product and how to configure and create a connection in order that you can see the payroll application source code. In this module, you will:
- Start the product
- Set the workspace location
- Set up a connection to an iSeries system
- Add a library to your library list

Chapter 2, "Module 2. Working with source in disconnected mode," on page 9 teaches you how to make source available for offline or disconnected development. Here you will learn to edit and verify the source disconnected from the iSeries system. You will also learn how to compile your changed member. In this module, you will:
- Download selected files or members to your workstation
- Verify downloaded source
- Launch the editor on a local source member under an iSeries Project
- Put an iSeries connection into offline mode
- Edit a local source member under an iSeries Project
- View an outline of the local RPG source
- Open a local DDS member for edit from the iSeries Projects Navigator
- Verify changed RPG source
- Compile a single member

Chapter 3, "Module 3. Creating and Building an iSeries Project," on page 19 teaches you another approach to working with iSeries Projects. You will learn how to create a new iSeries project, create new physical files and source members and how to populate the iSeries projects with members. You will also learn how to show remote objects or local objects or both and how to submit a build for the entire project. In this module, you will:
- Create a personal development library (single developer sandbox)
- Create an empty iSeries Project
- Populate the empty iSeries Project
- Select a build style for the iSeries Project
- Build the iSeries Project
- View all the remote objects in the iSeries Project Navigator
- Filter the objects to remove the remote only objects from the iSeries Project Navigator
- Remove the Local/Remote Flags from the iSeries Project Navigator
- Push only a changed file to the iSeries

When you are ready, begin Chapter 1, "Module 1. Starting the product and connecting to an iSeries," on page 1.

Chapter 1. Module 1. Starting the product and connecting to an iSeries

This module teaches you how to start the product and create and configure a connection to an iSeries system.

In this module, you will:
- Start the product
- Set the workspace location
- Set up a connection to an iSeries system
- Add a library to your library list

Requirement: Before beginning this module, you should have the prerequisite knowledge outlined in "Introduction" on page v.

Exercises

The exercises within this module must be completed in order.
- "Exercise 1.1: Starting the product"
- "Exercise 1.2: Configuring a connection to an iSeries system" on page 2
- "Exercise 1.3: Connecting to an iSeries system" on page 4

Time required

This module will take approximately **10 minutes** to complete.

Exercise 1.1: Starting the product

To start the product:
1. Click **Start** on the task bar of your desktop
2. Select **Programs** > **IBM® Rational** > **IBM WebSphere® Development Studio Client for iSeries V6.0** > **WebSphere Development Studio Client for iSeries**

3. A dialog will appear. Here you specify the name of the workspace where your projects and other resources such as folders, subfolders and files that you are developing in the workbench will reside.
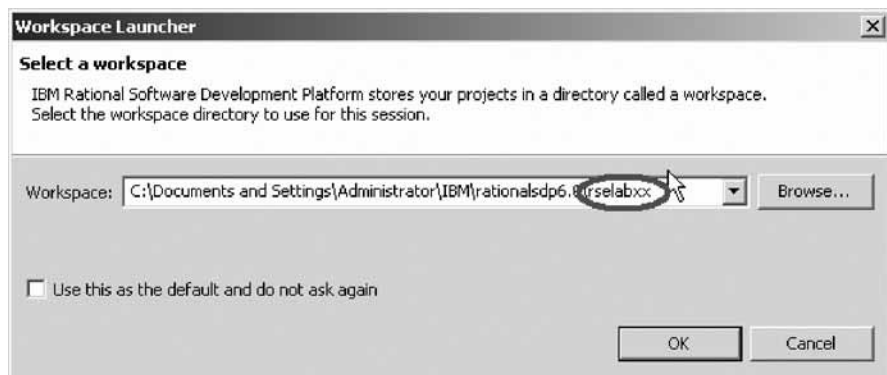
4. Use the same name of the workspace that you used for the tutorial named Maintain an ILE application with Remote System Explorer, for example, RSElabXX (where XX is a unique number), or use a different workspace.

5. Click **OK** to open the workbench.

You have started the product and set the workspace location and now you are ready to begin "Exercise 1.2: Configuring a connection to an iSeries system."

Exercise 1.2: Configuring a connection to an iSeries system

Before you begin, you must complete "Exercise 1.1: Starting the product" on page 1.

When you first start the workbench and open a perspective, you are not connected to any system except your local hard drive on your workstation.

Before you can work offline you need to connect to an iSeries system to download the source files that you need.

You may skip this chapter if you want to use an existing connection from the previous tutorial named Maintain an ILE RPG application with Remote System Explorer.

To configure a connection:

1. Click the X to close the Welcome view.

2. In the Remote Systems view **New Connection** shows the various remote systems types you can connect to through the Remote Systems view.
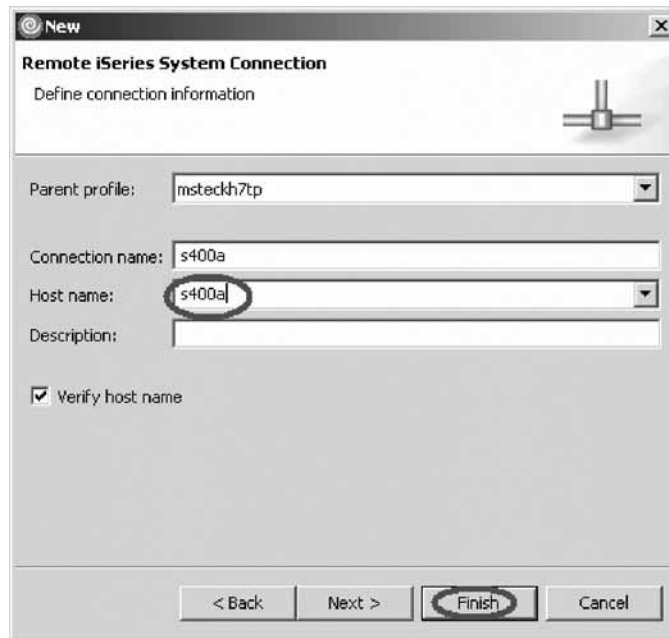
3. Expand **iSeries** to configure a connection to an iSeries system.

   The Name personal profile page opens if this is the first time you are creating a connection.

4. Click **Next** to accept the default value.

The profile defaults to the name of the workstation. Your profile will be different from the one shown here.

The Remote iSeries System Connection page opens:

On this second page you specify the information for your connection. The cursor on this page is positioned in the **Host Name** field.

5. In the **Host name** field, type the name of your host system.

   The Connection name is automatically filled with the host name. Leave it this way. This name displays in your Remote Systems view and must be unique to the profile.

6. Leave the **Parent profile** default value. You don't need to change it.

7. Leave the **Verify host name** check box selected.

8. Click **Finish** to create your connection.

You have set up a connection to an iSeries system and now you are ready to begin "Exercise 1.3: Connecting to an iSeries system."

Exercise 1.3: Connecting to an iSeries system

Before you begin, you must complete "Exercise 1.2: Configuring a connection to an iSeries system" on page 2.

After you configure a connection to an iSeries system, you can easily connect and expand your new connection to show the subsystems. Subsystems are a function grouping of the various types of remote resources that can be explored in the remote system.
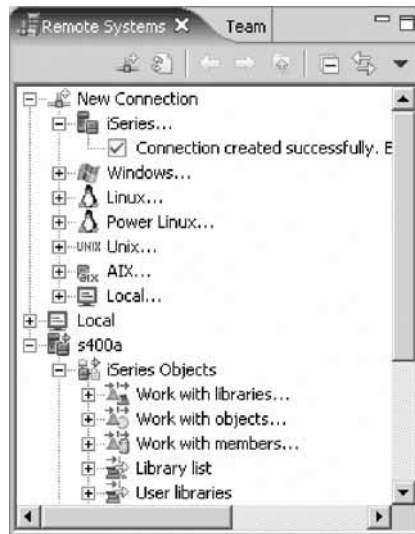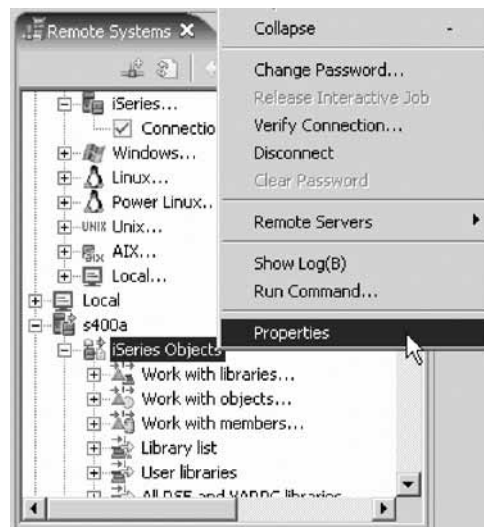
To connect to an iSeries system:

1. In the Remote Systems view, expand your new connection to show the subsystems.

   Now let's work with a library in our library list and add the library that you'll be using in this tutorial:

   a. Right click **iSeries Objects** and click **Properties** on the pop-up menu.

   b. Select **Initial Library List** on the left pane.
   c. Type RSELABXX in the **Library** field and click **Add**.
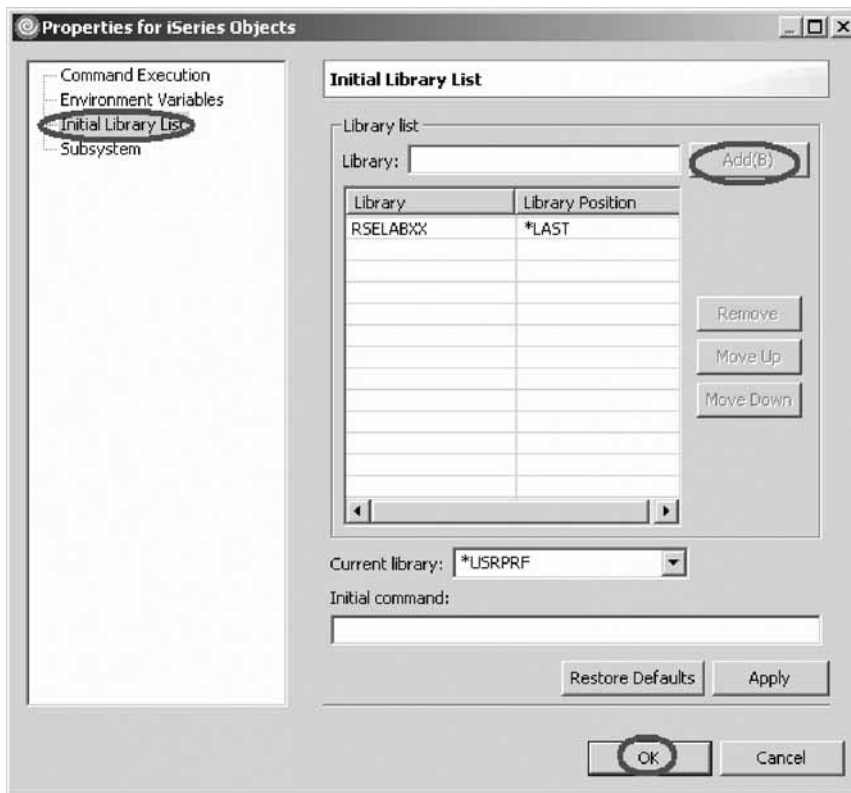
    d. Click **OK**.

       This will add the library RSELABxx to your library list every time you use this connection.

> **Note:** You can also change your library list using the pop-up menu items **Add Library List Entry** or **Change Current Library** on the **Library list** folder in the iSeries Objects subsystem. These changes are only valid until you disconnect and only known to the Remote Systems Communication server job.

2. Expand **iSeries Objects** if it is not already.
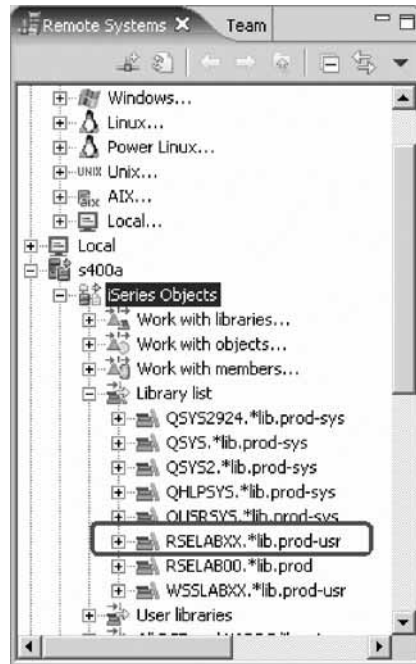3. Expand the **Library list** folder.

   Now the connection will be activated and you will be prompted for a user ID and password.

4. Enter your user ID and password.
5. Select the **Save user ID** check box.
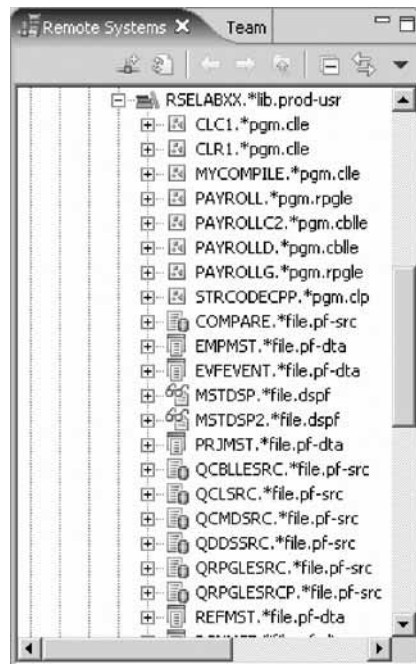6. Select the **Save password** check box.

7. Click **OK**.

   Back in the workbench you will see the libraries in your job's library list.

   Notice that the s400a node now has a small green arrow in the icon to indicate it is an active connection.

8. Expand RSELABxx to see the source files within this library. You will use some of these files in the next module.

   You have added a library to the library list.

   Module recap

You have completed Chapter 1, "Module 1. Starting the product and connecting to an iSeries," on page 1. You have learned how to:

- Start the product
- Set the workspace location
- Set up a connection to an iSeries system
- Add a library to your library list

Now that you have started the product, created and configured a connection, and located the source code for the payroll application, you are ready to download source to your workstation. Continue to Chapter 2, "Module 2. Working with source in disconnected mode," on page 9.

Chapter 2. Module 2. Working with source in disconnected mode

This module teaches you how to download selected files or members in order that you can work on these files disconnected from an iSeries. An iSeries project will be created for you automatically. You will verify the downloaded source before you disconnect. You will then disconnect from the iSeries. Next you will edit and verify source on your workstation. Finally you will compile your source. During the compile step you are automatically reconnected to the iSeries system.

In this module, you will:
- Download selected files or members to your workstation
- Verify downloaded source
- Launch the editor on a local source member under an iSeries Project
- Put an iSeries connection into offline mode
- Edit a local source member under an iSeries Project
- View an outline of the local RPG source
- Open a local DDS member for edit from the iSeries Projects Navigator
- Verify changed RPG source
- Compile a single member

Exercises

The exercises within this module must be completed in order.
- "Exercise 2.1: Making source available offline"
- "Exercise 2.2: Editing local source members in disconnected mode" on page 12
- "Exercise 2.3: Verifying RPG source in disconnected mode" on page 15
- "Exercise 2.4: Compiling a member" on page 16

Time required

This module will take approximately **20 minutes** to complete.

## Exercise 2.1: Making source available offline

You want to edit and verify your source on a workstation. You do not have an existing iSeries Project. You can use the Make Available Offline option.

The Make Available Offline option downloads the selected files or members locally to their corresponding iSeries project. The corresponding iSeries project has the same host and the same library properties as the host name and the library of the selected source physical files or members. If such an iSeries project is not found, one will automatically be created.

To make source available offline:
1. Expand the RSELABxx library (if it is not already) that contains the source files that you want to work with offline.

2. Select the source physical files QDDSSRC and QRPGLESRC as you want the members in these source files to be made available for offline development. You can use the Shift key to make multiple selections.

3. Right-click and click **Make Available Offline** on the pop-up menu.
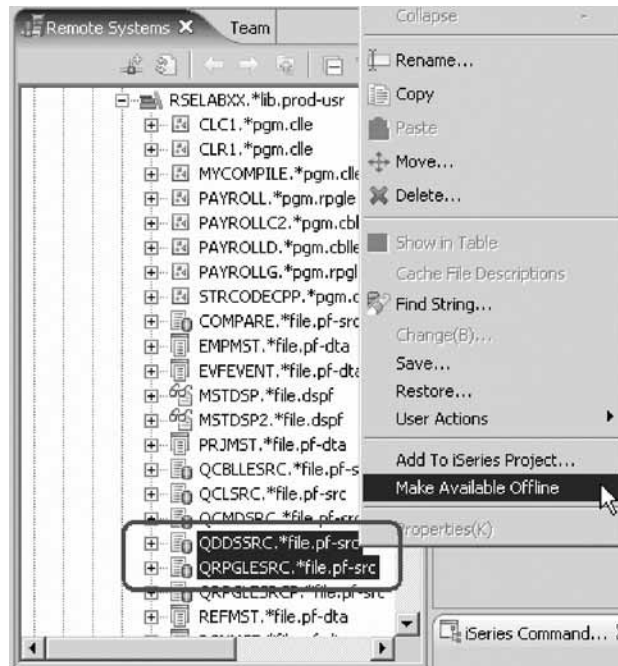
The iSeries Projects Question dialog opens.

4. Click **Yes** to proceed.
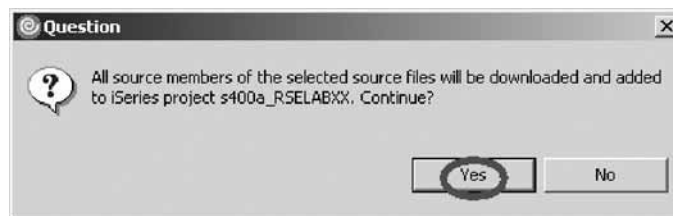
The iSeries Projects perspective opens with your new iSeries Project named s400a_RSELABxx. This project is created for you automatically and the source files QDDSSRC and QRPGLESRC are added to this project.

Note: If you want to add more source files later to this iSeries project, you can use the Make Available Offline option again.
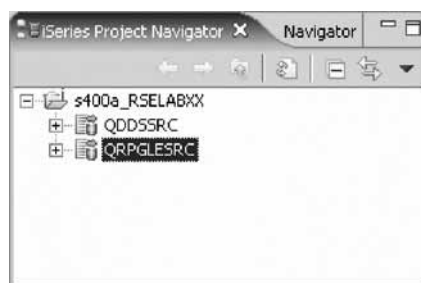
Now you need to verify that you have the latest cached source available before you begin to work offline.

10 Development Studio Client: iSeries Projects

5. Expand QRPGLESRC and double-click PAYROLLG to open this source file in the LPEX Editor.

6. Click **Source** and click **Verify** on the workbench menu.
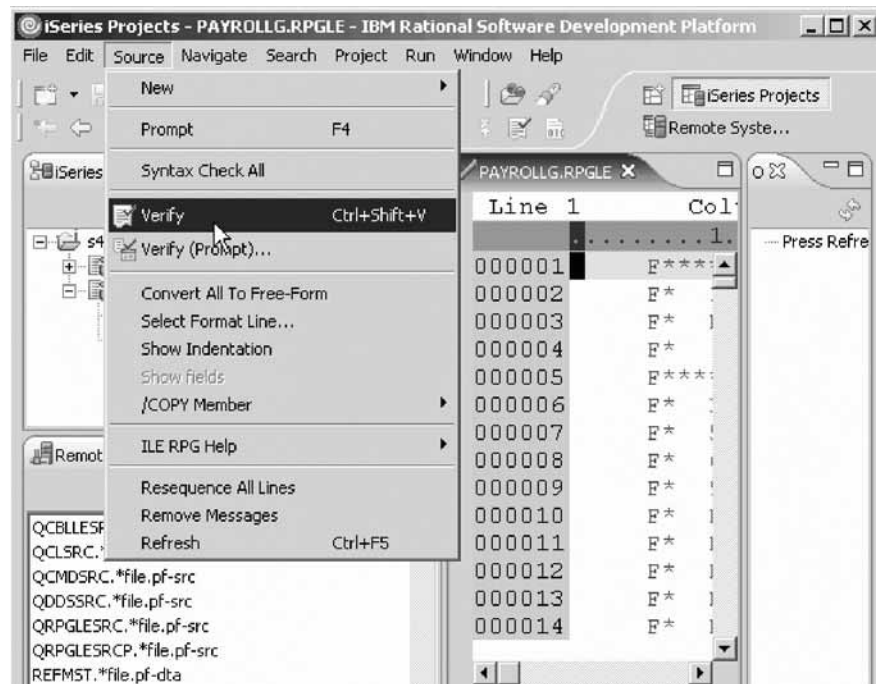
This action will verify your member for syntax and semantic errors and any errors will appear in the iSeries Error List and cache all information required for subsequent offline verifys.

Now you are ready to disconnect from the iSeries server.

7. From the Remote Systems view, just below the iSeries Project Navigator, right-click on the connection and click **Disconnect**.

> **Note:** You can also put an entire iSeries connection, and all the contents defined with that connection, into offline mode by right-clicking the connection and clicking Work Offline on the pop-up menu. This method of working offline prevents the Remote System Explorer from attempting to connect to the iSeries server if a request is made for remote information. For example, if you are verifying a local source file or member in an iSeries project, the project would attempt to connect to the server if the required information was not available in the cache. This can be prevented by switching the connection to offline mode.

You have downloaded selected files or members to your workstation, verified downloaded source, launched the editor on a local source member under an iSeries Project and put an iSeries connection into offline mode and now you are ready to begin "Exercise 2.2: Editing local source members in disconnected mode."

Exercise 2.2: Editing local source members in disconnected mode

Before you begin, you must complete "Exercise 2.1: Making source available offline" on page 9.

Editing local source members under iSeries projects is the same as editing remote source members in the Remote Systems Explorer. The LPEX editor can be launched on any local member and the same functionality is available, such as the Outline view, prompter and the content assist feature.

There is one major difference though when editing source members using the LPEX editor in the Remote System Explorer perspective as opposed to using the LPEX editor in the iSeries Projects perspective. When a source member is opened in LPEX in the Remote System Explorer perspective, it is locked, and it cannot be modified on the host while the local edit session is active. Any code changes that you make locally will be saved to the host when the Save action is performed in the LPEX editor.

On the other hand, when a member is opened in the LPEX editor in the iSeries Projects perspective, the member is not locked on the host. It can be modified by anyone with enough authority to the object. If you do a Refresh on the iSeries Project, a double sided red arrow will indicate that the member has been modified on the host. When you push changes back to the host, (something you will do when you are connected again), you will be warned that you will be overwriting remote changes, and you are given the chance to cancel the operation.

Now let's edit the member PAYROLLG that you already have open in the editor.

To edit PAYROLLG:
1. In the PAYROLLG Editor window move the cursor to line 211 which contains EXSR ACDESR.
2. Append an X to the EXSR op-code to make it EXSRX.
3. Click **File** > **Save** from the workbench menu to save this change.
   You will fix this error later.

Displaying an outline of an ILE RPG member

The Outline view acts as an excellent resource when you want to edit RPG and COBOL source in the Remote Systems LPEX Editor. The Outline view displays a structural outline of items defined in the source that you currently have open in the Remote Systems LPEX Editor window. With the editor active, you can expand the file structure in the Outline view, and click various elements in the view to jump to that location in the source itself.

To see an Outline view of your RPG source:
1. Click the **PAYROLLG** tab in the editor and click **Refresh** on the Outline view toolbar.
   The Outline view contains your source program in a tree view without the lines containing logic.
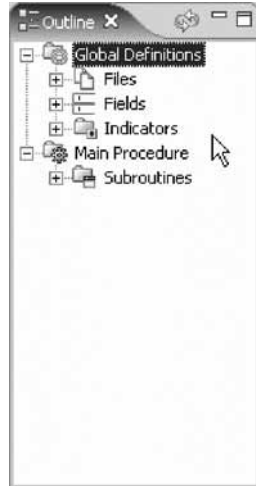
Now you want to see more details of your source member.

2. Expand **Files**.
3. Expand the MSTDSP workstation file.
4. Expand the EMPSEL record format.
5. Double-click on any of the entries in the Outline view.

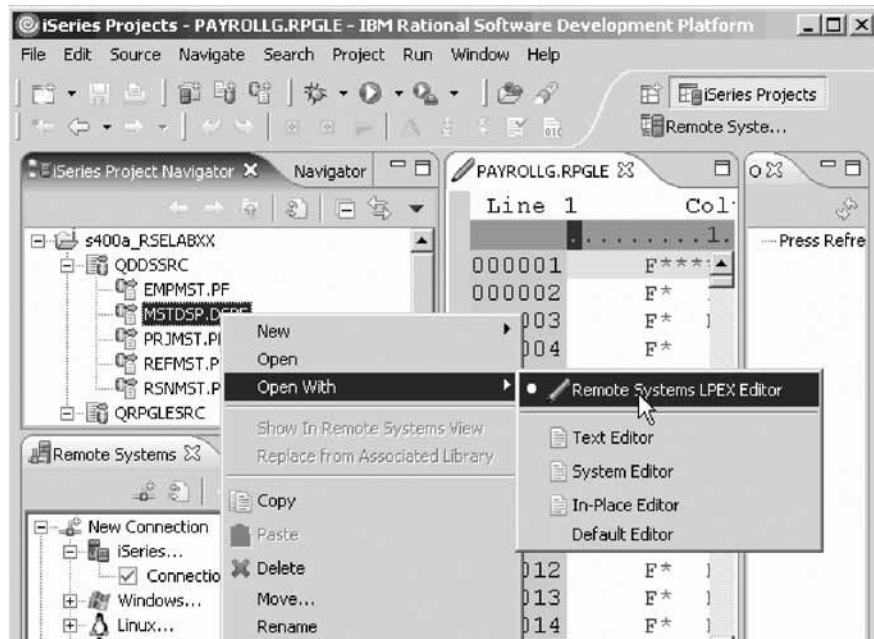This will position the source editor accordingly.
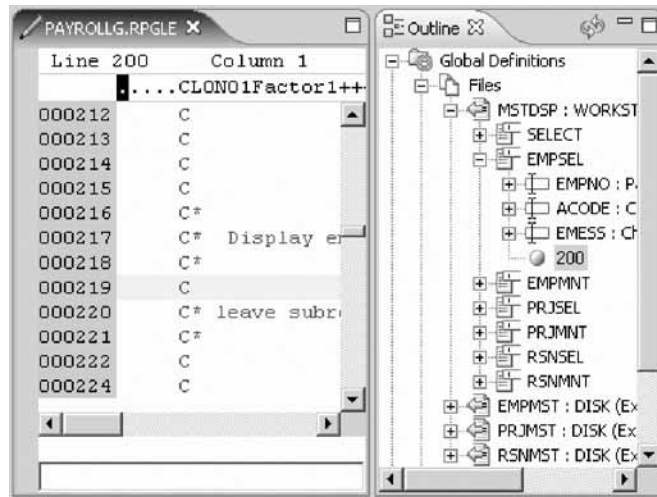
Opening a DDS member

You can also edit DDS members locally from the iSeries Projects Navigator.

To open a DDS member:

1. In the iSeries Project Navigator, expand QDDSSRC, if it is not already expanded.

2. Right-click MSTDSP member and click **Open With**.

As in the Remote System Explorer perspective, you now have the options to open the Remote Systems LPEX Editor or other editors. CODE Designer is not registered as an external editor for display files, but you can change that by manually adding it through the preference page, and it will then appear in the Open With menu. To do this:

   a. Select **Window** > **Preferences** from the main menu.

   b. Expand **Workbench**. Select **File Associations**.

   c. Select *dspf from **File types**.

   d. Click **Add** next to the **Associated editors** list view.

   e. Choose **External Programs** and then browse to the location where CODE Designer (codedsu.exe), is installed. (Look for the directory named WDSC.) Click **Open**. Click **OK**.

      Next you will add the CODE Editor.

   f. Click **Add** next to the **Associated editors** list view.

   g. Choose **External Programs** and then browse to the location where CODE Editor (codeedit.exe), is installed. (Look for the directory named WDSC.) Click **Open**. Click **OK**.

   h. Click **OK** to close the Preferences window.

      Now, if you right click MSTDSP member and click **Open With** you will see the codedsu and codeedit added to the pop-up menu.

You have edited a local source member under an iSeries Project, viewed an outline of the local RPG source and opened a local DDS member for edit from the iSeries Project Navigator and now you are ready to begin "Exercise 2.3: Verifying RPG source in disconnected mode."

Exercise 2.3: Verifying RPG source in disconnected mode

Before you begin, you must complete "Exercise 2.2: Editing local source members in disconnected mode" on page 12.

While you are disconnected from an iSeries, you can make certain that there are no errors by invoking the Program Verifier. The verifier checks for syntactic and semantic (compile) errors on your workstation so that you can guarantee a clean compile when you are connected to an iSeries. The iSeries 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.

To verify source:
1. Click the **PAYROLLG** tab.
2. Click **Source** > **Verify** from the workbench menu. (Similarly, you can also use the Verify tool button — you need the source in the editor and the tab needs to be selected for the button to be enabled.)

    After a moment the verifier will display an iSeries Error List below the Editor window.
3. Use the iSeries Error List to locate the error you created earlier and fix it.
4. Save the change to the source that you just made.
5. Verify the source again.

You have verified changed RPG source and now you are ready to begin "Exercise 2.4: Compiling a member."

Exercise 2.4: Compiling a member

Before you begin, you must complete "Exercise 2.3: Verifying RPG source in disconnected mode" on page 15.

So now that the code has been edited locally and verified, you are ready to perform a single member compile on the source member in your iSeries project.

To compile a single member:
1. First change the preferences to compile interactively. Click **Window** > **Preferences** from the workbench menu.
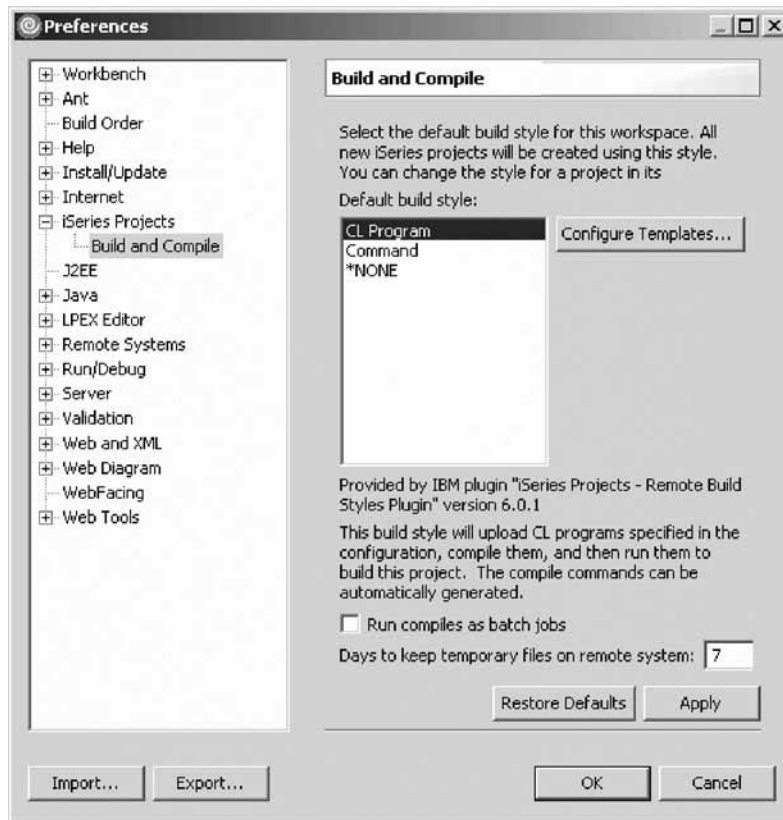
2. Expand **iSeries Projects**.

3. Select **Build and Compile**.

4. Clear the **Run compiles as batch jobs** check box.

5. Click **OK**.

6. Expand QRPGLESRC if it is not already.

7. Right-click PAYROLLG and select **Remote Actions** > **Compile** > **CRTBNDRPG** on the pop-up menu.
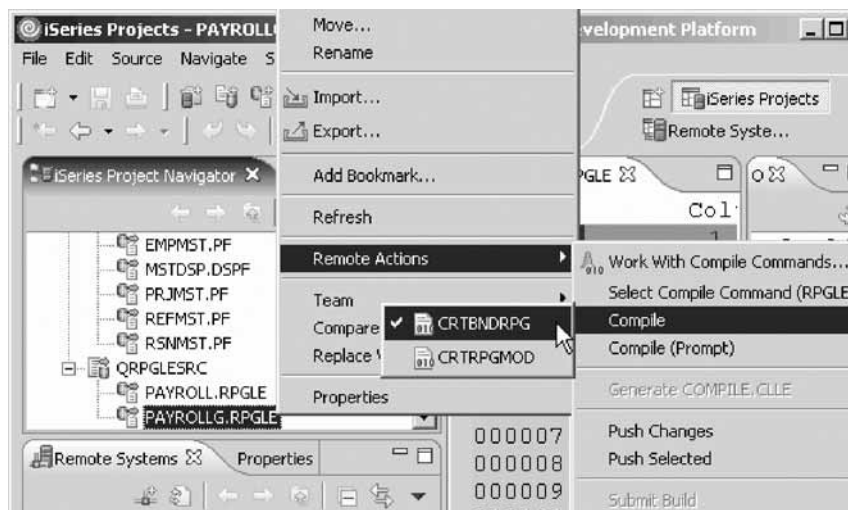
Now the connection will be activated and you may be prompted for a user ID and password.

8. You automatically connect to your iSeries system.

Any local changes are uploaded to the same file in the associated library on the iSeries system.

This action will compile your member and any errors will appear in the iSeries Error List.

If you are not sure that the compile was successful, you can check the iSeries Commands Log.

You could now run or debug your changed application.

You have compiled a single member.

Module recap

You have completed Chapter 2, "Module 2. Working with source in disconnected mode," on page 9. You have learned how to:
- Download selected files or members to your workstation
- Verify downloaded source
- Launch the editor on a local source member under an iSeries Project
- Put an iSeries connection into offline mode
- Edit a local source member under an iSeries Project
- View an outline of the local RPG source
- Open a local DDS member for edit from the iSeries Projects Navigator
- Verify changed RPG source
- Compile a single member

Another way to work with iSeries Projects is to create a new project, populate that project with members and build the entire project. Continue to Chapter 3, "Module 3. Creating and Building an iSeries Project," on page 19.

Chapter 3. Module 3. Creating and Building an iSeries Project

Instead of making remote objects available offline as you did with the Make Available Offline option, you can create an iSeries Project and import or add objects to that project. To import objects into an iSeries Project you must be connected to an iSeries system.

This module teaches you how to create an iSeries Project and import objects into that project. You will also learn how to show remote objects and filter out remote objects. Finally you will learn how to build the full project.

In this module, you will:
- Create a personal development library (single developer sandbox)
- Create an empty iSeries Project
- Populate the empty iSeries Project
- Select a build style for the iSeries Project
- Build the iSeries Project
- View all the remote objects in the iSeries Project Navigator
- Filter the objects to remove the remote only objects from the iSeries Project Navigator
- Remove the Local/Remote Flags from the iSeries Project Navigator
- Push only a changed file to the iSeries

Exercises

The exercises within this module must be completed in order.
- "Exercise 3.1: Creating a scratch library"
- "Exercise 3.2: Creating an iSeries Project" on page 22
- "Exercise 3.3: Creating a source file and a source member" on page 24
- "Exercise 3.4: Importing members into your project" on page 28
- "Exercise 3.5: Selecting the build style" on page 30
- "Exercise 3.6: Submitting a build" on page 31
- "Exercise 3.7: Showing remote objects" on page 33
- "Exercise 3.8: Filtering out remote objects" on page 34
- "Exercise 3.9: Changing PAYROLLG again and submitting another build" on page 35

Time required

This module will take approximately **30 minutes** to complete.

Exercise 3.1: Creating a scratch library

You will need an associated library when you create an iSeries Project in the next exercise. The associated library is a ″scratch″ library on the remote iSeries system that an iSeries project maps to. It is used for pushing code changes made locally to the remote host and for performing builds on that remote host. It is expected that this library is a ″personal″ library and the goal is that there should not be any need

for any source in it to be edited directly on the host. It is meant to be a sandbox in which a single developer can perform tasks.
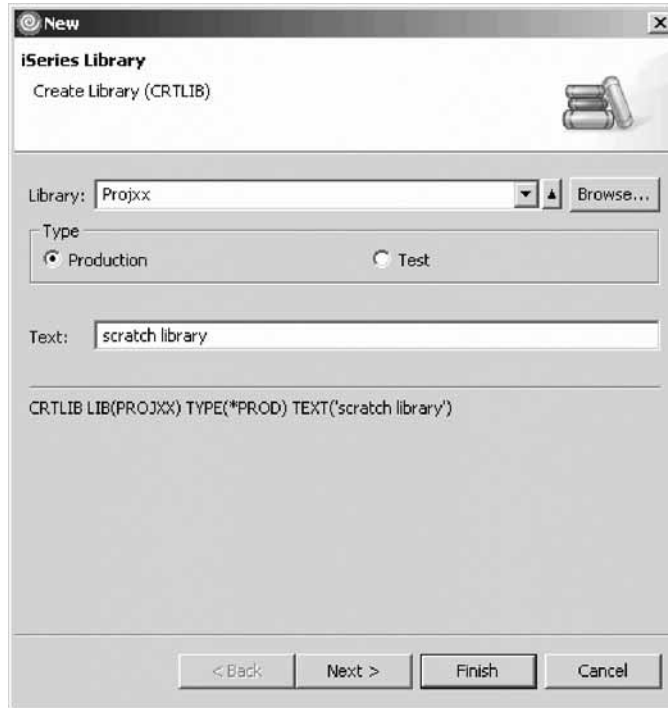
To create a scratch library:

1. In the Remote Systems view below the iSeries Project Navigator, right-click **iSeries Objects** and click **New** > **Library** on the workbench menu.
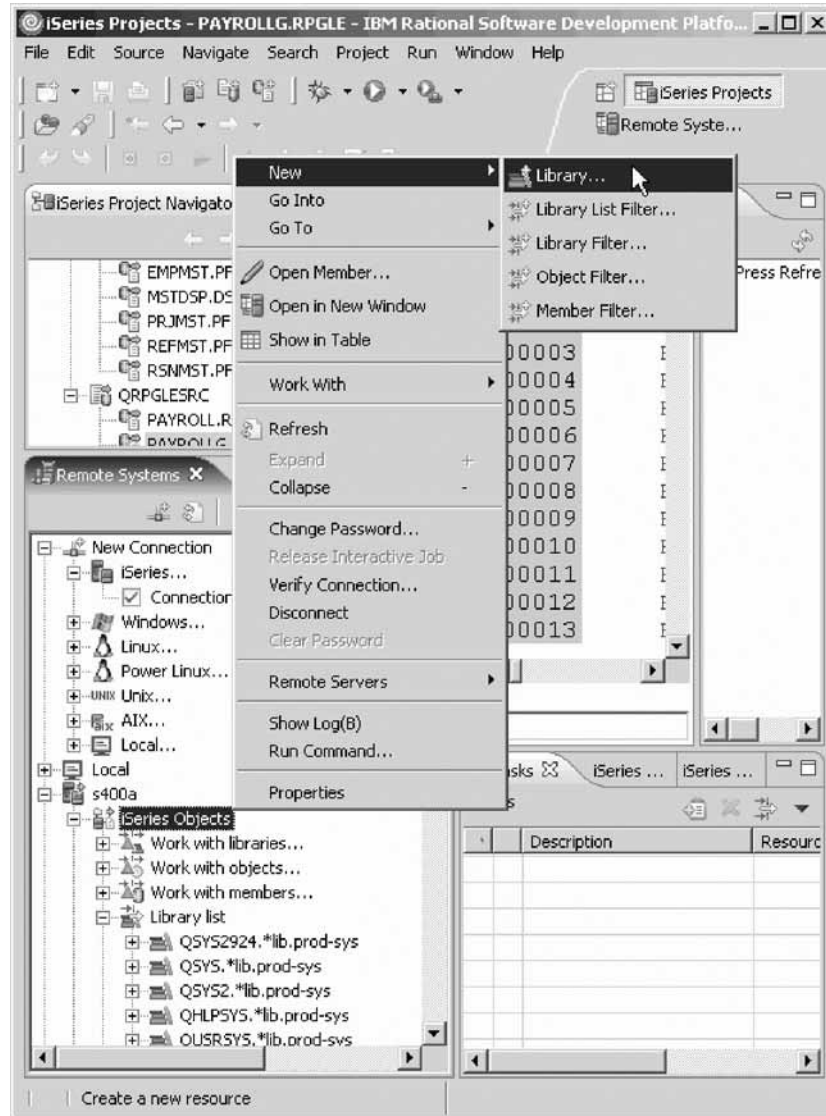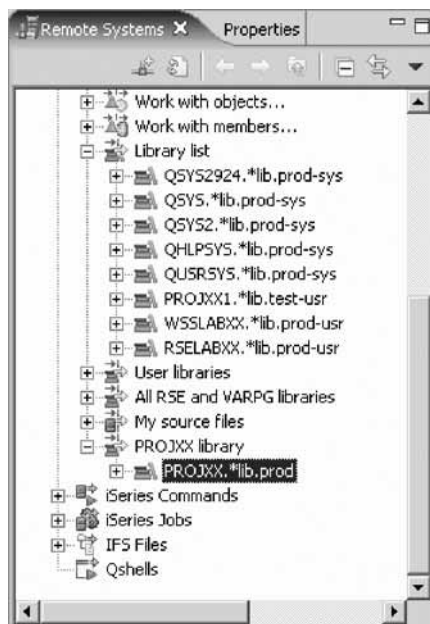
The iSeries Library page opens.

2. In the **Library** field, type `Projxx`.

3. Select the **Test** radio button to change the type of library to Test.
4. Type `scratch library` for the **Text** field.
5. Click **Finish**.

   This will create an empty iSeries Library.

You have created a personal development library (single developer sandbox) and now you are ready to begin "Exercise 3.2: Creating an iSeries Project" on page 22.

Exercise 3.2: Creating an iSeries Project

Before you begin, you must complete "Exercise 3.1: Creating a scratch library" on page 19.

iSeries Projects are projects that are dedicated for holding iSeries source members destined to be pushed and compiled on an iSeries machine. They follow the Workbench paradigm in that they have a special perspective, the iSeries Projects perspective, which is dedicated to showing and working with these projects. When a new iSeries project is created three properties need to be associated with it. These are an iSeries Connection, an Associated Library, and a Build Style.
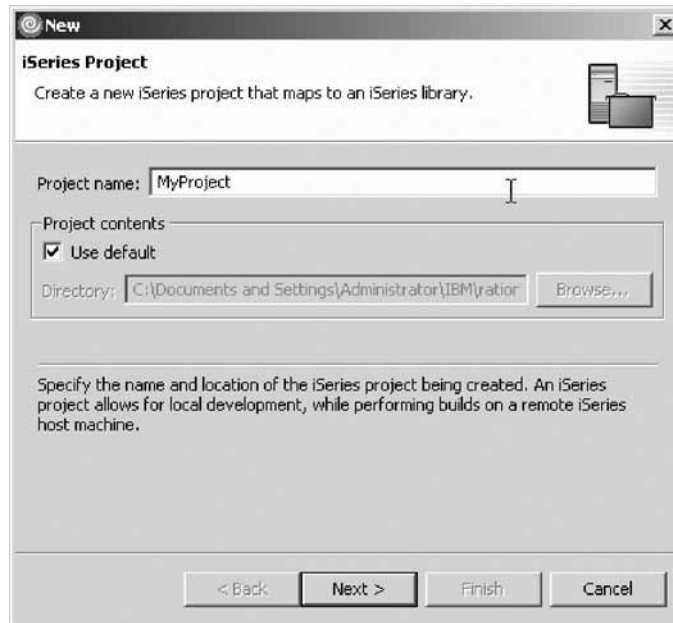
To create an iSeries project:
1. Click **File** > **New** > **iSeries Project** on the workbench menu.
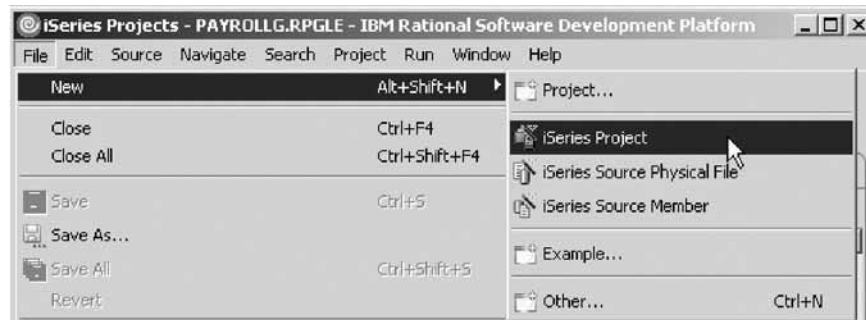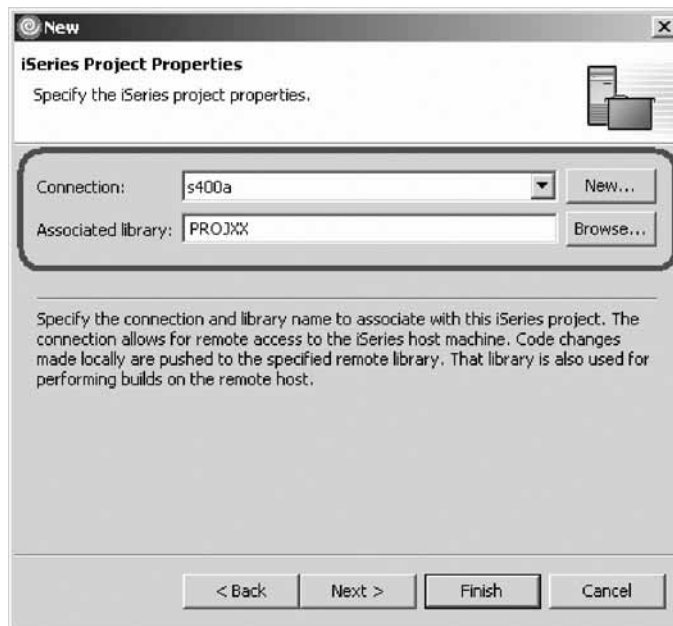

The iSeries Project page opens.
2. In the **Project name** field, type MyProject.

3. Click **Next**.
4. Select your connection from the **Connection** list, for example, s400a.

    The property iSeries connection allows for remote access to the iSeries host machine, and handles various aspects, including user ID and password authentication to the remote host. It also provides the initial environment for builds, including the base library list.

5. In the **Associated library** field, type PROJXX

The associated library is the library on the iSeries system that your iSeries project maps to.

6. Click **Next**.

7. Under **Active build style**, select **CL Program**.
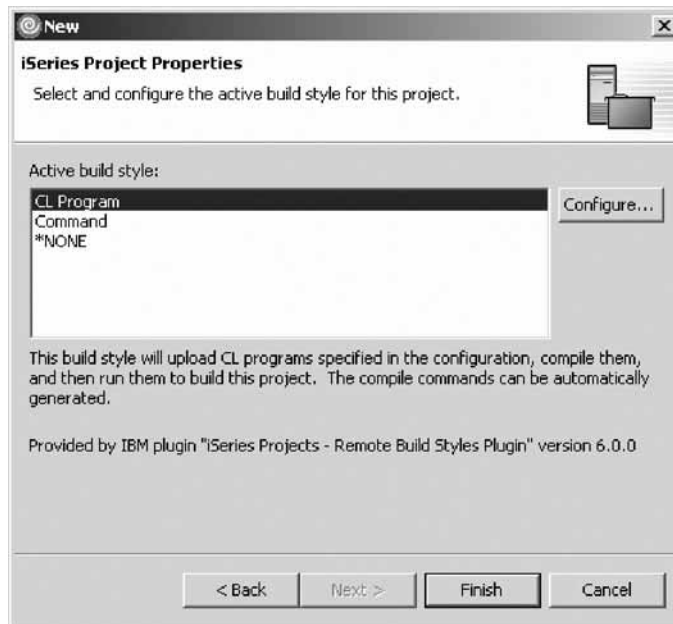
The property build style determines how a given iSeries Project is built. It determines how source physical files and members are pushed to the remote host, and how these members are compiled there.

8. Click **Finish**.

This will create an empty iSeries project that you have to populate. This is useful when you are creating new iSeries members from scratch and you don't have them already on a host.
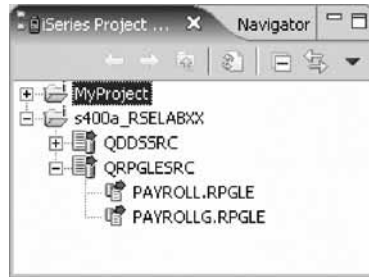
You have created an empty iSeries Project and now you are ready to begin "Exercise 3.3: Creating a source file and a source member."
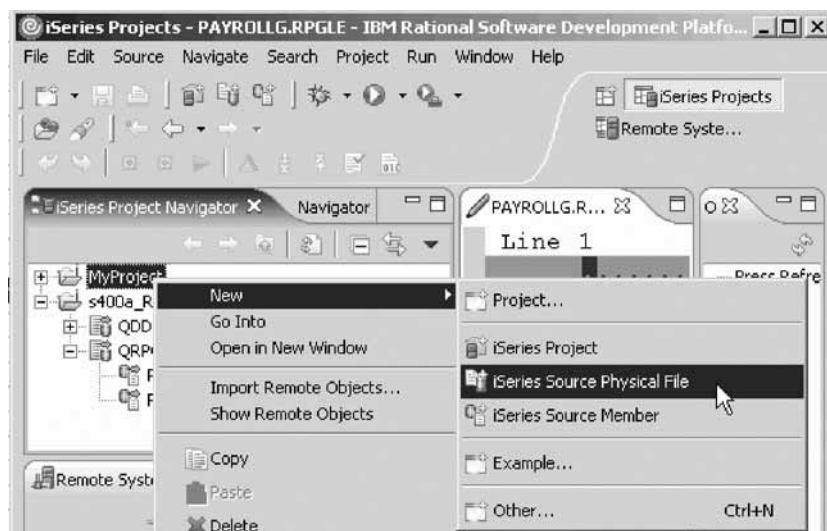
Exercise 3.3: Creating a source file and a source member

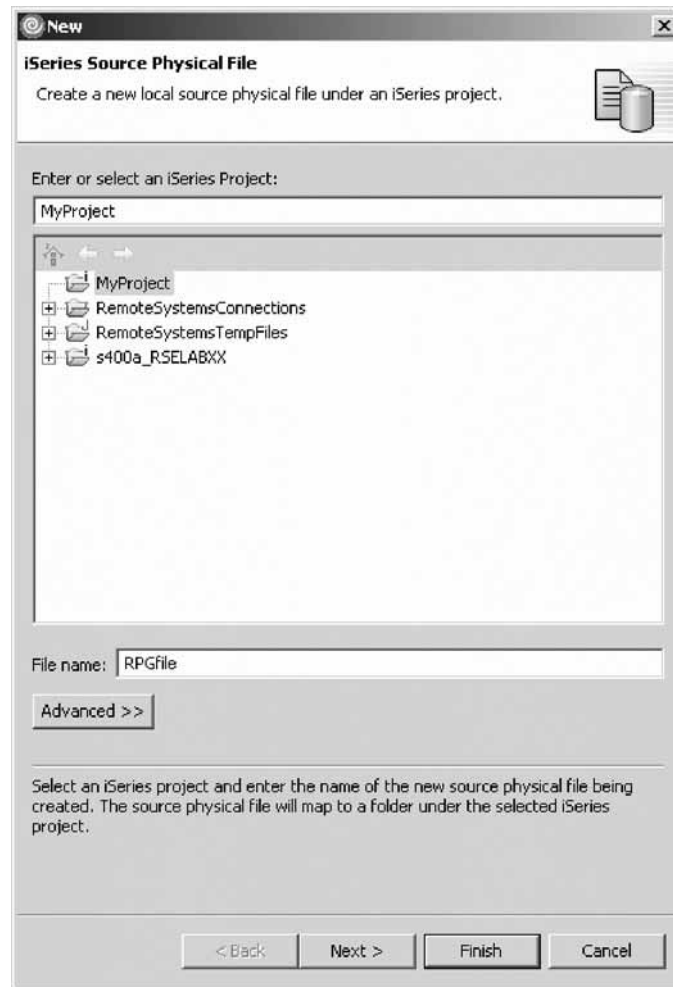Before you begin, you must complete "Exercise 3.2: Creating an iSeries Project" on page 22.

You can create a source physical file and a source member for your project.

To create a source physical file:

1. Right-click your iSeries Project.
2. Click **New** then click **iSeries Source Physical File** on the workbench menu.

The iSeries Source Physical File page opens.

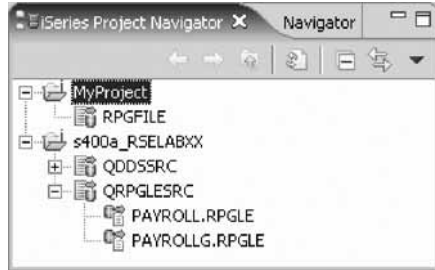3. Enter a name for your file in the **File name** field.

4. Click **Finish**.

   A Connection dialog may appear. Here you specify the connection to use for this remote action.
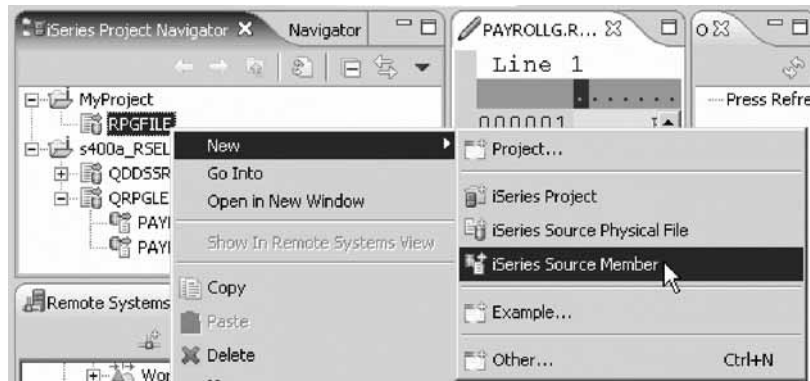

5. Click **OK**.

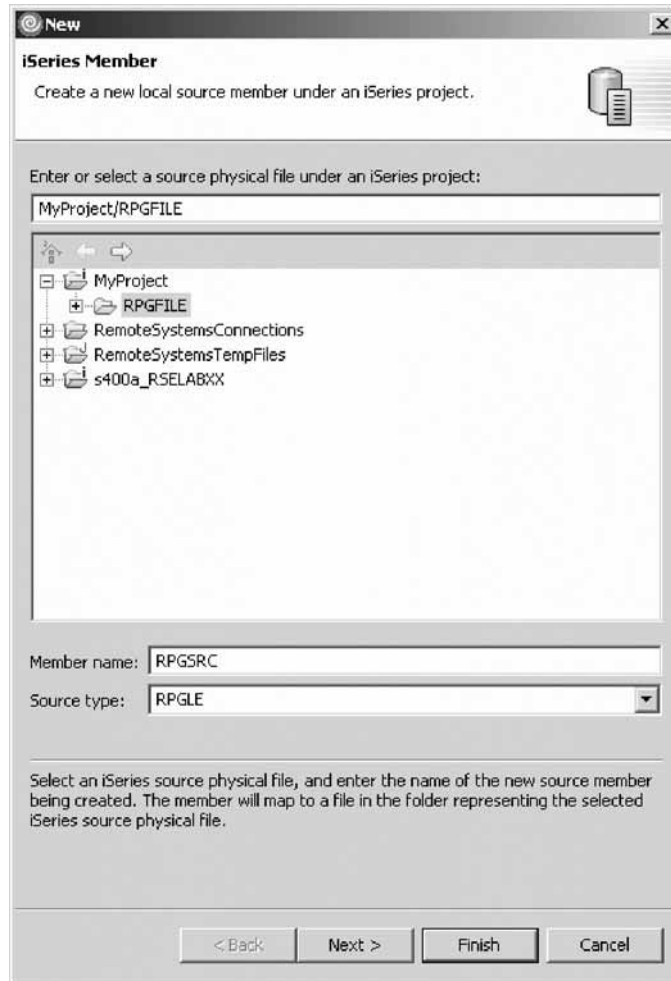   The file is added to your project.

Next add a member to the file that you just created.

6. Right-click your iSeries source physical file.

7. Click **New** then click **iSeries Source Member** on the pop-up menu.

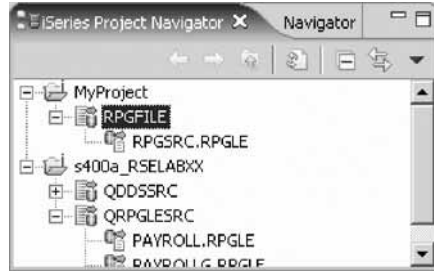The iSeries Member page opens.

8. Enter the name of the member in the **Member name** field, for example, RPGSRC.

9. Enter the source type in the **Source type** field, for example, RPGLE.

10. Click **Finish**.

A Connection dialog may appear. Here you specify the connection to use for this remote action.

11. Click **OK**.

The member is added to your project.

You can now double-click the member to open it in the editor.

You have populated the iSeries Project with a source physical file and a source member and now you are ready to begin "Exercise 3.4: Importing members into your project."

Exercise 3.4: Importing members into your project

Before you begin, you must complete "Exercise 3.3: Creating a source file and a source member" on page 24.

You can also import remote objects into an existing iSeries project. You can use the Import Remote Objects action or the Add to iSeries Project action.

To import remote objects:
1. Right-click your iSeries Project.
2. Click **Import Remote Objects** on the pop-up menu.
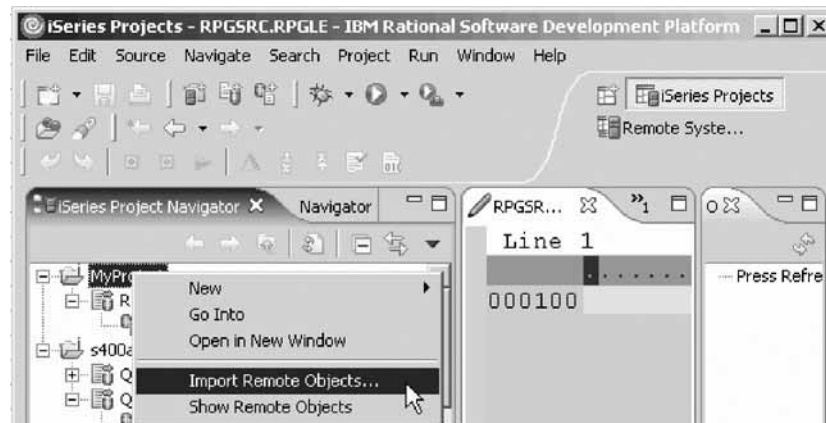
3. Expand *LIBL.
4. If you see a logon display, enter your user ID and password.
5. If you don't see RSELABxx in the library list, you will need to create a library filter.
   a. Expand Work with libraries.
   b. Type RSELABxx for the Library name.
   c. Click **OK**.
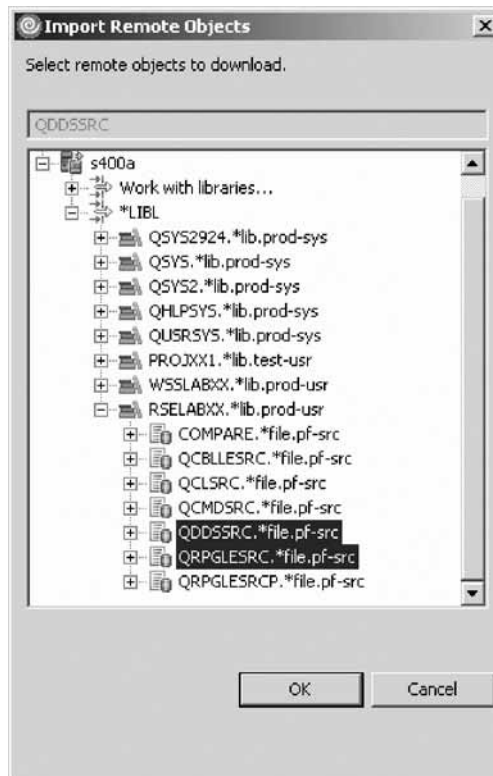
6. Expand the library that contains the source files that you want to import. In this case select RSELABxx.

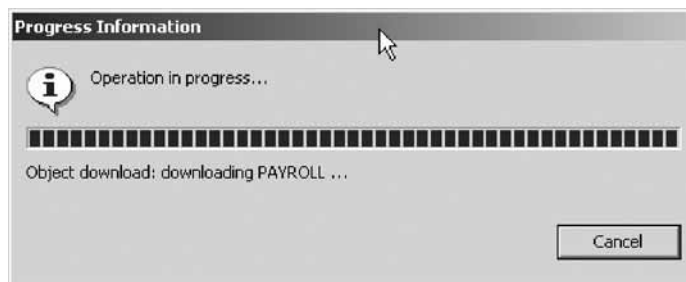7. Select the source physical files QDDSSRC and QRPGLESRC to be added to your project. You can use the Shift key to make multiple selections.
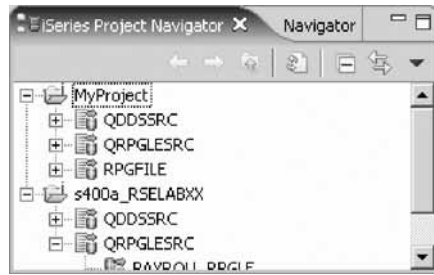
8. Click **OK**.

   A Connection dialog may appear. Here you specify the connection to use for this remote action.

9. Click **OK**.

   The selected files are downloaded.

The project now shows the copied files and they are ready for you to work with.

You have added existing members to your iSeries Project and now you are ready to begin "Exercise 3.5: Selecting the build style."

Exercise 3.5: Selecting the build style

Before you begin, you must complete "Exercise 3.4: Importing members into your project" on page 28.

So now that you know how to populate your iSeries Project, you could next disconnect and edit and verify the files on your workstation. Let's assume that you did just that but in this case you changed multiple members and that you want to build (compile) the full iSeries project.

Associated with each iSeries project is a Build Style which determines how source members in that project will be pushed and compiled on the host. A build style is a set of actions that provide push and build operations for a given iSeries Project.

There are three IBM-supplied build styles:

*NONE
> This build style contributes no actions and is useful if you want to disable the menu items associated with a build style. This style will always be available.

Command
> The command build style contributes the following actions to the Remote Actions menu:
> - **Push Changes** — Pushes any new or changed resources found in the selected project. An item is changed if it has been updated locally since the last time it was pushed, either by Push Selected or Push Changes. The first time it is done, all resources are uploaded to the associated library. Each subsequent upload is incremental.
> - **Push Selected** — This action uploads all selected resources and their children to their corresponding iSeries source files and members in the associated library. It creates any resources that do not exist. It will overwrite any changes that have been made on the server only if confirmed by the user. Use this action to force the contents of the associated library to match the workspace.
> - **Submit Build Command** — Submits the command found in this build style configuration to the server.

CL Program
> This style uploads the COMPILE.CLLE and BIND.CLLE members from the source file that you designated in the style configuration for this project. It

then compiles and runs them to build this project on the server. This build style contributes the following actions to the Remote Actions menu:

- **Push Changes** — Pushes any new or changed items found in the selected project. An item is changed if it has been updated locally since the last time it was pushed, either by Push Selected or Push Changes. The first time it is done, all resources are uploaded to the associated library. Each subsequent upload is incremental.
- **Push Selected** — This action uploads all selected resources and their children to their corresponding iSeries source files and members in the associated library. It creates any resources that do not exist. It will overwrite any changes that have been made on the server only if confirmed by the user. Use this action to force the contents of the associated library to match the workspace.
- **Submit Build** — Submits a job to compile and run the iSeries Project on the server using the COMPILE and BIND programs from the source file you specify.

Each iSeries Project can have only one active build style. This can be configured through the Properties page for that project (right-click on the iSeries project and select Properties).

By default, when a new iSeries project is created it has the CL Program Build Style associated with it. This can be changed by going to Window > Preferences > iSeries Projects > Build and Compile.

Once submitted, a build can be monitored through the iSeries Build Status View. This view can be set to update at regular intervals and monitors all iSeries Project build jobs that you have submitted. Also compile errors can be retrieved from that view by right-clicking on the submitted build and selecting the Retrieve Errors action.

You have selected a build style for your iSeries Project and now you are ready to begin "Exercise 3.6: Submitting a build."

Exercise 3.6: Submitting a build

Before you begin, you must complete "Exercise 3.5: Selecting the build style" on page 30.

Note: You must have a build style defined for your project before you can compile.

Besides compiling a single member like in "Exercise 2.4: Compiling a member" on page 16, you can compile the project in full. The actual compile behavior depends on the Build Style associated with the project. The default build style for iSeries Projects is the CL Program Build Style.

This build style will generate a COMPILE.CLLE member in a QCLSRC file that will contain a compile entry for each source member in the iSeries project.

You can verify or change the configuration of a build style. You can specify the name of the source file to hold the COMPILE.CLLE member and whether or not you want to automatically push changes and re-generate the COMPILE.CLLE member before a push.

To compile a project in full:

1. Right-click the iSeries Project and click **Remote Actions** > **Submit Build**.
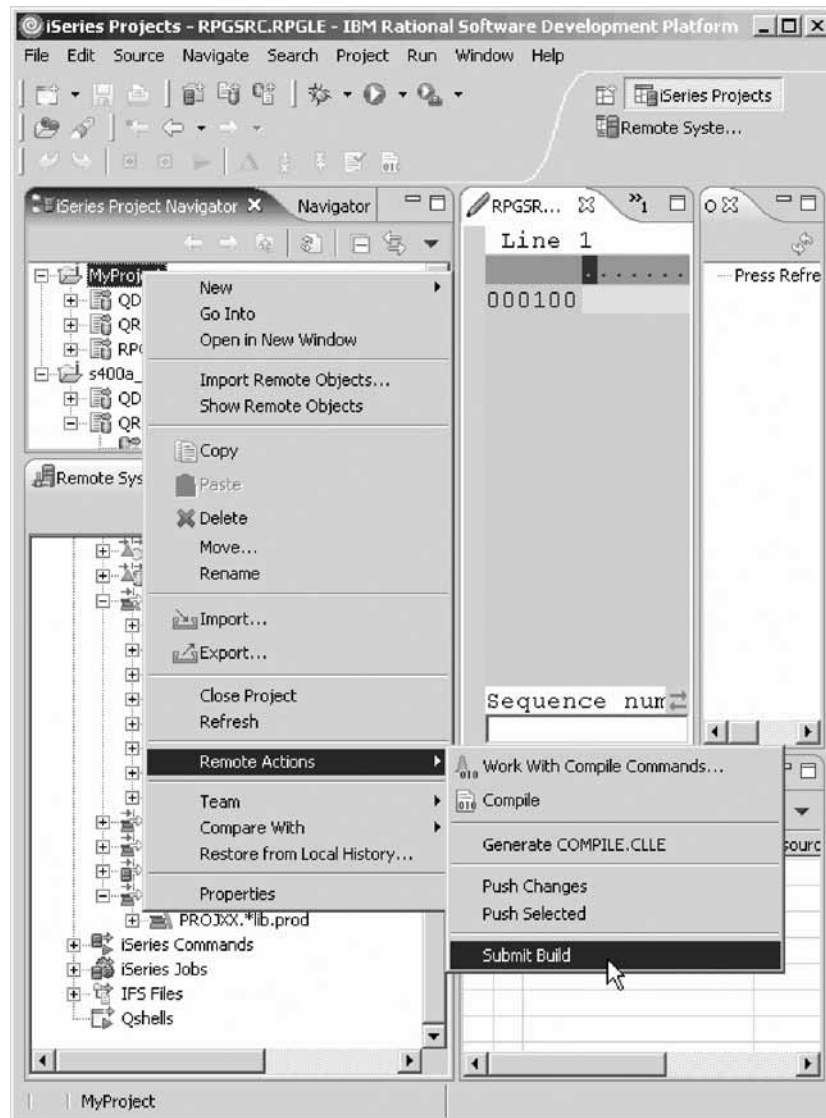
Now the connection will be activated and you may be prompted for a user ID and password if you disconnected.

2. Enter your user ID and password.

The changes are pushed to the iSeries. (For the first time all changes are pushed to the iSeries. Subsequent builds only push changed files.)
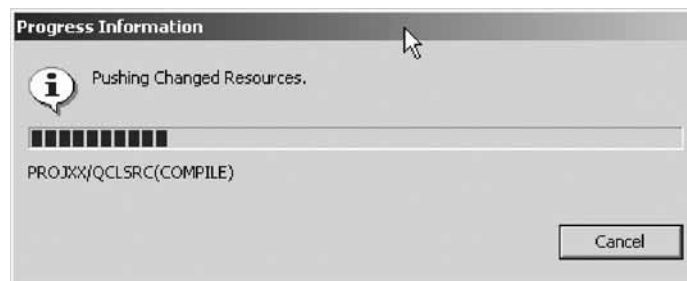
3. This action, with the default configuration, will perform all of the following:
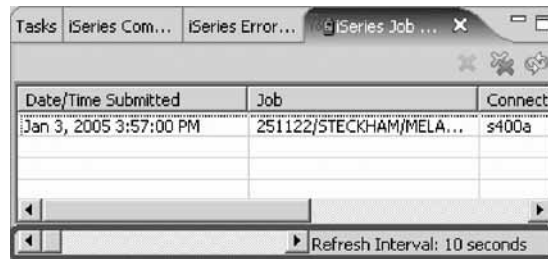
a. Generate the following COMPILE.CLLE member:

b. Display a Build Job entry in the iSeries Build Job Status View indicating that a build has been submitted. The entry will look like this:

You can change the Refresh Interval using the side arrow next to the **Refresh Interval** field shown above.

c. Push all project changes to the host, including the newly created COMPILE.CLLE member.

d. Compile COMPILE.CLLE into a program in the QTEMP library.

e. Run it against the source in the PROJxx project library.

f. If there where any errors during the submitted build (see PAYROLL), after the job completes, refresh the iSeries Job Status View, then right-click on the entry in the iSeries Job Status View, and select **Task Actions** > **Retrieve Errors**.

This will open the iSeries Error list and you will be able to double-click the error to open the corresponding member.

You have built your iSeries Project and now you are ready to begin "Exercise 3.7: Showing remote objects."
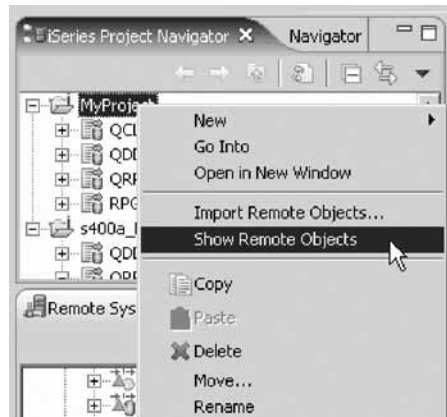
Exercise 3.7: Showing remote objects

Before you begin, you must complete "Exercise 3.6: Submitting a build" on page 31.

You can refresh your iSeries Project Navigator and look at all the remote objects.
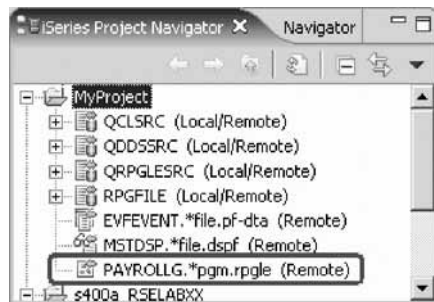
To show remote objects:

1. In the iSeries Project Navigator right-click the iSeries Project.

2. Click **Show Remote Objects** on the pop-up menu.

3. Your iSeries Project Navigator will now look like this:

You have seen the PAYROLLG program object in the iSeries Project Navigator and now you are ready to begin "Exercise 3.8: Filtering out remote objects."

Exercise 3.8: Filtering out remote objects

Before you begin, you must complete "Exercise 3.7: Showing remote objects" on page 33.

You can filter the objects to remove the remote only objects from the view.

To filter out remote objects:
1. Go to the iSeries Project Navigator toolbar.
2. Click the drop-down arrow.
3. Click **Show** > **Remote Objects** on the drop-down menu.
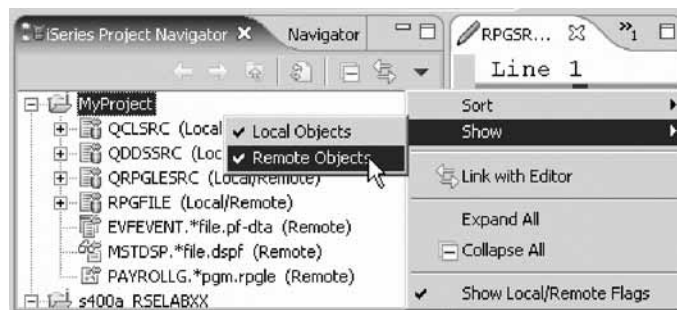
This would remove all objects that are only remote from the view.
4. Expand QCLSRC, QDDSSRC, QRPGLESRC and RPGFILE

Now you see only Local/Remote objects.
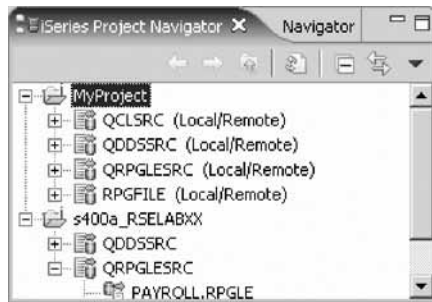
Now you want to remove the Local/Remote Flags from the view.

5. Go to the iSeries Project Navigator toolbar.
6. Click the drop-down arrow.
7. Click **Show Local/Remote Flags** on the pop-up menu.
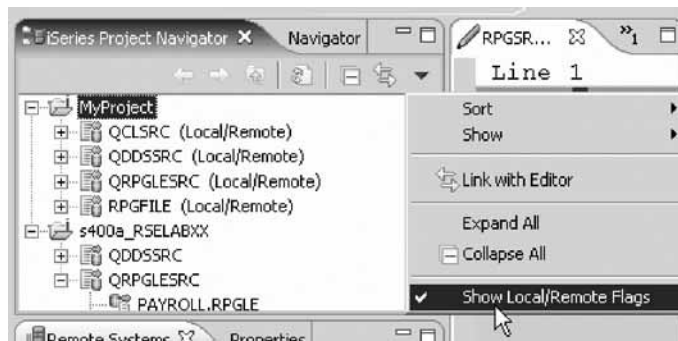
Now you see your objects without the flags to indicate whether the object is local, remote or both local and remote.
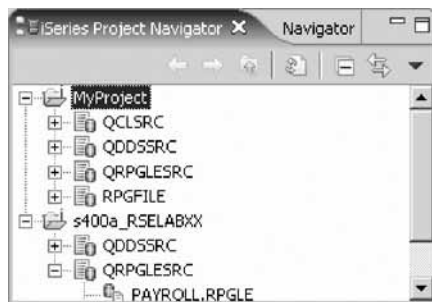
You have filtered the objects to remove the remote only objects from the iSeries Project Navigator and removed the local/remote flags from the view and now you are ready to begin "Exercise 3.9: Changing PAYROLLG again and submitting another build."

Exercise 3.9: Changing PAYROLLG again and submitting another build

Before you begin, you must complete "Exercise 3.8: Filtering out remote objects" on page 34.

Now let's change PAYROLLG again and the submit another build. You will see that only the changed file is pushed to the iSeries.

To change PAYROLLG and submit another build:

1. Double-click PAYROLLG in QRPGLESRC.
2. In the Editor add a comment line to the source.
3. Save the file and close the Editor.
4. Right-click the iSeries Project MyProject and click **Remote Actions** > **Submit Build** on the pop-up menu.
5. Only the member and COMPILE.CLLE are pushed to the iSeries.
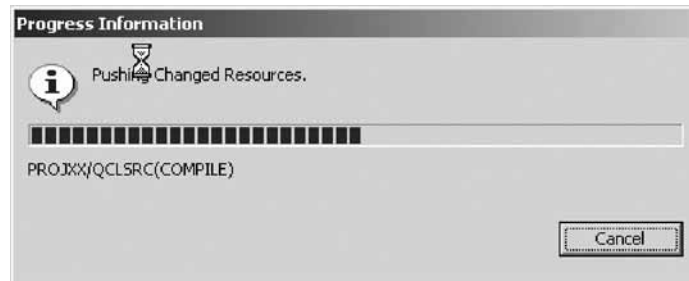

You have pushed only the changed file to the iSeries.

Module recap

You have completed Chapter 3, "Module 3. Creating and Building an iSeries Project," on page 19. You have learned how to:

- Create a personal development library (single developer sandbox)
- Create an empty iSeries Project
- Populate the empty iSeries Project
- Select a build style for the iSeries Project
- Build the iSeries Project
- View all the remote objects in the iSeries Project Navigator
- Filter the objects to remove the remote only objects from the iSeries Project Navigator
- Remove the Local/Remote Flags from the iSeries Project Navigator
- Push only a changed file to the iSeries

Now that you know how to build source you could start an interactive connection and then run your program interactively. You could also start the iSeries Integrated Debugger to debug your application. You can do the same run and debug tasks that are outlined in the tutorial named Maintain an ILE RPG application using the Remote System Explorer.

Finish the tutorial by reviewing the materials in Chapter 4, "Summary," on page 37.

Chapter 4. Summary

This tutorial has shown you how to use the iSeries Projects perspective in a typical development cycle for offline application development. After providing you information about the perspective, the tutorial outlined the development cycle working offline with an iSeries Project, then elaborated on each step in maintaining an ILE RPG project-based application.

If you have completed all the modules in this tutorial, you should now be able to:
- Start the product, configure and create a connection to an iSeries system
- Make source available for offline or disconnected development
- Edit and verify source in disconnected mode
- Compile a changed member
- Create a new iSeries project and populate with new source physical files and members
- Show remote objects or local objects or both
- Submit a build for the entire iSeries project

More information

For more information on the product and the iSeries Projects, see http://ibm.com/software/adwtools/iseries.

Appendix. Notices

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

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario
Canada
L6G 1C7

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

COPYRIGHT LICENSE:

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1992, 2005. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

ClearCase
IBM
iSeries
Rational
WebSphere

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT®, Win32, Win32s and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX® is a registered trademark of The Open Group.

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

IBM ®

Printed in USA