Collaborative Development using IBM Rational Team Concert

Work Items, Software Configuration Management, and Team Collaboration Platform for COBOL, Java, PL/I, C/C++, and Assembler Development

Student Exercises

Liam Doherty (dohertl@au1.ibm.com)
Rosalind Radcliffe (rradclif@us.ibm.com)

Visit Jazz.net at:
http://jazz.net/

Fourth Edition (August 2012)

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 customer's ability to evaluate and integrate them into the customer's 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 environments do so at their own risk.

Table of Contents

Overview

This lab is for use with Rational Developer for z (RDz) integrated with Rational Team Concert (RTC). RDz is an integrated development environment (IDE) that consists of a workbench and a common set of tools to help you develop and maintain Assembler, COBOL, C/C++, and PL/I. RTC provides project tracking, planning, build, collaboration and source code management. RDz and RTC can be installed separately or together in an integrated environment where both RDz and RTC share the same eclipse client. The lab will demonstrate how to use RTC to track progress using Work Items and Iteration Plans. It will show how to manage source code with the Jazz Source Code repository. This lab is split into parts to allow you to work on the pieces you most want to learn about.

Depending on what your interest is in RTC you may want to look at different components. Chose the modules you wish to do based on your interests.

Basic RTC knowledge

If you are just after some basic knowledge of RTC, maybe you are a manager and want to see what RTC can do through the use of work items, then this is a good place to start.

1. *Module 3.* *Planning your work*
2. *Module 4.* *Work Item queries*
3. *Module 5.* *Web user interface*

RTC Planning and Administration

If you are likely to be setting up RTC, creating users or as a project leader, creating plans then these modules will be best suited for you.

1. *Module 1.* *Using the Admin Console*
2. *Module 2.* *Setting up the team*
3. *Module 3.* *Planning your work*

RTC Mainframe code development using Eclipse and RDz

If you are a developer but are more used to working in a graphical IDE, then these modules will show you some of the source code capabilities of RTC.

1. *Module 6.* *Source Code Management (SCM)*

2. *Module 7. Enterprise Build (SCM)*
3. *Module 9. Optional: Exploring Changes and Traceability*

RTC Mainframe code development using the RTC ISPF Client

Still love the ISPF editor? Then have a look at how source code management through RTC is still achieved through an ISPF interface, so you can still use the editor you love.

1. *Module 8. Source code management with the ISPF Client*
2. *Module 7. Enterprise Build (SCM) – To understand how build definitions are defined*

If you get through a set of modules then take a look at a different one, or use this lab at your site and install RTC as a trail.

Once you have installed the RTC server and build toolkit you can get the sample data used in this lab, and the instructions for setting it up here: https://jazz.net/wiki/bin/view/Main/DependencyBuildScenarioV4

Module 1   Using the Admin Console

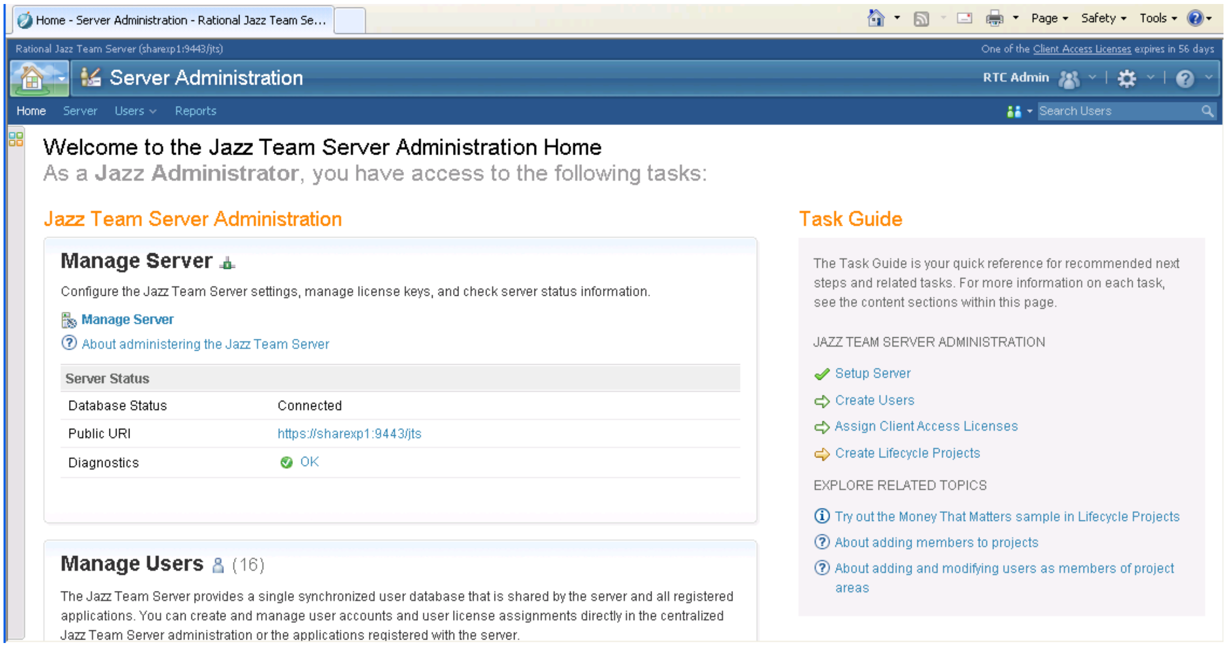
Lab Scenario

The admin console provides the ability to configure the server, project areas, and users. In this scenario you will learn about the different parts of the admin console, as well as create a new user for your use during the lab.

1.1. Admin Console

a) It may be necessary to start the Jazz team server. Check on your desktop for a command prompt window with a title of “Tomcat”. If there is one running then you do not need to do anything. If there isn’t one running then double click on the “**Start Jazz Team Server**” icon on your desktop. The wait until you see the “**INFO: Server startup**” message in the command prompt box.

b) The Web admin console is used to perform crucial administrative functions that you can’t perform elsewhere. Open Internet Explorer to https://sharexp1:9443/jts/admin (this is also bookmarked for you as “**Home – Server Administration - Rational Jazz Team Server**”) and login with user id **rtcadmin** and the password **rtcadmin** to access the admin console. Prior to getting the login screen you may get a certificate error screen. If you get this then click on the option to “**Continue to this website (not recommended)**”. The first screen you will receive is the Jazz Team Server Administration home, where you can choose between Server Administration and Application Administration.



Home - Server Administration - Rational Jazz Team Se...
Page
Safety
Tools
Rational Jazz Team Server (sharexp1:9443/jts)
One of the Client Access Licenses expires in 56 days
Server Administration
RTC Admin
Home
Server
Users
Reports
Search Users

Welcome to the Jazz Team Server Administration Home

As a **Jazz Administrator**, you have access to the following tasks:

Jazz Team Server Administration

Manage Server

Configure the Jazz Team Server settings, manage license keys, and check server status information.

Manage Server

About administering the Jazz Team Server

Server Status	
Database Status	Connected
Public URI	https://sharexp1:9443/jts
Diagnostics	✓ OK

Manage Users (16)

The Jazz Team Server provides a single synchronized user database that is shared by the server and all registered applications. You can create and manage user accounts and user license assignments directly in the centralized Jazz Team Server administration or the applications registered with the server.

Task Guide

The Task Guide is your quick reference for recommended next steps and related tasks. For more information on each task, see the content sections within this page.

JAZZ TEAM SERVER ADMINISTRATION

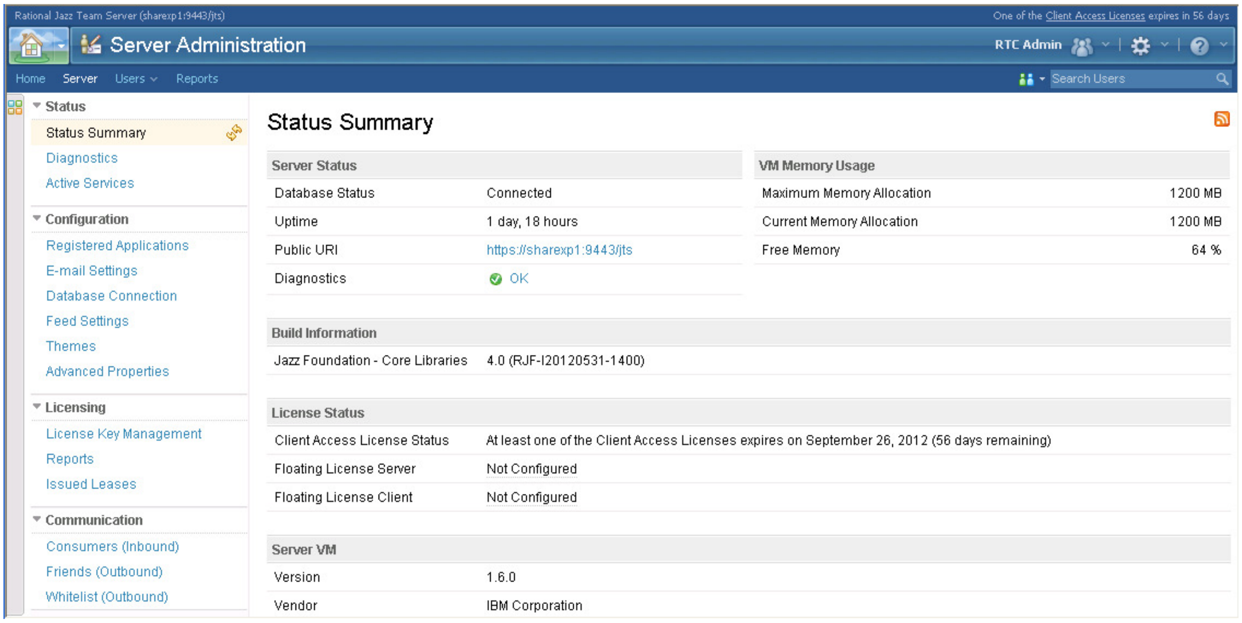
- ✓ Setup Server
- Create Users
- Assign Client Access Licenses
- Create Lifecycle Projects

EXPLORE RELATED TOPICS

- Try out the Money That Matters sample in Lifecycle Projects
- About adding members to projects
- About adding and modifying users as members of project areas

c) Click on **Jazz Team Server – Manage Server**. On the administration screen shown next the main links or pages are located across the top. The submenu is on the left. The Status

Summary is an overview of general server statistics and is a good starting point for server troubleshooting. It should look like the screenshot below once you are logged in.



Status Summary

Server Status	
Database Status	Connected
Uptime	1 day, 18 hours
Public URI	https://sharexp1:9443/jts
Diagnostics	✓ OK

VM Memory Usage	
Maximum Memory Allocation	1200 MB
Current Memory Allocation	1200 MB
Free Memory	64 %

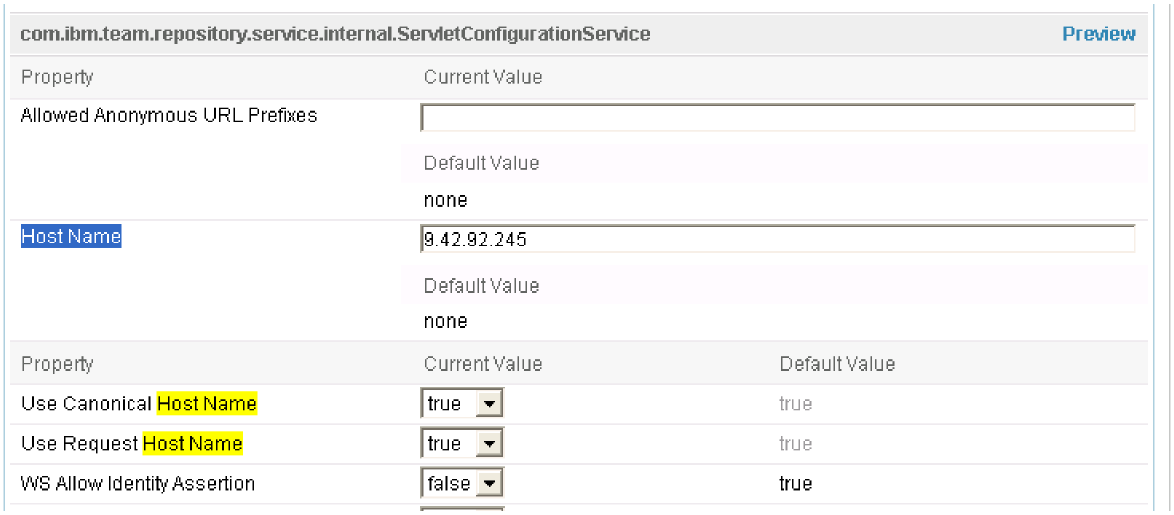
Build Information	
Jazz Foundation - Core Libraries	4.0 (RJF-I20120531-1400)

License Status	
Client Access License Status	At least one of the Client Access Licenses expires on September 26, 2012 (56 days remaining)
Floating License Server	Not Configured
Floating License Client	Not Configured

Server VM	
Version	1.6.0
Vendor	IBM Corporation

d) Explore the **Configuration** categories of **Registered Applications**, **Email Settings, Database Connection, Feed Settings, Themes** and **Advanced Properties.** We will make one change in the **Advanced Properties** find the host name field and click edit, add the *ip address* of your local image to the hostname, so that it will be reachable from the z/OS machine when doing build. You will need to return to the top of the page and click **Save** to save your changes.

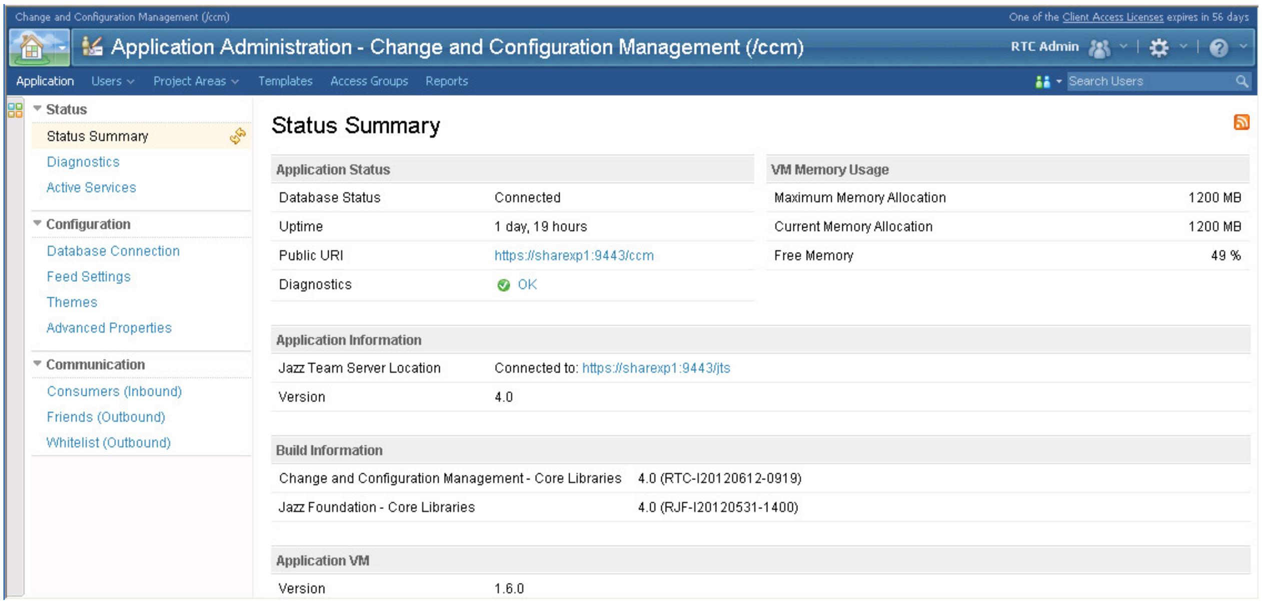
Note : To find the Host Name field press ctrl-F to bring up a find box, then enter Host Name in the box. This will quickly locate the Host Name field. To find your *ip address*, open a command prompt and type **ipconfig**. The *ip address* should be listed there.



com.ibm.team.repository.service.internal.ServletConfigurationService

Property	Current Value	
Allowed Anonymous URL Prefixes		
	Default Value	
	none	
Host Name	9.42.92.245	
	Default Value	
	none	
Property	Current Value	Default Value
Use Canonical Host Name	true	true
Use Request Host Name	true	true
WS Allow Identity Assertion	false	true

e) Let’s now look at application administration. From RTC version 3.0 onwards the change and configuration component (ccm) now runs as an application that depends on the Jazz Team Server (jts). Other applications, such as Rational Quality Manager (qm) also run as applications under the Jazz team Server. Click on the **Home** link at the top of the page to take you back to the Jazz Team Server Administration Home. In the **Application Administration** section click on the **Change and Configuration Management (*/ccm) – Application*** link and you will now be presented with the Status Summary page for the ccm application. Here you can see the version of both the ccm application and the Jazz foundation core libraries.



Status Summary

Application Status	
Database Status	Connected
Uptime	1 day, 19 hours
Public URI	https://sharexp1:9443/ccm
Diagnostics	✓ OK

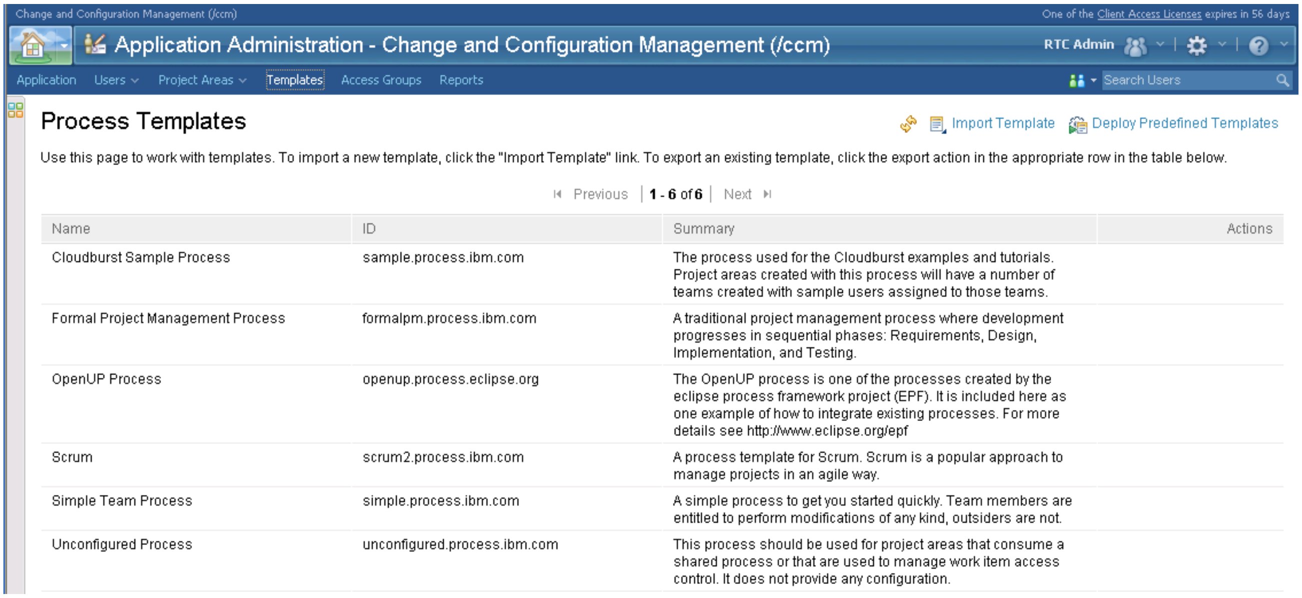
VM Memory Usage	
Maximum Memory Allocation	1200 MB
Current Memory Allocation	1200 MB
Free Memory	49 %

Application Information	
Jazz Team Server Location	Connected to: https://sharexp1:9443/jts
Version	4.0

Build Information	
Change and Configuration Management - Core Libraries	4.0 (RTC-I20120612-0919)
Jazz Foundation - Core Libraries	4.0 (RJF-I20120531-1400)

Application VM	
Version	1.6.0

f) On the page links across the top, click the **Templates** link. This page is where you can deploy predefined process templates (this has already been done for you), import templates, or manage existing templates.

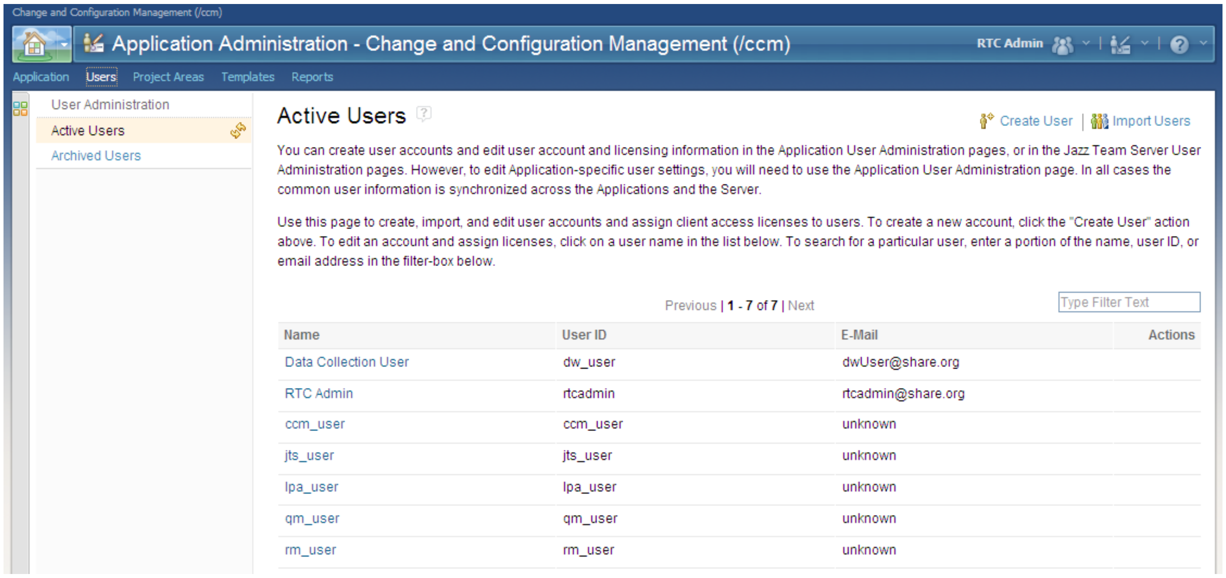

Change and Configuration Management (/ccm)
One of the Client Access Licenses expires in 56 days
Application Administration - Change and Configuration Management (/ccm)
RTC Admin
Application
Users
Project Areas
Templates
Access Groups
Reports
Search Users
Import Template
Deploy Predefined Templates
Previous
1 - 6 of 6
Next

Process Templates

Use this page to work with templates. To import a new template, click the "Import Template" link. To export an existing template, click the export action in the appropriate row in the table below.

Name	ID	Summary	Actions
Cloudburst Sample Process	sample.process.ibm.com	The process used for the Cloudburst examples and tutorials. Project areas created with this process will have a number of teams created with sample users assigned to those teams.	
Formal Project Management Process	formalpm.process.ibm.com	A traditional project management process where development progresses in sequential phases: Requirements, Design, Implementation, and Testing.	
OpenUP Process	openup.process.eclipse.org	The OpenUP process is one of the processes created by the eclipse process framework project (EPF). It is included here as one example of how to integrate existing processes. For more details see http://www.eclipse.org/epf	
Scrum	scrum2.process.ibm.com	A process template for Scrum. Scrum is a popular approach to manage projects in an agile way.	
Simple Team Process	simple.process.ibm.com	A simple process to get you started quickly. Team members are entitled to perform modifications of any kind, outsiders are not.	
Unconfigured Process	unconfigured.process.ibm.com	This process should be used for project areas that consume a shared process or that are used to manage work item access control. It does not provide any configuration.	

g) The next page we will explore is **User Management.** On the menu bar at the top of the screen click the **Users** option. Use this page to create, edit, or archive users. Archiving a user is the only way to not have a user show up in search results. You can't delete users.



Active Users

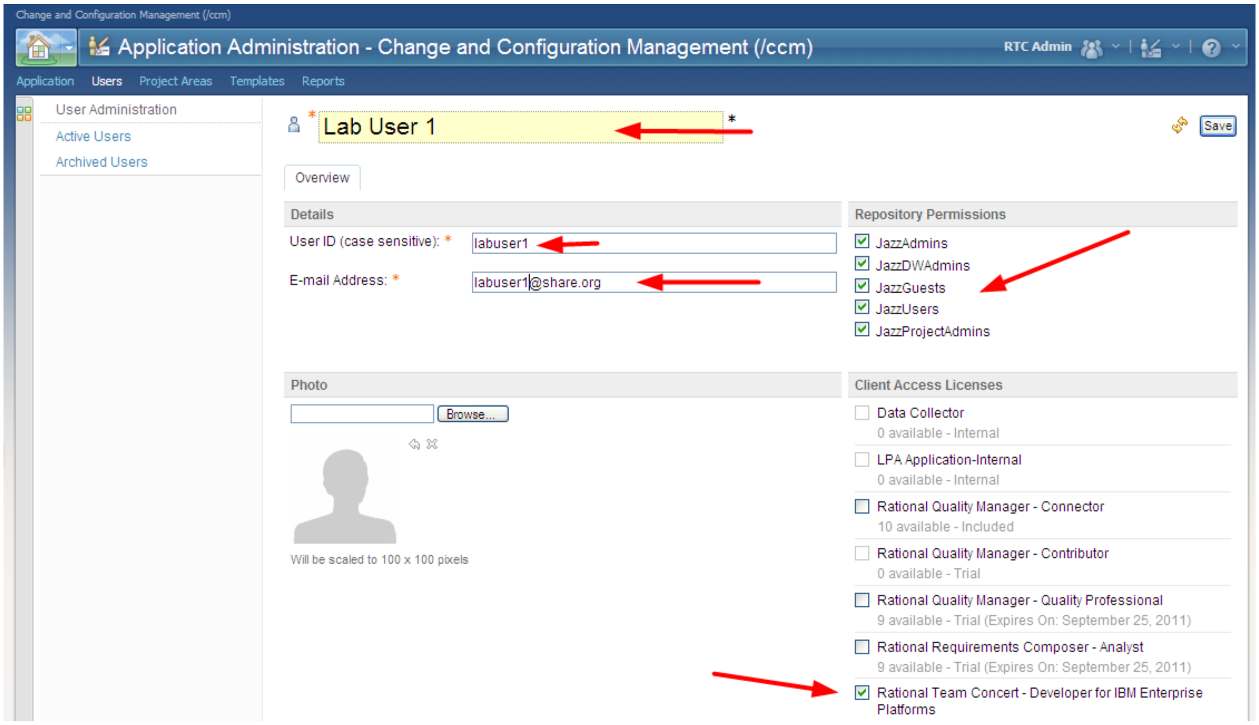
You can create user accounts and edit user account and licensing information in the Application User Administration pages, or in the Jazz Team Server User Administration pages. However, to edit Application-specific user settings, you will need to use the Application User Administration page. In all cases the common user information is synchronized across the Applications and the Server.

Use this page to create, import, and edit user accounts and assign client access licenses to users. To create a new account, click the "Create User" action above. To edit an account and assign licenses, click on a user name in the list below. To search for a particular user, enter a portion of the name, user ID, or email address in the filter-box below.

Previous | **1 - 7** of **7** | Next

Name	User ID	E-Mail	Actions
Data Collection User	dw_user	dwUser@share.org	
RTC Admin	rtcadmin	rtcadmin@share.org	
ccm_user	ccm_user	unknown	
jts_user	jts_user	unknown	
lpa_user	lpa_user	unknown	
qm_user	qm_user	unknown	
rm_user	rm_user	unknown	

h) Create a user by clicking the **Create User** option at the top right of the screen. You will create the team user you will use for this lab. Please create the user substituting the number of your lab machine for the number in the user name. Create a user **labuser#** when you create a user this way, the password will be the same as the user id (**labuser#**) You can modify a users repository permissions and the client access licenses on this page. You must give users a ***Rational Team Concert - Developer for IBM Enterprise Platforms*** client access license to have write privileges which are needed to save work items, save queries, and check in code. You can also upload a picture which will be displayed with that user. For this lab, add your user name, user id, and email address and assign your user ***Developer for IBM Enterprise Platforms*** license, and all of the repository permissions.



Change and Configuration Management (/ccm)

Application Administration - Change and Configuration Management (/ccm)

RTC Admin

Application Users Project Areas Templates Reports

User Administration

Active Users

Archived Users

Lab User 1 *

Save

Overview

Details

User ID (case sensitive): * labuser1

E-mail Address: * labuser1@share.org

Repository Permissions

☑ JazzAdmins

☑ JazzDWAdmins

☑ JazzGuests

☑ JazzUsers

☑ JazzProjectAdmins

Photo

Browse...

Will be scaled to 100 x 100 pixels

Client Access Licenses

☐ Data Collector
0 available - Internal

☐ LPA Application-Internal
0 available - Internal

☐ Rational Quality Manager - Connector
10 available - Included

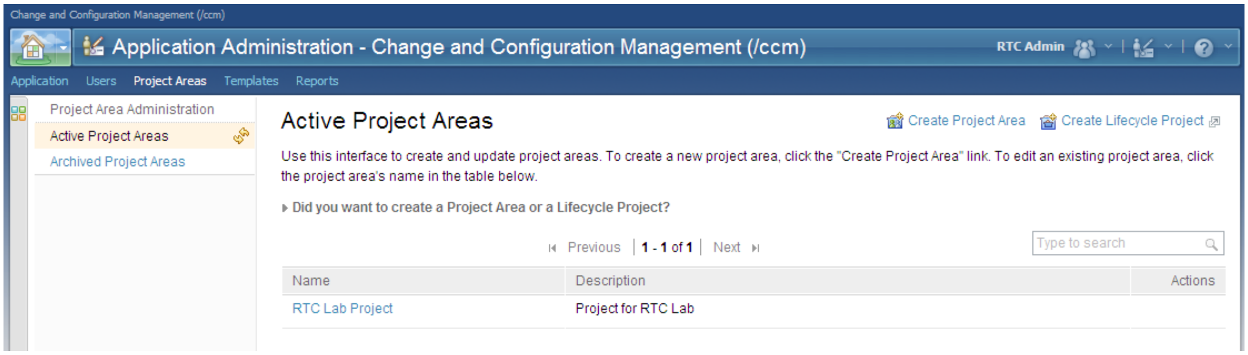
☐ Rational Quality Manager - Contributor
0 available - Trial

☐ Rational Quality Manager - Quality Professional
9 available - Trial (Expires On: September 25, 2011)

☐ Rational Requirements Composer - Analyst
9 available - Trial (Expires On: September 25, 2011)

☑ Rational Team Concert - Developer for IBM Enterprise Platforms

i) The last page we will explore in the RTC admin console is **Project Area Administration.** Click on Project Areas on the menu bar at the top of the screen**.** As the name implies, this is the interface used to manage Project Areas. You can create new Project Areas, archive Project Areas, view the active Project Areas, and manage Project Areas. To manage a particular Project Area, click on the Project Area name as seen below. You can add administrative users to project areas, add members, and change the process template with this editor.



Change and Configuration Management (/ccm)

Application Administration - Change and Configuration Management (/ccm)

RTC Admin

Application Users Project Areas Templates Reports

Project Area Administration

Active Project Areas

Archived Project Areas

Active Project Areas

Create Project Area Create Lifecycle Project

Use this interface to create and update project areas. To create a new project area, click the "Create Project Area" link. To edit an existing project area, click the project area's name in the table below.

Did you want to create a Project Area or a Lifecycle Project?

Previous | 1 - 1 of 1 | Next

Type to search

Name	Description	Actions
RTC Lab Project	Project for RTC Lab	

Let's add the user id we just created to the project area as an administrator. Go to the administrator section and click **add…** then in the **Select Users** box click **Show All**. Then select your user id followed by the **Add and Close** button.

j) Click OK to return to the project Administration screen. You must now click **Save** to save your changes

You may be asked if you want to send an invitation email to your new member, as email notification is disabled just hit the cancel button.

Conclusion

This concludes Module 1. You now should be familiar with the functions available from the admin console and have a user created to perform the lab.

Module 2 Setting up the Team

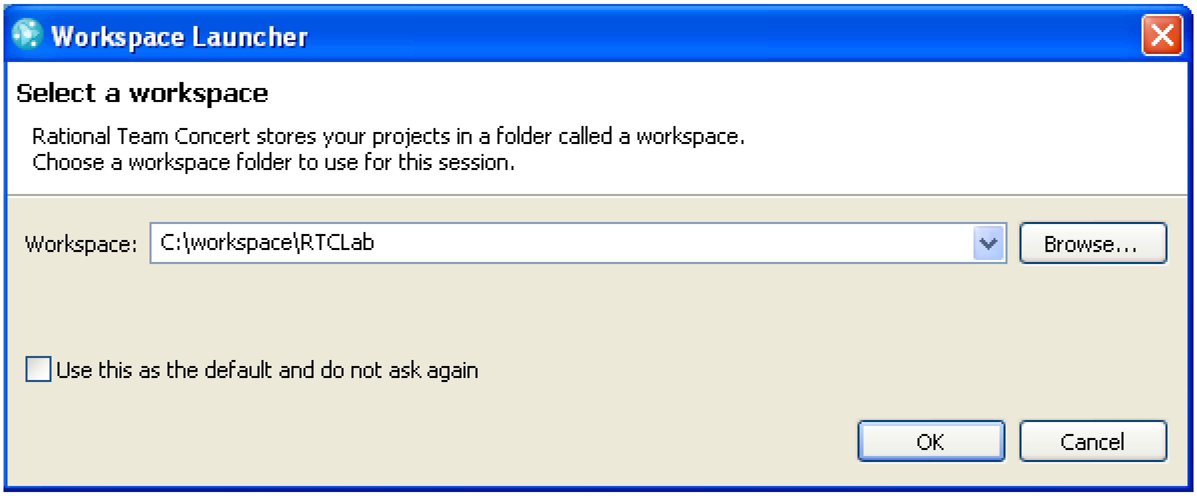
Lab Scenario

In this lab you will focus on how to connect to a repository and its defined project areas. You will also learn about teams, and add your new lab user to the team area for the lab.

2.1. Start the Rational Team Concert client

a) You should find the icon for Rational Team Concert on your desktop (IBM Rational Team Concert), double click to start the client. (The client has the Rational Developer for z client also installed for use during this lab).

You will be prompted with the Workspace Launcher where you can select your Eclipse workspace. This is a directory where all your settings are managed and where all your source code is stored. The workspace, **RTClab**, should be listed by default but if it is not just click the downward pointing arrow in the box and select **C:\workspace\RTCLab.** Click **OK.** Use the **RTClab** workspace for this lab, as it is preconfigured for you.



Workspace Launcher

Select a workspace

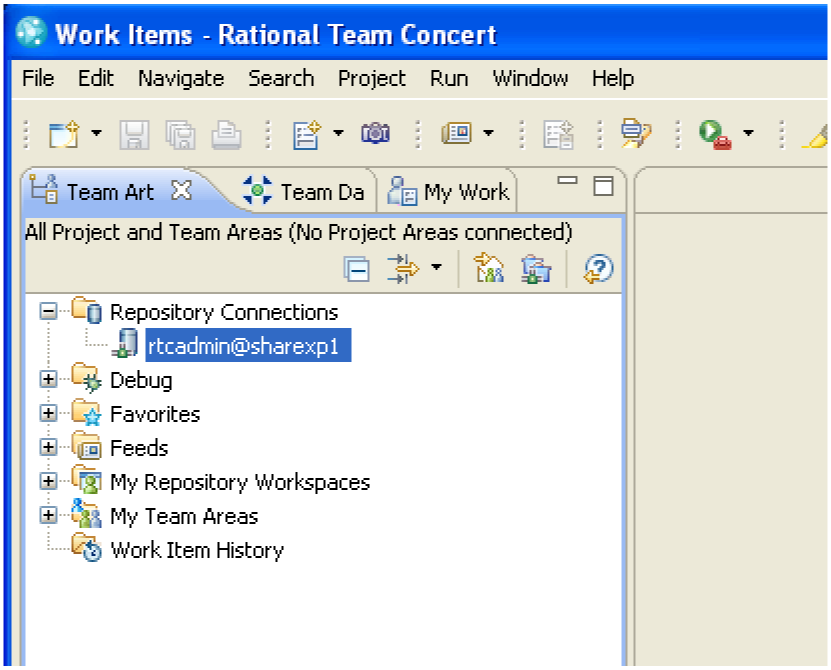
Rational Team Concert stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\workspace\RTCLab

□ Use this as the default and do not ask again

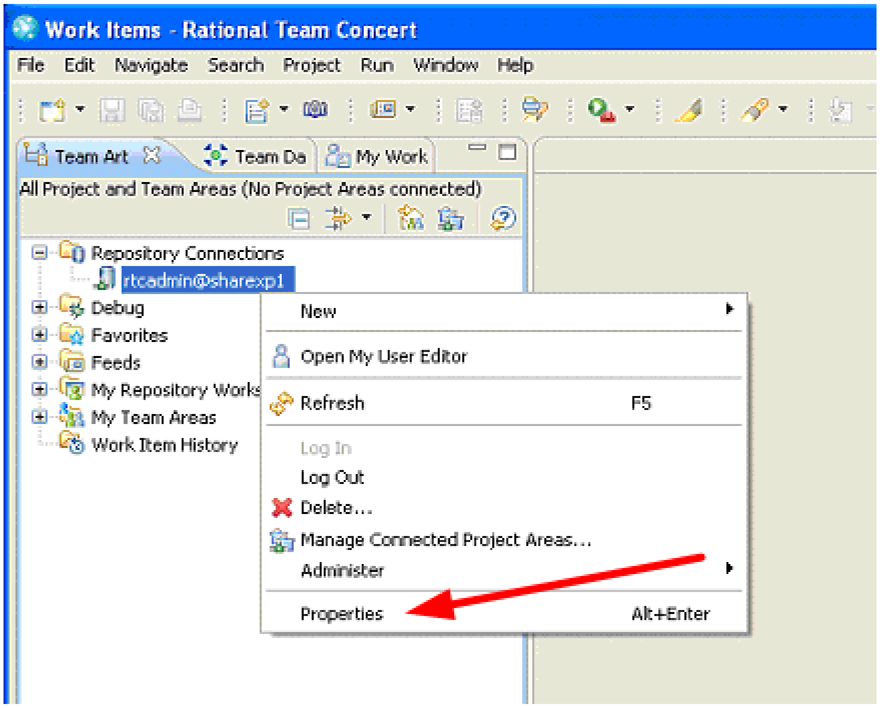
Note : If you have already gone past the workspace launcher selection box, you can still get to the required workspace by switching workspaces. Go to **File** → **Switch Workspace** and then either choose the workspace name or choose **Other** and type the workspace name in the field, or use the downward pointing arrow, or Browse to the workspace.

b) You should now be in the **Work Items** Perspective, as shown in the picture below. If you are not, you can open the Perspective by selecting **Window → Open Perspective → Work Items**

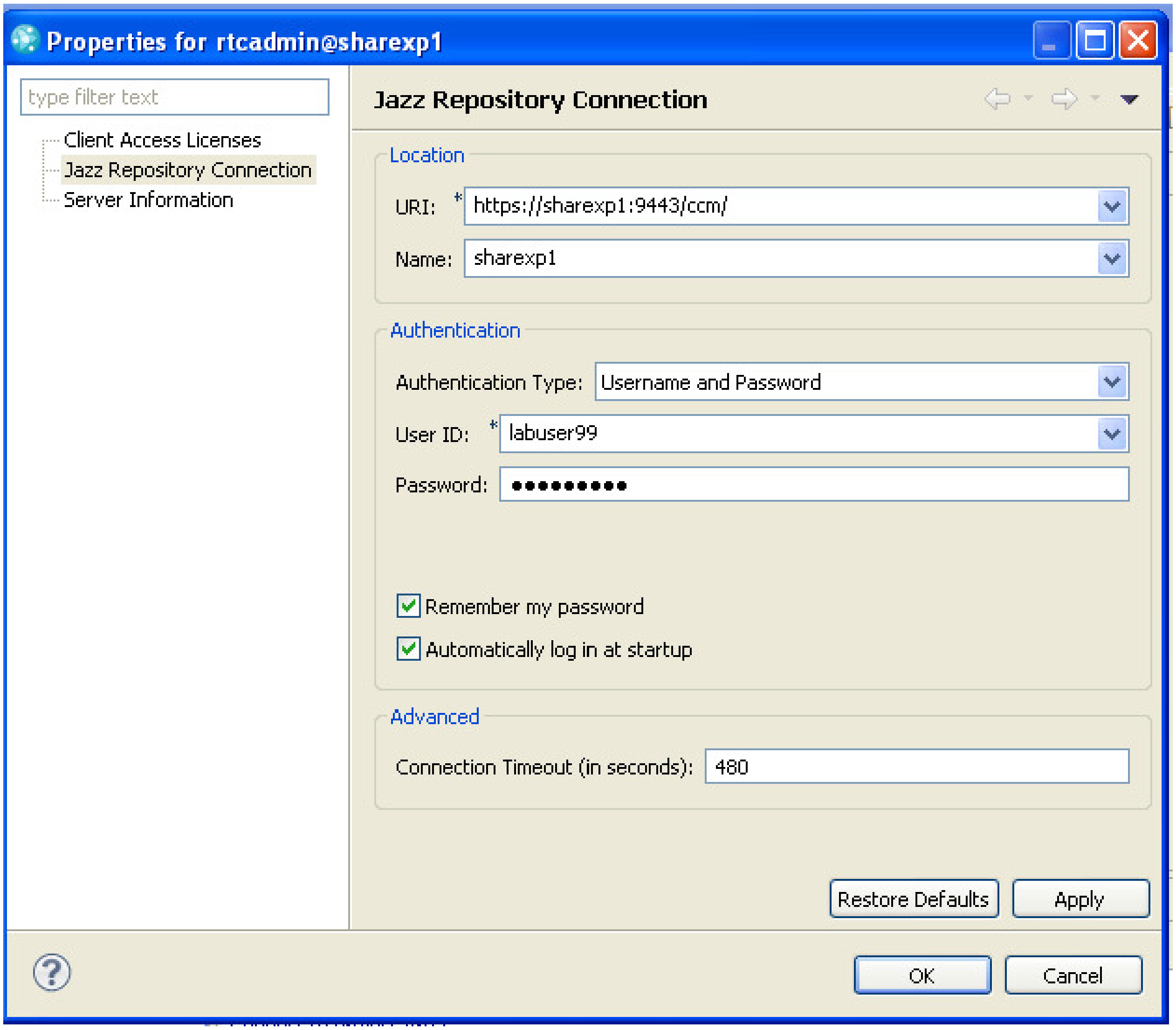

Work Items - Rational Team Concert
File Edit Navigate Search Project Run Window Help
Team Art
Team Da
My Work
All Project and Team Areas (No Project Areas connected)
Repository Connections
rtcadmin@sharexp1
Debug
Favorites
Feeds
My Repository Workspaces
My Team Areas
Work Item History

2.2. Create a repository connection and connect to a project

a) A repository connection has already been created for you to your local server, however it is using the rtcadmin userid and we need to log in with the user id we just created in the previous exercise. Right click on the repository connection node that says ***rtcadmin@sharexp1*** and select **Properties** as shown below



b) In the dialog box select the Jazz Repository Connection on the left side of the screen. Then change the user id and password to the one you created previously (labuser#). The password is the same as the user id. Alternatively if you did not create a userid just login with labuser99 (password labuser99) which has already been created for you. Press **OK**.



Properties for rtcadmin@sharexp1

Jazz Repository Connection

Location

URI: * https://sharexp1:9443/ccm/

Name: sharexp1

Authentication

Authentication Type: Username and Password

User ID: * labuser99

Password: ●●●●●●●●●●

☑ Remember my password

☑ Automatically log in at startup

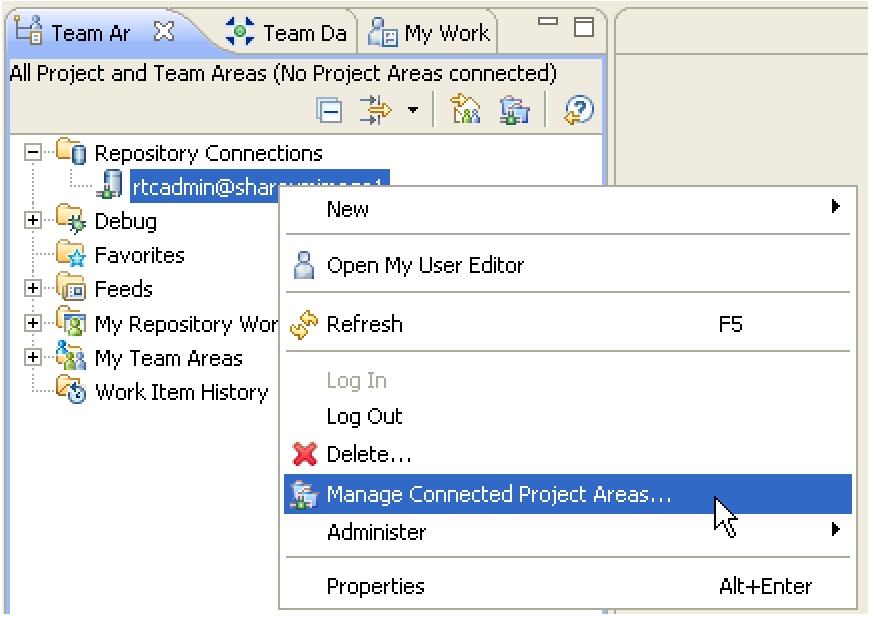
Advanced

Connection Timeout (in seconds): 480

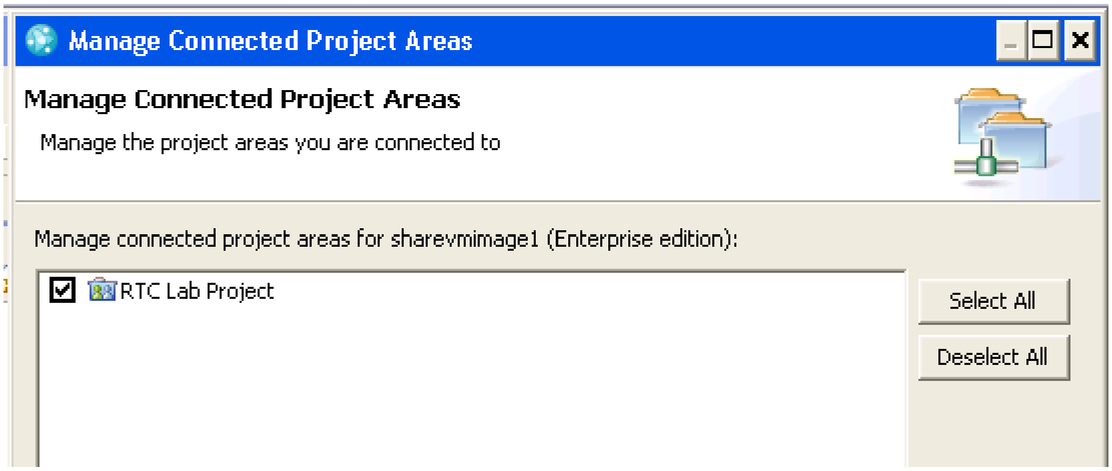
c) If you get any certificate problems when connecting, just accept the certificate permanently:

R

d) Next you need to connect to the project area. Right click on the repository connection you have just logged into and select **Manage Connected Project areas…**



e) On the Connect to project areas screen select the RTC Lab Project and click Finish.


Manage Connected Project Areas

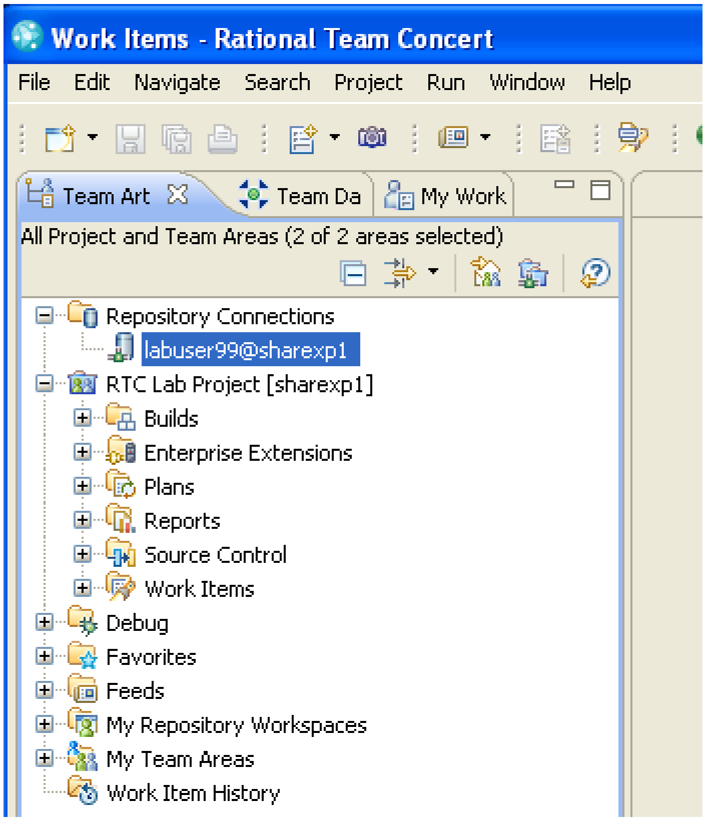
Manage Connected Project Areas

Manage the project areas you are connected to

Manage connected project areas for sharevmimage1 (Enterprise edition):

☑ RTC Lab Project

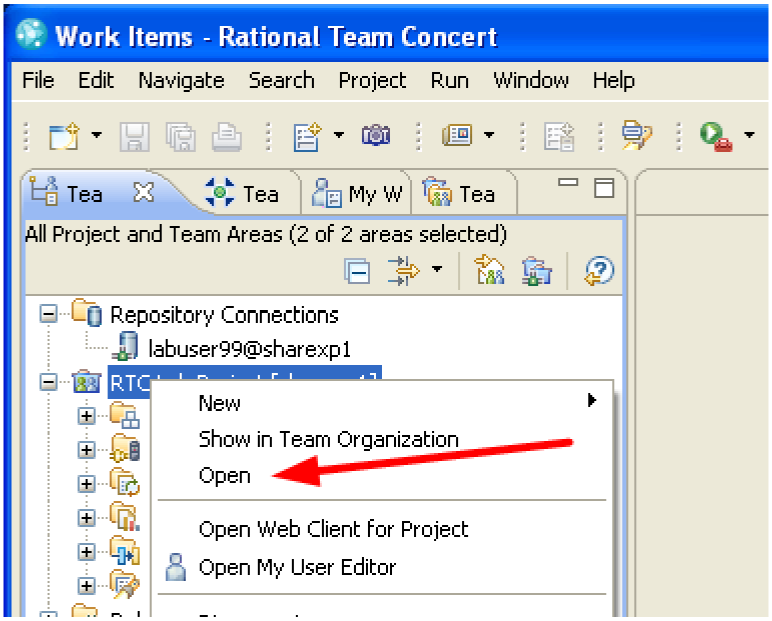
f) You will now see the project added to the tree, and if you expand it you will see a collection of new nodes that we will explore later:



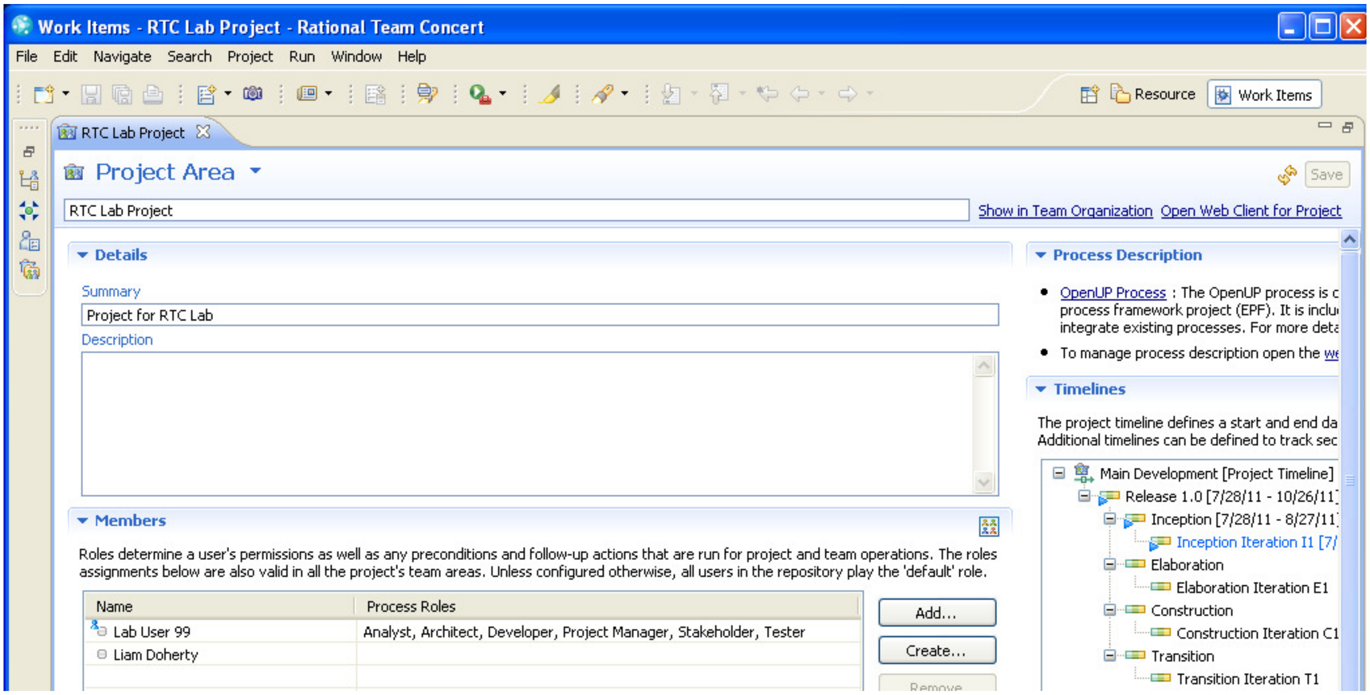
2.3. Meet the team and add your new user

a) The next step is to add yourself to the RTC Lab Project. Start by opening the project area definition by right clicking on the project in the Team Artifacts view and selecting **Open.**

You will need to select the Team Artifacts tab to get back to the correct view. This is the left most tab.


Work Items - Rational Team Concert
File
Edit
Navigate
Search
Project
Run
Window
Help
Tea
Tea
My W
Tea
All Project and Team Areas (2 of 2 areas selected)
Repository Connections
labuser99@sharexp1
RTC
New
Show in Team Organization
Open
Open Web Client for Project
Open My User Editor

c) Here you will see the project area definition, including, in the **Members** area, the users assigned to the project area. Click on the **Add…** button to add your new user id, **labuser#**



Project Area: RTC Lab Project

Summary: Project for RTC Lab

Description:

- OpenUP Process : The OpenUP process is c process framework project (EPF). It is inclu integrate existing processes. For more deta
- To manage process description open the we

The project timeline defines a start and end da Additional timelines can be defined to track sec

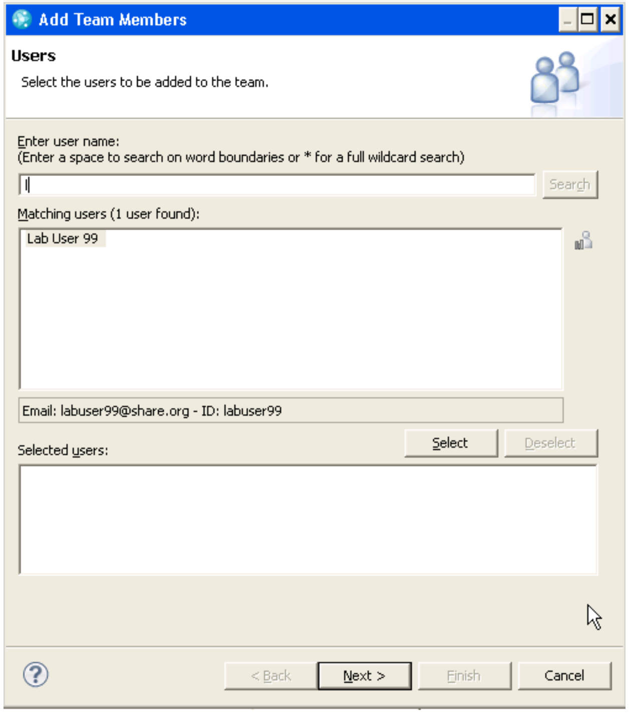
- Main Development [Project Timeline]
  - Release 1.0 [7/28/11 - 10/26/11]
    - Inception [7/28/11 - 8/27/11]
      - Inception Iteration I1 [7/
    - Elaboration
      - Elaboration Iteration E1
    - Construction
      - Construction Iteration C1
    - Transition
      - Transition Iteration T1

Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role.

Name	Process Roles
Lab User 99	Analyst, Architect, Developer, Project Manager, Stakeholder, Tester
Liam Doherty	

Note : Double clicking on a tab, such as the RTC Lab Project Tab, will open the tab into full screen. Double clicking it again will put it back into it's collapsed mode.

d) In the search windows search for users that start with "l" and press **Search** and you should find your new user. Click **Select** so that your new user is in the selected users list.



Add Team Members

Users

Select the users to be added to the team.

Enter user name:
(Enter a space to search on word boundaries or * for a full wildcard search)

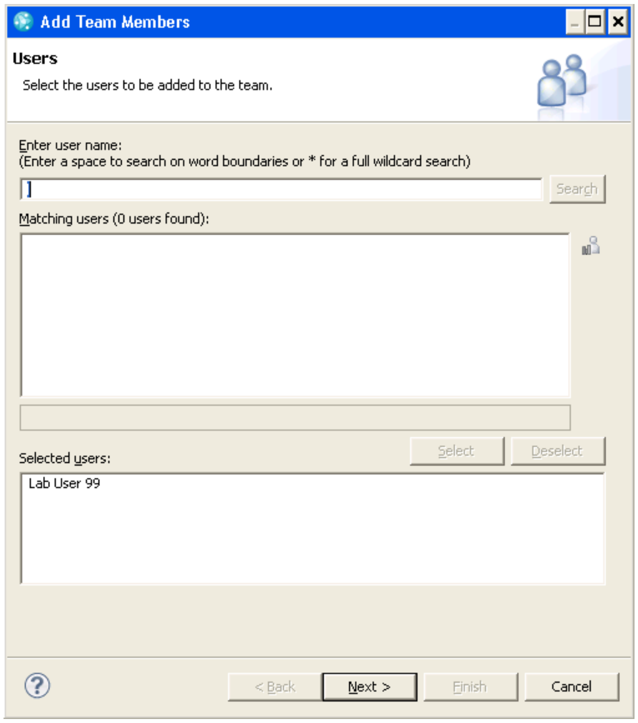
l

Matching users (1 user found):

Lab User 99

Email: labuser99@share.org - ID: labuser99

Selected users:



Add Team Members

Users

Select the users to be added to the team.

Enter user name:
(Enter a space to search on word boundaries or * for a full wildcard search)

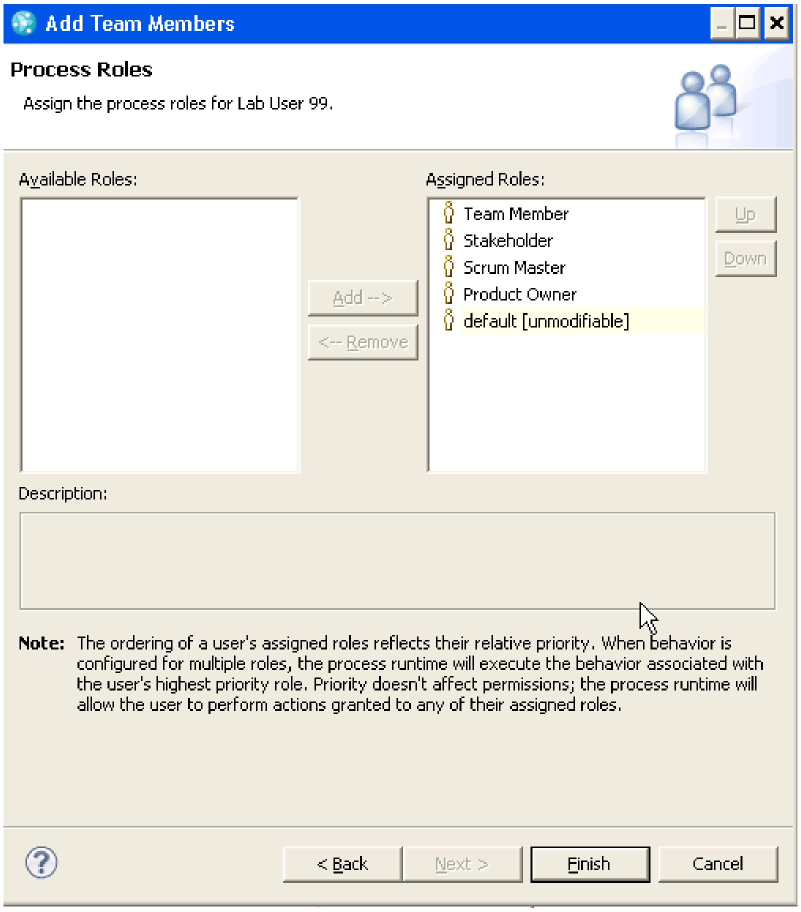
]

Matching users (0 users found):

Selected users:

Lab User 99

e) Click **Next,** Grant all permissions to your user **labuser#** by selecting all Available Roles and using the **Add** button to move to the Assigned Roles.



Process Roles

Assign the process roles for Lab User 99.

Available Roles:

Assigned Roles:

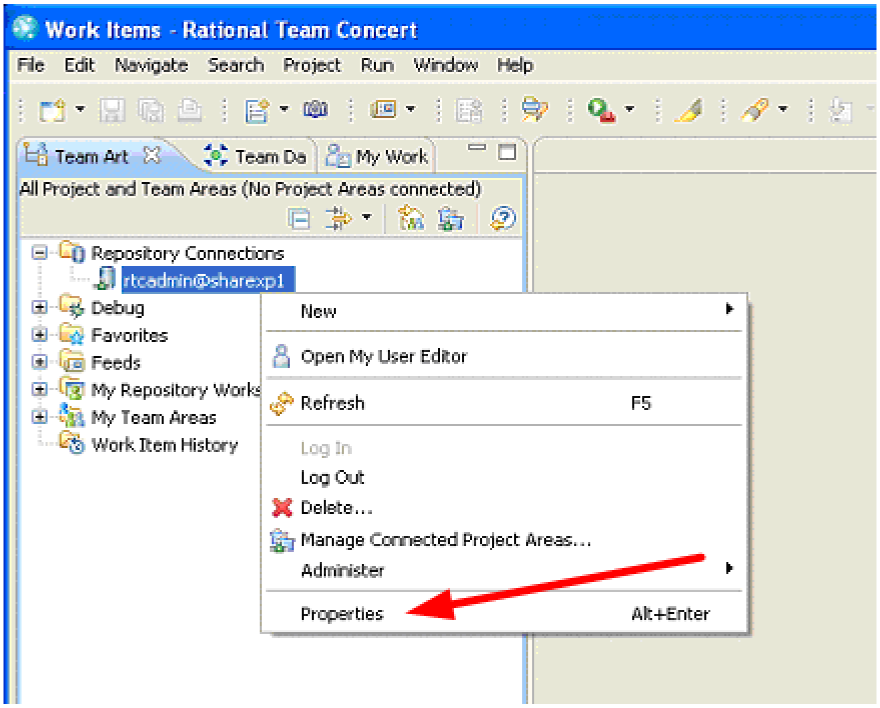
- Team Member
- Stakeholder
- Scrum Master
- Product Owner
- default [unmodifiable]

Note: The ordering of a user's assigned roles reflects their relative priority. When behavior is configured for multiple roles, the process runtime will execute the behavior associated with the user's highest priority role. Priority doesn't affect permissions; the process runtime will allow the user to perform actions granted to any of their assigned roles.

f) Click Finish and then make sure you click **Save** on the project area to save your changes.

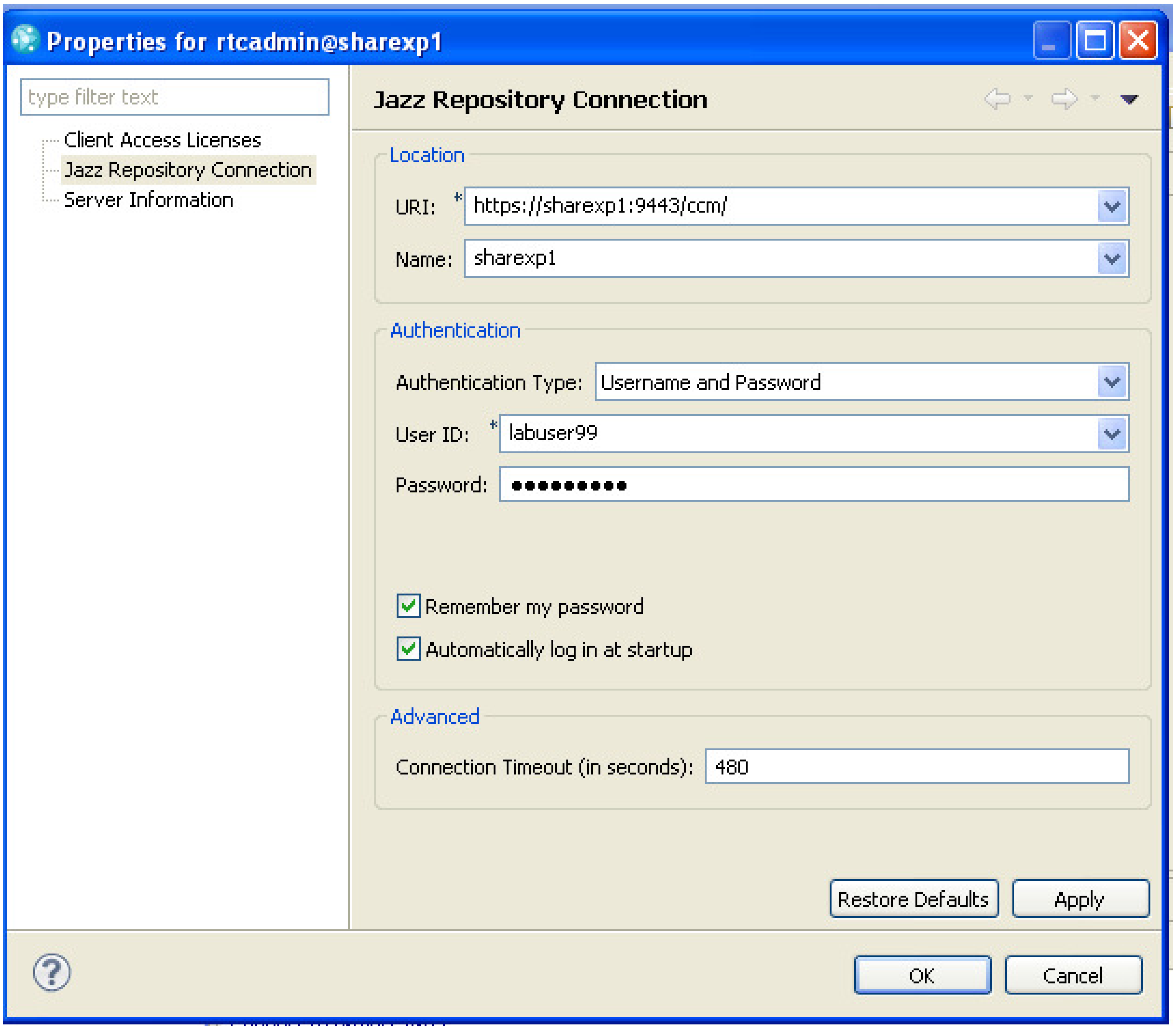
2.4. Changing users and repository connection

a) Another way to change the user that is connected to the jazz repository is to go via the connection properties. From the **Team Artifacts** view (**Window** → **Show View** → **Team Artifacts**), right-click on the repository connection and click **Properties**



b) Here you can enter a new URI if connecting to a different repository.

Note: This is for instructional purposes only. This lab image is already configured to communicate with the appropriate repository. Leave the repository URI the same and change the **user**/**password** to **labuser#**/**labuser#, where # is the number you used.** Select Jazz Repository Connection on the left if it is not selected by default. Press Ok.



URI: * https://sharexp1:9443/ccm/

Name: sharexp1

Authentication Type: Username and Password

User ID: * labuser99

Password: ●●●●●●●●●●

☑ Remember my password

☑ Automatically log in at startup

Connection Timeout (in seconds): 480

Now you are connected to the Jazz Team Server on your local machine denoted by the URI https://sharexp1:9443/ccm/. You are connected as user **labuser#**.

.

Conclusion

This concludes Module 2. You are now able to easily connect to an existing Jazz repository and project area as any valid user and begin collaborating with your teammates. You have also added your individual user to the team to begin work.

Module 3 Planning Your Work

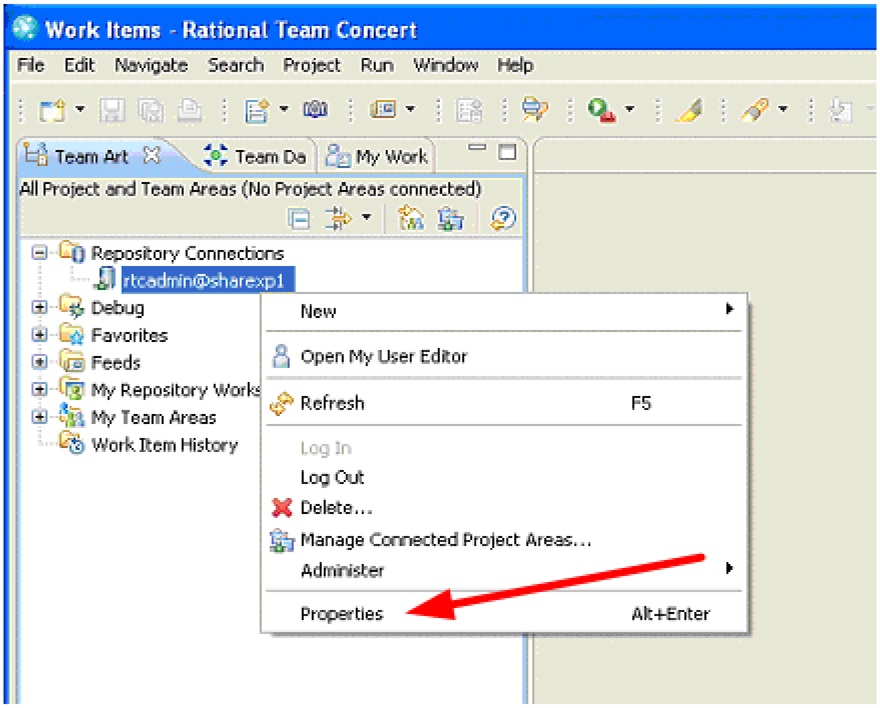
Lab Scenario

You want to track all the work on your projects. All your work (tasks, defects, enhancements) is based around the concept of Work Items. You hear how Work Items are fundamental to Rational Team Concert and how you use these work items to track the work you do in each item. You prioritize your work so that you can do the right things at the right time in the plan.

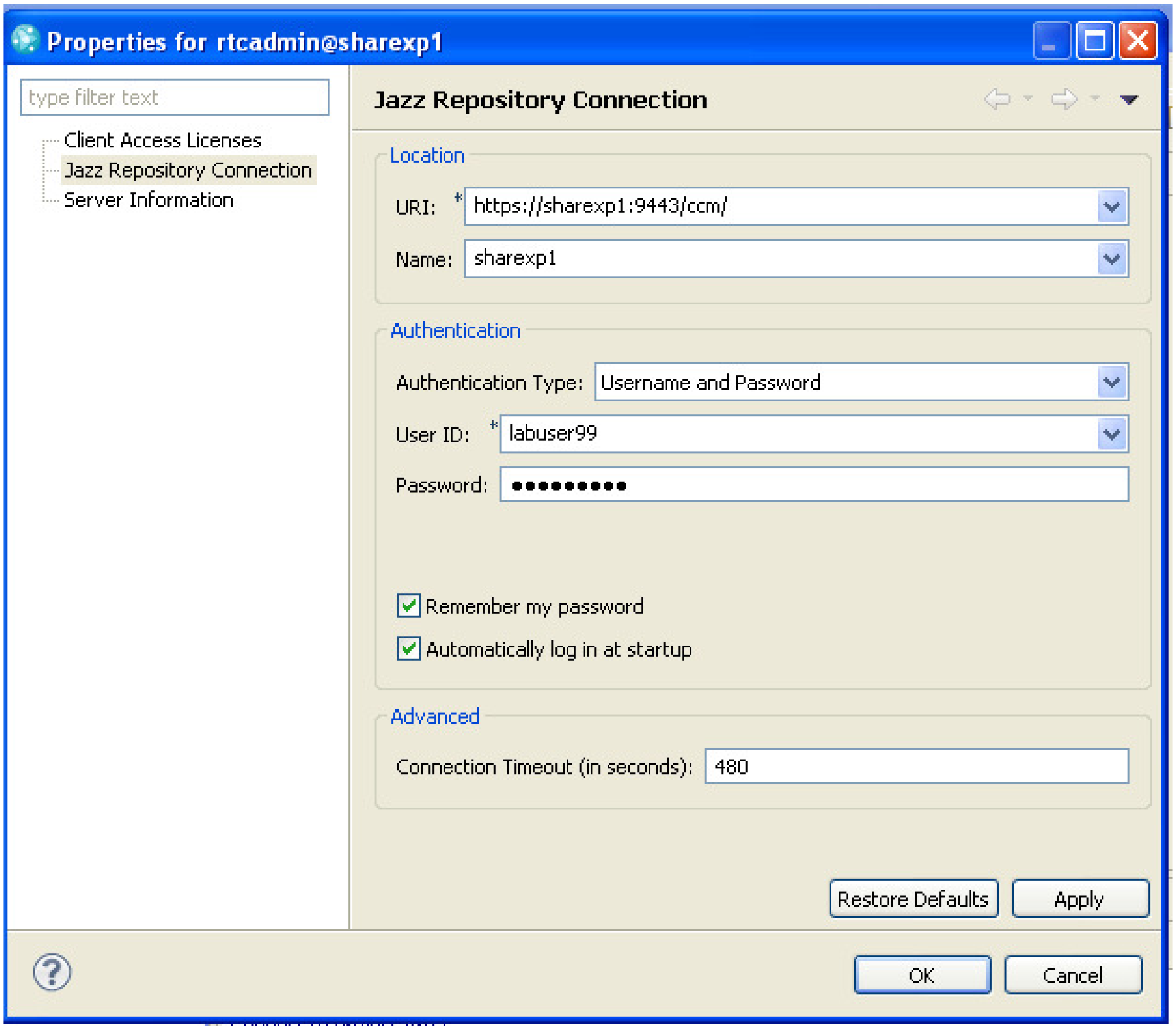
In Rational Team Concert, you will track the creation of an iteration plan using work items. You will create a new work item, set the priority, estimate how long the work will take, complete the work, and update the work item accordingly.

3.1. Creating Work Items

a) Ensure you are logged into RTC. If you created a user in Module 1 then you can use that user. Alternatively you can use labuser99 which has been created for you already. To login right click on the labuser99@sharexp1 node in the Repository connections and select **login**. Or if using your own userid right click on labuser99 and select **Properties** as shown below



b) In the dialog box select the Jazz Repository Connection on the left side of the screen. Then change the user id and password to the one you created previously (labuser#). The password is the same as the user id. Press **OK**.



Properties for rtcadmin@sharexp1

Jazz Repository Connection

Location

URI: * https://sharexp1:9443/ccm/

Name: sharexp1

Authentication

Authentication Type: Username and Password

User ID: * labuser99

Password: ●●●●●●●●●●

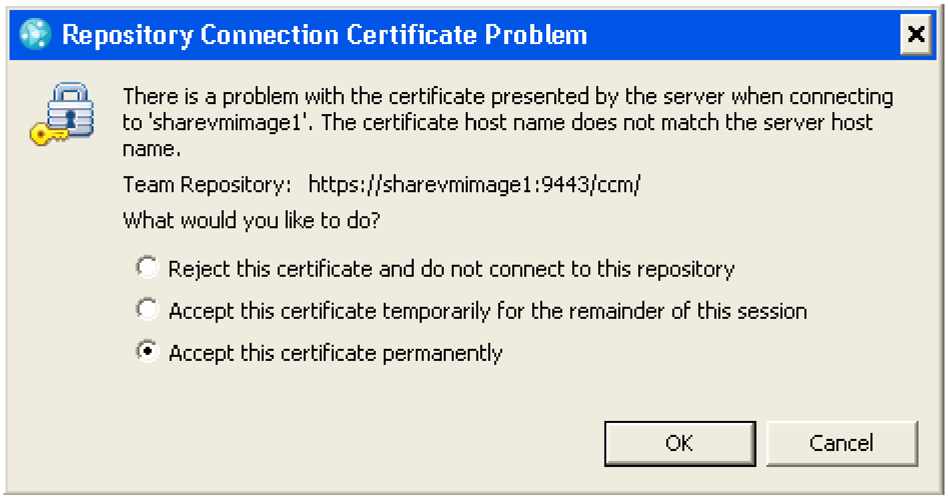
☑ Remember my password

☑ Automatically log in at startup

Advanced

Connection Timeout (in seconds): 480

c) If you get any certificate problems when connecting, just accept the certificate permanently:



Repository Connection Certificate Problem

There is a problem with the certificate presented by the server when connecting to 'sharevmimage1'. The certificate host name does not match the server host name.

Team Repository: https://sharevmimage1:9443/ccm/

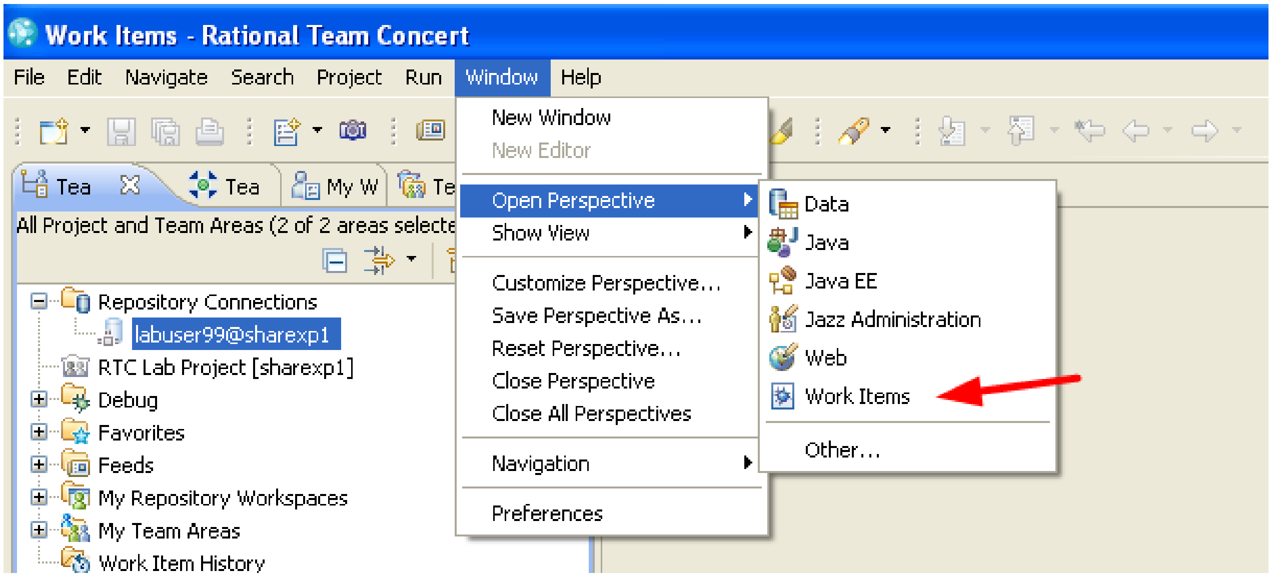
What would you like to do?

○ Reject this certificate and do not connect to this repository

○ Accept this certificate temporarily for the remainder of this session

● Accept this certificate permanently

d) Ensure that you are using the **Work Items** perspective (it should be open by default). Go to **Window** → **Open Perspective** → **Work Items** to open it if it is not already your perspective.

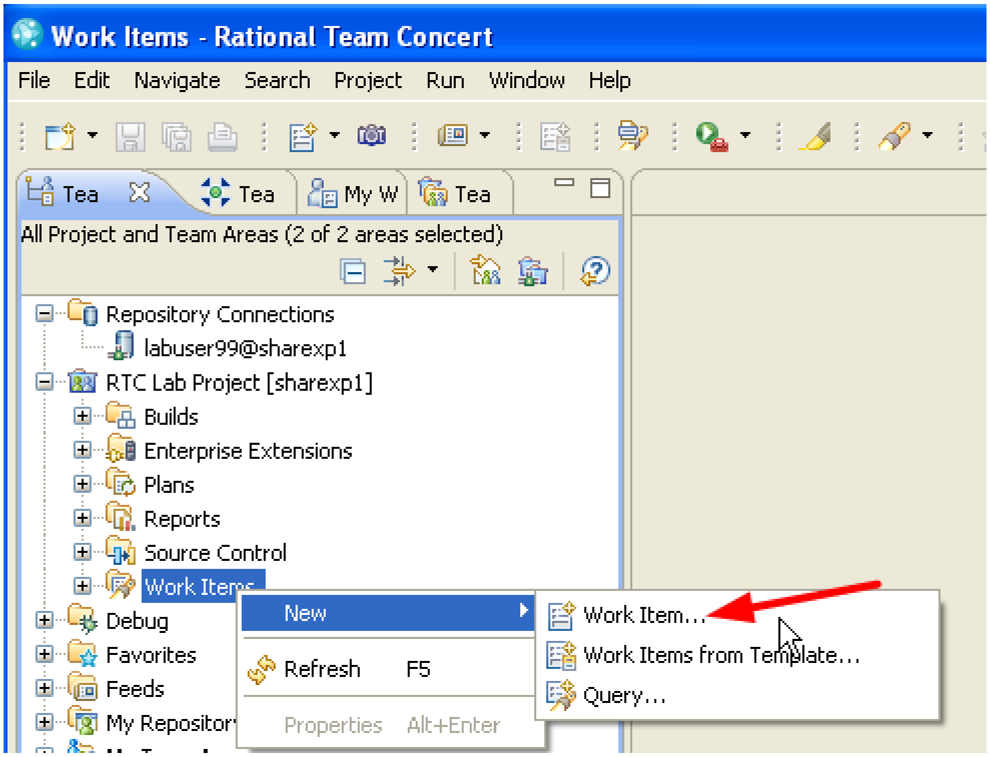


e) Create a new Task work item to implement a new Iteration Plan.

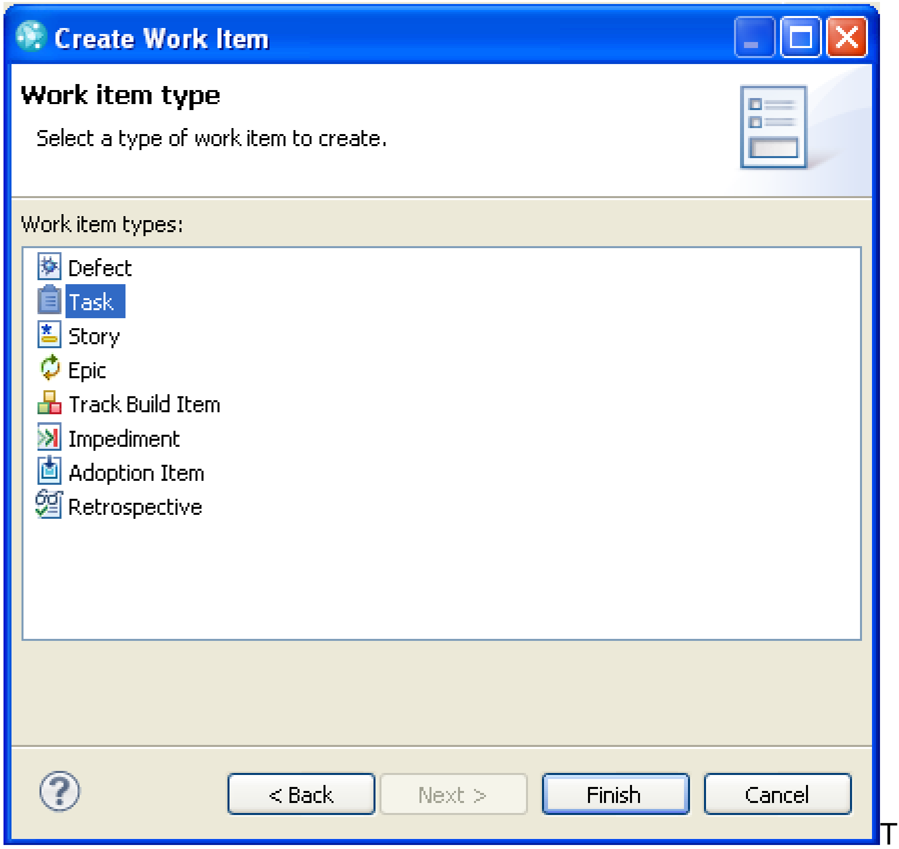
i. Select the **Team Artifacts** view

ii. Expand the **RTC Lab Project** project

iii. Right-click on the **Work items** folder and then select **New->Work Item**.



iv. Select Work item type **Task** and click **Finish**

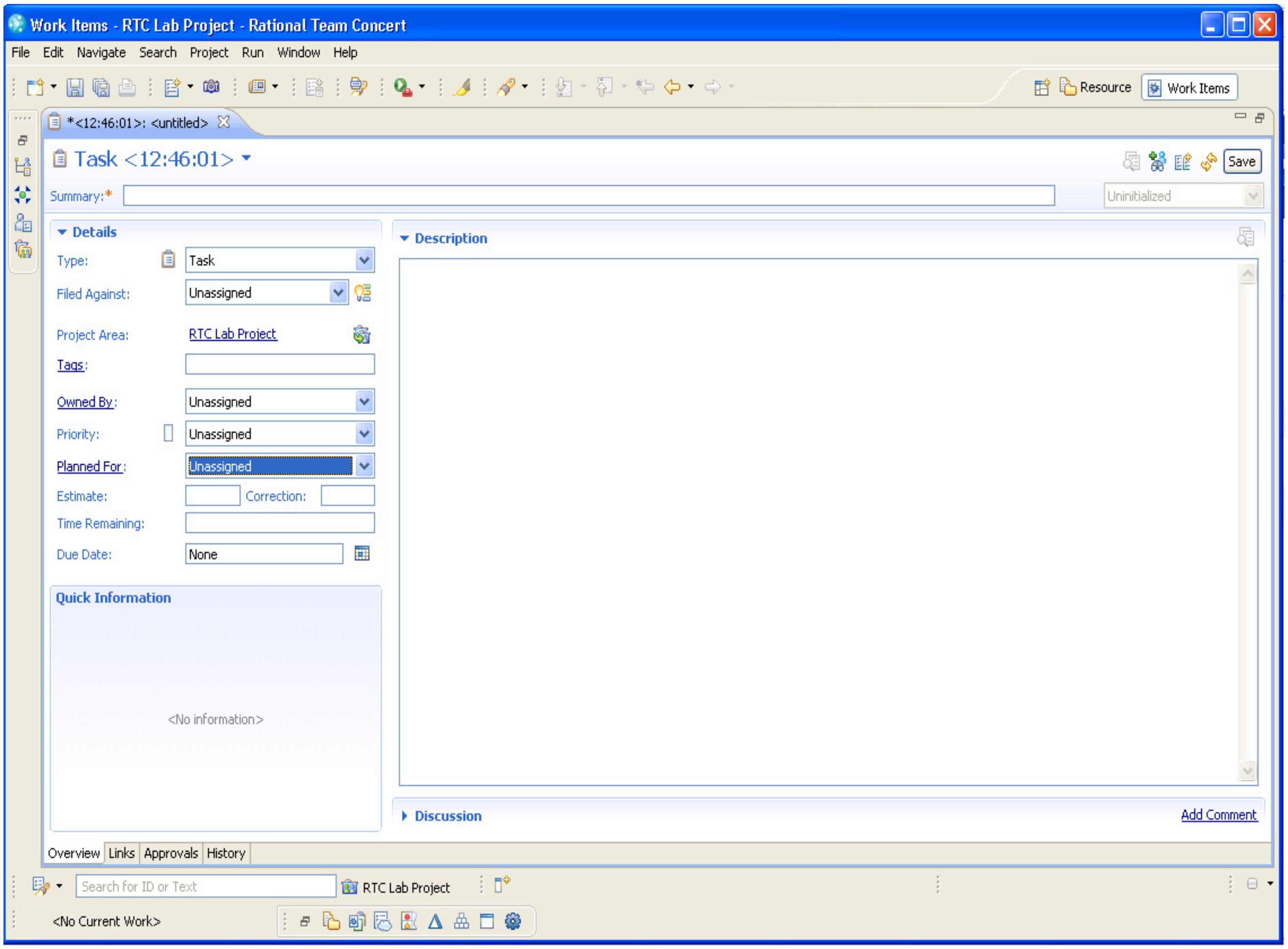


Work item type

Select a type of work item to create.

Test

f) The Work Item editor will automatically open and will look like the picture below.



Summary:*

Uninitialized

Type: Task

Filed Against: Unassigned

Project Area: RTC Lab Project

Tags:

Owned By: Unassigned

Priority: Unassigned

Planned For: Unassigned

Estimate: Correction:

Time Remaining:

Due Date: None

<No information>

g) For the details of the Task:

i. Type `Create Iteration Plan` in the **Summary**

ii. Fill in something useful and descriptive for the **Description**.

iii. Set **File Against** to RTC Lab Project

iv. Set **Owned By** to your lab user. You may need to select More… and then search for your userid by entering an * in the user name area.

v. Set **Priority** to `High`

vi. Leave **Planned For** as Unassigned - we will come back and fill this in later once we create an iteration plan.

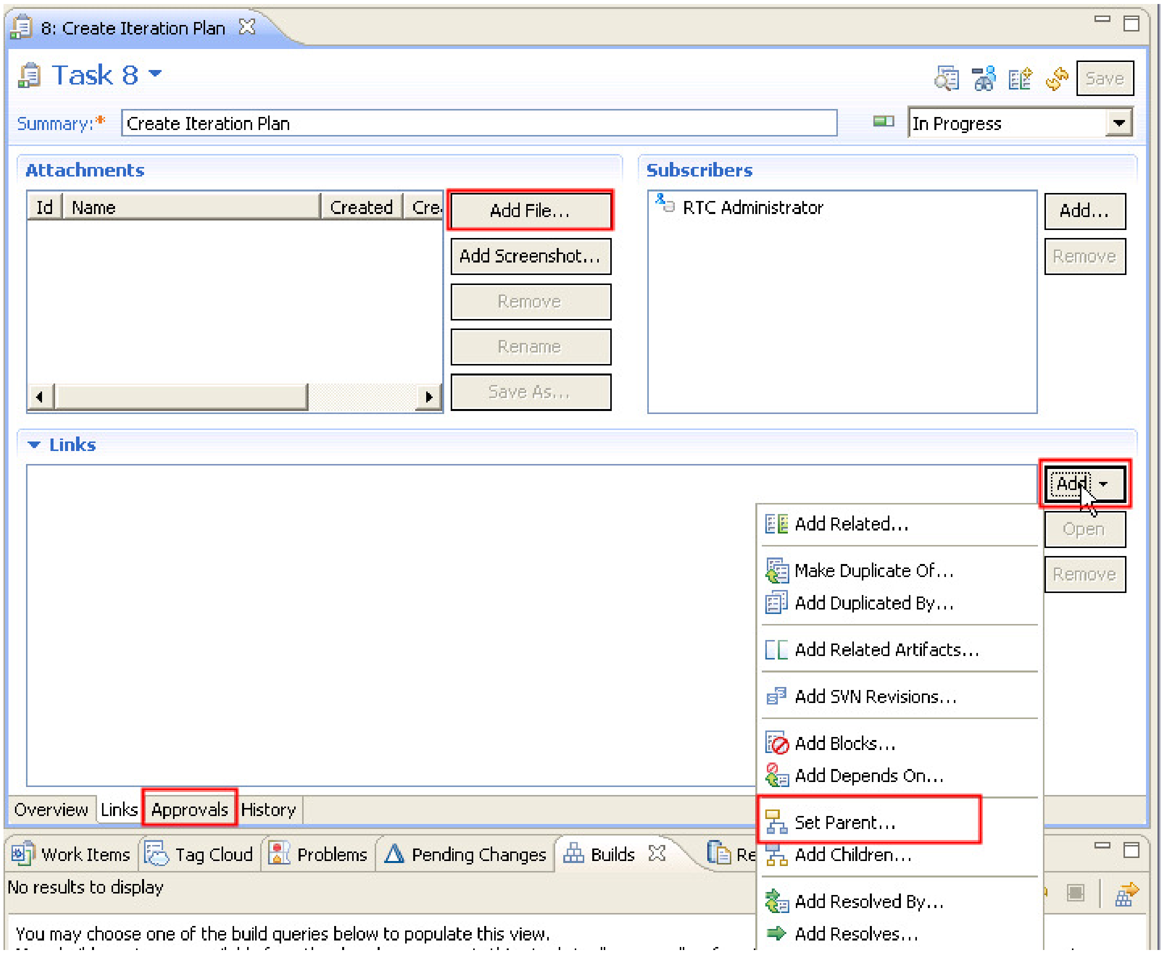
vii. In **Estimate**, type `10 minutes`

viii. **Save** the work item by clicking on the button in the upper-right corner titled **Save**.

ix. From the drop-down in the upper-right corner, select **Start Working**. You will close this work item once you are done with the next section.

x. Click **Save** in the upper-right corner of **Work Item** editor view.

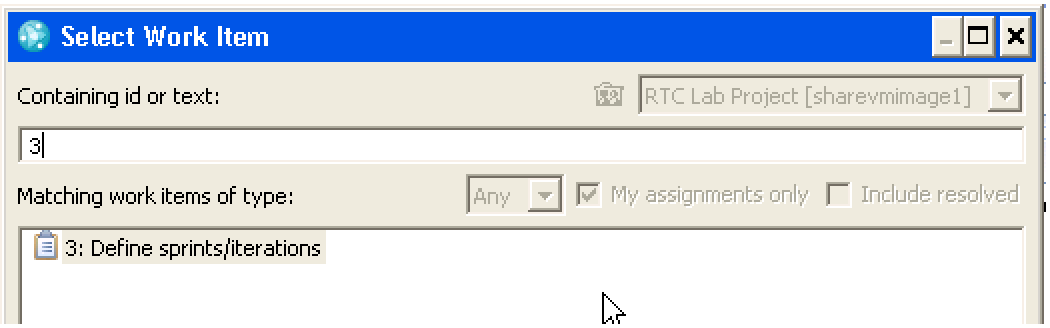
h) To create a parent-child relationship with work items, to make a work item depend on another work item, create a blocking relationship, or categorize as a duplicate, go to the **Links** of the work item. This is also the tab where you can attach a file to a work item or add subscribers. Subscribers are users that get notified when changes are made to a work item. Create another work item so you can associate your first work item with the second. Click the **Add** button as seen in the picture below to link a Work Item to another with the aforementioned relationships.



Summary:* Create Iteration Plan

In Progress

Select **Set Parent** in the presented drop down menu, and then on the next panel enter **3** as the ID. A workitem with the id of 3 should appear. Select this as the parent.



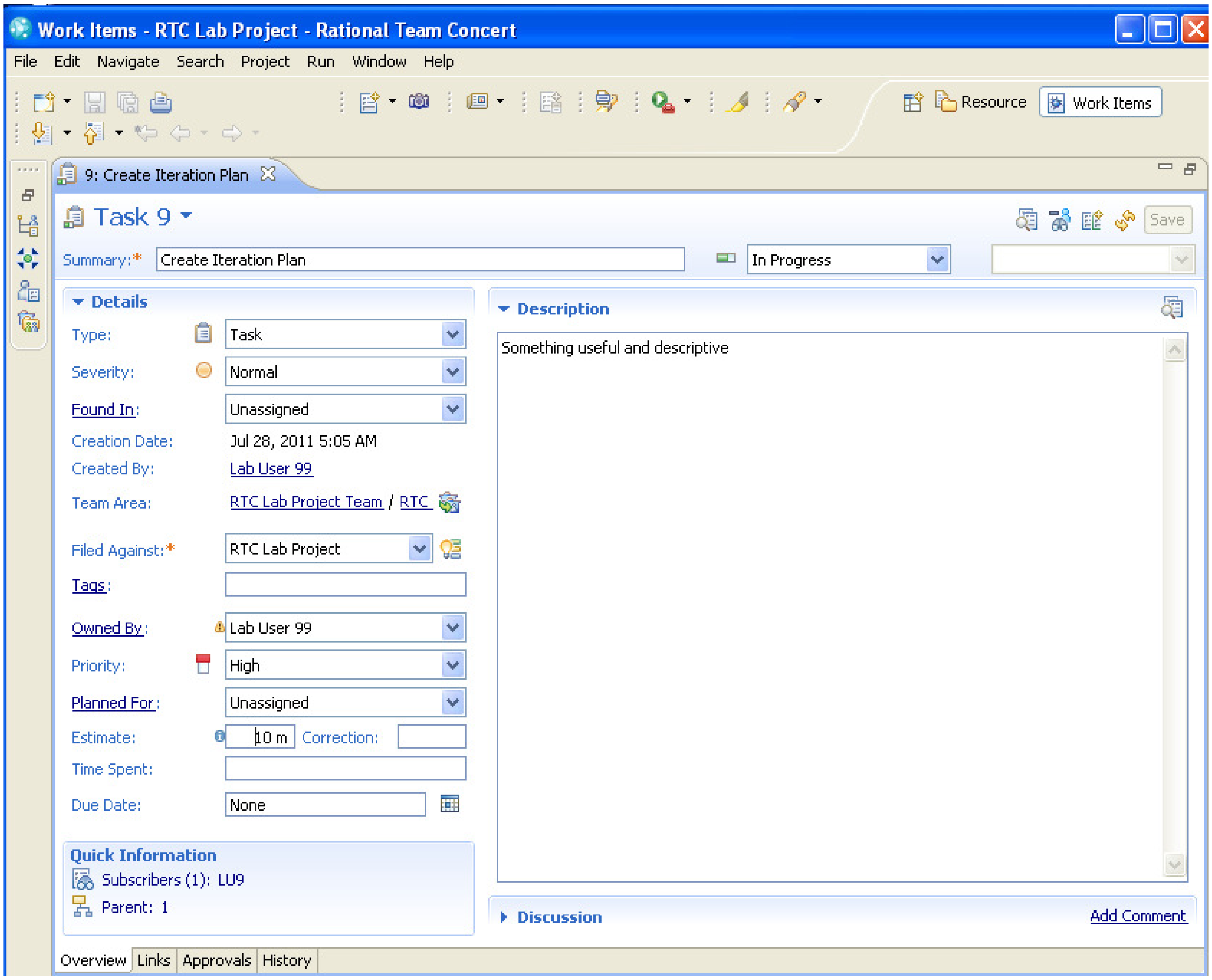
Containing id or text: 3

RTC Lab Project [sharevmimage1]

Matching work items of type: Any ☑ My assignments only □ Include resolved

3: Define sprints/iterations

i) Save the task by clicking the save button in the upper-right corner. Your task should look similar to this after a successful save:



Task 9

Summary:* Create Iteration Plan

In Progress

Type: Task

Severity: Normal

Found In: Unassigned

Creation Date: Jul 28, 2011 5:05 AM

Created By: Lab User 99

Team Area: RTC Lab Project Team / RTC

Filed Against:* RTC Lab Project

Tags:

Owned By: Lab User 99

Priority: High

Planned For: Unassigned

Estimate: 10 m Correction:

Time Spent:

Due Date: None

Description: Something useful and descriptive

Subscribers (1): LU9

Parent: 1

Working on a work item

There are several ways to designate a work item as the active work item:

1. *Open the work item, set its status to Start Working and click Save.*
2. *From the Work Items view, drag the work item into the lower right-hand corner of the Eclipse main frame.*
3. *Right-click the <no current work> string lower right-hand corner of the Eclipse main frame and select Start Working or Recent Work to see a list of work items to work on.*

There are several ways to deactivate an active work item:

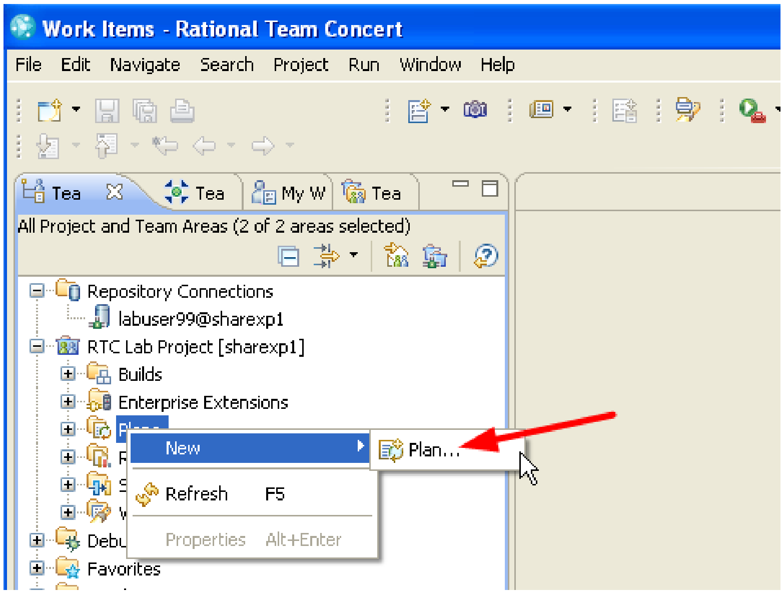
1. *Open the work item and set its status to Stop Working and click Save.*
2. *Right-click the work item in the lower right-hand corner of the Eclipse main frame and select Stop Working.*

When you resolve an active work item, you are prompted to deliver the associated change sets. If you stop working on a work item or start a new change set while a work item is active, you will be prompted to suspend the changes associated with the active work item.

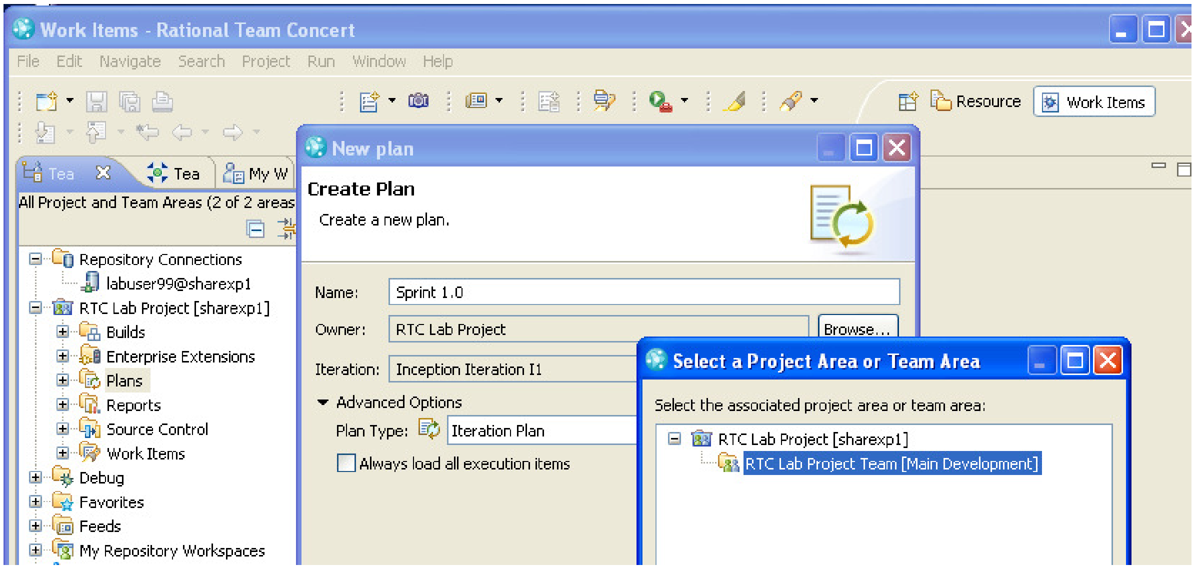
3.2. Creating an Iteration Plan

a) Ensure that you are using the **Work Items** perspective (it should be open by default).

b) You are ready to create a **Plan** what you will use to track your progress as a team over a period of time. Right-click on **Plans**, a child of RTC Lab Project, and select **New** → **Plan**.


Work Items - Rational Team Concert
File Edit Navigate Search Project Run Window Help
Tea Tea My W Tea
All Project and Team Areas (2 of 2 areas selected)
Repository Connections
labuser99@sharexp1
RTC Lab Project [sharexp1]
Builds
Enterprise Extensions
Plans
Debu
Favorites
New
Plan...
Refresh F5
Properties Alt+Enter

c) Name the Iteration Plan **Sprint 1.0.** This is what your team (you) will use to track your progress throughout this Iteration (lab). You will create work items and associate them with this Iteration Plan. Associate the plan with a Project Area by clicking the Browse button next to Owner, selecting RTC Lab Project. Click **Finish**.


New plan

Create Plan

Create a new plan.

Name: Sprint 1.0

Owner: RTC Lab Project

Iteration: Inception Iteration I1

Plan Type: Iteration Plan

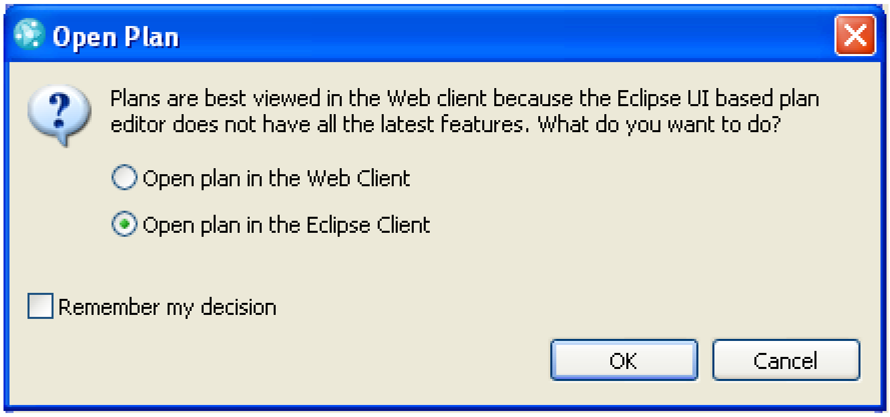
□ Always load all execution items

Select a Project Area or Team Area

Select the associated project area or team area:

RTC Lab Project [sharexp1]
RTC Lab Project Team [Main Development]

You may get an Open Plan dialog that asks where you want to open the plan, in the Web Client or in the Eclipse Client. We will use the Eclipse Client for now



Open Plan

Plans are best viewed in the Web client because the Eclipse UI based plan editor does not have all the latest features. What do you want to do?

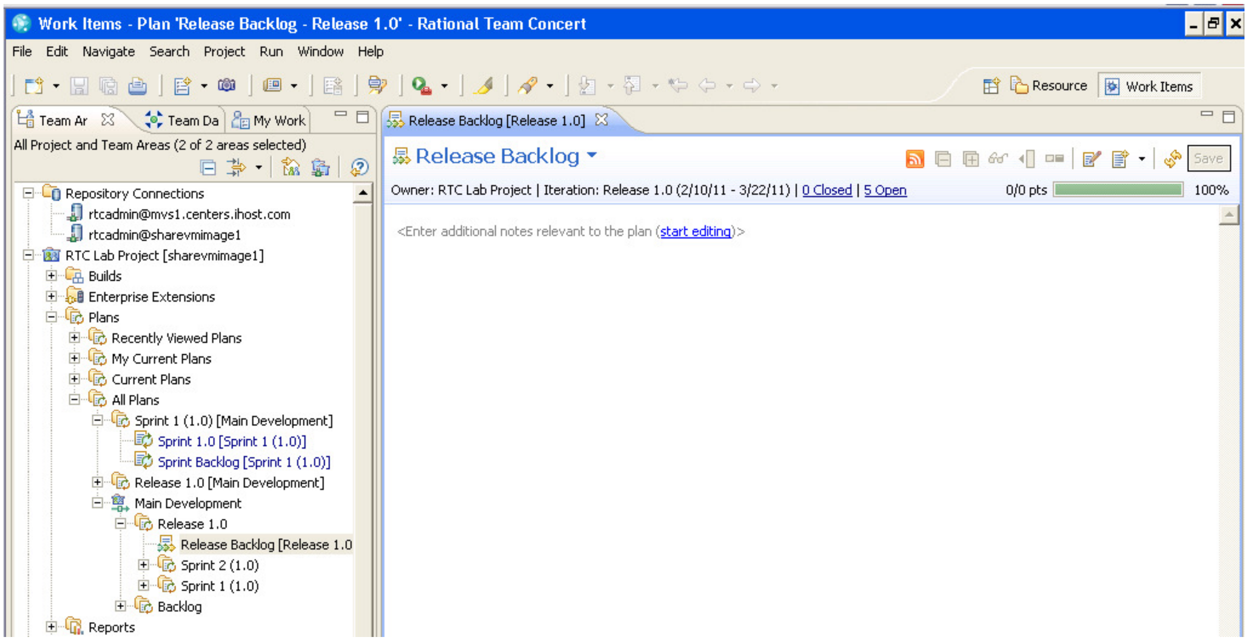
○ Open plan in the Web Client

● Open plan in the Eclipse Client

□ Remember my decision

3.3. Populating plans with Work Items

a) Take some time to create a few work items following the steps in Section 3.1. Set them to be Planned for Release 1.0, and leave the owner as unassigned. Open the plan by navigating to **RTC Lab Project → Plans → All Plans → Main Development → Release 1.0 → Release Backlog.** You will see you start out with a blank page, this is a place for a textual description of the goals or plans for the sprint. Fill in some text and click save.



Work Items - Plan 'Release Backlog - Release 1.0' - Rational Team Concert

All Project and Team Areas (2 of 2 areas selected)

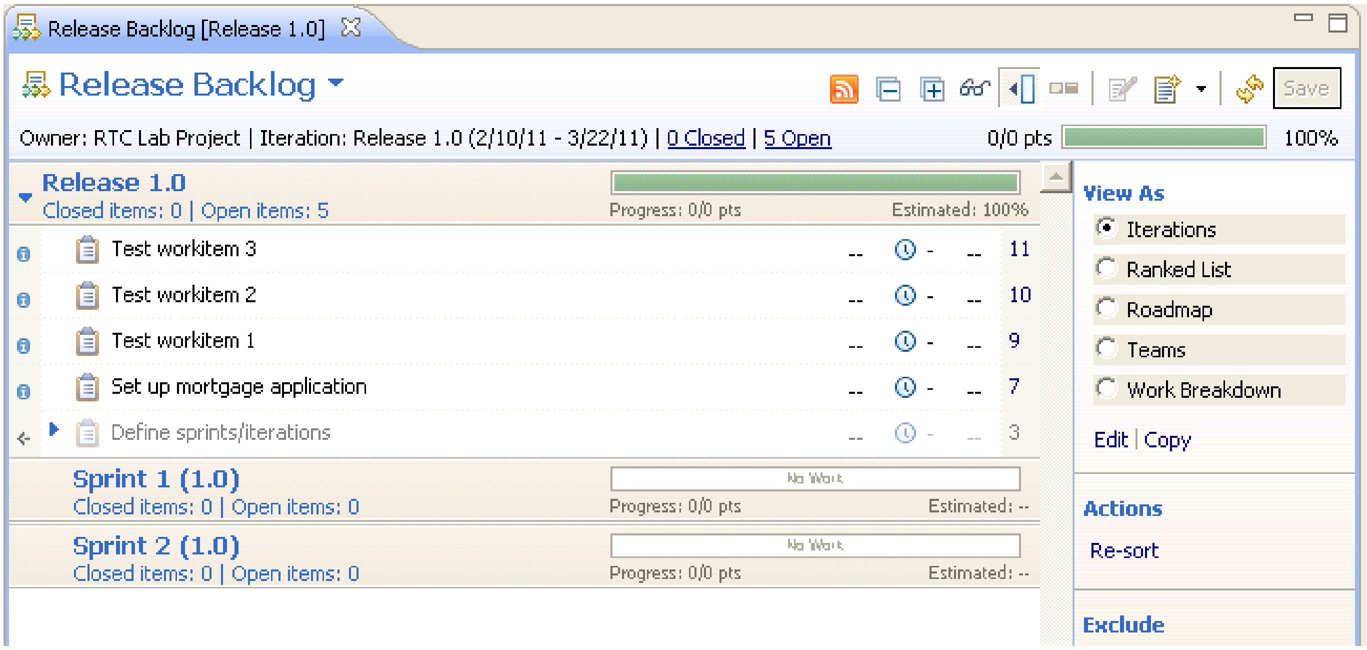
- Repository Connections
  - rtcadmin@mvs1.centers.ihost.com
  - rtcadmin@sharevmimage1
- RTC Lab Project [sharevmimage1]
  - Builds
  - Enterprise Extensions
  - Plans
    - Recently Viewed Plans
    - My Current Plans
    - Current Plans
    - All Plans
      - Sprint 1 (1.0) [Main Development]
        - Sprint 1.0 [Sprint 1 (1.0)]
        - Sprint Backlog [Sprint 1 (1.0)]
      - Release 1.0 [Main Development]
      - Main Development
        - Release 1.0
          - Release Backlog [Release 1.0
          - Sprint 2 (1.0)
          - Sprint 1 (1.0)
        - Backlog
  - Reports

Release Backlog

Owner: RTC Lab Project | Iteration: Release 1.0 (2/10/11 - 3/22/11) | 0 Closed | 5 Open 0/0 pts 100%

<Enter additional notes relevant to the plan (start editing)>

b) Click on the tab **Planned Items** at the bottom of the iteration plan window. You should see the following screen with with no work items yet assigned to any users.



Release Backlog [Release 1.0]

Release Backlog

Owner: RTC Lab Project | Iteration: Release 1.0 (2/10/11 - 3/22/11) | 0 Closed | 5 Open 0/0 pts 100%

Release 1.0
Closed items: 0 | Open items: 5 Progress: 0/0 pts Estimated: 100%

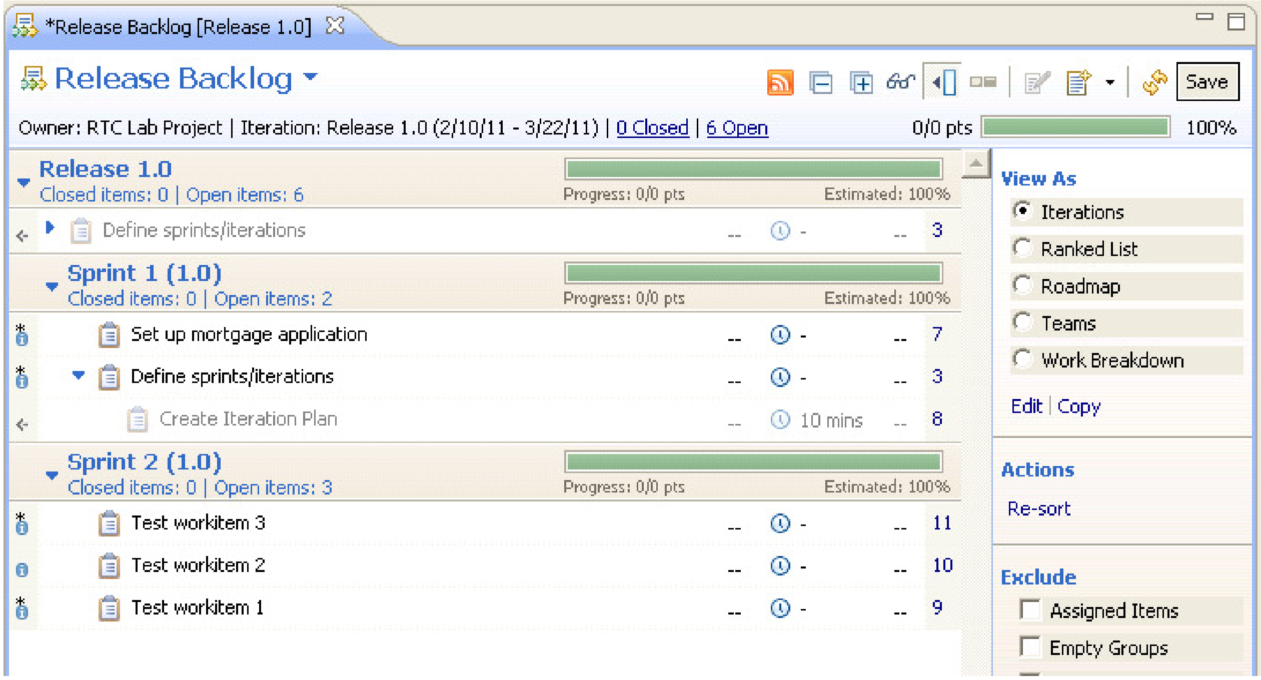
Test workitem 3	--	-	--	11
Test workitem 2	--	-	--	10
Test workitem 1	--	-	--	9
Set up mortgage application	--	-	--	7
Define sprints/iterations	--	-	--	3

Sprint 1 (1.0)
Closed items: 0 | Open items: 0 Progress: 0/0 pts No Work Estimated: --

Sprint 2 (1.0)
Closed items: 0 | Open items: 0 Progress: 0/0 pts No Work Estimated: --

View As
● Iterations
○ Ranked List
○ Roadmap
○ Teams
○ Work Breakdown

c) We are first going to add items in the release into various sprints. So add items to the Sprint 1.0 or Sprint 2.0 plans by dragging them from the Release Work items List to the Sprint plans. Make sure you **save** once you are done. You will then have something that looks like this:



*Release Backlog [Release 1.0]

Release Backlog

Owner: RTC Lab Project | Iteration: Release 1.0 (2/10/11 - 3/22/11) | 0 Closed | 6 Open 0/0 pts 100%

Release 1.0
Closed items: 0 | Open items: 6 Progress: 0/0 pts Estimated: 100%

Define sprints/iterations	--	-	--	3

Sprint 1 (1.0)
Closed items: 0 | Open items: 2 Progress: 0/0 pts Estimated: 100%

Set up mortgage application	--	-	--	7
Define sprints/iterations	--	-	--	3
Create Iteration Plan	--	10 mins	--	8

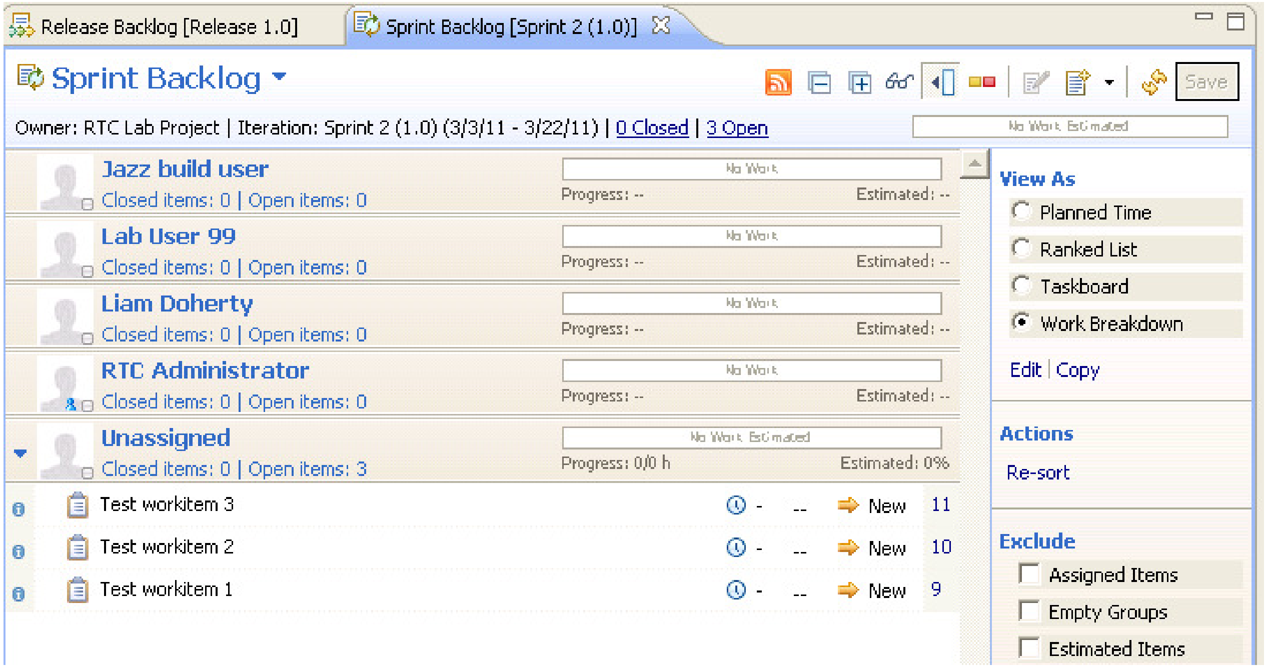
Sprint 2 (1.0)
Closed items: 0 | Open items: 3 Progress: 0/0 pts Estimated: 100%

Test workitem 3	--	-	--	11
Test workitem 2	--	-	--	10
Test workitem 1	--	-	--	9

View As
● Iterations
○ Ranked List
○ Roadmap
○ Teams
○ Work Breakdown

Exclude
☐ Assigned Items
☐ Empty Groups

d) Now we want to assign these to specific users. So double click on Sprint 1.0 or Sprint 2.0, which ever has the most work items in it, and if presented with a screen select the sprint backlog. You will need to now change the **View As** from ***Iterations*** to ***Work Breakdown*** to view work items by owner. You will then see the following unassigned work items listed in the Sprint plan:


Release Backlog [Release 1.0] | Sprint Backlog [Sprint 2 (1.0)]

Sprint Backlog

Owner: RTC Lab Project | Iteration: Sprint 2 (1.0) (3/3/11 - 3/22/11) | 0 Closed | 3 Open — No Work Estimated

Jazz build user — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

Lab User 99 — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

Liam Doherty — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

RTC Administrator — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

Unassigned — Closed items: 0 | Open items: 3 — No Work Estimated — Progress: 0/0 h — Estimated: 0%

Test workitem 3 - -- New 11

Test workitem 2 - -- New 10

Test workitem 1 - -- New 9

View As
○ Planned Time
○ Ranked List
○ Taskboard
● Work Breakdown

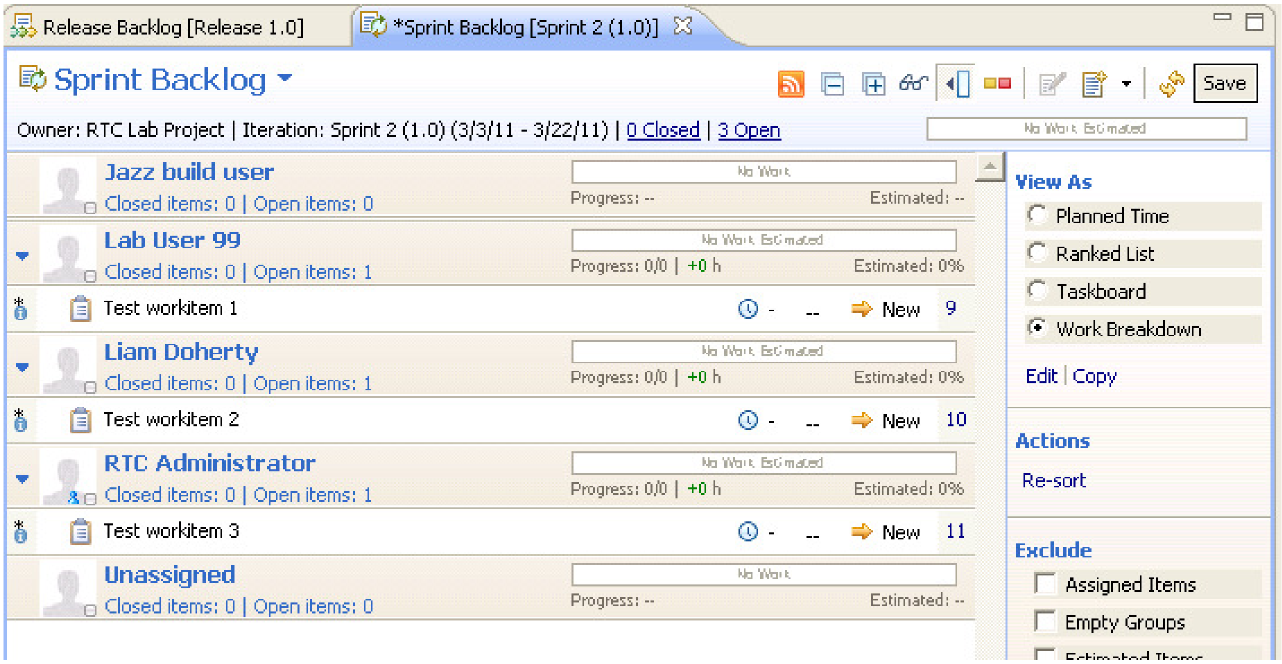
Edit | Copy

Actions

Re-sort

Exclude
□ Assigned Items
□ Empty Groups
□ Estimated Items

e) You can now drag a drop them on the user they should be assigned to. Open the work item and notice that both the **Planned for** and the **assigned to** has changed, save the changes and assign more tasks. Don’t forget to save the plan after adding work items.


Release Backlog [Release 1.0] | *Sprint Backlog [Sprint 2 (1.0)]

Sprint Backlog

Owner: RTC Lab Project | Iteration: Sprint 2 (1.0) (3/3/11 - 3/22/11) | 0 Closed | 3 Open — No Work Estimated

Jazz build user — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

Lab User 99 — Closed items: 0 | Open items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Test workitem 1 - -- New 9

Liam Doherty — Closed items: 0 | Open items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Test workitem 2 - -- New 10

RTC Administrator — Closed items: 0 | Open items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Test workitem 3 - -- New 11

Unassigned — Closed items: 0 | Open items: 0 — No Work — Progress: -- — Estimated: --

View As
○ Planned Time
○ Ranked List
○ Taskboard
● Work Breakdown

Edit | Copy

Actions

Re-sort

Exclude
□ Assigned Items
□ Empty Groups
□ Estimated Items

Conclusion

This concludes the Lab for Module 3. You now know how iteration plans can be created, how to create work items and understand how they fit in the context of plans.

Module 4 Work Item Queries

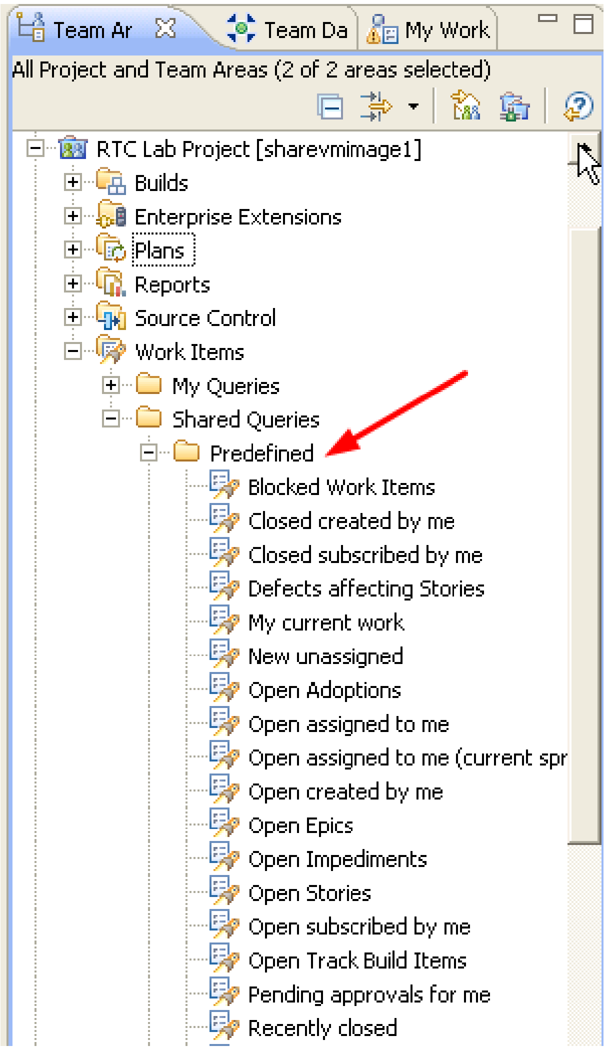
Module 4 Work Item Queries

Lab Scenario

Now that you have a few work items in the server repository, it's time to learn how to search through them. Queries are the search mechanism for Work Items. You can use predefined, queries, share queries amongst your team, or create queries from scratch and save them later.

4.1. Predefined Queries

a) To search for work items, either create a query from scratch or use a predefined one. Try using predefined queries first. We will create queries in the next section. To use predefined queries, in the **Team Artifacts** view, navigate to **RTC Lab Project** → **Work Items** → **Shared Queries** → **Predefined**

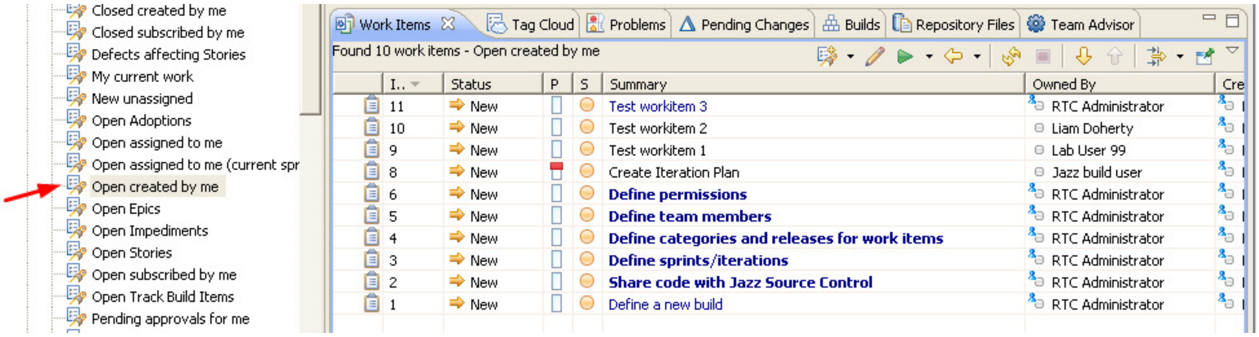


All Project and Team Areas (2 of 2 areas selected)

- RTC Lab Project [sharevmimage1]
  - Builds
  - Enterprise Extensions
  - Plans
  - Reports
  - Source Control
  - Work Items
    - My Queries
    - Shared Queries
      - Predefined
        - Blocked Work Items
        - Closed created by me
        - Closed subscribed by me
        - Defects affecting Stories
        - My current work
        - New unassigned
        - Open Adoptions
        - Open assigned to me
        - Open assigned to me (current spr
        - Open created by me
        - Open Epics
        - Open Impediments
        - Open Stories
        - Open subscribed by me
        - Open Track Build Items
        - Pending approvals for me
        - Recently closed

b) The predefined queries provide a useful set of basic queries that can be used. You can save queries you build to use in the future and even share them with your teammates! It can be useful to have queries that return results for a specific Iteration Plan, with a specific word or tag, to find work items based on any user, and even build complex Boolean statements using AND and OR expressions.

c) Results are displayed in the **Work Item** view as seen below when running the predefined query **Open created by me** by double clicking it. To open the **Work Item** view, go to **Window** → **Show View** → **Work Items,** but by running the query the view should be displayed automatically.

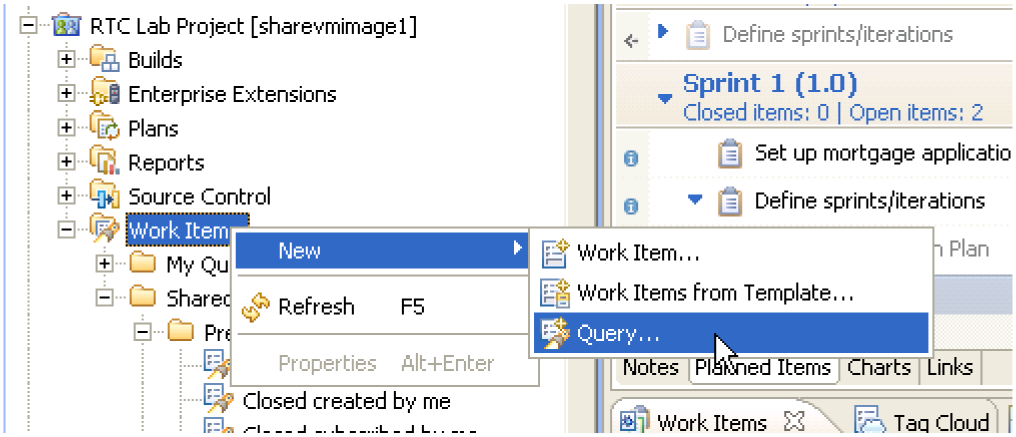


Found 10 work items - Open created by me

I..	Status	P	S	Summary	Owned By	Cre
11	New			Test workitem 3	RTC Administrator	
10	New			Test workitem 2	Liam Doherty	
9	New			Test workitem 1	Lab User 99	
8	New			Create Iteration Plan	Jazz build user	
6	New			Define permissions	RTC Administrator	
5	New			Define team members	RTC Administrator	
4	New			Define categories and releases for work items	RTC Administrator	
3	New			Define sprints/iterations	RTC Administrator	
2	New			Share code with Jazz Source Control	RTC Administrator	
1	New			Define a new build	RTC Administrator	

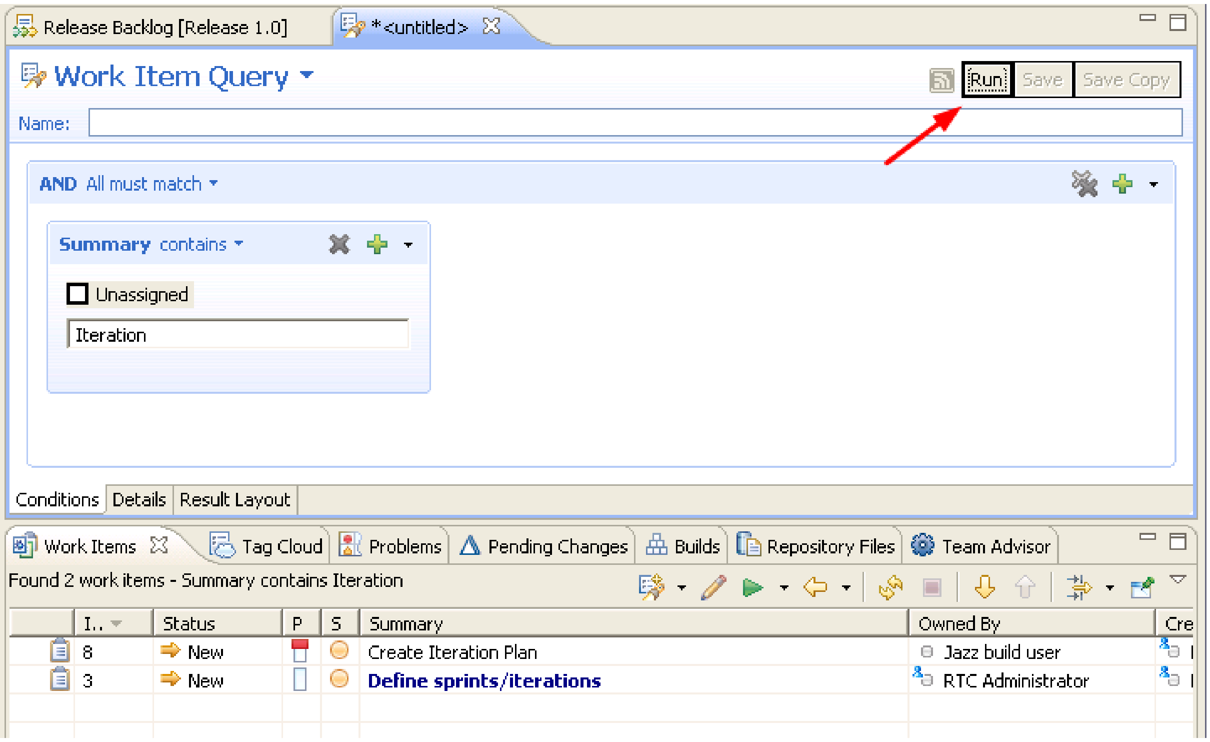
4.2. Creating Queries from scratch

a) Right-click on Work Items and select **New → Query.**



b) Click the link to **Start from Scratch.** Name your query something that describes what it searches for so you will know what it does. As an example we will create a query to search the summary text of a work item for the string 'Iteration'.

- Use the ✚ sign in the upper-right corner to add a new condition. You can add as many as you need. .
- In the **Add Conditions…** screen select **Summary**
- In the box provided enter the word **Iteration**. This is case sensitive so be careful. Click the Run button to see the results.



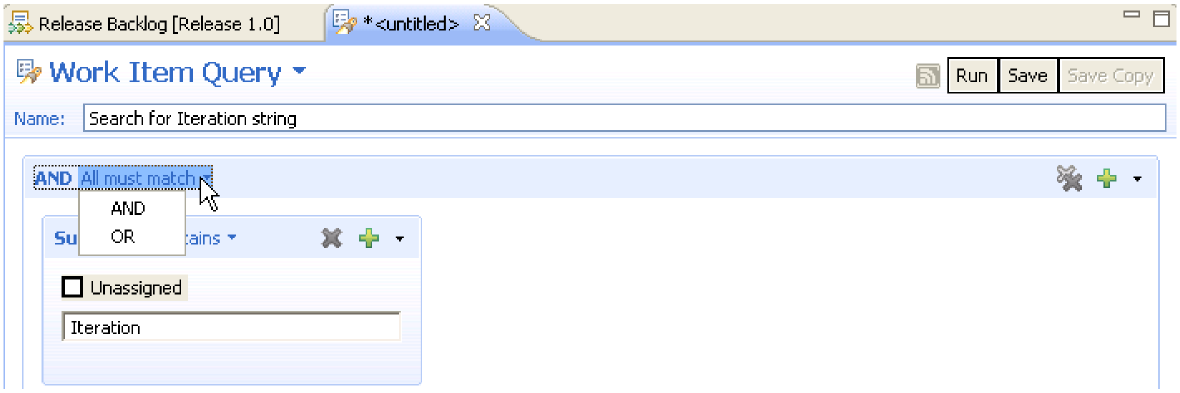
☐ Unassigned

Iteration

Found 2 work items - Summary contains Iteration

I..	Status	P	S	Summary	Owned By	Cre
8	New			Create Iteration Plan	Jazz build user	
3	New			**Define sprints/iterations**	RTC Administrator	

c) Try creating a more interesting query that searches based on more than one condition. By default, the conditions are added with a Boolean AND but you can change this to a Boolean OR by selecting OR from the dropdown menu as seen below.



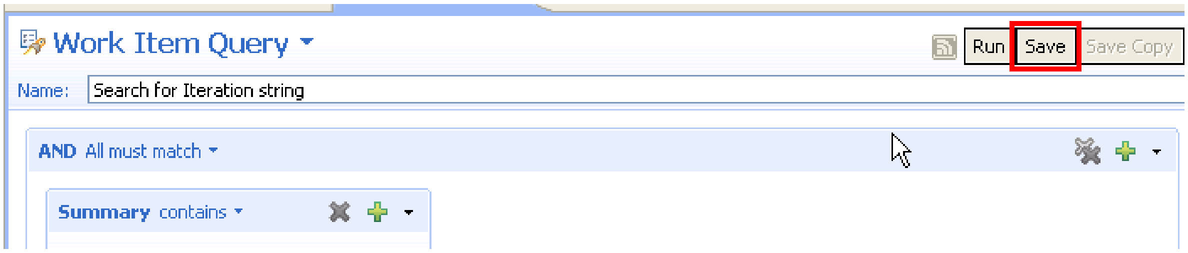
Name: Search for Iteration string

☐ Unassigned

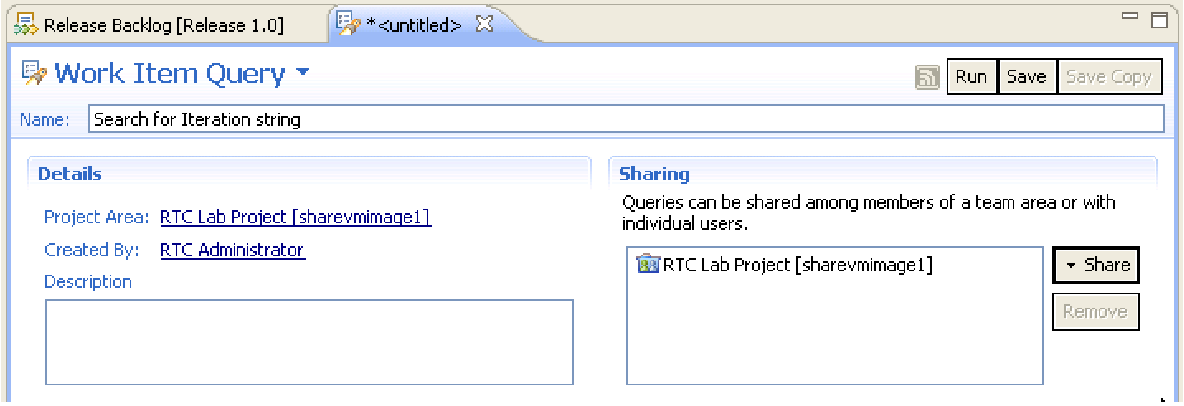
Iteration

d) Try creating complex queries with multiple conditions. When using queries, it is important to keep in mind that searches for text return true if the work item field *contains* the search text. There is no need to insert wildcards before and after search text.

e) To **save** a query, simply click on the **Save** button in the upper-right corner as seen below (the query must be named before it can be saved). Once a query is saved, it will appear in the **My Queries** subfolder as a child of Work Items.

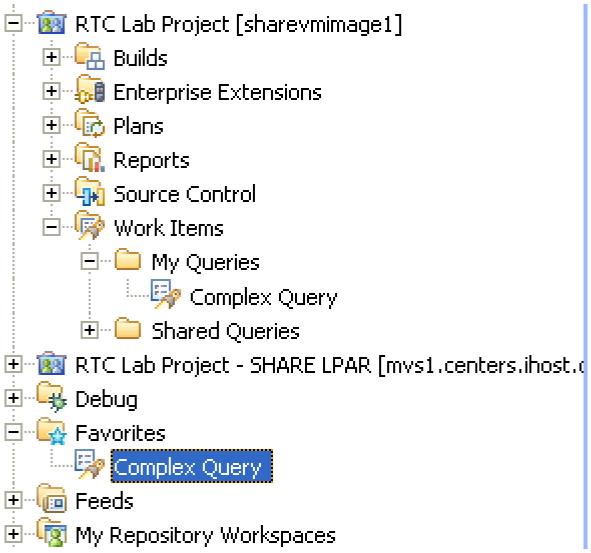


f) To share a query amongst team members, go to the **Details** of the query as seen below with the Complex query example.


Release Backlog [Release 1.0]
*<untitled>
Work Item Query
Run
Save
Save Copy
Name: Search for Iteration string
Details
Project Area: RTC Lab Project [sharevmimage1]
Created By: RTC Administrator
Description
Sharing
Queries can be shared among members of a team area or with individual users.
RTC Lab Project [sharevmimage1]
Share
Remove

g) Click on **Share** → **Team or Project Area** and then choose **RTC Lab Project.** Your query is now shared amongst the team.

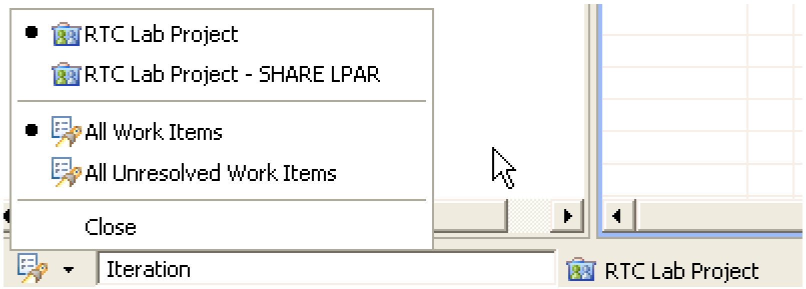
h) To further keep track of your work, you can add the new query to your **Favorites** folder. The **Favorites** folder is used to organize queries, plans and other artifacts that are important for your work. Drag the **Complex Query** query to the **Favorites** folder in the **Team Artifacts** view.



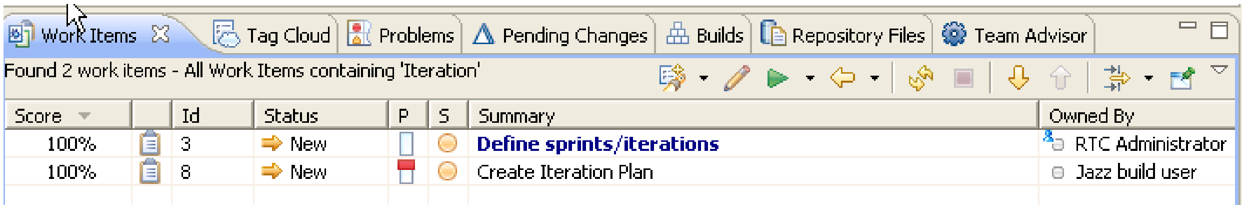
i) It is not necessary to have the **Team Artifacts** view open to access the **Favorites** folder. At any time, press **Ctrl-Alt-F** (press the three keys simultaneously) and a popup dialog will appear with your favorites.

j) In the **Favorites** or **My Queries** folder, double-click your query titled **Complex Query**. The resulting work items are displayed in the **Work Items** view (That should be by default displayed at the bottom of the Eclipse client).

k) Experiment with quick search. In the bottom-left corner of the Eclipse client interface you find a drop-down list, expand it and select **All Work Items**.



l) In the text field next to the drop down list, type in a word that appears in the title of several work items, for example type `Iteration` and press **Enter**. The result of this query is displayed in the **Work Items** view.



Found 2 work items - All Work Items containing 'Iteration'

Score		Id	Status	P	S	Summary	Owned By
100%		3	New			**Define sprints/iterations**	RTC Administrator
100%		8	New			Create Iteration Plan	Jazz build user

Quick search

Try typing in a partial word (rat). Note that partial words do not match. Jazz uses a whole word indexer for work items for fast text queries. Try typing in a phrase that appears in a work item description somewhere. Note that the quick search does a full text search too.

Conclusion

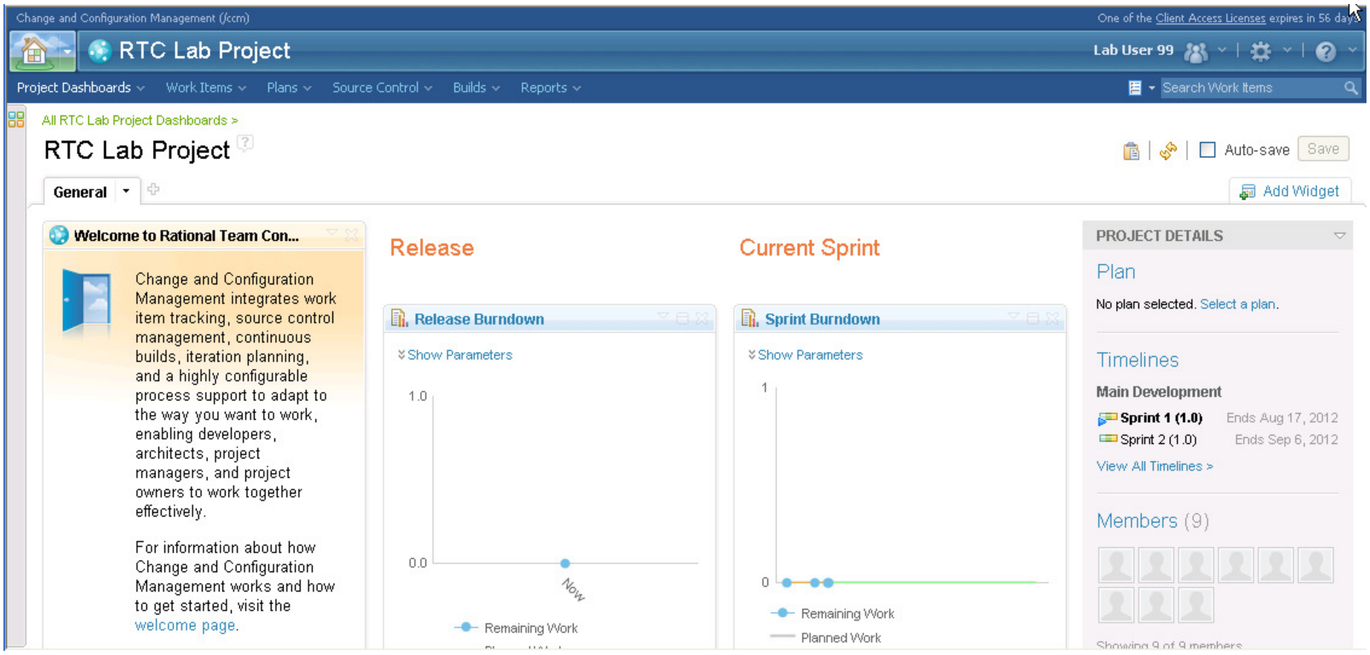
This concludes the Lab for Module 4. You now know how to use predefined queries, create queries of your own, save them, add them to your favorites, and use the quick search to find work items.

Module 5 Web User Interface

5.1. Web User Interface: Work Items, Iteration Plans, Reports, Dashboards

a) The RTC web interface is another way to access work items, queries, and plans. To access the web interface, open Internet Explorer and go to [https:// sharexp1:9443/jts/web](https://sharexp1:9443/jts/web) (also bookmarked as RTC Web interface) and login as user/password **labuser99/labuser99** (or any valid user such as the user you created). You may need to get past a certificate error. If this is the case select Continue to this website.

b) **Project Dashboards** are customizable collections of viewlets which are meant to be a quick summary of important data. By default you are shown your personal dashboard which was created for you. There is also an RTC Lab Project Dashboard which was created by default when the project area was created. Try clicking the RTC Lab Project in the My Projects box to display it. Experiment with the dashboard by adding new tabs and adding viewlets to those tabs.



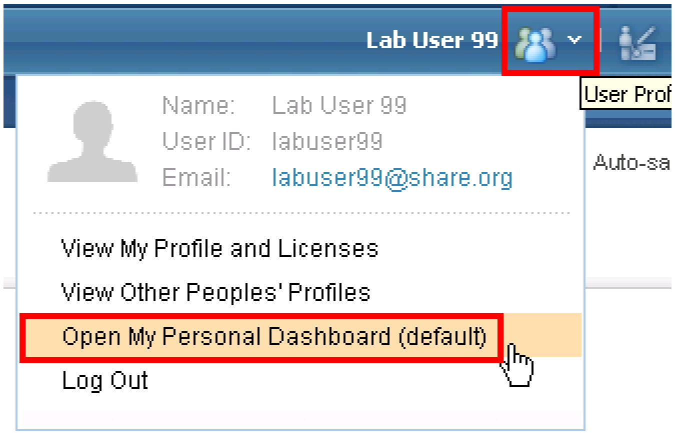
One of the Client Access Licenses expires in 56 days

Change and Configuration Management integrates work item tracking, source control management, continuous builds, iteration planning, and a highly configurable process support to adapt to the way you want to work, enabling developers, architects, project managers, and project owners to work together effectively.

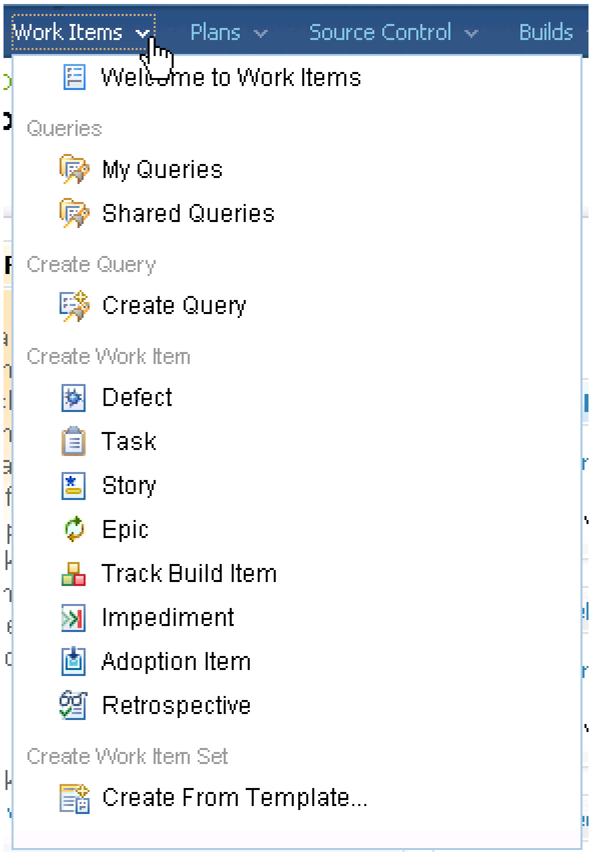
For information about how Change and Configuration Management works and how to get started, visit the welcome page.

No plan selected. Select a plan.

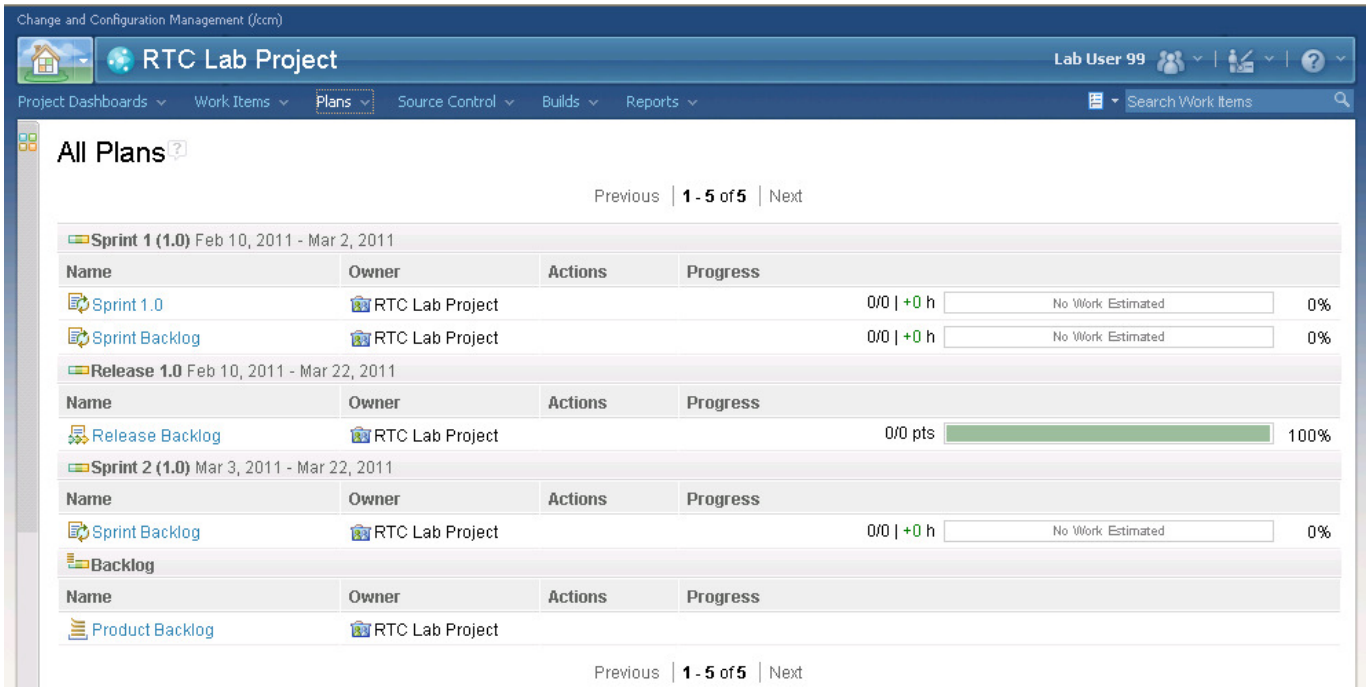
To get back to your own personal dashboard click on the drop down arrow next to your user name at the top right of the screen and then select Open My Personal Dashboard. However this is just for information; do not select your dashboard for now.


Lab User 99
User Prof
Name: Lab User 99
User ID: labuser99
Email: labuser99@share.org
Auto-sa
View My Profile and Licenses
View Other Peoples' Profiles
Open My Personal Dashboard (default)
Log Out

c) The next link on the top is **Work Item.** Click on it and you will see options to search for a work item, create a new work item, create a new query, and execute predefined or user-defined saved queries. Notice the queries created previously in the client will display in the web interface too. This is because both interfaces point to the same data. Creating a work item and query is so similar to the instructions in the client that it will not be covered in the web interface section.



d) The next link at the top is **Plans.** Click on it and select **all plans** to see a list of the iteration plans associated with this project area as seen below.



All Plans

Previous | 1 - 5 of 5 | Next

Sprint 1 (1.0) Feb 10, 2011 - Mar 2, 2011

Name	Owner	Actions	Progress		
Sprint 1.0	RTC Lab Project		0/0 | +0 h	No Work Estimated	0%
Sprint Backlog	RTC Lab Project		0/0 | +0 h	No Work Estimated	0%

Release 1.0 Feb 10, 2011 - Mar 22, 2011

Name	Owner	Actions	Progress		
Release Backlog	RTC Lab Project		0/0 pts		100%

Sprint 2 (1.0) Mar 3, 2011 - Mar 22, 2011

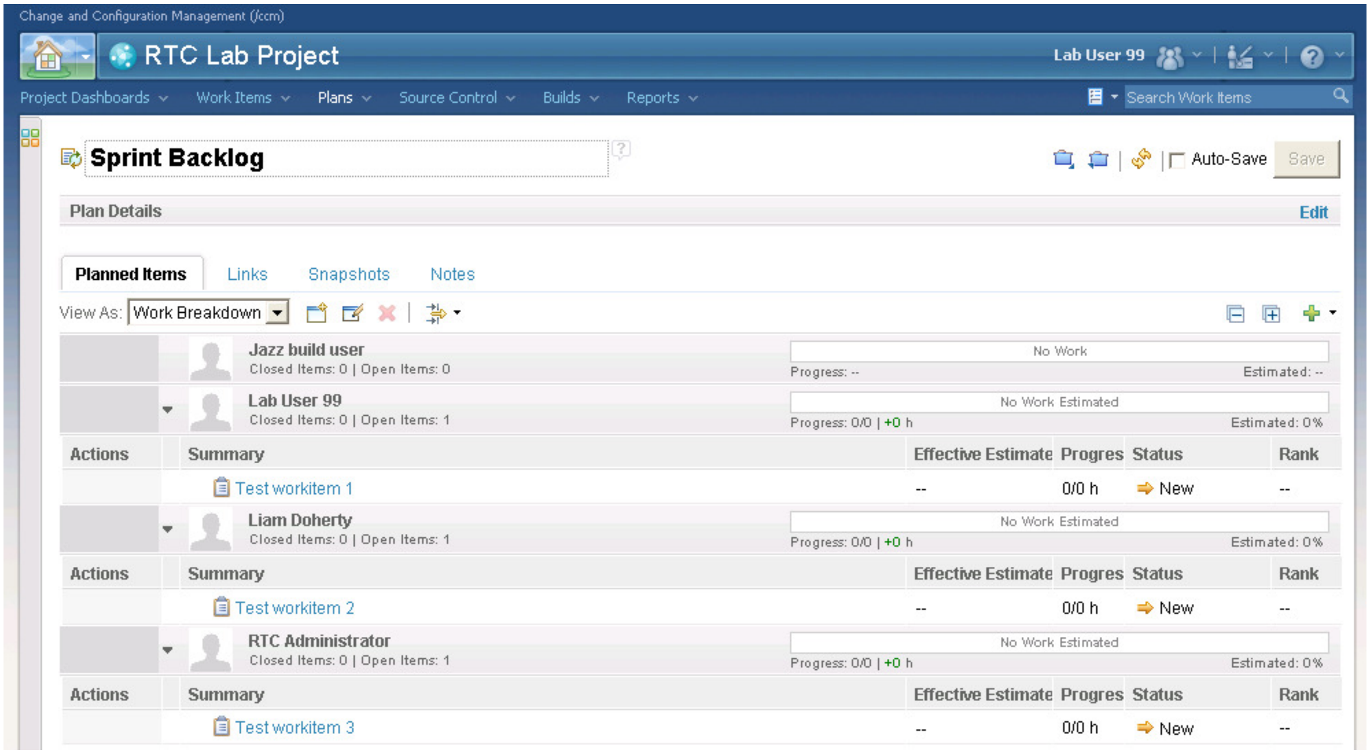
Name	Owner	Actions	Progress		
Sprint Backlog	RTC Lab Project		0/0 | +0 h	No Work Estimated	0%

Backlog

Name	Owner	Actions	Progress
Product Backlog	RTC Lab Project		

Previous | 1 - 5 of 5 | Next

e) Click on the **Sprint 2 → Sprint Backlog** plan to see the **Planned Items** as highlighted below to see a breakdown of the tasks, defects, enhancements, etc. associated with this project area.



Sprint Backlog

Plan Details

View As: Work Breakdown

Jazz build user
Closed Items: 0 | Open Items: 0 — No Work — Progress: -- — Estimated: --

Lab User 99
Closed Items: 0 | Open Items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Actions	Summary	Effective Estimate	Progres	Status	Rank
	Test workitem 1	--	0/0 h	New	--

Liam Doherty
Closed Items: 0 | Open Items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Actions	Summary	Effective Estimate	Progres	Status	Rank
	Test workitem 2	--	0/0 h	New	--

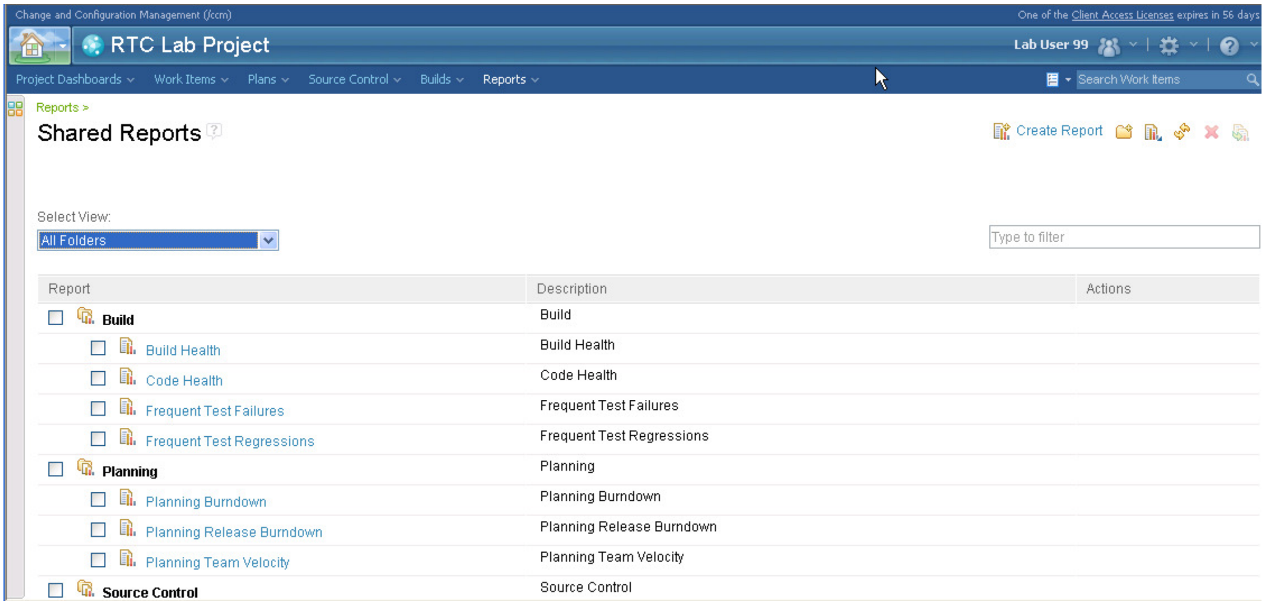
RTC Administrator
Closed Items: 0 | Open Items: 1 — No Work Estimated — Progress: 0/0 | +0 h — Estimated: 0%

Actions	Summary	Effective Estimate	Progres	Status	Rank
	Test workitem 3	--	0/0 h	New	--

f) The **Source Control** link at the top is where you can find history for streams and components. We will cover this more extensively in the source control section. Source control access is provided in the web for viewing of artifacts.

g) The **Build** link at the top is where you can look at the existing build definitions, build queues and build engines, look at the results of builds that have run, and submit a build. Builds will be discussed later in the lab.

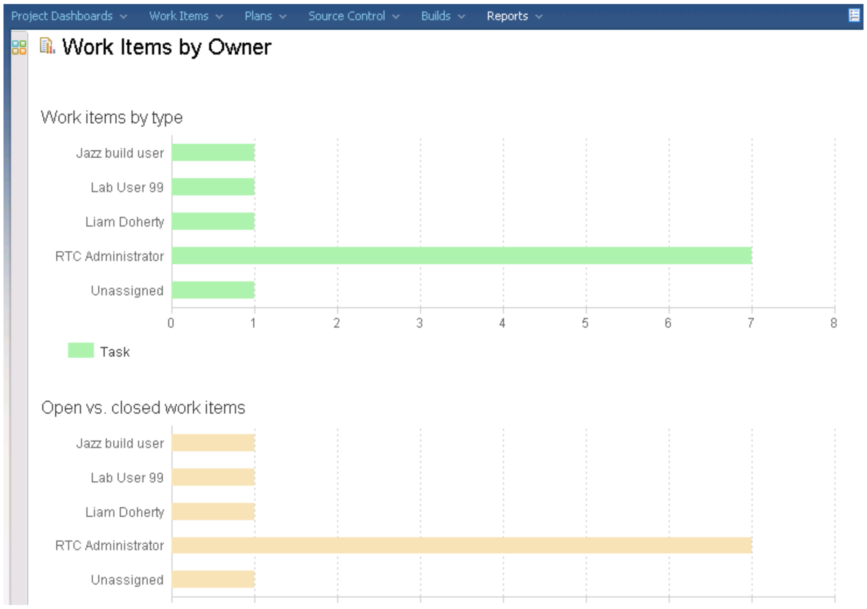
h) Click on the **Reports** link at the top of the page. Select the **Shared Reports** link. On the left there are several predefined reports.



Shared Reports

Report	Description	Actions
□ **Build**	Build	
□ Build Health	Build Health	
□ Code Health	Code Health	
□ Frequent Test Failures	Frequent Test Failures	
□ Frequent Test Regressions	Frequent Test Regressions	
□ **Planning**	Planning	
□ Planning Burndown	Planning Burndown	
□ Planning Release Burndown	Planning Release Burndown	
□ Planning Team Velocity	Planning Team Velocity	
□ **Source Control**	Source Control	

Play around with some of the reports, for example the **Work Items by owner** report



Conclusion

This concludes Module 5. You now should be familiar with the functions available in the web user interface. As you can see this is a valid alternative interface for those users who don't need to work with source control, or do not want the heavy Eclipse client.

Module 6 Source Code Management (SCM)

6.1. Source Code Management with RTC

Some Jazz Terms Defined:

<u>Project Area</u>

The project area is the system's representation of a software project. The project area provides definition for the project deliverables, team structure, process, and schedule.

<u>Team Area</u>

The structure of the project teams is defined by one or more team areas. A team area serves these functions:

1. *Defines the users (team members) on the team and specify their roles.*
2. *Defines the development line in which the team is participating.*
3. *Customizes the project process for the team.*

<u>Streams and Components</u>

A stream is a repository object that includes one or more components. A component is a collection of related artifacts, such as an Application, an Eclipse plug-In, or a group of documents that comprise Web site content.

<u>Change Set</u>

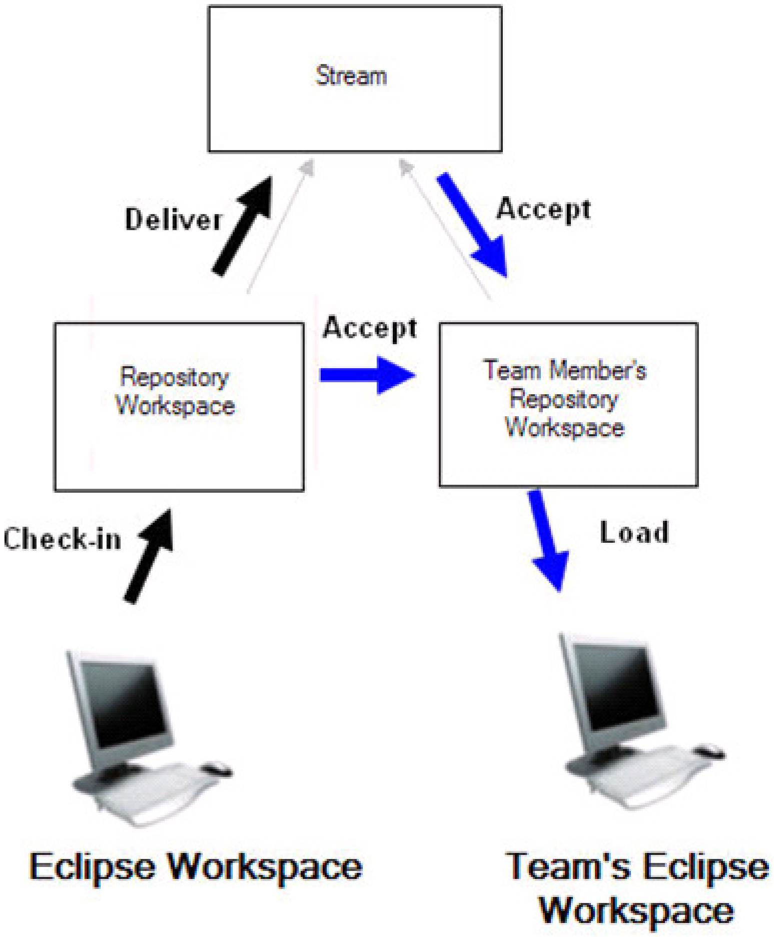
A change set is a repository object that collects a related group of changes so that they can be applied to a flow target (workspace or stream) in a single operation.

The change set is the fundamental unit of change in Jazz source control. The contents of any workspace, component, or stream can be expressed as a collection of change sets, beginning with the one created when the initial set of projects was checked in. A change set can include changes to the contents of individual files and changes to a component namespace (such as delete, rename, and move operations).

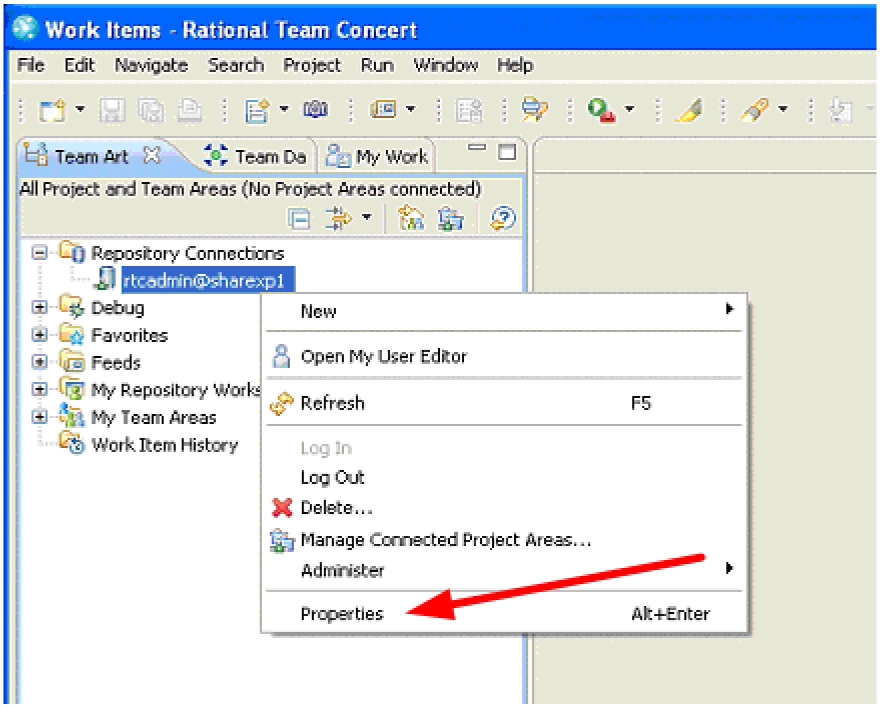
The source control component handles the storing, retrieving and sharing of source code and other artifacts in your project. It is important to understand the terminology and relationships involved in any model so we will give some background information on managing source code with RTC including terminology.

In Rational Team Concert, we want users to benefit from SCM's ability to track and version your changes, whether or not you are ready to share those changes with your team. Accordingly, as a user you have your own private **repository workspace** which stores the changes you've made, regardless of when you decide to make them available to your team. Sometimes it will be only a couple of hours before you decide to share your changes; sometimes it will be longer. You decide when to make the changes available to your team. When you **load** your repository workspace, the files and folders in it are transferred to your Eclipse workspace on your computer. To push a change from your Eclipse workspace to the repository workspace you **check-in** the change.

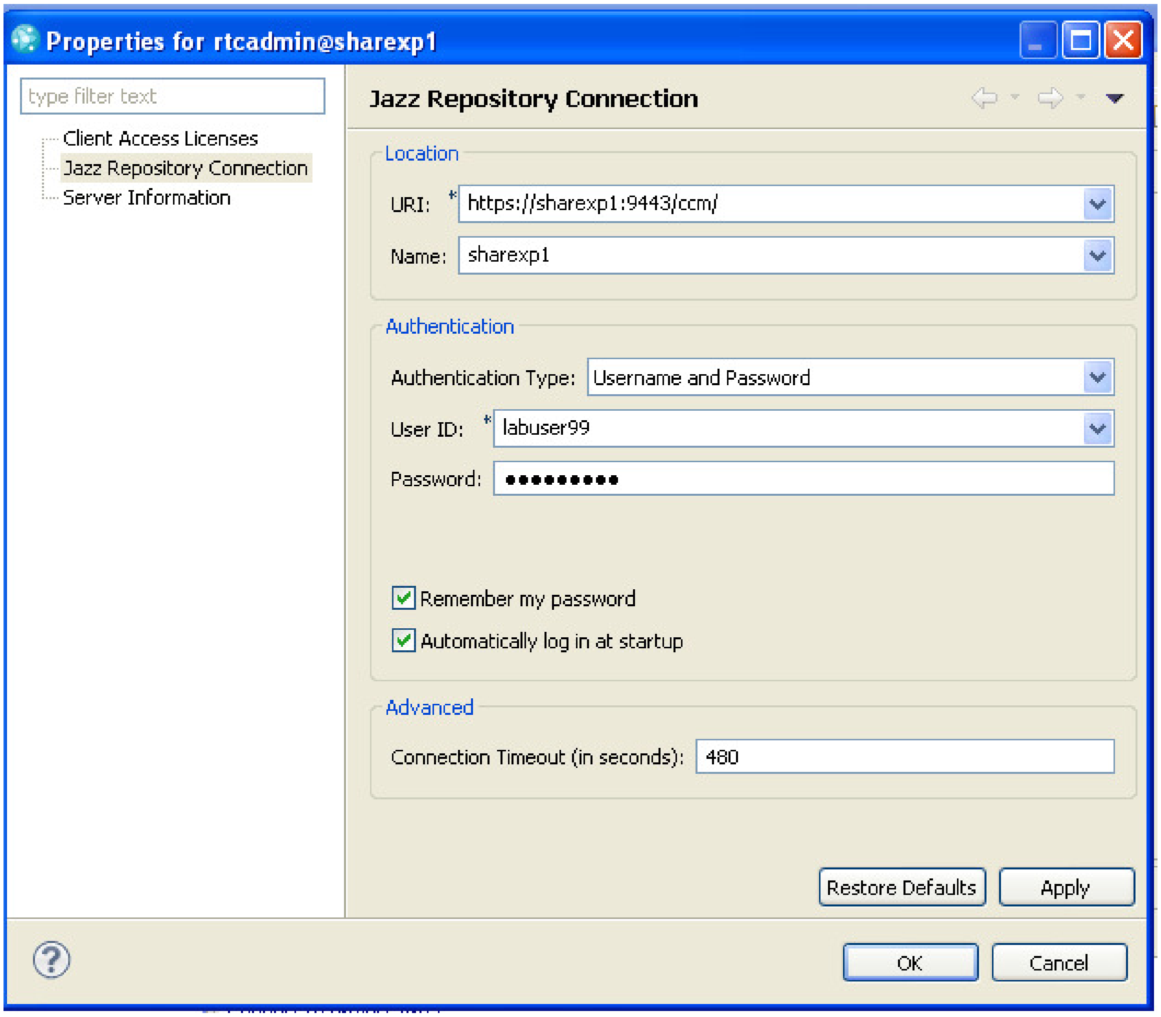
A **stream** is used to store the team's work. When you want to make your changes available to your team, you **deliver** them from a repository workspace to a stream. When you wish to incorporate other team members' changes, you **accept** them from the stream (Note that you can also accept changes directly from another repository workspace, allowing for fine-grained sharing of changes between team members). For example, two team members might collaborate on a small bug fix; or, if someone starts a change and has to go on vacation, another team member could continue the work and then deliver it later. All changes you make in your repository workspace are tracked within change sets. Each change set is composed of a collection of explicit, primitive changes to one or more files or folders. The following picture shows the source control process.



a) Ensure you are logged into RTC. If you created a user in Module 1 then you can use that user. Alternatively you can use labuser99 which has been created for you already. To login right click on the labuser99@sharexp1 node in the Repository connections and select **login**. Or if using your own userid right click on labuser99 and select **Properties** as shown below



b) In the dialog box select the Jazz Repository Connection on the left side of the screen. Then change the user id and password to the one you created previously (labuser#). Alternatively use the **labuser99** id that already exists. The password is the same as the user id. Press **OK**.



Jazz Repository Connection

Location

URI: * https://sharexp1:9443/ccm/

Name: sharexp1

Authentication

Authentication Type: Username and Password

User ID: * labuser99

Password: ●●●●●●●●●●

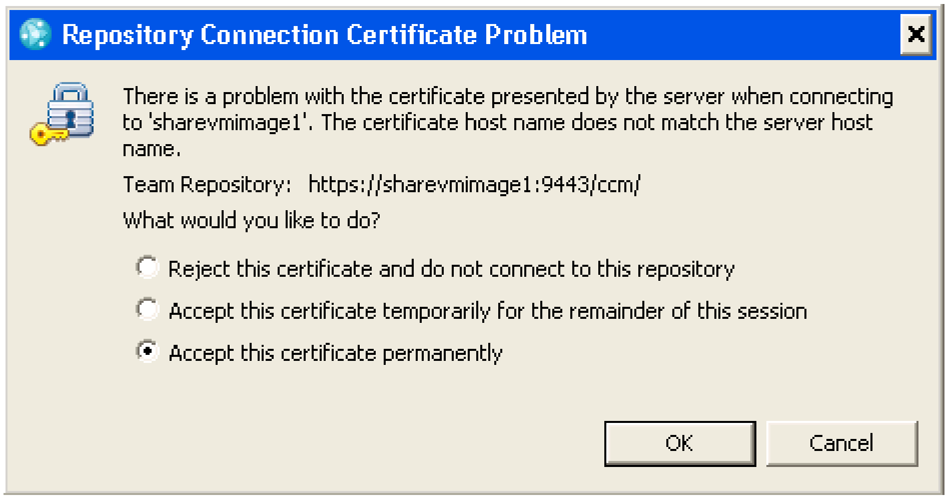
☑ Remember my password

☑ Automatically log in at startup

Advanced

Connection Timeout (in seconds): 480

c) If you get any certificate problems when connecting, just accept the certificate permanently:



Repository Connection Certificate Problem

There is a problem with the certificate presented by the server when connecting to 'sharevmimage1'. The certificate host name does not match the server host name.

Team Repository: https://sharevmimage1:9443/ccm/

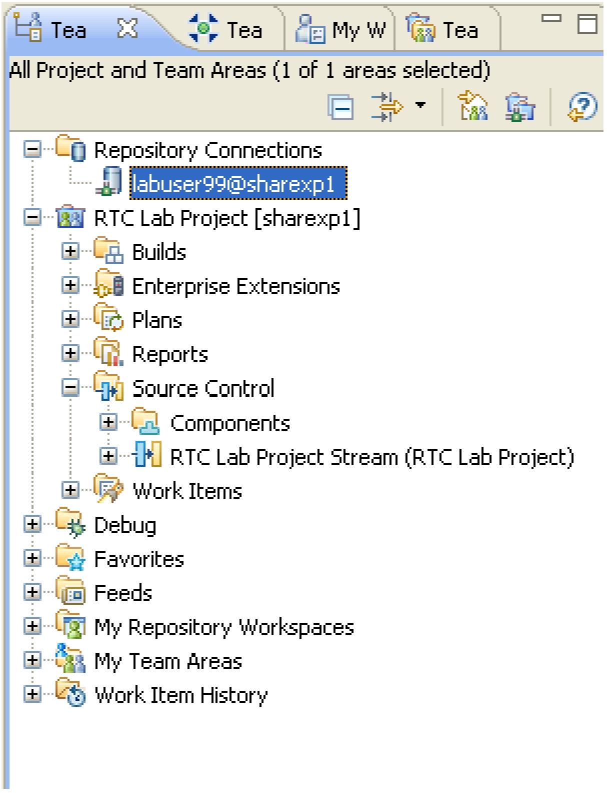
What would you like to do?

○ Reject this certificate and do not connect to this repository

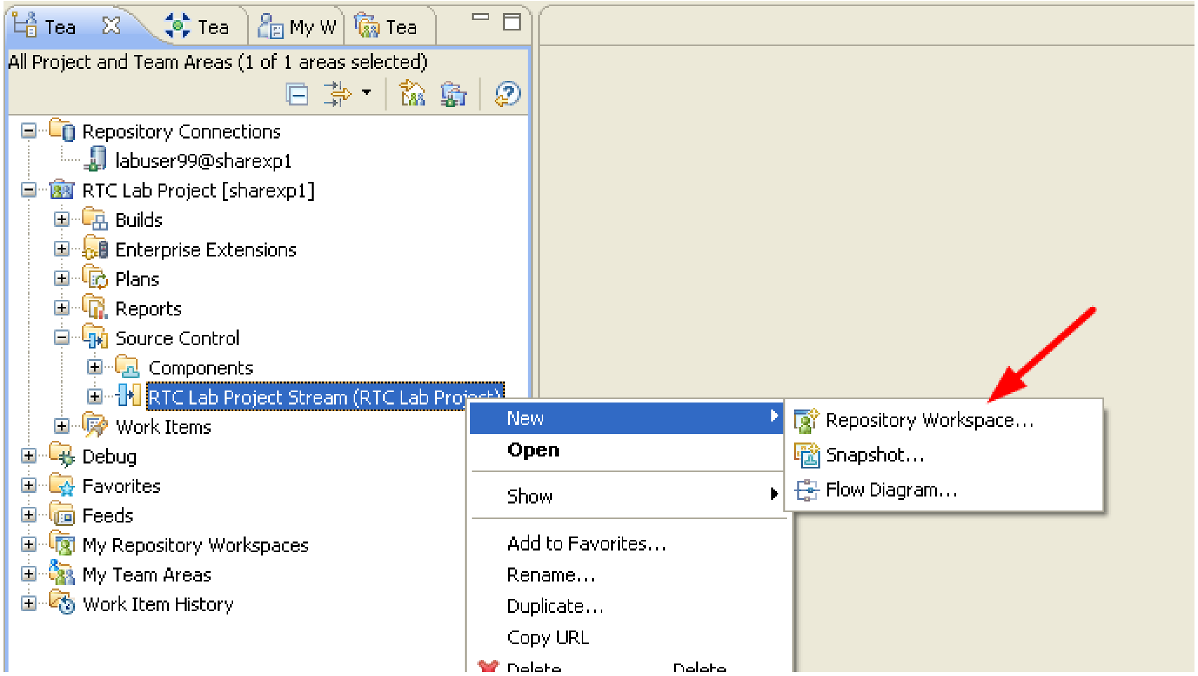
○ Accept this certificate temporarily for the remainder of this session

● Accept this certificate permanently

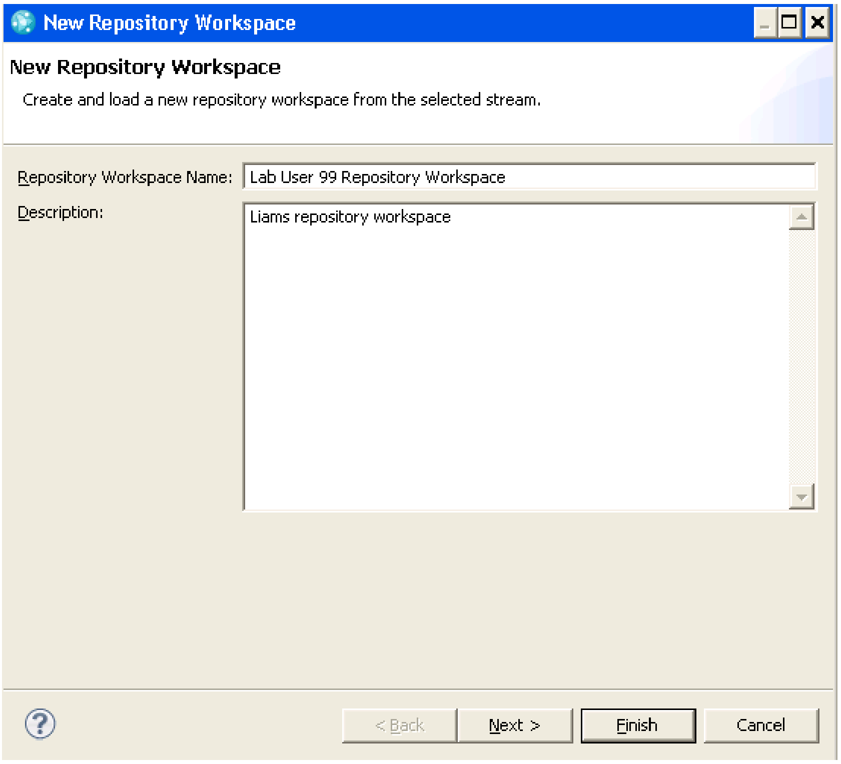
d) First, we are going to create a local **repository workspace,** to work with files in the repository. To do this, first locate your team stream for the project area **RTC Lab Project**. From the **Team Artifacts View**, expand **RTC Lab Project** project area and then expand the **Source Control** node. You should see a team stream named **RTC Lab Project Stream**.



e) Right-click on the **RTC Lab Project Stream** and select **New** → **Repository Workspace**.



f) Give the **repository workspace** a name such as **Lab User # Repository Workspace** where **#** is your user number and click **Next.**



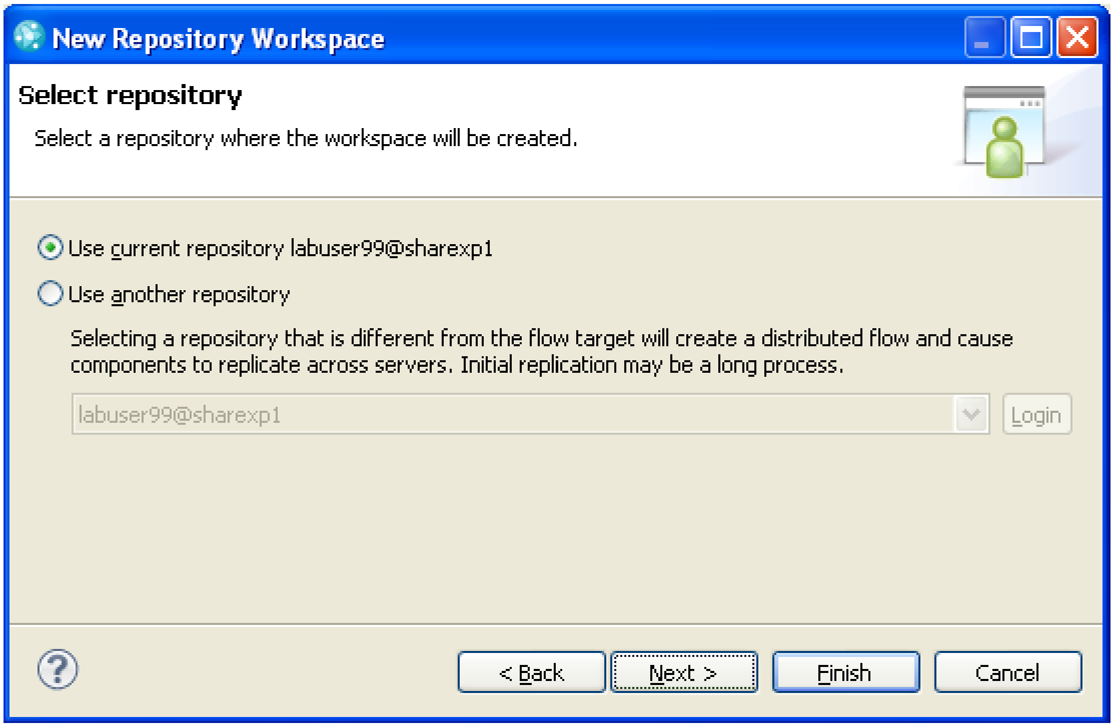
New Repository Workspace

Create and load a new repository workspace from the selected stream.

Repository Workspace Name: Lab User 99 Repository Workspace

Description: Liams repository workspace

g) Select the current repository for the workspace



Select repository

Select a repository where the workspace will be created.

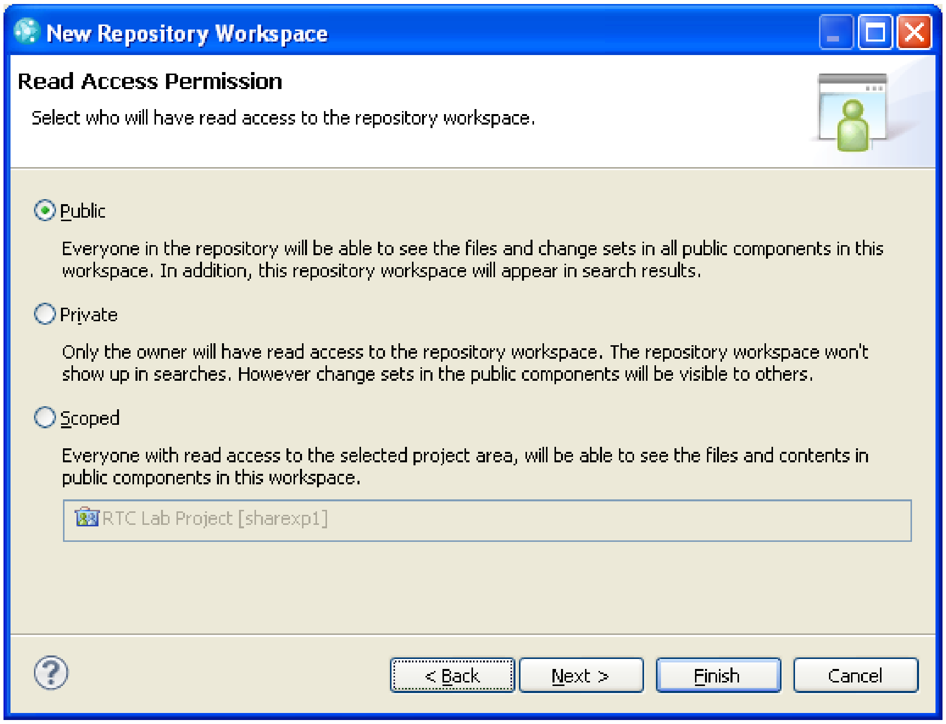
● Use current repository labuser99@sharexp1

○ Use another repository

Selecting a repository that is different from the flow target will create a distributed flow and cause components to replicate across servers. Initial replication may be a long process.

labuser99@sharexp1

h) Next select the read access permissions, for the lab, make the workspace public, then click next..



Read Access Permission

Select who will have read access to the repository workspace.

● Public

Everyone in the repository will be able to see the files and change sets in all public components in this workspace. In addition, this repository workspace will appear in search results.

○ Private

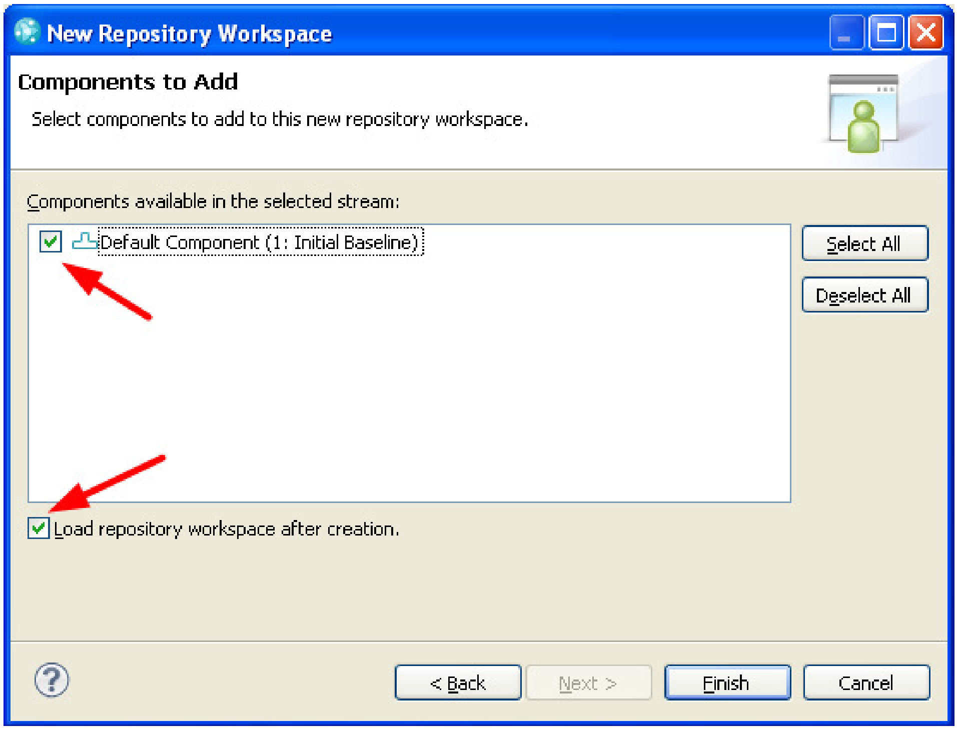
Only the owner will have read access to the repository workspace. The repository workspace won't show up in searches. However change sets in the public components will be visible to others.

○ Scoped

Everyone with read access to the selected project area, will be able to see the files and contents in public components in this workspace.

RTC Lab Project [sharexp1]

i) Make sure all components are selected and load the repository workspace after creation is checked, click Finish.



Components to Add

Select components to add to this new repository workspace.

Components available in the selected stream:

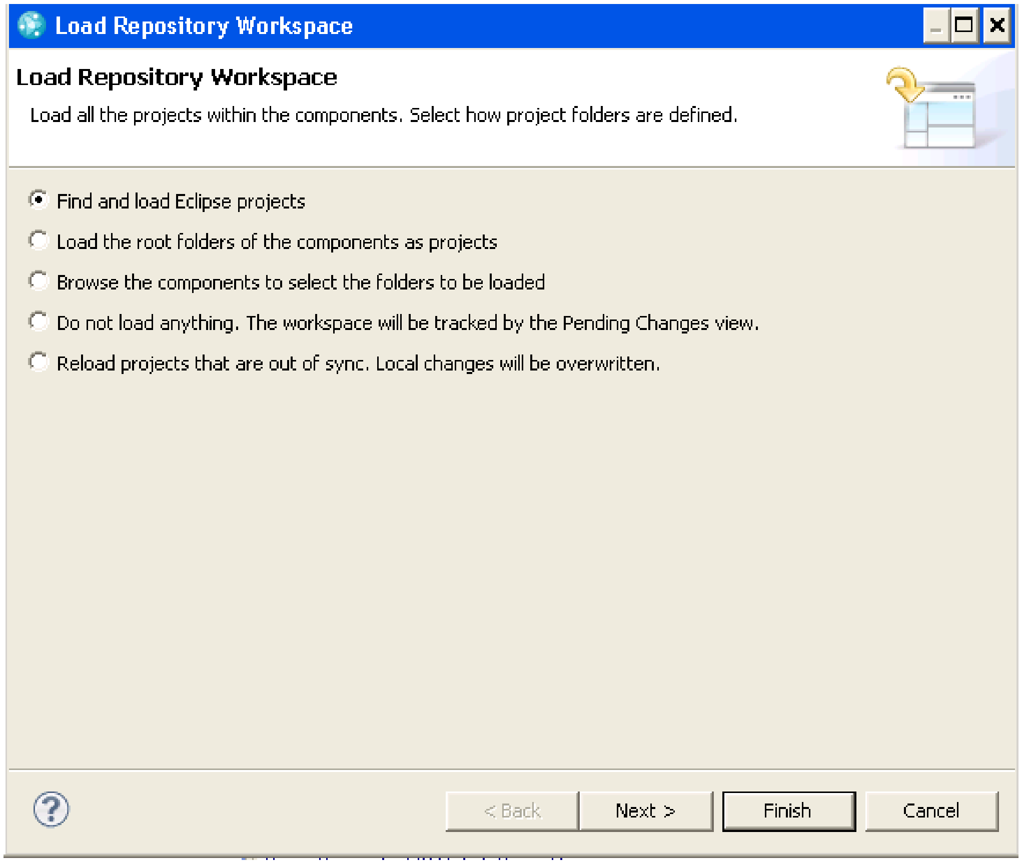
☑ Default Component (1: Initial Baseline)

☑ Load repository workspace after creation.

Some Jazz terms reviewed

*You use a personal **Repository Workspace** to work on project files under **Source Control**. You **Load** the repository workspace to copy the files and folders into your local workspace (Eclipse) on your computer. Jazz tracks all changes made to source-controlled files with **Change-Sets**. Each change-set itemizes the changes to one or more individual files or folders, carries a comment, and references the relevant work item motivating the changes. You **Check-in** your change-sets to upload copies of the modified files from your local workspace to the **repository workspace.***

j) You should now get a prompt for loading projects into your repository workspace. Make sure **Find and Load Eclipse Projects** is selected, and then click **Finish**. The components were already loaded as the original user of RTCadmin, so you will probably get a message asking which components to overwrite, select them all and then **Finish.**



Load Repository Workspace

Load Repository Workspace

Load all the projects within the components. Select how project folders are defined.

● Find and load Eclipse projects

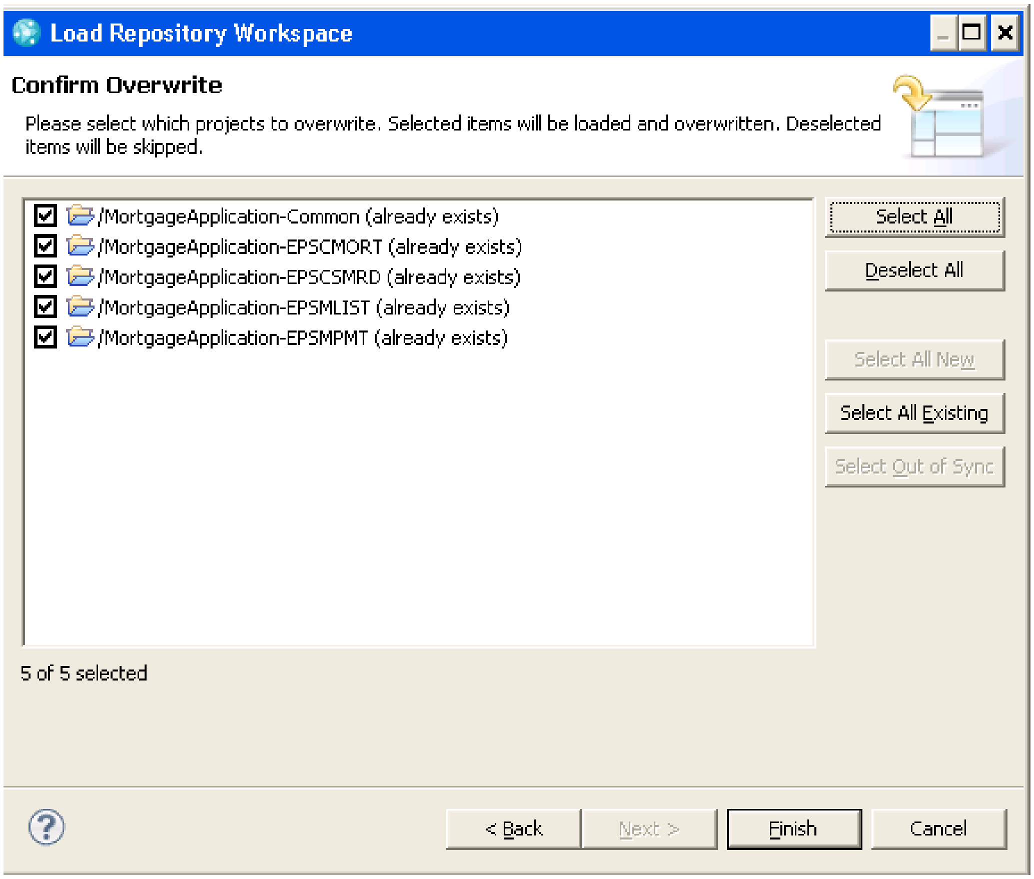
○ Load the root folders of the components as projects

○ Browse the components to select the folders to be loaded

○ Do not load anything. The workspace will be tracked by the Pending Changes view.

○ Reload projects that are out of sync. Local changes will be overwritten.

k) If you get a request to overwrite then select all the projects and press Finish.



Confirm Overwrite

Please select which projects to overwrite. Selected items will be loaded and overwritten. Deselected items will be skipped.

- ☑ /MortgageApplication-Common (already exists)
- ☑ /MortgageApplication-EPSCMORT (already exists)
- ☑ /MortgageApplication-EPSCSMRD (already exists)
- ☑ /MortgageApplication-EPSMLIST (already exists)
- ☑ /MortgageApplication-EPSMPMT (already exists)

5 of 5 selected

l) A loaded repository workspace will appear in the Pending Changes View. Loaded repository workspaces are special in that whenever you change a loaded file or folder in your Eclipse workspace, the changes are tracked and shown in the Pending Changes view. Here, you can manage your changes and perform common tasks such as

- Check-in changes to your repository workspace.
- Organize changes into change sets.
- Undo changes you've made.
- Associate change sets with work items.

If this view is not visible, go to **Window** → **Show View** → **Other** → **Jazz Source Control** → **Pending Changes**.

Note: Loaded means that the code in the repository has been copied down to your PC into a folder so that you can work on it. A loaded component is identified by the decorator. If you have a component in your Repository workspace that is not loaded then it is not filled out in blue, thus: .

Other change-set operations:

***Suspending a change set** (⏸ Suspend)*

*There might be times when you need to begin working on a new change set for a given set of items before you are finished with one that is in progress in your workspace. When in this situation, you can **suspend** the current change set, which removes it from your workspace but preserves it in the repository. The files in a suspended change set revert to the state they were in before the change set started, and the change set itself is moved to a special Suspended folder so that you can **resume** the work when you are ready.*

***Discarding a change set** (Discard...)*

There are two basic scenarios for discarding a change set:

Appendix A. If you have accepted an incoming change set but decide later that you don't want it in your workspace, you can discard it to undo the accept and return the change set to the component's Incoming folder.

Appendix B. You can also discard a change set that you created but have not yet delivered. Discarded change sets of this type remain in the repository but are not placed in any special folder. To make it easier to retrieve a discarded change set that does not exist in any other stream or workspace, you can associate it with a work item before you discard it and then accept it from the work item later.

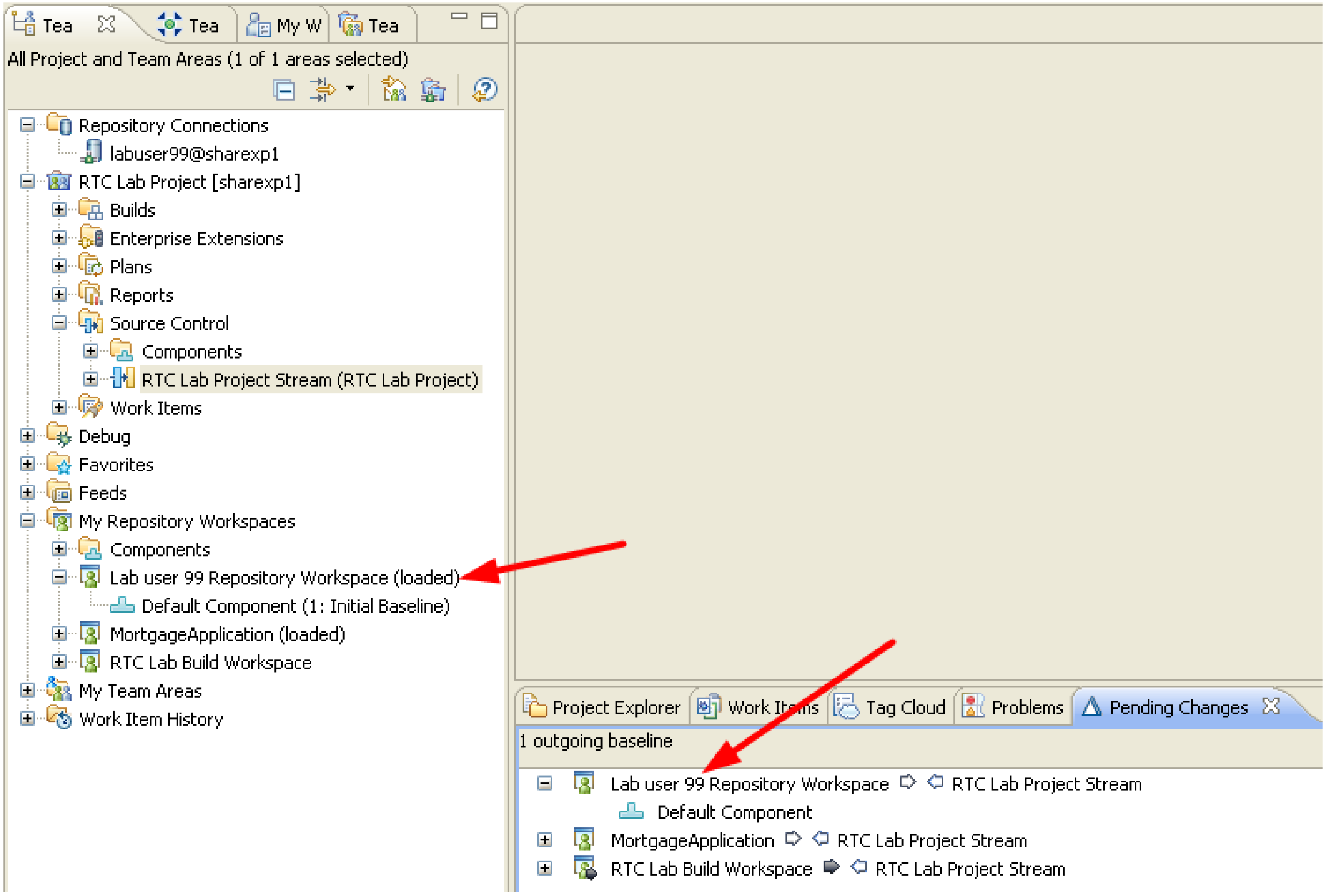
***Reversing a change set** (Reverse)*

If you want to undo the delivery of a change set, you can create a new change set that reverses all the changes in it and then deliver the reversed change set.

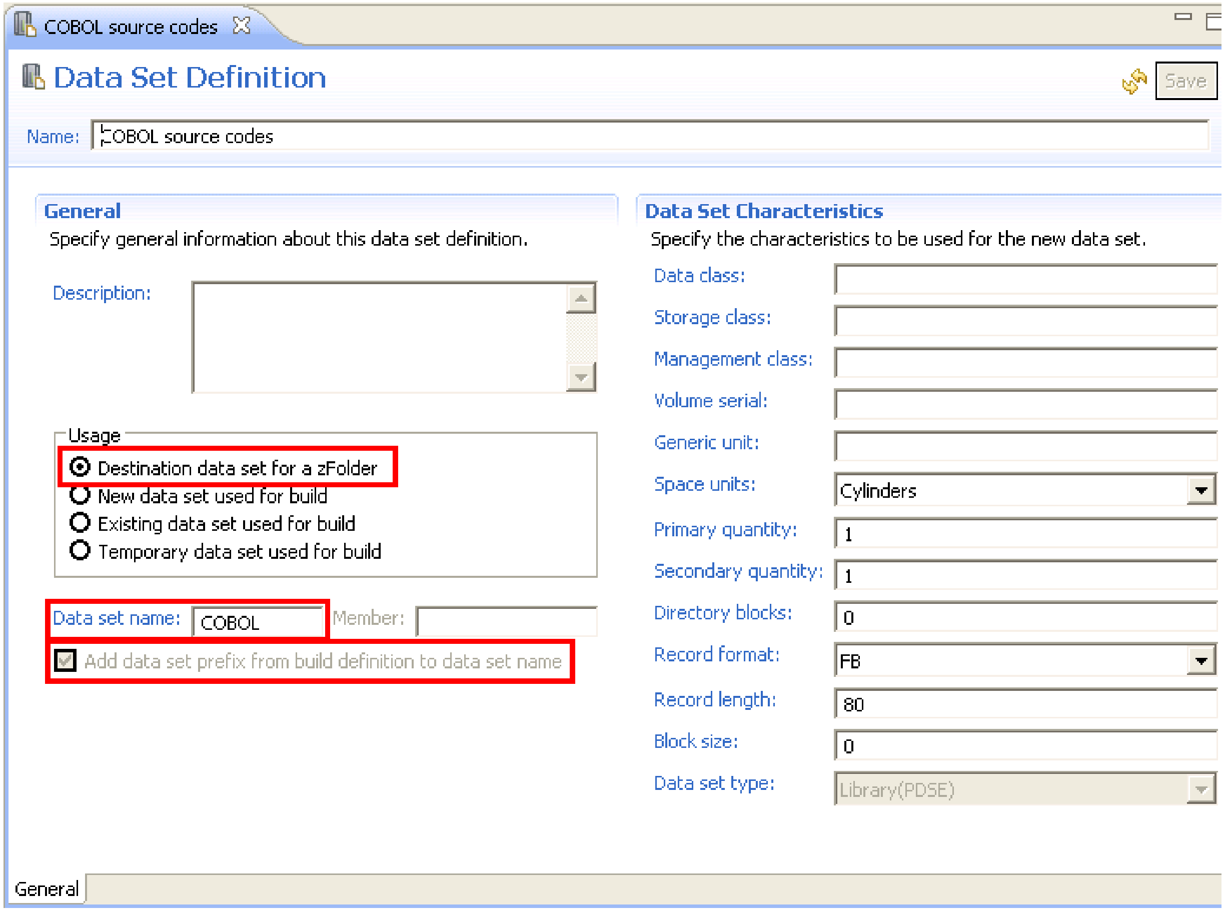
6.2. Working with Local Projects

In this section we are going to walk through the different artifacts related to resources for building them on z/OS. This will be based on an existing COBOL project that has already been created and is stored in the RTC SCM. As part of this lab you will explore the different artifacts, modify a COBOL part, and then build the sample application using Antz.

a) In this section we will work with an existing COBOL project loaded in RTC. We will use the functions of RDz to work with the parts. Make sure your repository workspace is loaded from the stream as done in the prior step, if it is not; load it now as you will be working with the files. First make sure you are in the work items perspective, if not Switch to the Work items perspective by going to **Window → Open Perspective → Work Items**



b) In this view you will explore the new Data Set Definitions and Language Definitions sections. First start by expanding the **Enterprise Extensions → System Definitions → z/OS → Data Set Definitions**, and double click to open the **COBOL source codes** definition. Data Set Definitions are used to define the datasets used on z/OS. These will include definitions for datasets used for source code that is placed in PDS/Es as well as the definition for the location of compilers, temporary datasets for build, and output datasets for build. Looking at this definition more closely you see this is a dataset used for source files copied to the PDS/Es. You will also notice this dataset will have a prefix added when used, this will be the users prefix when using RDz and the prefix defined in the build definition for team level builds. You will also notice this is where you define the dataset characteristics.



Data Set Definition

Name: COBOL source codes

General

Specify general information about this data set definition.

Description:

Usage

- ● Destination data set for a zFolder
- ○ New data set used for build
- ○ Existing data set used for build
- ○ Temporary data set used for build

Data set name: COBOL Member:

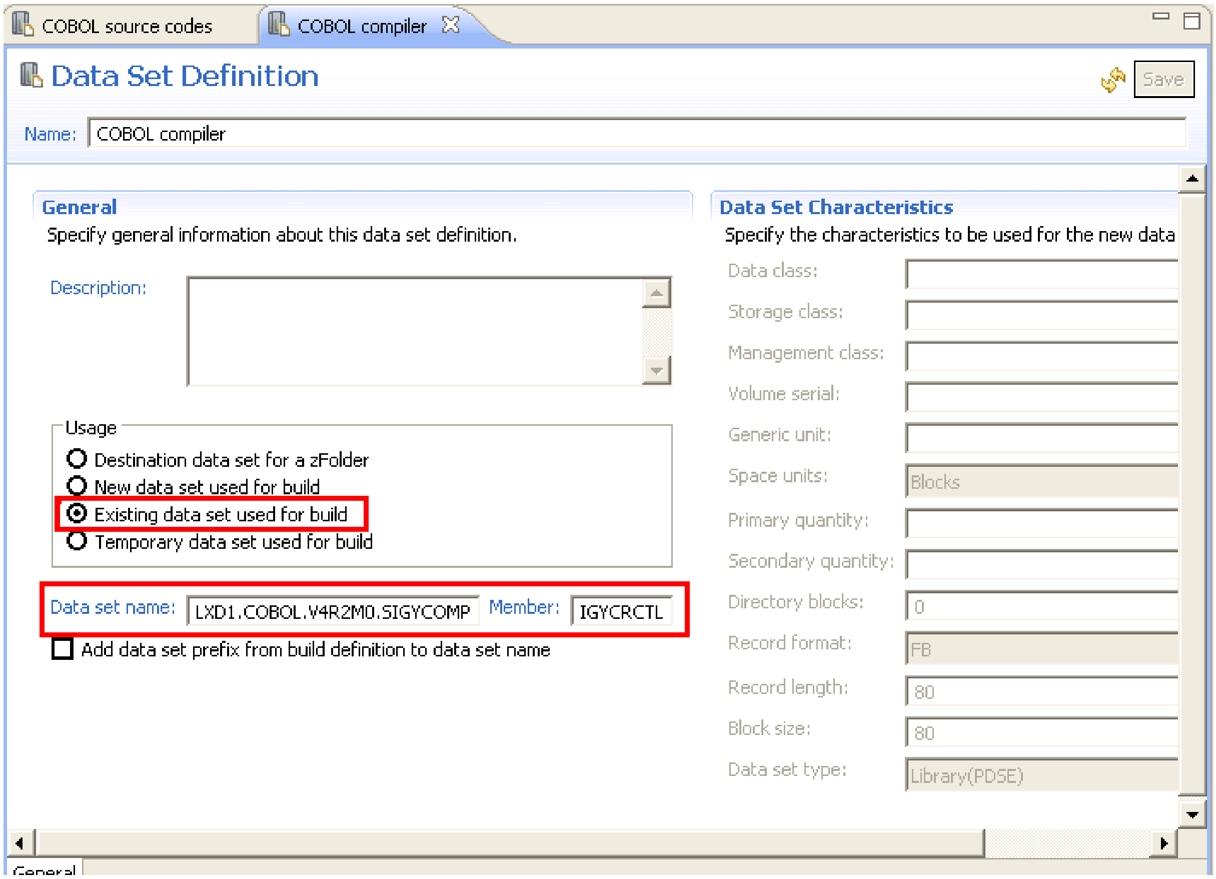
☑ Add data set prefix from build definition to data set name

Data Set Characteristics

Specify the characteristics to be used for the new data set.

Data class:	
Storage class:	
Management class:	
Volume serial:	
Generic unit:	
Space units:	Cylinders
Primary quantity:	1
Secondary quantity:	1
Directory blocks:	0
Record format:	FB
Record length:	80
Block size:	0
Data set type:	Library(PDSE)

c) Now open the **COBOL compiler** dataset definition. In this definition you will see the compiler to be used. You will see more about how these dataset definitions are used in the next few steps. You notice in this definition, an existing dataset is used for build and the specific member is specified.



Data Set Definition

Name: COBOL compiler

General

Specify general information about this data set definition.

Description:

Usage

○ Destination data set for a zFolder
○ New data set used for build
● Existing data set used for build
○ Temporary data set used for build

Data set name: LXD1.COBOL.V4R2M0.SIGYCOMP Member: IGYCRCTL

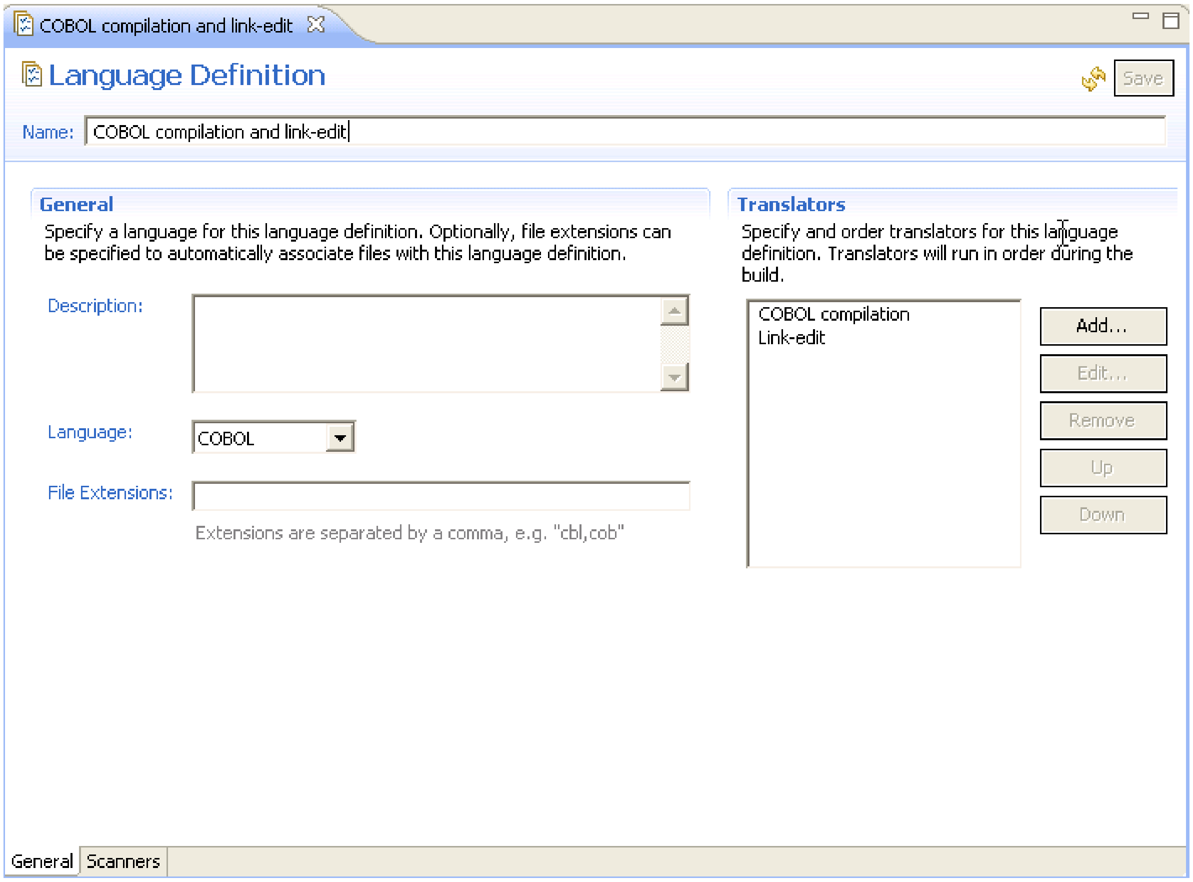
□ Add data set prefix from build definition to data set name

Data Set Characteristics

Specify the characteristics to be used for the new data

Data class:	
Storage class:	
Management class:	
Volume serial:	
Generic unit:	
Space units:	Blocks
Primary quantity:	
Secondary quantity:	
Directory blocks:	0
Record format:	FB
Record length:	80
Block size:	80
Data set type:	Library(PDSE)

d) Expand **Language Definitions** Section and you will see Language Definitions, as well as Translators. Open the **COBOL compilation and link-edit** language definition. In this definition you specify the name of the language definition, any optional descriptions, language related, any associated file extensions such as **.cbl** and the translators used. The name and translators are required. The Translators are similar to the job steps to be performed for this language definition. Multiple translators can be used in each language definition. Each file, or module, will be associated with a language definition. The Language Definitions are then used when using Antz to perform the build and for integration with RDz.



Language Definition

Name: COBOL compilation and link-edit

General

Specify a language for this language definition. Optionally, file extensions can be specified to automatically associate files with this language definition.

Description:

Language: COBOL

File Extensions:

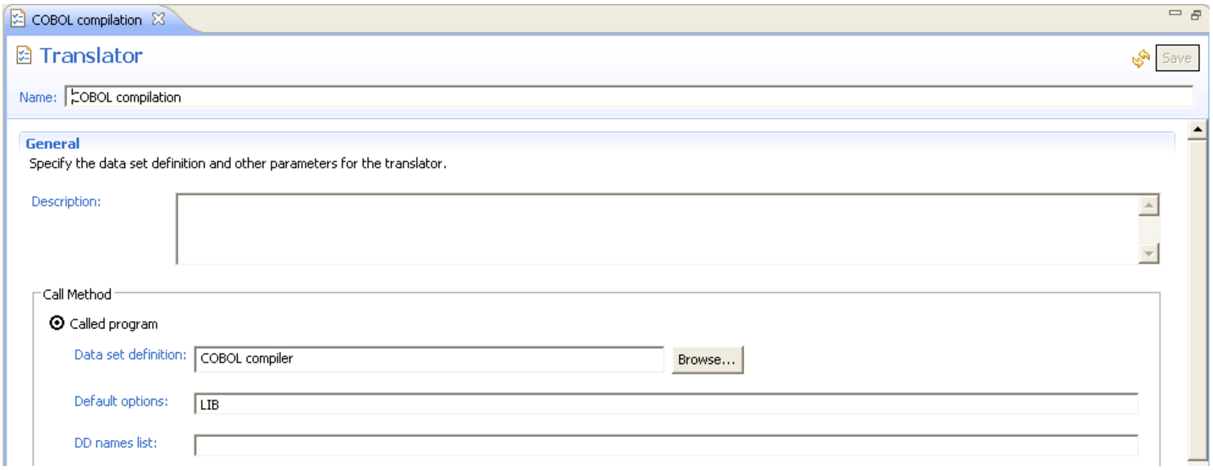
Extensions are separated by a comma, e.g. "cbl,cob"

Translators

Specify and order translators for this language definition. Translators will run in order during the build.

COBOL compilation
Link-edit

e) Open the associated **COBOL compilation** Translator by expanding the Translators node. In this translator you see the actual step to be preformed, similar to a job step. The translator specifies the dataset definition which includes the name of the program to be run. The translator also specifies the parameters to be supplied for the compile, and the associated datasets.



Translator

Name: COBOL compilation

General

Specify the data set definition and other parameters for the translator.

Description:

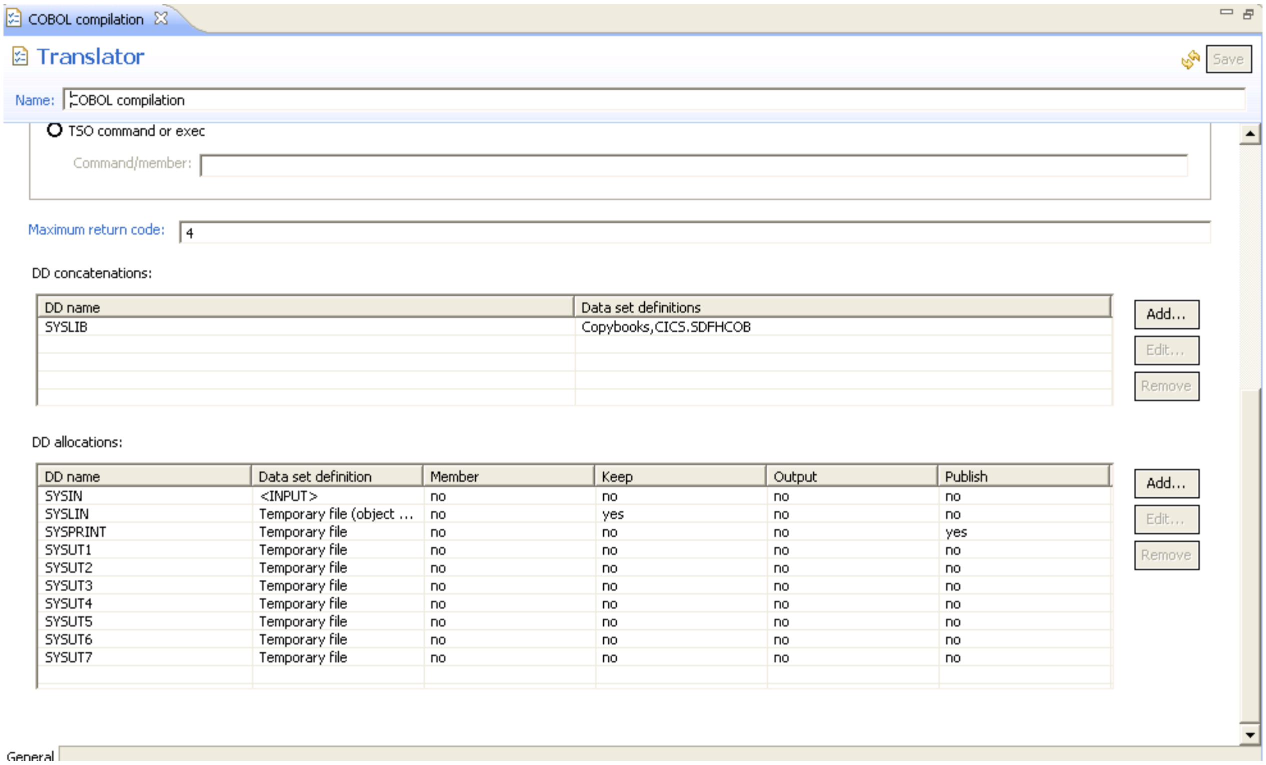
Call Method

● Called program

Data set definition: COBOL compiler

Default options: LIB

DD names list:



Translator

Name: COBOL compilation

○ TSO command or exec

Command/member:

Maximum return code: 4

DD concatenations:

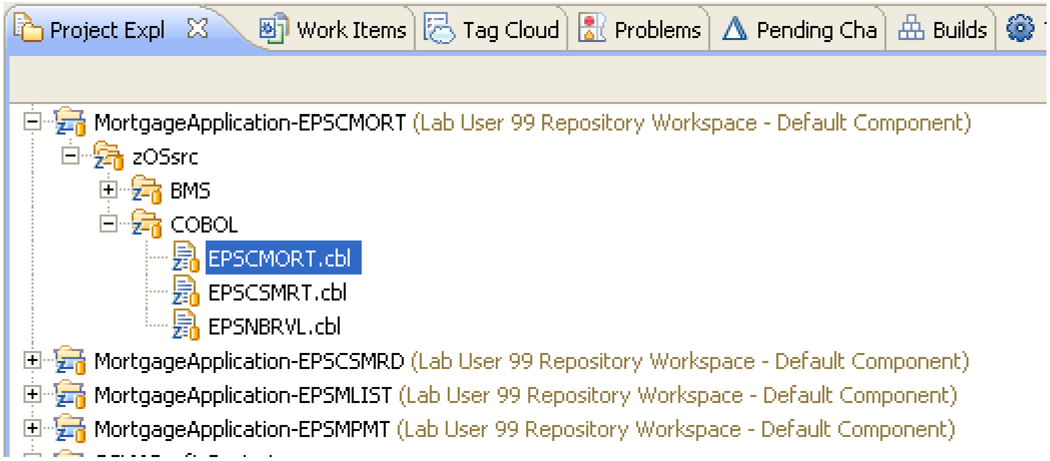
DD name	Data set definitions
SYSLIB	Copybooks,CICS.SDFHCOB

DD allocations:

DD name	Data set definition	Member	Keep	Output	Publish
SYSIN	<INPUT>	no	no	no	no
SYSLIN	Temporary file (object ...	no	yes	no	no
SYSPRINT	Temporary file	no	no	no	yes
SYSUT1	Temporary file	no	no	no	no
SYSUT2	Temporary file	no	no	no	no
SYSUT3	Temporary file	no	no	no	no
SYSUT4	Temporary file	no	no	no	no
SYSUT5	Temporary file	no	no	no	no
SYSUT6	Temporary file	no	no	no	no
SYSUT7	Temporary file	no	no	no	no

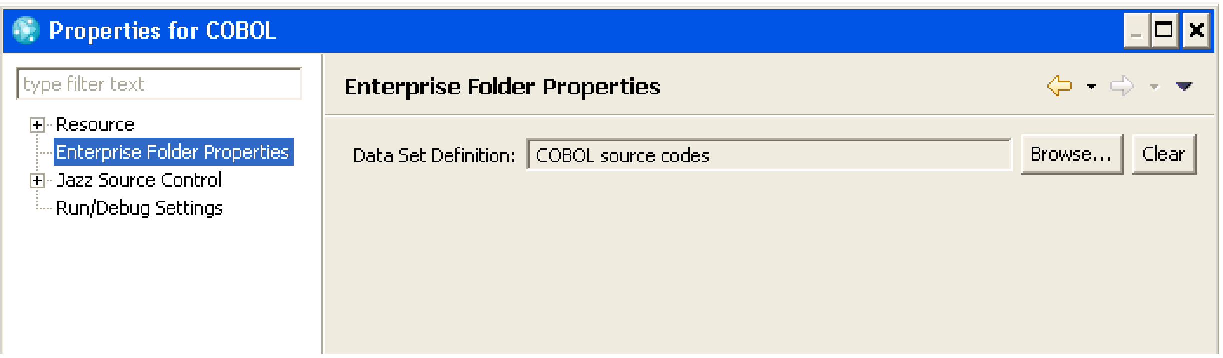
f) Now let's drill into the **zComponent** Projects. A zComponent project is a special type of Eclipse project that has a z/OS nature. This means that there will be special processing involved for this type of project, such as translation to EBCDIC when files are sent to z/OS for compiling.

Add the Project explorer view to your workspace by selecting **Window → Show View → Other…**, in the filter box start typing Project Explorer until you see the node appear, then select it and click OK. In the Project Explorer open the **MortgageApplication-EPSCMORT** project by expanding the node. Open the zOSsrc folder and then the COBOL folder and look at the files.



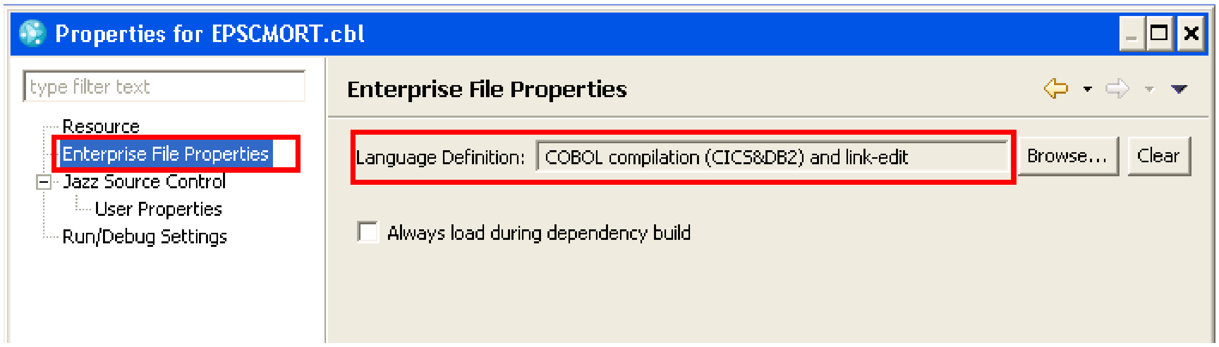
MortgageApplication-EPSCMORT (Lab User 99 Repository Workspace - Default Component)
zOSsrc
BMS
COBOL
EPSCMORT.cbl
EPSCSMRT.cbl
EPSNBRVL.cbl
MortgageApplication-EPSCSMRD (Lab User 99 Repository Workspace - Default Component)
MortgageApplication-EPSMLIST (Lab User 99 Repository Workspace - Default Component)
MortgageApplication-EPSMPMT (Lab User 99 Repository Workspace - Default Component)

Open the properties for the COBOL folder by right clicking on the COBOL folder and selecting properties. If you click on the ***Enterprise Folder Properties*** you will notice the specification of the associated dataset definition



Data Set Definition: COBOL source codes

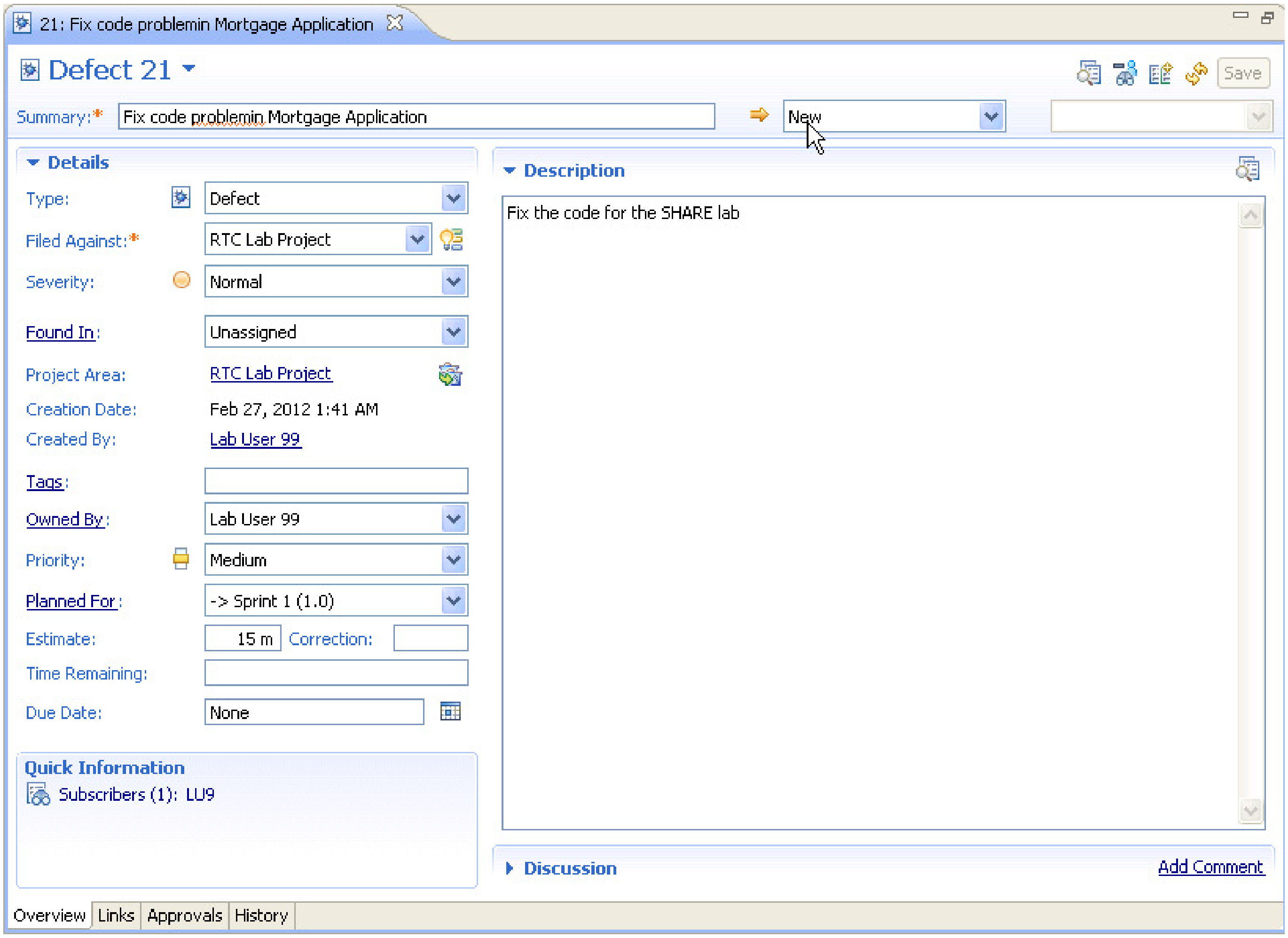
Similarly right click on the EPSCMORT file and select properties, again if you click on the ***Enterprise File Properties*** you will see the language definition associated with the file. You can change the language definition used at this point if required.



Language Definition: COBOL compilation (CICS&DB2) and link-edit

☐ Always load during dependency build

g) Now that we have an understanding of the definition for the zComponent project, let's make a source code change. First create a new work item to track your changes, and start working on it. (See the working with work items section for detailed instructions.)



21: Fix code problemin Mortgage Application

Defect 21

Summary:* Fix code problemin Mortgage Application

New

Details

Type: Defect

Filed Against:* RTC Lab Project

Severity: Normal

Found In: Unassigned

Project Area: RTC Lab Project

Creation Date: Feb 27, 2012 1:41 AM

Created By: Lab User 99

Tags:

Owned By: Lab User 99

Priority: Medium

Planned For: -> Sprint 1 (1.0)

Estimate: 15 m Correction:

Time Remaining:

Due Date: None

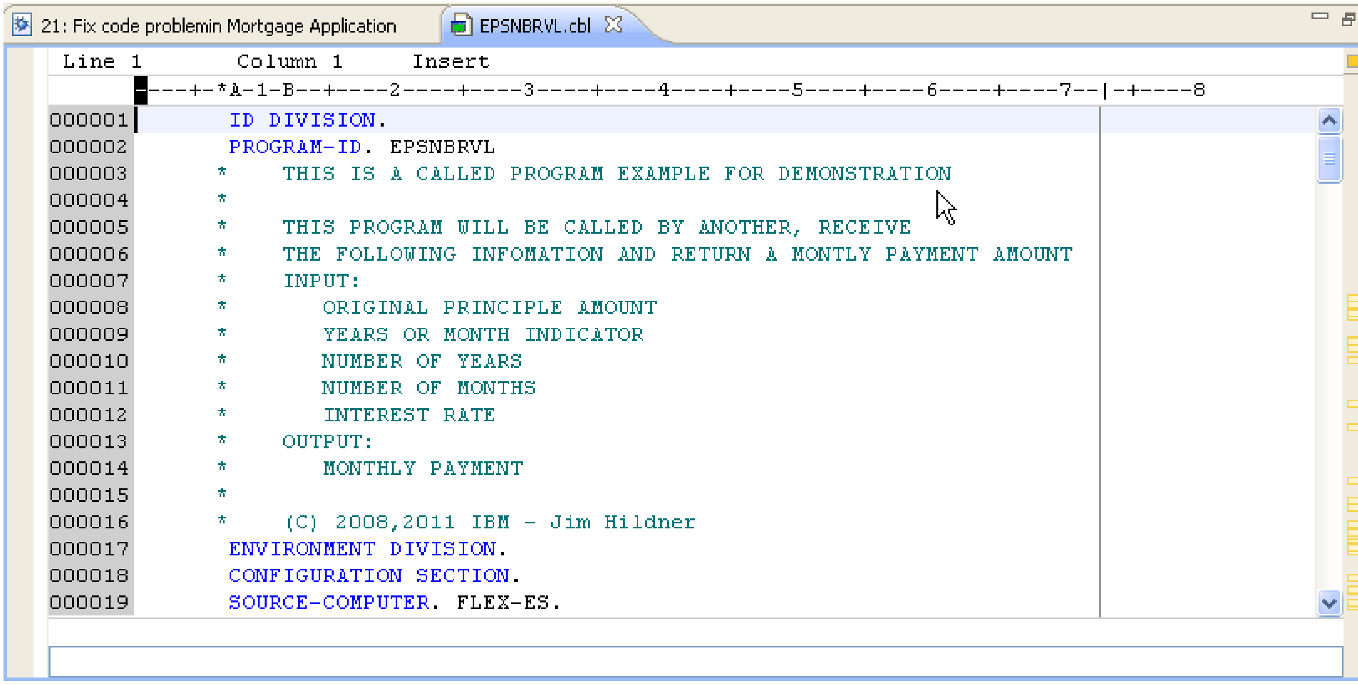
Quick Information

Subscribers (1): LU9

Description

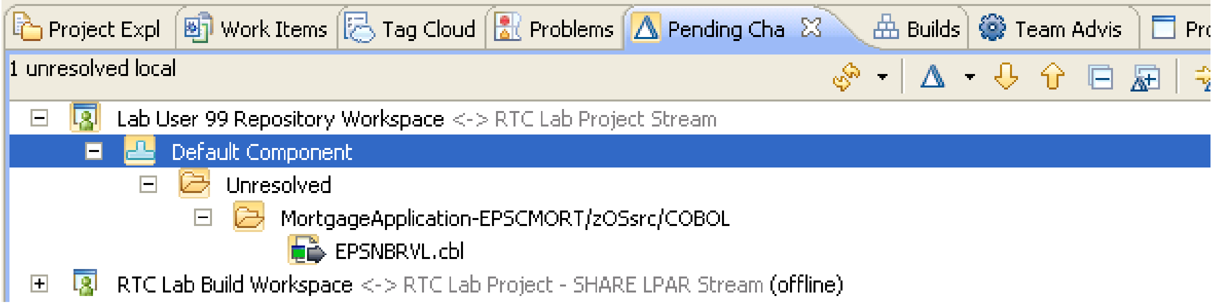
Fix the code for the SHARE lab

h) Once you are working on a work item. Double click on the EPSNBRVL COBOL file with the .cbl extension. You will see it has been opened in the RDz editor. Make some change in the comments, and save. To save a file, either right click inside the file and select Save, Just close the file and confirm you wish to save it when asked, select **file → Save** from the main menu bar, or click the icon of the floppy disk on the toolbar.



```
 Line 1        Column 1       Insert
      ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--|-+----8
000001        ID DIVISION.
000002        PROGRAM-ID. EPSNBRVL
000003       *    THIS IS A CALLED PROGRAM EXAMPLE FOR DEMONSTRATION
000004       *
000005       *    THIS PROGRAM WILL BE CALLED BY ANOTHER, RECEIVE
000006       *    THE FOLLOWING INFOMATION AND RETURN A MONTLY PAYMENT AMOUNT
000007       *    INPUT:
000008       *       ORIGINAL PRINCIPLE AMOUNT
000009       *       YEARS OR MONTH INDICATOR
000010       *       NUMBER OF YEARS
000011       *       NUMBER OF MONTHS
000012       *       INTEREST RATE
000013       *    OUTPUT:
000014       *       MONTHLY PAYMENT
000015       *
000016       *    (C) 2008,2011 IBM - Jim Hildner
000017        ENVIRONMENT DIVISION.
000018        CONFIGURATION SECTION.
000019        SOURCE-COMPUTER. FLEX-ES.
```

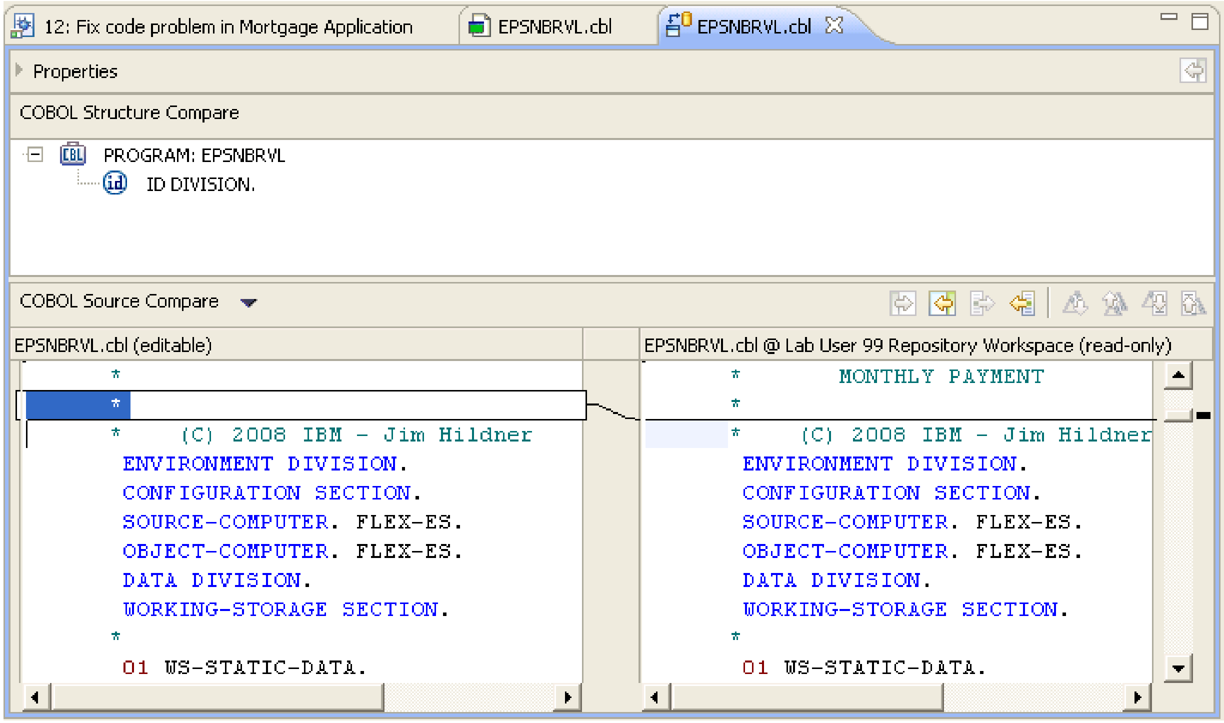
i) Look at the pending changes view, it should be displayed below the source code. If not, open the view with **Window → Show View → Pending Changes.** You will now see that the Mortgage Application has a set of unresolved changes.



1 unresolved local

- Lab User 99 Repository Workspace <-> RTC Lab Project Stream
  - Default Component
    - Unresolved
      - MortgageApplication-EPSCMORT/zOSsrc/COBOL
        - EPSNBRVL.cbl
- RTC Lab Build Workspace <-> RTC Lab Project - SHARE LPAR Stream (offline)

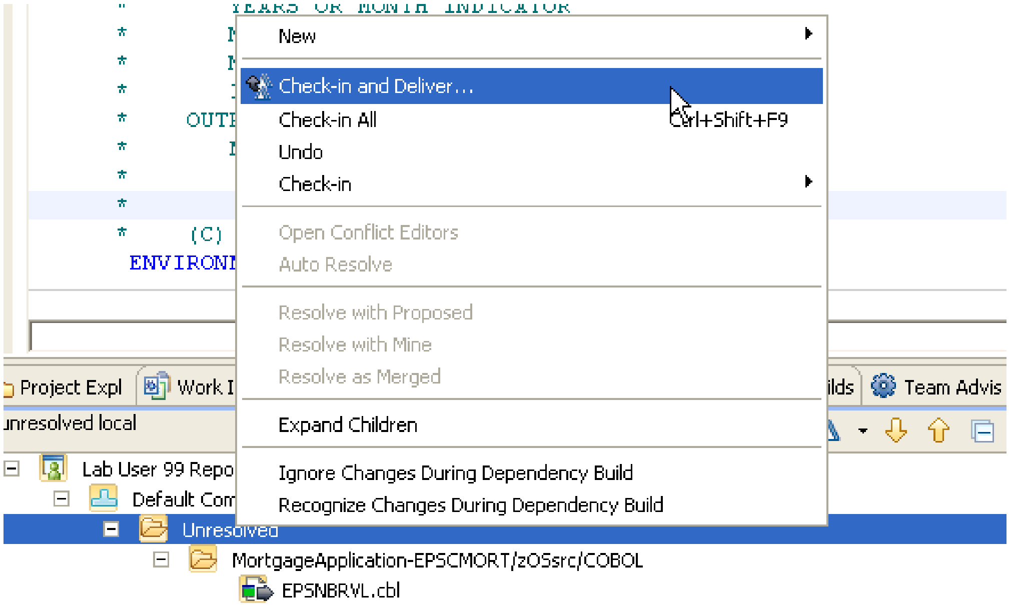
If you expand to see the file that changed, and double click on the file, the Text compare window will be displayed so that you can see what was added.



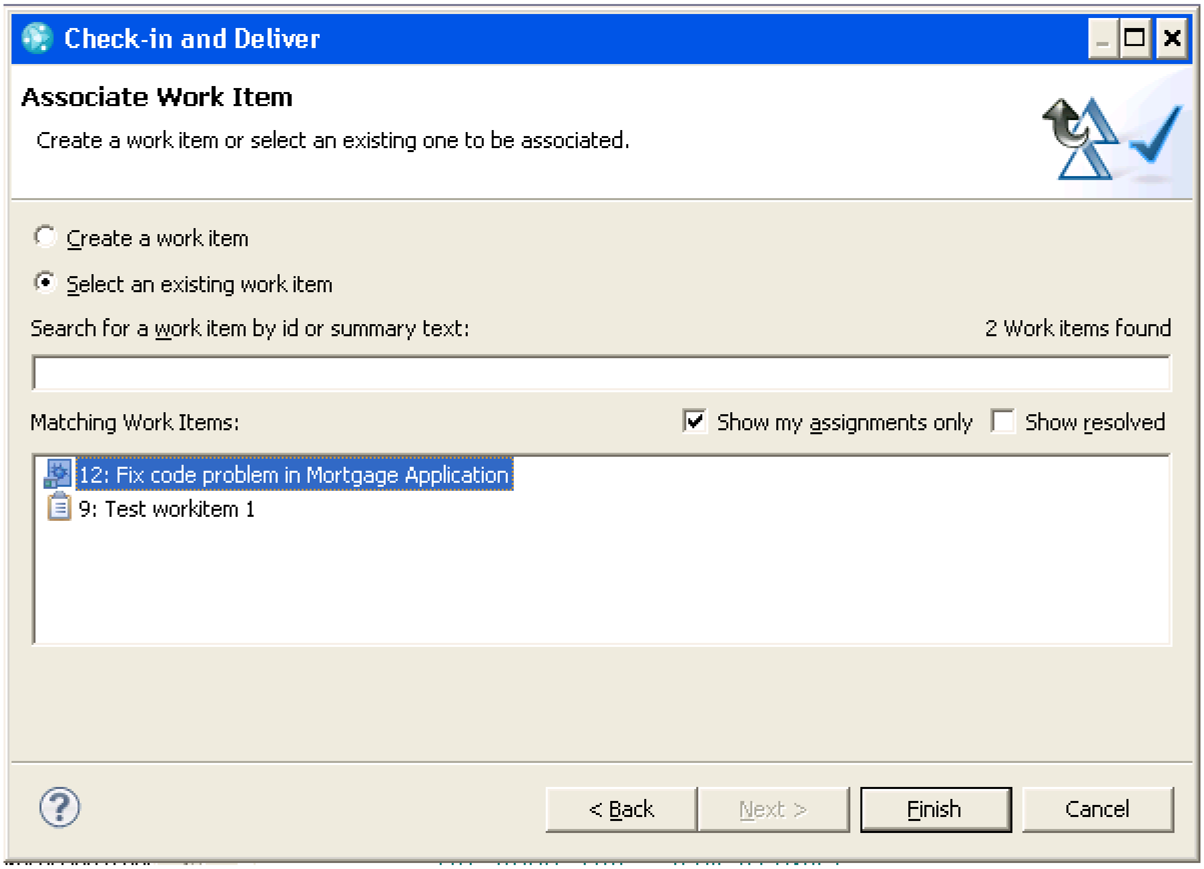
```
      *
      *
      *     (C) 2008 IBM - Jim Hildner
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. FLEX-ES.
       OBJECT-COMPUTER. FLEX-ES.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
      *
       01 WS-STATIC-DATA.
```

```
      *         MONTHLY PAYMENT
      *
      *     (C) 2008 IBM - Jim Hildner
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. FLEX-ES.
       OBJECT-COMPUTER. FLEX-ES.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
      *
       01 WS-STATIC-DATA.
```

j) Go back to the Pending Changes View and right click on the **Unresolved** line, you will get the option to **Check-in** or **Check-in and Deliver…** This allows you to save the changes to your repository workspace, as well as to deliver them to the stream. Go ahead and select **Check-in and Deliver.**



k) Type a comment to go with the change set, and click **Next**. Select the work item you created that you are currently working on to associate to the change set. You may need to add the work item number into the search box to find it. And click **Finish**. Your code will now be delivered to the stream and the pending changes view will show no changes.



Check-in and Deliver

Associate Work Item

Create a work item or select an existing one to be associated.

○ Create a work item

● Select an existing work item

Search for a work item by id or summary text: 2 Work items found

Matching Work Items: ☑ Show my assignments only □ Show resolved

- 12: Fix code problem in Mortgage Application
- 9: Test workitem 1

You could have just checked in the file, then worked on a bunch of other files and checked them into the same changeset. Then associate a workitem at one time to the complete changeset prior to delivering.

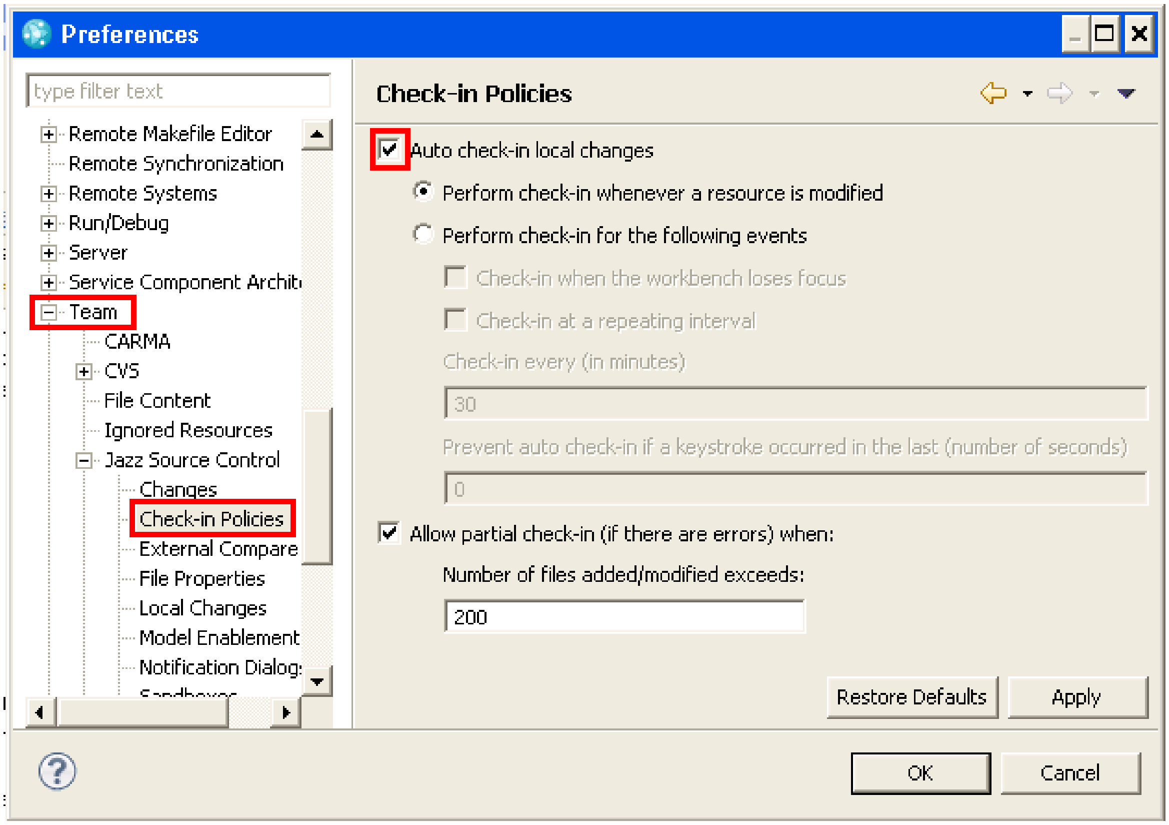
6.3. Optional: Enabling Automatic Check-Ins

a. Once a change is made to a versioned file (a file already being tracked by the Jazz Source Code repository), its corresponding change-set must be checked in to commit the change to your repository workspace. By default, check-ins must be initiated by the user. However, Rational Team Concert provides the option for making check-ins occur automatically.

b. Select **Window -> Preferences** to open the Preferences settings.

c. Expand **Team -> Jazz Source Control** and select **Check-in Policies**

d. Check **Auto check-in local changes** and press OK.



Check-in Policies

☑ Auto check-in local changes

● Perform check-in whenever a resource is modified

○ Perform check-in for the following events

□ Check-in when the workbench loses focus

□ Check-in at a repeating interval

Check-in every (in minutes): 30

Prevent auto check-in if a keystroke occurred in the last (number of seconds): 0

☑ Allow partial check-in (if there are errors) when:

Number of files added/modified exceeds: 200

Conclusion

This concludes Module 6. You now should be familiar with source code management using the Eclipse GUI, along with the integration with the work items.

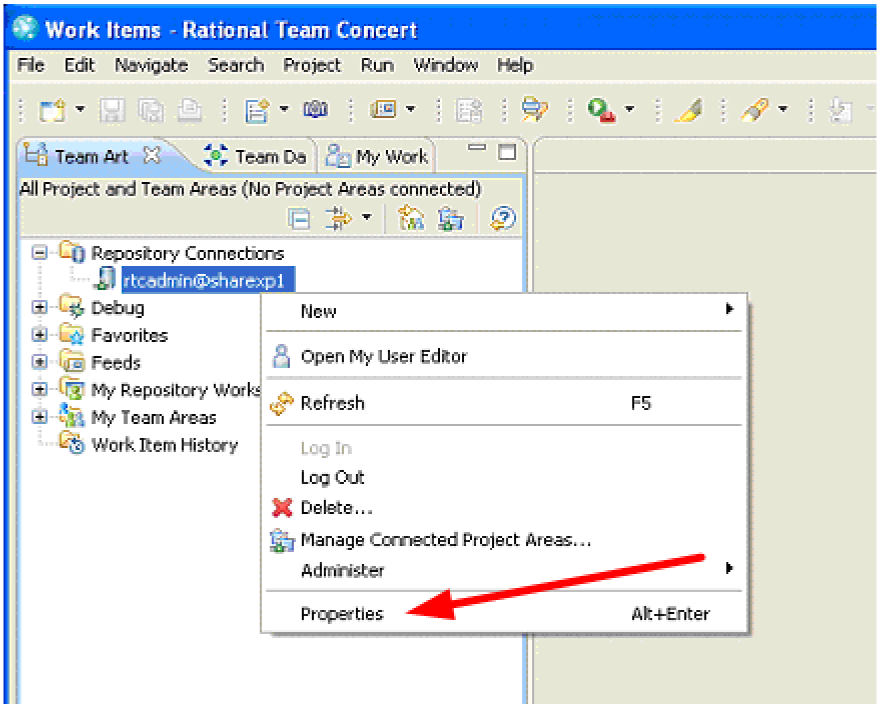
Module 7 Enterprise Build (SCM)

7.1. z/OS Antz Build

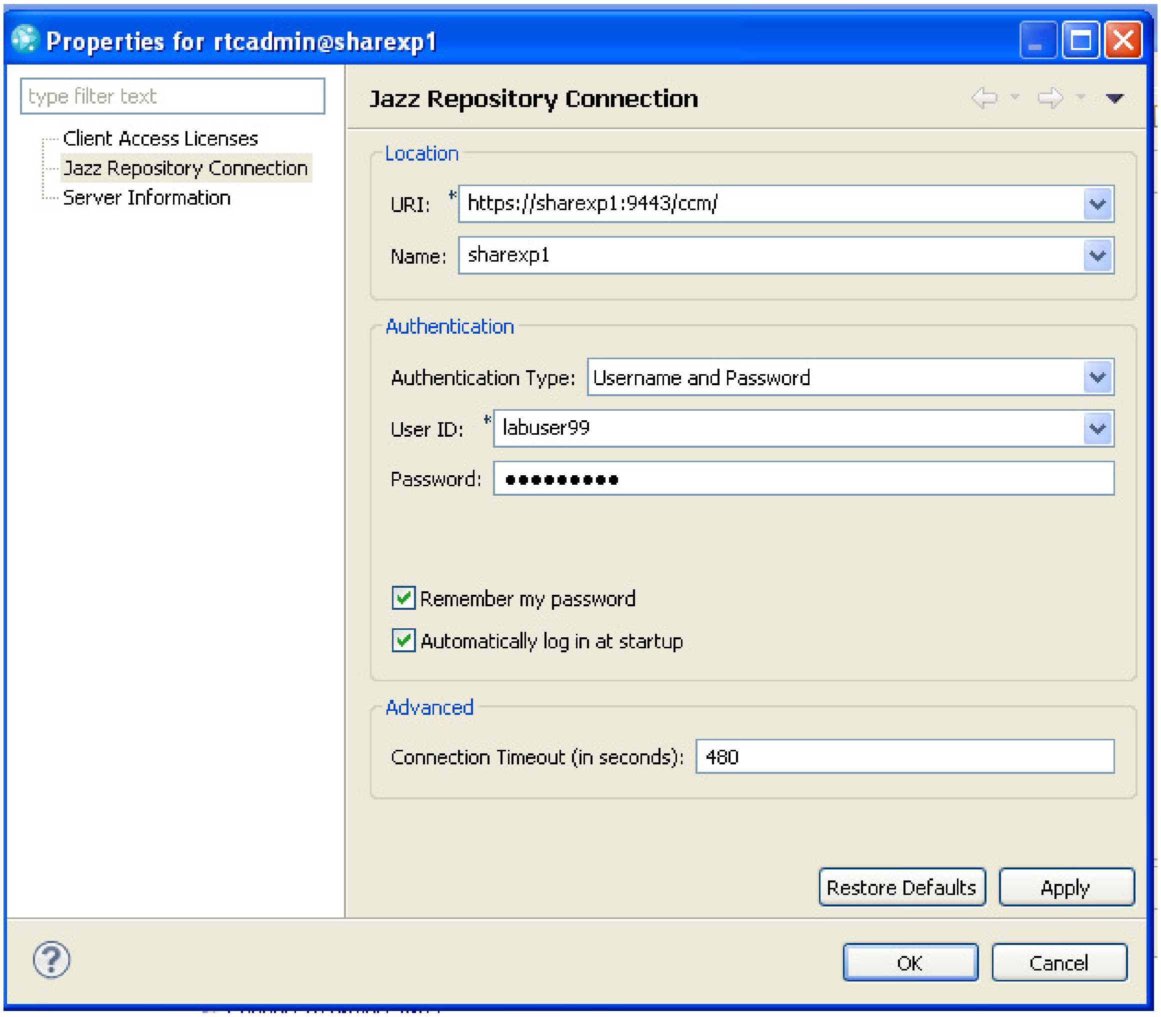
In this section we are going to create a build definition that will build the mortgage application. This build definition will use the RTC Ant with Enterprise Extensions capabilities to perform the build.

RTC uses a build agent that is running on the z/OS machine. This agent handles the transport of the parts to and from the RTC repository, plus the invocation of the build through the use of Ant XML build scripts. These are generally automatically generated for you.

a) Ensure you are logged into RTC. If you created a user in Module 1 then you can use that user. Alternatively you can use **labuser99** which has been created for you already. To login right click on the ***labuser99@sharexp1*** node in the Repository connections and select **login**. Or if using your own userid right click on labuser99 and select **Properties** as shown below



b) In the dialog box select the Jazz Repository Connection on the left side of the screen. Then change the user id and password to the one you created previously (labuser#). The password is the same as the user id. Press **OK**.



Properties for rtcadmin@sharexp1

Jazz Repository Connection

Location

URI: * https://sharexp1:9443/ccm/

Name: sharexp1

Authentication

Authentication Type: Username and Password

User ID: * labuser99

Password: ●●●●●●●●●●

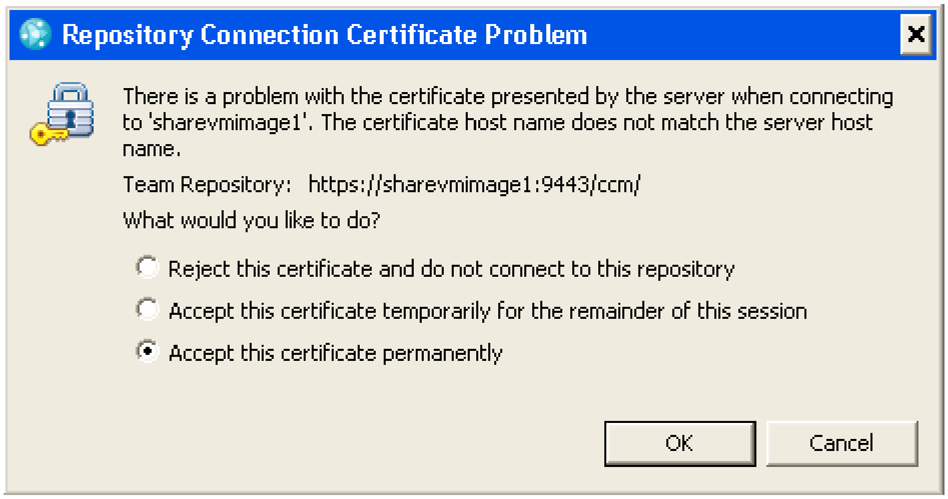
☑ Remember my password

☑ Automatically log in at startup

Advanced

Connection Timeout (in seconds): 480

c) If you get any certificate problems when connecting, just accept the certificate permanently:



Repository Connection Certificate Problem

There is a problem with the certificate presented by the server when connecting to 'sharevmimage1'. The certificate host name does not match the server host name.

Team Repository: https://sharevmimage1:9443/ccm/

What would you like to do?

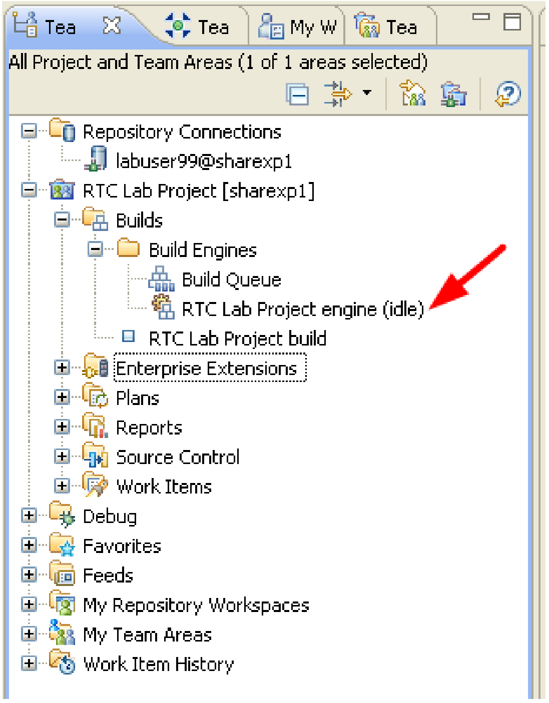
○ Reject this certificate and do not connect to this repository

○ Accept this certificate temporarily for the remainder of this session

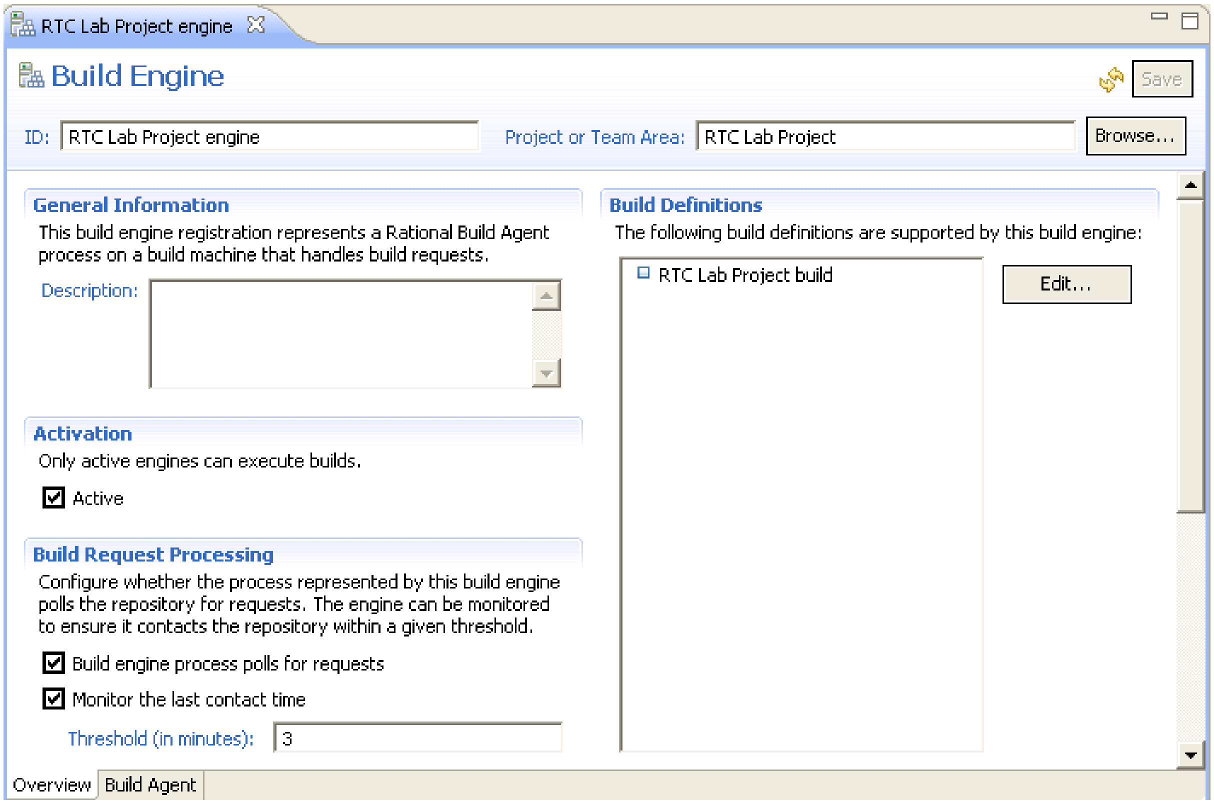
● Accept this certificate permanently

d) Now that we have the artifacts defined let’s look at doing a build. Navigate back to the Team artifacts view and open the builds section. Under the builds, you should see a folder with the existing build engines, and a build definition for the Mortgage Application called **RTC Lab Project build**. We will step through this definition to make sure it is set up.

Note : On the lab system there may be other build engines and build definitions, such as the RTC Lab - MVS269 Engine and the RTC Lab Project MVS269 build. These are used for lab system set-up and can be ignored. Although if you are short of time, you can look at the RTC Lab Project MVS269 build to see what a successful build looks like.



e) Open the RTC Lab Project engine by right clicking the build engine and selecting **Open Build engine**. The first tab is an overview and mainly shows us the build definitions that use this build engine, this screen should look like this:



RTC Lab Project engine

Build Engine

Save

ID: RTC Lab Project engine

Project or Team Area: RTC Lab Project

Browse...

General Information

This build engine registration represents a Rational Build Agent process on a build machine that handles build requests.

Description:

Activation

Only active engines can execute builds.

☑ Active

Build Request Processing

Configure whether the process represented by this build engine polls the repository for requests. The engine can be monitored to ensure it contacts the repository within a given threshold.

☑ Build engine process polls for requests

☑ Monitor the last contact time

Threshold (in minutes): 3

Build Definitions

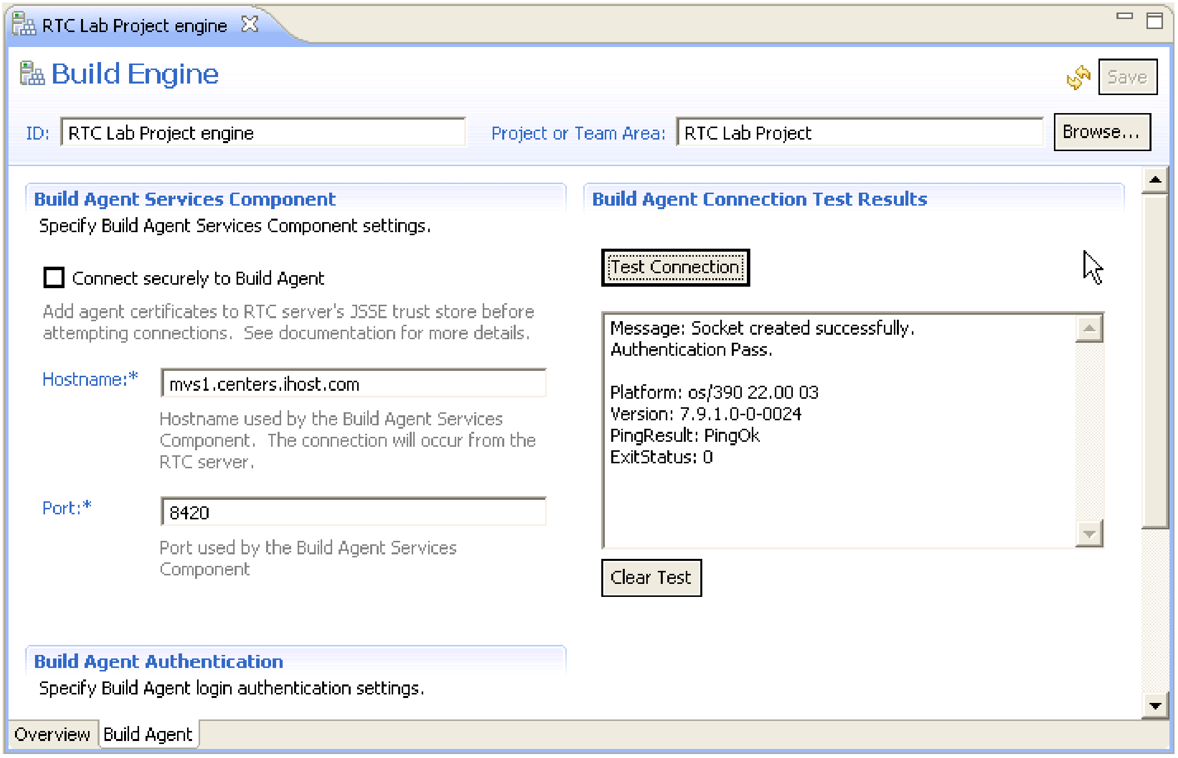
The following build definitions are supported by this build engine:

- RTC Lab Project build

Edit...

Overview | Build Agent

The Build Agent tab tells us where the build engine is running and provides connection information. For this lab the build agent is running on the SHARE machine, make sure the build agent hostname is specified as **mvs1.centers.ihost.com** and the port number is **8420**. The user name is set to LXD1with a password, this has been set up as the build user id. To verify you have a connection to the SHARE machines build agent you can click the Test Connection button. You should get a connection with the following feedback:



ID: RTC Lab Project engine

Project or Team Area: RTC Lab Project

Build Agent Services Component

Specify Build Agent Services Component settings.

☐ Connect securely to Build Agent

Add agent certificates to RTC server's JSSE trust store before attempting connections. See documentation for more details.

Hostname:* mvs1.centers.ihost.com

Hostname used by the Build Agent Services Component. The connection will occur from the RTC server.

Port:* 8420

Port used by the Build Agent Services Component

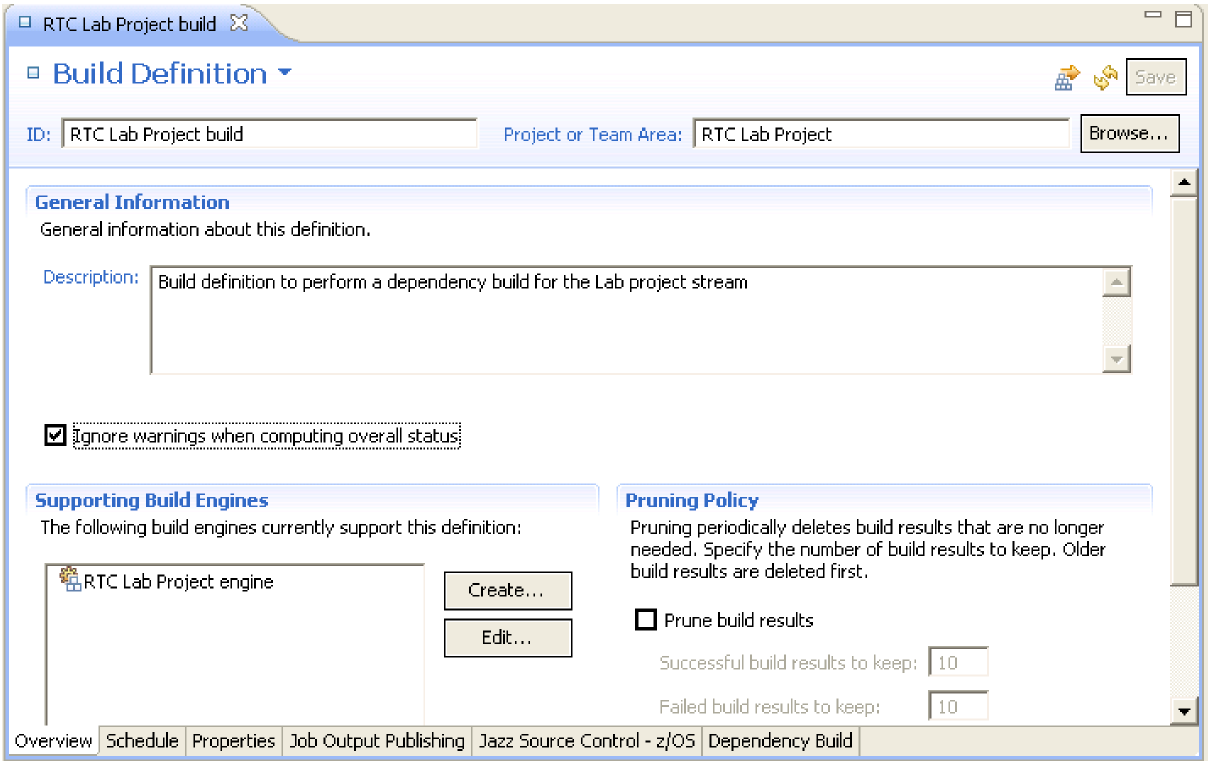
Build Agent Connection Test Results

```
Message: Socket created successfully.
Authentication Pass.

Platform: os/390 22.00 03
Version: 7.9.1.0-0-0024
PingResult: PingOk
ExitStatus: 0
```

Build Agent Authentication

Specify Build Agent login authentication settings.

f) Open the RTC Lab Project Build Definition by right clicking on the build definition and selecting **Open Build Definition**. The first tab is the overview tab, it provides the build name, the project or team area the build is associated with, any description for the build, supporting build definitions and pruning policy. The Build Definition should look like this:



ID: RTC Lab Project build

Project or Team Area: RTC Lab Project

General Information

General information about this definition.

Description: Build definition to perform a dependency build for the Lab project stream

☑ Ignore warnings when computing overall status

Supporting Build Engines

The following build engines currently support this definition:

RTC Lab Project engine

Pruning Policy

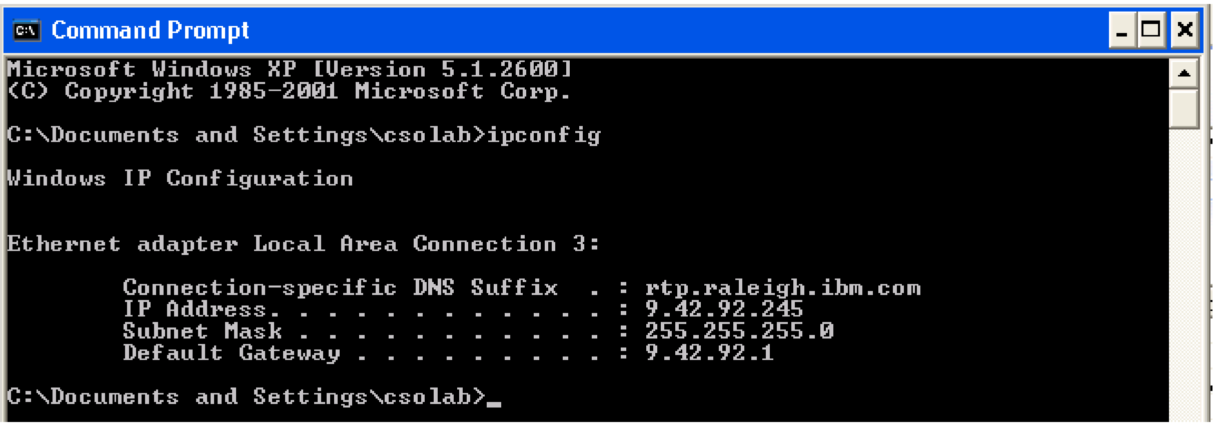
Pruning periodically deletes build results that are no longer needed. Specify the number of build results to keep. Older build results are deleted first.

☐ Prune build results

Successful build results to keep: 10

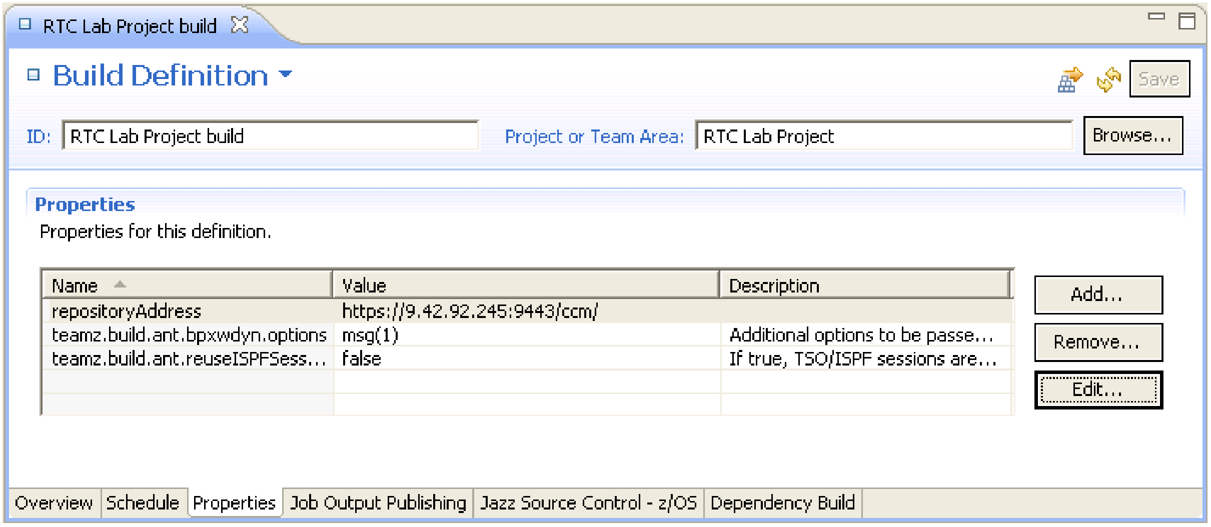
Failed build results to keep: 10

g) Notice there are a number of tabs associated with the build definition. The second tab is the schedule tab, if this build would be a scheduled build, you could select when to have the build run automatically. The third tab is the properties tab, which allows you to specify additional properties for the build. You will need to set one of the properties in this tab, to specify the IP address of your server, such that the build engine can contact you back. To find your IP address, open a command prompt and type **ipconfig** and press enter. You should see this:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\csolab>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 3:

        Connection-specific DNS Suffix  . : rtp.raleigh.ibm.com
        IP Address. . . . . . . . . . . . : 9.42.92.245
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 9.42.92.1

C:\Documents and Settings\csolab>_
```

Remember your IP address, in this case 9.42.92.245, then open the properties tab of the build definition. You can edit the existing build property by highlighting the repositoryAddress line and clicking enter. Change https://sharexp1:9443/ccm/ to be https://x.x.x.x:9443/ccm/ where x.x.x.x is your IP address. The properties tab should look like this:



ID: RTC Lab Project build

Project or Team Area: RTC Lab Project

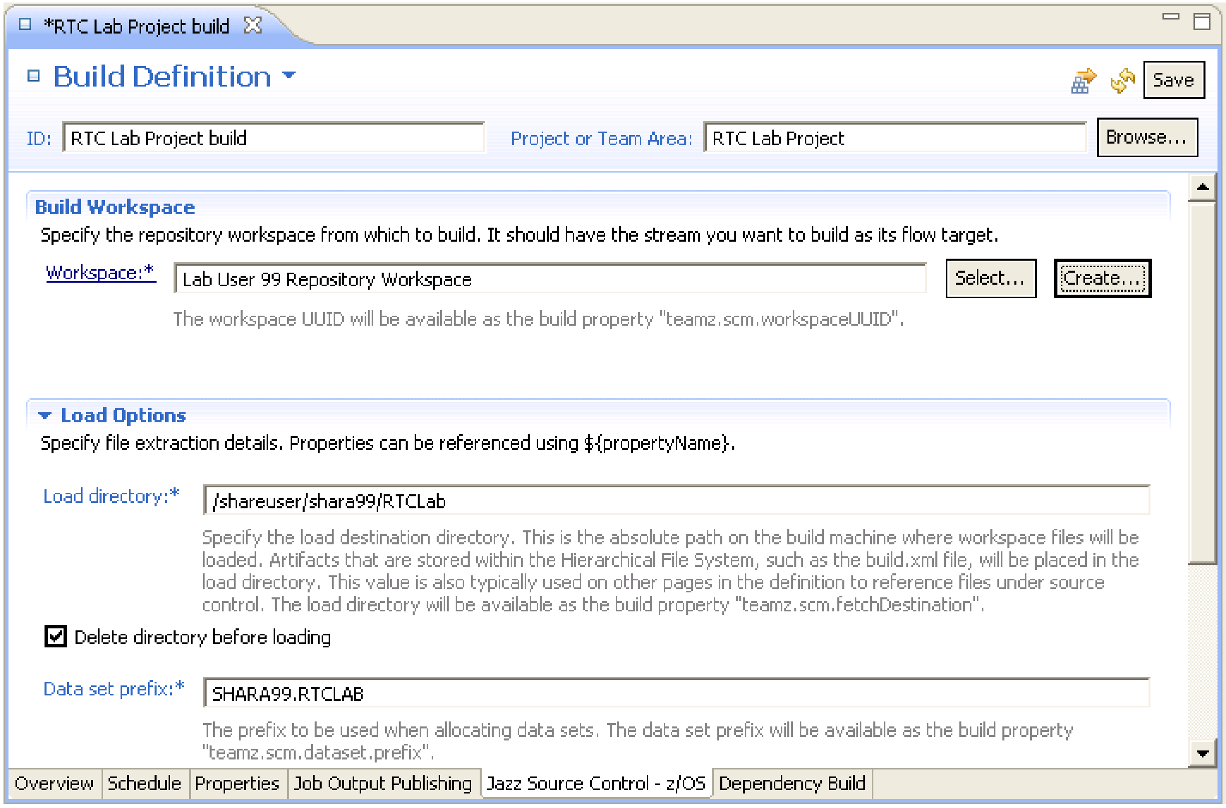
Properties

Properties for this definition.

Name	Value	Description
repositoryAddress	https://9.42.92.245:9443/ccm/	
teamz.build.ant.bpxwdyn.options	msg(1)	Additional options to be passe...
teamz.build.ant.reuseISPFSess...	false	If true, TSO/ISPF sessions are...

h) Select the **Job Output Publishing** page, here you specify to publish job output logs. The check box should be selected.

i) The next page is the **Jazz Source Control – z/OS**. This is the page that specifies what repository workspace should be used, and the specific load options. Make sure your repository workspace is selected, and specify the load directory as **/shareuser/sharaxx/RTCLab** and the dataset prefix as **SHARA01.RTCLAB**. Change **sharaxx** to the userid that you were told to log in with. Also make sure the **delete directory before loading** option is specified.



ID: RTC Lab Project build

Project or Team Area: RTC Lab Project

Build Workspace

Specify the repository workspace from which to build. It should have the stream you want to build as its flow target.

Workspace:* Lab User 99 Repository Workspace

The workspace UUID will be available as the build property "teamz.scm.workspaceUUID".

Load Options

Specify file extraction details. Properties can be referenced using ${propertyName}.

Load directory:* /shareuser/shara99/RTCLab

Specify the load destination directory. This is the absolute path on the build machine where workspace files will be loaded. Artifacts that are stored within the Hierarchical File System, such as the build.xml file, will be placed in the load directory. This value is also typically used on other pages in the definition to reference files under source control. The load directory will be available as the build property "teamz.scm.fetchDestination".

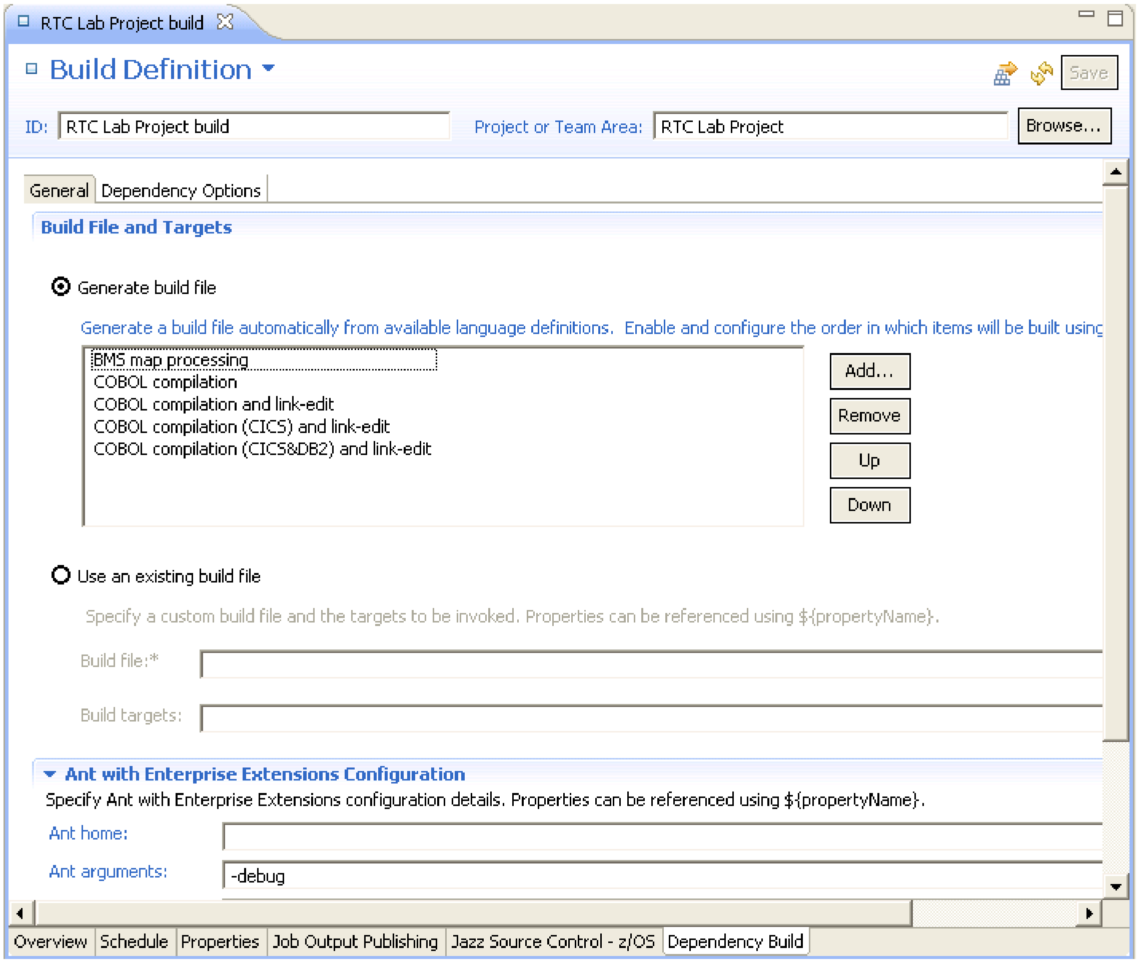
☑ Delete directory before loading

Data set prefix:* SHARA99.RTCLAB

The prefix to be used when allocating data sets. The data set prefix will be available as the build property "teamz.scm.dataset.prefix".

Note : If you have logged in as LABUSER99, you will still need to use your assigned mainframe ID or SHARAxx for the load directory and Data set prefix. This is because everyone cannot use and overwrite the Labuser99 data sets.

j) The last tab is the Dependency Build tab. This build definition has been set up as a dependency based build to only build what has changed, so in essence a smart build. There are some sub-tabs in the main tab. The first is a general tab where you can specify things here such as a modified build.xml file. However we are letting RTC generate a build file for us based on the language definitions used in our project This tab is also used to specify any other Antz configuration parameters.



ID: RTC Lab Project build

Project or Team Area: RTC Lab Project

Build File and Targets

● Generate build file

Generate a build file automatically from available language definitions. Enable and configure the order in which items will be built using

- BMS map processing
- COBOL compilation
- COBOL compilation and link-edit
- COBOL compilation (CICS) and link-edit
- COBOL compilation (CICS&DB2) and link-edit

○ Use an existing build file

Specify a custom build file and the targets to be invoked. Properties can be referenced using ${propertyName}.

Build file:*

Build targets:

Ant with Enterprise Extensions Configuration

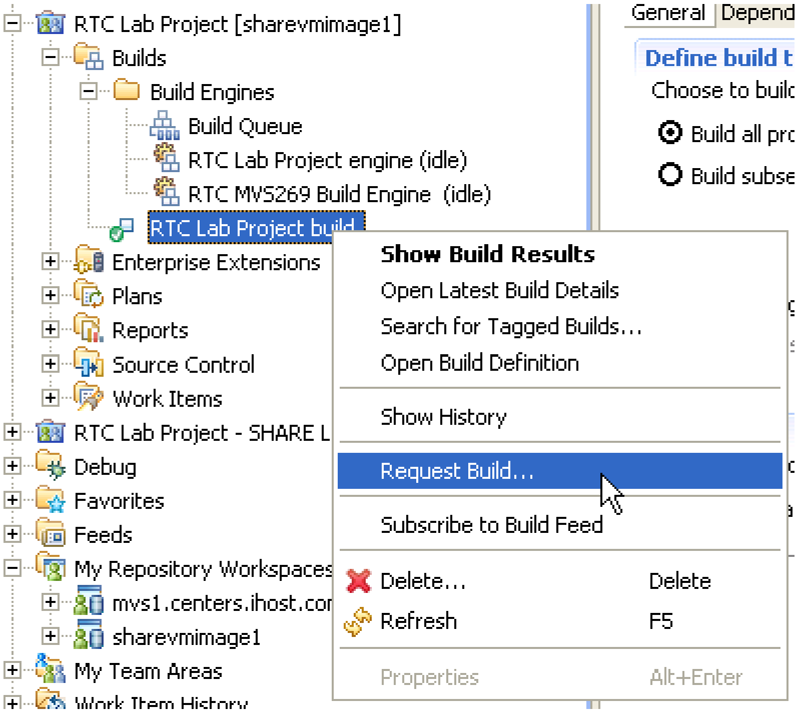
Specify Ant with Enterprise Extensions configuration details. Properties can be referenced using ${propertyName}.

Ant home:

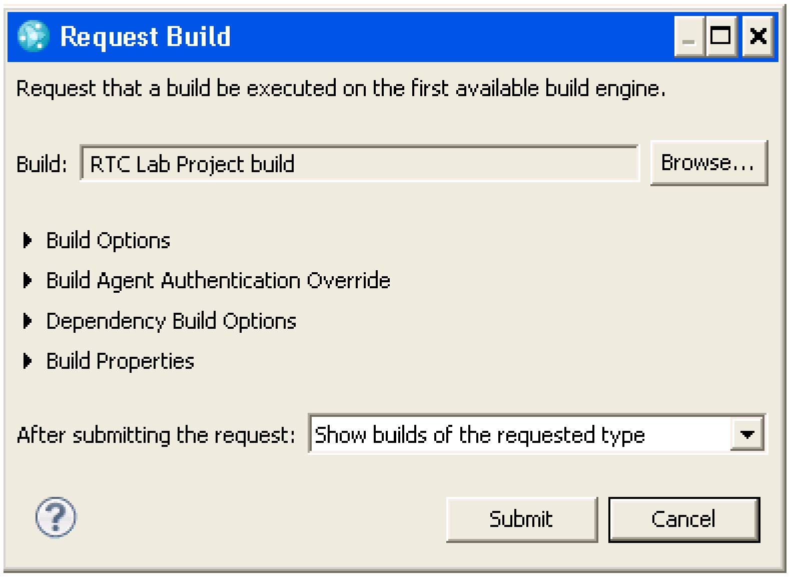
Ant arguments: -debug

The Dependency options allow you to define some options such as to build only changed items or to only build certain programs. We don’t need to change anything here.

k) Once the build definition is updated, save the build definition using the **Save** button at the top right of the Build Definition. Once the Build Definition is complete it can be submitted. To submit the job, right click on the definition and click **Request Build….**



l) The Request Build definition is displayed click on the submit button to request the build.

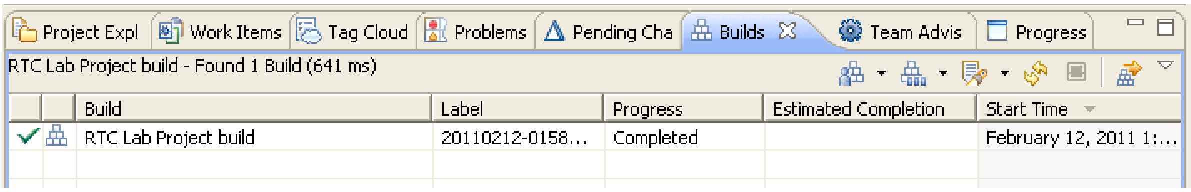


Request that a build be executed on the first available build engine.

Build: RTC Lab Project build

After submitting the request: Show builds of the requested type

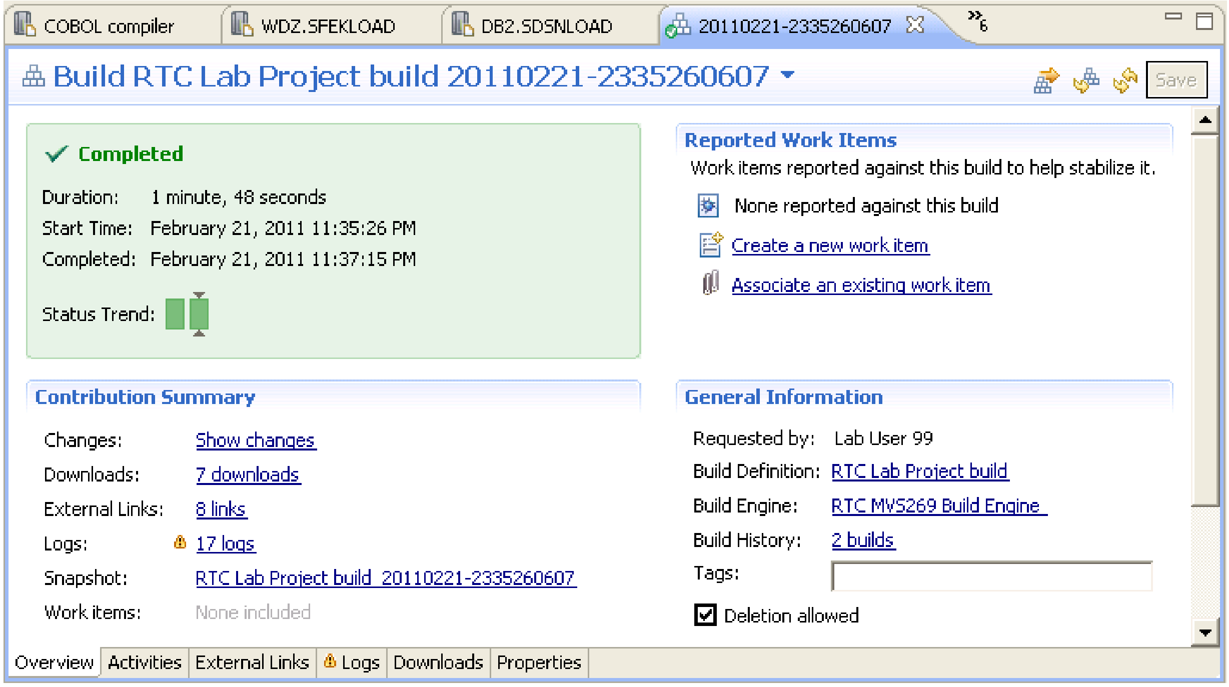
m) When the build runs, it will display in the builds view. If the builds view is not displayed double click on the build definition and it will display the builds window.



RTC Lab Project build - Found 1 Build (641 ms)

		Build	Label	Progress	Estimated Completion	Start Time
✓		RTC Lab Project build	20110212-0158…	Completed		February 12, 2011 1:…

n) You should now have now successfully run a build on the Mortgage Application. Your build should only have built what was required to be built based on the change you made. So, let's go look at a build report. Double click on the build you just ran, and you can see the build report. On the first page you see the general information about the build.



Build RTC Lab Project build 20110221-2335260607

✓ Completed

Duration: 1 minute, 48 seconds
Start Time: February 21, 2011 11:35:26 PM
Completed: February 21, 2011 11:37:15 PM

Status Trend:

Reported Work Items

Work items reported against this build to help stabilize it.

None reported against this build

Create a new work item

Associate an existing work item

Contribution Summary

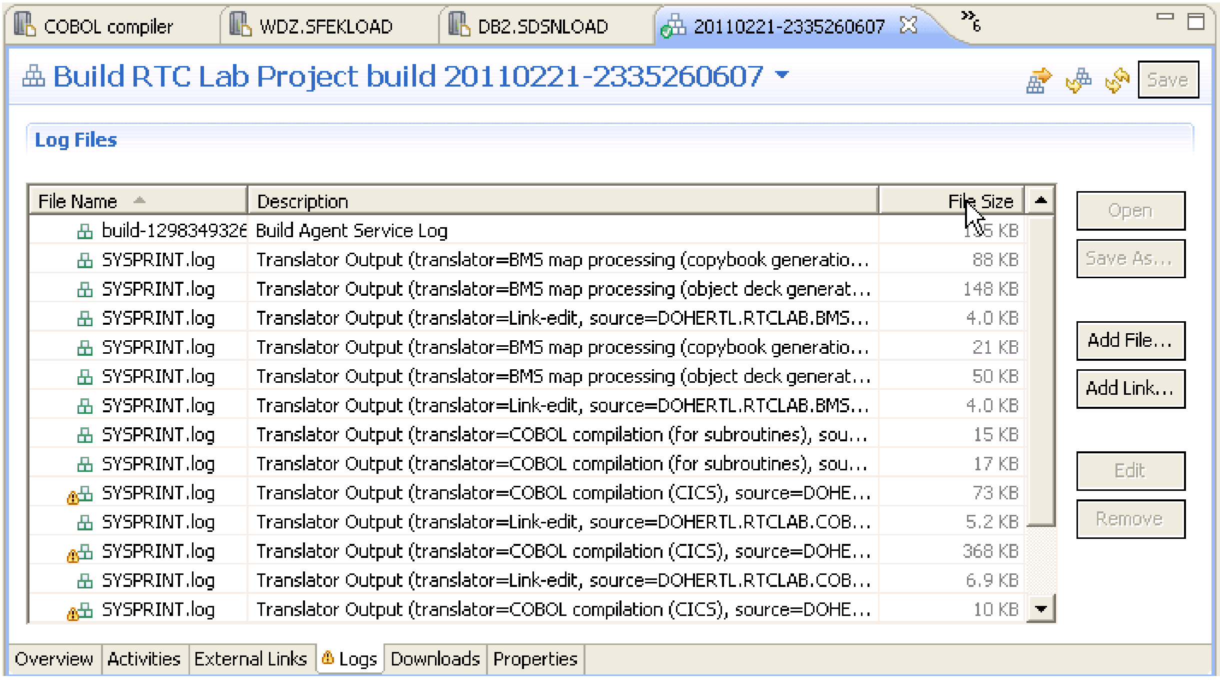
Changes:	Show changes
Downloads:	7 downloads
External Links:	8 links
Logs:	17 logs
Snapshot:	RTC Lab Project build_20110221-2335260607
Work items:	None included

General Information

Requested by:	Lab User 99
Build Definition:	RTC Lab Project build
Build Engine:	RTC MVS269 Build Engine
Build History:	2 builds
Tags:	

☑ Deletion allowed

There are also links to downloads and logs. If you click on the Logs page you will see all the output produced for the build, in this case the compiler output for COBOL compiles and BMS Map assembles:



Build RTC Lab Project build 20110221-2335260607

Log Files

File Name	Description	File Size
build-1298349326	Build Agent Service Log	135 KB
SYSPRINT.log	Translator Output (translator=BMS map processing (copybook generatio...	88 KB
SYSPRINT.log	Translator Output (translator=BMS map processing (object deck generat...	148 KB
SYSPRINT.log	Translator Output (translator=Link-edit, source=DOHERTL.RTCLAB.BMS...	4.0 KB
SYSPRINT.log	Translator Output (translator=BMS map processing (copybook generatio...	21 KB
SYSPRINT.log	Translator Output (translator=BMS map processing (object deck generat...	50 KB
SYSPRINT.log	Translator Output (translator=Link-edit, source=DOHERTL.RTCLAB.BMS...	4.0 KB
SYSPRINT.log	Translator Output (translator=COBOL compilation (for subroutines), sou...	15 KB
SYSPRINT.log	Translator Output (translator=COBOL compilation (for subroutines), sou...	17 KB
⚠ SYSPRINT.log	Translator Output (translator=COBOL compilation (CICS), source=DOHE...	73 KB
SYSPRINT.log	Translator Output (translator=Link-edit, source=DOHERTL.RTCLAB.COB...	5.2 KB
⚠ SYSPRINT.log	Translator Output (translator=COBOL compilation (CICS), source=DOHE...	368 KB
SYSPRINT.log	Translator Output (translator=Link-edit, source=DOHERTL.RTCLAB.COB...	6.9 KB
⚠ SYSPRINT.log	Translator Output (translator=COBOL compilation (CICS), source=DOHE...	10 KB

Conclusion

This concludes Module 7. You now should be familiar with Enterprise build using Ant and Buildforge

Module 8 Source code management with the ISPF Client

In this module, you will see how to start the RTC ISPF Client and use it to edit, check-in, deliver and build z/OS programs.

Many z/OS shops wanted to continue using the tools they were used to to work on mainframe programs, in particular ISPF edit. So RTC provided an ISPF Client that allowed the SCM operations of RTC to be performed through ISPF. Other operations around workitems, plans and admin would still be handled either from the Web UI or the Eclipse GUI.

Lab Scenario

You have been assigned the task of making a change to a COBOL program, testing it and then delivering it the stream for you colleagues to pick up. You will create a workitem through the web interface, and then use that workitem to make the changes to your code. Compile the code and then deliver it.

8.1. Logging into the SHARE LPAR

First of all you will need to logon to the SHARE LPAR.

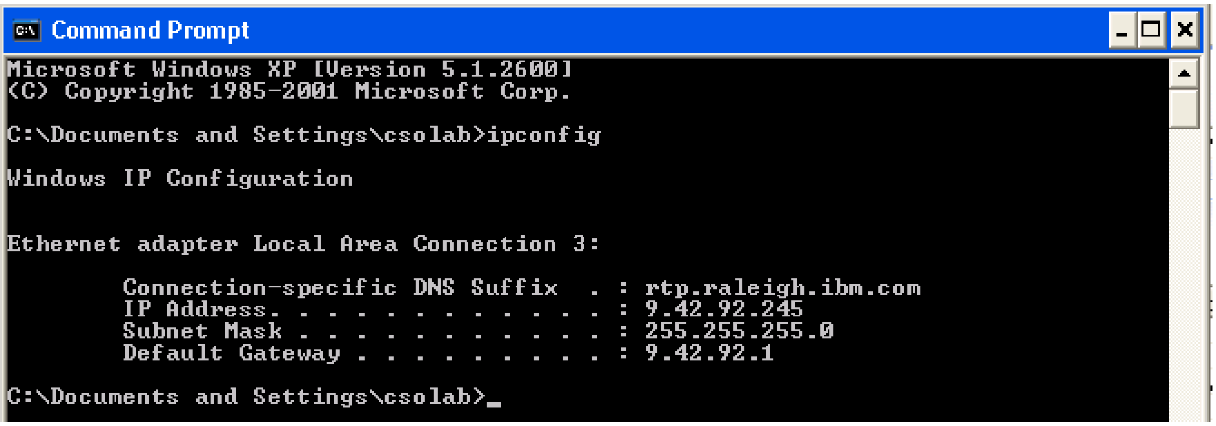
To login, follow these steps:

- Double click on the “mainframe” or “TSO” icon on your desktop. TSO.WS
- At the application prompt, type **TSO** and press Enter.
- At the prompt, enter your userid (**SHARAxx** – you will be allocated a number).
- On the TSO/E Logon Panel, enter your password which is **FIRSTPW**.
- You should be presented with the SHARE ISPF Primary Option menu

8.2. Starting and logging into to the ISPF Client

The ISPF Daemon is a started task that runs to handle all requests to the various RTC servers. The Daemon should already be running. If you are interested in looking at the daemon, go to SDSF with a prefix of ISPFDMN, and you will see the started task. Have a look through the job if you wish.

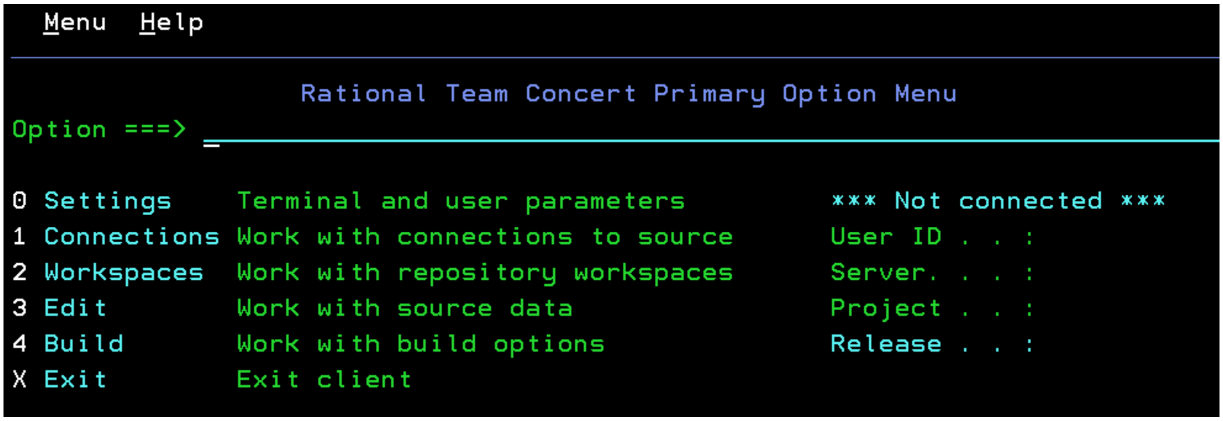
a) To start using the ISPF Client you will need to know the IP address of your RTC server running on your desktop. To find your IP address, open a command prompt and type **ipconfig** and press enter. You should see this:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\csolab>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 3:

        Connection-specific DNS Suffix  . : rtp.raleigh.ibm.com
        IP Address. . . . . . . . . . . . : 9.42.92.245
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 9.42.92.1

C:\Documents and Settings\csolab>_
```

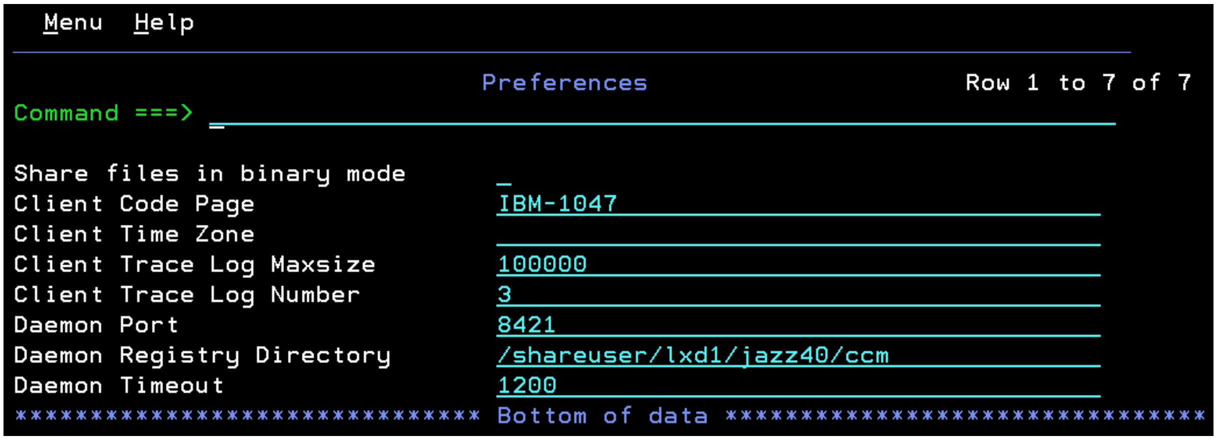
Remember your IP address, in this case 9.42.92.245.

b) To start the ISPF Client enter **RTC** on the command line and press Enter, you will see the RTC primary screen :



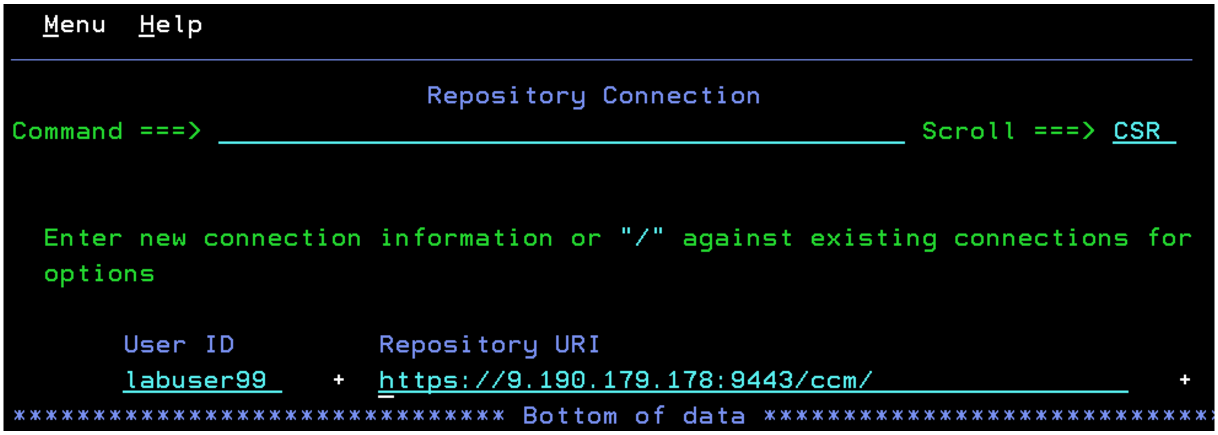
```
  Menu  Help
 ------------------------------------------------------------------------------
                  Rational Team Concert Primary Option Menu
Option ===> _

0 Settings     Terminal and user parameters          *** Not connected ***
1 Connections  Work with connections to source       User ID . . :
2 Workspaces   Work with repository workspaces       Server. . . :
3 Edit         Work with source data                 Project . . :
4 Build        Work with build options                Release . . :
X Exit         Exit client
```

c) First thing we will check are your settings. This will tell the RTC Client on which port the daemon is running and additionally what the daemon registry directory is. These have to match what is set in the daemon you are pointing to. For this LPAR the daemon port is **8421** and daemon registry is /**shareuser/lxd1/jazz/ccm.**



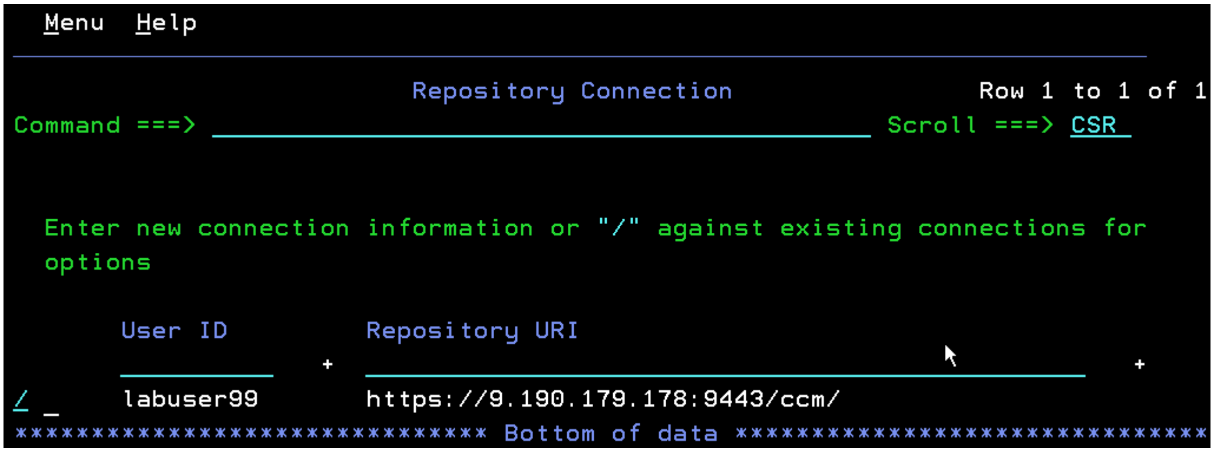
```
  Menu  Help
 ------------------------------------------------------------------------------
                              Preferences                          Row 1 to 7 of 7
Command ===> _

Share files in binary mode     _
Client Code Page               IBM-1047
Client Time Zone
Client Trace Log Maxsize       100000
Client Trace Log Number        3
Daemon Port                    8421
Daemon Registry Directory      /shareuser/lxd1/jazz40/ccm
Daemon Timeout                 1200
******************************* Bottom of data ********************************
```

d) Press Enter to go back to the primary menu. We are now going to connect to our RTC Repository through the daemon. Select option 1 – Connections and press Enter, you will be presented with an empty repository connection screen. First step is to add your userid and repository address into the fields provided. If you have created your own labuserxx userid you can use this, or else use labuser99 which has been created for you. For the repository address enter **https://*your-IP-address*:9443/ccm/** where your-IP-address is the IP address returned from your IP config command.



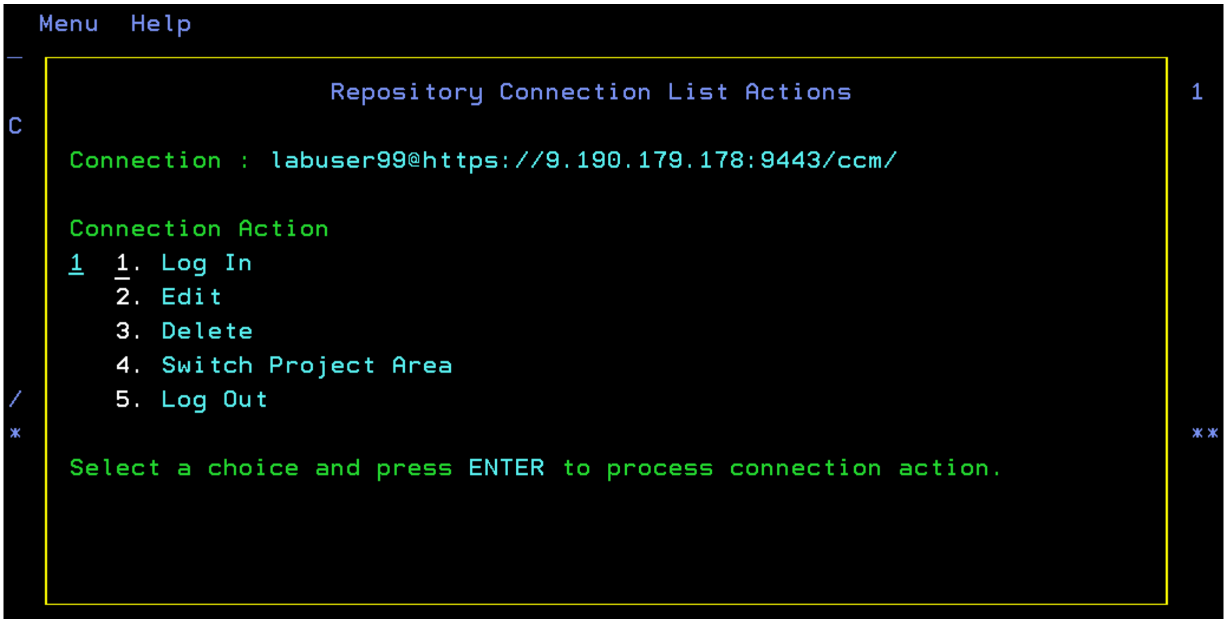
```
  Menu  Help
 ------------------------------------------------------------------------------
                         Repository Connection
Command ===> ____________________________________________ Scroll ===> CSR_

  Enter new connection information or "/" against existing connections for
  options

       User ID          Repository URI
       labuser99    +   https://9.190.179.178:9443/ccm/________________   +
******************************* Bottom of data ******************************
```

e) Press enter and the connection will be added to the list. If you have multiple servers you can connect to you can add them all to the list to use for connection. Now you can enter an **L** next to the connection to login, or alternatively enter a "/" to see all available options:



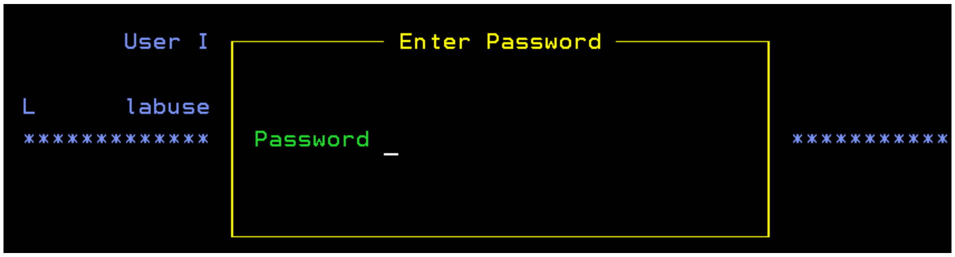
```
  Menu  Help
 ------------------------------------------------------------------------------
                         Repository Connection                 Row 1 to 1 of 1
Command ===> ____________________________________________ Scroll ===> CSR_

  Enter new connection information or "/" against existing connections for
  options

       User ID          Repository URI
       __________   +   ______________________________________________   +
/ _    labuser99        https://9.190.179.178:9443/ccm/
******************************* Bottom of data ******************************
```

You will be presented with a list of options, so just select **1** to login:



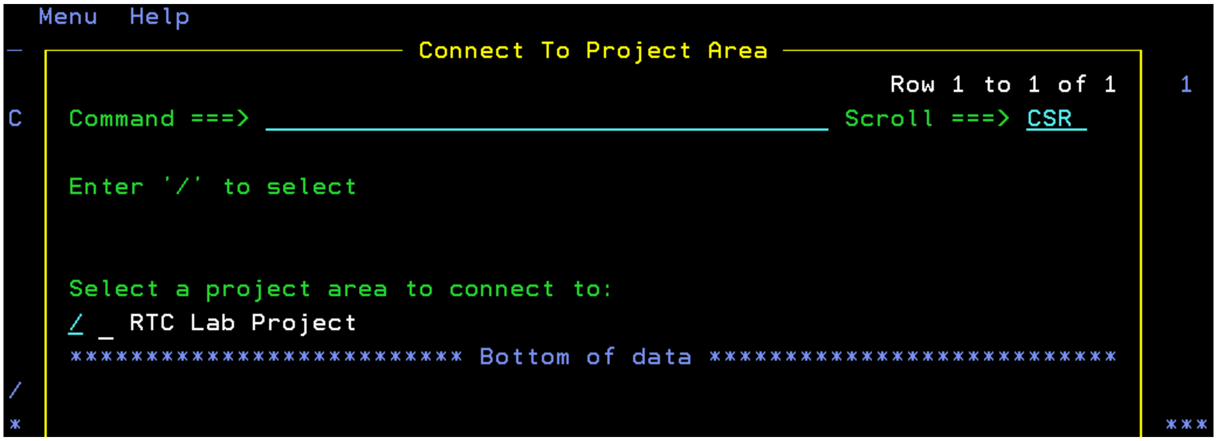
```
  Menu  Help
_ ┌──────────────────────────────────────────────────────────────────┐
  │                 Repository Connection List Actions               │ 1
C │                                                                  │
  │ Connection : labuser99@https://9.190.179.178:9443/ccm/           │
  │                                                                  │
  │ Connection Action                                                │
  │ 1  1. Log In                                                     │
  │    2. Edit                                                       │
  │    3. Delete                                                     │
  │    4. Switch Project Area                                        │
/ │    5. Log Out                                                    │
* │                                                                  │ **
  │ Select a choice and press ENTER to process connection action.    │
  │                                                                  │
  └──────────────────────────────────────────────────────────────────┘
```

f) You will be presented now with a password screen where you can enter your jazz repository password, which is the same as your userid, or if you are using labuser99, it will be labuser99. The password is a protected field so you will not see it as your type.



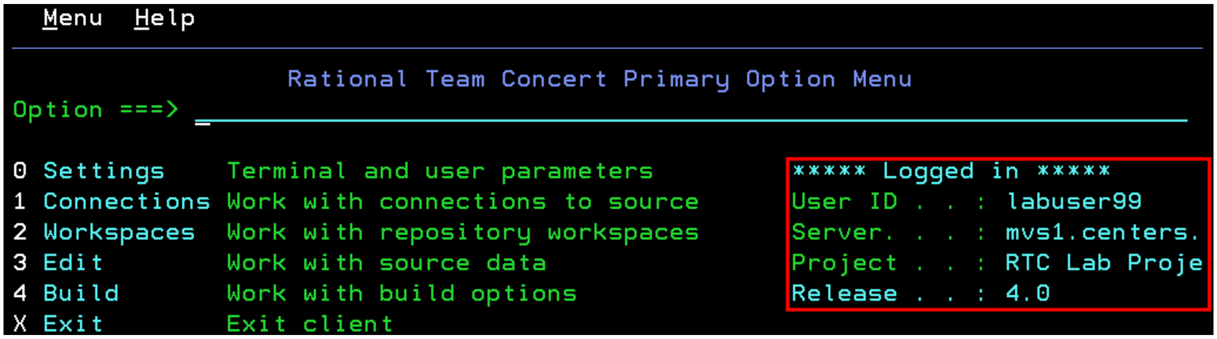
```
      User I ┌───────── Enter Password ─────────┐
             │                                  │
L     labuse │                                  │
************ │ Password _                       │ ***********
             │                                  │
             └──────────────────────────────────┘
```

g) If you have never logged in before you will be presented with a screen to select a project area. Your RTC repository may have multiple project areas you can select from. The connection panel is also the place you can switch between project areas. For this lab there is a single project area, so select it with a "/":



```
  Menu  Help
_ ┌──────────────────────── Connect To Project Area ────────────────┐
  │                                                 Row 1 to 1 of 1 │ 1
C │ Command ===> ________________________________   Scroll ===> CSR  │
  │                                                                 │
  │ Enter '/' to select                                             │
  │                                                                 │
  │                                                                 │
  │ Select a project area to connect to:                            │
  │ / _ RTC Lab Project                                             │
  │ ************************* Bottom of data ************************** │
/ │                                                                 │
* │                                                                 │ ***
```

h) You should now be logged in and presented with the primary options panel to inform you of that:



```
 Menu  Help
 ------------------------------------------------------------------------------
                  Rational Team Concert Primary Option Menu
Option ===> _

0 Settings    Terminal and user parameters          ***** Logged in *****
1 Connections Work with connections to source       User ID . . : labuser99
2 Workspaces  Work with repository workspaces       Server. . . : mvs1.centers.
3 Edit        Work with source data                 Project . . : RTC Lab Proje
4 Build       Work with build options               Release . . : 4.0
X Exit        Exit client
```

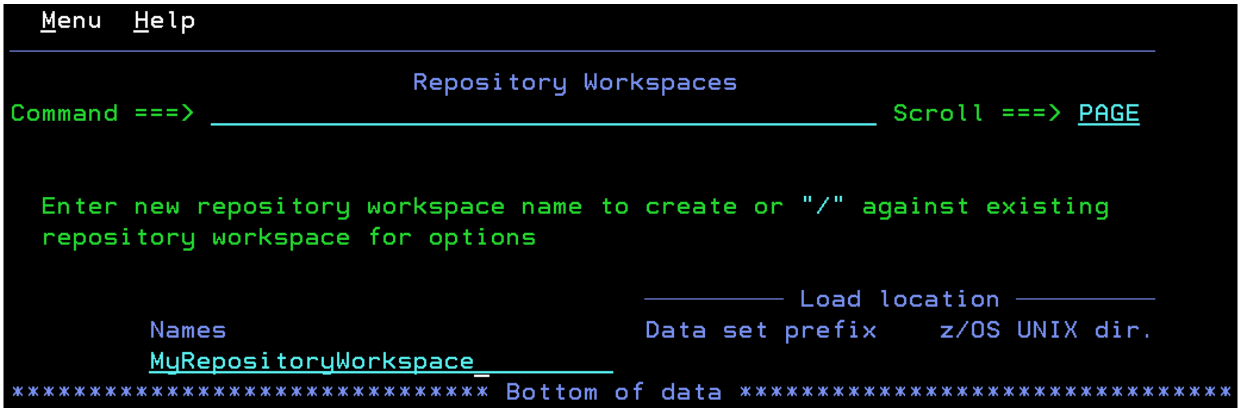
Note: Many of the fields in RTC are what are know as scrollable fields in ISPF. So for fields where the information is too long for the available width you can still get the rest of the information. Position the cursor somewhere in a field, for example under the actual project name “RTC Lab Proje..”, then press PF4. A panel is displayed with the information. Some scrollable fields are updatable, so you can press PF4 to enter additional data. Alternatively, with the cursor in a scrollable field just press PF10/PF11 to scroll the available data left and right.

8.3. Working with your repository workspaces

The first thing we need to do is load some PDSs with source code data from the RTC repository. On z/OS the source needs to be in normal PDSs so that ISPF can edit the code and compilers can compile it. The actions of the Eclipse GUI such as LOAD and SHARE have been propagated to the ISPF Client, so regardless of which interface you use the processes are basically the same. So lets get started…

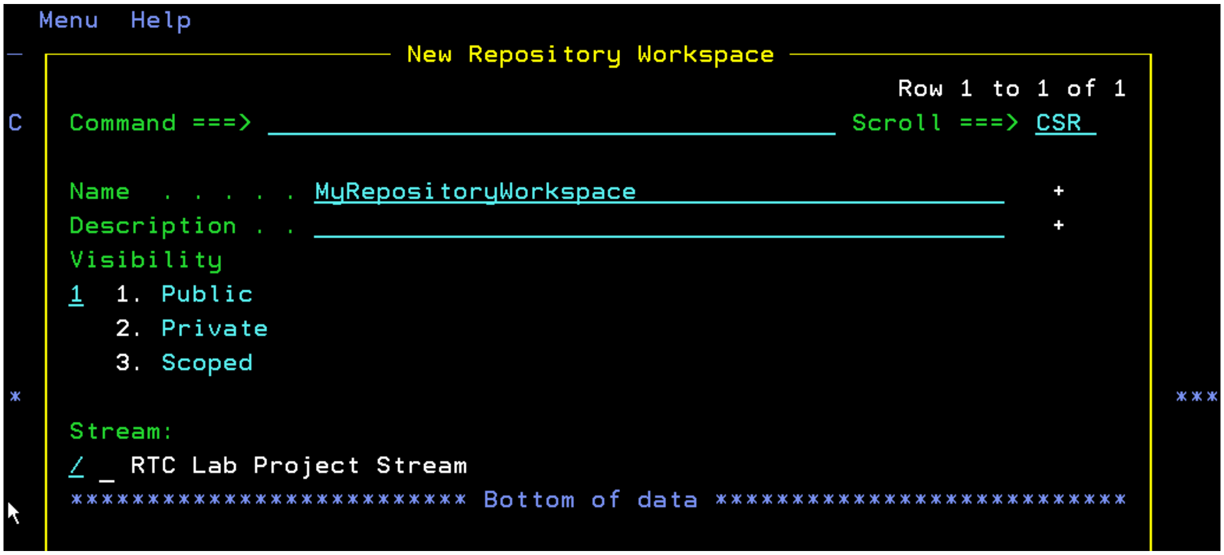
a) From the primary option menu sect option 2, to work with your repository workpaces. Depending on which other lab modules you have worked will have an affect on what you see next. If you are using LABUSER99 to log in, then you will see a screen with an existing repository workspace. If you previously created a repository workspace, then again, you will have one listed on the screen presented. If this is the case, skip to bullet **d)** to continue.

However, if you have dived straight into the ISPF Lab with your own userid, you may be presented with an empty screen. If that is the case, enter a repository workpace name, whatever you like, in the Name field provided and press enter:



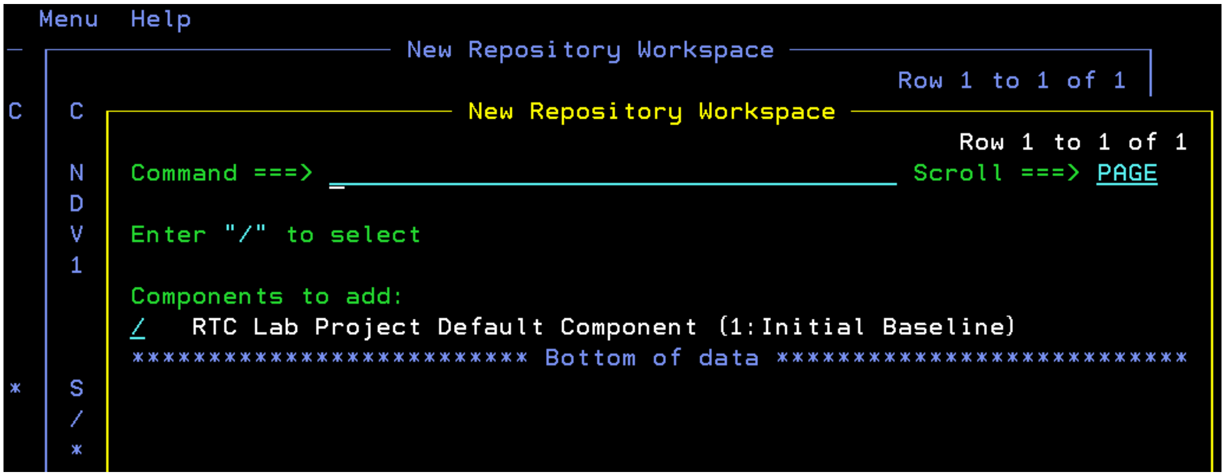
```
 Menu  Help
 ---------------------------------------------------------------------------
                         Repository Workspaces
Command ===> ____________________________________________ Scroll ===> PAGE


  Enter new repository workspace name to create or "/" against existing
  repository workspace for options

                                  --------- Load location ---------
         Names                    Data set prefix     z/OS UNIX dir.
         MyRepositoryWorkspace_
******************************* Bottom of data ********************************
```

b) Enter the workspace visibility and select a stream to flow with using a “/”, there should only be one in the list, then press enter:



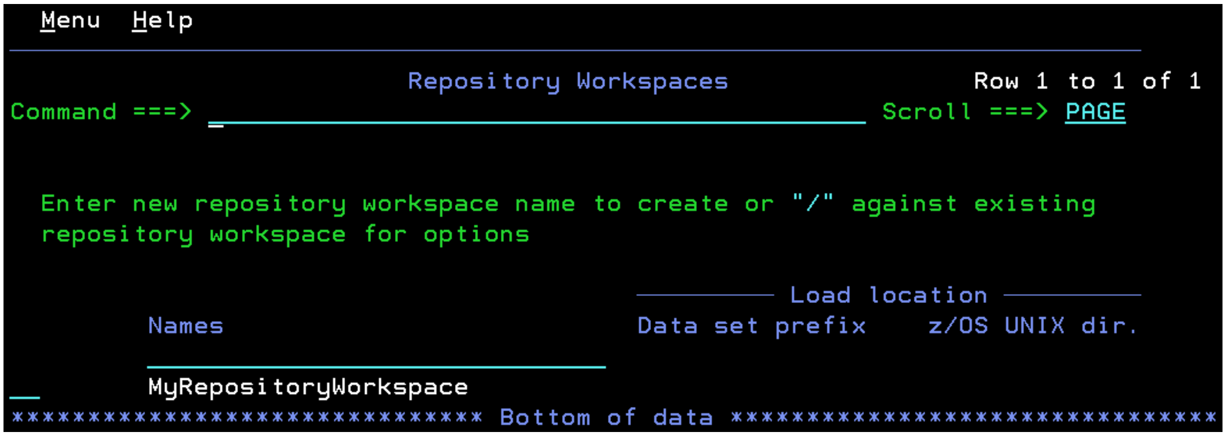
```
   Menu   Help
 ─ ┌──────────────────────── New Repository Workspace ─────────────────────┐
   │                                                         Row 1 to 1 of 1
 C │ Command ===> ______________________________________ Scroll ===> CSR
   │
   │ Name . . . . . MyRepositoryWorkspace_____________________________   +
   │ Description . . ______________________________________________   +
   │ Visibility
   │ 1  1. Public
   │    2. Private
   │    3. Scoped
 * │                                                                         ***
   │ Stream:
   │ / _ RTC Lab Project Stream
   │ ************************** Bottom of data ***************************
```

c) Select the component you want to include, again in this lab there should be only one, and press enter:



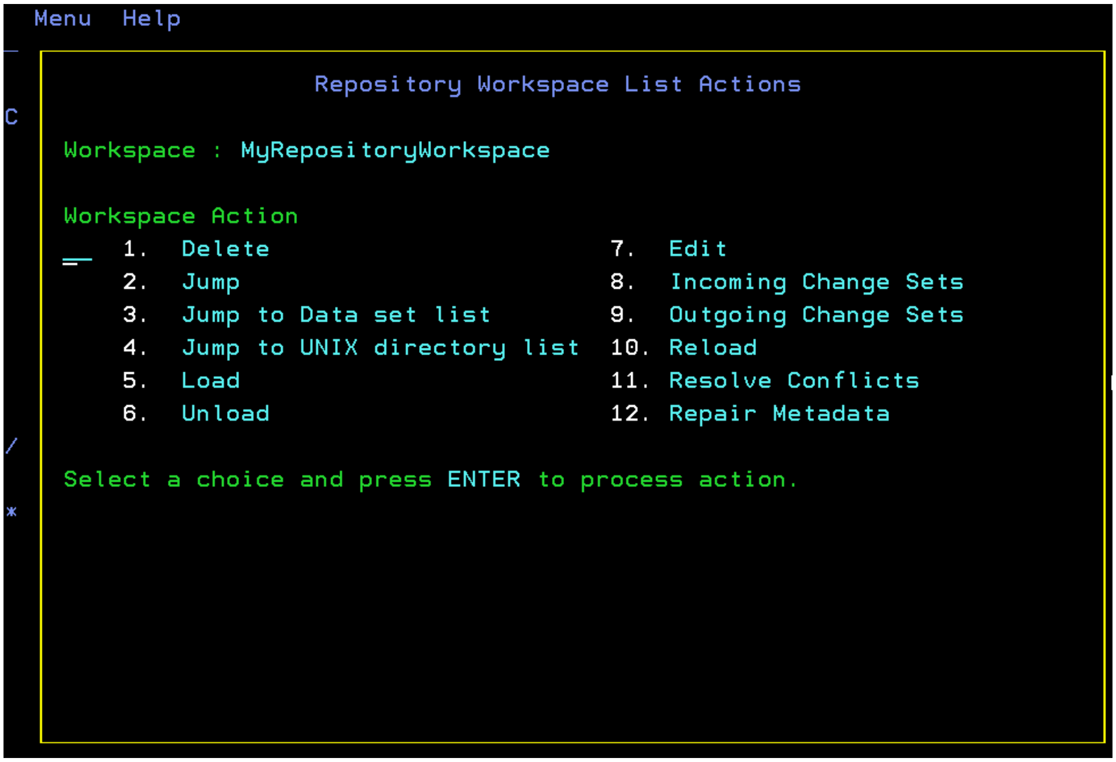
```
   Menu   Help
 ─ ┌──────────────────────── New Repository Workspace ─────────────────────┐
   │                                                         Row 1 to 1 of 1
 C │ C ┌──────────────────────── New Repository Workspace ─────────────────────┐
   │   │                                                         Row 1 to 1 of 1
   │ N │ Command ===> _______________________________________ Scroll ===> PAGE
   │ D │
   │ V │ Enter "/" to select
   │ 1 │
   │   │ Components to add:
   │   │ /   RTC Lab Project Default Component (1:Initial Baseline)
   │   │ ************************** Bottom of data ***************************
 * │ S │
   │ / │
   │ * │
```

d) You will now be presented with a list containing your repository workspace, where you will now be able to perform your next actions:



```
   Menu   Help
 ──────────────────────────────────────────────────────────────────────────────
                             Repository Workspaces                Row 1 to 1 of 1
 Command ===> ___________________________________________ Scroll ===> PAGE


   Enter new repository workspace name to create or "/" against existing
   repository workspace for options

                                                ──────── Load location ────────
          Names                                 Data set prefix    z/OS UNIX dir.
          ______________________________
 __       MyRepositoryWorkspace
 ******************************** Bottom of data ********************************
```

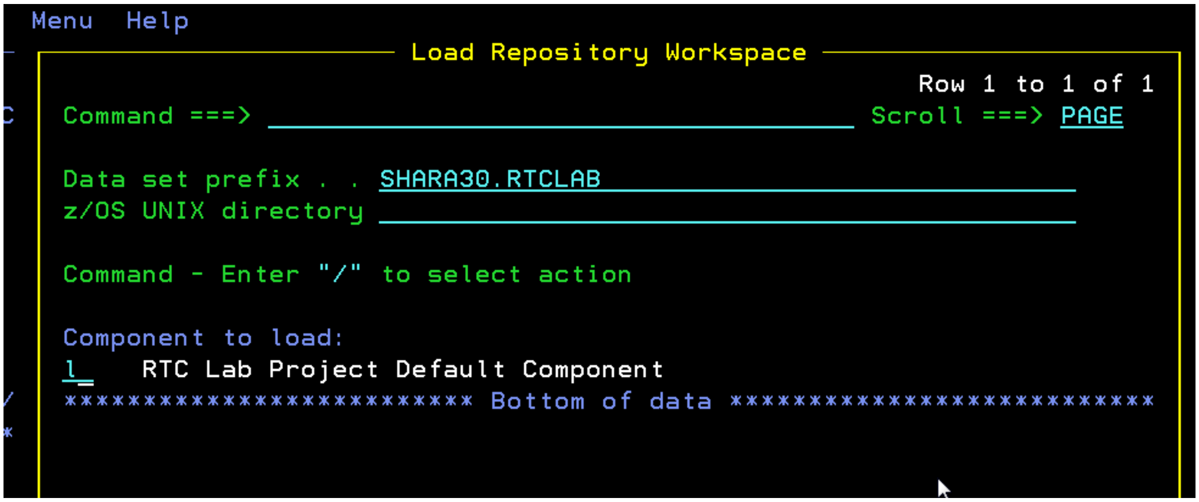
e) If you enter a “/” next to the repository workspace you will see a list of actions that can be performed.



```
  Menu  Help
 ─
     ┌───────────────────────────────────────────────────────────────────────┐
     │                   Repository Workspace List Actions                   │
 C   │                                                                       │
     │ Workspace : MyRepositoryWorkspace                                     │
     │                                                                       │
     │ Workspace Action                                                      │
     │ __  1.  Delete                         7.  Edit                       │
     │     2.  Jump                           8.  Incoming Change Sets       │
     │     3.  Jump to Data set list          9.  Outgoing Change Sets       │
     │     4.  Jump to UNIX directory list   10.  Reload                     │
     │     5.  Load                          11.  Resolve Conflicts          │
     │     6.  Unload                        12.  Repair Metadata            │
 /   │                                                                       │
     │ Select a choice and press ENTER to process action.                    │
 *   │                                                                       │
     └───────────────────────────────────────────────────────────────────────┘
```

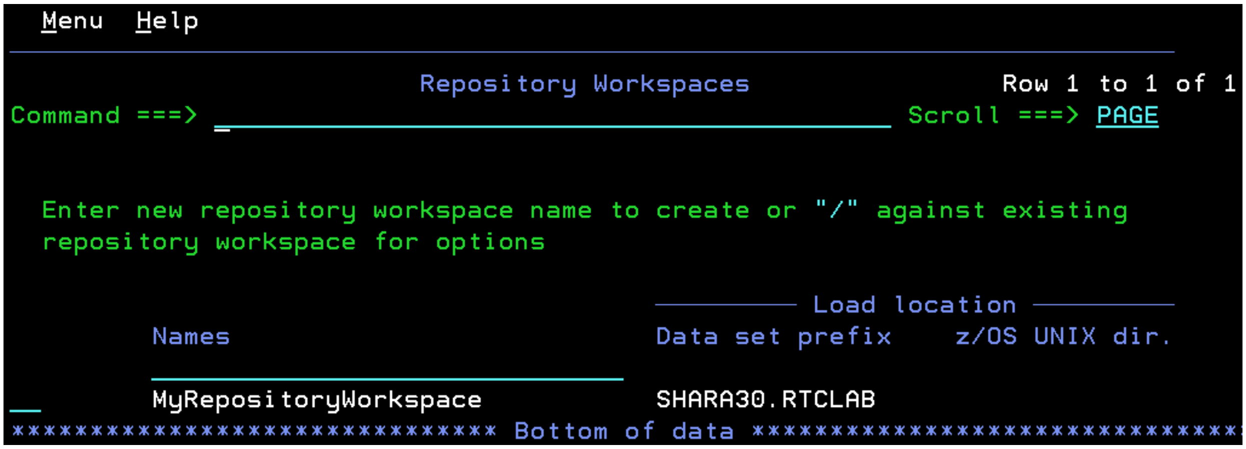
The main ones will be Load, Jump, Incoming Change Sets and Outgoing Change Sets. So we will concentrate on these for now.

f) The first thing we need to do is load a set of PDSs with the source code held in the repository so that we can begin working on it. Either from the list shown above select option **5** to perform a Load, or if you are still on the repository workspace list screen, enter a **L** to load. You will be presented with a screen where you have to enter your data set prefix. This is a high level qualifier, and most likely a middle level qualifier to identify where RTC is going to load the source data. You high level qualifier for this lab will need to be you SHARE LPAR userid, **SHARAxx**, that was assigned to you. For the middle level qualifier use something like **RTCLAB**. In the example shown the HLQ is **DOHERTL**. You can ignore the z/OS UNIX directory location for this Lab.



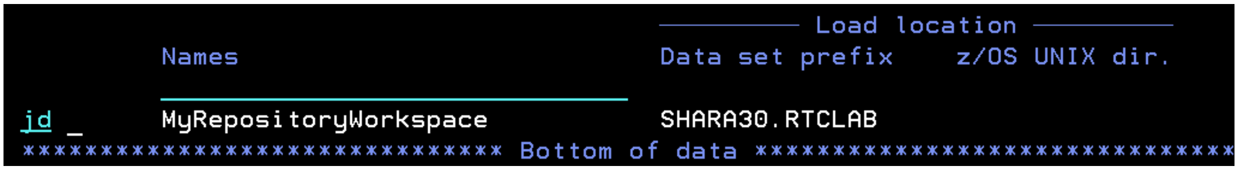
```
  Menu  Help
 ─
     ┌───────────────────── Load Repository Workspace ────────────────────────┐
     │                                                       Row 1 to 1 of 1  │
 C   │ Command ===> ____________________________________ Scroll ===> PAGE     │
     │                                                                        │
     │ Data set prefix . . SHARA30.RTCLAB____________________________________ │
     │ z/OS UNIX directory ___________________________________________        │
     │                                                                        │
     │ Command - Enter "/" to select action                                   │
     │                                                                        │
     │ Component to load:                                                     │
     │ l_   RTC Lab Project Default Component                                 │
 /   │ ************************** Bottom of data **************************** │
 *   │                                                                        │
```

Enter an **L** next to the component to load and press **enter**. You will be presented with a confirmation screen where you can just press enter again. The load may take some time as it transfers all of the source code data from the RTC repository to the PDS data sets it will create for you. RTC uses the data set definitions discussed in Module 6 to know what attributes to allocate a data set with. Once the load has finished you will be presented with the repository workspace screen, except the load location will now be filled in:



```
  Menu   Help
 ------------------------------------------------------------------------------
                          Repository Workspaces                 Row 1 to 1 of 1
Command ===> _                                             Scroll ===> PAGE


  Enter new repository workspace name to create or "/" against existing
  repository workspace for options

                                          --------- Load location ---------
         Names                            Data set prefix    z/OS UNIX dir.

         ______________________________
__       MyRepositoryWorkspace            SHARA30.RTCLAB
******************************* Bottom of data ********************************
```

g) From here we can now easily jump to a data set list with our loaded data sets so we can begin making code changes. Enter a **JD** next to the repository workspace, and you will be taken to the data set list (similar to ISPF option 3.4) except for the data set pattern specified in the load location.



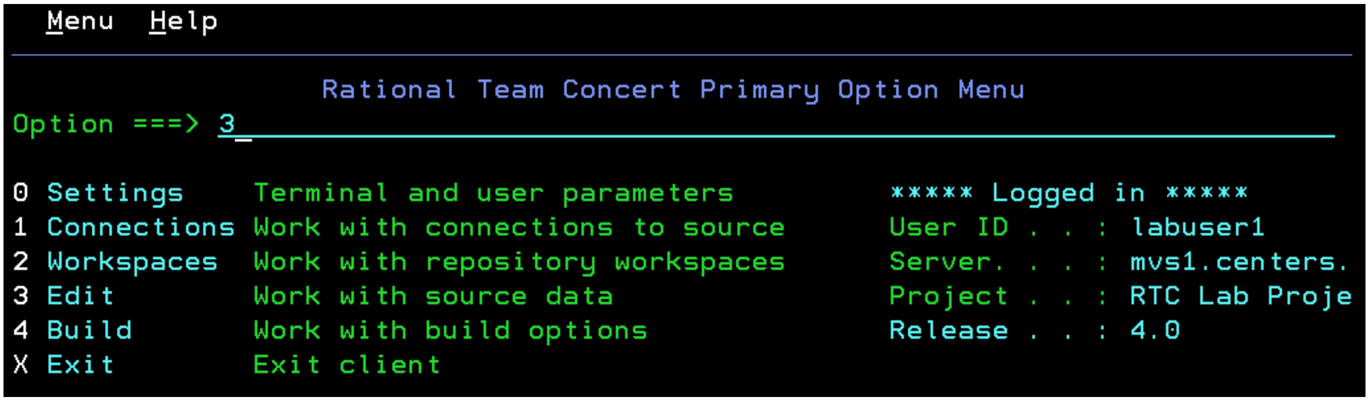
```
                                          --------- Load location ---------
         Names                            Data set prefix    z/OS UNIX dir.

         ______________________________
jd _     MyRepositoryWorkspace            SHARA30.RTCLAB
******************************* Bottom of data ********************************
```

The next section will talk you though making program changes and checking programs in.

8.4. Making changes to programs

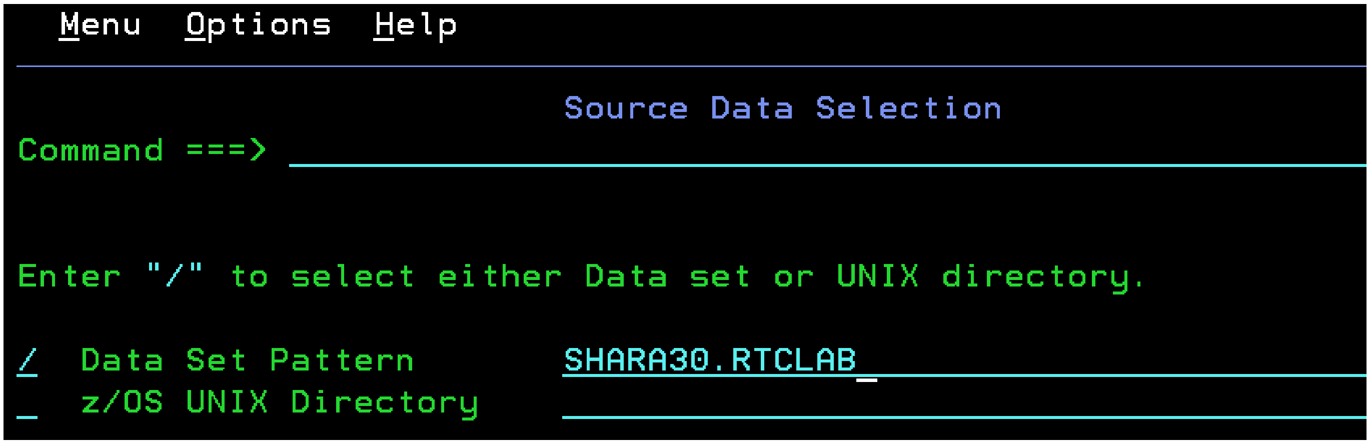
There are two ways to start editing members. The first is to use the Jump option from the repository workspace panel, to jump directly to the source code data panels, which we used in the previous section. You can use the **J** command to jump to the main option 3 panel, or the **JD** to jump directly to the data set list, or the **JU** to jump directly to the z/OS Unix directory list.

Alternatively you can start with option 3 on the RTC primary menu:



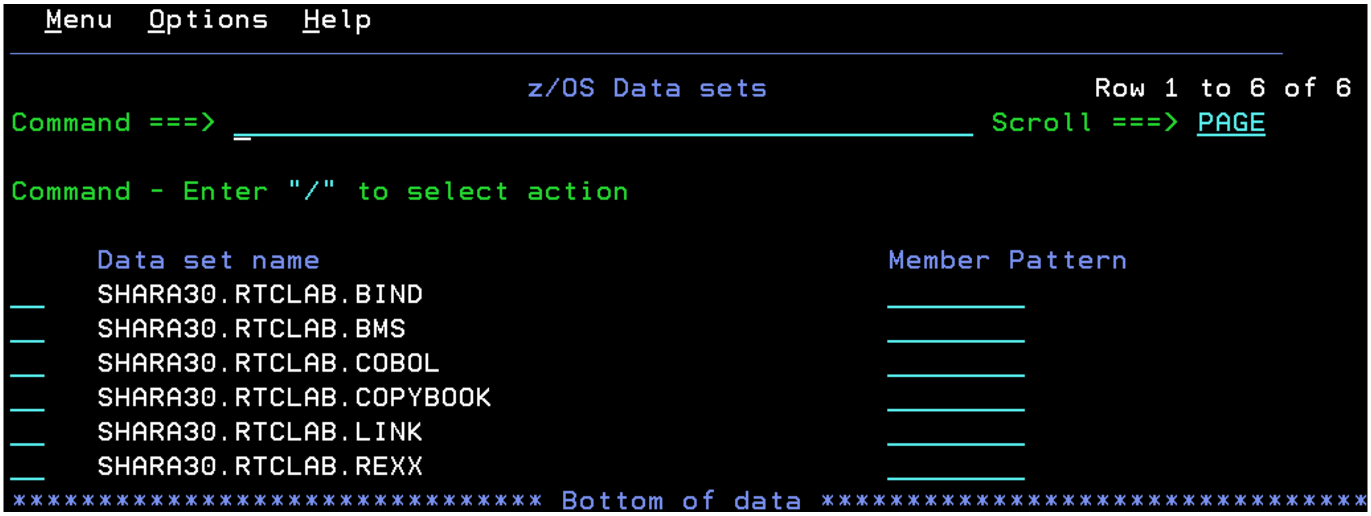
```
  Menu  Help
 ------------------------------------------------------------------------------
                  Rational Team Concert Primary Option Menu
Option ===> 3_

0 Settings    Terminal and user parameters       ***** Logged in *****
1 Connections Work with connections to source    User ID . . : labuser1
2 Workspaces  Work with repository workspaces    Server. . . : mvs1.centers.
3 Edit        Work with source data              Project . . : RTC Lab Proje
4 Build       Work with build options            Release . . : 4.0
X Exit        Exit client
```

In this case you will be given a panel where you can enter your data set pattern, which works the same as ISPF option 3.4 or ISPF option 3.17 for z/OS Unix directories. In this case enter your data set pattern, which will be SHARAxx.RTCLAB, or whatever you used when you did your load.



```
  Menu  Options  Help
 ------------------------------------------------------------------------------
                            Source Data Selection
Command ===> ____________________________________________________________


Enter "/" to select either Data set or UNIX directory.

/  Data Set Pattern          SHARA30.RTCLAB_
_  z/OS UNIX Directory       ______________________________
```

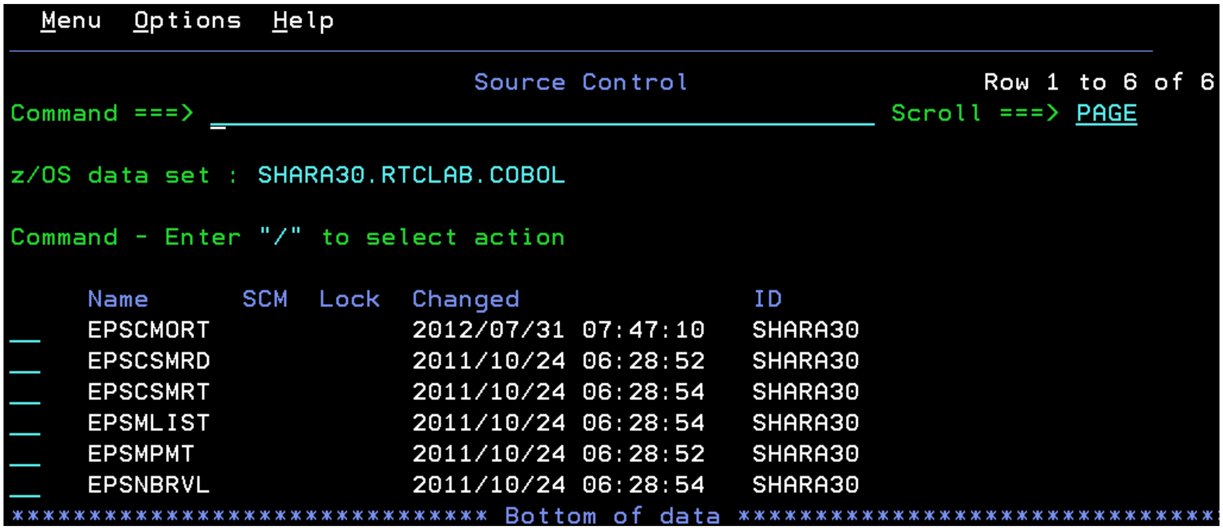
Entering a “/” next to the Data Set Pattern and pressing enter will show data sets that match this pattern, which is the same panel you would be presented with if you had used the Jump option from the repository workspace panel:



```
  Menu  Options  Help
 ------------------------------------------------------------------------------
                          z/OS Data sets                     Row 1 to 6 of 6
Command ===> _________________________________________    Scroll ===> PAGE

Command - Enter "/" to select action

     Data set name                                     Member Pattern
__   SHARA30.RTCLAB.BIND                               ________
__   SHARA30.RTCLAB.BMS                                ________
__   SHARA30.RTCLAB.COBOL                              ________
__   SHARA30.RTCLAB.COPYBOOK                           ________
__   SHARA30.RTCLAB.LINK                               ________
__   SHARA30.RTCLAB.REXX                               ________
******************************* Bottom of data ********************************
```

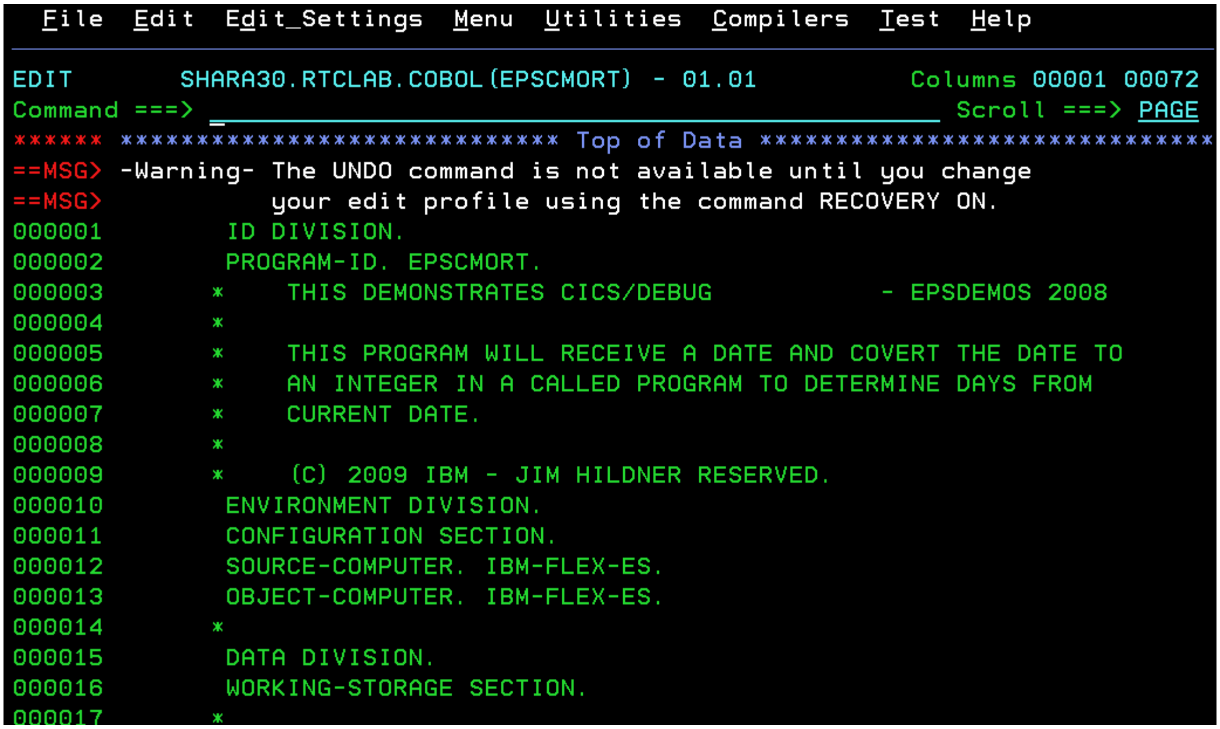
So we now have a list of data sets we can start working with to make some code changes. So let’s get started on that…

a) Enter an **E** next to the **SHARAxx.RTCLAB.COBOL** data set. Press enter and you will be presented with the member list for this dataset.



```
                              Source Control                        Row 1 to 6 of 6
Command ===> _                                           Scroll ===> PAGE

z/OS data set : SHARA30.RTCLAB.COBOL

Command - Enter "/" to select action

     Name       SCM  Lock  Changed               ID
___  EPSCMORT              2012/07/31 07:47:10   SHARA30
___  EPSCSMRD              2011/10/24 06:28:52   SHARA30
___  EPSCSMRT              2011/10/24 06:28:54   SHARA30
___  EPSMLIST              2011/10/24 06:28:54   SHARA30
___  EPSMPMT               2011/10/24 06:28:52   SHARA30
___  EPSNBRVL              2011/10/24 06:28:54   SHARA30
******************************* Bottom of data ********************************
```

b) Enter an **E** next to the **EPSCMORT** member and press enter. You are now in the normal ISPF editor for this member. We are going to make a simple change. Just add an extra comment line with a comment of your choice, using the previous comment lines as an example:

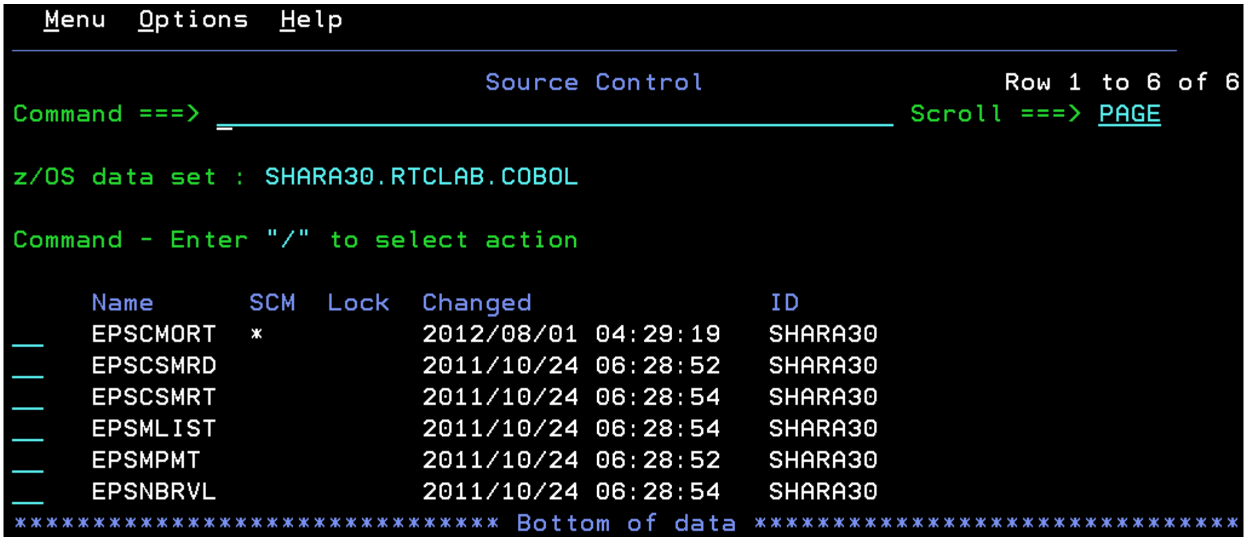


File Edit Edit_Settings Menu Utilities Compilers Test Help

```
EDIT       SHARA30.RTCLAB.COBOL(EPSCMORT) - 01.01          Columns 00001 00072
Command ===> _                                                 Scroll ===> PAGE
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001        ID DIVISION.
000002        PROGRAM-ID. EPSCMORT.
000003       *    THIS DEMONSTRATES CICS/DEBUG           - EPSDEMOS 2008
000004       *
000005       *    THIS PROGRAM WILL RECEIVE A DATE AND COVERT THE DATE TO
000006       *    AN INTEGER IN A CALLED PROGRAM TO DETERMINE DAYS FROM
000007       *    CURRENT DATE.
000008       *
000009       *    (C) 2009 IBM - JIM HILDNER RESERVED.
000010        ENVIRONMENT DIVISION.
000011        CONFIGURATION SECTION.
000012        SOURCE-COMPUTER. IBM-FLEX-ES.
000013        OBJECT-COMPUTER. IBM-FLEX-ES.
000014       *
000015        DATA DIVISION.
000016        WORKING-STORAGE SECTION.
000017       *
```

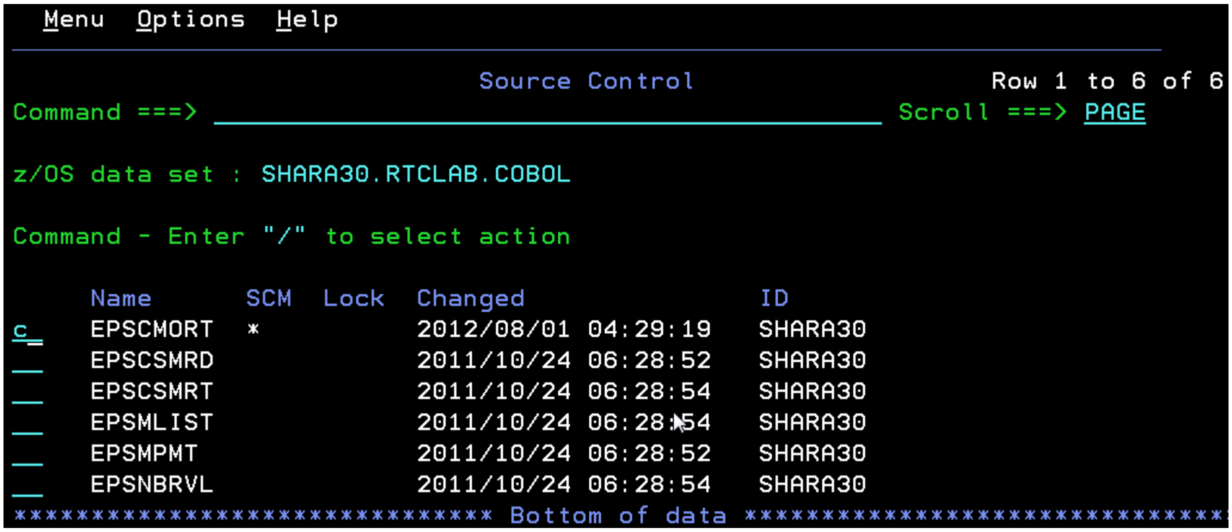
Note: In COBOL an asterix in column 7 denotes a comment line.

c) Press **PF3** to save and exit the member, and you will see that the member list display has been updated to show that you have edited the member. The * in the SCM column shows that you have an unchecked in change.



```
   Menu  Options  Help
 ──────────────────────────────────────────────────────────────────────────
                                Source Control                 Row 1 to 6 of 6
Command ===> _                                            Scroll ===> PAGE

z/OS data set : SHARA30.RTCLAB.COBOL

Command - Enter "/" to select action

     Name       SCM  Lock  Changed              ID
__   EPSCMORT   *          2012/08/01 04:29:19  SHARA30
__   EPSCSMRD              2011/10/24 06:28:52  SHARA30
__   EPSCSMRT              2011/10/24 06:28:54  SHARA30
__   EPSMLIST              2011/10/24 06:28:54  SHARA30
__   EPSMPMT               2011/10/24 06:28:52  SHARA30
__   EPSNBRVL              2011/10/24 06:28:54  SHARA30
******************************* Bottom of data ********************************
```

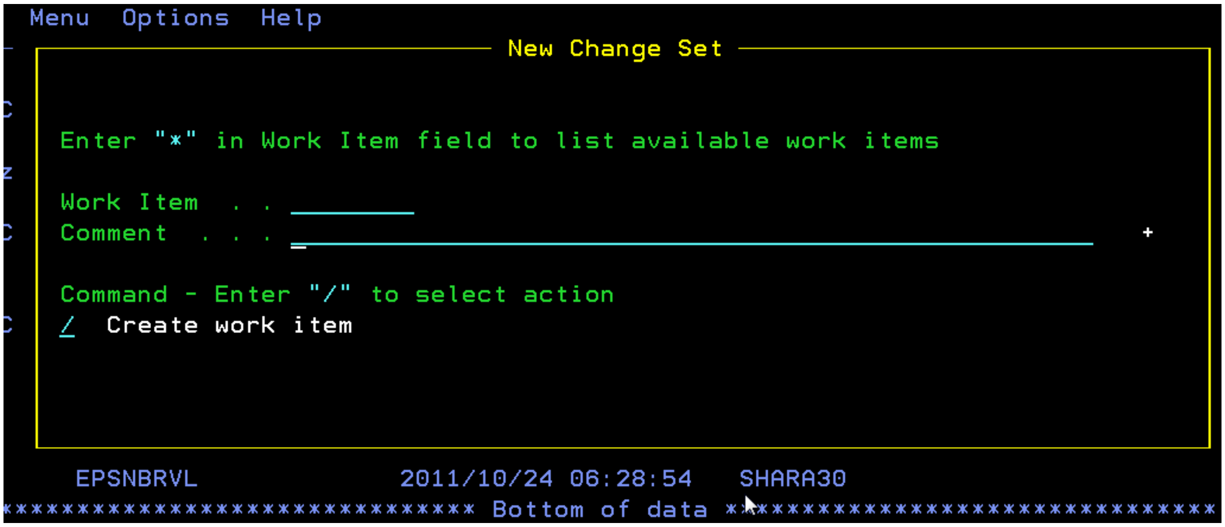
d) In order to check the member in, you are going to need to associate it with a work item. Next to the member you changed, enter a **C** to check the change in, and press enter:



```
   Menu  Options  Help
 ──────────────────────────────────────────────────────────────────────────
                                Source Control                 Row 1 to 6 of 6
Command ===>                                              Scroll ===> PAGE

z/OS data set : SHARA30.RTCLAB.COBOL

Command - Enter "/" to select action

     Name       SCM  Lock  Changed              ID
c_   EPSCMORT   *          2012/08/01 04:29:19  SHARA30
__   EPSCSMRD              2011/10/24 06:28:52  SHARA30
__   EPSCSMRT              2011/10/24 06:28:54  SHARA30
__   EPSMLIST              2011/10/24 06:28:54  SHARA30
__   EPSMPMT               2011/10/24 06:28:52  SHARA30
__   EPSNBRVL              2011/10/24 06:28:54  SHARA30
******************************* Bottom of data ********************************
```

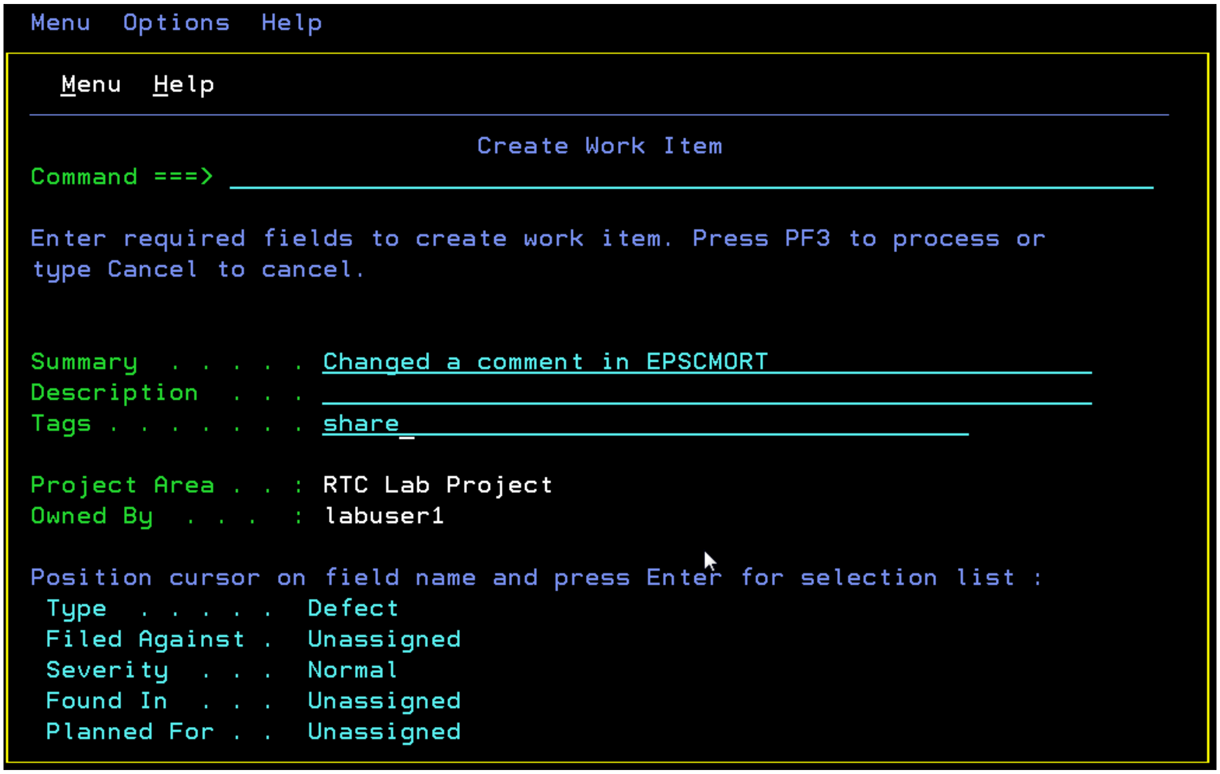
e) You will be presented with a panel where you can now associate your changeset with an existing work item and add an optional comment or create a new work item. If you wish to enter it with an existing work item (less typing ☺) enter a * in the work item field and if you wish a comment in the comment line. Or if you know the work item number you can, rather than entering an * to get a list, just enter the work item number directly.

However for this lab we will create a new work item. Enter a “/” in next to the Create work item field on the screen.



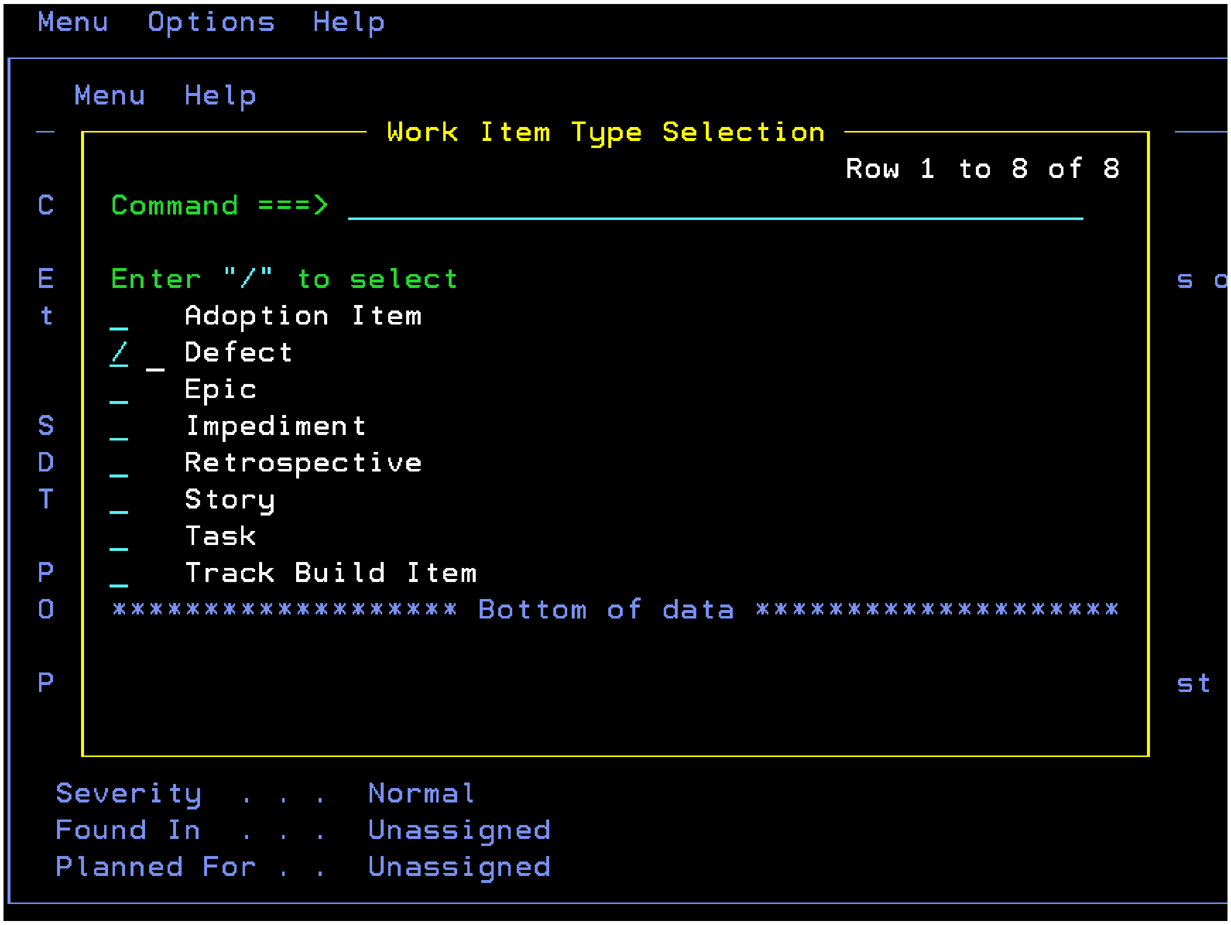
```
 Menu  Options  Help
                                New Change Set

  Enter "*" in Work Item field to list available work items

  Work Item  . . ________
  Comment  . . . _____________________________________________________  +

  Command - Enter "/" to select action
  /  Create work item

    EPSNBRVL             2011/10/24 06:28:54    SHARA30
****************************** Bottom of data ******************************
```

f) Once you press enter you will be presented with the Create Work Item screen. Enter a summary and description if you wish. These fields are scrollable, so pressing **PF4** while the cursor is in the field will enable you to enter extra data.



```
 Menu  Options  Help

   Menu  Help

                             Create Work Item
 Command ===> ____________________________________________________________

 Enter required fields to create work item. Press PF3 to process or
 type Cancel to cancel.

 Summary  . . . . . . Changed a comment in EPSCMORT____________________
 Description  . . .  __________________________________________________
 Tags . . . . . . . . share________________________________________

 Project Area . . :  RTC Lab Project
 Owned By  . . .  :  labuser1

 Position cursor on field name and press Enter for selection list :
  Type  . . . . . .  Defect
  Filed Against .    Unassigned
  Severity  . . .    Normal
  Found In  . . .    Unassigned
  Planned For . .    Unassigned
```

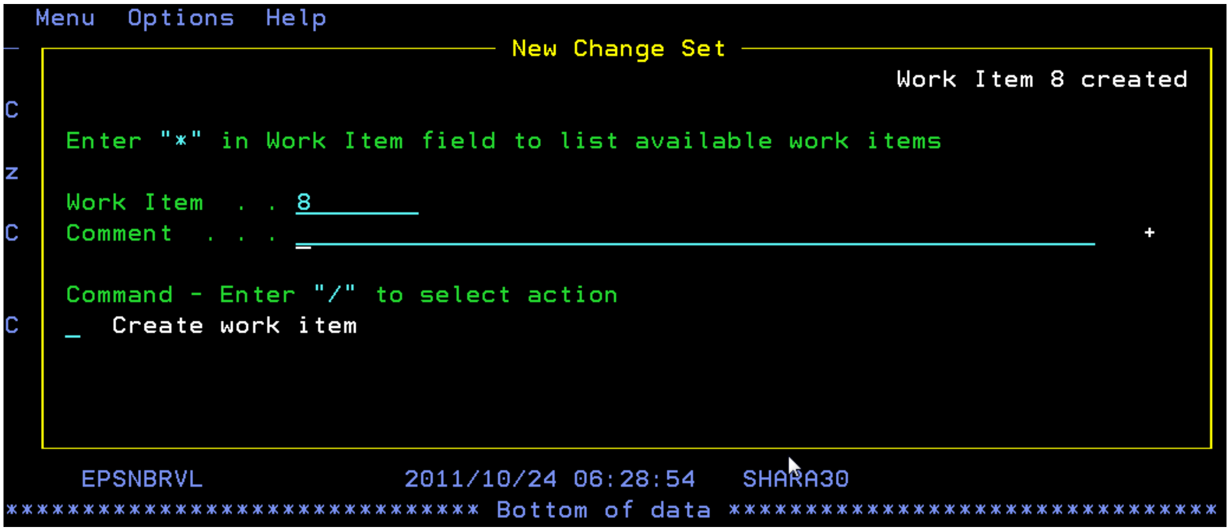
g) To set the ***Type***, ***Filed Against***, ***Severity***, ***Found in*** and ***Planned for*** values position your cursor on those fields (either the name or the value) and press enter. You will be presented with a selection panel that presents you with valid values retrieved from the server. You can select the value you wish with a “/”.



```
 Menu  Options  Help
 ──────────────────────────────────────────────────────────────────
   Menu  Help
 ─  ─────────────── Work Item Type Selection ──────────────  ─
   │                                        Row 1 to 8 of 8 │
 C │ Command ===> ________________________________________  │
   │                                                        │  
 E │ Enter "/" to select                                    │ s c
 t │ _   Adoption Item                                      │
   │ / _ Defect                                             │
   │ _   Epic                                               │
 S │ _   Impediment                                         │
 D │ _   Retrospective                                      │
 T │ _   Story                                              │
   │ _   Task                                               │
 P │ _   Track Build Item                                   │
 O │ ******************* Bottom of data ******************** │
   │                                                        │
 P │                                                        │ st
   │                                                        │
   └────────────────────────────────────────────────────────┘
  Severity . . .  Normal
  Found In  . . . Unassigned
  Planned For . . Unassigned
```

h) Once you have entered all the required fields on the Create Work Item screen press PF3 to create the work item.

i) You will be returned to the new change set screen where, now that you have a work item number assigned you can press enter again to associate the work item to the change.



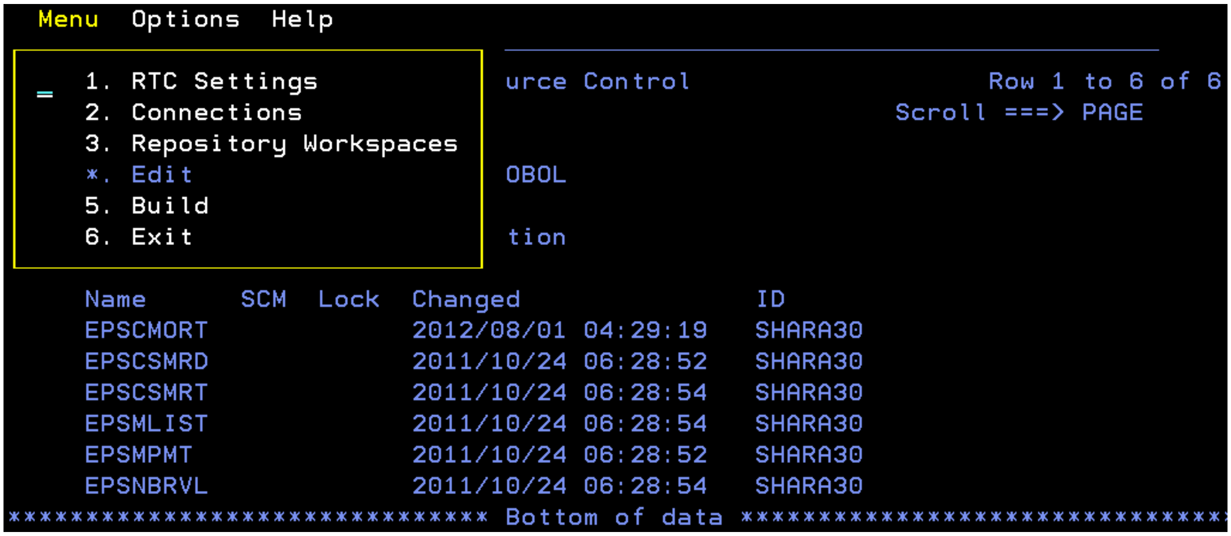
```
 Menu  Options  Help
 ─ ──────────────────────────── New Change Set ─────────────────────────
  │                                                    Work Item 8 created │
 C│                                                                       │
  │ Enter "*" in Work Item field to list available work items             │
 z│                                                                       │
  │ Work Item  . . 8_______                                               │
 C│ Comment  . . . _____________________________________________    +    │
  │                                                                       │
  │ Command - Enter "/" to select action                                  │
 C│ _  Create work item                                                   │
  │                                                                       │
  └───────────────────────────────────────────────────────────────────────┘
    EPSNBRVL            2011/10/24 06:28:54   SHARA30
******************************** Bottom of data ********************************
```

j) You will see that now you have pressed enter, the * in the SCM column has disappeared, showing that you have checked the code in. The next stage we need to perform is to build the program to make sure our change compiles.

8.5. Performing Builds

Before you can deliver your changes and make them available to the rest of the team you might want to compile and test your program. So let's explore the build options available from the ISPF Client.

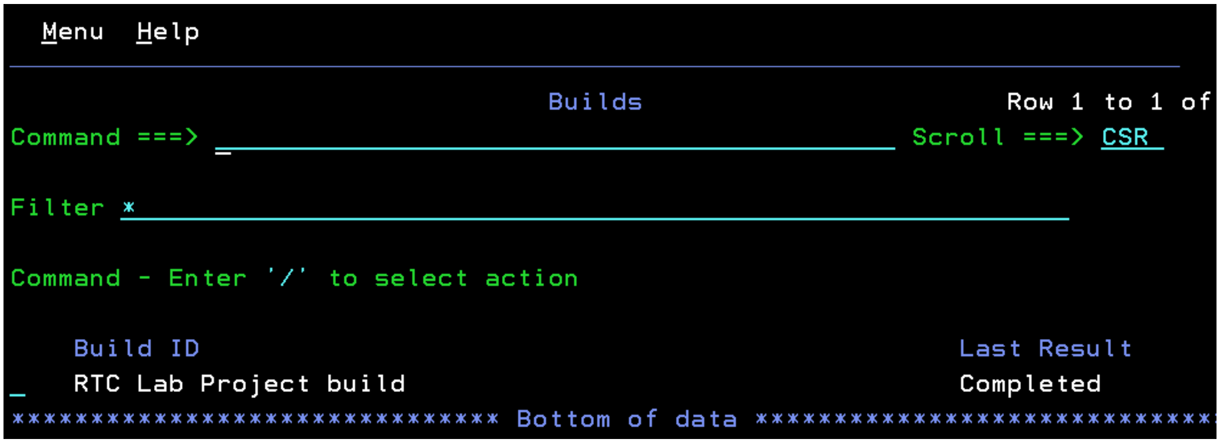
a) To get to the build options we could PF3 out until we went back to the RTC Primary option menu. But instead we are going to use the Action Bar choices on the RTC menus. Position your cursor on the Menu bar and press enter and the action bar menu should be displayed.



```
  Menu  Options  Help

 ┌──────────────────────────┐ ─────────────────────────────────────────
 │ _   1. RTC Settings      │ urce Control                 Row 1 to 6 of 6
 │     2. Connections       │                          Scroll ===> PAGE
 │     3. Repository Workspaces
 │     *. Edit              │ OBOL
 │     5. Build             │
 │     6. Exit              │ tion
 └──────────────────────────┘
     Name      SCM  Lock  Changed              ID
     EPSCMORT             2012/08/01 04:29:19  SHARA30
     EPSCSMRD             2011/10/24 06:28:52  SHARA30
     EPSCSMRT             2011/10/24 06:28:54  SHARA30
     EPSMLIST             2011/10/24 06:28:54  SHARA30
     EPSMPMT              2011/10/24 06:28:52  SHARA30
     EPSNBRVL             2011/10/24 06:28:54  SHARA30
******************************* Bottom of data ********************************
```

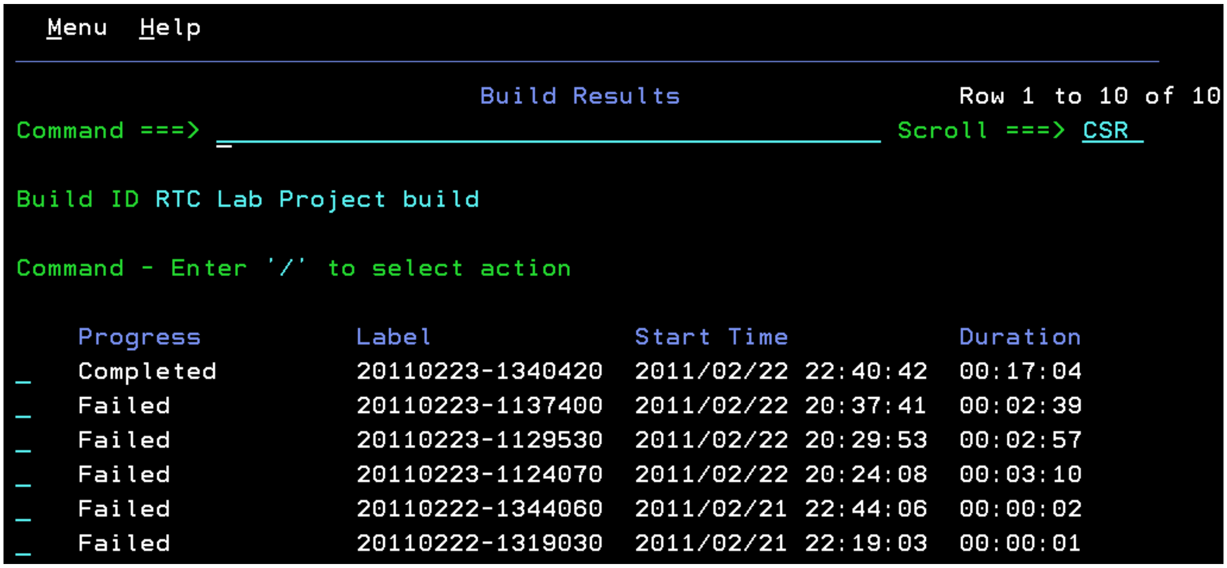
From here we are going to select option **5** for Build and press enter.

b) The Builds panel should be displayed with a list of the different build definitions. You may need to enter an * in the filter and then press enter. There should only be one build definition for this Lab and it should show that last status of the build. In the example below, that is completed.



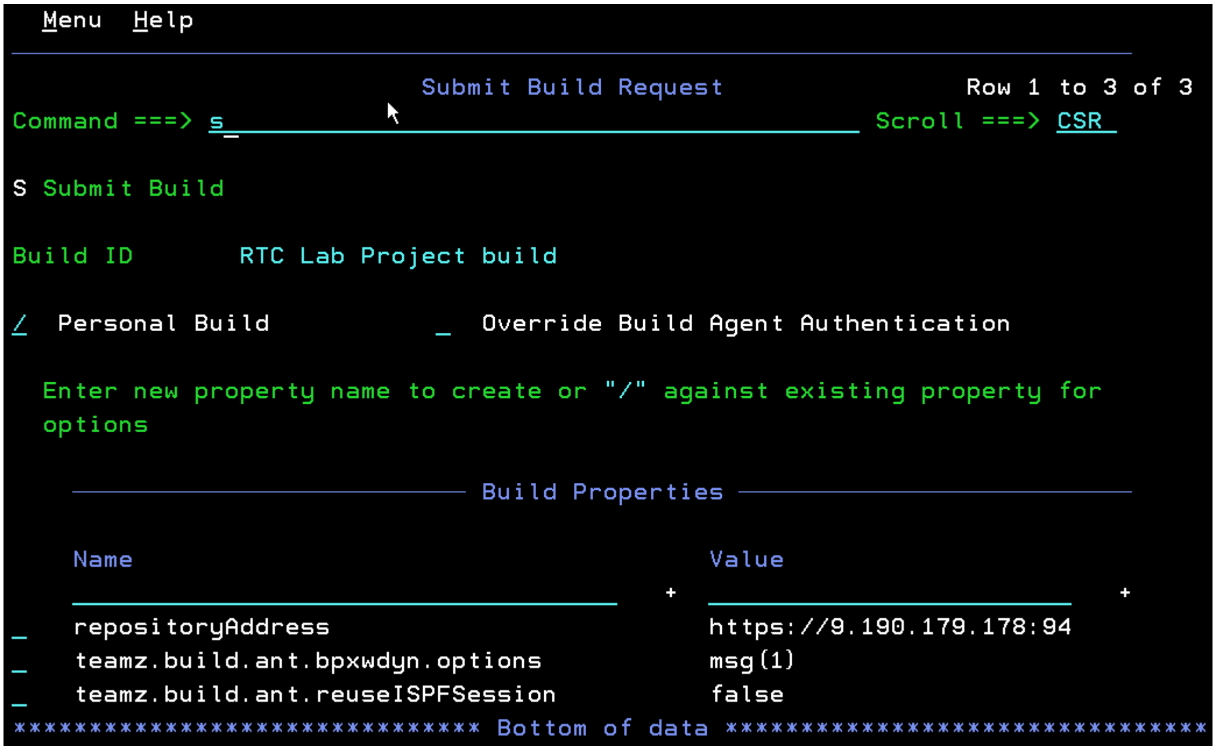
```
  Menu  Help
 ──────────────────────────────────────────────────────────────────────────
                                 Builds                      Row 1 to 1 of
Command ===> _                                           Scroll ===> CSR

Filter *

Command - Enter '/' to select action

    Build ID                                               Last Result
_   RTC Lab Project build                                  Completed
******************************* Bottom of data ********************************
```

c) To view previous builds enter a **V**, or if you want to request a new build enter an **R**, or enter a "/" to see available options. For this lab we will enter a V to view any previous builds. This will show us, much like the Eclipse GUI a list of all the previous builds and their current status.



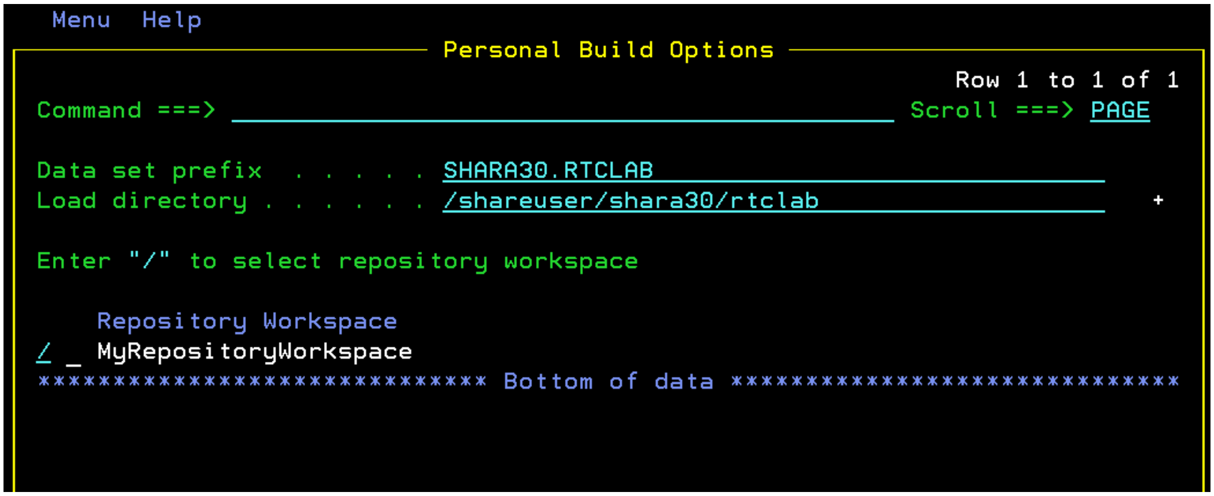
```
  Menu  Help
 ---------------------------------------------------------------------------
                              Build Results                  Row 1 to 10 of 10
 Command ===> _                                              Scroll ===> CSR

 Build ID RTC Lab Project build

 Command - Enter '/' to select action

     Progress          Label             Start Time           Duration
 _   Completed         20110223-1340420  2011/02/22 22:40:42  00:17:04
 _   Failed            20110223-1137400  2011/02/22 20:37:41  00:02:39
 _   Failed            20110223-1129530  2011/02/22 20:29:53  00:02:57
 _   Failed            20110223-1124070  2011/02/22 20:24:08  00:03:10
 _   Failed            20110222-1344060  2011/02/21 22:44:06  00:00:02
 _   Failed            20110222-1319030  2011/02/21 22:19:03  00:00:01
```

d) We now want to start a personal build to build the code changes we have made in our own personal repository workspace. So next to any one of the existing builds enter an **R** to request a build. This will display the Submit Build Request panel, which shows the build properties we will be using. However we want to submit a personal build so that it uses our own repository workspace and build data sets. So select the Personal Build option with a "/" and enter an S on the command line to submit the build.



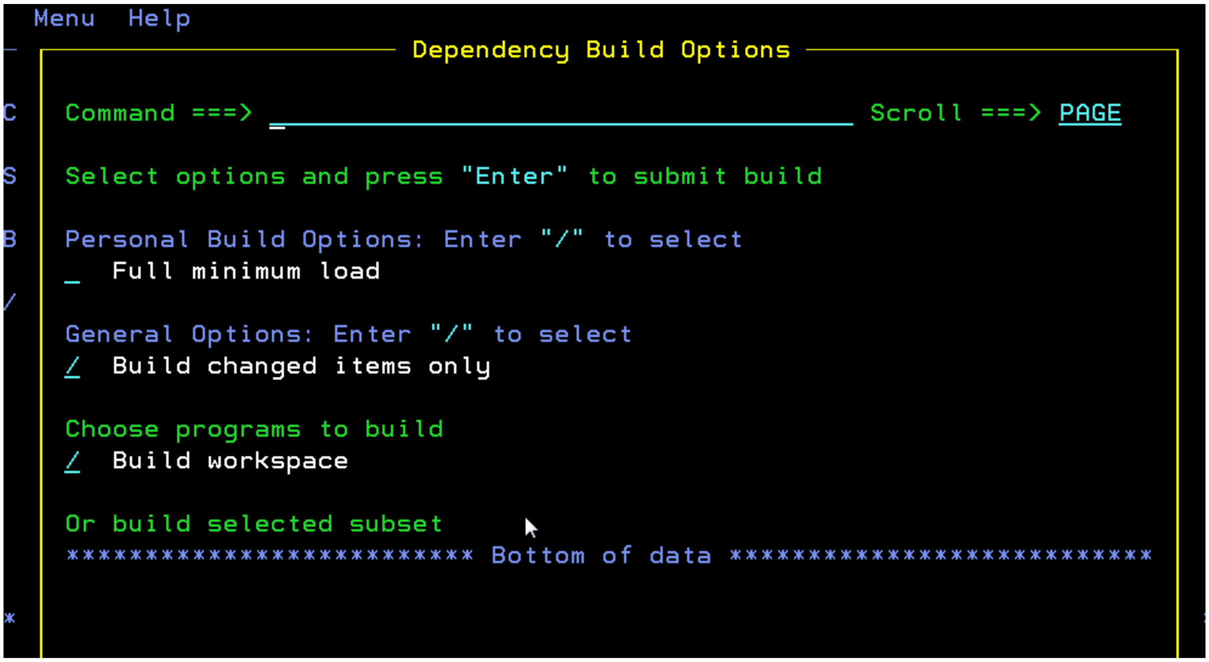
```
  Menu  Help
 --------------------------------------------------------------------------
                           Submit Build Request                Row 1 to 3 of 3
 Command ===> s_                                              Scroll ===> CSR

 S Submit Build

 Build ID       RTC Lab Project build

 / Personal Build              _ Override Build Agent Authentication

   Enter new property name to create or "/" against existing property for
   options

     ---------------------------- Build Properties ----------------------------

     Name                                       Value
     ____________________________________ +     ________________________ +
 _   repositoryAddress                          https://9.190.179.178:94
 _   teamz.build.ant.bpxwdyn.options            msg(1)
 _   teamz.build.ant.reuseISPFSession           false
******************************* Bottom of data ********************************
```

e) On the personal build options screen you need to select the repository workspace to use with a “/”. You also need to enter your data set prefix and load directory. For the data set prefix you can use the same settings as you did for the load, so **SHARAxx.RTCLAB**. For the load directory we need to specify a work location that RTC will use for the build xml files. Use /**shareuser/sharaxx/rtclab** where **sharaxx** is your SHARE LPAR userid.



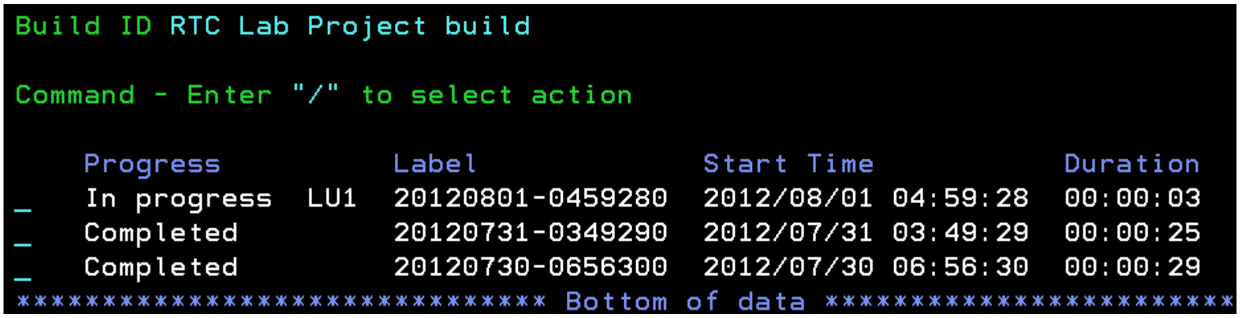
```
   Menu  Help
 ------------------------- Personal Build Options ---------------------------
                                                           Row 1 to 1 of 1
  Command ===> ____________________________________________ Scroll ===> PAGE

  Data set prefix  . . . . . SHARA30.RTCLAB____________________________
  Load directory . . . . . . /shareuser/shara30/rtclab_________________   +

  Enter "/" to select repository workspace

      Repository Workspace
  / _ MyRepositoryWorkspace
  ****************************** Bottom of data ******************************
```

Press Enter and you may be presented with the Dependency Build options screen. In this case make sure Build all programs is set with a “/”.



```
   Menu  Help
 -  ----------------------- Dependency Build Options -------------------------
 C
 C   Command ===> _____________________________________ Scroll ===> PAGE
 
 S   Select options and press "Enter" to submit build

 B   Personal Build Options: Enter "/" to select
     _  Full minimum load
 /
     General Options: Enter "/" to select
     /  Build changed items only

     Choose programs to build
     /  Build workspace

     Or build selected subset
     *************************** Bottom of data ***************************

 *
```

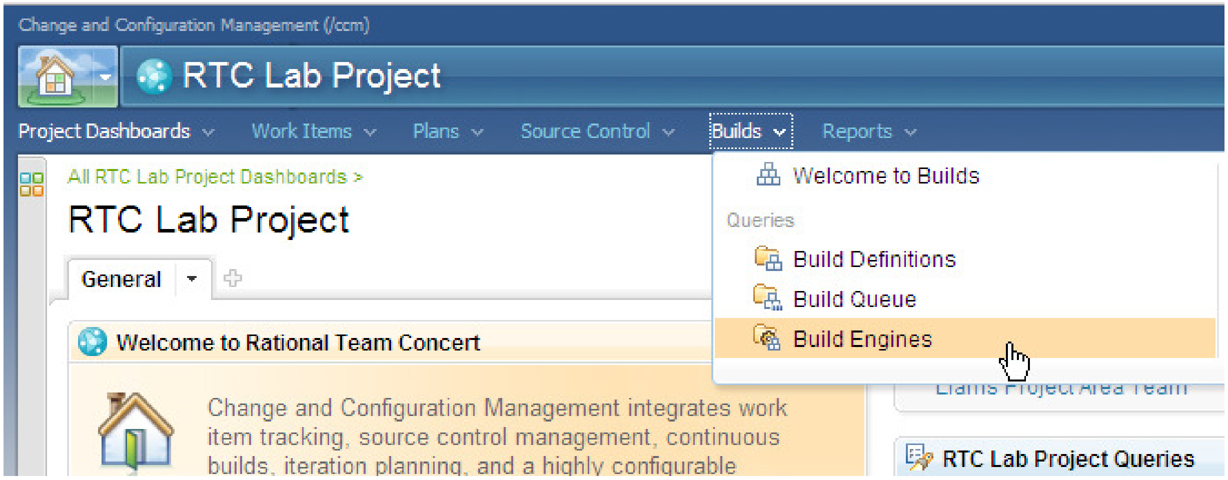
Press Enter and you will be returned to the Build results screen, where you will see the build is **pending**. If you press Enter you will see the status change to **In Progress**. The build may take some time, but should only build your 1 changed program.



```
Build ID RTC Lab Project build

Command - Enter "/" to select action

    Progress            Label              Start Time           Duration
_   In progress   LU1   20120801-0459280   2012/08/01 04:59:28  00:00:03
_   Completed           20120731-0349290   2012/07/31 03:49:29  00:00:25
_   Completed           20120730-0656300   2012/07/30 06:56:30  00:00:29
******************************* Bottom of data ************************
```

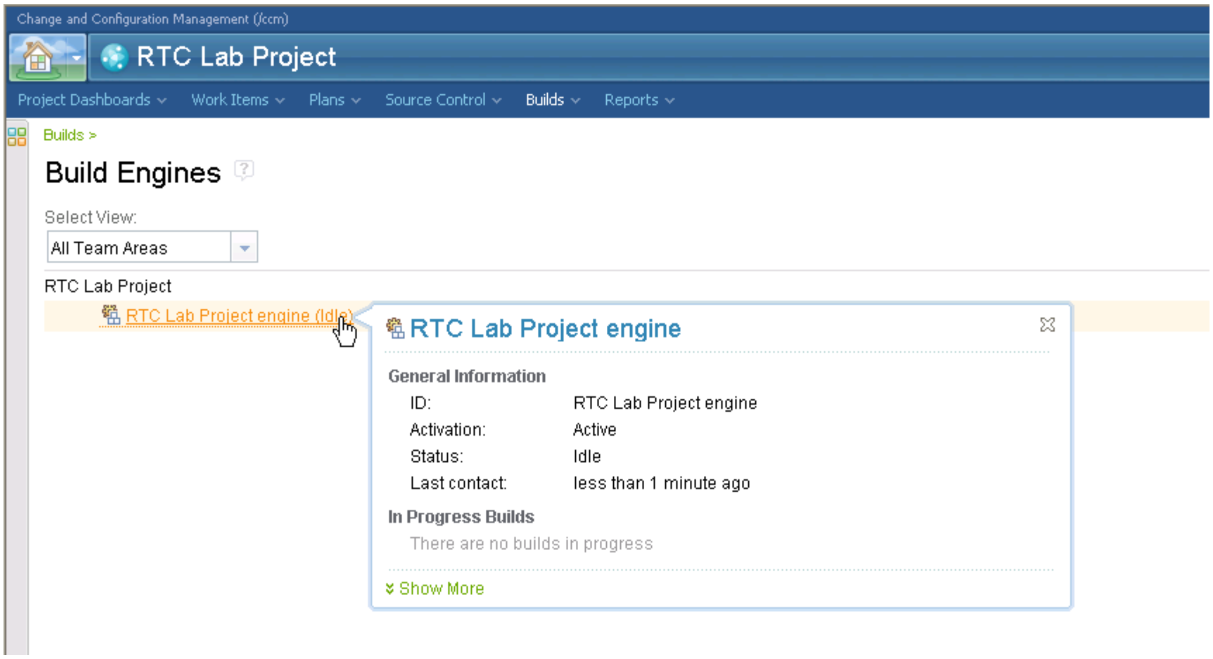
Note: There is a chance the build may have built the BMS Maps for your project as well.

f) If you want to check your build results you can do so through the Eclipse GUI or through the Web UI. As the Eclipse GUI is covered in Modules 6 and 7, we will check the build output via the Web UI. In a web browser use the URL **https://localhost:9443/ccm/web/projects/RTC Lab Project**. You my nee to login again using your userid or labuser99. On the main dashboard screen you will see the builds tab along the menu bar. Select that and then select the Build Engines option.



Change and Configuration Management integrates work item tracking, source control management, continuous builds, iteration planning, and a highly configurable

g) On the next screen, click on the link for the RTC Build engine.



Build Engines

Select View: All Team Areas

RTC Lab Project

RTC Lab Project engine (Idle)

RTC Lab Project engine

General Information

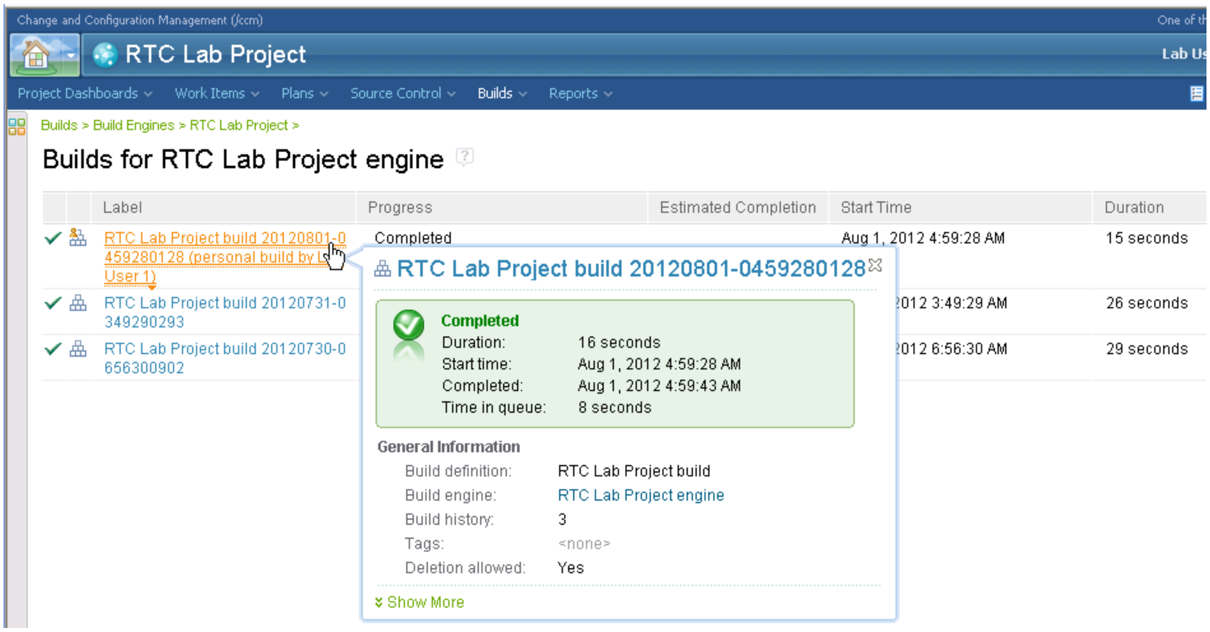
ID:	RTC Lab Project engine
Activation:	Active
Status:	Idle
Last contact:	less than 1 minute ago

In Progress Builds

There are no builds in progress

Show More

h) You will then be presented with a list of build results, similar to that shown in the ISPF Client. You should see that the last one is the personal build that you submitted from your ISPF session. If you hover your mouse over the link you will get popup giving you some details about the build.



Builds for RTC Lab Project engine

Label	Progress	Estimated Completion	Start Time	Duration
✓ RTC Lab Project build 20120801-0459280128 (personal build by L User 1)	Completed		Aug 1, 2012 4:59:28 AM	15 seconds
✓ RTC Lab Project build 20120731-0349290293			2012 3:49:29 AM	26 seconds
✓ RTC Lab Project build 20120730-0656300902			2012 6:56:30 AM	29 seconds

RTC Lab Project build 20120801-0459280128

Completed

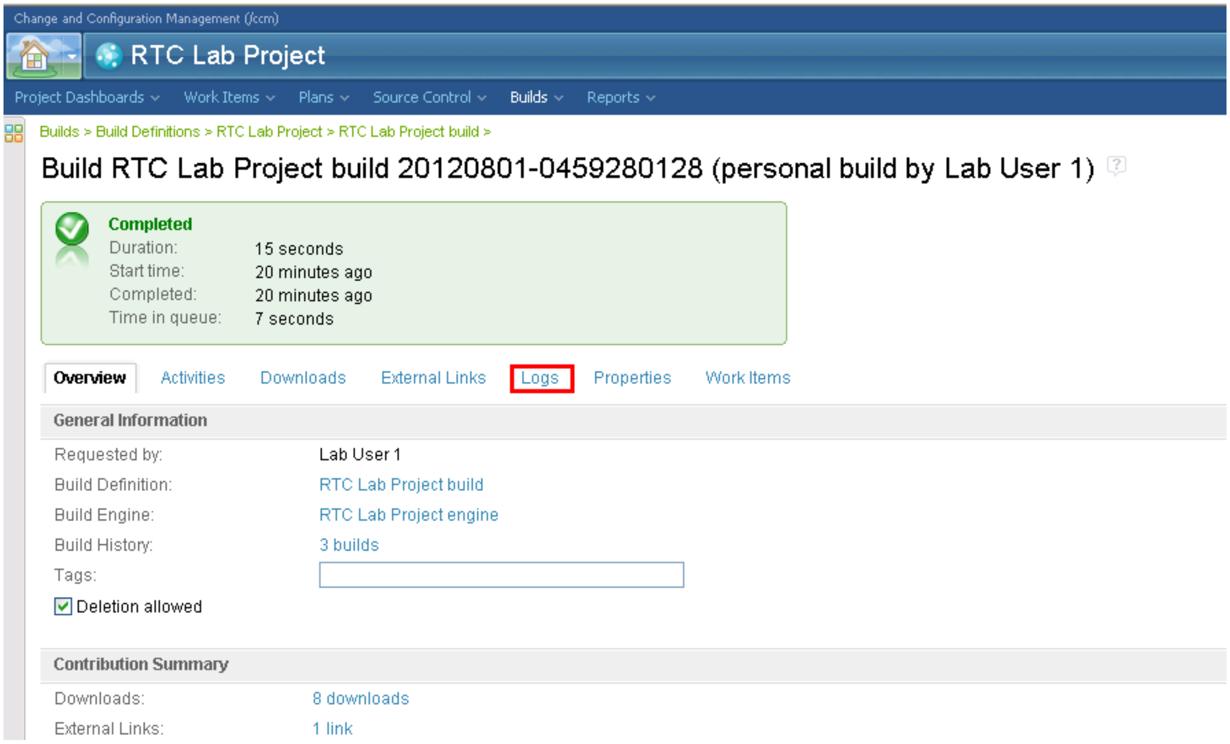
Duration:	16 seconds
Start time:	Aug 1, 2012 4:59:28 AM
Completed:	Aug 1, 2012 4:59:43 AM
Time in queue:	8 seconds

General Information

Build definition:	RTC Lab Project build
Build engine:	RTC Lab Project engine
Build history:	3
Tags:	<none>
Deletion allowed:	Yes

Show More

i) You can then select the link to take you into the build results



Change and Configuration Management (/ccm)
RTC Lab Project
Project Dashboards
Work Items
Plans
Source Control
Builds
Reports
Builds > Build Definitions > RTC Lab Project > RTC Lab Project build >

Build RTC Lab Project build 20120801-0459280128 (personal build by Lab User 1)

Completed	
Duration:	15 seconds
Start time:	20 minutes ago
Completed:	20 minutes ago
Time in queue:	7 seconds

General Information

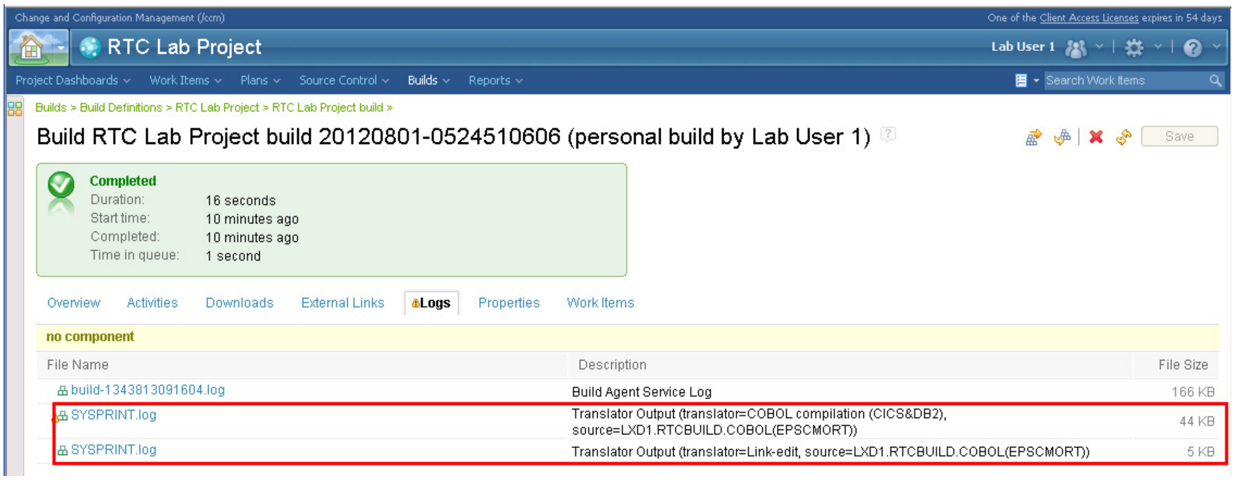
Requested by:	Lab User 1
Build Definition:	RTC Lab Project build
Build Engine:	RTC Lab Project engine
Build History:	3 builds
Tags:	

☑ Deletion allowed

Contribution Summary

Downloads:	8 downloads
External Links:	1 link

From here you can click on the tabs to show you various things. The one we are interested in is the Logs tab. Once you click that you will see all the compiler outputs. In this example we also have the BMS output, but we are interested in the SYSPRINT output from the compile of EPSCMORT, that we changed previously.



Change and Configuration Management (/ccm)
One of the Client Access Licenses expires in 54 days
RTC Lab Project
Lab User 1
Project Dashboards
Work Items
Plans
Source Control
Builds
Reports
Search Work Items
Builds > Build Definitions > RTC Lab Project > RTC Lab Project build >
Save

Build RTC Lab Project build 20120801-0524510606 (personal build by Lab User 1)

Completed	
Duration:	16 seconds
Start time:	10 minutes ago
Completed:	10 minutes ago
Time in queue:	1 second

no component

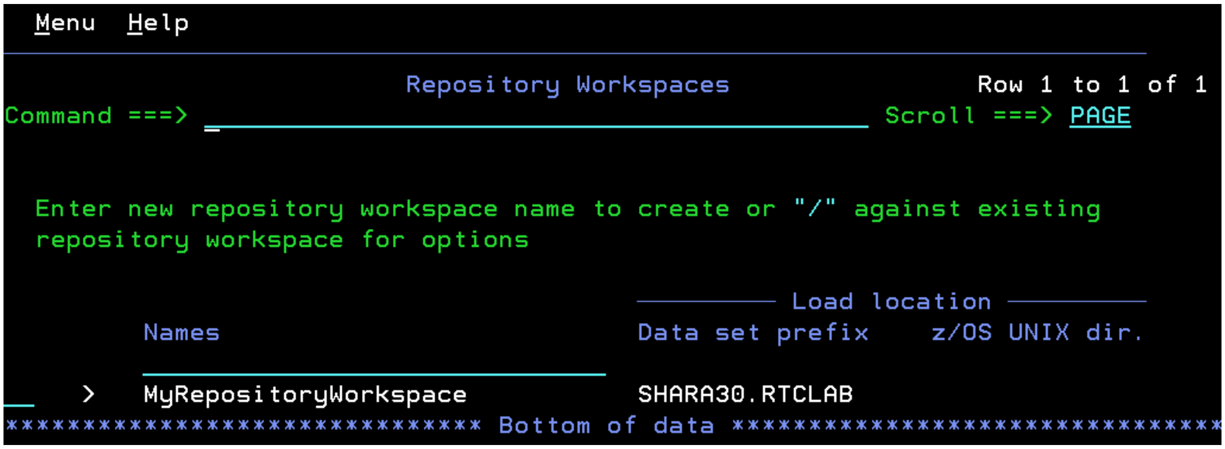
File Name	Description	File Size
build-1343813091604.log	Build Agent Service Log	166 KB
SYSPRINT.log	Translator Output (translator=COBOL compilation (CICS&DB2), source=LXD1.RTCBUILD.COBOL(EPSCMORT))	44 KB
SYSPRINT.log	Translator Output (translator=Link-edit, source=LXD1.RTCBUILD.COBOL(EPSCMORT))	5 KB

Click on the link for the COBOL compile and the system should download the file. You should now be able to view the file with a text editor such as Notepad.

8.6. Delivering your changes to the stream

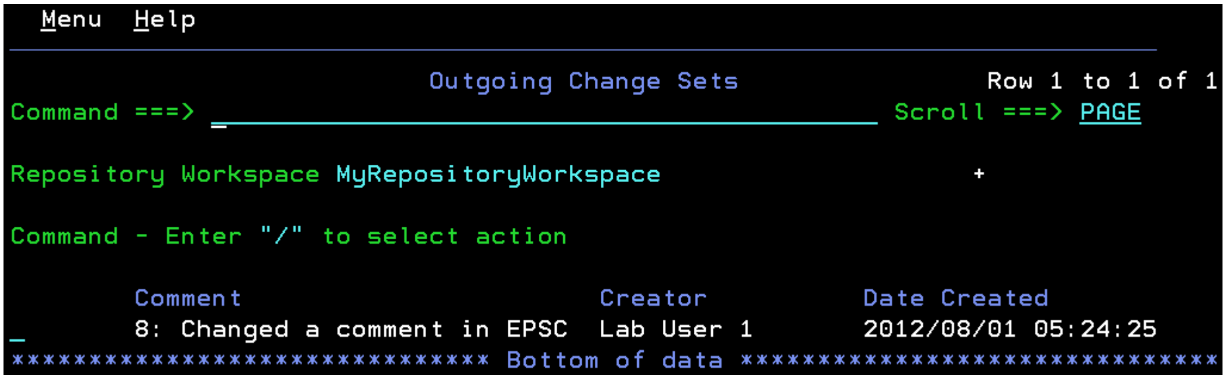
Now that you have successfully made a change to your programs, you are ready to deliver those changes to the projects stream so that you can share them with the rest of your team. This is a simple process.

a) Either from the RTC main menu, select option 2 – Workspaces or select the Menu action bar choice, press enter and 3 – Repository Workspaces. The repository workspaces panel will be displayed, and you will notice in the prefix area a **>** symbol. This shows we have outgoing changes that we can deliver. If the was a **<** symbol, then this would signify we had incoming changes to accept.



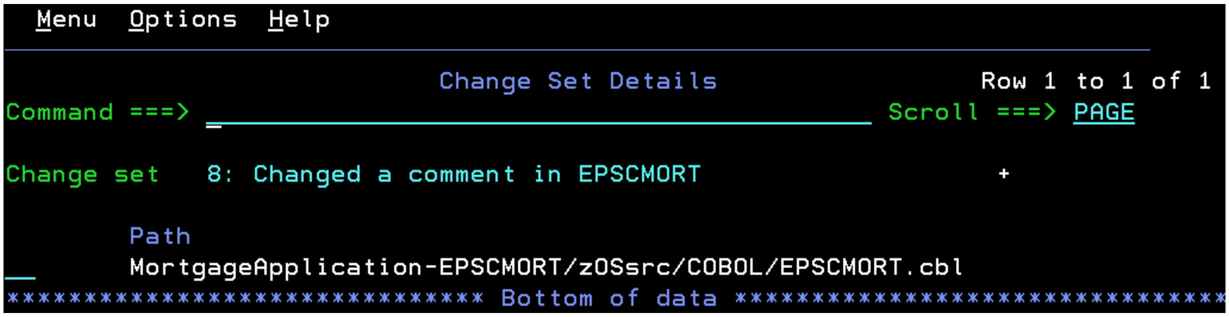
```
 Menu  Help
 ──────────────────────────────────────────────────────────────────────────
                          Repository Workspaces                 Row 1 to 1 of 1
Command ===> _____________________________________________  Scroll ===> PAGE


  Enter new repository workspace name to create or "/" against existing
  repository workspace for options

                                        ───────── Load location ─────────
       Names                            Data set prefix    z/OS UNIX dir.
       ──────────────────────────────
__   >   MyRepositoryWorkspace          SHARA30.RTCLAB
******************************* Bottom of data *******************************
```

b) You can use the “/” to bring up a list of available options, or in this case just enter an O to bring up the outgoing changes panel.



```
 Menu  Help
 ──────────────────────────────────────────────────────────────────────────
                          Outgoing Change Sets                 Row 1 to 1 of 1
Command ===> _____________________________________________  Scroll ===> PAGE

Repository Workspace MyRepositoryWorkspace                          +

Command - Enter "/" to select action

        Comment                       Creator          Date Created
_       8: Changed a comment in EPSC  Lab User 1       2012/08/01 05:24:25
******************************* Bottom of data *******************************
```

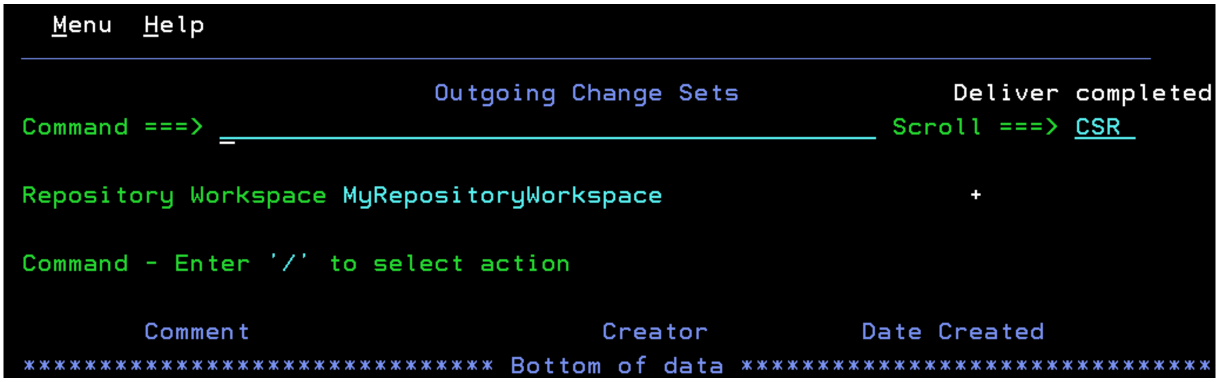
c) You will see the changeset, and the workitem you attached to it listed. If you wish to see what files are being delivered you can enter a **V** to list the files.



```
 Menu  Options  Help
 ──────────────────────────────────────────────────────────────────────────
                          Change Set Details                   Row 1 to 1 of 1
Command ===> _____________________________________________  Scroll ===> PAGE

Change set   8: Changed a comment in EPSCMORT                       +

        Path
__      MortgageApplication-EPSCMORT/zOSsrc/COBOL/EPSCMORT.cbl
******************************* Bottom of data *******************************
```

From here you can then enter a “/” for available options such as browsing the local or remote files, or comparing your local file with what is on the server.

d) Press PF3 to go back to the Outgoing Change Sets panel. Now enter a **D** to deliver the change to the stream. Once complete you will get a confirmation message.



```
  Menu  Help
 ──────────────────────────────────────────────────────────────────────────
                          Outgoing Change Sets                Deliver completed
Command ===> ________________________________________________ Scroll ===> CSR

Repository Workspace MyRepositoryWorkspace                        +

Command - Enter '/' to select action

        Comment                        Creator          Date Created
******************************* Bottom of data *******************************
```

e) Your changes have now been delivered to the project stream, so any team mates who have this component in their repository workspaces will now be able to accept your changes and merge them with their own.

Conclusion

This concludes Module 8. You now should be familiar with source code management using the RTC ISPF Client.

Module 9 Optional: Exploring Changes and Traceability

In this module, you will explore a work item to demonstrate how information is linked within Jazz. There are some basic questions that this lab will answer:

- What changes were made for a work item?
- Who changed this file, and why?
- What are the specific changes made on this resource?
- How can we visualize the change history for this resource?

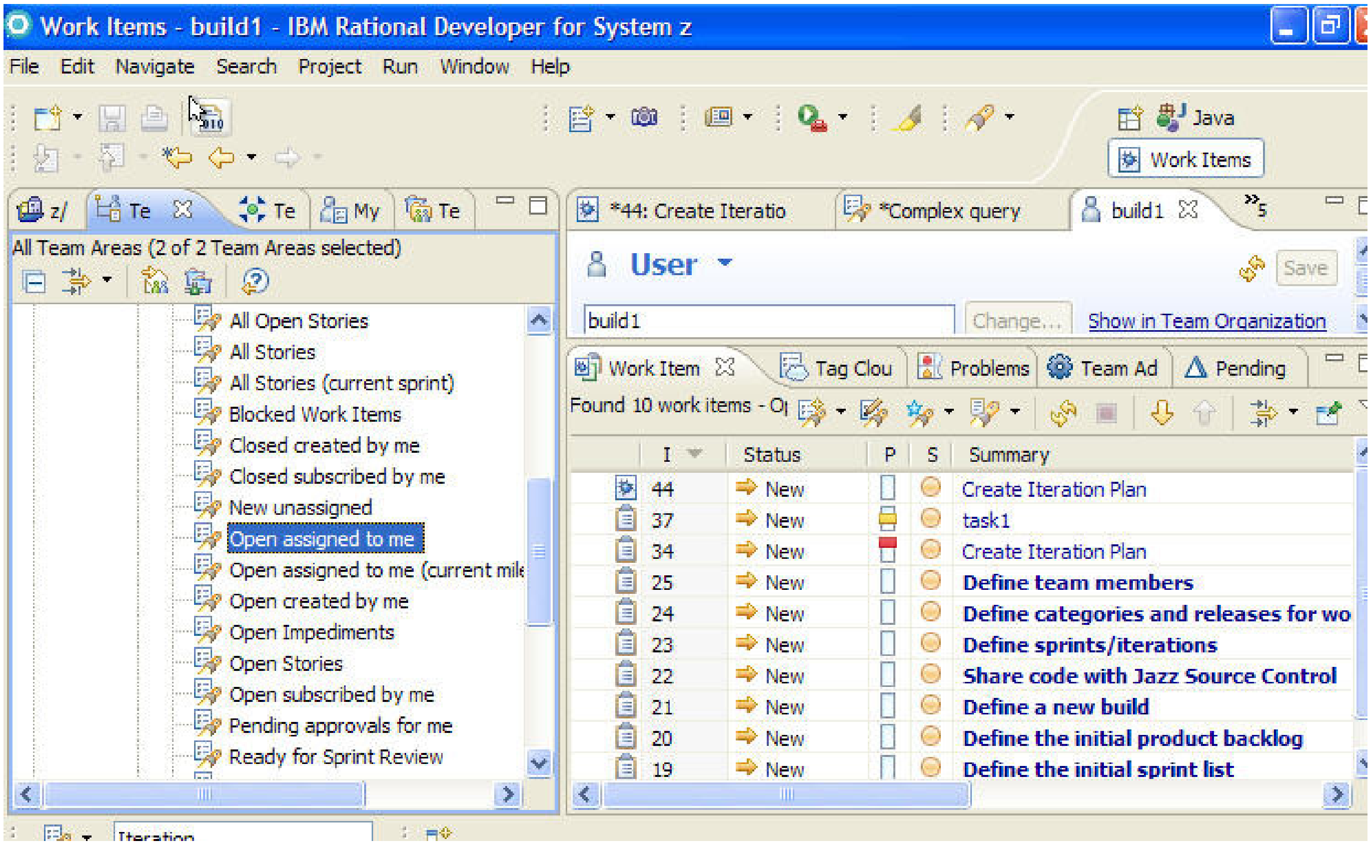
Lab Scenario

This is a good time to explore how Jazz links software artifacts in useful ways. You will now explore a part of the integration of Jazz Source Control Management with the rest of Jazz. You will also explore some of what can be done with change history.

9.1. What changes were made for a Work Item

Now you will explore the changes for a specific work item. This will allow you to see the specific files that changed. This makes the review process for the change easier.

a) Open a work item. Run a predefined query to search for **Open Assigned To Me** work items (See Section 3.1 for help with predefined queries). You should see something like the example picture below in your **Work Items** view (Window -> Show View -> Work Items).


Work Items - build1 - IBM Rational Developer for System z
File Edit Navigate Search Project Run Window Help
Java
Work Items
z/
Te
Te
My
Te
*44: Create Iteratio
*Complex query
build1
User
Save
Change...
Show in Team Organization
Work Item
Tag Clou
Problems
Team Ad
Pending
Iteration

All Team Areas (2 of 2 Team Areas selected)

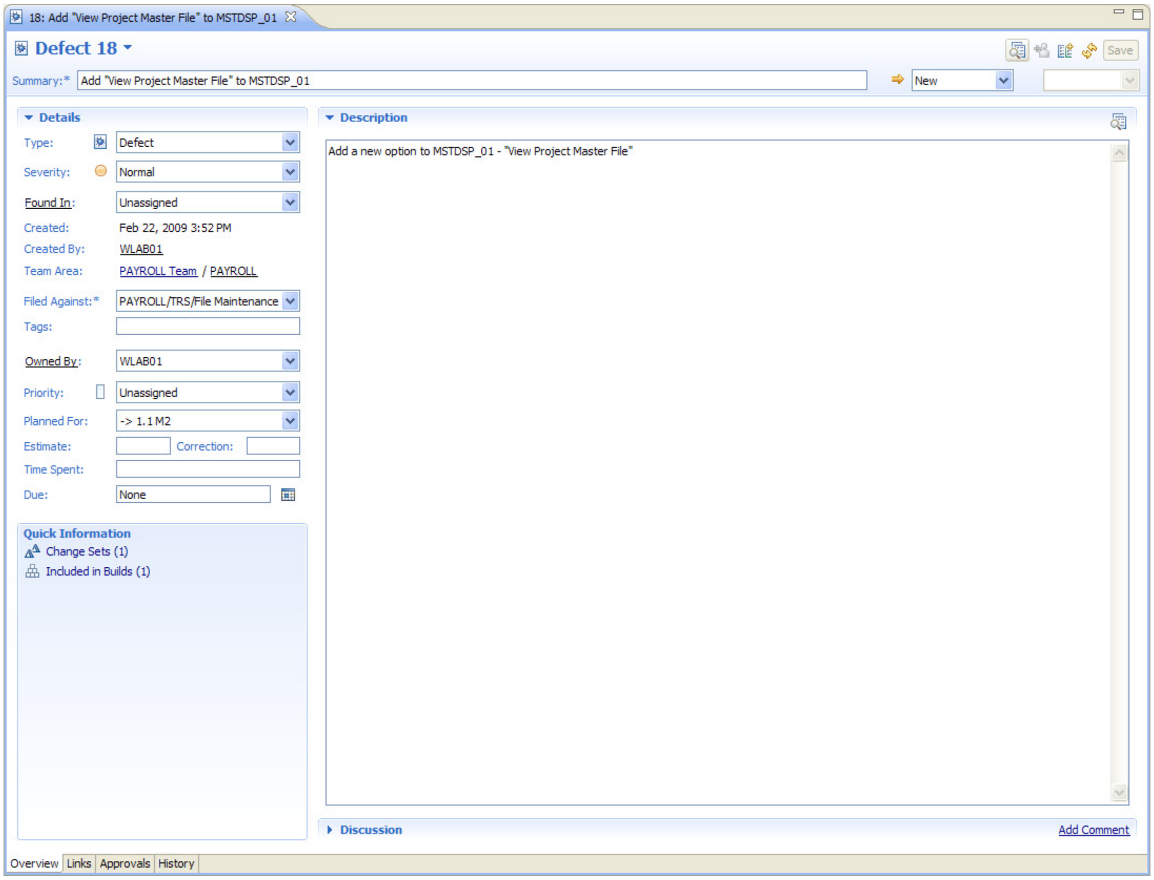
- All Open Stories
- All Stories
- All Stories (current sprint)
- Blocked Work Items
- Closed created by me
- Closed subscribed by me
- New unassigned
- Open assigned to me
- Open assigned to me (current mile
- Open created by me
- Open Impediments
- Open Stories
- Open subscribed by me
- Pending approvals for me
- Ready for Sprint Review

build1

Found 10 work items - O

I	Status	P	S	Summary
44	New			Create Iteration Plan
37	New			task1
34	New			Create Iteration Plan
25	New			Define team members
24	New			Define categories and releases for wo
23	New			Define sprints/iterations
22	New			Share code with Jazz Source Control
21	New			Define a new build
20	New			Define the initial product backlog
19	New			Define the initial sprint list

b) From the **Work Items** view, double-click a work item. The work item below is just an example. Choose any work item from the search results.



18: Add "View Project Master File" to MSTDSP_01

Defect 18

Summary:* Add "View Project Master File" to MSTDSP_01

New

Details

Type: Defect

Severity: Normal

Found In: Unassigned

Created: Feb 22, 2009 3:52 PM

Created By: WLAB01

Team Area: PAYROLL Team / PAYROLL

Filed Against:* PAYROLL/TRS/File Maintenance

Tags:

Owned By: WLAB01

Priority: Unassigned

Planned For: -> 1.1 M2

Estimate: Correction:

Time Spent:

Due: None

Quick Information

Change Sets (1)

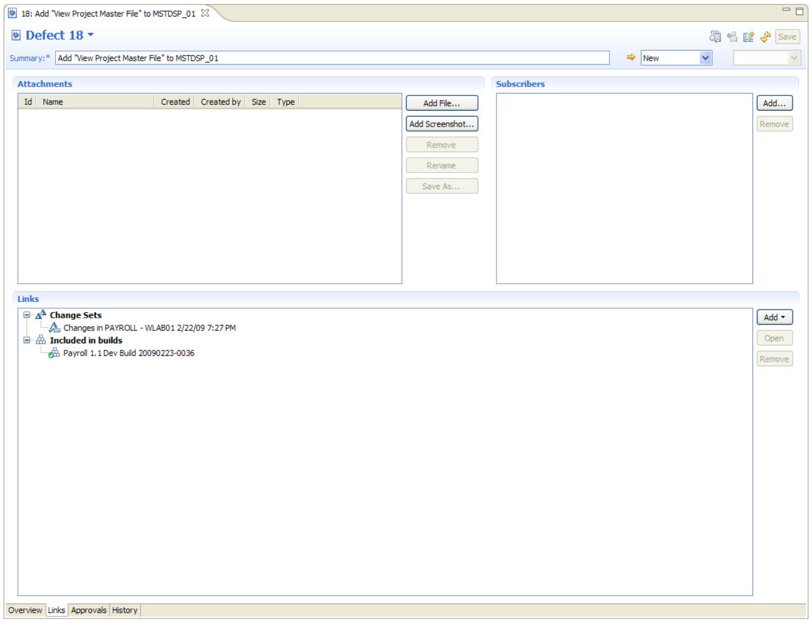
Included in Builds (1)

Description

Add a new option to MSTDSP_01 - "View Project Master File"

Discussion

c) Note the **Change-sets** link in the **Quick Information** section of the **Overview** tab and the detail in the **Links** tab. You will not see any change set information if a change set has not been associated with this work item (see Section 6 for information on how to associate a work item with a change set)



18: Add "View Project Master File" to MSTDSP_01

Defect 18

Summary:* Add "View Project Master File" to MSTDSP_01

New

Attachments

Id	Name	Created	Created by	Size	Type

Subscribers

Links

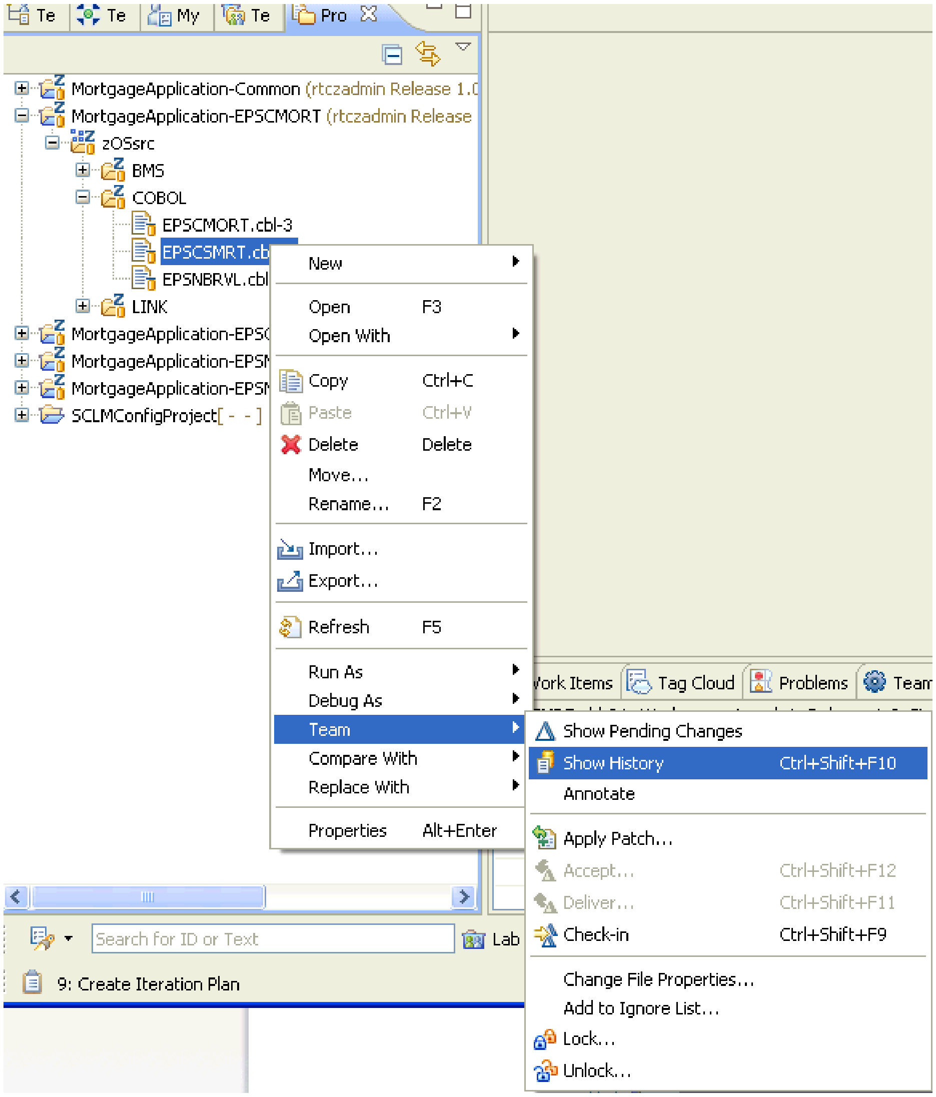
- **Change Sets**
  - Changes in PAYROLL - WLAB01 2/22/09 7:27 PM
- **Included in builds**
  - Payroll 1.1 Dev Build 20090223-0036

d) Note the work item's **Change Sets** section in the **Links** tab above. Double click a change set to open the change set and see which source members were modified for the change. In the **Change Explorer** view you can double click on a source member to open the compare editor and to see exactly which lines were modified for the change. This can be done for any work item. Try exploring changes made in a Work Item.

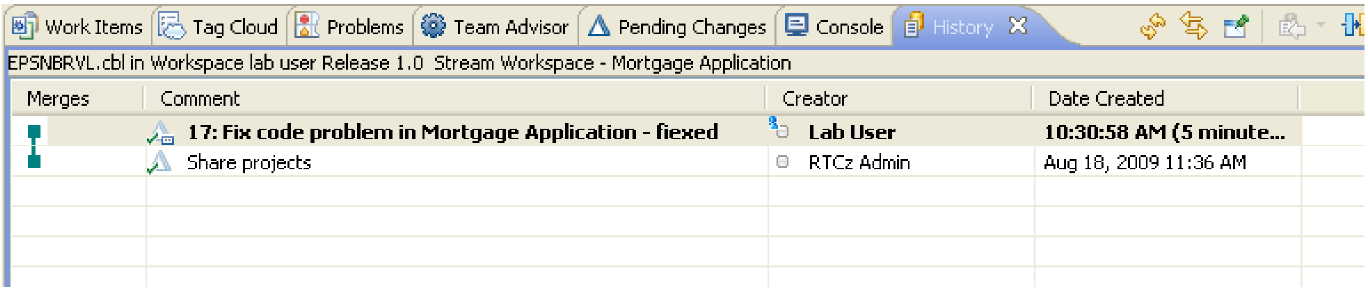
9.2. Who changed this file, when and why?

This section covers looking at the capabilities of finding out what changes happened with a specific file. We will use the **MortgageApplication** program loaded in Section 6.

a. From the **Package Explorer** view (**Window -> Show View -> Package Explorer**), select any of the files that you have modified right-click on the file and select **Team -> Show History**.



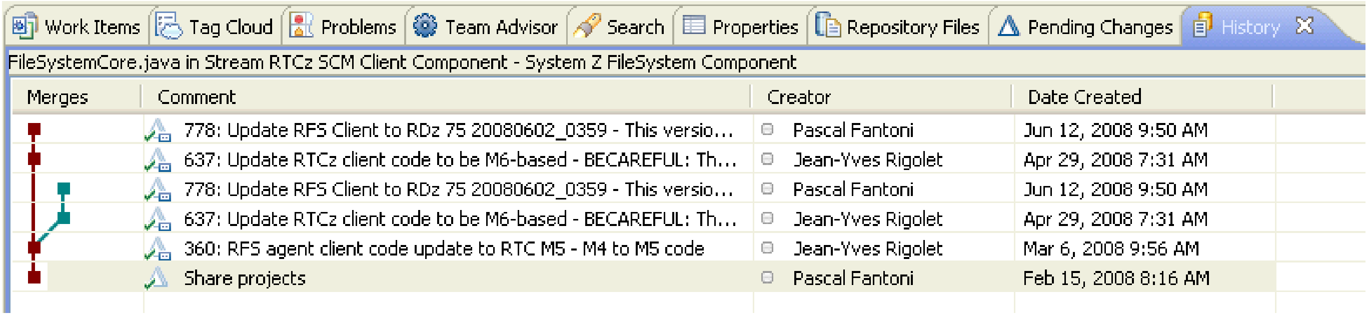
b. The **History** view opens and shows who has made changes to the file and when those changes were made. You can now start to understand how the file has evolved over time. This can also help with someone who is new to the project so they know who to talk to when they have questions over certain files. The **Merges** column gives a graphical view of the change-set merges that have occurred over time on this file.



EPSNBRVL.cbl in Workspace lab user Release 1.0 Stream Workspace - Mortgage Application

Merges	Comment	Creator	Date Created
	17: Fix code problem in Mortgage Application - fiexed	**Lab User**	**10:30:58 AM (5 minute...**
	Share projects	RTCz Admin	Aug 18, 2009 11:36 AM

c. Here is a second example of history for a part within the RTC source repository. It shows the different updates, who made them, and how the files were merged.



FileSystemCore.java in Stream RTCz SCM Client Component - System Z FileSystem Component

Merges	Comment	Creator	Date Created
	778: Update RFS Client to RDz 75 20080602_0359 - This versio...	Pascal Fantoni	Jun 12, 2008 9:50 AM
	637: Update RTCz client code to be M6-based - BECAREFUL: Th...	Jean-Yves Rigolet	Apr 29, 2008 7:31 AM
	778: Update RFS Client to RDz 75 20080602_0359 - This versio...	Pascal Fantoni	Jun 12, 2008 9:50 AM
	637: Update RTCz client code to be M6-based - BECAREFUL: Th...	Jean-Yves Rigolet	Apr 29, 2008 7:31 AM
	360: RFS agent client code update to RTC M5 - M4 to M5 code	Jean-Yves Rigolet	Mar 6, 2008 9:56 AM
	Share projects	Pascal Fantoni	Feb 15, 2008 8:16 AM

Conclusion

This concludes Module 9. You should now be familiar with the some of the traceability provided with Rational Team Concert. As you explore you will learn more of the history and traceability features.

Lab Conclusion

This lab has demonstrated the basic functionality of Rational Team Concert. Download your own copy from jazz.net and play all you want to learn more.

To use the sample code used in this lab go to:

https://jazz.net/wiki/bin/view/Main/DependencyBuildScenarioV4

or for the V3.0.1 version

https://jazz.net/wiki/bin/view/Main/DependencyBuildScenario

and for information on setting the build up go to :

https://jazz.net/wiki/bin/view/Main/DependencyBuildScenarioBuildEnvironmentSetup

Notices

U.S. Government Users Restricted Rights - Use, duplication or disclosure restrictedby 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 thisdocumentation in other countries. Consult your local IBM representative forinformation on the products and services currently available in your area. Anyreference to an IBM product, program, or service is not intended to state or implythat only that IBM product, program, or service may be used. Any functionallyequivalent product, program, or service that does not infringe any IBM intellectualproperty right may be used instead. However, it is the user's responsibility toevaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matterdescribed in this documentation. The furnishing of this documentation does notgive you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of LicensingIBM Corporation

North Castle Drive

Armonk, NY 10504-1785U.S.A.

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

IBM World Trade Asia CorporationLicensing

2-31 Roppongi 3-chome, Minato-kuTokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any othercountry where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THISPUBLICATION AS IS WITHOUT WARRANTY OF ANY KIND, EITHEREXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIEDWARRANTIES OR CONDITIONS OF NON-INFRINGEMENT,MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some statesdo 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 beincorporated in new editions of the publication. IBM may make improvementsand/or changes in the product(s) and/or the program(s) described in thispublication at any time without notice.

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

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

Intellectual Property Dept. for WebSphere SoftwareIBM Corporation

3600 Steeles Ave. East

Markham, Ontario

Canada L3R 9Z7

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 documentation and all licensed materialavailable for it are provided by IBM under terms of the IBM Customer Agreement,IBM International Program License Agreement or any equivalent agreementbetween us.

Any performance data contained herein was determined in a controlledenvironment. Therefore, the results obtained in other operating environments mayvary significantly. Some measurements may have been made on development-levelsystems and there is no guarantee that these measurements will be the same ongenerally available systems. Furthermore, some measurements may have beenestimated through extrapolation. Actual results may vary. Users of this documentshould verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change orwithdrawal without notice, and represent goals and objectives only.

Copyright license

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp.Sample Programs. © Copyright IBM Corp. 1992, 2007. All rights reserved

Trademarks and service marks

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

1. IBM
2. z/OS
3. System z
4. Rational

Intel® and Pentium® are trademarks of Intel Corporation in the United States, othercountries, or both.

Microsoft, Windows, Windows NT® and the Windows logo are trademarks ofMicrosoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in theUnited States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds in the United States, othercountries, or both.

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