

[tutorial](#) > DevOps

Migration from IBM Rational Team Concert to GitHub

Simplified migration and implementation



Kabilan Saravanabava
Distribution June 29, 2017



In this tutorial, we present the experience of implementation on my team and the tips and tricks I learned during the implementation process, for developers or administrators who are currently using IBM Rational Team Concert and can quickly move to GitHub To do.

The contents of this tutorial are as follows.

- Overview of Rational Team Concert and GitHub
- Precondition
- How to create a GitHub repository
- How to set up GitHub Desktop and Rational Team Concert client
- How to configure rtc2git package
- How to migrate from IBM Rational Team Concert to GitHub
- How to set access authority for GitHub
- Common problems and solutions

Overview

Migrating from Rational Team Concert to GitHub requires a series of scripts. With these scripts, identify the baseline of the stream in Rational Team Concert that is going to GitHub and publish the files and folders in the stream to GitHub.

If there is no interest in a file or folder of history that is included in the stream, create a copy of the latest stream on the local workspace, and then commit the stream to GitHub. If you want to migrate along with the source code data and history, and then migrated using rtc2git tool.

Precondition

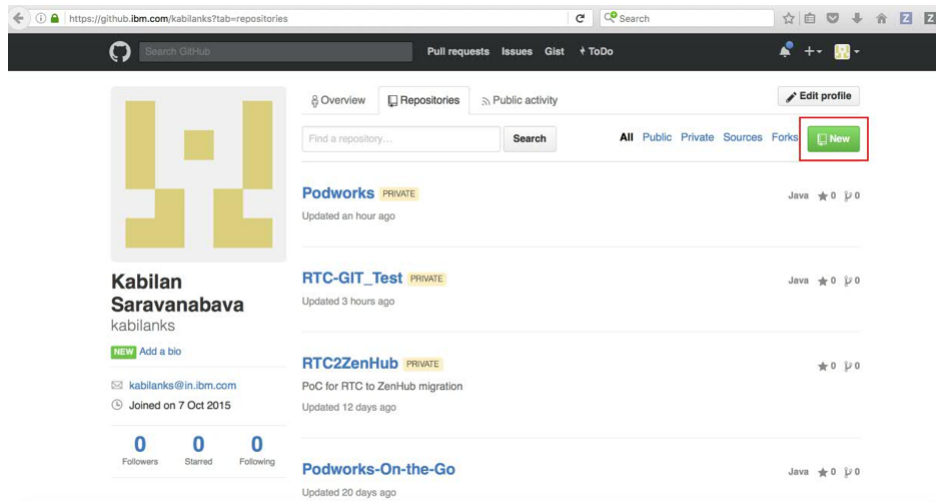
The prerequisites for completing the steps in this tutorial are as follows.

- Rational Team Concert Server 5.0 or later
- The [Collaborative Lifecycle Management](#) tool for your version of Rational Team Concert Server
- The [Rational Team Concert CLI](#) has been configured
- A Rational Team Concert client that matches your version of [Rational Team Concert Server](#)
- Python version 3.4 or later is installed on the machine that runs rtc2git tool (migration script will not work on Python 2 or earlier versions)
- [Rtc2git](#) tool
- GitHub Desktop client

Create a GitHub repository

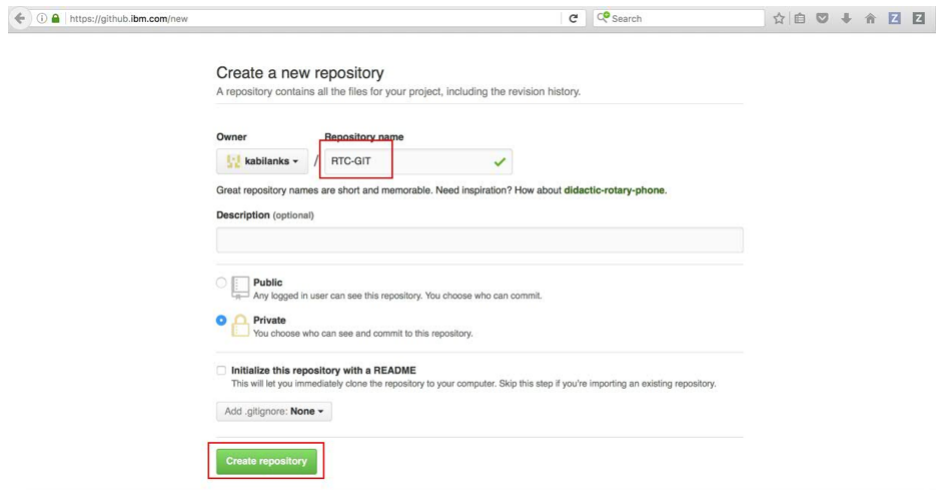
Before you begin the migration, you need to set up a GitHub repository for migrating the source code data of Rational Team Concert. The following figure shows a list of GitHub repositories. To create a repository, click **New**.

Figure 1. Creating a GitHub repository



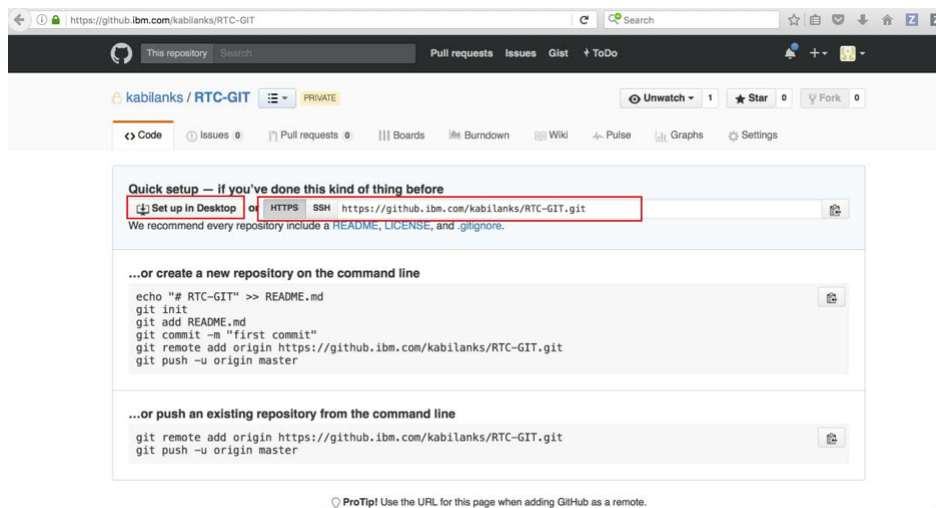
Enter the name of the repository in the text field. You can choose whether to create a private repository and the public repository. In the following example, the private repository is selected. Access to the repository can be expanded after migration is complete. Click **"Create repository"**. You can change the name of the repository at any time.

Figure 2. Private repository created within GIT



Specify a name, **"Create repository (create a repository)"** please click. Make a copy to paste the URL of the newly created repository in GitHub Desktop client.

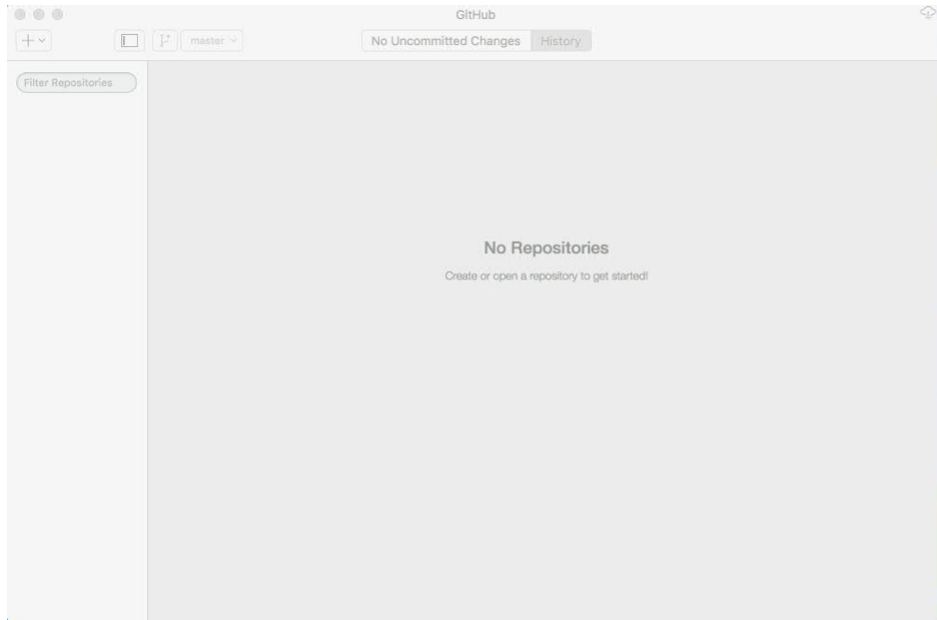
Figure 3. Setting up the GIT Desktop client



Set up GitHub Desktop and Rational Team Concert Eclipse client

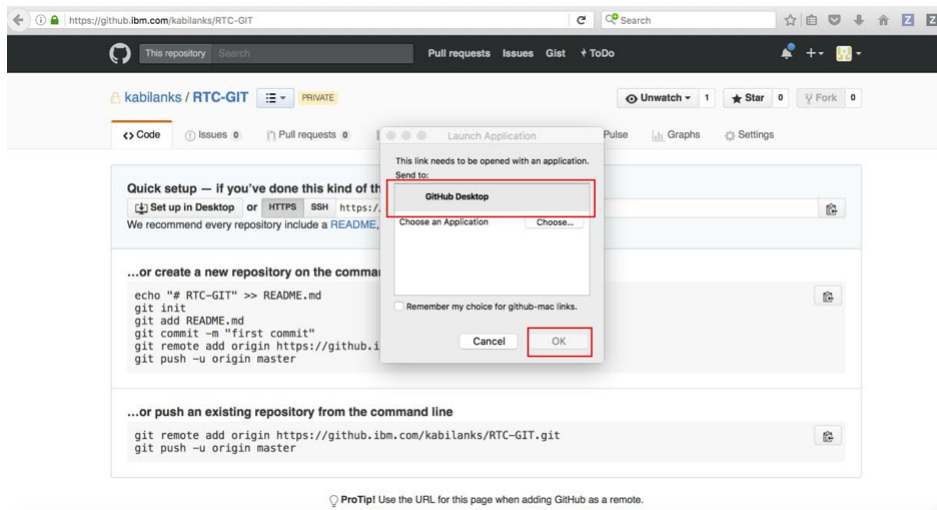
Open the GitHub Desktop client. Blank page that GitHub repository is not connected will be displayed.

Figure 4. GitHub Desktop client



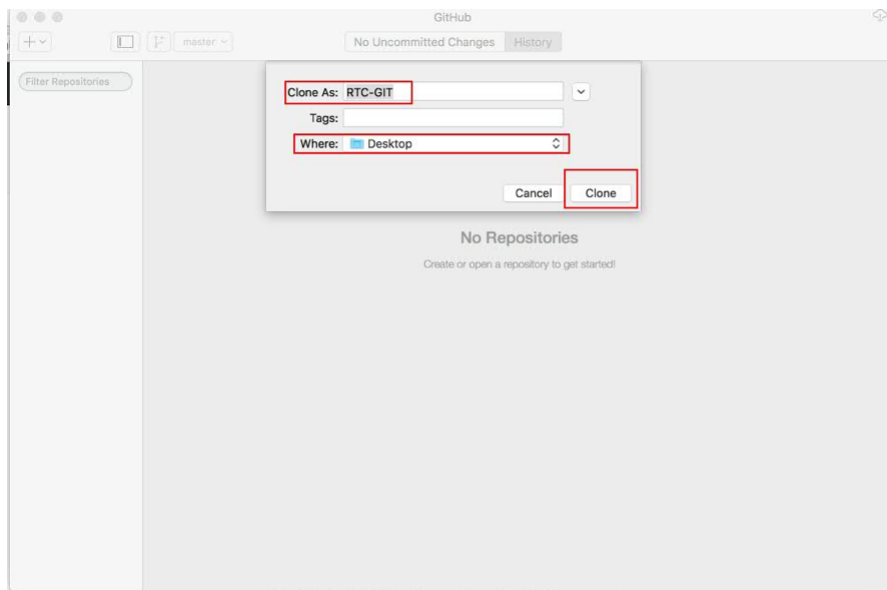
The following figure shows how to set up the GitHub Desktop client from the GitHub server. On the newly created GitHub repository, click the "Set up in Desktop" button. A dialog box opens prompting you to send to GitHub Desktop. Click "OK". GitHub Desktop opens.

Figure 5. Setting up GitHub Desktop



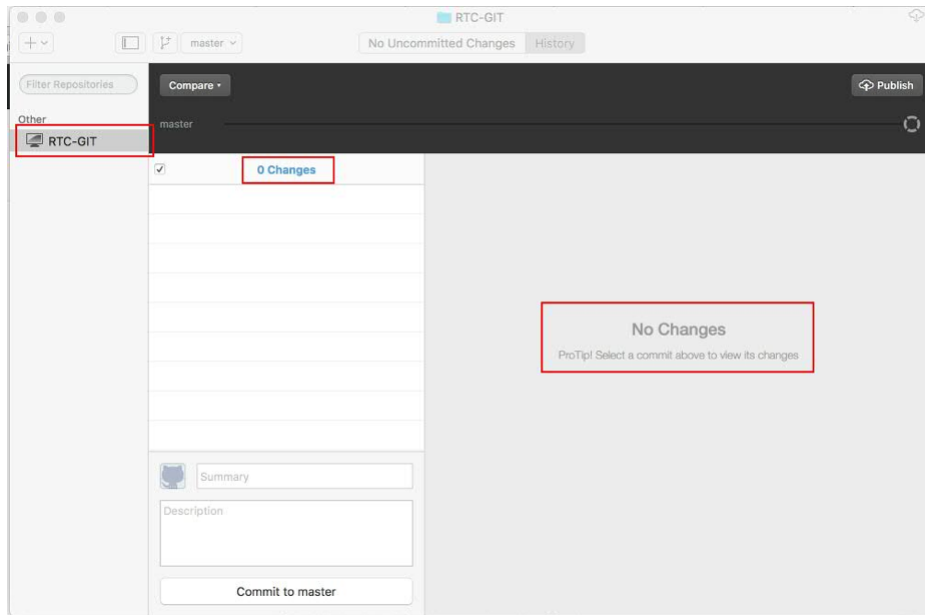
Next, choose where you want to replicate the local version of the repository. In the figure below, we chose GitHub Desktop to replicate the repository with the name RTC-GIT. Enter the name of the repository, select multiple places, then click "Clone".

Figure 6. Enter name and specify Git Desktop



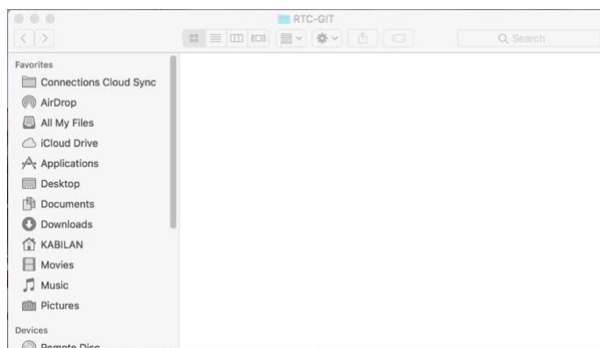
Opening a duplicated GitHub repository shows that the files in this repository are not yet visible.

Figure 7. Git Repository opened in Git Desktop



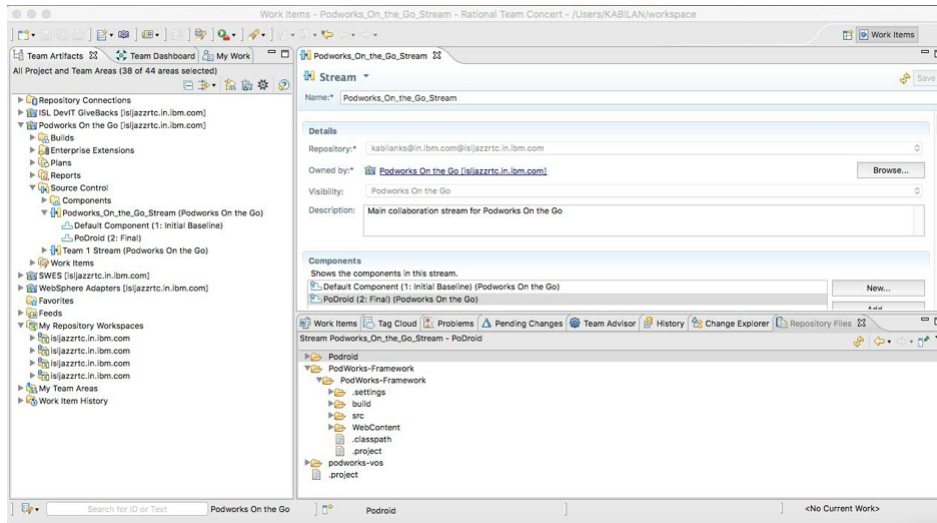
When opening the RTC-GIT folder on GitHub Desktop, it seems that the folder of the GitHub project is empty while the space for the local repository is allocated.

Figure 8. Extracting data Empty space on GitHub Desktop



Open the Rational Team Concert Eclipse client and connect to the Rational Team Concert project area you want to migrate to GitHub. Make sure you have the necessary access rights to perform all operations in the Rational Team Concert project area.

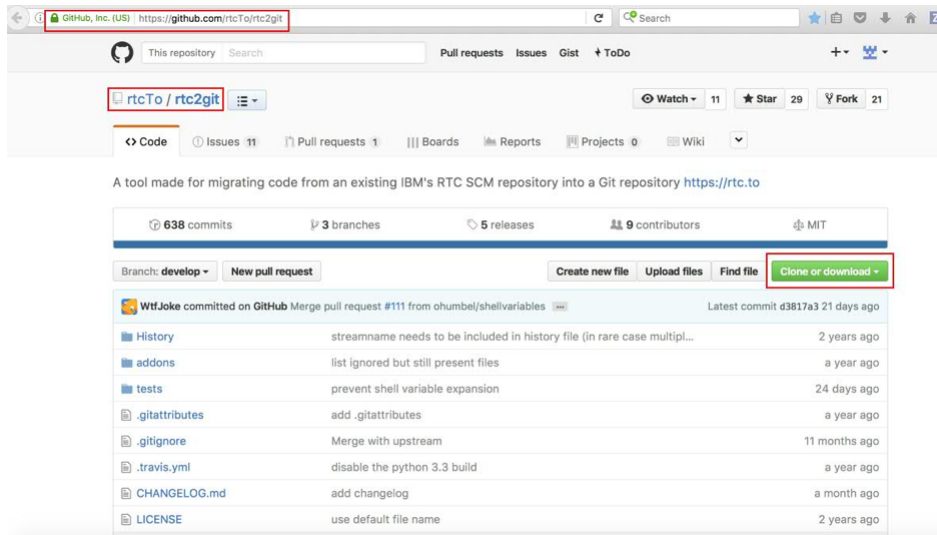
Figure 9. View of source files in Rational Team Concert



Configure the rtc2git package

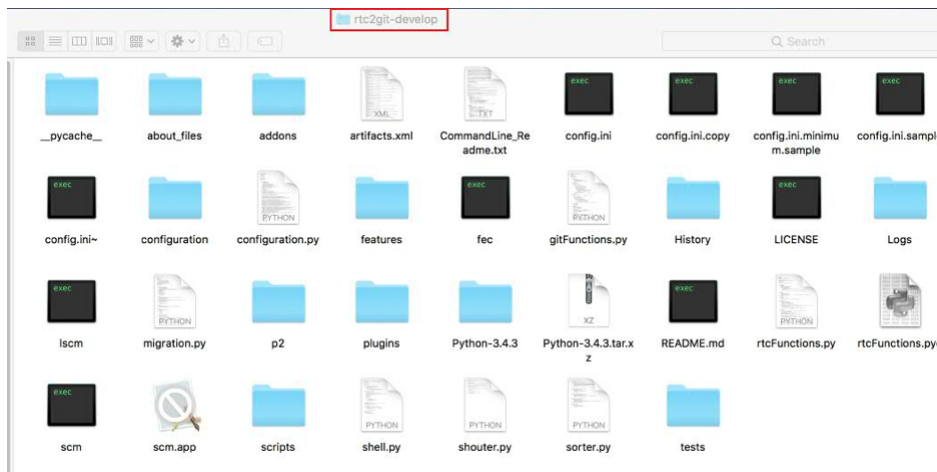
Download rtc2git (see "Prerequisites").

Figure 10. Downloading the rtc2git package



Extract the contents of the downloaded file into the rtc2git-develop folder. The following figure shows the files included in the package.

Figure 11. Folder extracted from rtc2git package



To specify details of this particular migration, open config.ini. The necessary entries in the config.ini file are as follows.

- Repo - The name of the Rational Team Concert repository that contains the data to migrate.
- User - The Rational Team Concert user name of the person making the transition.
- Password - The password for the Rational Team Concert user.

- GIT - Reponame - The name of the GIT repository that will migrate data from Rational Team Concert.
- WorkspaceName - Where to temporarily load the stream (make sure there are no existing workspaces of the same name).
- UseExistingWorkspace - Select this item if you specified the name of an existing workspace as "WorkspaceName" as the workspace to use.
- Directory - The location of the local directory on the machine used to store temporary data during migration.
- Scmcommand - The place you downloaded the Rational Team Concert SCM tool.
- Streamtomigrate - The name of the stream to be migrated from Rational Team Concert to GitHub (make sure the stream name is correct and that stream exists in Rational Team Concert).
- UseProvidedHistory - Select this item if you also want to migrate history along with the source code data.

The following figure shows the config.ini file with the necessary information entered.

Figure 12. Example of a config.ini file with repository details entered

```

config.ini - Edited
[General]
Repo=https://isljazzrtc.in.ibm.com:9551/ccm
User=kabilanks@in.ibm.com
Password=<Password>
GIT-Reponame = RTC-GIT.git
WorkspaceName=Podworks_On_the_Go_Stream Workspace
# If this value is set to True, the workspace referred with workspaceName will just be loaded instead of newly created
useExistingWorkspace = False
# Folder to be created - where migration will take place
Directory = /Users/KABILAN/Desktop/RTC-GIT.git
# Scm command to use (lscm or scm)
ScmCommand = /Users/KABILAN/Desktop/Jazz_5.0.2/scmtools/eclipse/lscm
# Optional - Set encoding of files (For example encoding = UTF-8)
# See "https://github.com/rtcToRtc2git/wiki/Encoding" for further instructions
encoding =

[Migration]
# Stream to be migrated, referenced by Name or UUID
StreamToMigrate = Podworks_On_the_Go_Stream

# Optional - Used by author for prepare a workspace for migrating an earlier release-stream to a later one
# (eg. When Migrating Stream_Version2, Previous-Stream would be Stream_Version1)
PreviousStream=

# Optional, can be defined additionally to set the workspace to a specific baseline
# Use following format: ComponentName = BaselineName, AnotherComponentName=BaselineName
# If its not set, it will determine the oldest baseline (takes some time, depending of how much components you have in your stream)
InitialBaselines =

# False - Rely on order of changeset provided by the rtc cli compare command (due wrong order, more likely to cause merge-conflicts)
# True - When migrating stream, use the order of changesets from the stream
UseProvidedHistory = False

# Determines whether to prompt the user to accept change sets together to resolve accept errors.
# False - The user is prompted
# True - The user is not prompted
UseAutomaticConflictResolution = False

# The max number of change sets to accept together in order to resolve a merge conflict.
MaxChangeSetsToAcceptTogether = 10

# Optional, defines a prefix for commit-messages, which were linked to an rtc workitem.
# In case you have migrate your workitems to another issue system (like jira/github/bitbucket) by using rtc2jira,
# you can define a prefix for the commit-message, in order that previously linked rtc commits get linked to the new issue system.
#
# Examples:
# In jira: AP- (Project has the key AP and is followed by a dash)
# see https://confluence.atlassian.com/jiracloud/processing-jira-issues-with-commit-messages-740098538.html for more details)
# On bitbucket/github: #
CommitMessageWorkItemPrefix=

```

Migrate from IBM Rational Team Concert to GitHub

Once you have preconfigured the config.ini file, migrate the Rational Team Concert source code data to GitHub along with its history. To start the migration, run the migration.py script. Note: Before executing the command, make sure that Python installed is version 3.4 or later. The migration is started by the Python 3 migration.py script. The following figure shows the commands executed by this script. If an error occurs while executing the script, please make sure all prerequisites are satisfied. Common errors and solutions are discussed at the end of this tutorial.

Figure 13. Commands to handle migration

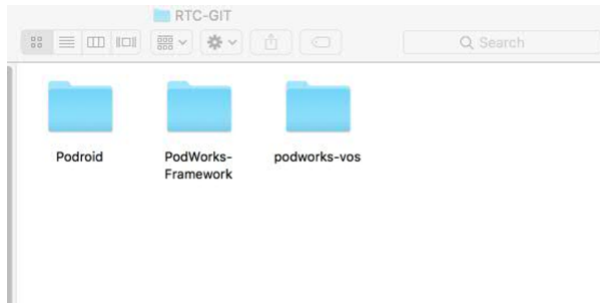
```

bash-3.2$ python3 migration.py
refs/heads/Podworks_On_the_Go_Stream
refs/heads/Podworks_On_the_Go_Stream_branchpoint
16:47:15 - Migration will take place in /Users/KABILAN/Desktop/RTC-GIT.git
Initialized empty Git repository in /Users/KABILAN/Desktop/RTC-GIT.git/RTC-GIT.git/
16:47:15 - Repository was created in /Users/KABILAN/Desktop/RTC-GIT.git
Cloning into 'RTC-GIT'...
warning: You appear to have cloned an empty repository.
done.
16:47:15 - Set core.ignorecase to false
Logged in to https://isljazzrtc.in.ibm.com:9551/ccm
16:47:16 - get UUID of configured stream Podworks_On_the_Go_Stream
Workspace (1811) "Podworks_On_the_Go_Stream Workspace" successfully created
16:47:19 - Determine initial baseline of Default Component
/bin/sh: scm: command not found
16:47:19 - Command 'scm --show-alias n --show-uuid y list baselines --components 40KAPM0EeCq3Z7Rkz45A -r https://isljazzrtc.in.ibm.com:9551/ccm -u kabilanks@in.ibm.com -P
[Error] - 20000' returned non-zero exit status 127
16:47:19 - Determine initial baseline of Podroid
/bin/sh: scm: command not found
16:47:19 - Command 'scm --show-alias n --show-uuid y list baselines --components _Z15MKLBEEcCa-rq6s9K5w -r https://isljazzrtc.in.ibm.com:9551/ccm -u kabilanks@in.ibm.com -P
[Error] - 20000' returned non-zero exit status 127
16:47:19 - Set component 'Default Component' (_40KAPM0EeCq3Z7Rkz45A) to baseline 'Initial Baseline' (_48IacPM0EeCq3Z7Rkz45A)
16:47:20 - Set component 'Podroid'[_Z15MKLBEEcCa-rq6s9K5w] to baseline 'Final' [_1xSuccLBEEcCa-rq6s9K5w]
16:47:22 - Start (re)loading current workspace: /Users/KABILAN/Desktop/Jazz_5.0.2/scmtools/eclipse/lscm load -r https://isljazzrtc.in.ibm.com:9551/ccm 'Podworks_On_the_Go_Stream W
orkspace' --force
Problem running 'load':
Exception during loadError closing metadata store
16:48:10 - Load of workspace finished
16:48:10 - Push of branch master
Counting objects: 3, done.
Writing objects: 100% (3/3), 249 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /Users/KABILAN/Desktop/RTC-GIT.git/RTC-GIT.git
* [new branch] master -> master
16:48:11 - Set new flowtargets
The flow target was updated.
16:48:13 - Warning - UseProvidedHistory is set to false, merge-conflicts are more likely to happen.
For more information see https://github.com/rtcToRtc2git/wiki/Getting-your-History-Files
16:48:13 - Start collecting changeentries
16:48:13 - Collect changes until baseline Initial Baseline of component Default Component
16:48:14 - Collect changes until baseline Final of component Podroid
Switched to a new branch 'Podworks_On_the_Go_Stream'
16:48:14 - Start collecting changes since baseline creation
11/07/16 16:48:15 - Start accepting 0 changesets
Successfully logged out of https://isljazzrtc.in.ibm.com:9551/ccm/
16:48:15 -
All changes accepted - Migration of stream 'Podworks_On_the_Go_Stream' is completed.
You can distribute the git-repo 'RTC-GIT.git'.
bash-3.2$

```

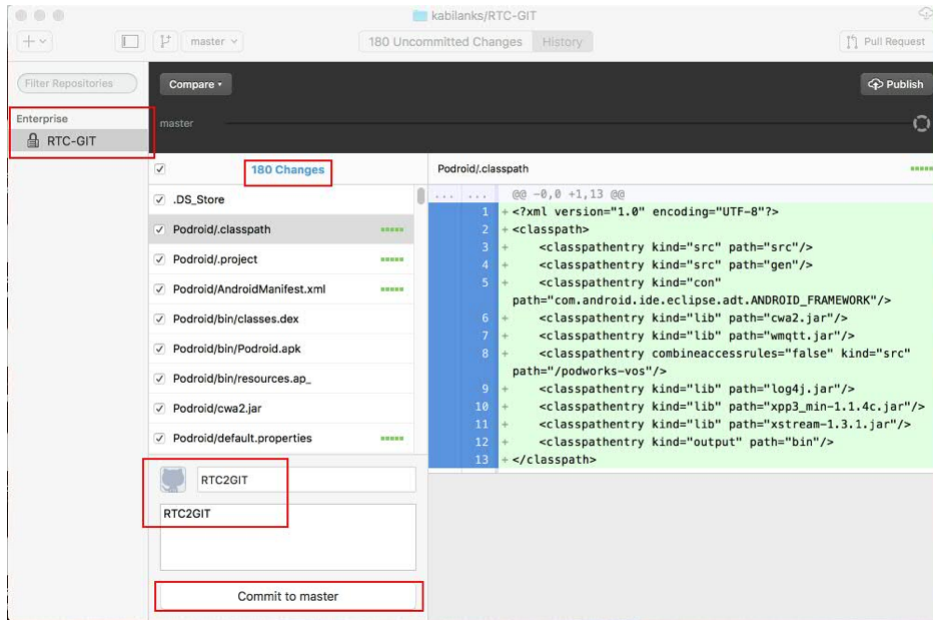
Once the migration script has been executed, the source code data should be stored in the local workspace created as the data storage destination on GitHub Desktop. The storage location of that state is shown below.

Figure 14. Where migration source files are stored on GitHub Desktop



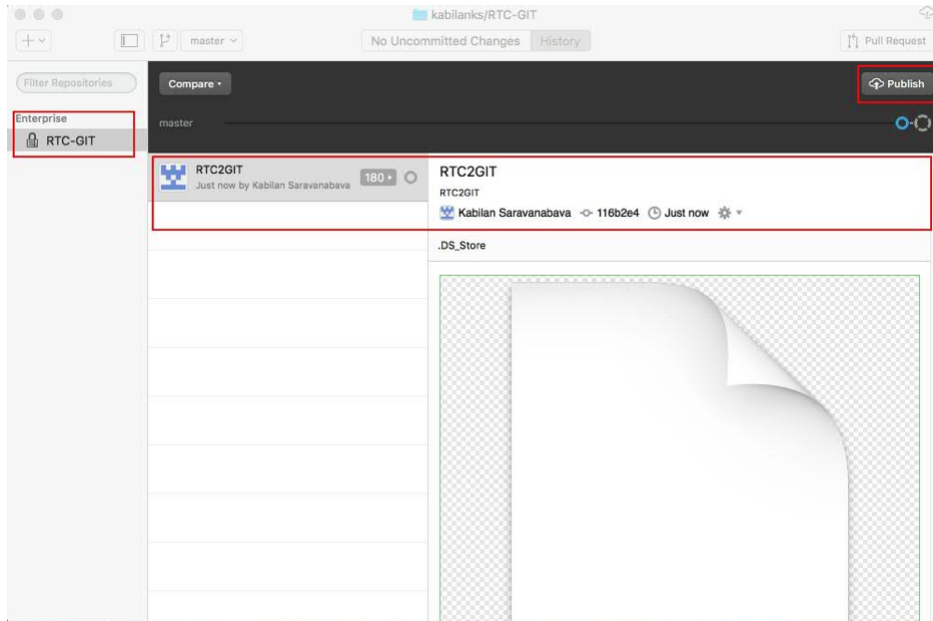
Open the GitHub Desktop client and check the files and folders that the rtc2git migration script pulled from Rational Team Concert.

Figure 15. Source file on Git Desktop and "Commit to master" button



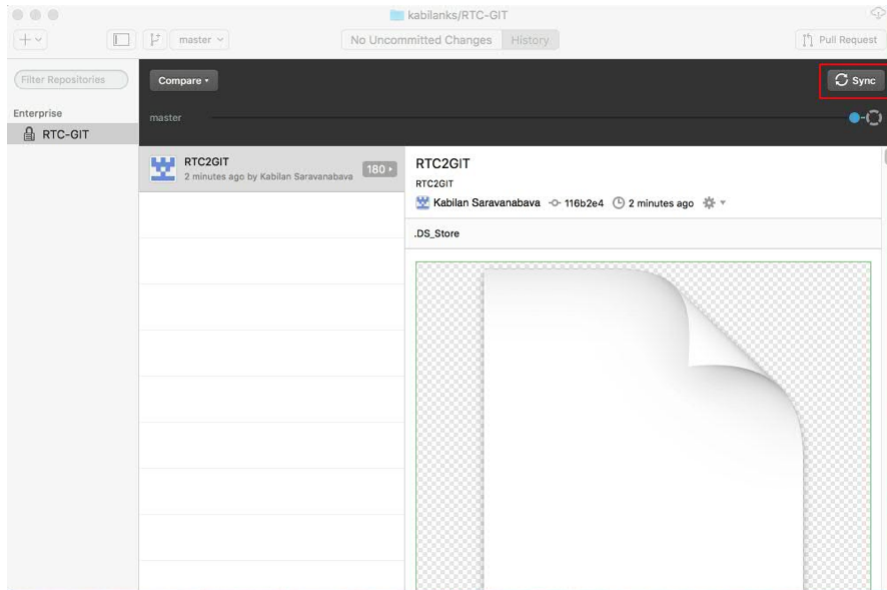
In the GitHub Desktop client, enter the comment you want to add to the data in the text field, then click the "Commit to master" button. This puts the data in a committed state and is ready to be published to the GitHub server. Click "Publish" in the upper right corner of GitHub Desktop .

Figure 16. Publishing source files to GitHub



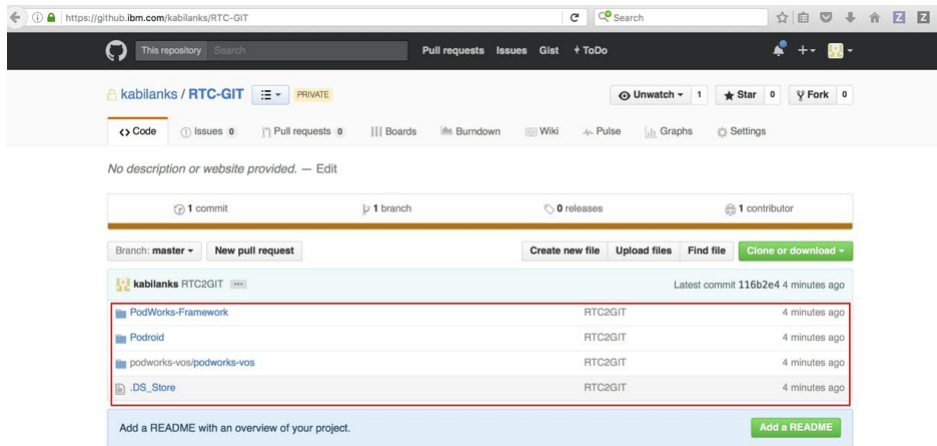
When the source code data is published to the GitHub server, the label of the "Publish" button changes to "Sync". If you add files or folders, you can use these buttons to synchronize those files / folders with the GitHub server.

Figure 17. Synchronizing Git Desktop with GitHub



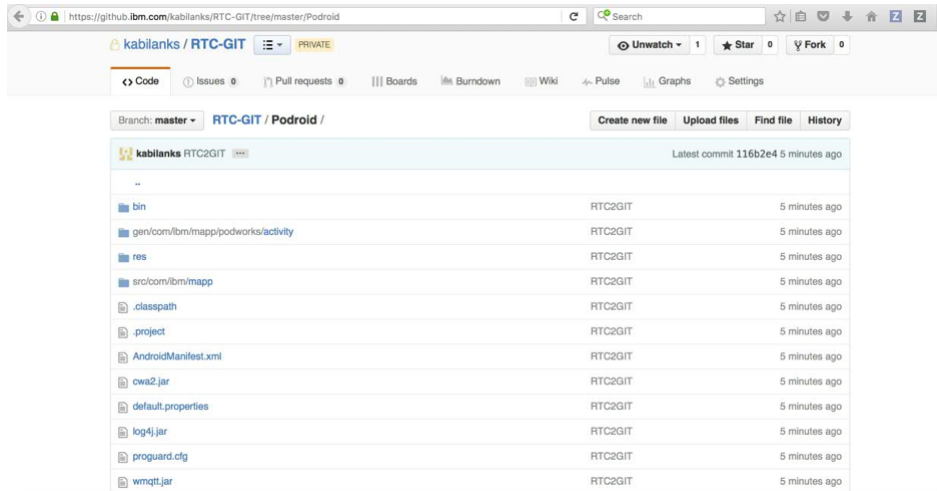
Update the page to display the source code data on the migrated GitHub. The repository shown in the following figure lists all files and folders migrated from Rational Team Concert to GitHub.

Figure 18. GitHub display



Open the folder and make sure that all files and folders are migrated without errors. After you verify that the data is complete, next, look at the logs created during the execution of the migration.py script. This step is to confirm that no errors or warnings are issued during the execution of the migration activity.

Figure 19. Files and folders migrated from IBM Rational Team Concert to GitHub

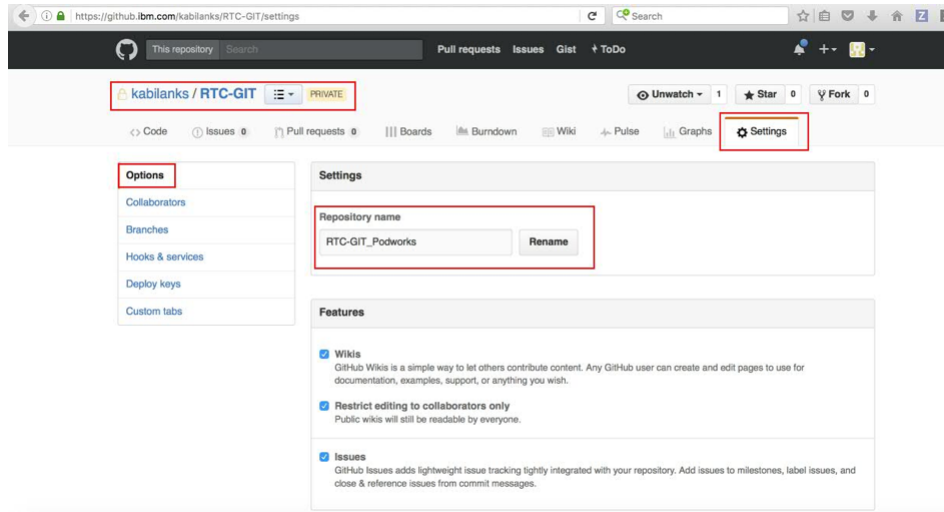


Set access authority for GitHub

After transferring data from Rational Team Concert to GitHub, you can change the name of the GitHub repository or project. If you want to change the name of the GitHub repository, display the "Settings" tab in the repository and change the name set in

the "Repository name" text field.

Figure 20. Changing the GitHub repository name



On the "Settings" page, you can also add collaborators to the GitHub repository. In that case, search for the name of the user you want to add as a collaborator. In order to retrieve the user name in the search field, the user must have logged in to GitHub at least once.

Figure 21. Adding a collaborator to the GitHub repository

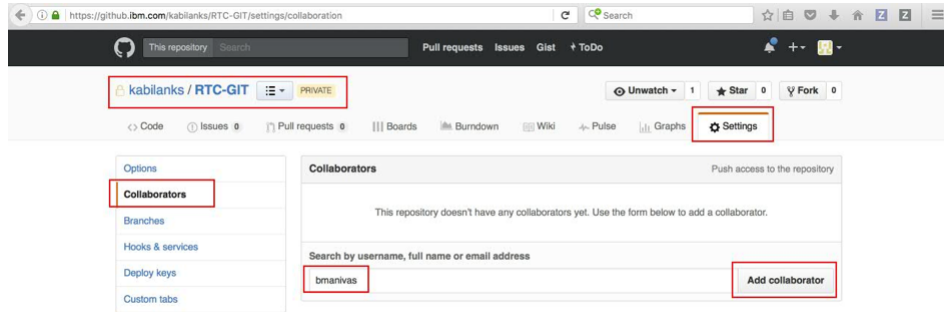
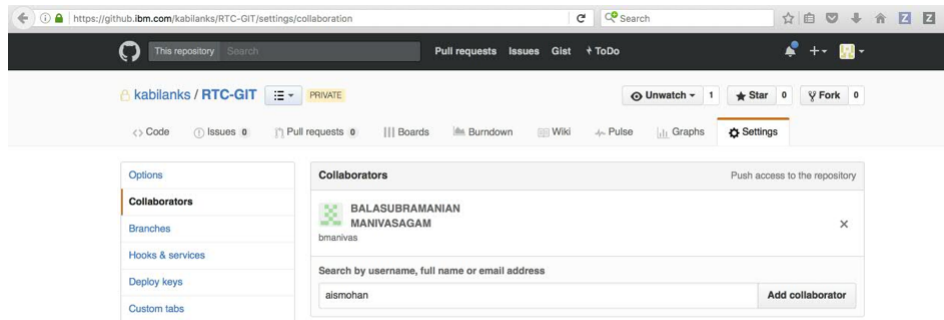
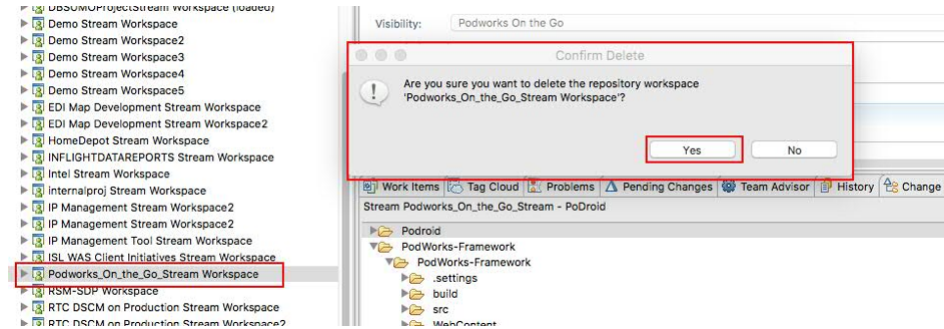


Figure 22. Adding multiple collaborators to the GitHub repository



After adding the collaborator to the GitHub repository, release the project so that the collaborator can start working. This will send e-mails with access information to the repository to users added to the GitHub repository as collaborators.

Figure 23. Adding multiple collaborators to the GitHub repository



Common Errors and Solutions

When migrating a stream to an existing workspace, enter the exact same workspace name in the config.ini file. If the names do not match or add a workspace name, an error will occur while running the migration.py script. The solution to this problem is to delete the existing workspace and rerun the migration script. By doing so, the migration activity from Rational Team Concert to GitHub will complete successfully.

Figure 24. Deleting workspace

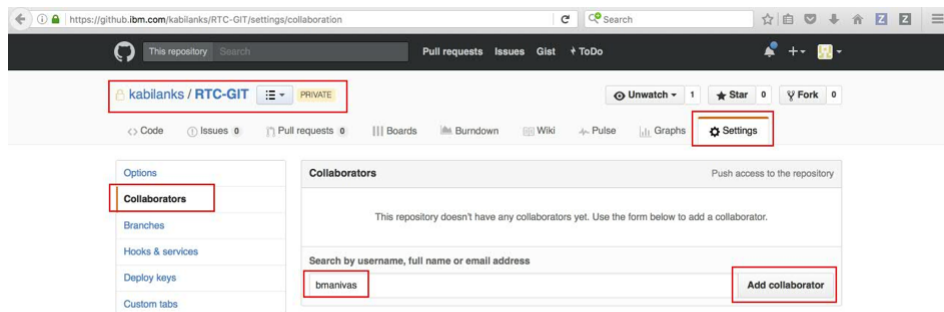
```

bash-3.2$ python3 migration.py
refs/heads/podworks_on_the_go_stream
refs/heads/podworks_on_the_go_stream_branchpoint
16:39:10 - Migration will take place in /Users/KABILAN/Desktop/RTC-GIT.git
Initialized empty Git repository in /Users/KABILAN/Desktop/RTC-GIT.git/RTC-GIT.git/
16:39:10 - Repository was created in /Users/KABILAN/Desktop/RTC-GIT.git
Cloning into 'RTC-GIT'...
warning: You appear to have cloned an empty repository.
done.
16:39:10 - Set core.ignorecase to false
Logged in to https://is1jazrztc.in.ibm.com:9551/ccm
16:39:14 - Get UUID of configured stream Podworks_On_the_Go_Stream
Workspace (1009) "Podworks_On_the_Go_Stream Workspace" successfully created
16:39:18 - Determine initial baseline of Default Component
/bin/sh: scm: command not found
16:39:18 - Command 'scm --show-alias n --show-uid y list baselines --components _48KnAPM8EcQ3Z7Kz45A -r https://is1jazrztc.in.ibm.com:9551/ccm --u kabilanks@in.ibm.com -P
[REDACTED] --m 20000' returned non-zero exit status 127
16:39:18 - Determine initial baseline of PoDroid
/bin/sh: scm: command not found
16:39:18 - Command 'scm --show-alias n --show-uid y list baselines --components _Zt54MKL8EcEa-rq65Hk5w -r https://is1jazrztc.in.ibm.com:9551/ccm --u kabilanks@in.ibm.com -P
[REDACTED] --m 20000' returned non-zero exit status 127
16:39:18 - Set component 'Default Component'({_48KnAPM8EcQ3Z7Kz45A}) to baseline 'Initial Baseline'({_48KnAPM8EcQ3Z7Kz45A})
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(1009) "Podworks_On_the_Go_Stream Workspace"
(1008) "Podworks_On_the_Go_Stream Workspace"
Problem running 'set component':
Ambiguous workspace/stream "Podworks_On_the_Go_Stream Workspace"
16:39:18 - Set component 'PoDroid'({_Zt54MKL8EcEa-rq65Hk5w}) to baseline 'Final'({_1x5ucl8EcEa-rq65Hk5w})
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(1009) "Podworks_On_the_Go_Stream Workspace"
(1008) "Podworks_On_the_Go_Stream Workspace"
Problem running 'set component':
Ambiguous workspace/stream "Podworks_On_the_Go_Stream Workspace"
16:39:18 - Start (re)loading current workspace: /Users/KABILAN/Desktop/Jazz_5.0.2/scntools/eclipse/lscm -r https://is1jazrztc.in.ibm.com:9551/ccm 'Podworks_On_the_Go_Stream W
orkspace' --force
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(1009) "Podworks_On_the_Go_Stream Workspace"
(1008) "Podworks_On_the_Go_Stream Workspace"
Problem running 'load':
Ambiguous workspace "Podworks_On_the_Go_Stream Workspace"
16:39:18 - Load of workspace finished
16:39:18 - Push of branch master
Counting objects: 3, done.
Writing objects: 100% (3/3), 250 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /Users/KABILAN/Desktop/RTC-GIT.git/RTC-GIT.git
 * [new branch] master -> master
16:39:19 - Set new flowtargets
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(_JNsAQKTZEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
(_FpZACKTFEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
Problem running 'list flowtargets':
Ambiguous workspace/stream "Podworks_On_the_Go_Stream Workspace"
16:39:19 - Command '/Users/KABILAN/Desktop/Jazz_5.0.2/scntools/eclipse/lscm --show-uid y --show-alias n list flowtargets -r https://is1jazrztc.in.ibm.com:9551/ccm 'Podworks_On_th
e_Go_Stream Workspace' returned non-zero exit status 9
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(1009) "Podworks_On_the_Go_Stream Workspace"
(1008) "Podworks_On_the_Go_Stream Workspace"
Problem running 'add flowtarget':
Ambiguous workspace/stream "Podworks_On_the_Go_Stream Workspace"
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(1009) "Podworks_On_the_Go_Stream Workspace"
(1008) "Podworks_On_the_Go_Stream Workspace"
Problem running 'set flowtarget':
Ambiguous workspace/stream "Podworks_On_the_Go_Stream Workspace"
16:39:20 - Warning - UseProvidedHistory is set to false, merge-conflicts are more likely to happen.
For more information see https://github.com/rtcTo/rct2git/wiki/Getting-your-History-Files
16:39:20 - Start collecting changeentries
16:39:20 - Collect changes until baseline Initial Baseline of component Default Component
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(_JNsAQKTZEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
(_FpZACKTFEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
Problem running 'compare':
Ambiguous workspace "Podworks_On_the_Go_Stream Workspace"
16:39:20 - Collect changes until baseline Final of component PoDroid
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(_JNsAQKTZEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
(_FpZACKTFEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
Problem running 'compare':
Ambiguous workspace "Podworks_On_the_Go_Stream Workspace"
Switched to a new branch 'Podworks_On_the_Go_Stream'
16:39:20 - Start collecting changes since baseline creation
Ambiguous selector "Podworks_On_the_Go_Stream Workspace" matched multiple items:
Possible matches:
(_JNsAQKTZEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
(_FpZACKTFEe8s4iGdVbVEg) "Podworks_On_the_Go_Stream Workspace"
Problem running 'compare':
Ambiguous workspace "Podworks_On_the_Go_Stream Workspace"
11/07/16 16:39:20 - Start accepting 0 changesets
Successfully logged out of https://is1jazrztc.in.ibm.com:9551/ccm/
16:39:21 -
All changes accepted - Migration of stream 'Podworks_On_the_Go_Stream' is completed.
You can distribute the git-repo 'RTC-GIT.git'.

```

The following figure shows an error that is displayed when an incorrect item is selected in the config.ini file when using an existing workspace.


Figure 25. Error when there is an existing repository workspace in Rational Team Concert



Summary

In this tutorial we provide hints based on the experience of implementing Rational Team Concert and GitHub and give you the hints on migration requirements, how to create a GitHub repository, how to setup it, how to configure it, how to migrate from Rational Team Concert to GitHub, I explained the setting method according to the procedure. Finally, I explained about common problems and their solutions.

Downloadable resources

 [PDF of this content](#)
