

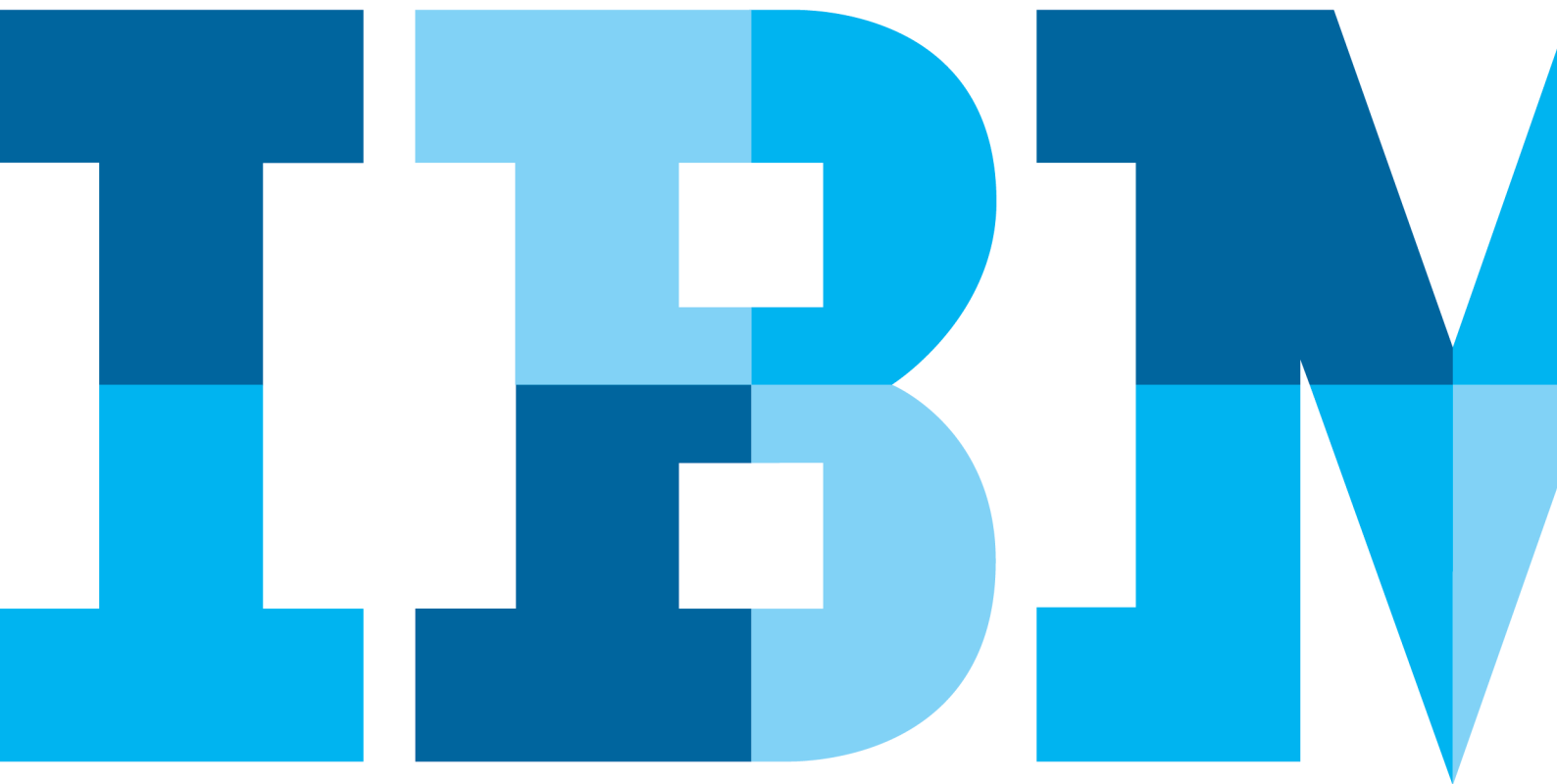
IBM Software

Rational

Simulink Projects Integration with Rational Team Concert

User Guide for Release 1.0

Steve Rooks, IBM IoT UK



IBM

Contents

1.	GLOSSARY	3
2.	SCOPE	3
3.	CONNECT TO RTC.....	3
4.	ADD NEW MSP PROJECT TO RTC.....	3
5.	OPEN AN EXISTING PROJECT FROM RTC.....	6
6.	ADD AND MODIFY FILES.....	6
7.	ACCEPT INCOMING CHANGES FROM YOUR TEAM	7
8.	HANDLING CONFLICT	8
9.	CHEAT SHEET	10
10.	ADVANCED	11
11.	CREATING YOUR OWN RTC WORKSPACE.....	11
12.	RTC COMMAND LINE MAPPING	12
13.	INSTALLATION OF RATIONAL TEAM CONCERT INTEGRATION	12
	THE FOLLOWING PRODUCTS MUST BE INSTALLED ON THE WORKSTATION:.....	12
14.	EXTERNAL MERGE TOOL SETUP.....	13
15.	TROUBLE SHOOTING.....	14

1. Glossary

RTC – Rational Team Concert, providing file configuration and change management with planning

Jazz – IBM’s engineering collaboration platform

MSP – Mathwork’s Simulink Project

2. Scope

This document describes how to use Mathwork’s Simulink Project with Rational Team Concert using the direct integration adapter.

For a new user to MSP, please supplement this guide with the built in help available from MSP on using source control.

This guide also doesn’t cover how to setup new projects and streams within RTC. This should be done by a projet admin.

3. Connect to RTC

For connecting MSP to the RTC server you need two things:

1. URL of RTC (E.g. <https://jlr.rtc.com>)
2. Your RTC username and password

With this information, goto the Matlab command window and type the following. Note the use of ‘!’ to invoke an external command.

```
>>!lscm login -r https://<rtcURL>/ccm -n RTC -u <username> -P <password> -c
```

Here’s an example:

```
>>!lscm login -r https://jlrtt.icds.ibmcloud.com/ccm -n RTC -u SteveR -P foobar -c
```

```
Logged in to https://jlrtt.icds.ibmcloud.com/ccm
```

By specifying the –c option, your password will be cached permanently

To confirm the connection is good, try listing the workspaces on the repository:

```
>>!lscm list workspaces -r RTC
```

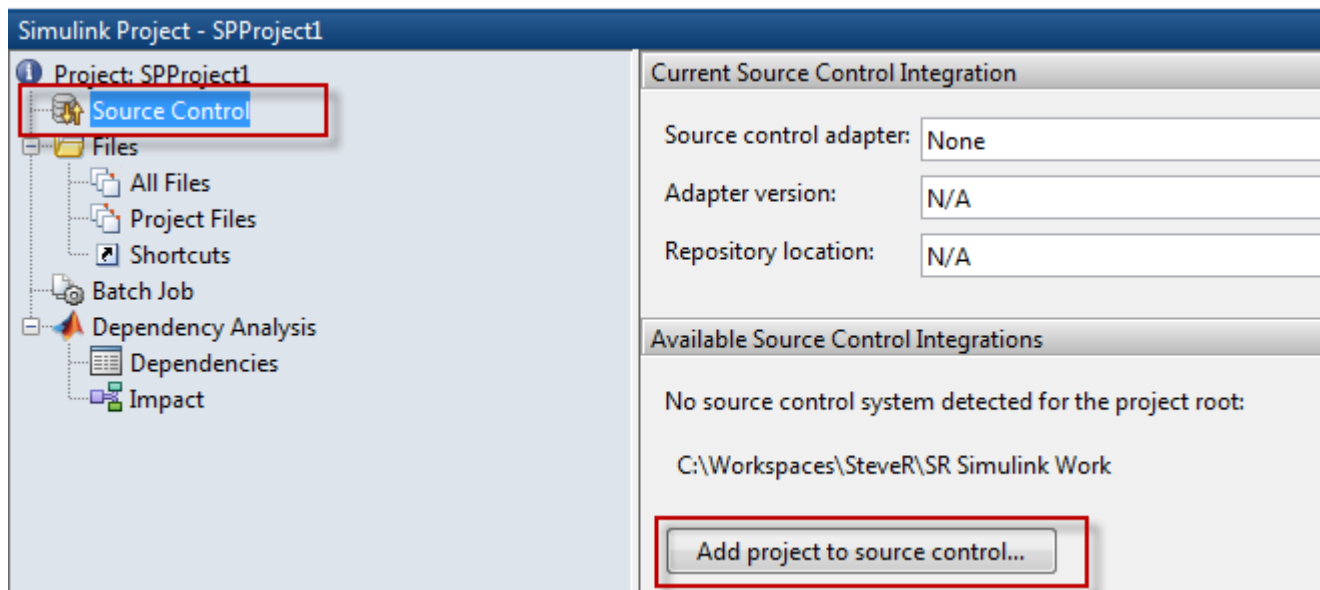
```
<list here...>
```

4. Add new MSP project to RTC

To create a new MSP project on RTC, you’ll need a RTC repository workspace with a component. Note that you could also create your own private workspace which will backup your work whenever you check-in changes, See 11.

To create the MSP project and share it with RTC:

1. Ensure you can connect to RTC via the command line (See 3)
2. Have an empty RTC component in which you’ll store the model
3. Create a directory for RTC on your local file system which will become your sandbox (E.g. C:\RTCSandbox)
4. Within this directory, create a directory which will correspond to your component (E.g. C:\RTCSandbox\MyComponent)
5. Start MSP and navigate to the component directory (E.g. C:\RTCSandbox\MyComponent)
6. Within MSP, do New->Simulink Project->Blank Project and fill in the name, click create
7. Within MSP choose Source Control, Add Project to Source Control



8. Now choose the RTC SCM Integration as the source control adapter and fill in the component and workspace information. To get the names you can run the following commands:

```
>>!lscm list components -r RTC
```

```
(1332) "Dev Env for Chassis Default Component"
```

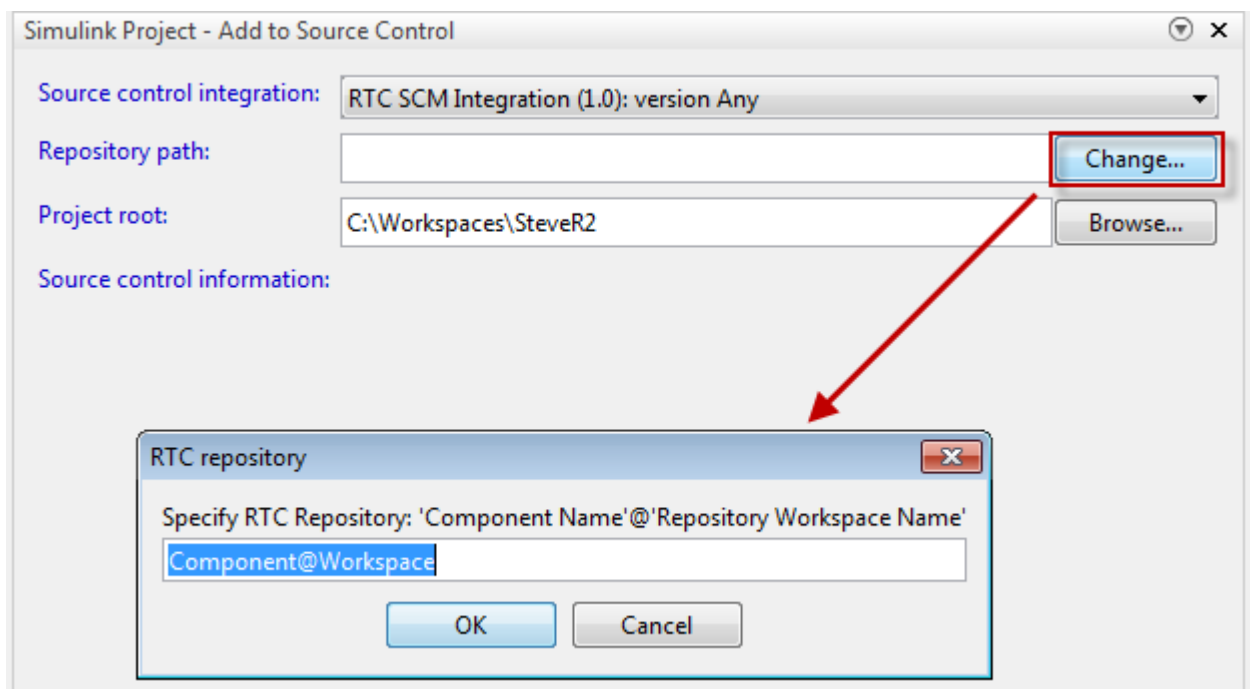
```
....
```

```
>>!lscm list workspaces -r RTC
```

```
(1322) "Dev Env for Chassis Stream Workspace" mike staunton
```

```
....
```

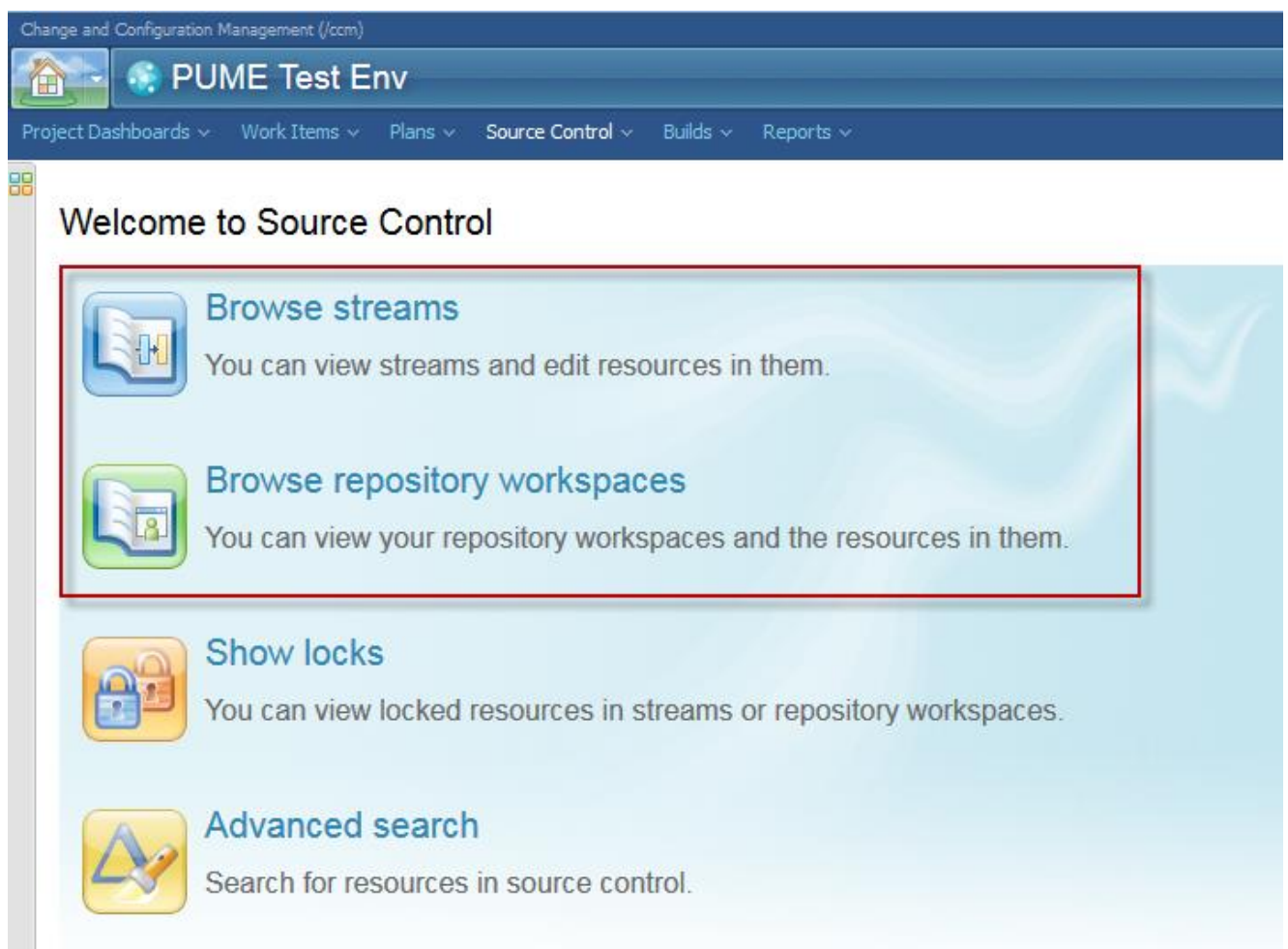
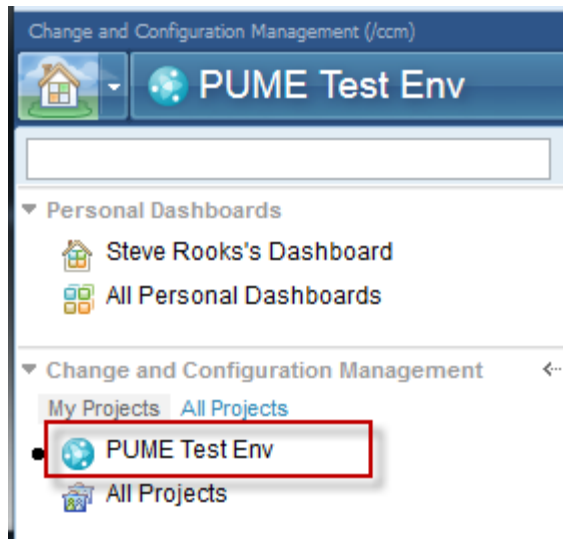
9. Now enter the Workspace and Component information, note the format! Also note you can enter the numerical values instead (E.g. 1332@1322)



10. Finally, click **Convert** and the project will be shared with the RTC workspace
11. If you're curious you can see the files shared by typing:

>> `!lscm status -C`

12. Another way of looking at the files is via the web. Log into the RTC project under Change and Configuration Management, and browse the workspaces and streams (note you must have delivered the changes to see them in the stream).



13. If this is not a private workspace, you can deliver the new project to the stream for sharing with the team, to-do this the changeset must be delivered. Type the following:

```
>>!lscm deliver
```

5. Open an existing project from RTC

To open a MSP project that has already been shared to RTC, do the following:

1. Create a directory for RTC on your local file system which will become your sandbox (E.g. C:\RTCSandbox)
2. Start MSP and navigate to the sandbox directory (E.g. C:\RTCSandbox)
3. Within MSP, do New->Simulink Project->From Source Control
4. Now enter the component and workspace information, see Section 4 on how to search for this information and retrieve the component
5. MSP will likely complain that it cannot find the project and ask you create a new one. Click No and manually navigate to the project file within the sub directory (which represents the component), double click to open

6. Add and Modify files

With an MSP project under RTC management, files (scripts, models, etc) can be modified and checked-in (to your Repository Workspace), then you can deliver the changes to the stream for sharing with your team.

To check-in changes:

1. Goto the Modified Files view in MSP
2. To check-in all changes click on the **Commit Modified Files** button
3. You are prompted for a comment.
 - a. If you leave this blank, the current active changeset will be used (error if this doesn't exist)
 - b. If you enter a comment and an active changeset exists, you will be prompted to choose.
4. You are then prompted to complete the changeset. If you intend to add further changes to the set, then leave it open. If however you have completed all of the changes and want to share it with your team then click to complete.
5. The modified and added files are checked-in
6. If you want to manually share the changeset with your team (not possible for private workspaces without a flow target), then goto the commandline and type:

```
>>!lscm deliver
```

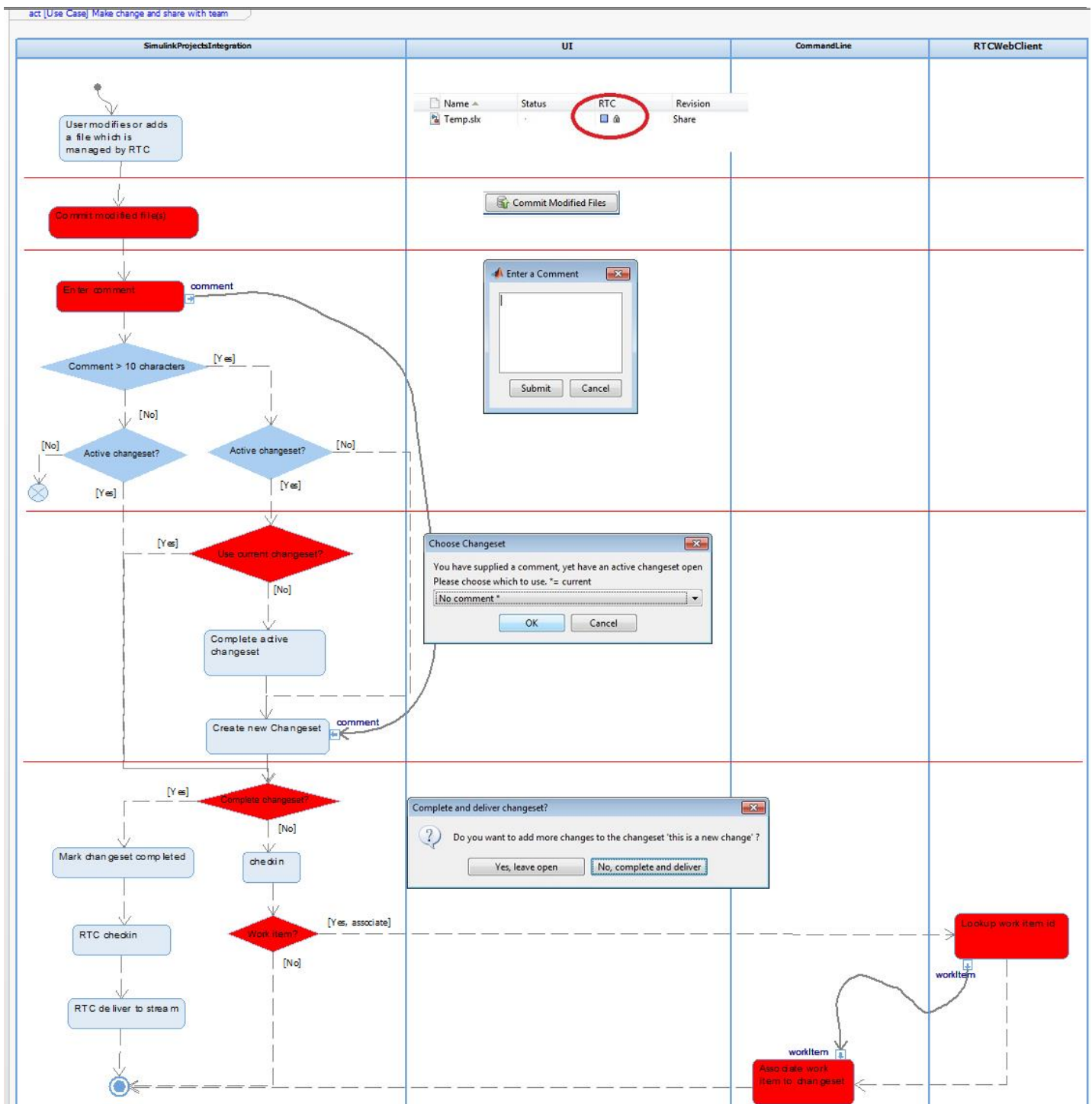


FIGURE 1 CHECK-IN WORKFLOW

7. Accept incoming changes from your team

When a team member has delivered changes to your stream (the stream your workspace flows to) , you'll have incoming changes that you can accept. RTC will not allow you to deliver your own changes to the stream if you have a conflict!

To accept incoming changes:

- Click **Update**

- Click **Continue**. If you have un-checked-in changes which conflict with the incoming changes, an error will be displayed. You can either:
 - Abandon your local changes using **Revert Local Changes and Release Locks**, then run **Update** again
 - Or
 - Check-in your local changes using **Commit Modified Files**, then run **Update** again. If you see conflict move to 8

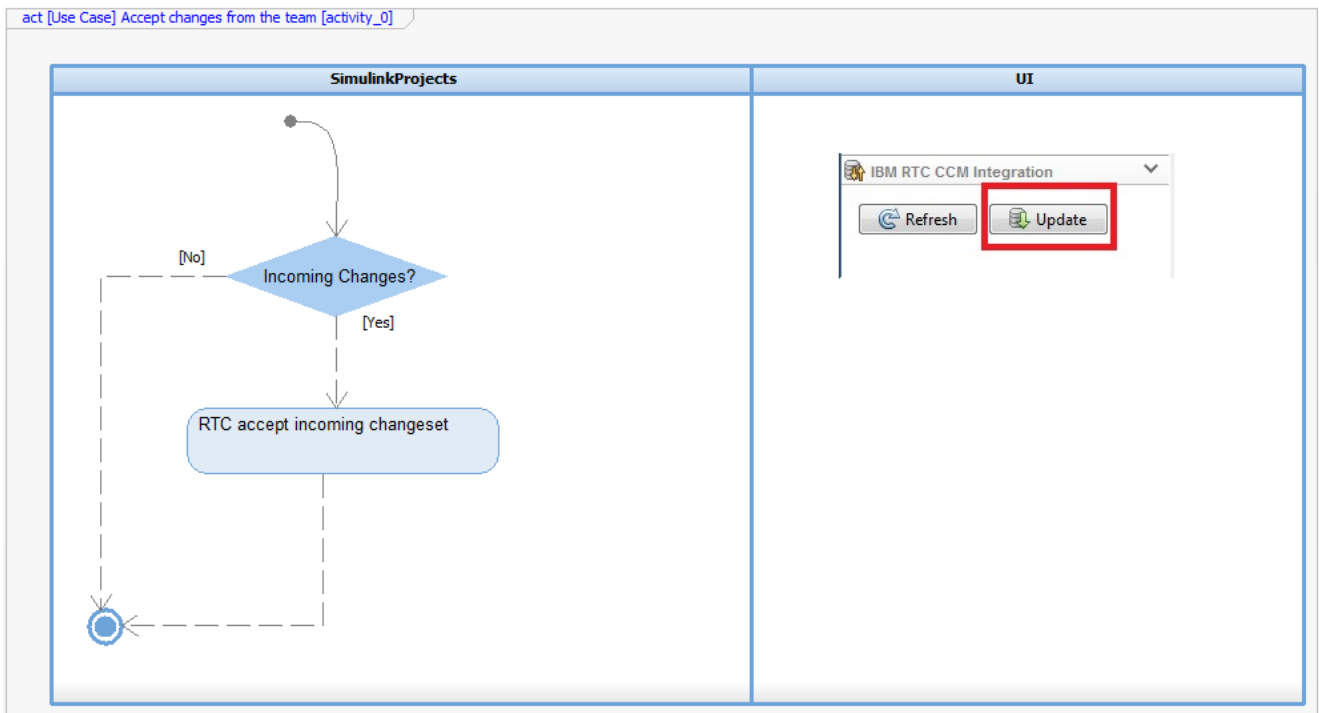


FIGURE 2 ACCEPT INCOMING CHANGES

8. Handling conflict

Conflict can occur when you try to deliver a change which overlaps (conflicts) with changes already delivered by your team to the same stream.

Within Simulink Projects, to resolve conflict, use the **Mark Conflict Resolved** menu. You will be prompted to choose either the proposed (incoming), mine (your change) or merge. If you choose merge, the Mathworks' visdiff() tool will be invoked which will allow you to perform a merge. To deliver this merge to your team, you must:

1. Merge and save back into the sandbox, overwriting your existing file
2. Check-in the new copy
3. Use the **Mark Conflict Resolved** menu option again, this time choosing mine (your change).

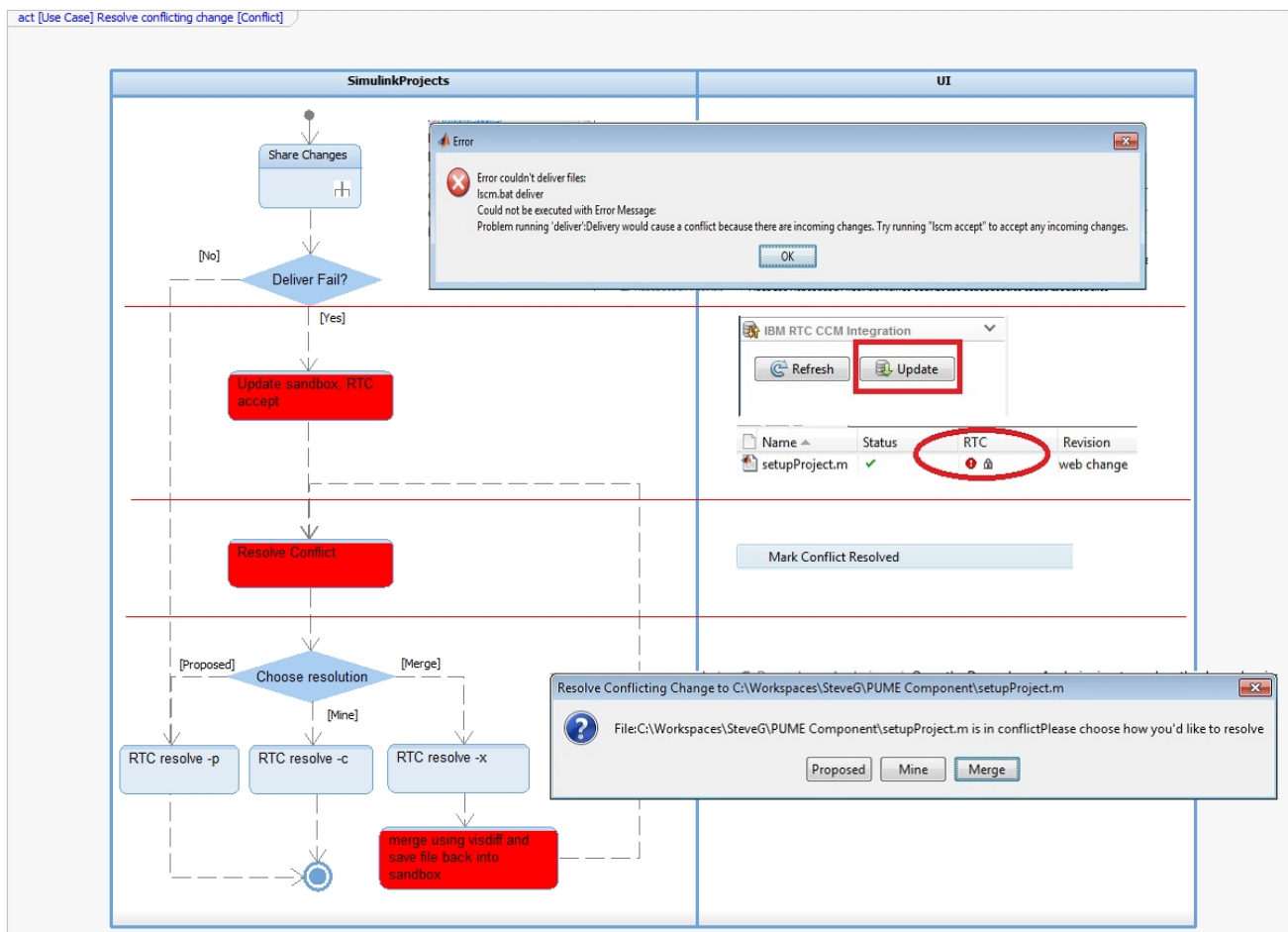


FIGURE 3 RESOLVING CONFLICT

9. Cheat sheet

The screenshot shows the 'Simulink Project - SPProject1' window. The top toolbar contains icons for Source Control, Files, Project Files, Shortcuts, Modified Files, Batch Job, Dependency Analysis, and Impact. The left pane shows 'Classification' and 'Review (String, Single Valued)'. The main area displays a table with columns: Name, Status, RTC, Revision, Date Modified, Review, and Classification. The 'Name' column lists 'sr1.slx (Simulink Model)'. The 'Status' column shows 'Enter'. The 'RTC' column has a checkbox and a folder icon. The 'Revision' column shows 'modified model for xyz'. The 'Date Modified' column shows '8/16/2015 5:26 AM'. The 'Review' column is empty. The 'Classification' column shows '62'. Below the table, there are buttons for 'Open', 'Run', 'Show in Explorer', 'Rename', 'Compare Selected Files/Folders', 'Compare Against', 'Add to Project', 'Commit to IBM RTC CCM Integration Repository', 'Update from IBM RTC CCM Integration', 'Refresh IBM RTC CCM Integration Status', 'Check for Modifications', 'Revert Local Changes and Release Locks', 'Show Revisions', 'Compare to Revision', 'Compare to Ancestor', 'Check for Dependency Analysis...', 'Label for Review', and 'Commit Modified Files'. Red arrows point from text labels to specific elements in the interface: 'status' points to the 'Status' column header; 'changeset name' points to the 'Revision' column header; 'check-in to repository workspace' points to the 'Commit to IBM RTC CCM Integration Repository' button; 'accept incoming changes' points to the 'Update from IBM RTC CCM Integration' button; 'undo latest change' points to the 'Refresh IBM RTC CCM Integration Status' button; 'compare to history (will invoke visdiff)' points to the 'Compare to Revision' button; and 'compare to latest change' points to the 'Refresh IBM RTC CCM Integration Status' button. The bottom of the window shows a 'Commit Modified Files' button.

Name	Status	RTC	Revision	Date Modified	Review	Classification
sr1.slx (Simulink Model)	Enter	<input type="checkbox"/>	modified model for xyz	8/16/2015 5:26 AM		62

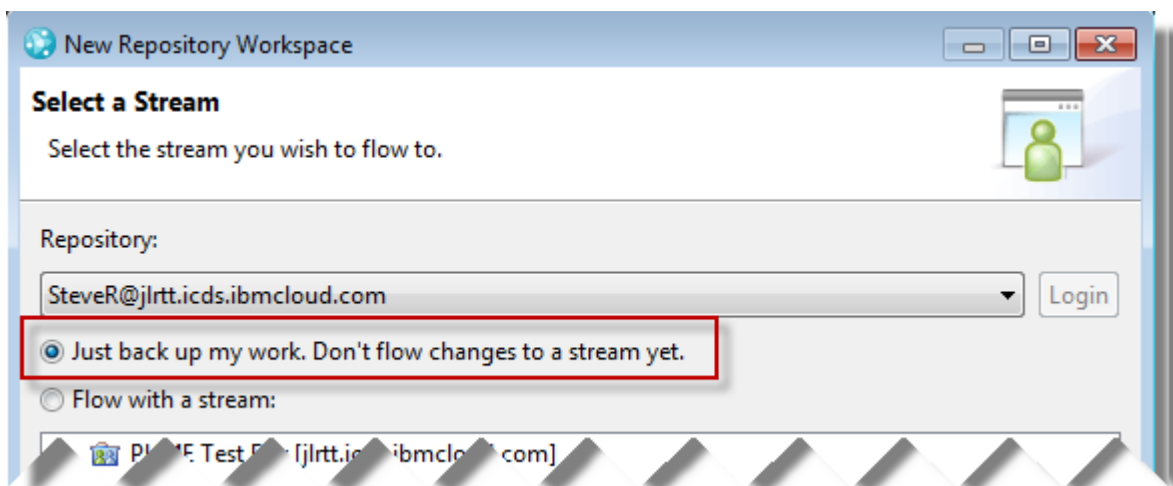
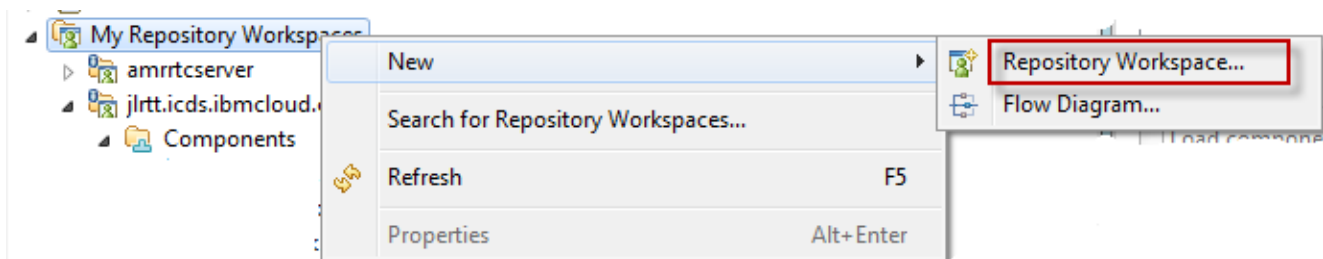
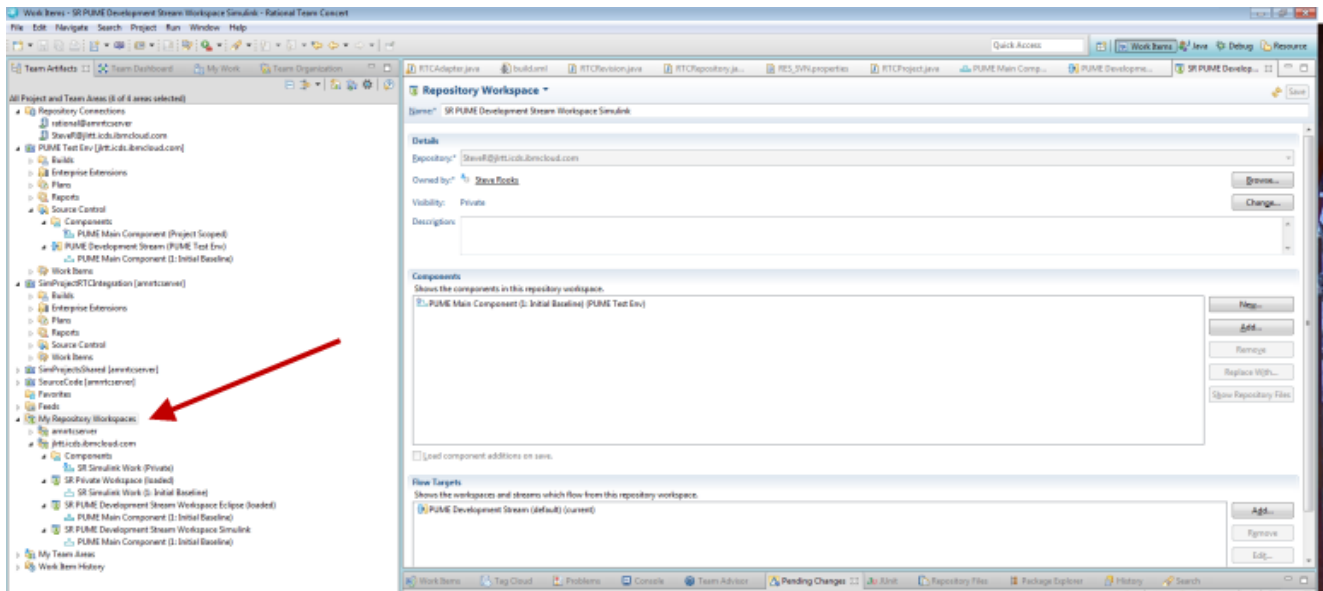
Annotations:

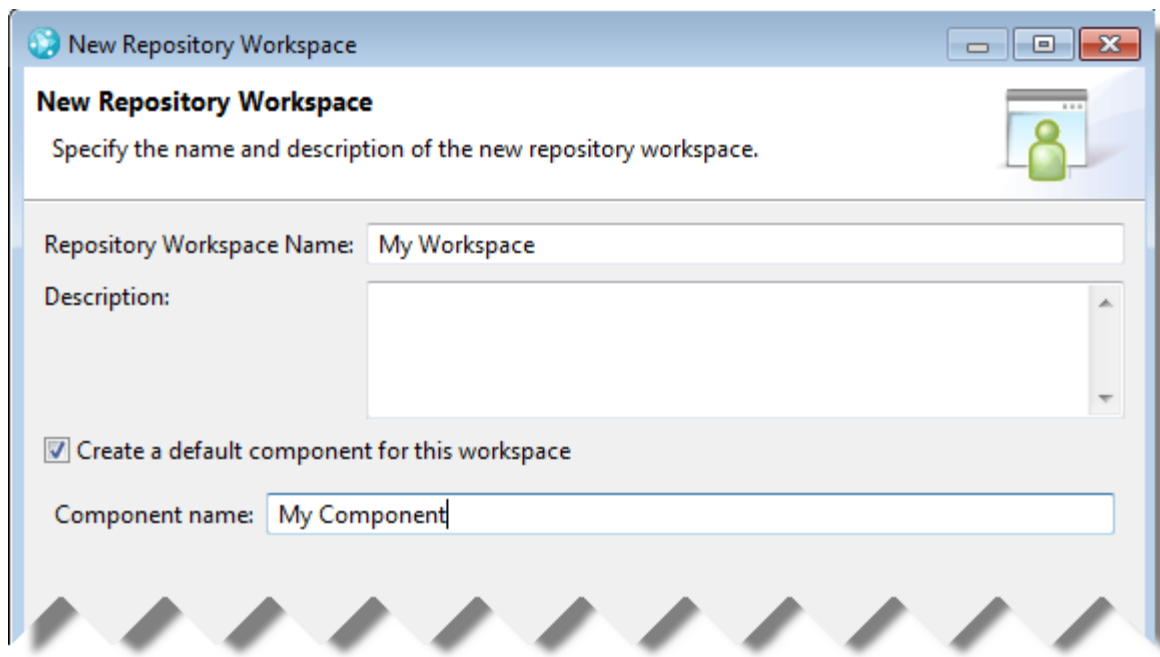
- status
- changeset name
- check-in to repository workspace
- accept incoming changes
- undo latest change
- compare to history (will invoke visdiff)
- compare to latest change

10. Advanced

11. Creating your own RTC Workspace

To create a new Repository Workspace you must use the RTC Eclipse client, via the Work Item perspective:





Finish and you'll have your own area.

12. RTC Command Line mapping

Simulink Projects	RTC scm command
Initial load	lscm load -r RTC <workspace> <component>
Check-in	lscm checkin
Update	lscm accept
Revision	

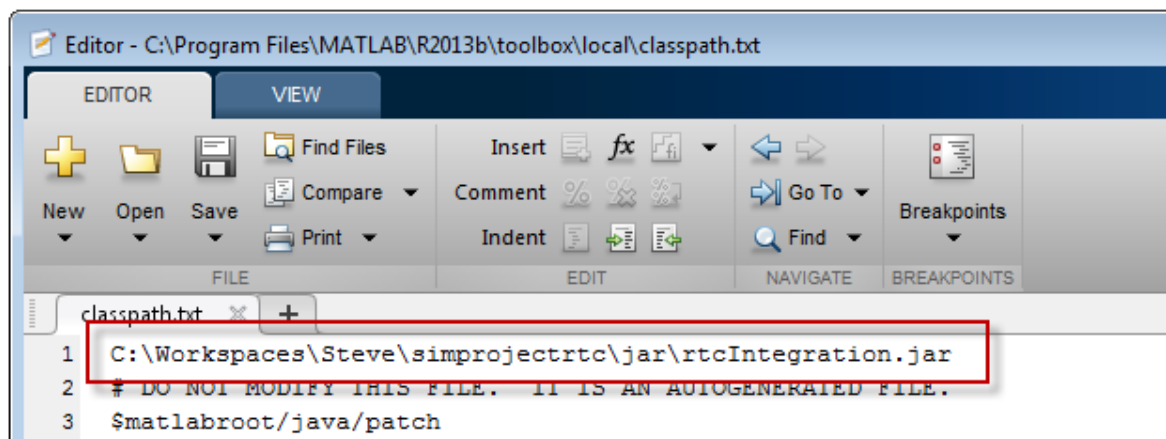
13. Installation of Rational Team Concert integration

The following products must be installed on the workstation:

- RTC command line (from jazz.net [5.0.2 <https://jazz.net/downloads/rational-team-concert/releases/5.0.2?p=allDownloads>]). Note that scm.exe needs to be available from the path.
- Simulink Projects from Mathworks
- RTC Integration adapter for Simulink Projects rtcIntegration.jar (from IBM)

Register the adapter with Matlab, by editing the classpath.txt file.

```
Trial>> edit classpath.txt
Trial>>
```



14. External Merge tool setup

To setup external merge tool (E.g. Mathworks Comparison Tool) configure RTC using preferences.

To see the current configuration, run:

```
>>! lscm list pref -v
```

...

...

Name : external.compare.2way

Description : External 2-way compare tool

Format: <full path to the executable> <options>

Note: Some characters such as " or \$ might have a special meaning in certain operating systems and you might need to be escape characters to represent them.

Examples:

external.compare.2way: <bcompare path> -title1="" -title2="" -rightreadonly

external.compare.2way: <xcleardiff path>

external.compare.2way: <diffmerge path> -ro1 -title2="" -title1=""

external.compare.2way: <kdiff3 path> --L1 "" --L2 "" --output

external.compare.2way: <p4merge path> -nl "" -nr ""

external.compare.2way: <winmerge path> -dl "" -dr "" -wr /e /u

Default Value : None

Current Value : None

Name : external.compare.3way

Description : External 3-way compare tool

Format: <full path to the executable> <options>

This tool compares items that have conflicts. If this tool is not configured, the 2-way compare tool is used by default.

Examples:

Note: Some characters such as " or \$ might have a special meaning in certain operating systems and you might need to be escape characters to represent them.

external.compare.3way: <bcompare path> -title1="" -title2="" -title3="" -title4=""

external.compare.3way: <xcleardiff path> -base "" -out "" ""

external.compare.3way: <diffmerge path> -result="" -title1="" -title2="" -title3=""

external.compare.3way: <kdiff3 path> --L1 "" --L2 "" --L3 "" --output

external.compare.3way: <p4merge path> -nb "" -nl "" -nr "" -nm ""

Default Value : None

Current Value : None

To setup a new diff tool, run:

```
>>!scm set pref external.compare.2way "<matlab path>\matlab.exe -r visdiff('${file1Path}','${file2Path}')"

```

Note the <matlab path> cannot have spaces, so use the 8.x format (E.g. C:\Progra~2\MATLAB\R2013b\bin\win32\matlab.exe)

15. Trouble shooting

- To kill the RTC command line daemon

```
>>!scm daemon stop -a

```