The source control component of the Jazz™ technology platform manages the source code, documents, and other artifacts that a team creates. It provides change-flow management to facilitate sharing of controlled artifacts, retains a history of changes made to these artifacts, and supports simultaneous development of multiple versions of shared artifacts, so that teams can work on several development lines at the same time.

A software development team typically works with a large base of files that comprise the source code for a software product or system. As a team member, you can add new features or fix existing ones within this base of source code. After you build and test the code to verify that your changes are correct, you can share the changes with the rest of the team. Rational Team Concert™ source control organizes versionable items (files and folders) into components and streams, and provides workspaces where you can view and modify file and folder contents. Together, these repository objects represent the configuration of the system being developed and allow any configuration to be retrieved, shared, or built. They organize files for a team, and track and share changes so that the entire team can work simultaneously to achieve common goals.

Rational Team Concert source control is closely integrated with the other application development lifecycle tools included in Rational Team Concert.

* The Jazz Build Engine and Build System Toolkit have built-in support for loading files from Rational Team Concert source control, and capturing snapshots of build input so that a build can be reproduced exactly. It also provides direct access to a rich set of tools that you can use to view the component versions in a specific build and compare them with versions in other builds, streams, and workspaces.

For more information, see [Building with Rational Team Concert Build](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.build.doc/topics/t_build_overview.html?view=kc).

* Change sets can be linked to work items, which support traceability of individual changes and insight into the reasons why they were made.

For more information, see [Tracking work by using work items](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.workitem.doc/topics/t_tracking_work_items.html?view=kc).

* Process preconditions can be used to control the flow of change sets. For example, you can configure a process so that a change set must be reviewed and approved before it can be delivered to an integration stream.

For more information, see [Administering change and configuration management projects](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.jazz.platform.doc/topics/t_projects_teams_process.html?view=kc).

* [**The repository**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_repo.html?view=kc)

Rational Team Concert source control uses a secure repository that is hosted on a server and accessed by clients through a URL. The repository stores objects such as streams, work items, and workspaces that help manage change flow. It also stores controlled artifacts that represent and can be retrieved as files or folders in a file system.

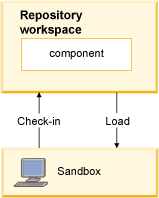
* [**Workspaces and sandboxes**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_workspaces.html?view=kc)

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 sandbox on the client, such as an Eclipse workspace or an ordinary folder.

To support integration with file-based tools such as editors, compilers, and debuggers, files, and folders in a repository workspace are loaded (copied) into a sandbox on your computer. As you modify files and folders in the sandbox, you periodically check them in, which copies the changes to the repository workspace. When all the changes in your sandbox are checked in, the repository and sandbox have the same content.

[Figure 1](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_workspaces.html?view=kc#c_workspaces__fig_ws) illustrates a simple configuration of a repository workspace and a single component. Check-in copies changes from the sandbox to the repository workspace. Load updates the sandbox with the contents of the repository workspace.

*Figure 1. Check-in copies work from your sandbox to your repository workspace*



* [**File content types and line delimiters**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_lineterm.html?view=kc)

When your team develops software on more than one platform, differences in platform conventions for line delimiters (line termination characters) in text files can affect content merges and other development tasks. Rational Team Concert source control stores text files in a normalized format. It also allows users to specify the line delimiter and MIME type properties on files.

* [**Read access permissions**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_read_access.html?view=kc)

Owners and administrators can change the read access permissions for streams, repository workspaces, components, and files and folders. If a component or stream is owned by a team or project area, the process area permissions determine who can set read access permissions.

* [**Creating custom attributes**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/t_creating_custom_att.html?view=kc)

You can create custom attributes on source file versions, streams, snapshots, components, and baselines. You define custom attributes in the project area. You can query for custom attributes from the command line interface.

Custom attributes support the extensibility and traceability for artifacts. For example, after you deliver a file, you can record extra comments about a specific file version in a comments attribute. You can write a script that runs a static analysis tool on the team's source code, and saves file-specific errors or warnings into an attribute on specific source file versions. Code inspection information can also be recorded on source code (for example, whether the file version was inspected and who inspected it). You can keep information about what release a baseline is part of, or what build generated a specific snapshot.

**Procedure**

1. Navigate to the project area in the administrative web interface.
2. In the project area or team area, click **Source Control** > **Attribute Definition**.
3. On the Source Control - Attribute Definition page, click **Add**.
4. In the Add Attribute window, complete the appropriate fields and select the types of objects that allow the custom attribute. If file is selected, in the **Behavior when a new file version is created** area, select one of the following options:
   * **Do not create the attribute**: If the custom attribute existed on the previous version of file, it is not created on the new version of the file when you check in your changes.
   * **Copy the attribute value from the previous version**: If the custom attribute existed on the previous version of the file, it is created on the new version of the file and its value is set to the value of the attribute on the previous version.
   * **Reset the attribute value to the default value**: If the custom attribute existed on the previous version of file, it is created on the new version of the file and its value is set to the default value of the attribute as defined in the project area.

**Note**If version of a file or baseline is part of a component that is owned by multiple project areas, you can set a custom attribute only if it defined in all parent project areas.

1. Click **Add**.
2. Click **Save**.
3. Optional: To edit an attribute, hover under the Actions column and click **Edit Attribute**. In the Edit Attribute window, you can change the name, types, default values and checkin behavior (when a new file version is created). Save the changes.
4. Optional: To delete an attribute, hover under the Actions column and click **Delete attribute** to delete the attribute.

* [**Streams and components**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_streams.html?view=kc)

A stream is a repository object that includes one or more components. A component is a collection of related artifacts, such as a set of related Eclipse plug-ins or a group of documents that comprise website content.

* [**Change sets and change flow**](https://www.ibm.com/support/knowledgecenter/SSYMRC_6.0.5/com.ibm.team.scm.doc/topics/c_csets_flow_intro.html?view=kc)

A change set is a repository object that collects a related group of file, folder, and symbolic link modifications within a component. Change sets flow between repository workspaces and streams. The workspace's flow targets specify the sources and destinations of incoming and outgoing change sets.

Setup Steam =>

1 : In Source Control Management/All Stream/Right Click/New/Stream/ give the name of Stream as well as grag down and add Component

2 : now Select Stream(Which you have made recently)/Right Click/new/repository

3 : now on All Repository Workspace/right click/load/browse the component to select the folder to be loaded/next

4 : now copy the java files or any file in components folders in local drive

Note-components folders=> automatically build on your system when you configure your eclips and gave path to save data locally on your machine

5 :now In pending change set Tab/right click on workspace/locate change set

6 : now windows Tab/Open perspective/select JAVA/ok

7 : Now In Eclips left corner you can see package --- TAb/click on it/you can see your components which you have made

previsily/right click/refress all components one by one

8 : Now In Eclips Right corner you can see WorkItem TAb/click

9 : Now In Pending changes Tab/Expand Repository workspace/select all components for which you want to checking the code/right click/check-in & Deliver/

Add some comments/next/create WI or Select any exesting WI for tracking this stream code/next/finish

10 : Now you can check all the component files on RTC Eclis client by right click on any component/show/repository files

----here you can see all files which you have lodaed

11 : Now you can check all files on RTC WEB by /Source control management/stream and you can open the Tracked WI to check change set

If you added any file from RTC Web Client or Made some changes on code then in Eclips /In pending change set Tab/Expand Repository workspace/

there would be a folde name incommine /right click/accept

If you want to add any file want some changes on code and save then follow the Steps from 5 - 11