



IBM Global Services - IBM eServer iSeries

A34

WDS*c*: iSeries Projects
Modern Tools for RPG and COBOL Developers
Don Yantzi



Miami, FL

Oct 17-21, 2005

why "I"? it's simple.

© 2005 IBM Corporation



Agenda

- iSeries Projects
 - What are they?
 - How do I use them?
 - Adding source files and members
 - Building your project

- Remote System Explorer
 - Using RSE features with iSeries projects

- Demo



RSE and iSeries Projects

- Remote System Explorer (RSE)
 - Designed to be familiar to PDE / SEU programmer
 - Remote edit, verify, compile, run / debug
 - Source members are still kept on the iSeries
 - Use existing OS/400 source configuration management (SCM) providers

- iSeries Projects
 - Designed to be similar to development of Web, Java, and XML in the workbench
 - Source is kept local on the PC in the workspace
 - Local edit and verify then push changes and build on remote system
 - Use any workbench based SCM provider
 - iSeries vendors or CVS, Rational ClearCase, PVCS, ...



Why use iSeries projects?

- Use for disconnected development
 - If you want to work on source while you're disconnected from a iSeries server
 - On the train
 - At home on the weekend (sorry)
- Use for structured development
 - Organize development into "projects", just like you would for Java, J2EE or Web
 - A project holds the required source and you build the project
 - Easy to develop and maintain versus having source in various locations
 - Version control (not included in WDS*c*)

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.



Why use iSeries projects?

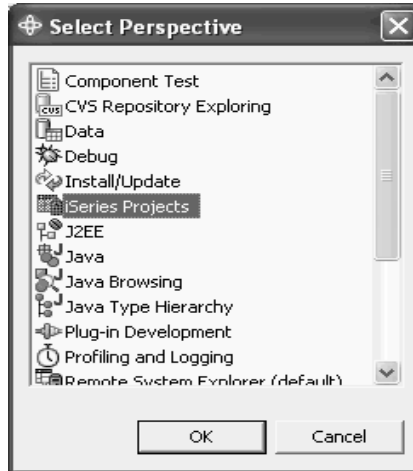
- RSE features such as:
 - ❖ Advanced editing capabilities in Remote Systems LPEX editor
 - ❖ Outline view for RPG and COBOL
 - ❖ Content Assist
 - ❖ Syntax checking
 - ❖ Prompting
 - ❖ Showing indentation
 - ❖ Verifying source
 - ❖ Integrated, online help for tools and languages

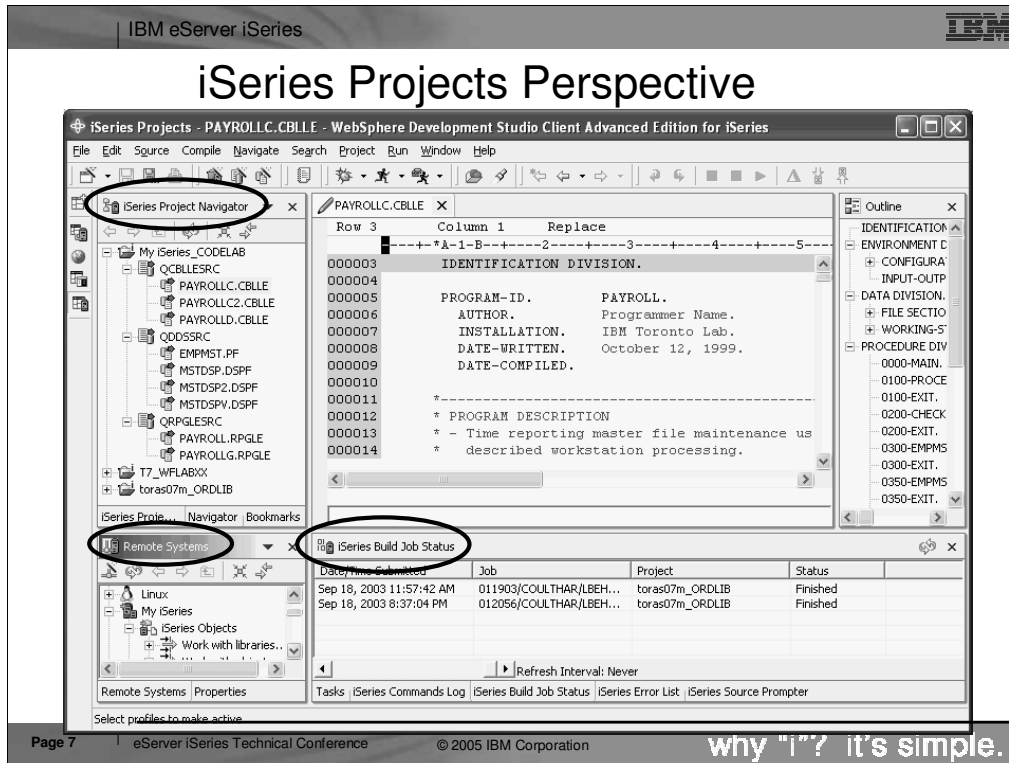
All are available!



iSeries Projects

Window->Open Perspective->Other...





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 Projects

- A dedicated perspective
 - Provides a collection of views and editor
 - iSeries Project Navigator view
- Typical workbench project
 - Contains folders and files, can be shared by a team
 - Has its own tools and perspectives
- But Also
 - Holds source destined to be compiled on iSeries
 - User “Pushes” source periodically to the host, and builds

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.



iSeries Project Library, files, members

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

- iSeries Source Physical File
 - Source physical files represented as folders in the project
 - CCSID, record length, IGC data and description

- iSeries Member
 - Members stored as files within “source” folders
 - Uses format: memberName.membertype
 - ORDRENT. RPGLE

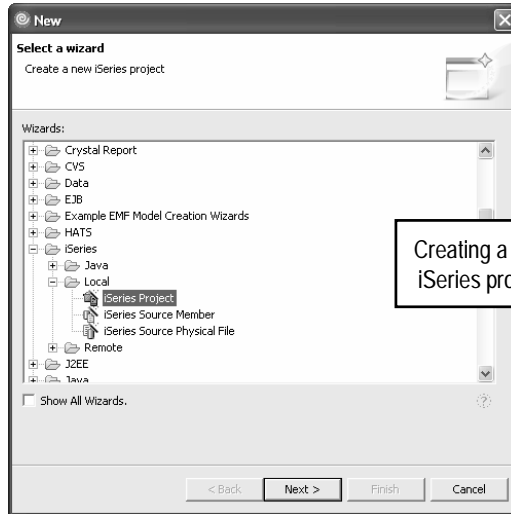


Setting Up Your iSeries Project

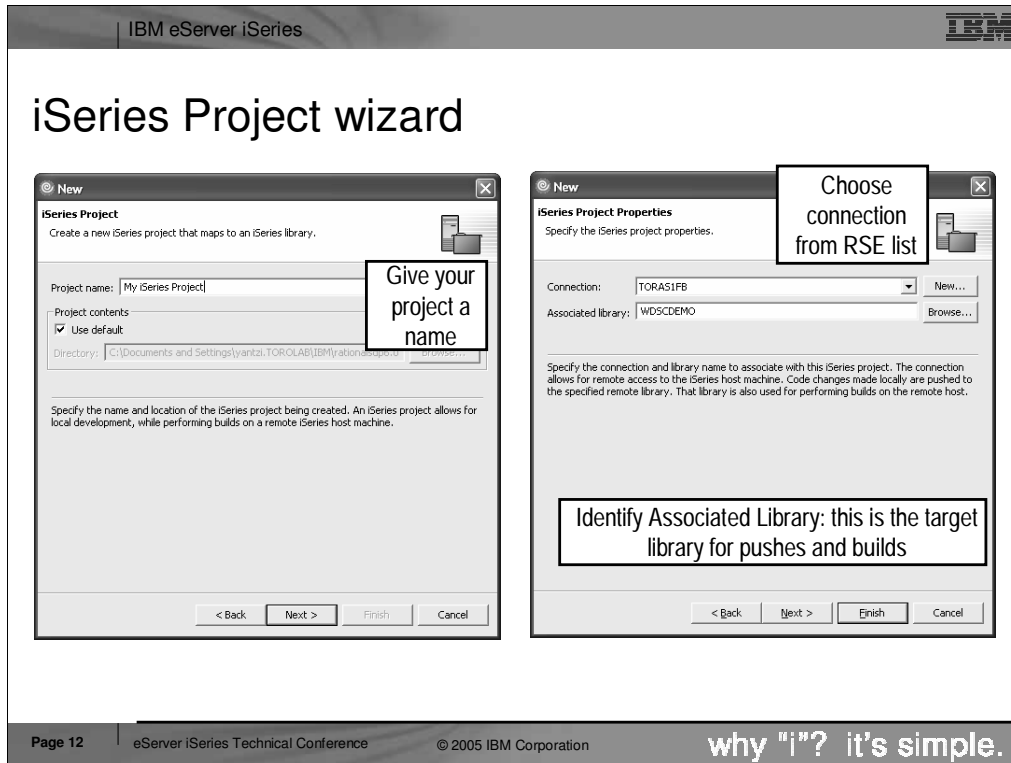
- Multiple ways to setup an iSeries project, source physical files and members
 - Using the workbench “New” wizards (under iSeries > Local)
 - Using the “Add to Project” actions from the iSeries project
 - Using the “Create iSeries Project” and “Add To iSeries Project” actions from the RSE



New iSeries Project



Creating a new
iSeries project



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.

IBM eServer iSeries

iSeries Project Wizard (next page)

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

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

The screenshot displays three overlapping windows from the iSeries Project Wizard. The 'New' window is in the foreground, showing 'iSeries Project Properties' with a list of build styles: 'CL Program', 'Command', and '*NONE'. The 'CL Program' style is selected, and a 'Configure...' button is visible. Two other windows are partially visible behind it: 'CL Program Build Style Configuration' and 'Command Build Style Configuration'. The 'CL Program Build Style Configuration' window has a 'Build Source File' field containing 'QCLSRC' and two checked options: 'Automatically generate COMPILE.CCLE prior to each build' and 'Automatically push all changed members prior to build'. The 'Command Build Style Configuration' window has a 'Build Command' field and an unchecked option: 'Automatically push all changed members prior to build'. Arrows point from the 'Configure...' button in the 'New' window to the 'CL Program Build Style Configuration' window, and from a text box to the 'Command Build Style Configuration' window.

Page 13 | eServer iSeries Technical Conference | © 2005 IBM Corporation | why "i"? it's simple.

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

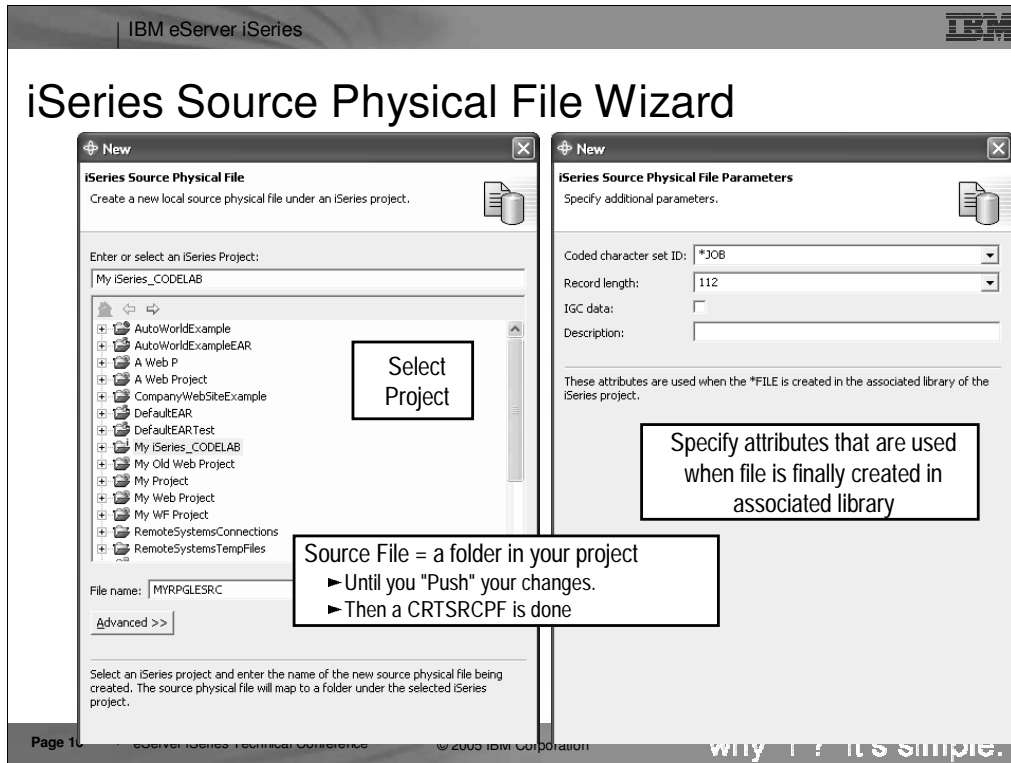


Build Styles

CL Program — Automatically generate COMPILE.CLLE in a source folder of your choice. This will be used to do the build on the host.

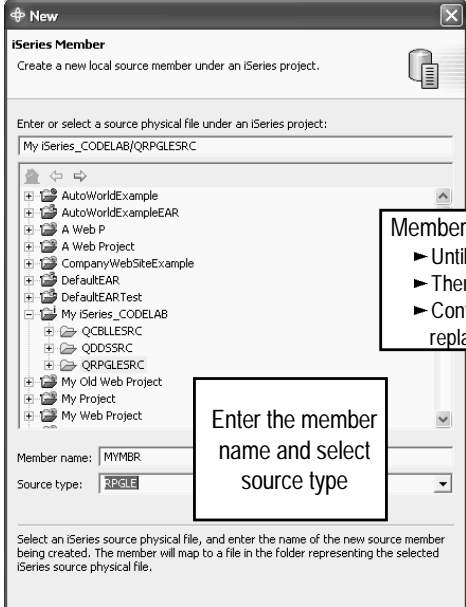
Command — Execute user specified command to do the build

***NONE** — No build style. Project can not be built on the host.



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 result 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.

Page 17 | eServer iSeries Technical Conference | © 2005 IBM Corporation | why "i"? it's simple.

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?

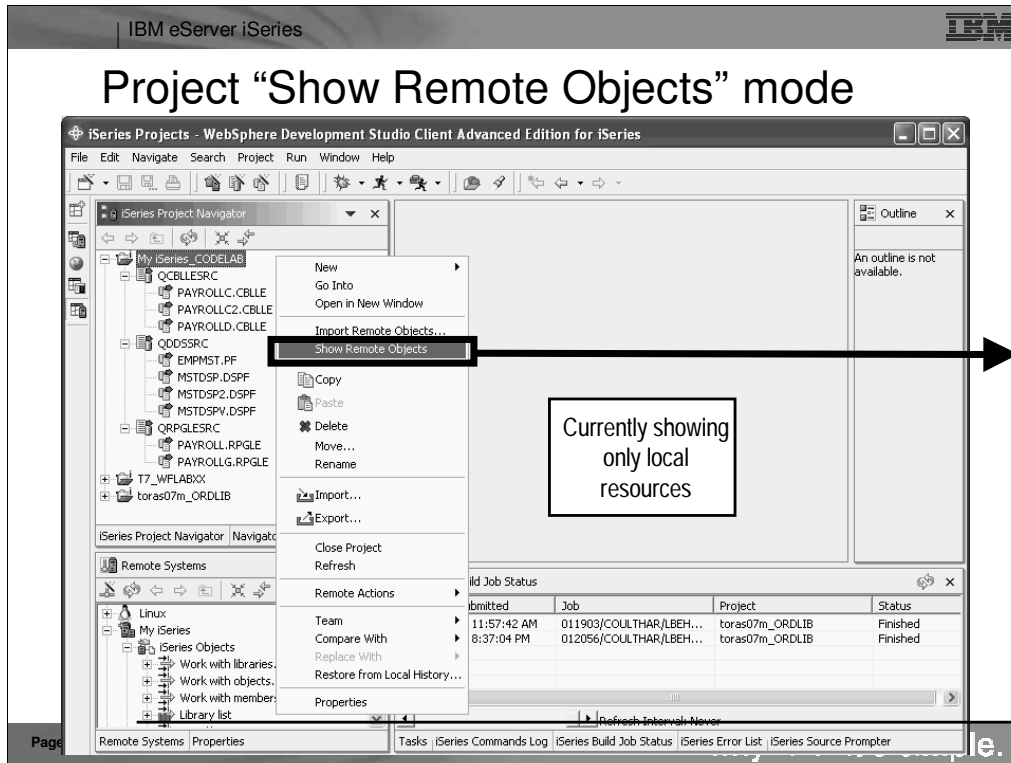
- Use Import or Add to Project action to copy them
- Source Files become folders locally
- Members added are copied to that folder
- Developer works on local copy
- Full power of workbench tools at your disposal

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

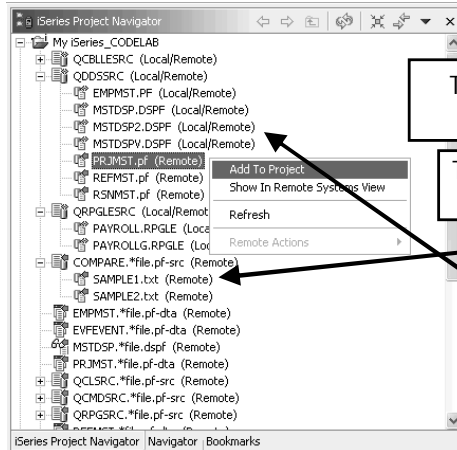
- Edits file
- Pushes changes using "Push" action
- Does a build using "Build" action
- Shares with team using "Synchronize" action



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**

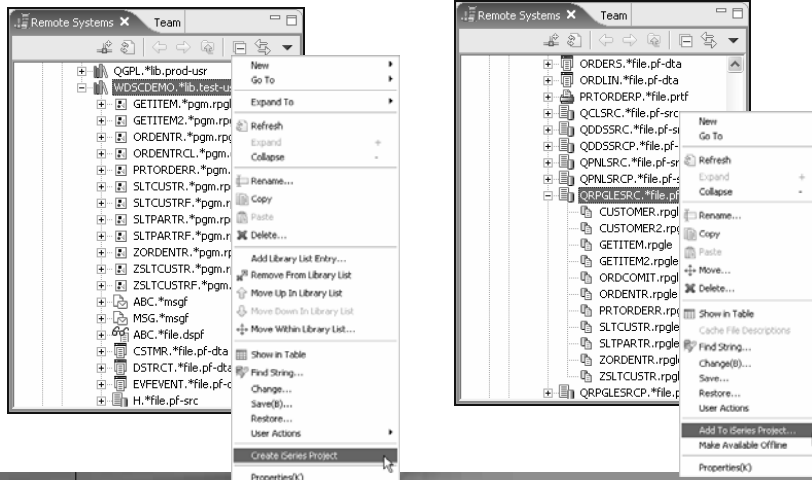
To do actions against remote objects, use **Show In Remote Systems View**

(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

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.

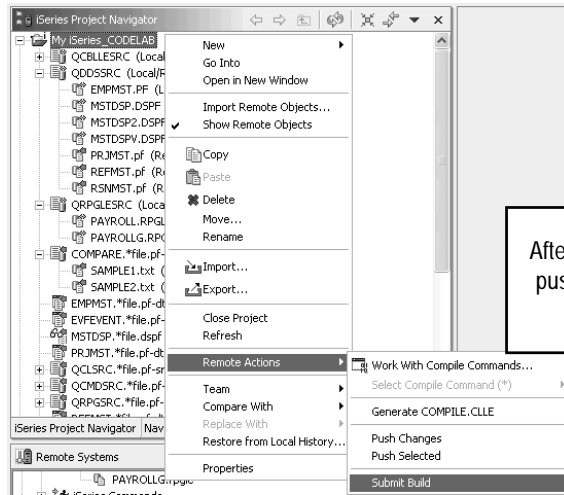
Populating iSeries Project from Remote Systems

- Can right click on Library and **create** an iSeries Project. Also can right click source file or member and **add** to an iSeries project.

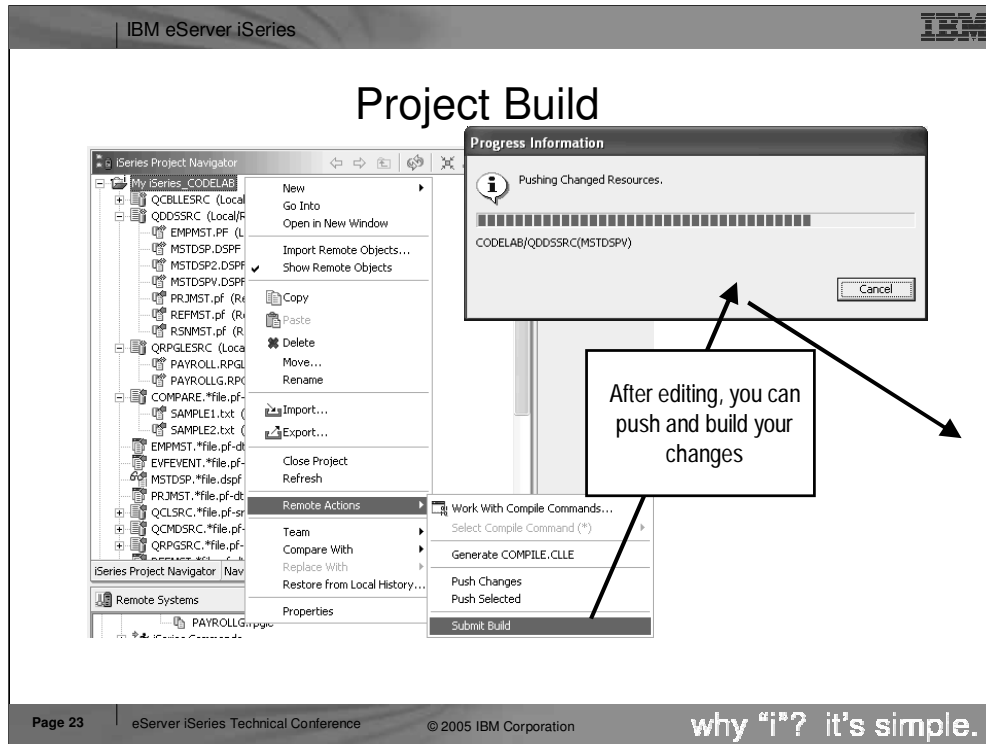




Project Build



After editing, you can push and build your changes



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 COMPILER.CLLE.

iSeries Build Job Status

The screenshot shows the 'iSeries Job Status' window with a table of jobs and a context menu. Callouts provide the following information:

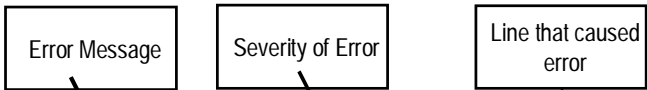
- Job where build is running:** Points to the 'Job' column in the table.
- Status of the build:** Points to the 'Status' column in the table.
- Refresh interval determines how frequently status is updated:** Points to the 'Refresh Interval: 30 seconds' control at the bottom of the window.
- Retrieve errors from build to the iSeries Error List:** Points to the 'Retrieve Errors' option in the 'Task Actions' menu.

Date/Time Submitted	Job	Connection	Status	Task
Sep 14, 2005 2:00:00 PM	389165/YANTZI/YANTZI	TORAS1FB	*OUTQ	TORAS1FB_WDSCDEMO Build
Sep 14, 2005 2:00:00 PM	389168/YANTZI/YANTZI	TORAS1FB	*ACTIVE	TORAS1FB_WDSCDEMO Build

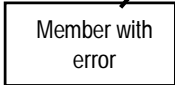
- Remove
- End
- Hold
- Release
- Display job log
- User Actions
- Debug As
- Properties
- Task Actions
 - Cancel
 - Retrieve Errors



iSeries Error List



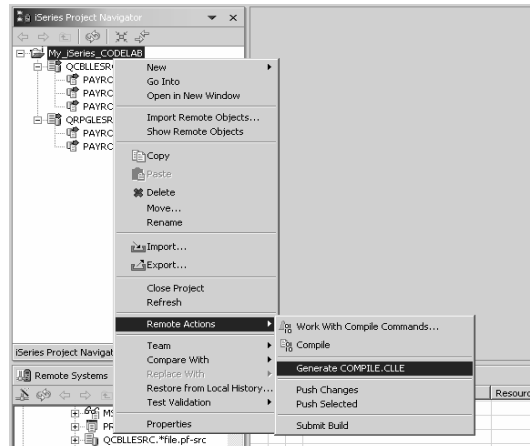
ID	Message	Se...	Line	Location	Connection
■ RN59308	Compilation stopped. Severity 30 errors found in pr...	50	0	WDSCDEMO/QRPGLES...	TORAS1FB
□ RNF7503	Expression contains an operand that is not defined.	30	24	WDSCDEMO/QRPGLES...	TORAS1FB
□ RNF5014	Operation code is not valid; specification is ignored.	30	22	WDSCDEMO/QRPGLES...	TORAS1FB
□ RNF7030	The name or indicator FEEDBACK is not defined.	30	24	WDSCDEMO/QRPGLES...	TORAS1FB
■ RNF7031	The name or indicator IID is not referenced.	00	1	WDSCDEMO/QRPGLES...	TORAS1FB
■ RNF7031	The name or indicator INAME is not referenced.	00	2	WDSCDEMO/QRPGLES...	TORAS1FB
■ RNF7031	The name or indicator IPRICE is not referenced.	00	3	WDSCDEMO/QRPGLES...	TORAS1FB
■ RNF7031	The name or indicator IDATA is not referenced.	00	4	WDSCDEMO/QRPGLES...	TORAS1FB
■ CPD0791	No labels used in program.	00	13	WDSCDEMO/QLSRCC(...	TORAS1FB
■ RME7066	Record-Format-ITRCD not used for input or output	00	6	WDSCDEMO/QRPGLES...	TORAS1FB

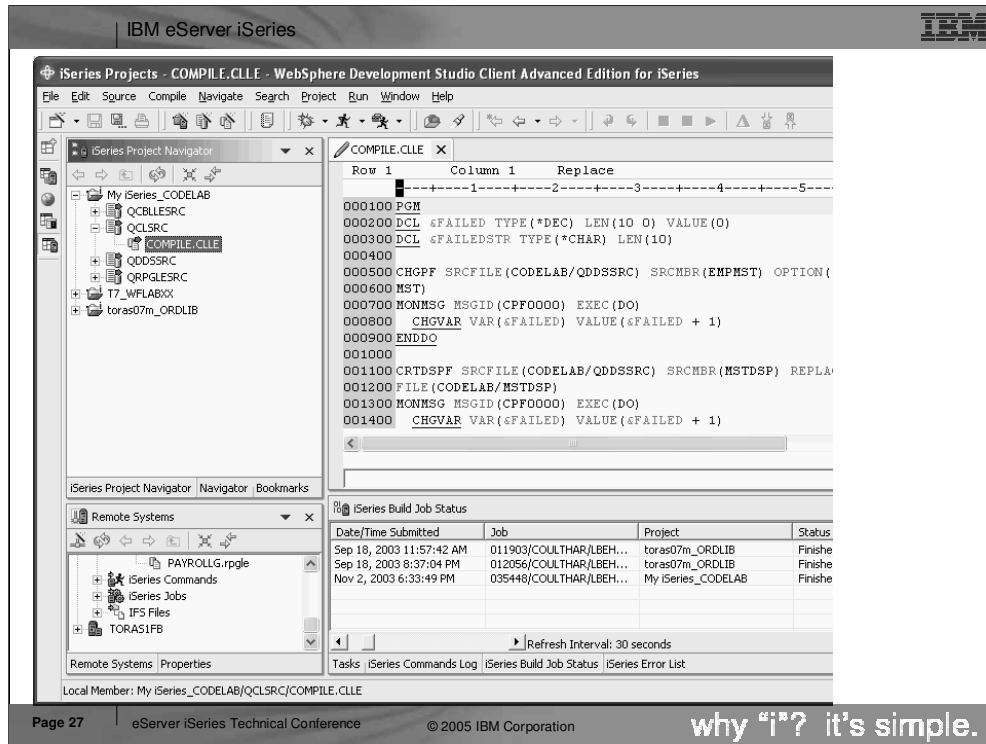




COMPILE.CLLE

- Can generate COMPILE.CLLE for project without building





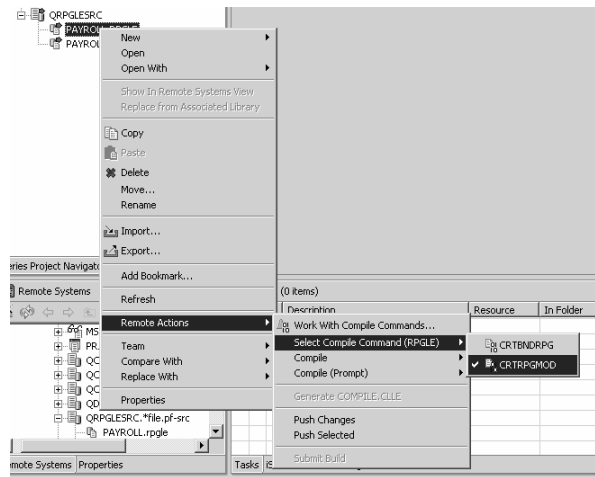
An example of what is generated for the build. This is a CL member. You can affect what compile command is used for each member type.

COMPILE.CLLE

- To change what is generated for COMPILE.CLLE:

–Right click on a member and select a different compile command

–New compile commands can be added or existing compile commands changed through Work With Compile Commands...





Work With Compile Commands

Work With Compile Commands

Parent profile: kmunir02

Member type: RPGLE

Compile Commands:

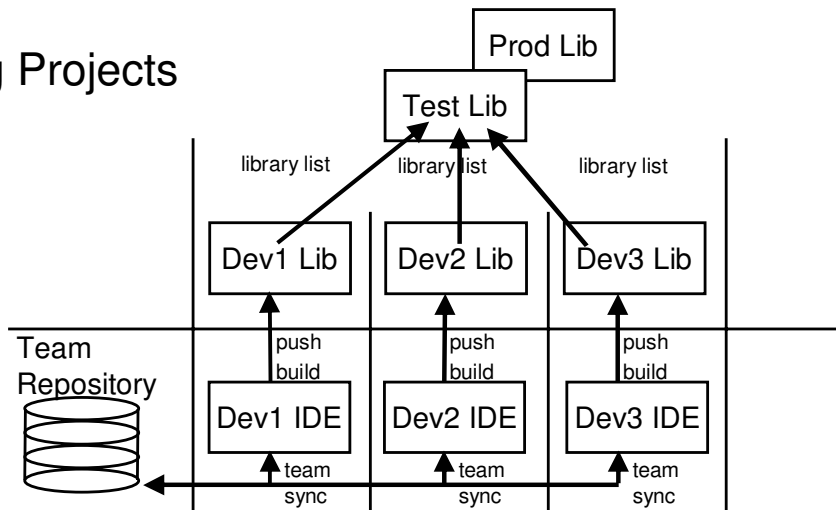
- New command
- CRTBNDRPG
- CRTRPGMOD**

Selected compile command:

Label: CRTRPGMOD

Command: CRTRPGMOD MODULE(&O/&N) SRCFILE(&L/&F) SRCMBR(&M) REPLACE(&R) OPTION(*EVENTF) DBGVIEW(*SOURCE)

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.



Brief Overview of RSE capabilities in iSeries projects

- ❖ Advanced editing capabilities of Remote Systems LPEX Editor
- ❖ Outline view for RPG and COBOL
- ❖ Content Assist
- ❖ Syntax checking
- ❖ Prompting
- ❖ Showing indentation
- ❖ Verifying source



Demo



Additional Information

- **homepage:**
<http://ibm.com/software/awdtools/series>
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, Don Yantzi and Kushal Munir

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.