

Source Control with RTC

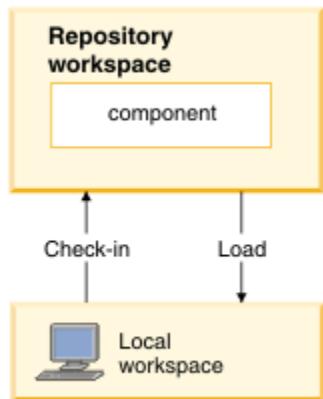
Workspaces

A workspace is an area where you can view or modify components. A repository workspace on the server stores components as versionable artifacts that can be loaded into a local workspace (such as an Eclipse workspace or an ordinary folder) on the client.

Repository workspaces are objects in the repository. Local workspaces are directories in your computer's file system.

In the repository, files and folders are stored as versionable items whose data and metadata can be viewed but not directly modified. To enable integration with file-based tools such as editors, compilers, and debuggers, files and folders in a repository workspace are loaded (copied) into a local workspace on your computer. As you make changes to files and folders in the local workspace, you periodically check them in, which copies the changes to the repository workspace. When all the changes in your local workspace are checked in, the repository and local workspaces have the same content.

The figure below illustrates a simple configuration of a repository workspace a single component. Check-in copies changes from the local workspace to the repository workspace. Load loads the local workspace with the contents of the repository workspace.



For more information about workspaces, see this link:

http://pic.dhe.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.team.scm.doc/topics/c_streams.html

Streams and components

Source control in RTC is organized using streams and components. A stream is a repository object that includes one or more components. A stream is a collection of one or more components.

A component is a collection of related artifacts, such as a website project or a group of source code that comprise a Windows application. Artifacts under source control are grouped into components. Any group of files and folders that share a common root can be a component.

A stream is similar to branches and trunks found in other source control systems. However, streams provide more functionality. A stream can include at most one version of any component. By using multiple streams, or different branches, a development organization can work on projects that use different versions of the same components. For example, a stream dedicated to the development of a new software release would include the most recent version of the release's components, while another stream dedicated to the maintenance of an earlier version of that software release would initially include the component versions that were part of that release.

A component is grouping of files with a common purpose. A web project can be configured as a component. Any component in a repository can be included in zero or more streams.

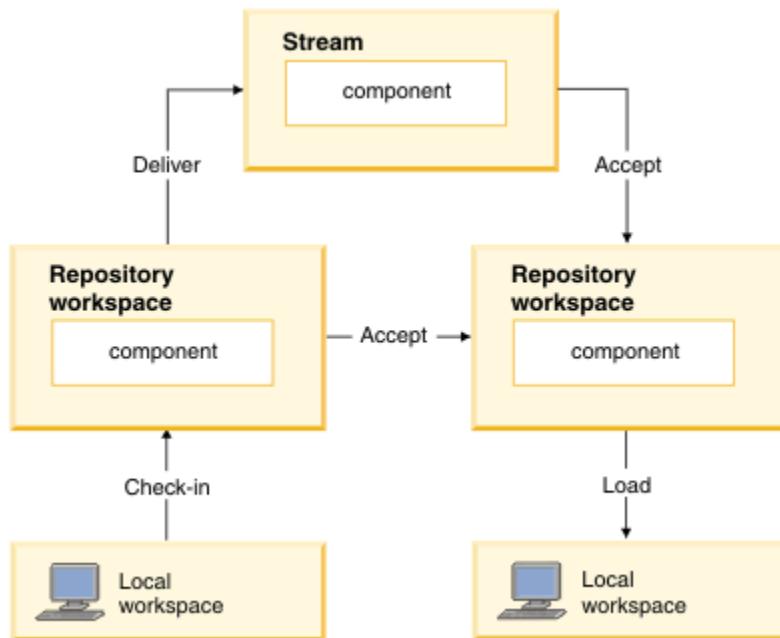
The separation of components to stream allows for the separation of business value and development value. The business prefers the stream aspect as it organizes work using versions and time-boxing. The developers prefer components as they visualize projects by their purpose, not by time or version. A developer sees one website project; the business sees a website with different sets of functionalities.

Because streams and components are repository abstracts that catalog workspace contents, a repository workspace is created to define the streams and components in a repository. Once you have created a repository workspace and added projects to it, (by checking them in) you can organize the projects into components.

Streams and components are not defined in a local workspace. The file and folder objects in each component in a repository workspace are loaded into the local file system as files and folders. Only one version of each file or folder (the version selected by the baseline or stream that you specified when creating the workspace) is loaded.

A local workspace can load more than one version of a component (this can happen if the local workspace includes the contents of more than one repository workspace), but only one version of a component can be present in a local workspace at a time. Loading the component from one repository workspace overwrites contents that were loaded from a different repository workspace, after prompting you to allow the overwrite.

Here is a visual description for the relationship among streams, components, and workspaces.



Code checked-in in one local workspace is uploaded to its responding repository workspace. From here, another repository workspace can accept changes from the first repository workspace and merge its changes onto itself. This way the changes are local and not all repository workspaces can see the changes. Or you can deliver changes in the repository workspace into the stream, making it available to all repository workspaces of that stream.

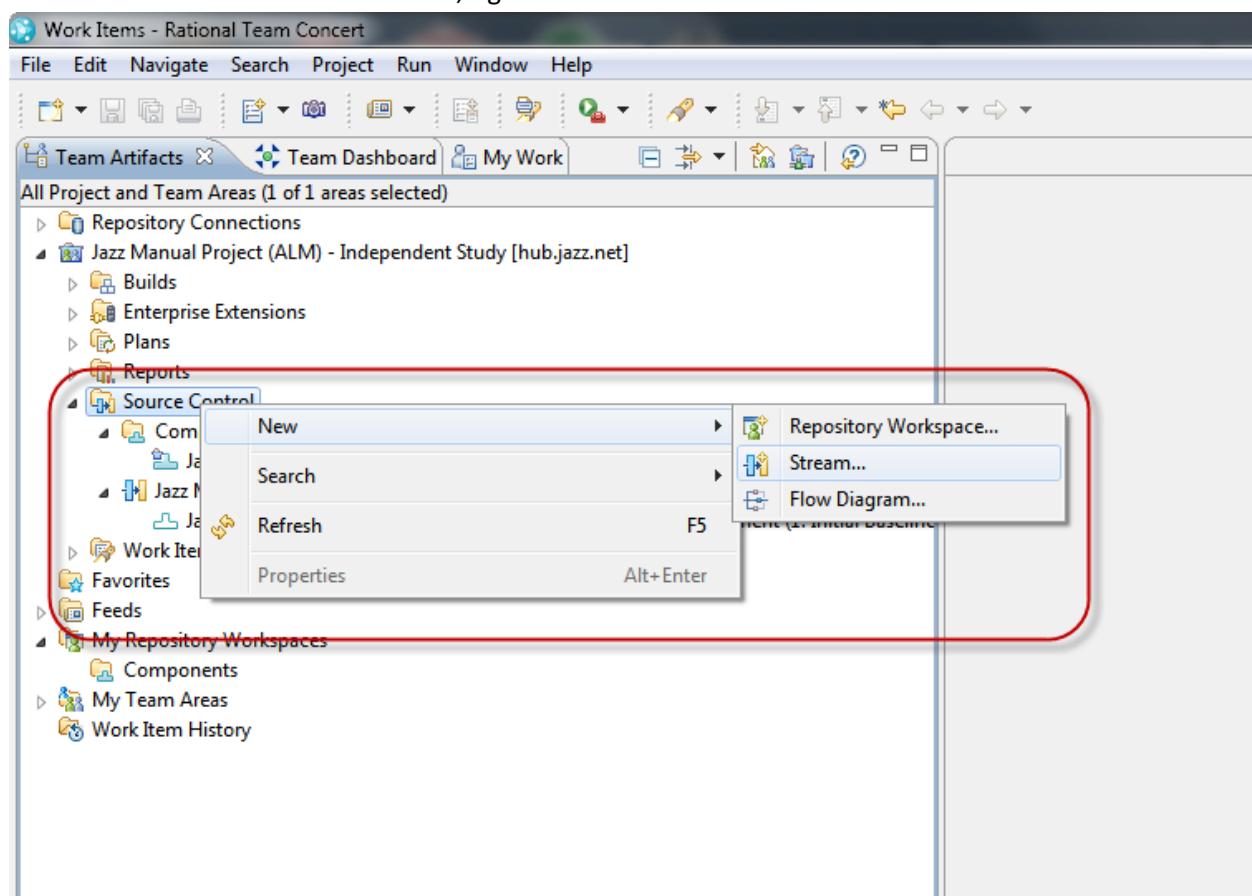
To read more, see this link:

http://pic.dhe.ibm.com/infocenter/rtc/v2r0m0/index.jsp?topic=/com.ibm.team.scm.doc/topics/c_streams.html

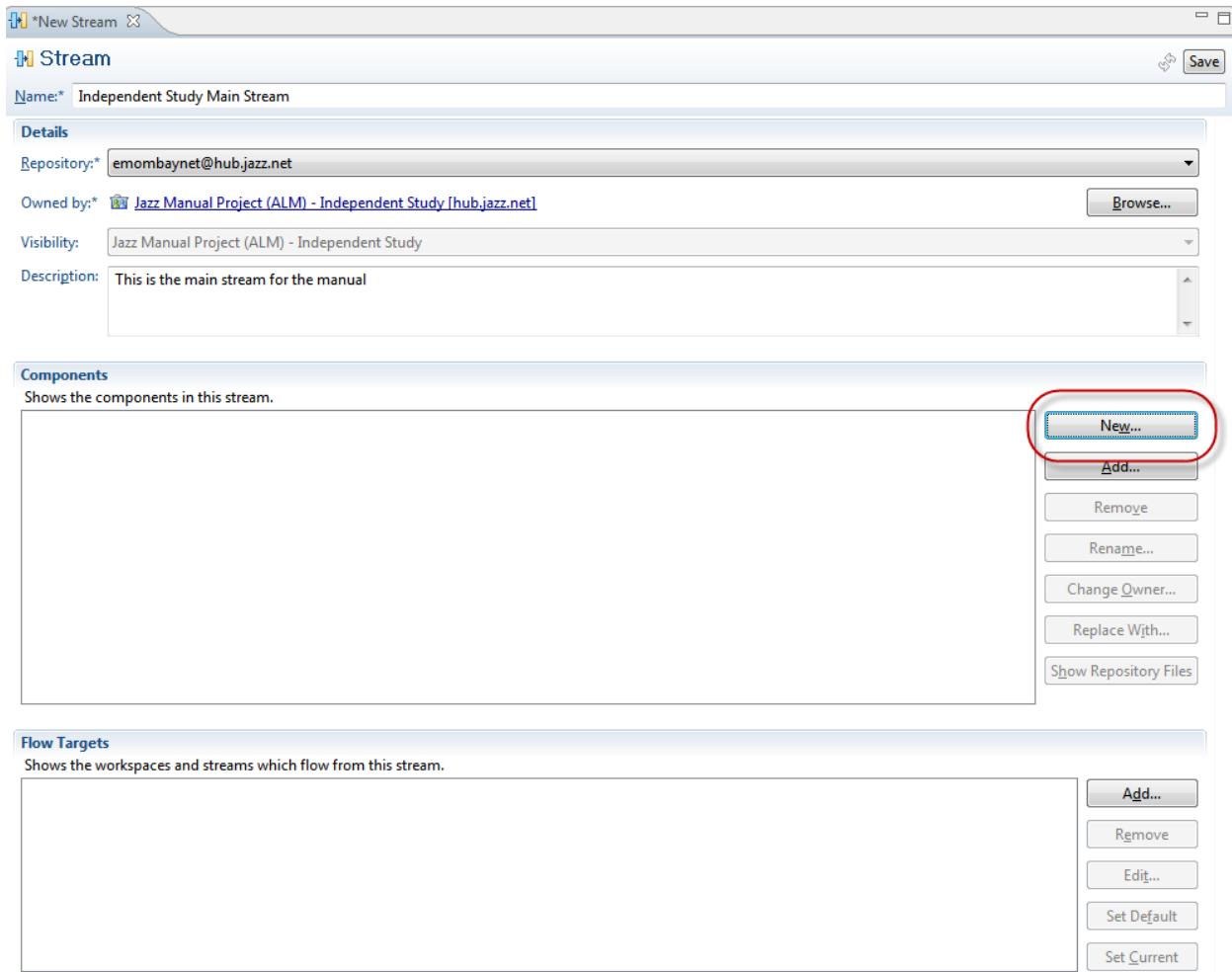
Create a new stream and component

1. Start the **Rational Team Concert** application.
2. Make sure you are logged in to a **Repository Connection**. If not, log in using your Jazz.Net username and password.

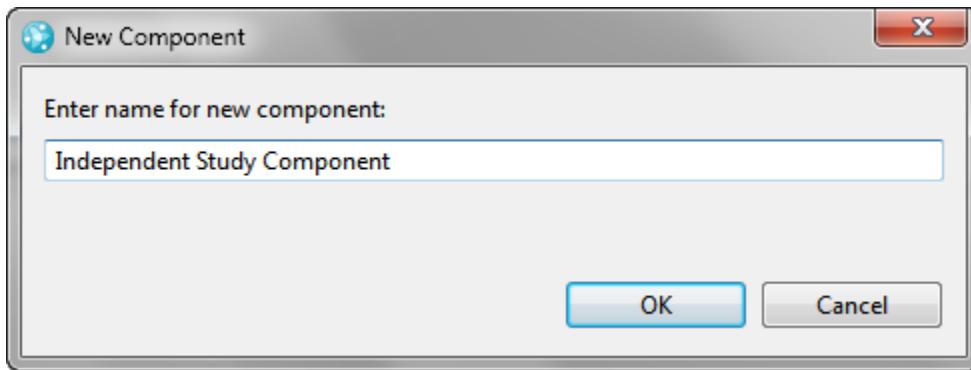
3. In the **Team Artifacts** window, right click on **Source Control > New... > Stream...**



4. **New Stream** window appears.

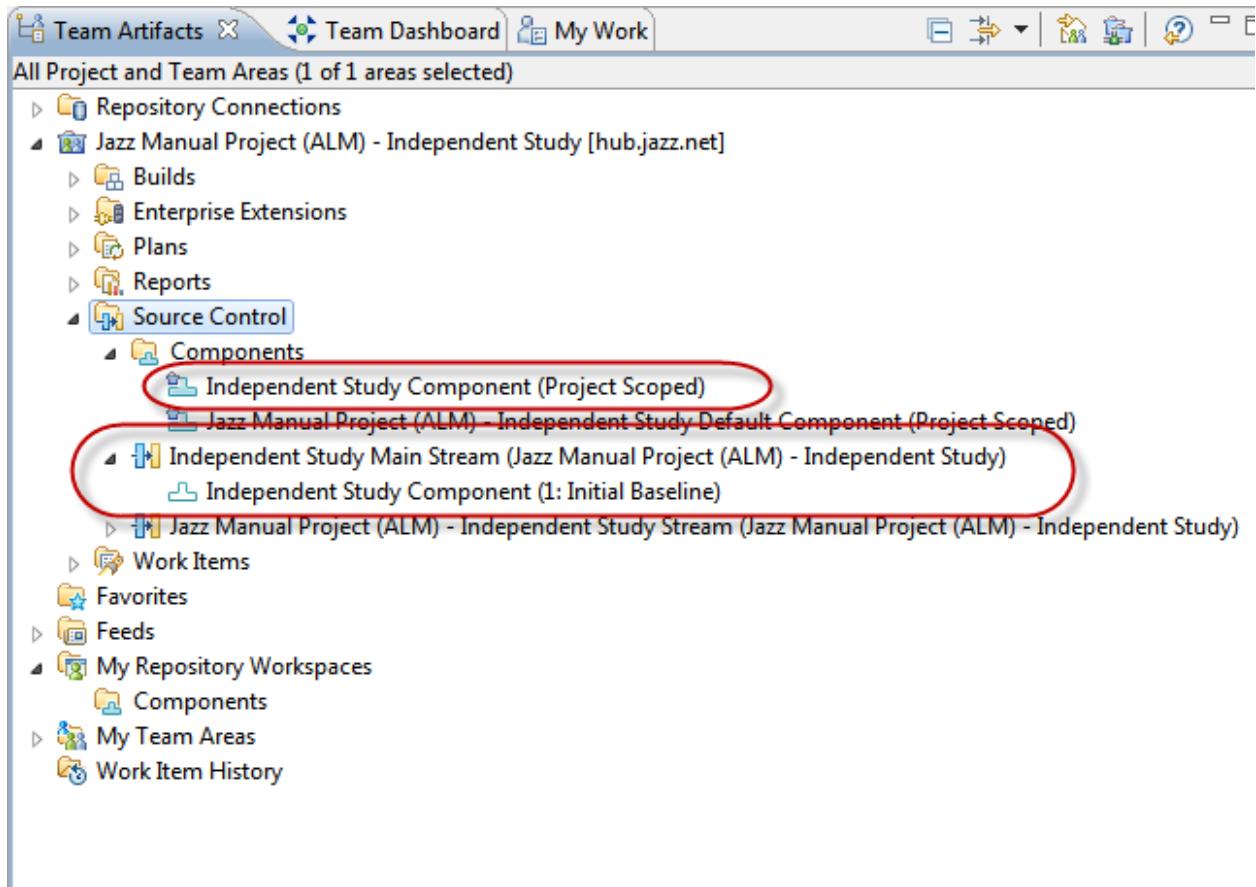


5. Fill in a Stream Name and Description. Verify that the stream is using the proper **Repository** and **Project**
6. To add a new component, click on the **New...** button within the **Component** section.



7. Enter a **New Component** name and click **OK**.

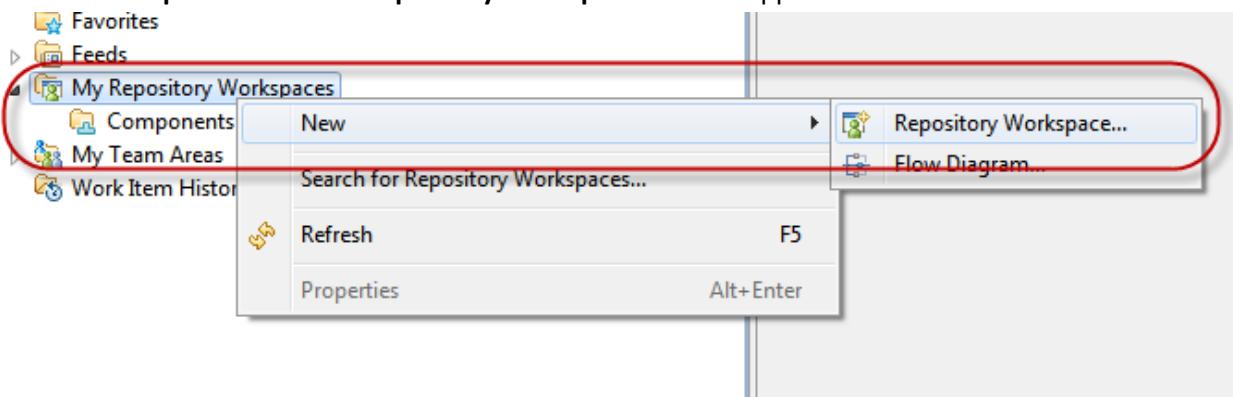
8. Click **Save** on the top right hand side of the **New Stream** window. The new stream and component should appear in the Team Artifacts Window.



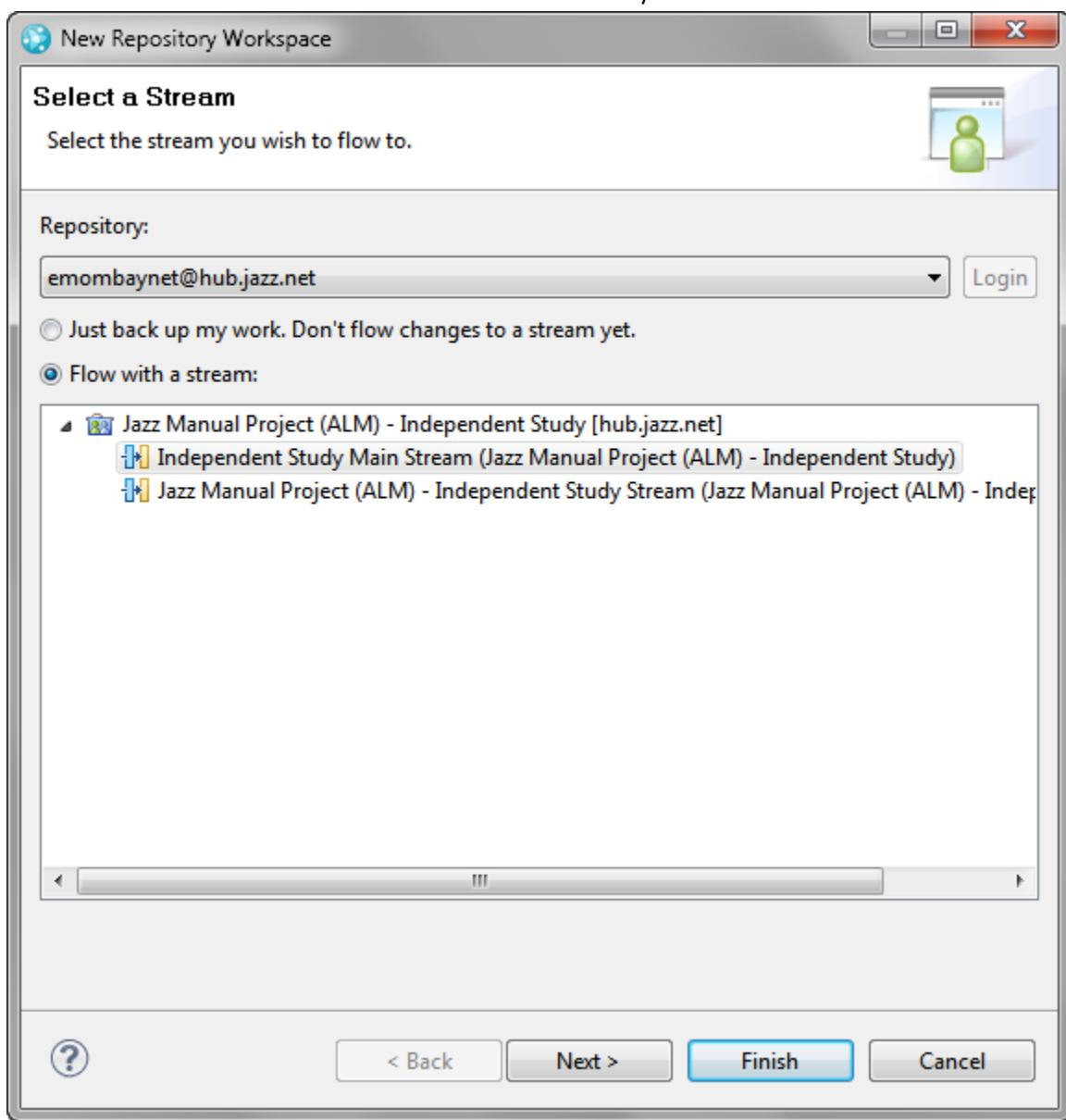
Create a workspace

As a note, you can only have one active repository workspace per local workspace. If you want to load a new repository workspace, your local workspace will be overridden with the new repo workspace files and folders.

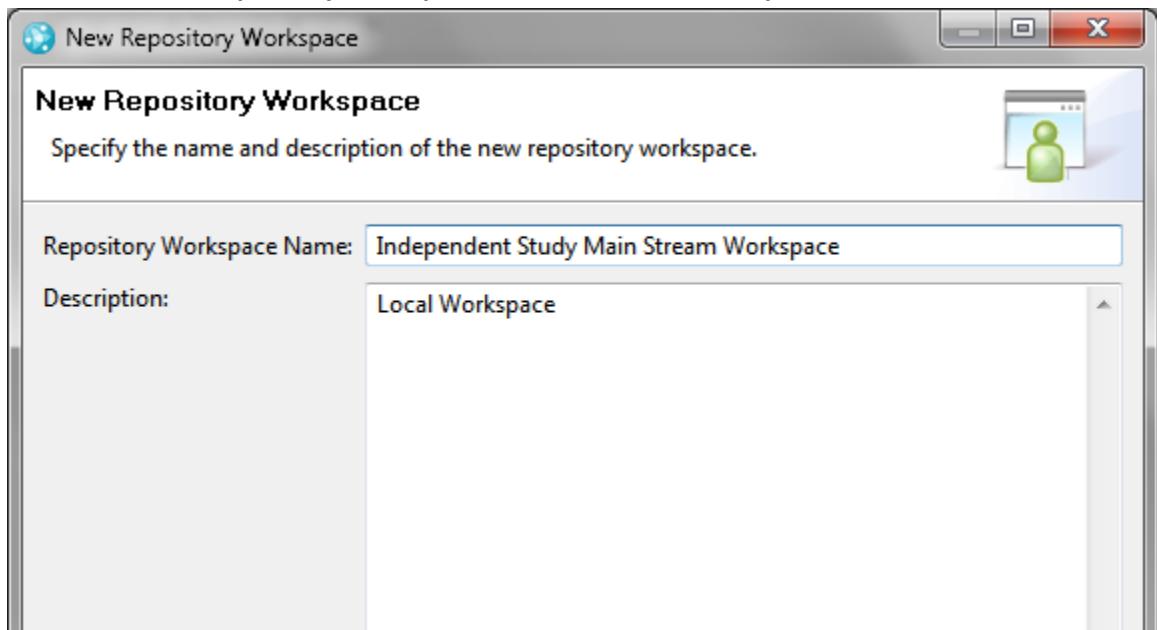
1. Start the **Rational Team Concert** application.
2. Make sure you are logged in to a **Repository Connection**. If not, log in using your Jazz.Net username and password.
3. In the **Team Artifacts** window, right click on **My Repository Workspaces** > **New... > Repository Workspace...** The **New Repository Workspace** window appears.



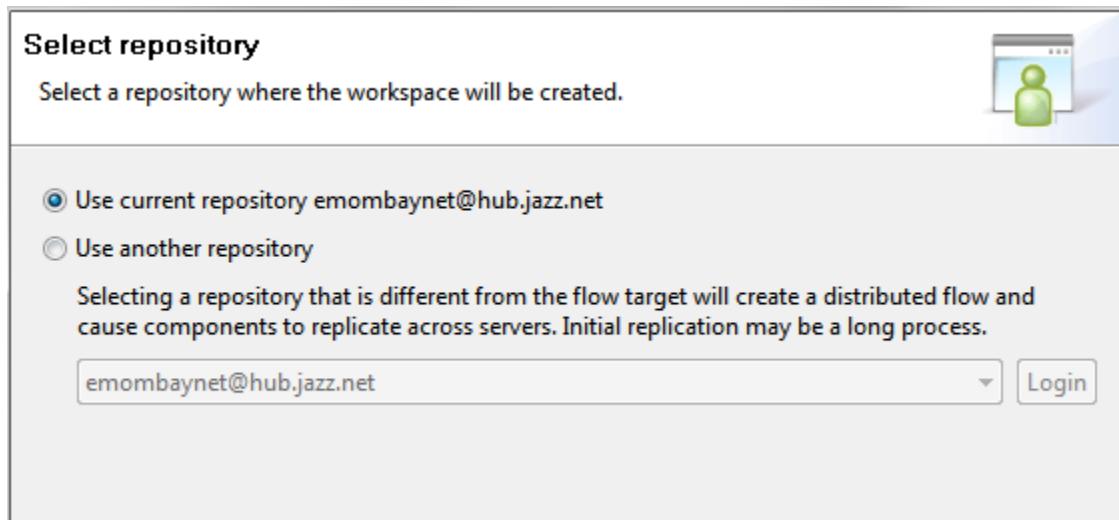
4. Select **Flow with a stream**: and select the stream you want to work off. Click **Next**.



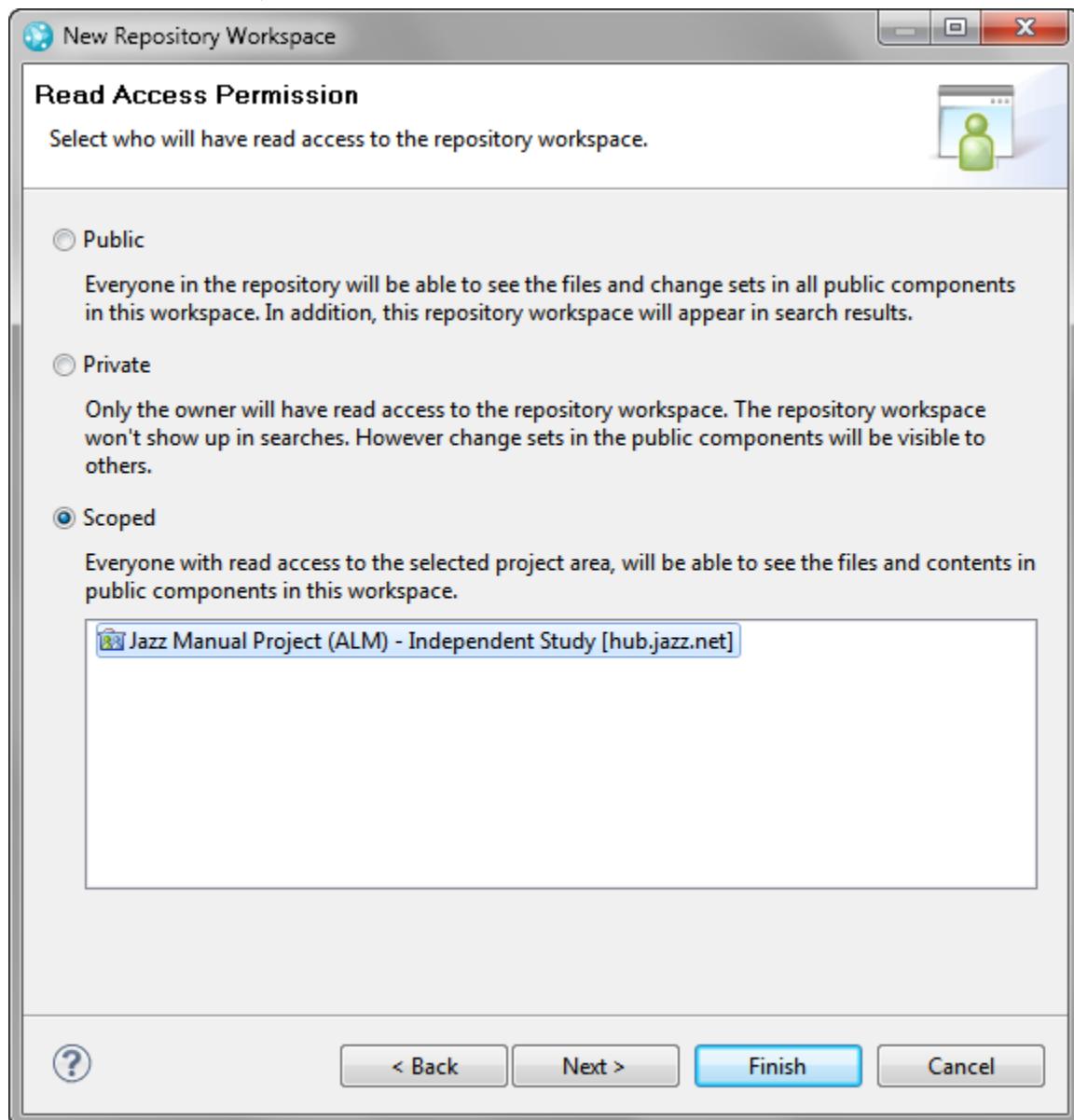
5. Create a **Repository Workspace Name** and add a **Description**. Click **Next**.



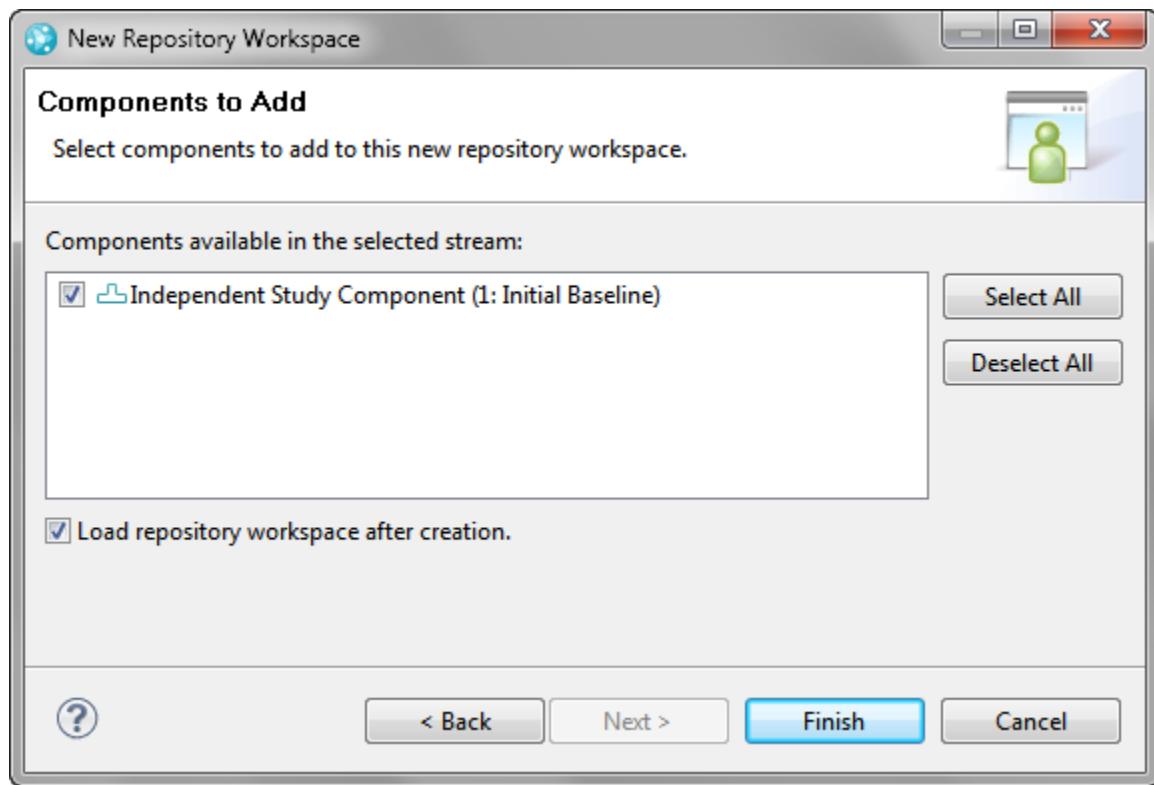
6. Use the current repository. Click **Next**.



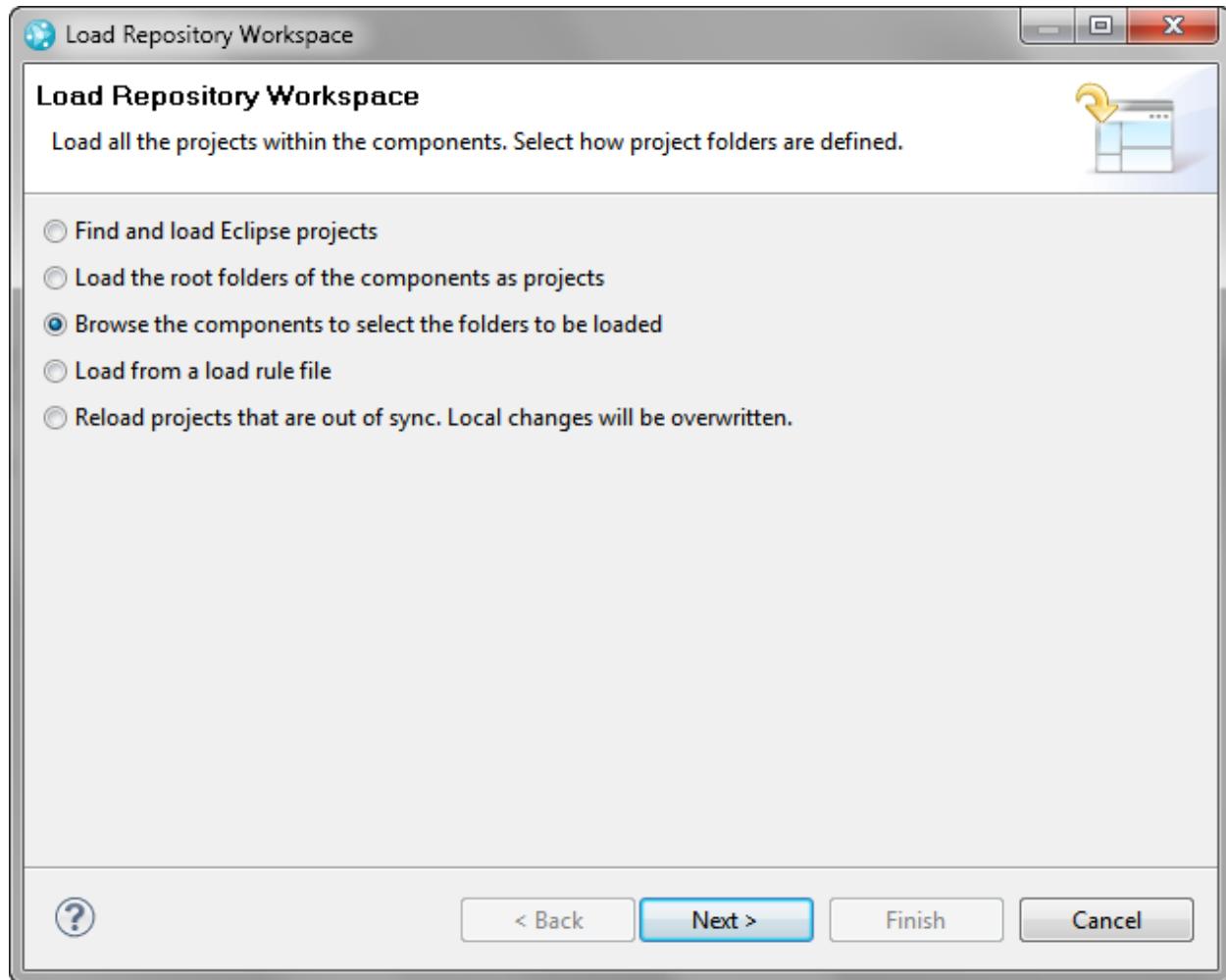
7. In the **Read Access Permission**, if you want to share your repository to anyone within the Project Area, select **Scoped**. If you want to keep the repository workspace private, as in you are the sole owner, select **Private**.



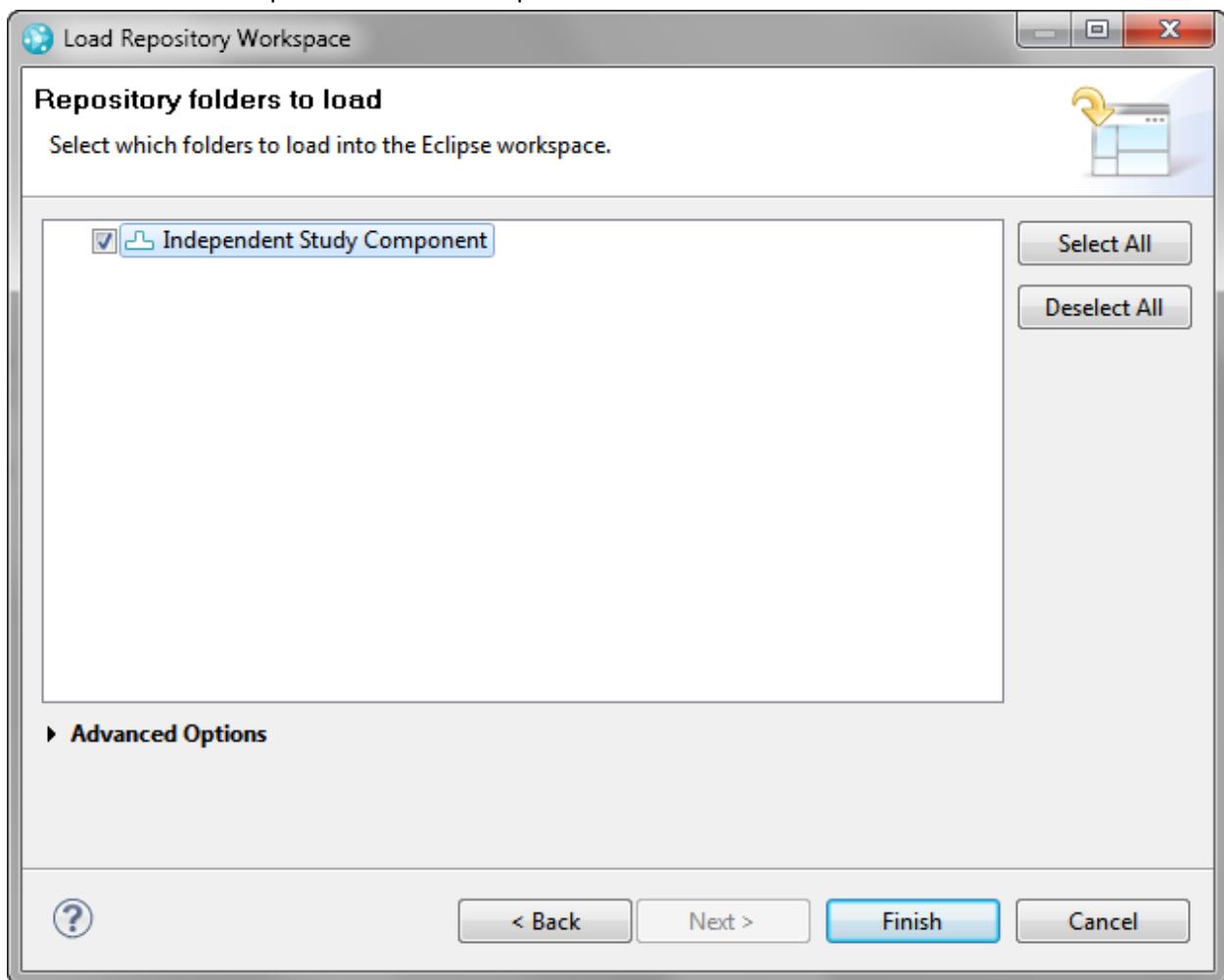
8. In the **Components to Add** section, select all the components you want to add to this repository workspace. If you want to load the workspace now, ensure the **Load repository workspace after creation** checkbox is checked. Click **Finish**.



9. Upon finish, RTC will attempt to load the workspace. The **Load Repository Workspace** window appears. You can select any type of loading based on the project type, but for our case, we can select **Browse the components to select the folders to be loaded**. If you are loading a code project, you can make Eclipse to load the project by selecting **Find and load Eclipse projects**.



10. Select the component for the workspace and click **Finish**.



11. The repository workspace should now be visible in the **Team Artifacts** window

