In Part 1 of this series, learn how to join a team project, create a workspace to contribute to your project, and load existing artifacts from your team’s project components.

**View more content in this series**

A significant amount of collateral on IBM Rational Team Concert™ is readily available for Administrators, Build Architects, and Release Engineers. This article series is for the often overlooked, and sometimes even unloved Software Developer.

The Software Developer role is critical to constructing the project's software deliverables, services, and apps. Project Leads are increasingly looking at building, testing, and deploying the artifacts that developers create in an ever-compressing, highly agile, lean, DevOps life cycle.

Fortunately, Rational Team Concert enables these goals while fulfilling every developer's wish of developing software using a non-intrusive, lightweight, and easy to use change management solution.

This article series describes many operations a developer is responsible for from the time they join the project to the time where they are ready to deliver their features and fixes. This series is meant to be a helpful collection of "developer's cheat sheets." Some of the operations outlined are ones you might use infrequently. You'll use other operations often enough for them to become second nature to your normal development routines.

Part 1 of this article series explores:

- The main concepts behind Rational Team Concert's change management mechanisms
- Joining a team project
- Creating a repository workspace and loading your project components and artifacts
Before you start
Figure 1 outlines the deployment of the Jazz™ Team Server and Rational Team Concert used to run the steps in this article.

If you would like to follow the steps in this article, you will need a deployment topology similar to that shown in Figure 1. This topology makes use of two physical workstations, although it is also possible to use a single workstation.

Figure 1. Rational Team Concert and Jazz Team Server deployment topology

Workstation 1. The Jazz Team Server

Workstation 1 hosts:

- The Jazz Team Server and the Rational solution for Collaborative Lifecycle Management (CLM) 5.0.2. The Knowledge Center has detailed instructions for installing the Jazz Team Server and the Rational solution for CLM.
- The Money that Matters sample lifecycle application. You'll find Steps for installing the Money that Matters sample in the Knowledge center.

Important: The developer operations you will carry out will be under the Money That Matters sample user id bob. You or your Team Administrator will need to assign the fictional user Bob at JKE Banking a Rational Team Concert Developer client access license. You can find the steps to do so here.

Workstation 2. The Rational Team Concert development client

There are many options for installing Rational Team Concert on workstation 2. These options include, but are not limited to installing Rational Team Concert:

- As a standalone product
- Into an existing Eclipse environment
- Into an existing IBM development offering also based on Eclipse (e.g. IBM® Rational Application Developer for WebSphere Software)

For this article I chose the second option: An installment of Rational Team Concert into an existing Eclipse 4.4 (Luna) IDE.

Read Assigning client access licenses to users for detailed directions for the various client install options.

Team development in Rational Team Concert: The vocabulary

Rational Team Concert strives to conceal the mechanics of change management as much as possible. Doing so, allows you, the developer to focus on constructing, testing, and deploying high
quality code to your project. Yet, to optimize your productivity in a team environment you need to be familiar with some of Rational Team Concert's change management concepts.

**Figure 2. Relationship between Rational Team Concert storage and change management mechanisms**

In Rational Team Concert, a team repository is your team's collaborative lifecycle management (CLM) server. Depending on the capabilities installed, the CLM server can provide all or any of the following services and more to teams: change management, requirements management, and quality management. The team repository hosts project areas and user profiles for contributors into the various project areas.

Among several other things, a project area contains one or more streams for development. A stream is similar in concept to a branch in other change management systems. In essence, a stream is a logical unit for collecting changes made to artifacts contained in one or more components.

Some projects use a single stream to deliver all changes. This happens when all team members focus on delivering a single release on a sequential project schedule. This is becoming somewhat rare, particularly in agile projects where there is a need to deliver multiple releases and service updates concurrently. Read the article "Multiple Stream Development" to learn more about single vs. multiple stream development.

A component is logical collection of the software artifacts that you and your team will develop (i.e. create, update, and test). More specifically, a component contains your Eclipse projects, folders, and the artifacts under them (Java, JavaScript, C++, scripts, images, etc.).

**Join a team project**

To join your development project area and begin contributing you need to either accept a team invitation sent to you by email or create a repository connection to the Jazz team server that hosts your project area.

**Create a repository connection to join a project**

The Money that Matters project area and the team assigned to it is fictional. Log in to join and contribute to the project. Use the credentials:

User id: bob
Password: **bob**

In your real-world projects, you will receive an invitation email that contains clear steps on how to accept the invitation and quickly join the project using Rational Team Concert. You will not receive an email invitation to join the Money that Matters project because you are using the fictional user bob. Instead, you will directly create a repository connection to join your project area.

1. On your developer workstation, Start Rational Team Concert/Eclipse with a new workspace.
2. Open a command prompt window and type `<eclipse_install_dir>\eclipse –data <workspaceName>` e.g. `C:\eclipse\eclipse C:\myWorkspaces\MoneyThatMatters`
3. Open the **Work Items** perspective, **Windows > Open Perspective > Work Items**.
4. In the **Team Artifacts** view, click the **Create a Repository Connection** link.
5. In the Create a Jazz Repository Connection dialog, type the following as shown in Figure 3:
   - `https://<your_JTS_server_hostname>:<jts_port>/ccm`
   - Enter **bob** for both the **User ID** and **Password** fields and click the **Finish** button.

**Figure 3. The Create a Jazz Repository Connection dialog**

Manage connected project areas

After you create a connection to the Jazz repository, identify the project area that you are joining.
1. In the **Team Artifacts** view, expand the **Repository Connections** folder to see the connection you created to your team repository.
2. Right click your repository connection and select **Manage Connected Project Areas...** as shown in Figure 4.

**Figure 4. Invoke the Manage Connected Project Areas dialog**

![Image of workspace with Team Artifacts view expanded and Manage Connected Project Areas highlighted]

3. In the **Manage Connected Project Areas** dialog select the checkbox next to the **JKE Banking (Change Management)** project area, and click the **Finish** button.

Figure 5 shows a new expandable entry representing the JKE Banking project area that opens after you select the just connected project area. Expand the entry to see a list of the various project artifact types that you can now access. This includes your project's builds, plans, reports, and work items (i.e. change requests, defects, enhancements, etc.). Actually, you likely have more than just the ability to view these artifacts, your team administrator has probably granted you access to launch your own builds, draft or edit shared project plans, and update and create your own work items.
Create a repository workspace and load your project components and artifacts

If your role on the team is that of Project Manager, you can probably stop here. You can view, create and update project plans, reports, and work items. However, as a developer you still require access to the code!

Before you load code into your Eclipse workspace, you need to create a repository workspace. A repository workspace is analogous to your Eclipse workspace, except it is stored in your team repository instead of your local file system. A repository workspace contains the undelivered changes (or change sets) you make to shared team artifacts. Changes include new code artifacts you create as well as those that you update.

Create a new Repository Workspace

1. In the Team Artifacts view, right click the My Repository Workspaces folder and select New > Repository Workspace...
2. On the Select a Stream page of the New Repository Workspace wizard:
   • Select the Flow with a stream: radio button.
   • Expand the JKE Banking project area and select BRM Stream (Business Recovery Matters) and click Next >
3. Provide a name for Repository Workspace Name: or accept the default name and click Next >
4. In the Select repository page, leave the Use current repository radio button selected and click Next >
5. Select the Public radio button on the Read Access Permission page and click Next >
6. In the Components to Add page shown in Figure 6:
   • Leave all of the checkboxes for the list of available components selected.
   • Leave the Load repository workspace after creation checkbox selected.
• Click the **Finish** button.

**Figure 6. Adding components to a new repository workspace**

![New Repository Workspace](image)

**Load source code into your repository workspace**

After completing its task, the **New Repository Workspace** wizard opens the **Load Repository Workspace** wizard to load both your repository workspace and your local Eclipse workspace with the Eclipse projects and the artifacts belonging to those projects.

The Eclipse projects available to load are the ones associated to the components selected when you create your repository workspace (shown in Figure 6).

1. In the **Load Repository Workspace** page, select the **Find and load Eclipse projects** radio button and click **Next >**
2. In the **Load Eclipse Projects** page, leave all of the check boxes next to the list of Eclipse projects selected and click **Finish**.
3. Switch to the Java EE perspective, **Window > Open Perspective > Java EE**.

Figure 7 shows the various projects now loaded into your Eclipse-based IDE. You are ready to code and make valued team contributions.
Summary

In this article, you used Rational Team Concert, to connect your Eclipse-based IDE to a function-rich, Jazz CLM server.

This article briefly explained some of the fundamental mechanisms that Rational Team Concert uses to carry out change management. It also showed, step-by-step how to join a team project, create a workspace to contribute to your project, and load existing artifacts from your team's project components.

In part 2 of this series you will learn key operations to use frequently to make code contributions and to manage your work, with the focus remaining on your role as a developer in a team environment.

Acknowledgements

The author thanks Cheng-Yee Lin and Guillermo A. Hurtado for their thoughtful review of this article.
Related topics

• Visit the Rational Team Concert deployment Wiki.
• More about Rational Team Concert:
  • Find Rational Team Concert articles and links to many other resources on IBM developerWorks, and check the product overview page and the features and benefits.
  • Check the Rational Team Concert page on Jazz.net.
  • Watch the Using Rational Team Concert in a globally distributed team webcast or a demonstration of the Dashboards and reports, or listen to the podcast about IBM Rational Team Concert and Jazz.
• Download Rational Team Concert from Jazz.net and try it free on up to 10 developers for as long as you want (requires registration).
• Try IBM Bluemix a single solution environment with the instant resources and infrastructure you need to develop and deploy apps across multiple domains

© Copyright IBM Corporation 2015
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)