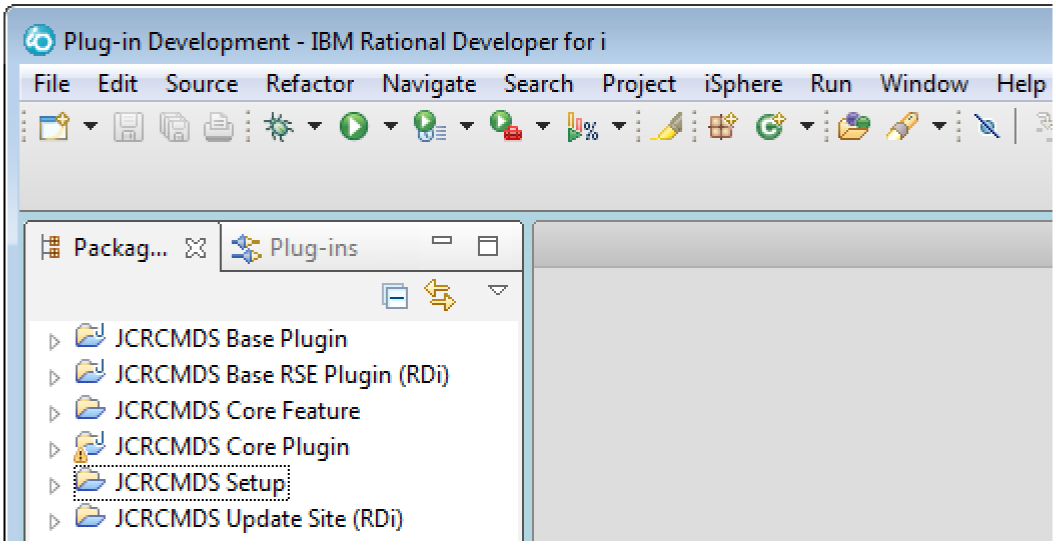
Initial Upload of JCRCMDS to SourceForge

Prerequisites

Installed IBM Rational Developer for i (RDi 9.5) with a dedicated workspace for JCRCMDS and the JCRCMDS plug-ins (Selected perspective: Plug-in Development):



This tutorial uses the following workspace location:

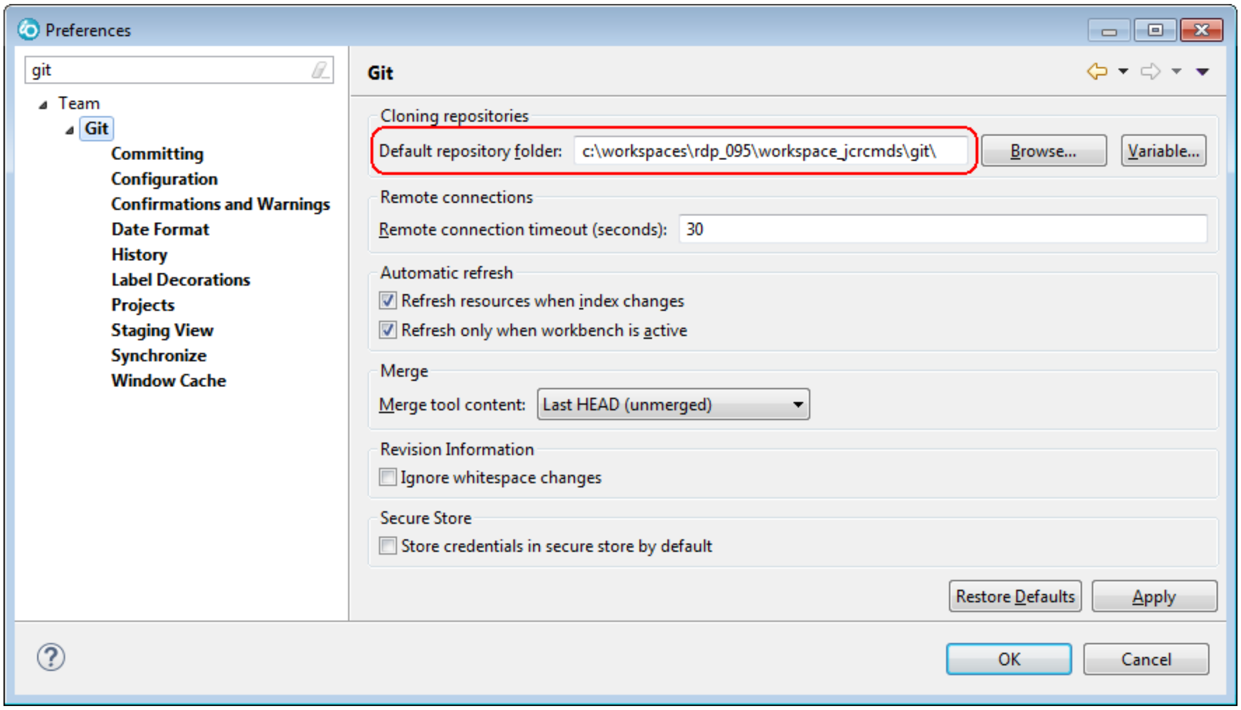
```
c:\workspaces\rdp_095\workspace_jcrcmds\
```

Setting Up Git

Open the Git preferences page and set the following values:

Default Repository Folder

```
c:\workspaces\rdp_095\workspace_jcrcmds\git\
```



Git

Cloning repositories

Default repository folder: c:\workspaces\rdp_095\workspace_jcrcmds\git\

Remote connections

Remote connection timeout (seconds): 30

Automatic refresh

☑ Refresh resources when index changes

☑ Refresh only when workbench is active

Merge

Merge tool content: Last HEAD (unmerged)

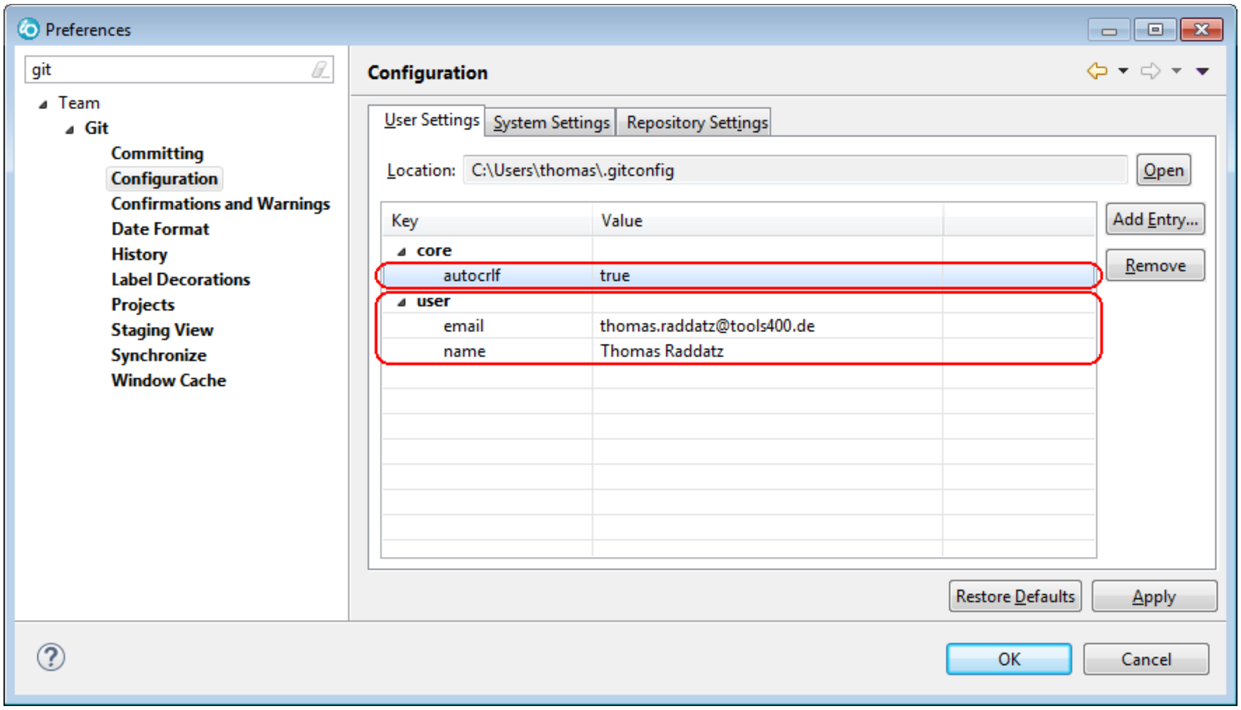
Revision Information

□ Ignore whitespace changes

Secure Store

□ Store credentials in secure store by default

CRLF Handling And User



Configuration

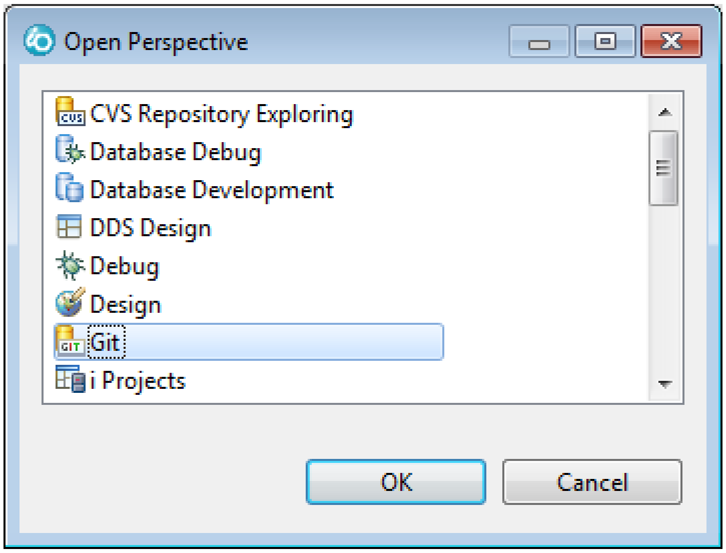
Location: C:\Users\thomas\.gitconfig

Key	Value
core	
autocrlf	true
user	
email	thomas.raddatz@tools400.de
name	Thomas Raddatz

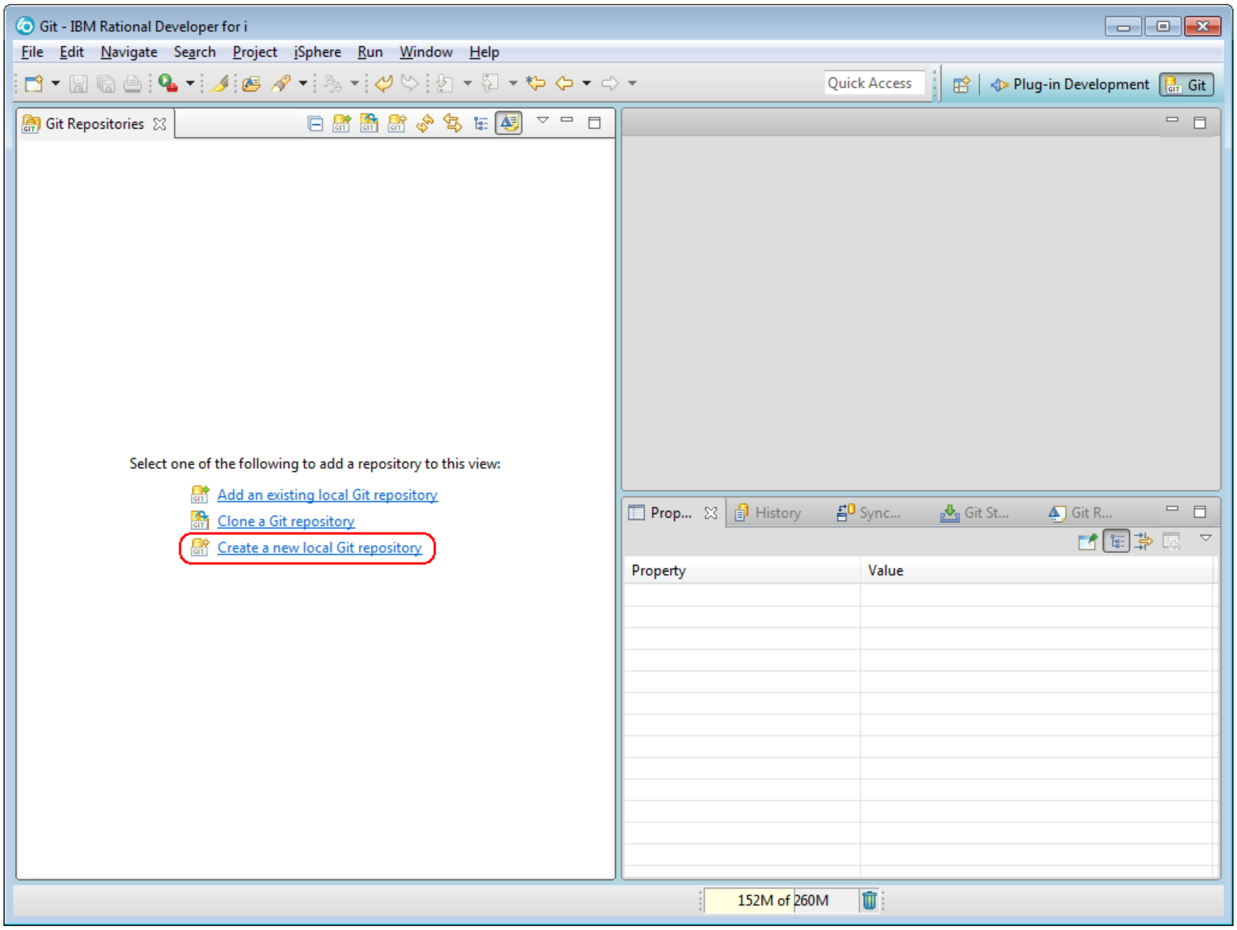
Of course you must not use my email address and name, but your own address and name.

Creating a Local Git repository

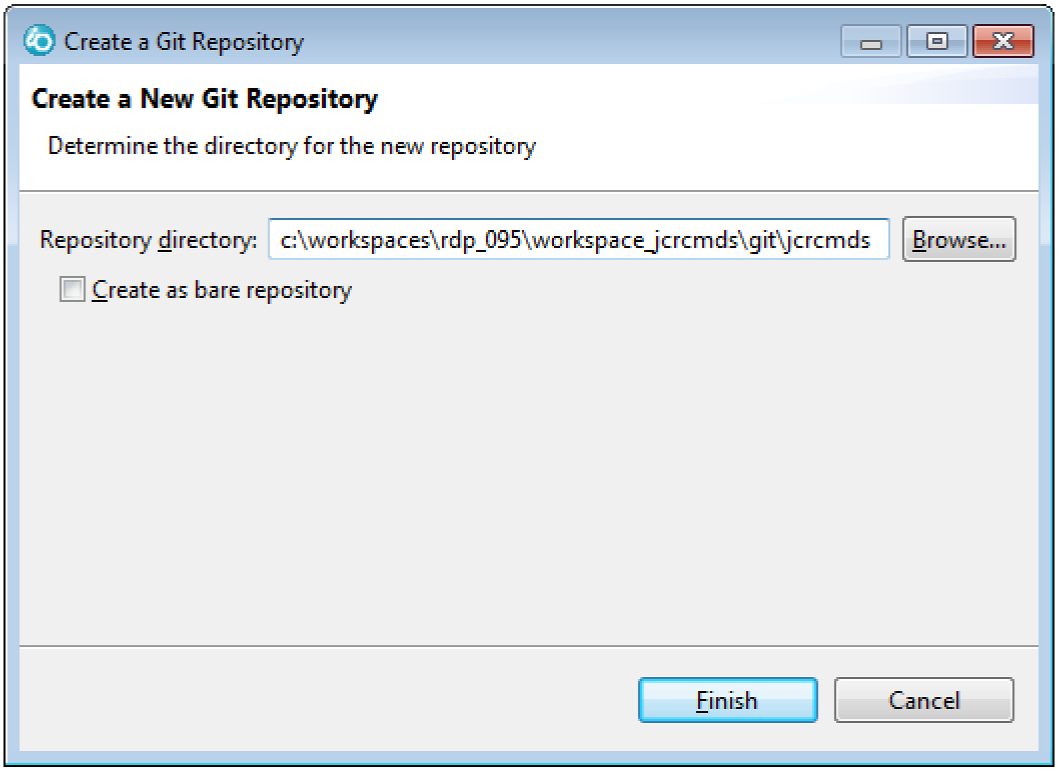
Select the "Git" perspective to create a local Git repository for JCRCMDS:



On the Git perspective select "Create a new local Git repository":



Enter the path of the new directory. In this example we create a new folder "jcrcmds" below the root Git path:



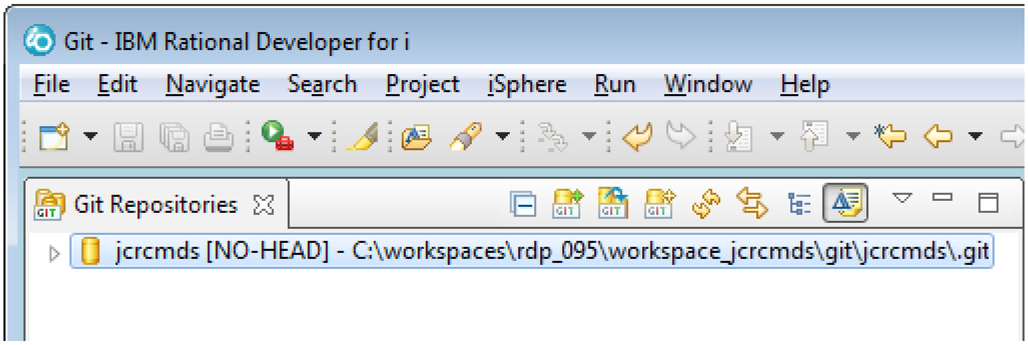
Create a New Git Repository

Determine the directory for the new repository

Repository directory: c:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds

□ Create as bare repository

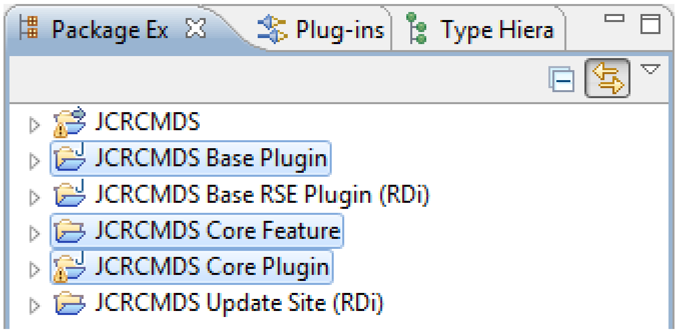
You should get the following repository entry in the Git perspective:


Git - IBM Rational Developer for i
File Edit Navigate Search Project iSphere Run Window Help
Git Repositories

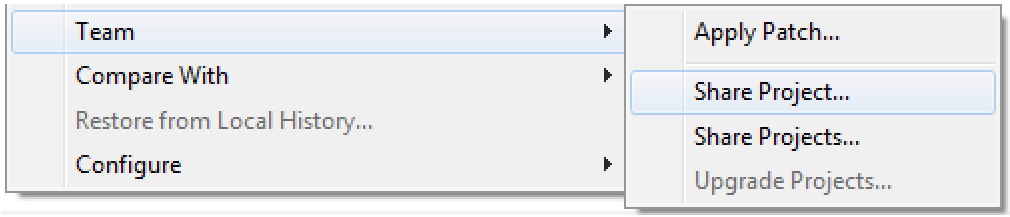
jcrcmds [NO-HEAD] - C:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\.git

Adding Projects to Git

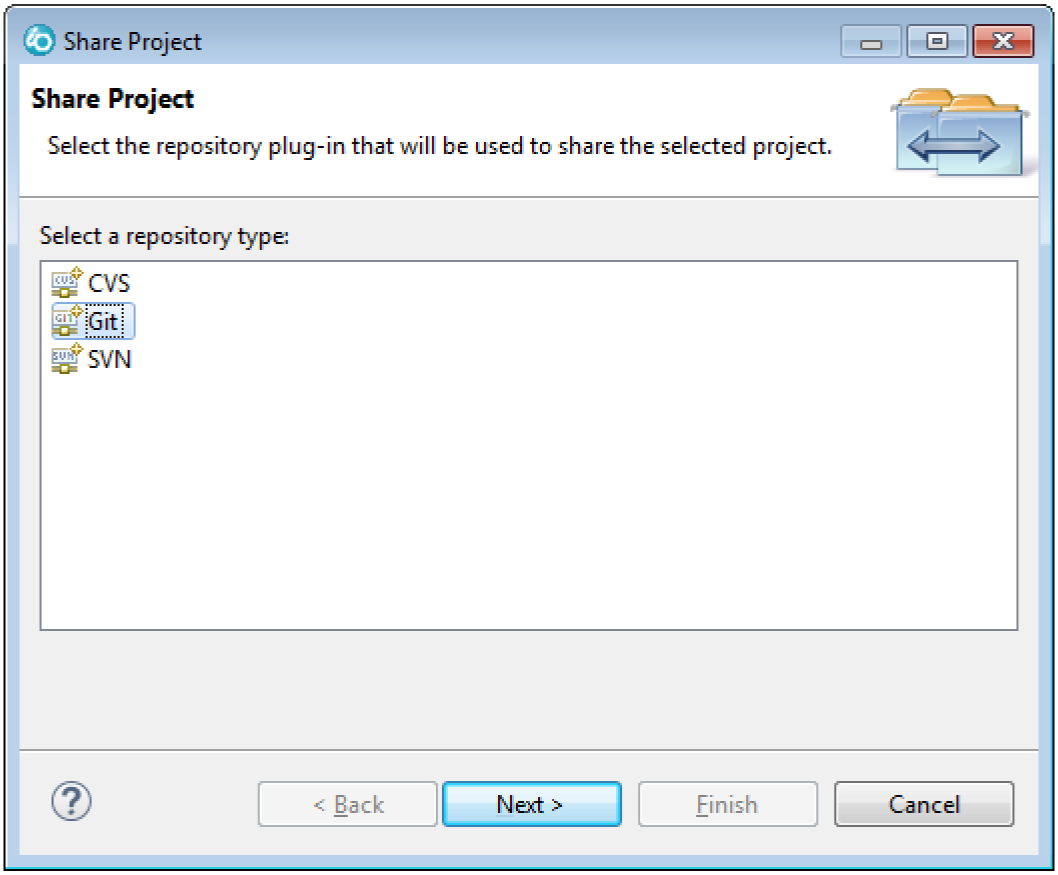
Select the JCRCMDS projects that go into the "eclipse" sub-path of your Git repository. We want to separate the plug-ins that are common for RDi and WDSCi from the plug-ins that are specific to the IBM IDEs:



Open the context menu and select "Team - Share Project...":



Then continue with "Git":

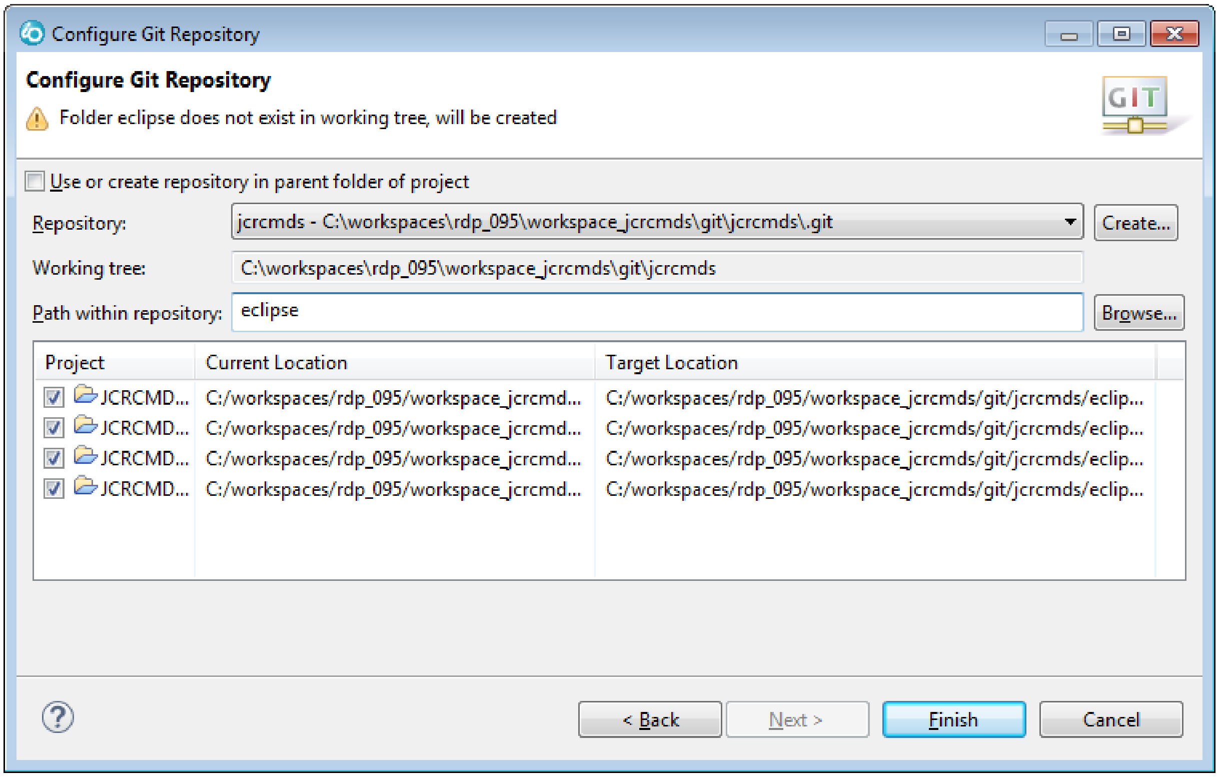

Share Project

Share Project

Select the repository plug-in that will be used to share the selected project.

Select a repository type:

Select the "jcrcmds" repository and set the "Path within repository" to "eclipse":



Configure Git Repository

Configure Git Repository

⚠ Folder eclipse does not exist in working tree, will be created

☐ Use or create repository in parent folder of project

Repository: jcrcmds - C:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\.git

Working tree: C:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds

Path within repository: eclipse

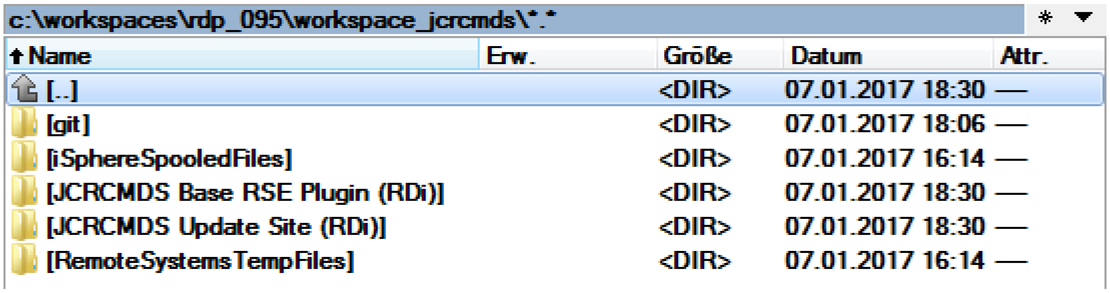
Project	Current Location	Target Location
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/eclip...
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/eclip...
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/eclip...
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/eclip...

Click the [Finish] button.

The question marks next to the projects indicate that the projects have been added to Git and that there are uncommitted changes.

The projects have been moved from the workspace to the git repository:

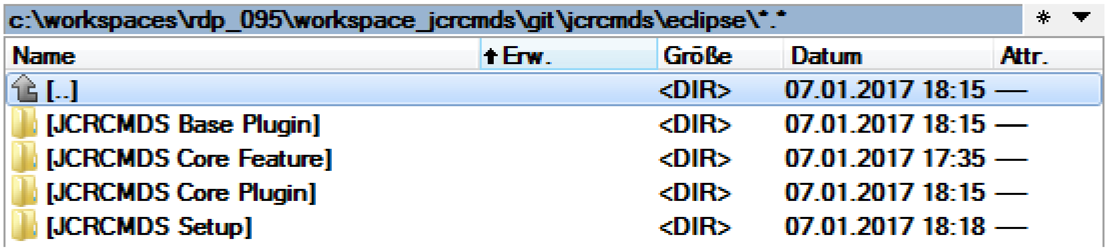
Workspace



c:\workspaces\rdp_095\workspace_jcrcmds\*.*

Name	Erw.	Größe	Datum	Attr.
[..]		<DIR>	07.01.2017 18:30	—
[git]		<DIR>	07.01.2017 18:06	—
[iSphereSpooledFiles]		<DIR>	07.01.2017 16:14	—
[JCRCMDS Base RSE Plugin (RDi)]		<DIR>	07.01.2017 18:30	—
[JCRCMDS Update Site (RDi)]		<DIR>	07.01.2017 18:30	—
[RemoteSystemsTempFiles]		<DIR>	07.01.2017 16:14	—

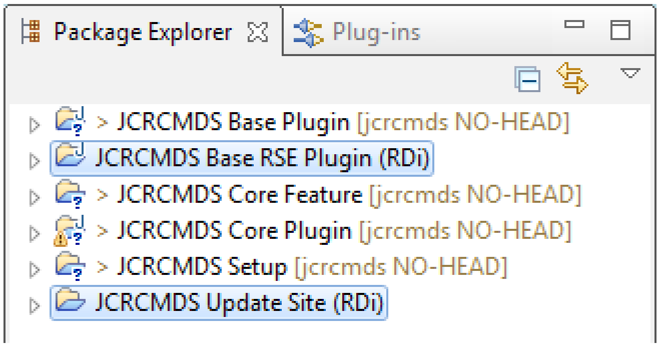
Local Git Repository



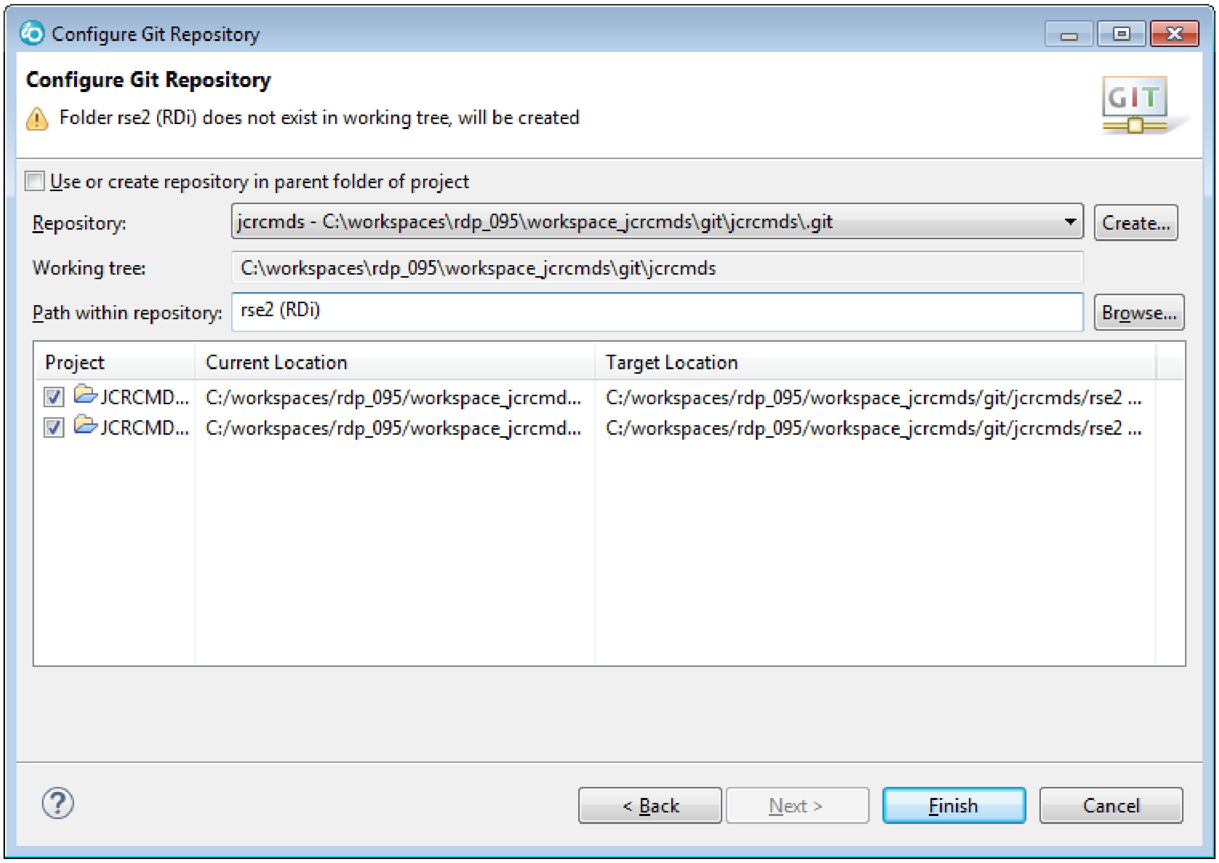
c:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\eclipse\*.*

Name	↑Erw.	Größe	Datum	Attr.
[..]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Base Plugin]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Core Feature]		<DIR>	07.01.2017 17:35	—
[JCRCMDS Core Plugin]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Setup]		<DIR>	07.01.2017 18:18	—

Go ahead and add the remaining IBM IDE specific projects to sub-folder "rse2 (RDi)". The WDSCi plug-in will go into folder "rse1 (WDSCi)":



- > JCRCMDS Base Plugin [jcrcmds NO-HEAD]
- JCRCMDS Base RSE Plugin (RDi)
- > JCRCMDS Core Feature [jcrcmds NO-HEAD]
- > JCRCMDS Core Plugin [jcrcmds NO-HEAD]
- > JCRCMDS Setup [jcrcmds NO-HEAD]
- JCRCMDS Update Site (RDi)



Configure Git Repository

Configure Git Repository

Folder rse2 (RDi) does not exist in working tree, will be created

☐ Use or create repository in parent folder of project

Repository: jcrcmds - C:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\.git

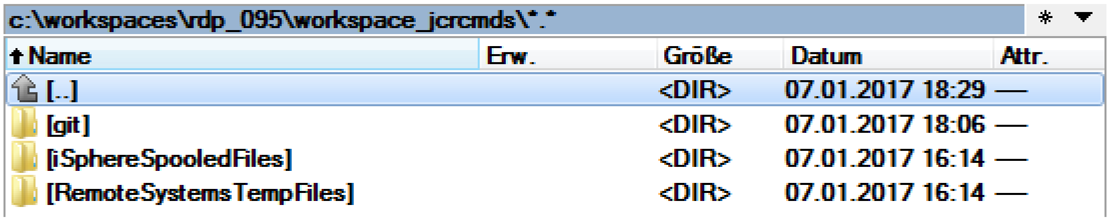
Working tree: C:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds

Path within repository: rse2 (RDi)

Project	Current Location	Target Location
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/rse2 ...
☑ JCRCMD...	C:/workspaces/rdp_095/workspace_jcrcmd...	C:/workspaces/rdp_095/workspace_jcrcmds/git/jcrcmds/rse2 ...

Now all projects are in the Git repository and the workspace is empty again:

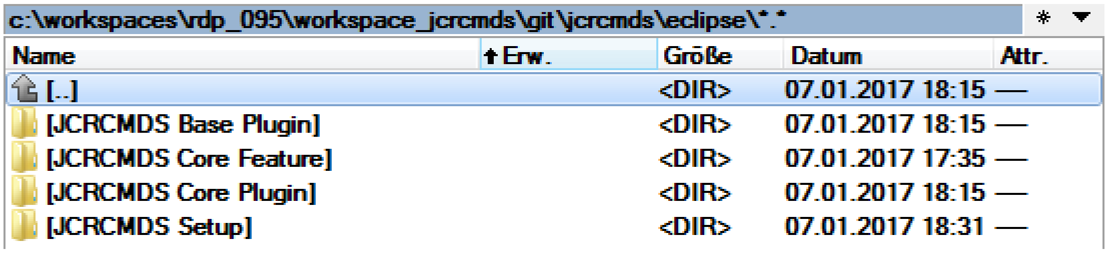
Workspace



c:\workspaces\rdp_095\workspace_jcrcmds\*.*

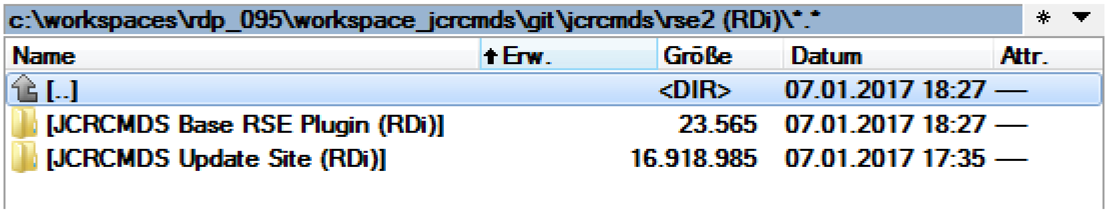
Name	Erw.	Größe	Datum	Attr.
[..]		<DIR>	07.01.2017 18:29	—
[git]		<DIR>	07.01.2017 18:06	—
[iSphereSpooledFiles]		<DIR>	07.01.2017 16:14	—
[RemoteSystemsTempFiles]		<DIR>	07.01.2017 16:14	—

Local Git Repository



c:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\eclipse\*.*

Name	Erw.	Größe	Datum	Attr.
[..]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Base Plugin]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Core Feature]		<DIR>	07.01.2017 17:35	—
[JCRCMDS Core Plugin]		<DIR>	07.01.2017 18:15	—
[JCRCMDS Setup]		<DIR>	07.01.2017 18:31	—



c:\workspaces\rdp_095\workspace_jcrcmds\git\jcrcmds\rse2 (RDi)*.*

Name	Erw.	Größe	Datum	Attr.
[..]		<DIR>	07.01.2017 18:27	—
[JCRCMDS Base RSE Plugin (RDi)]		23.565	07.01.2017 18:27	—
[JCRCMDS Update Site (RDi)]		16.918.985	07.01.2017 17:35	—

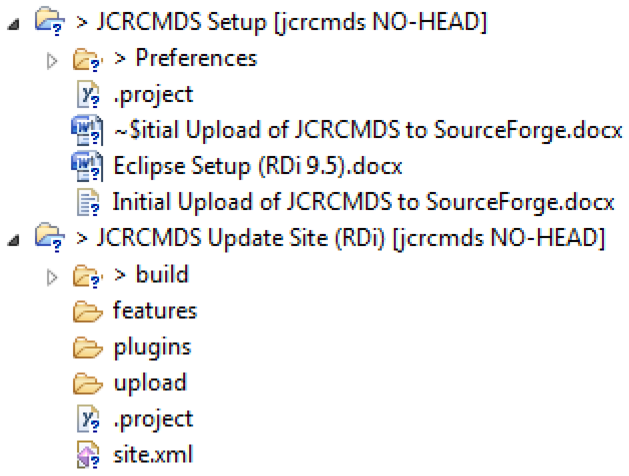
Preparing The Initial Commit

Before we can commit our projects, we need to add a few items to the "ignore list", because we do not want these items go into the repository. These items are temporary work items:

- MS Word temporary files
- Everything in the directories "features", "plugins" and "upload"

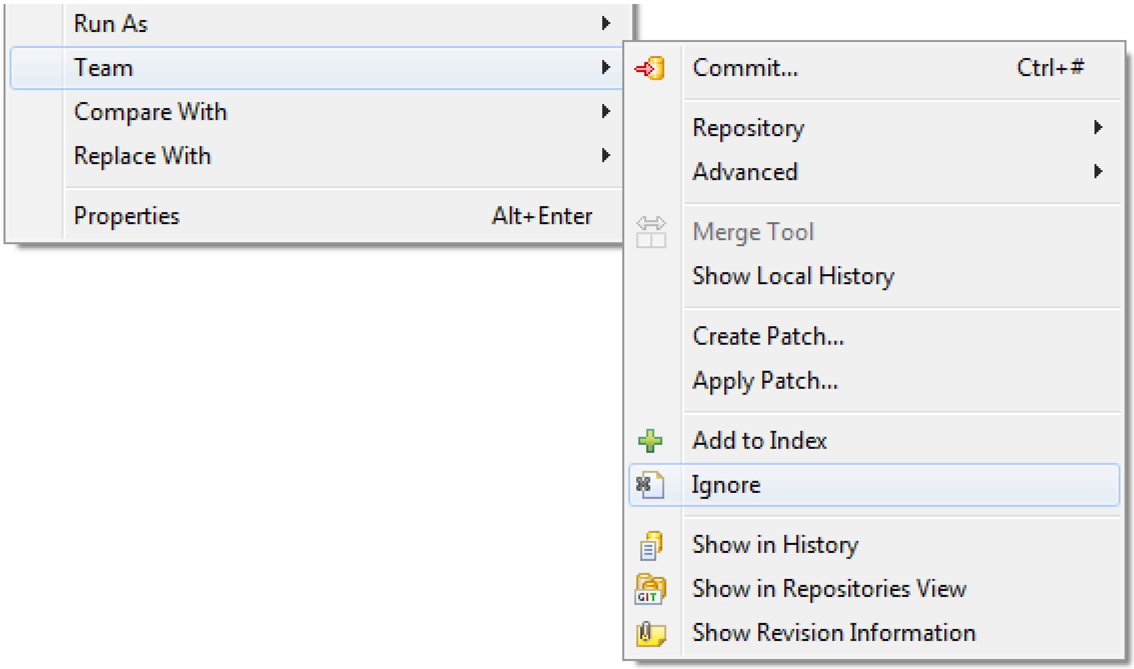
The MS Word file is there, because I write that tutorial while I practice it. Most likely you do not have temporary files when you are about for the first commit.

Let us start with the MS Word temporary files, which start with a tilde (~):

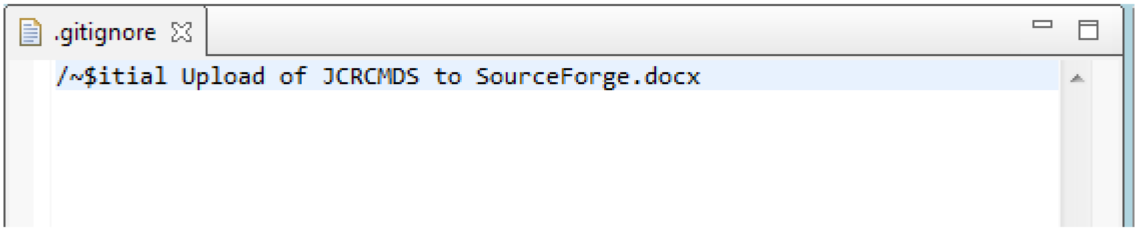


```
◢ > JCRCMDS Setup [jcrcmds NO-HEAD]
    ▷ > Preferences
      .project
      ~$itial Upload of JCRCMDS to SourceForge.docx
      Eclipse Setup (RDi 9.5).docx
      Initial Upload of JCRCMDS to SourceForge.docx
◢ > JCRCMDS Update Site (RDi) [jcrcmds NO-HEAD]
    ▷ > build
      features
      plugins
      upload
      .project
      site.xml
```

Select that file and add it to the "ignore list":

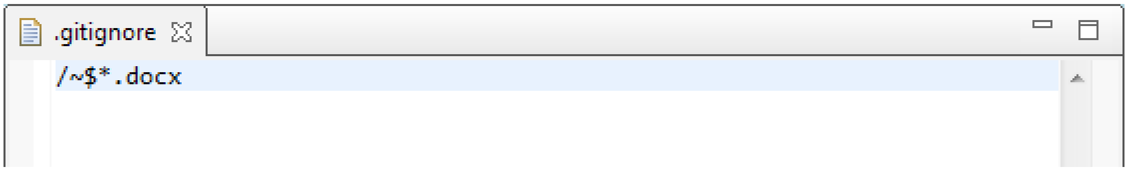


Now open the "Navigator" view and open the ".gitignore" file of project "JCRCMDS Setup":



```
/~$itial Upload of JCRCMDS to SourceForge.docx
```

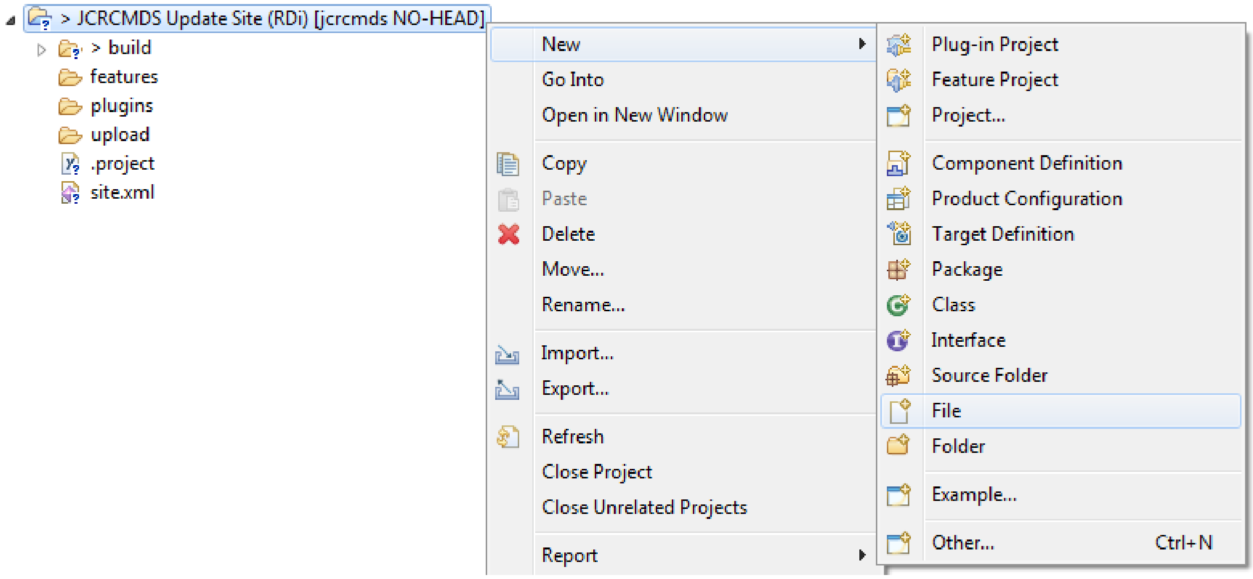
Change the item to a generic name like this:



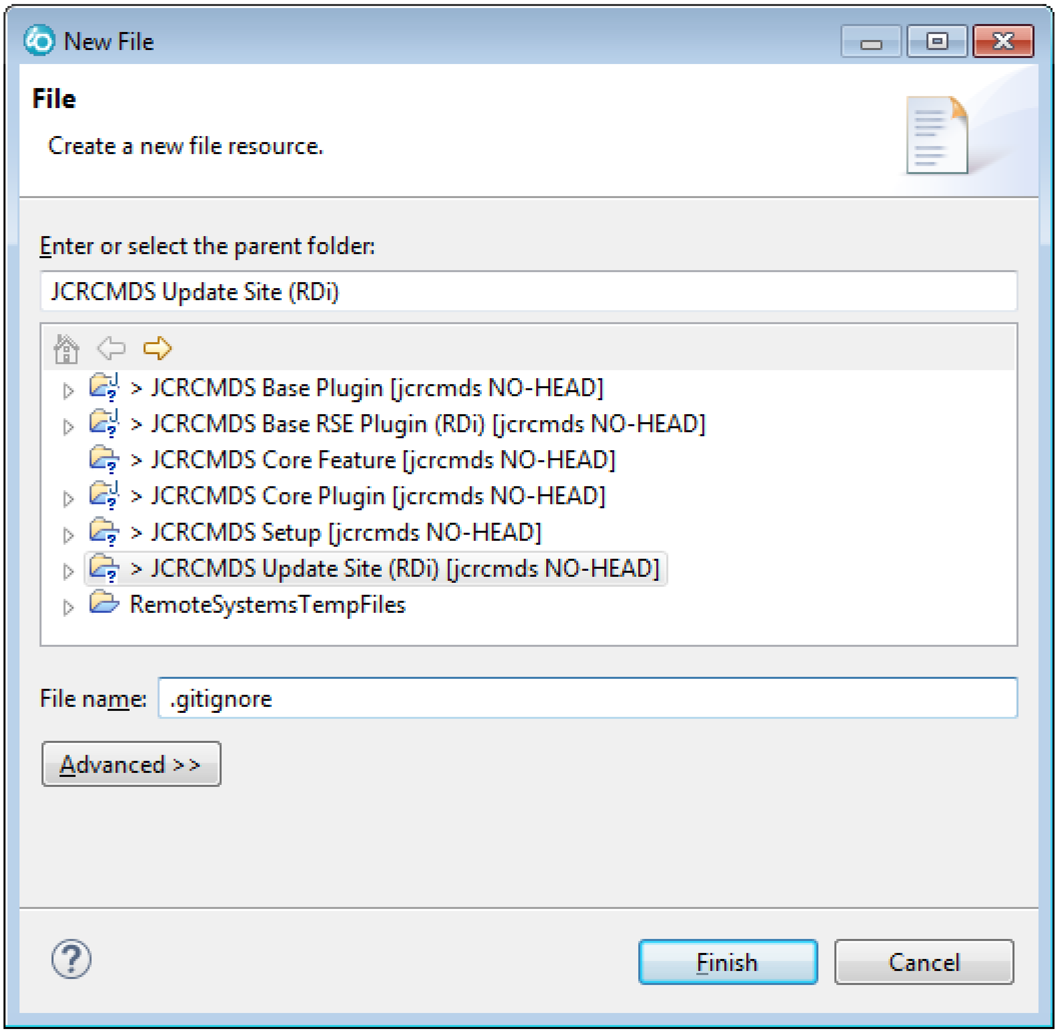
```
/~$*.docx
```

This way all MS Word temporary files are ignored.

Now go ahead with the former mentioned directories of the "JCRCMDS Update Site" project. For this project a ".gitignore" file has not yet been created. Therefore you should create it yourself. Right click the project and select "New - File":



Enter ".gitignore" and click the [OK] button:



New File

File

Create a new file resource.

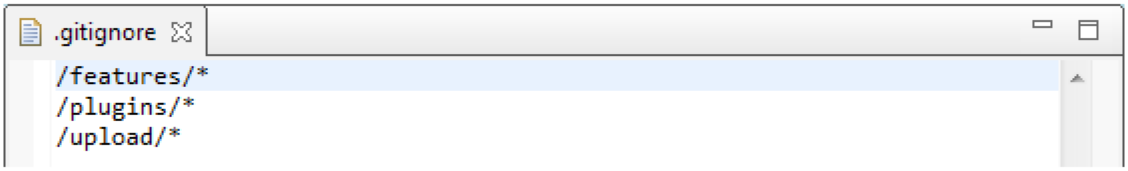
Enter or select the parent folder:

JCRCMDS Update Site (RDi)

- > JCRCMDS Base Plugin [jcrcmds NO-HEAD]
- > JCRCMDS Base RSE Plugin (RDi) [jcrcmds NO-HEAD]
- > JCRCMDS Core Feature [jcrcmds NO-HEAD]
- > JCRCMDS Core Plugin [jcrcmds NO-HEAD]
- > JCRCMDS Setup [jcrcmds NO-HEAD]
- > JCRCMDS Update Site (RDi) [jcrcmds NO-HEAD]
- RemoteSystemsTempFiles

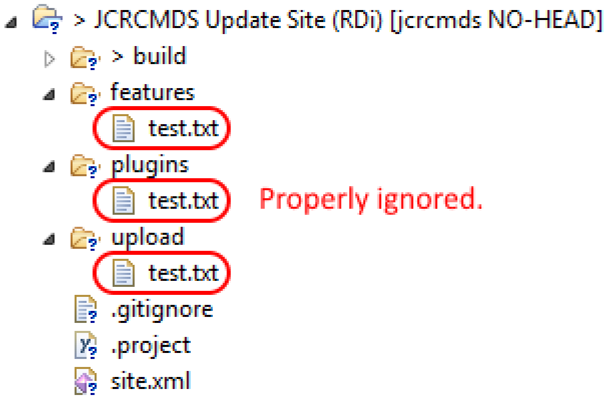
File name: .gitignore

Open the file and add the following entries:



```
/features/*
/plugins/*
/upload/*
```

Now feel free to add some test files to these folders to see, that the files are ignored whereas the folders are not:



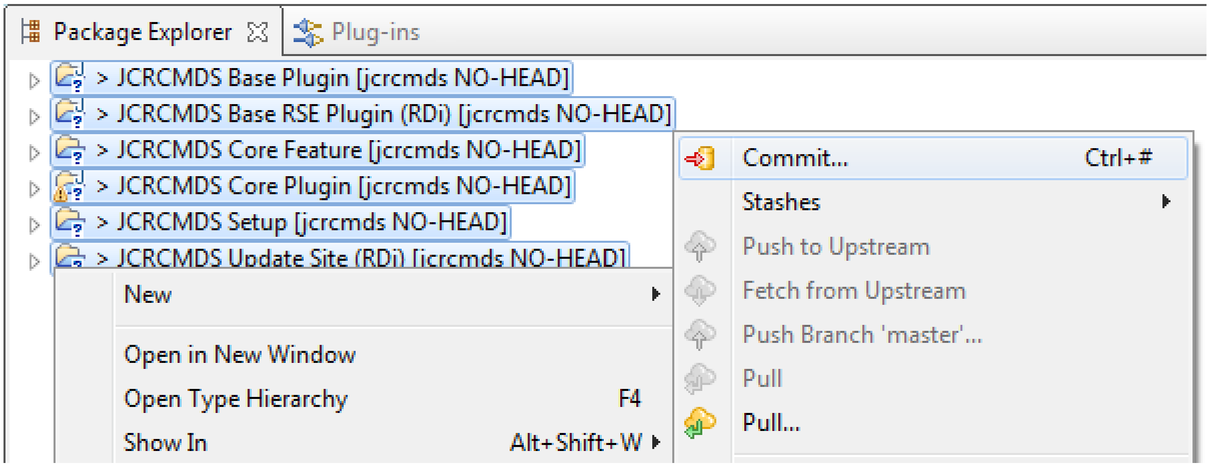
Properly ignored.

Removed the test files. You not need them any longer.

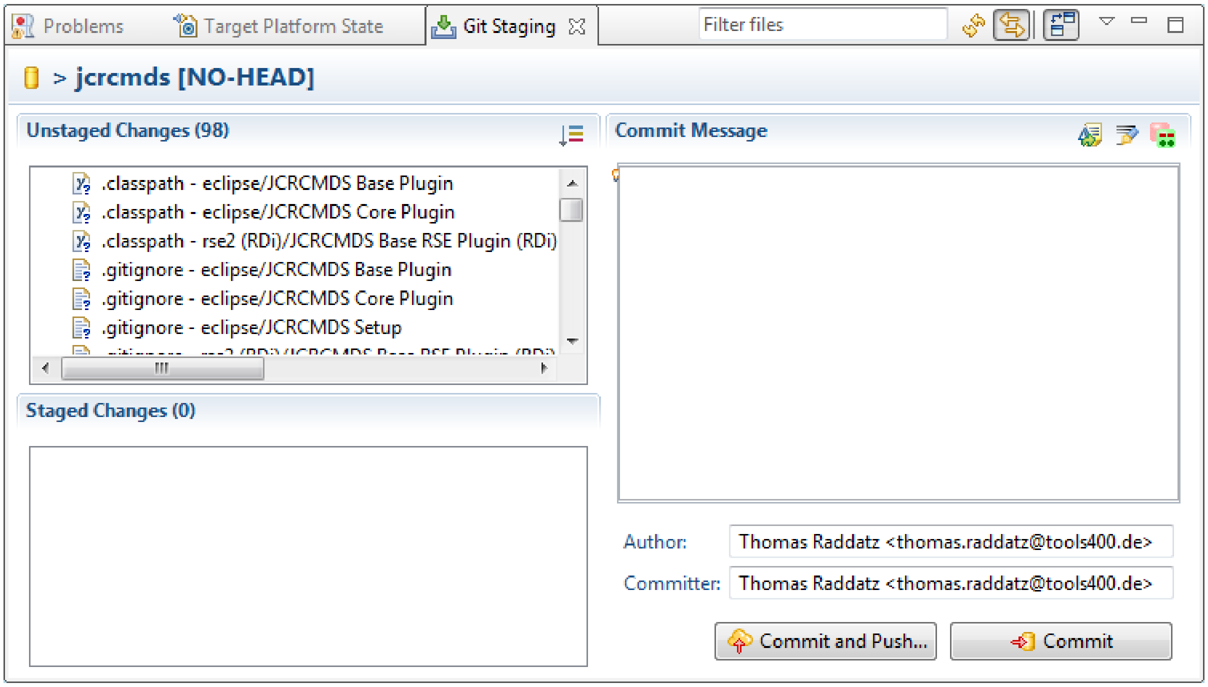
Now everything is set up for the first commit.

Initial Commit

Select all projects, then commit your changes:


Package Explorer
Plug-ins
> JCRCMDS Base Plugin [jcrcmds NO-HEAD]
> JCRCMDS Base RSE Plugin (RDi) [jcrcmds NO-HEAD]
> JCRCMDS Core Feature [jcrcmds NO-HEAD]
> JCRCMDS Core Plugin [jcrcmds NO-HEAD]
> JCRCMDS Setup [jcrcmds NO-HEAD]
> JCRCMDS Update Site (RDi) [jcrcmds NO-HEAD]
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Commit... Ctrl+#
Stashes
Push to Upstream
Fetch from Upstream
Push Branch 'master'...
Pull
Pull...

Notice, that the "Git staging" view has been opened. You may adjust its size to see all options:



> jcrcmds [NO-HEAD]

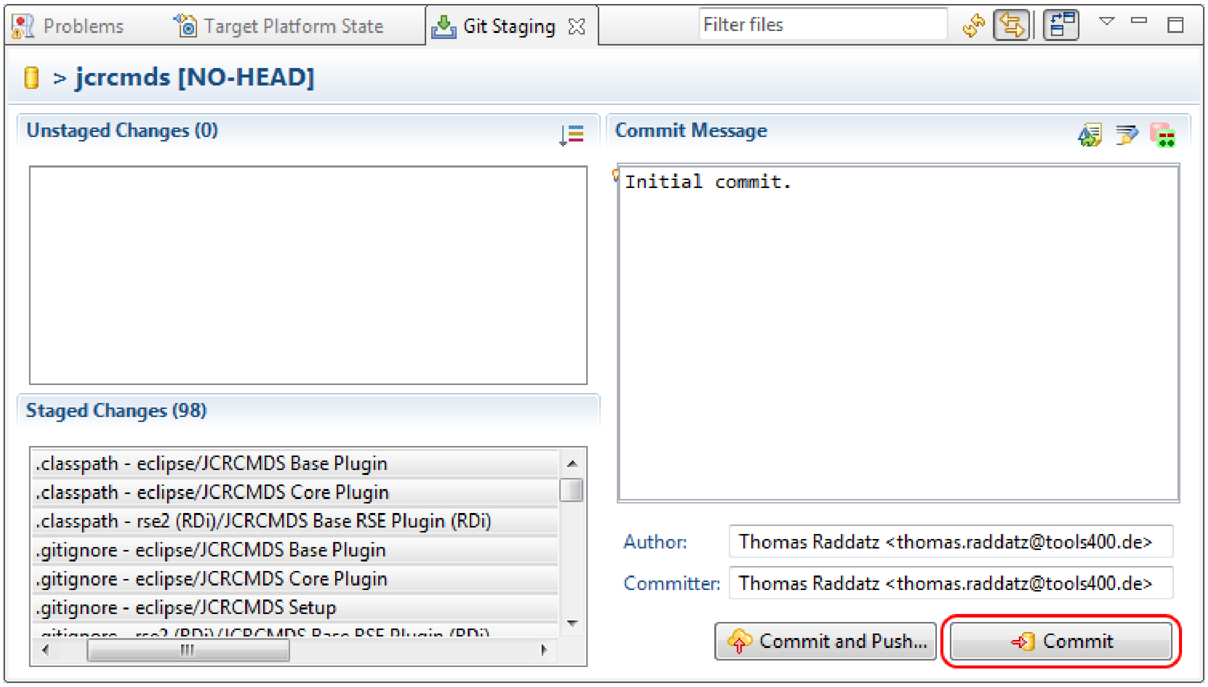
Unstaged Changes (98)

- .classpath - eclipse/JCRCMDS Base Plugin
- .classpath - eclipse/JCRCMDS Core Plugin
- .classpath - rse2 (RDi)/JCRCMDS Base RSE Plugin (RDi)
- .gitignore - eclipse/JCRCMDS Base Plugin
- .gitignore - eclipse/JCRCMDS Core Plugin
- .gitignore - eclipse/JCRCMDS Setup

Author: Thomas Raddatz <thomas.raddatz@tools400.de>

Committer: Thomas Raddatz <thomas.raddatz@tools400.de>

The "Unstaged Changes" window contains all uncommitted changes. In order to commit the changes you first need to move them to the "Staged Changes" window. Select one item and then press Ctrl+A to select all. Then drag and drop the items into the "Staged Changes" window or use the context menu and menu option "Add to index" to do so. Also add a commit message. The result should look like this:



> jcrcmds [NO-HEAD]

Commit Message:

```
Initial commit.
```

Staged Changes (98)

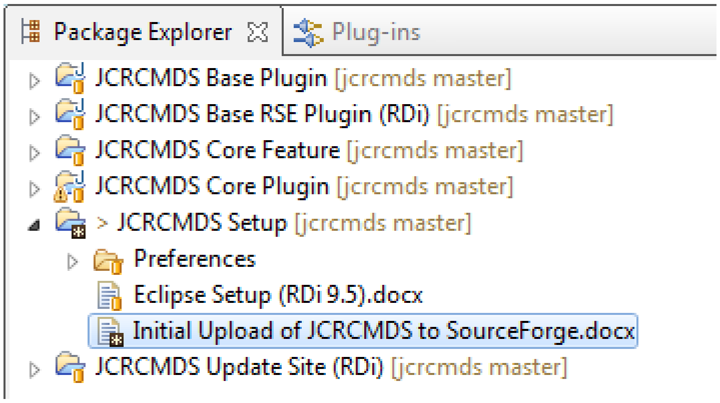
- .classpath - eclipse/JCRCMDS Base Plugin
- .classpath - eclipse/JCRCMDS Core Plugin
- .classpath - rse2 (RDi)/JCRCMDS Base RSE Plugin (RDi)
- .gitignore - eclipse/JCRCMDS Base Plugin
- .gitignore - eclipse/JCRCMDS Core Plugin
- .gitignore - eclipse/JCRCMDS Setup

Author: Thomas Raddatz <thomas.raddatz@tools400.de>

Committer: Thomas Raddatz <thomas.raddatz@tools400.de>

Now click the [Commit] button to commit the changes to your local repository.

Now everything but "Initial Upload of JCRCMDS to SourceForge.docx" is marked as committed (the orange little decoration). The black asterisk decoration of "Initial Upload of JCRCMDS to SourceForge.docx" indiactes, that this files has uncommited changes again. That is, because I often update that file while writing this tutorial.

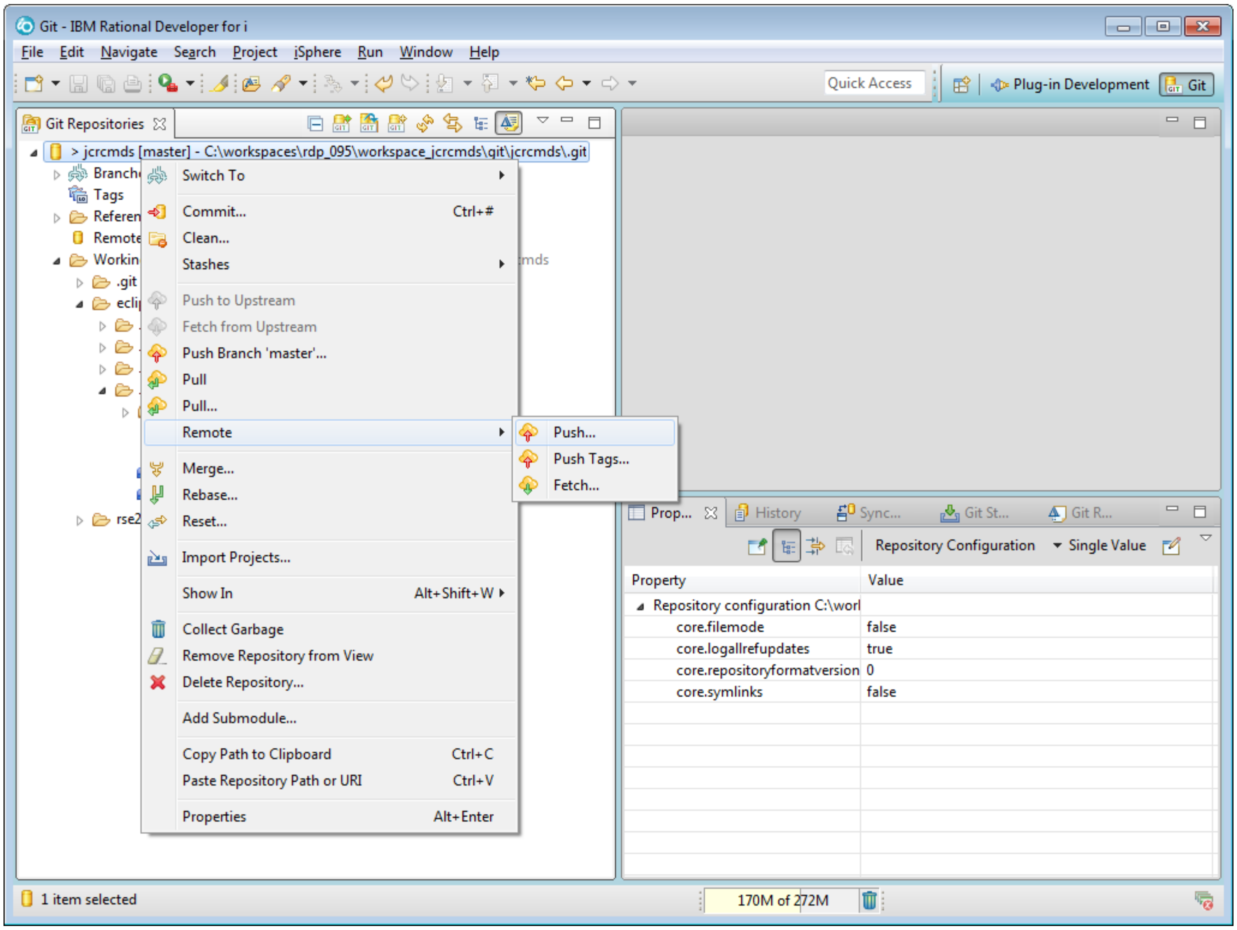


```
JCRCMDS Base Plugin [jcrcmds master]
JCRCMDS Base RSE Plugin (RDi) [jcrcmds master]
JCRCMDS Core Feature [jcrcmds master]
JCRCMDS Core Plugin [jcrcmds master]
> JCRCMDS Setup [jcrcmds master]
    Preferences
    Eclipse Setup (RDi 9.5).docx
    Initial Upload of JCRCMDS to SourceForge.docx
JCRCMDS Update Site (RDi) [jcrcmds master]
```

I also added "/~*.tmp" to the Git ignore list of project "JCRCMDS Setup" to exclude these temporary files as well.

Pushing Changes of Local Repository to SourceForge

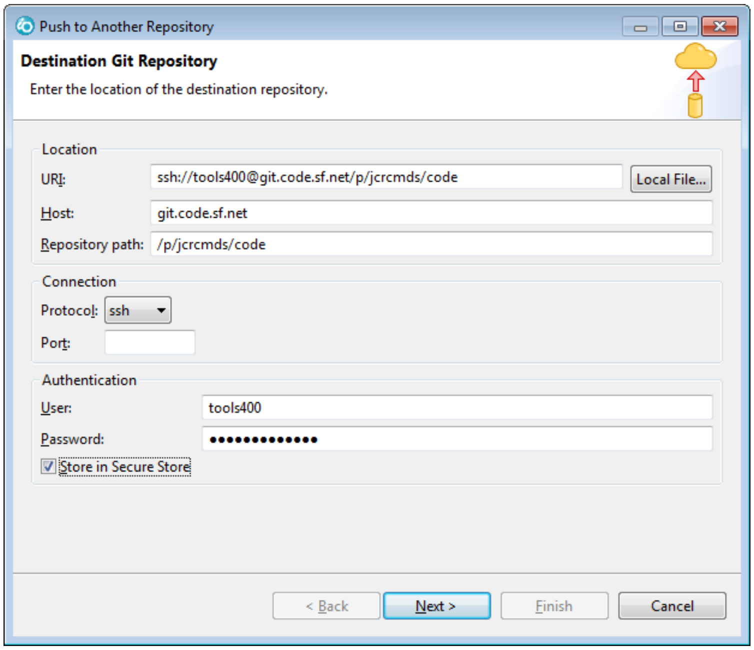
Switch to the "Git" perspective to push the local Git repository to SourceForge for the first time.



Property	Value
Repository configuration C:\worl	
core.filemode	false
core.logallrefupdates	true
core.repositoryformatversion	0
core.symlinks	false

Paste the URL of the SourceForge Git repository into the dialog that just popped up:

```
ssh://tools400@git.code.sf.net/p/jcrcmds/code
```



Push to Another Repository

Destination Git Repository

Enter the location of the destination repository.

Location

URI: ssh://tools400@git.code.sf.net/p/jcrcmds/code

Host: git.code.sf.net

Repository path: /p/jcrcmds/code

Connection

Protocol: ssh

Port:

Authentication

User: tools400

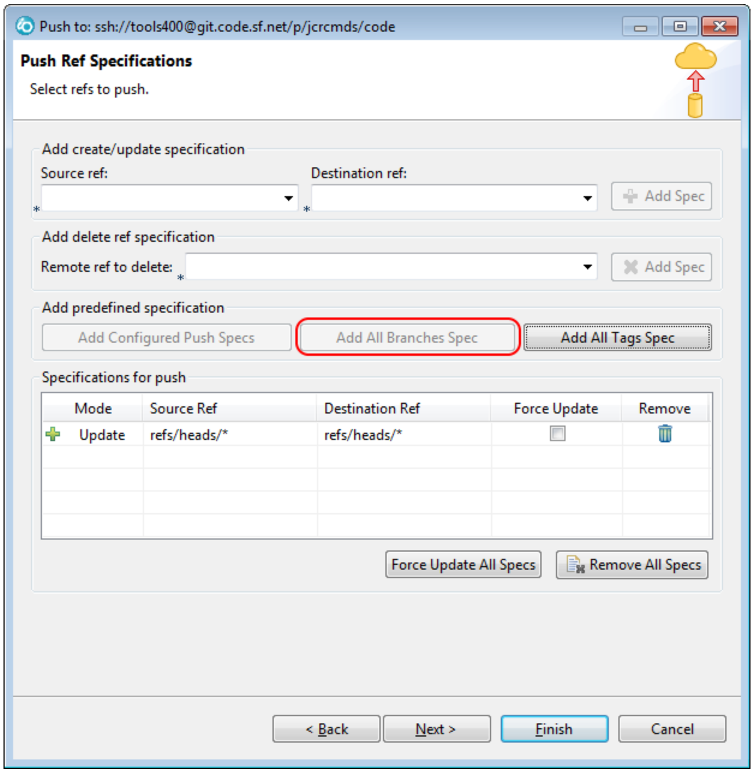
Password: ••••••••••••••

☑ Store in Secure Store

The user has been automatically retrieved from the global Git preferences.

Enter your SourceForge password and optionally check "Store in Secure Store". I usually do that, because I am a lazy developer and I do not like entering it again and again

On the next page click [Add All Branches Spec] and then [Next]:



Push to: ssh://tools400@git.code.sf.net/p/jcrcmds/code

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref:

Destination ref:

Add delete ref specification

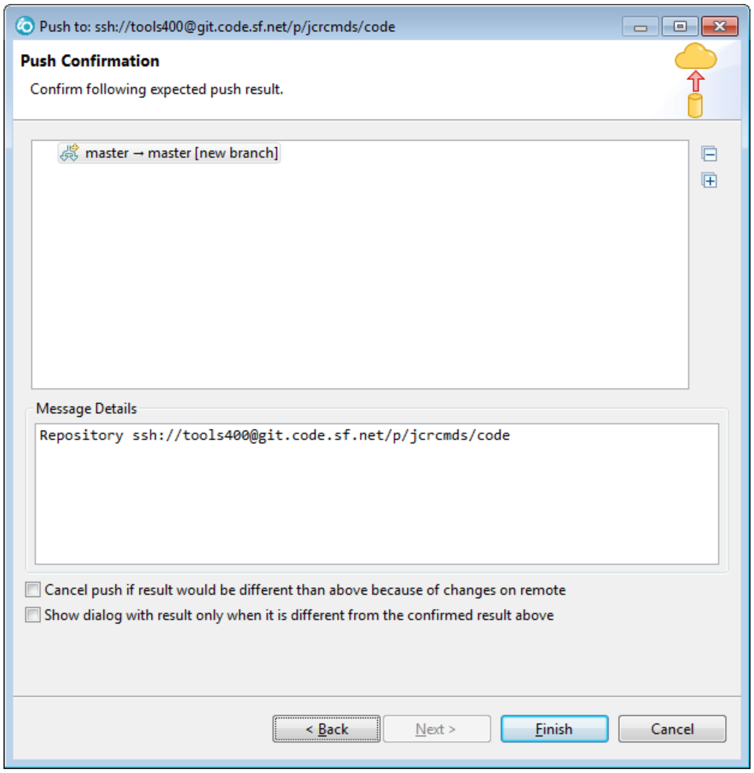
Remote ref to delete:

Add predefined specification

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
Update	refs/heads/*	refs/heads/*	☐	🗑

Now you are ready and about to push your projects to SourceForge.


Push to: ssh://tools400@git.code.sf.net/p/jcrcmds/code
Push Confirmation
Confirm following expected push result.
master → master [new branch]
Message Details

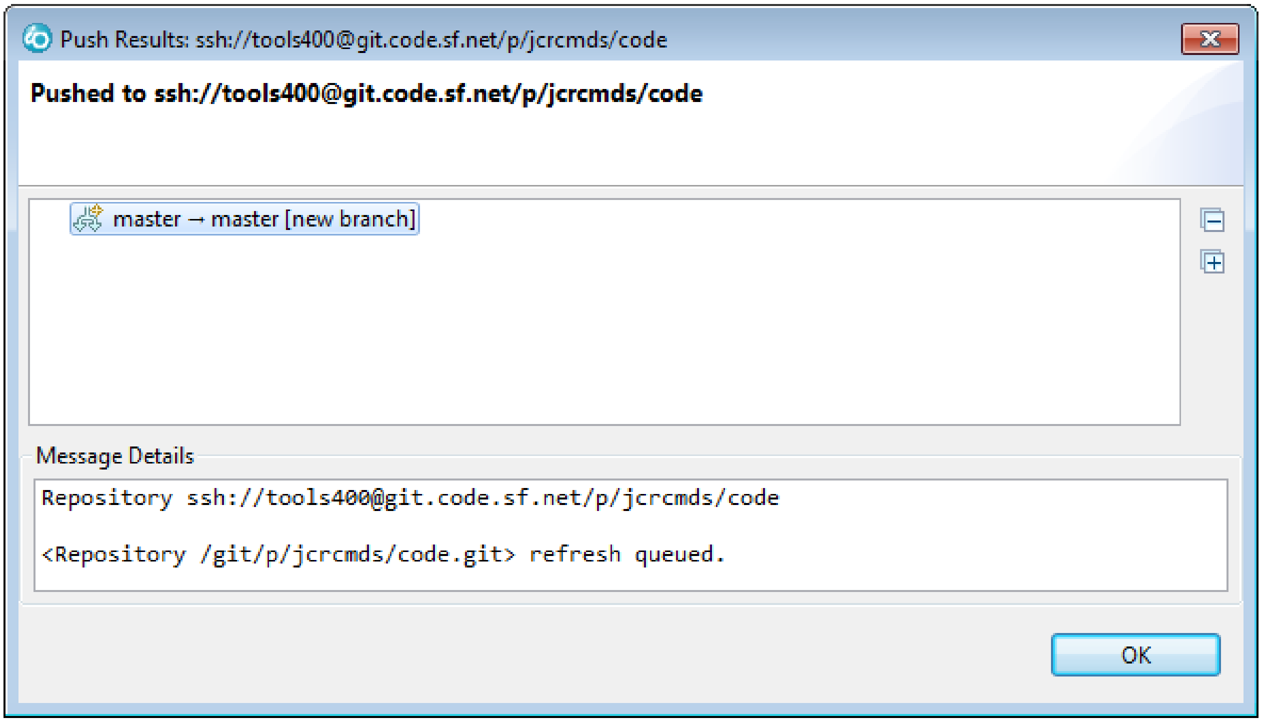
```
Repository ssh://tools400@git.code.sf.net/p/jcrcmds/code
```

☐ Cancel push if result would be different than above because of changes on remote
☐ Show dialog with result only when it is different from the confirmed result above

< Back
Next >
Finish
Cancel

Click the [Finish] button to do so.

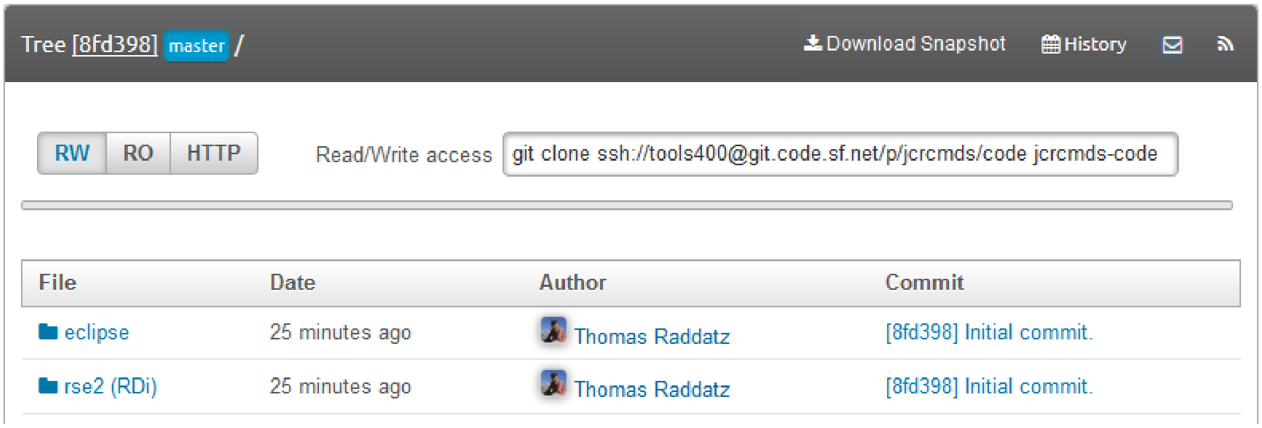
The following dialog confirms that it worked just fine:


Push Results: ssh://tools400@git.code.sf.net/p/jcrcmds/code
Pushed to ssh://tools400@git.code.sf.net/p/jcrcmds/code
master → master [new branch]
Message Details

```
Repository ssh://tools400@git.code.sf.net/p/jcrcmds/code

<Repository /git/p/jcrcmds/code.git> refresh queued.
```

Let us check that at SourceForge:

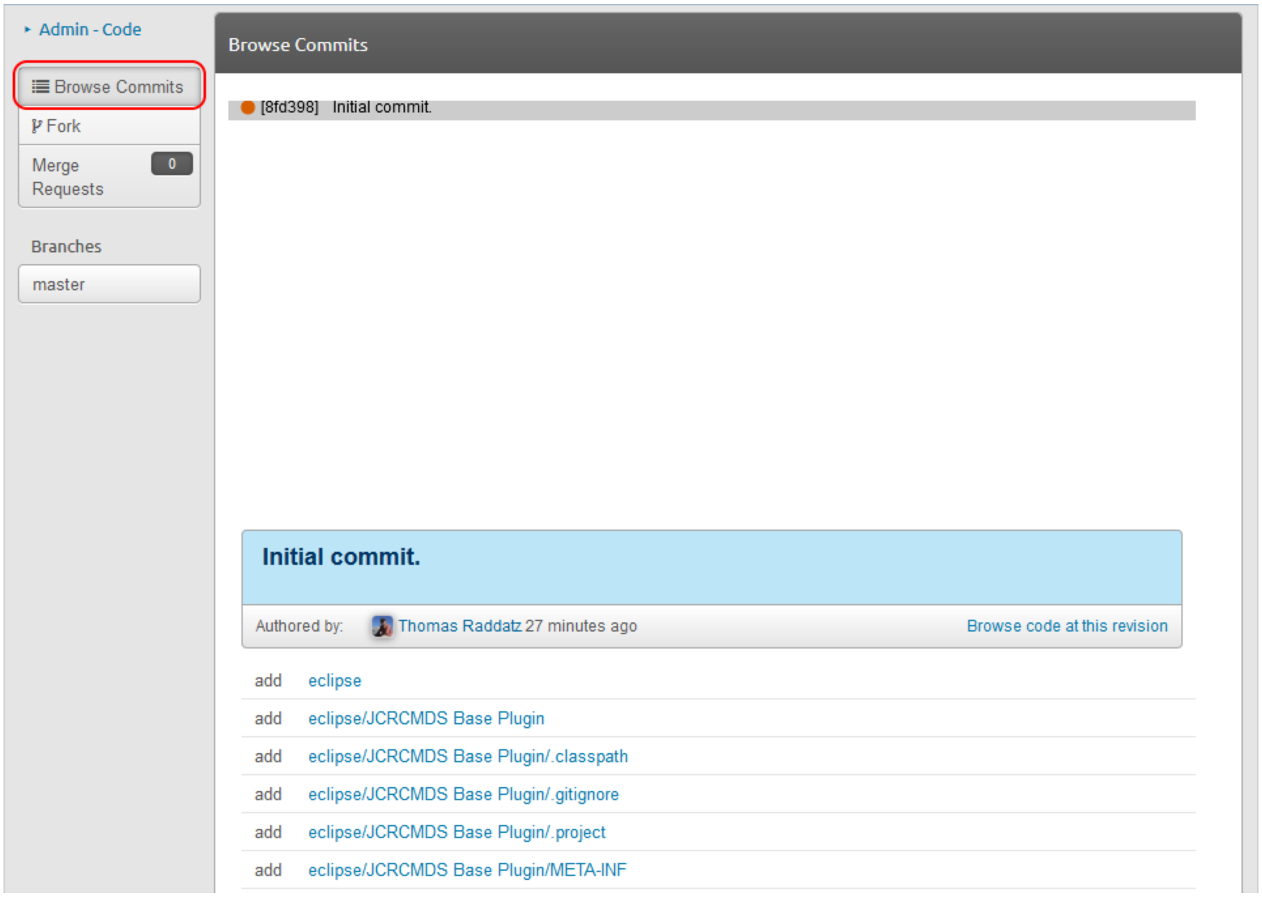


Read/Write access `git clone ssh://tools400@git.code.sf.net/p/jcrcmds/code jcrcmds-code`

File	Date	Author	Commit
eclipse	25 minutes ago	Thomas Raddatz	[8fd398] Initial commit.
rse2 (RDi)	25 minutes ago	Thomas Raddatz	[8fd398] Initial commit.

Looks good, doesn't it?

Let us also check the commits:



● [8fd398] Initial commit.

Initial commit.

Authored by: Thomas Raddatz 27 minutes ago Browse code at this revision

add	eclipse
add	eclipse/JCRCMDS Base Plugin
add	eclipse/JCRCMDS Base Plugin/.classpath
add	eclipse/JCRCMDS Base Plugin/.gitignore
add	eclipse/JCRCMDS Base Plugin/.project
add	eclipse/JCRCMDS Base Plugin/META-INF

Looks well, too.

Thomas Raddatz, 07.01.2017