

RTC 502 and its Git adapter, Git and its prerequisite software installation and setup guide

- In this doc, OS is RedHat Enterprise Server 6.4.
 - Note: Why ? Because open source software installation sometimes requires its “build”, using Linux is easier.
- Links to docs
 - **Existing “How to configure RTC Git adapter” docs**
 - ✧ <https://jazz.net/library/article/1426>
 - ✧ <https://jazz.net/library/article/1479>
 - ✧ https://www.ibm.com/developerworks/community/blogs/nfrsblog/entry/rational_team_concert_and_git_integration_setup_guide?lang=en
 - **RTC Git adapter Knowledge Center**
 - ✧ http://www-01.ibm.com/support/knowledgecenter/SSCP65_5.0.2/com.ibm.team.connector.cq.doc/topics/c_integ_git.html?lang=en
 - **Git Doc**
 - ✧ <http://git-scm.com/docs>
 - **Apache Doc**
 - ✧ <http://httpd.apache.org/docs/2.4/>
- OS configuration
 - Add a user and its group as a git functional id. (For example, group=git, user=git2)
 - ✧ Add a group to share the git repo
 - login as root
 - groupadd git
 - ✧ Add an user for a git functional id (The processes are git daemon and httpd's cgi)
 - login as root
 - useradd git2
 - usermod -G git,git2 -g git git2
 - passwd git2 <password>
 - ✧ Make sure that you can login as git2

- Note: These user id and group id affect 1) git xinetd configuration 2) git repository ownership and permission and 3) apache httpd.conf configuration.

- LDAP configuration parameters

- Try to get LDAP configuration parameters from LDAP Admin. They should be same as what are used for RTC setup. (See [this link](#))

- ✧ For example.

LDAP Registry Location	ldap://ldap.mycompany.com:389
Base User DN	ou=ourMembers,o=mycompany.com
User Property Names Mapping	userId=uid,name=cn,emailAddress=mail
Base Group DN	ou=myMembers,ou=myGroups,o=mycompany.com
Jazz to LDAP Group Mapping	JazzAdmins=MyJazzAdmins,JazzProjectAdmins=MyJazzProjectAdmins,JazzUsers=MyJazzUsers,JazzDWAdmins=MyJazzDWAdmins,JazzGuests=MyJazzGuests
Group Name Property	cn
Group Member Property	jazzMembers

- Note that you need to create following your LDAP groups. (See “Jazz to LDAP Group Mapping” above)

- MyJazzAdmins

- MyJazzProjectAdmins

- MyJazzUsers

- MyJazzDWAdmins

- MyJazzGuests

- Download prerequisite software images (Note: KnowledgeCenter requires specific version of prerequisite software, so please use the same version below.)

- git - v 1.7.9 or later version. In this doc, Git 2.3.0 is used.

1. <https://www.kernel.org/pub/software/scm/git/git-2.3.0.tar.gz>

- XAMPP(LAMPP) - In this doc, v 5.6.3 is used. (XAMPP is convenient because Apache is already configured.)

1. <http://downloads.sourceforge.net/project/xampp/XAMPP%20Linux/5.6.3/xampp-linux-x64-5.6.3-0-installer.run>

- python - v3.3.5. This specific version is required.

1. <http://www.python.org/ftp/python/3.3.5/Python-3.3.5.tgz>

- libgit2 - This is required by RTC Git hook. In this doc, v0.21.5 is used. (Note: According to [Knowledge Center](#), this is optional. However, if this parameter isn't specified, the hook won't work. So we need to assume that this is mandatory. See [this work item](#).)
 1. <https://github.com/libgit2/libgit2/archive/v0.21.5.zip>
- cmake – This is required to build libgit2. In this doc, v3.1.3 is used.
 1. <http://www.cmake.org/files/v3.1/cmake-3.1.3.tar.gz>
- RTC Git hook – This is used to hook git/gerrit/smart HTTP
 1. <https://jazz.net/downloads/rational-team-concert/releases/5.0.2?p=allDownloads> and look for “Git Integration Hooks”
- Build and install prerequisite software (Note : It is assumed that all download files are at **/root/Downloads**)
 - git - <http://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
 1. login as root
 2. cd /root/Downloads
 3. tar -zxf git-2.3.0.tar.gz
 4. cd git-2.3.0
 5. make configure
 6. ./configure --prefix=/usr/local/git230
 7. make all
 8. make install
 9. vi /etc/profile.d/git230.sh
 10. Write the following content. (This is to add the path to git binaries to PATH environment variable.)


```
if ! echo ${PATH} | /bin/grep -q /usr/local/git230/bin ; then  

    export PATH=/usr/local/git230/bin:/usr/local/git230/libexec/git-core:$PATH  

fi
```
 11. chmod 0644 /etc/profile.d/git230.sh
 12. logout and login as root again
 13. Make sure that “git --version” shows “git version 2.3.0”
 - python - <https://docs.python.org/3.3/using/unix.html#on-linux>
 1. login as root
 2. cd /root/Downloads
 3. tar -zxf Python-3.3.5.tgz
 4. cd Python-3.3.5
 5. ./configure --prefix=/usr/local/Python335

6. make
 7. make install
 8. vi /etc/profile.d/python335.sh
 9. Write the following content. (This is to add the path to python binaries to PATH environment variable.)


```

if ! echo ${PATH} | /bin/grep -q /usr/local/Python335/bin ; then
    export PATH=/usr/local/Python335/bin:$PATH
fi

```
 10. chmod 0644 /etc/profile.d/ python335.sh
 11. logout and login as root again
 12. Make sure that “python3.3 --version” shows “Python 3.3.5”
- xampp
 1. login as root
 2. cd /root/Downloads
 3. chmod +x xampp-linux-x64-5.6.3-0-installer.run
 4. ./xampp-linux-x64-5.6.3-0-installer.run
 5. Make sure that xampp is installed at the default location (/opt/lampp)
 - cmake - <http://www.cmake.org/install/>
 1. login as root
 2. cd /root/Downloads
 3. tar xf cmake-3.1.3.tar
 4. cd cmake-3.1.3
 5. ./bootstrap --prefix=/usr/local/cmake313
 6. make
 7. make install
 - libgit2 - <https://libgit2.github.com/docs/guides/build-and-link/>
 1. login as root
 2. cd /root/Downloads
 3. unzip -qq libgit2-0.21.5.zip
 4. cd libgit2-0.21.5
 5. mkdir build
 6. cd build
 7. export PATH=/usr/local/cmake313/bin:\$PATH
 8. cmake ..
 9. cmake --build .
 - RTC Git Hook

1. login as root
2. cd /root/Downloads
3. mkdir GitHook502
4. cd GitHook502
5. unzip -qq ../RTC-GIT-Hooks-5.0.2.zip

- git configuration

- git - xinetd configuration - This is for git client to access git service as "git://"

1. login as root
2. vi /etc/xinetd.d/git
3. Write the following content.

```

service git
{
    disable            = no
    socket_type        = stream
    wait               = no
    user               = git2
    group              = git
    server              = /usr/local/git230/bin/git
    server_args         = daemon --base-path=/var/lib/git --export-all
                        --user-path=public_git --syslog --inetd --verbose --enable=receive-pack
    log_on_failure     += USERID
}

```

4. chmod 0600 /etc/xinetd.d/git
5. Restart xinetd
 1. service xinetd restart

- git - setup a shared git repo. For example,


1. login as root
2. mkdir -p /var/lib/git/public_git/AudioDev.git
3. cd /var/lib/git/public_git/ AudioDev.git
4. git init --bare --shared
5. chown -R git2:git .
6. chmod -R g+w .

Note1: /var/lib/git is in the path because this value is specified in server_args' "--base-path=" argument

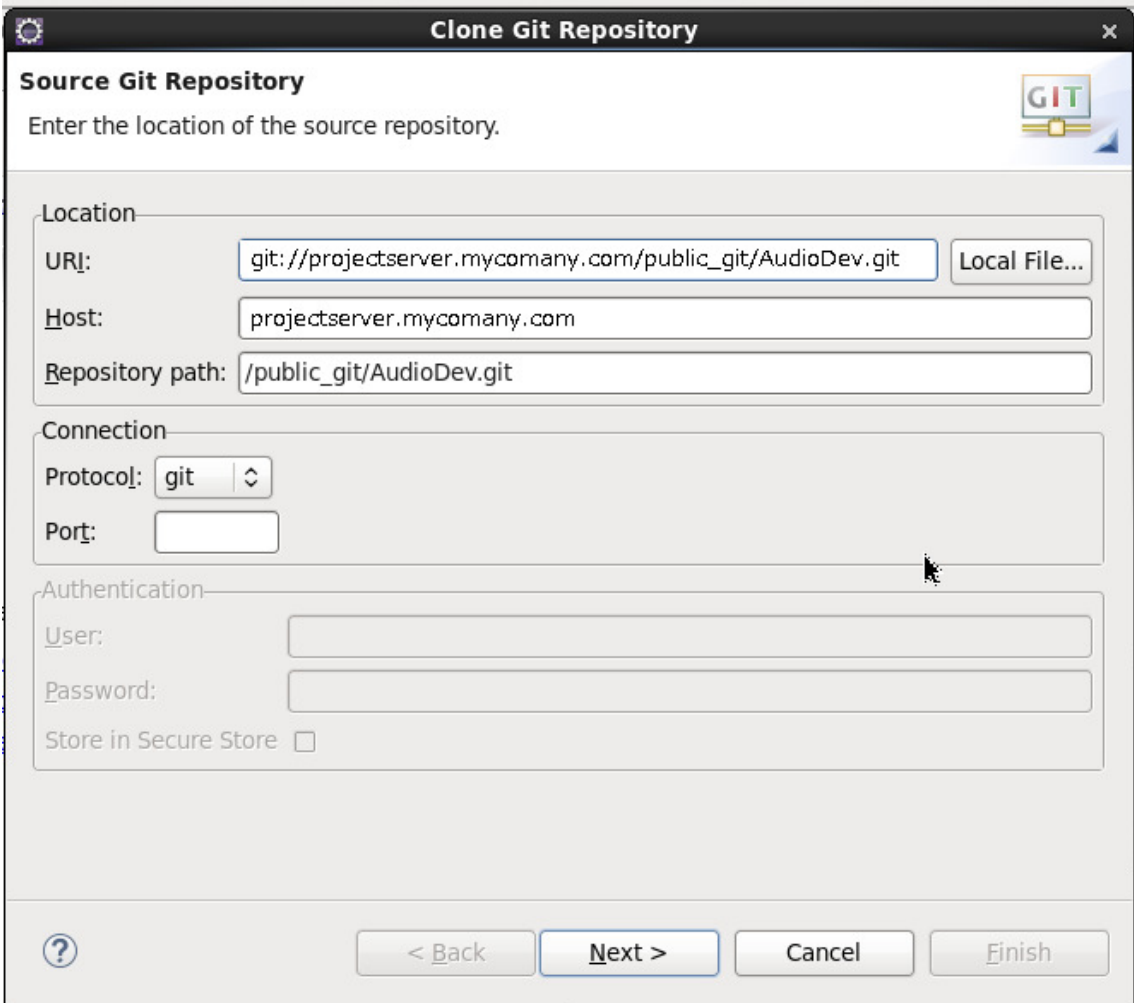
Note2: "public_git" is in the path because this value is specified in server_args' "--user-path=" argument.

- Verification for “git://” access
 - Start Eclipse with EGit.
 - Change git perspective
 - Click “Clone a Git repository”

Select one of the following to add a repository to this view:

-  [Add an existing local Git repository](#)
-  [Clone a Git repository](#)
-  [Create a new local Git repository](#)

- Use git:// protocol for Location URI. For example, it is something like git://projectserver.mycomany.com/public_git/AudioDev.git (Note: “public_git” comes from “--user-path=public_git” of “server_args” in /etc/xinetd.d/git. “AudioDev.git” comes from the directory name for git repo.)



Clone Git Repository

Source Git Repository
Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

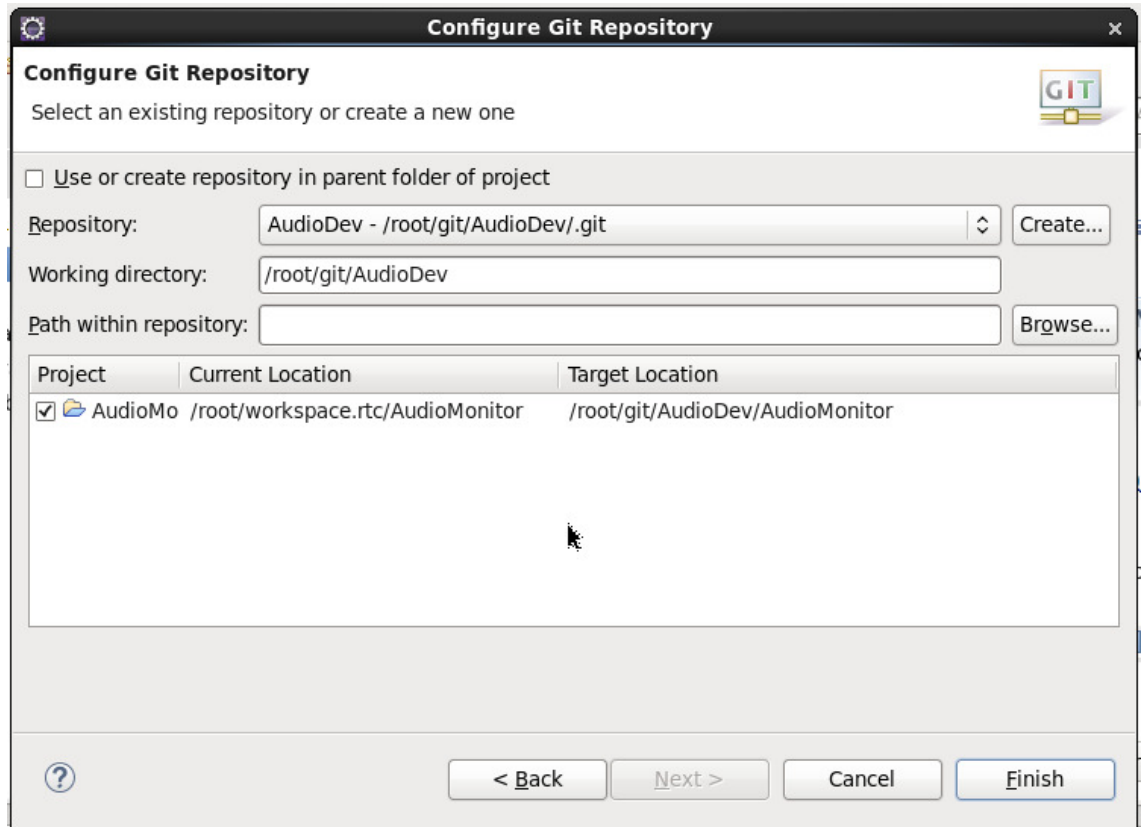
User:

Password:

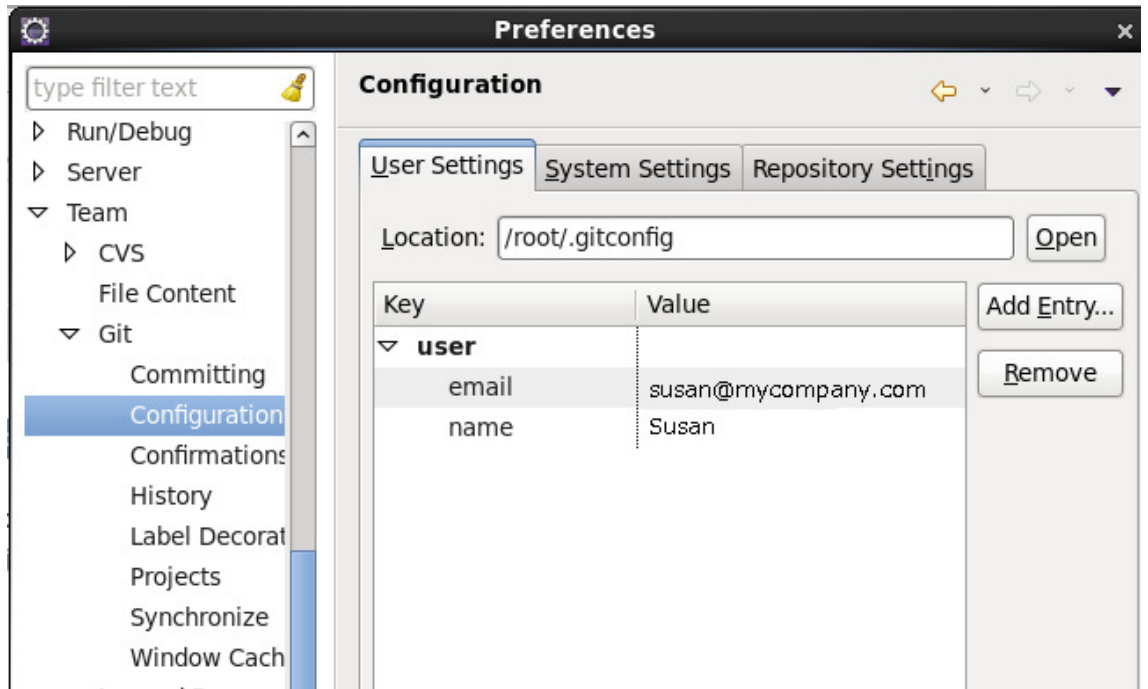
Store in Secure Store

- Change Java Perspective, create a Java project and add some Java classes.

- Select the project and open its context menu
- Team > Share > Git
- Select your local git repository. Default value is fine. For example,

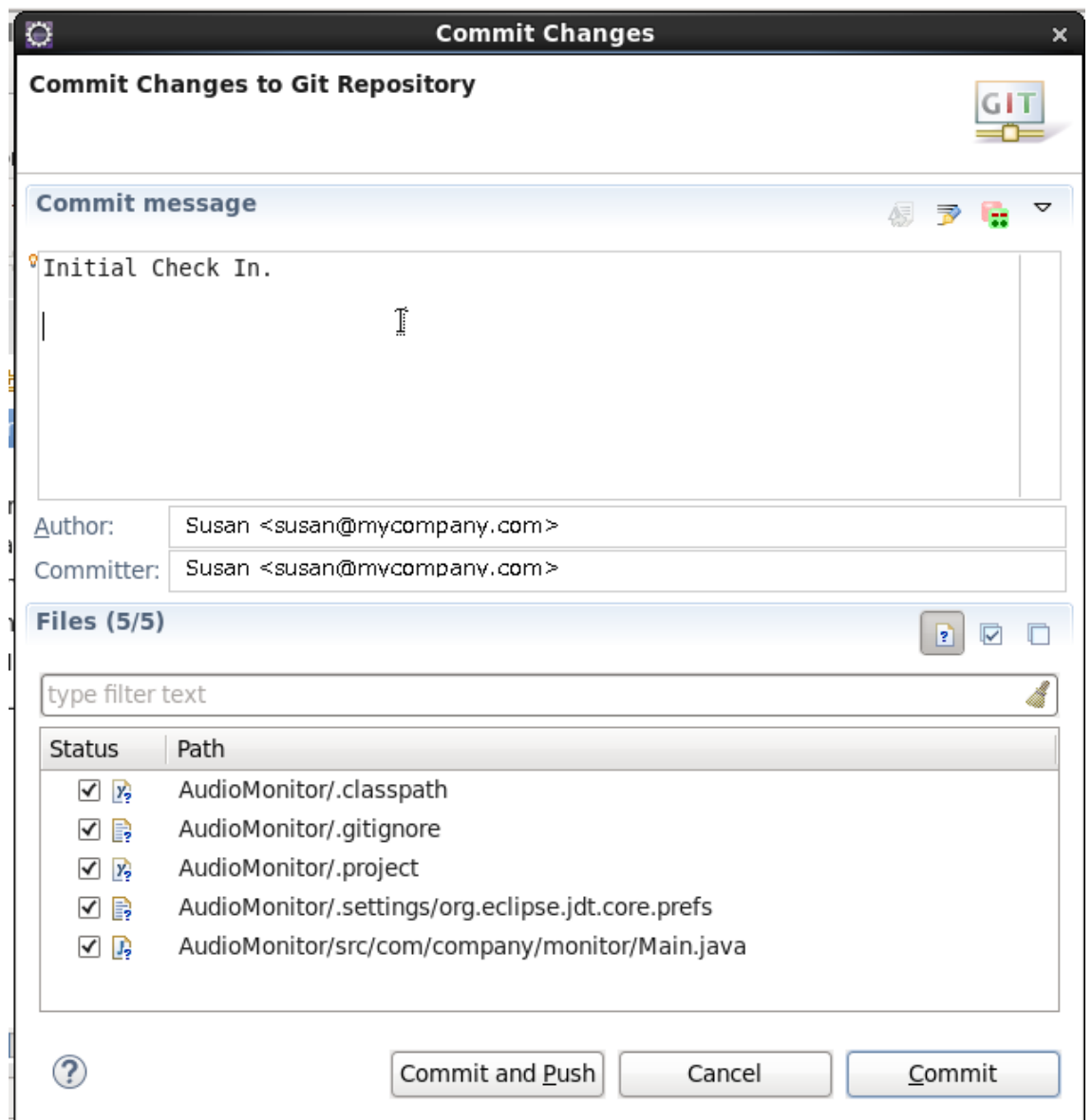


- Finish
- Windows > Preference > Team > Git > Configuration
- Add Entries for user.name and user.email properties. For example,

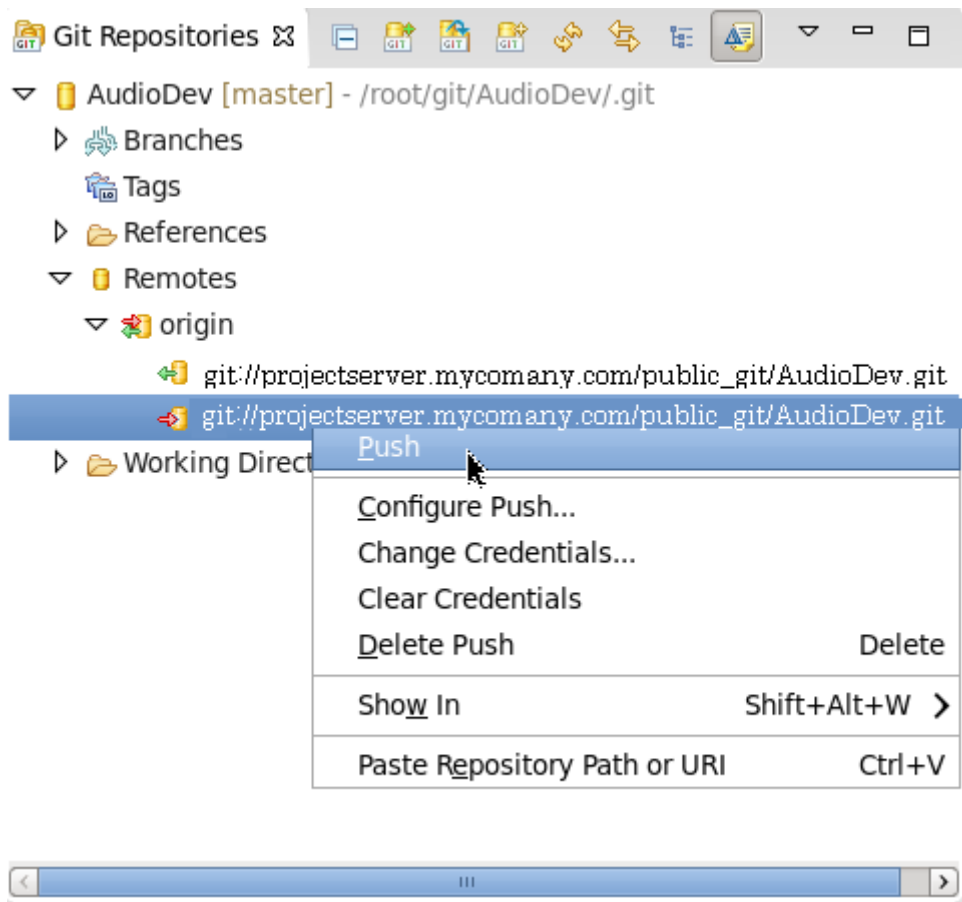


Note: email should be a valid email n your LDAP registry.

- Select the project again, and open its context menu
- Team > Commit
- Add a comment, check all files, and make sure that Author/Committer are same as what are specified in the preference page. For example,



- Commit
- Switch to Git perspective
- Push



- Make sure that everything is done without errors.
- Make sure that files are cloned correctly. For example,
 1. `mkdir tmp`
 2. `cd tmp`
 3. `git clone git://projectserver.mycomany.com/public_git/AudioDev.git`
 `AudioDev`
 4. Verify that files are cloned at “AudioDev” folder.
- Install and set up RTC
 - There are no special instructions other than a user registry.
 - LDAP should be used for the user registry.
 - The following is an example. (You can copy the text from [here in this doc.](#))

Property	Current Value
LDAP Registry Location The location of the LDAP registry.	<input type="text" value="ldap://ldap.mycompany.com:389"/> Default Value ldap://localhost:389
User Name User name to access LDAP registry. Anonymous mode is used if user name and password are not specified.	<input type="text"/> Default Value none
Password Password to access LDAP registry. Anonymous mode is used if user name and password are not specified.	<input type="password"/> Default Value none

Specify the following properties to configure Jazz Team Server to map to users in the LDAP server.

Property	Current Value
Base User DN Base distinguished name of users in the LDAP registry.	<input type="text" value="ou=ourMembers,o=mycompany.com"/> Default Value ou=people,dc=jazz,dc=net
User Property Names Mapping Mapping of Jazz user property names to LDAP registry entry attribute names. The mapping should be represented as {contributorAttributeName1}={LDAPEntryAttributeName1}, {contributorAttributeName2}={LDAPEntryAttributeName2}...	<input type="text" value="userId=uid,name=cn,emailAddress=mail"/> Default Value userId=uid,name=cn,emailAddress=mail

Specify the following properties to configure Jazz Team Server to map to groups in the LDAP server.

Property	Current Value
Base Group DN Base distinguished name of the Jazz application groups in the LDAP registry.	<input type="text" value="ou=myMembers,ou=myGroups,o=mycompany.com"/> Default Value ou=JazzGroups,dc=jazz,dc=net
Jazz to LDAP Group Mapping Mapping between Jazz groups and LDAP groups. One Jazz group can be mapped to multiple LDAP groups. The LDAP groups must be separated by a semi colon. For example, JazzAdmins=LDAPAdmins1; LDAPAdmins2 maps JazzAdmins group to LDAPAdmins1 and LDAPAdmins2.	<input type="text" value="JazzAdmins=MyJazzAdmins,JazzProjectAdmins=MyJazzProjectAdmins,JazzUsers=MyJazzU;"/> Default Value JazzAdmins=JazzAdmins, JazzUsers=JazzUsers, JazzDWAAdmins=JazzDWAAdmins, JazzProjectAdmins=JazzProjectAdmins, JazzGuests=JazzGuests
Group Name Property Property to represent the name of the Jazz groups in the LDAP registry.	<input type="text" value="cn"/> Default Value cn
Group Member Property Property to represent the members of a group in the LDAP registry.	<input type="text" value="jazzMembers"/> Default Value members

- Configure RTC Git Hooks - Basically we need to follow [Knowledge Center](#)
 - apache - Configure it to accept a hook as cgi
 1. login as root
 2. vi /opt/lamp/etc/httpd.conf.
 3. Change the port number. (Look for “Listen”) This example uses 9543.


```
#Listen 12.34.56.78:80
```

#For Git

#Listen 80

Listen 9543
 4. Change the user and group id to be used to start httpd. (Look for “daemon”)


```
#For Git
```

#User daemon

User git2

#Group daemon

Group git
 5. Append the following content at the last. (Note : “AuthLDAPURL” depends on your LDAP configuration.)

```
# For Git
<VirtualHost *:9543>
    SetEnv GIT_PROJECT_ROOT /var/lib/git/public_git
    SetEnv GIT_HTTP_EXPORT_ALL
    #ScriptAlias /git/ /usr/local/git230/libexec/git-core/git-http-backend/
    ScriptAlias /git/ /root/Downloads/GitHook502/rtc-git-http-interceptor/
    #SetEnv REMOTE_USER=$REDIRECT_REMOTE_USER

    <Directory "/usr/local/git230/libexec/git-core*">
        Options ExecCGI Indexes
        Order allow,deny

        Allow from all
        Require all granted
    </Directory>

    <Directory "/root/Downloads/GitHook502*">
        Options ExecCGI Indexes
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>

    #<LocationMatch "^/git./*/git-receive-pack$">
    #     AuthType Basic
    #     AuthName "Git Access"
    #     AuthUserFile /var/lib/git/.htpasswd
    #     Require valid-user
    #</LocationMatch>

    #<LocationMatch "^/git/.*">
    #     AuthType Basic
    #     AuthName "Git Access"
    #     AuthUserFile /var/lib/git/.htpasswd
    #     Require valid-user
```

#</LocationMatch>

<LocationMatch "/>

AuthType Basic

AuthName "Git Access"

AuthBasicProvider ldap

AuthLDAPURL

ldap:// ldap.mycompany.com:389/ou= ourMembers,o= mycompany.com?uid

Require valid-user

</LocationMatch>

</VirtualHost>

6. start apache
 1. login as root
 2. /opt/lampp/xampp startapache

➤ Register Git repo to RTC

1. login RTC as the user id you want to commit/push the files to Git. (For example, susan.) Note that this id/passwd is verified by LDAP.
2. Source Control > Work with Git
3. Register a Git Repository
4. Type something in Title field
5. Type Git Web Address. For example,

<http://projectserver.mycomany.com:9543/git/AudioDev.git>

(Note1: There is no "public_git" in URL because "SetEnv GIT_PROJECT_ROOT /var/lib/git/public_git" in httpd.conf has "public_git")

(Note2: The port number is specified in httpd.conf. In this example, 9543 is used.)

6. Set Owner and save it. You will see

The screenshot shows the 'Work with Git' interface in RTC. At the top, there's a search bar with 'AudioMonitorRepo' entered. Below it, there are tabs for 'Mapped Git References', 'Overview', and 'Settings'. The 'Overview' tab is active, showing the following fields:

- URL:
- Owner:
- Key:
- Description:

7. Copy the value of "Key.
8. Keep opening this web page.

➤ RTC Git - configuration

1. Copy git hooks to git repo and change their permission. For example,
2. login as root
3. `cp /root/Downloads/GitHook502/examples/pre-receive /var/lib/git/public_git/AudioDev.git/hooks`
4. `cp /root/Downloads/GitHook502/examples/post-receive /var/lib/git/public_git/AudioDev.git/hooks`
5. `cd /var/lib/git/public_git/AudioDev.git/hooks`
6. `chmod a+x pre-receive`
7. `chmod a+x post-receive`
8. Change the following values to the appropriate one in pre-receive and post-receive hooks. For example

`jazzRepoUrl=https://projectserver.mycomany.com:9443/ccm`

`gitRepoKey=<the string which is generated when Git repo is registered to RTC.>`

`pythonPath="/usr/local/Python335/bin/python3.3"`

`hooksPath="/root/Downloads/GitHook502"`

9. Add some git properties in git repo
 1. `cd /var/lib/git/public_git/AudioDev.git`
 2. `git config rtc.repourl https://projectserver.mycomany.com:9443/ccm`
 3. `git config rtc.repokey <the string which is generated when Git repo is registered to RTC.>`
 4. make sure that you see in "config" file at git repo.
`vi /var/lib/git/public_git/AudioDev.git/config`
then you will read

[rtc]

`repourl = https:// projectserver.mycomany.com:9443/ccm`

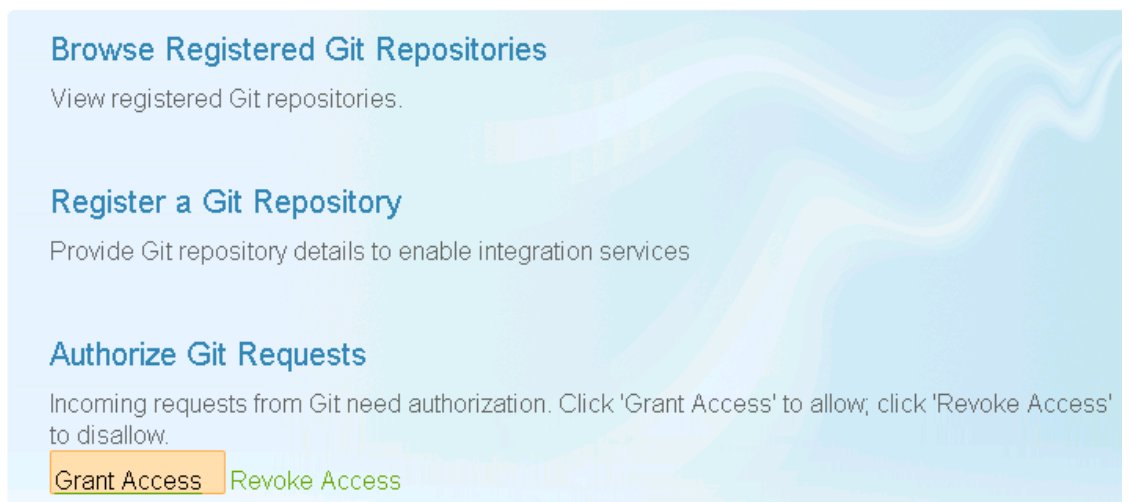
`repokey = <the string which is generated when Git repo is registered to RTC.>`

➤ Apache - configuration

1. Update apache hook. For example,
2. login as root
3. `chmod a+x /root/Downloads/GitHook502/rtc-git-http-interceptor`
4. `vi /root/Downloads/GitHook502/rtc-git-http-interceptor`

5. Change the following values to the appropriate one. For example
`PYTHON_PATH="/usr/local/Python335/bin/python3.3"`
`RTC_SCRIPTS_PATH="/root/Downloads/GitHook502"`
`GIT_HTTP_BACKEND_PATH="/usr/local/git230/libexec/git-core/git-http-backend"`
`LIBGIT2_PATH="/root/Downloads/libgit2-0.21.5/build/libgit2.so"`
 6. restart apache
 1. login as root
 2. `/opt/lampp/xampp restartapache`
- Save git smart HTTP server credential
 1. Go back to RTC web. You still see “Register a Git Repository”
 2. Click Settings tab
 3. Specify user id and its password to access Git through Apache. This id and password should be valid for LDAP.
 4. Save it.
 - Authorize Git Request
 1. login RTC as the user id you want to commit/push the files to Git. (For example, susan.)
 2. Source Control > Work with Git
 3. Grant Access

Work with Git



- Verification for “<http://>” access and WorkItem and Git integration
 - Login RTC login RTC as the user id you want to commit/push the files to Git. (For example, susan.)
 - Create a WorkItem and remember its ID.

- Start Eclipse with EGit
- Clone a git repo by http:// protocol. For example
Location URI is <http://projectserver.mycompany.com:9543/git/AudioDev.git>, and you need to specify the valid User and Password for LDAP, and also they are the same as what you register id and password at “Register a Git Repository”

Clone Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

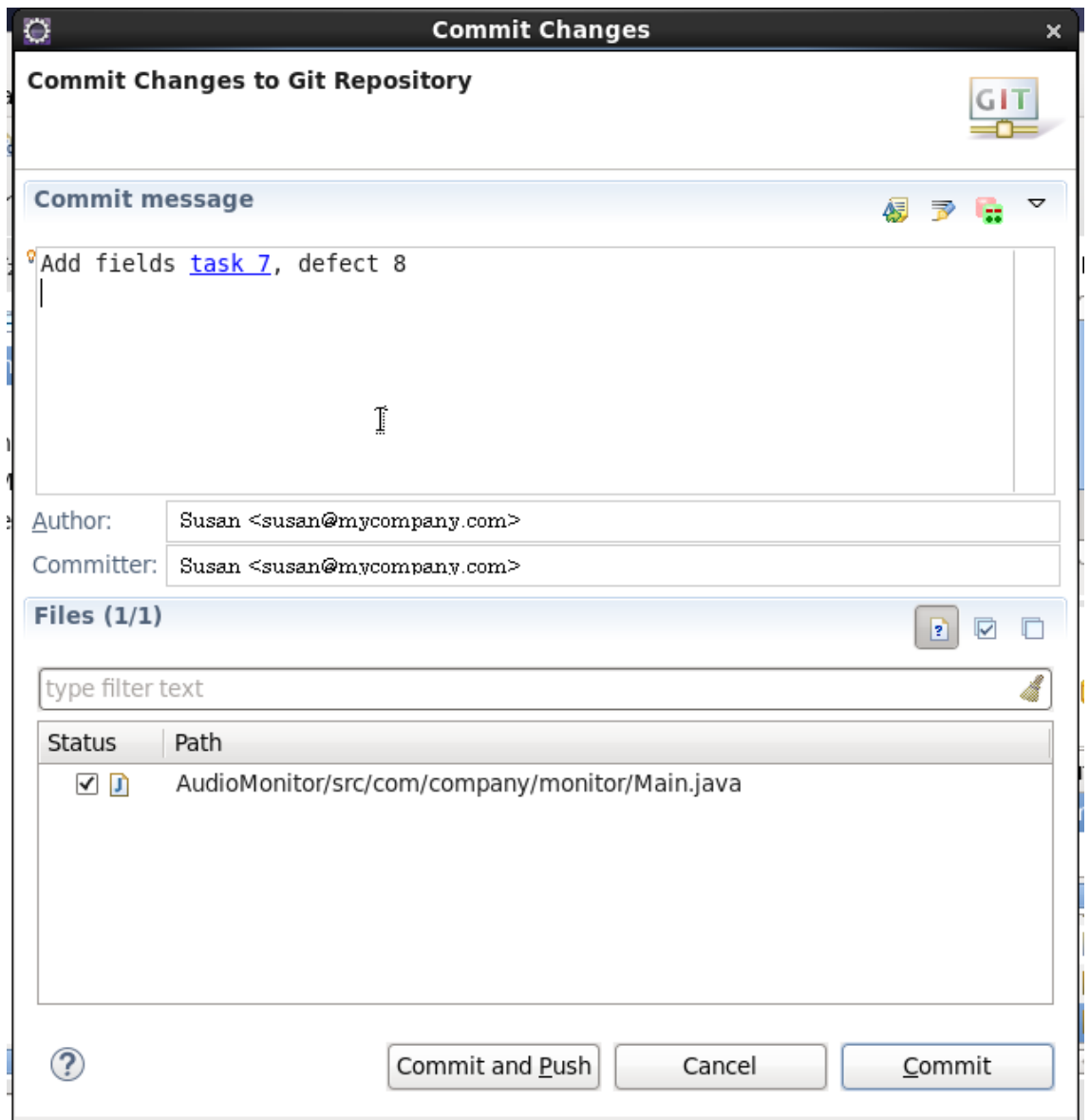
Authentication

User:

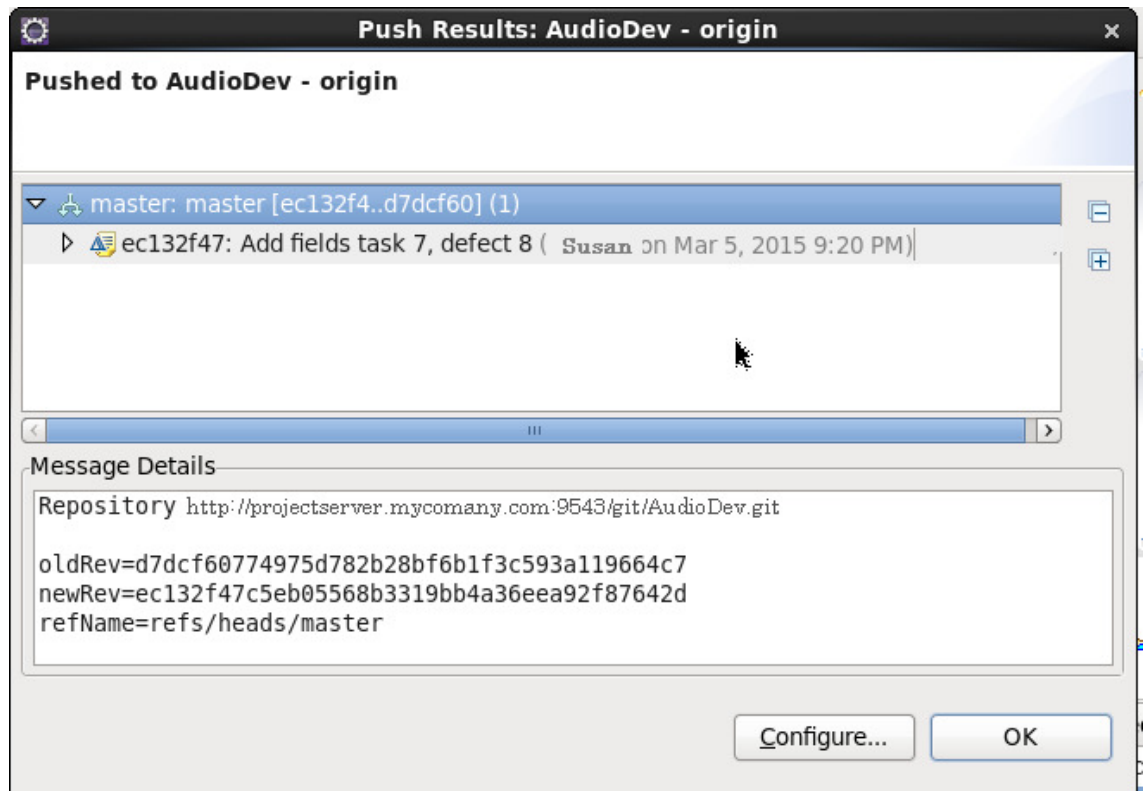
Password:

Store in Secure Store

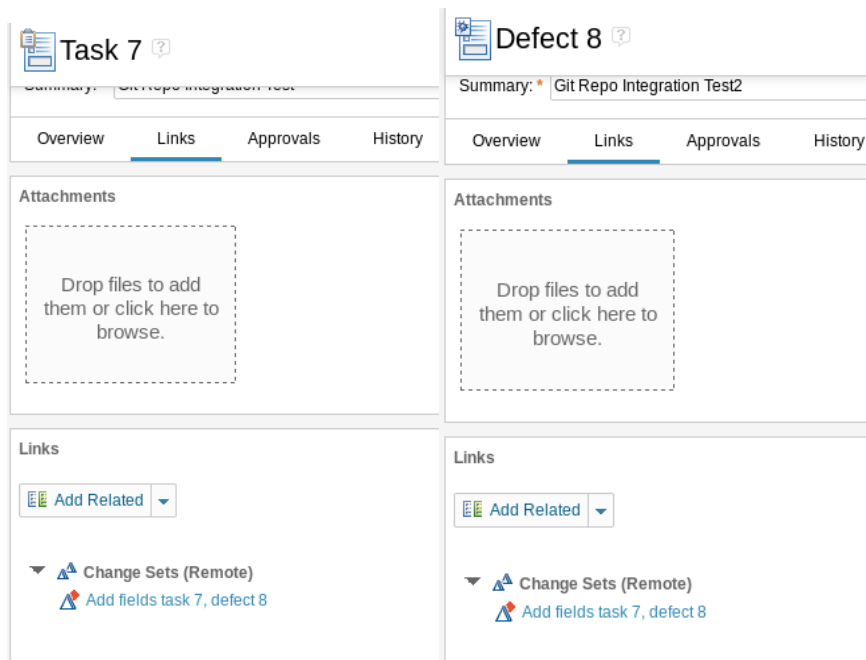
- Import a project, (For example, AudioMonitor in this doc.) and make changes. Or you can share the new project. But please don't commit changes.
- Select the project, open its context menu and Team > Commit.
- In this time, you need to write either viz, workitem, bug, defect, or task with the workitem number you created. (Note: Don't recommend to append "." (dot) after workitem number. "," comma seems to be okay, though.) For example,



- Then Commit, and Push. You will see



- Click OK. You can repeat these Commit and Push actions
- Take a look at WorkItem's Links area to see if Git ChangeSet is there or not.



- Make sure that its preview works

browse.

Start work : Task 13

Comment: Start work : Task 13
Committer Name: [Susan](#)
Git Commit SHA1: 9c375d7f64117aae9d8d707bff01d051b7615c3b
Date Created: "Thu Mar 5 21:58:22 2015 +0900"
Git Repository: http://projectserver.mycompany.com:9543/git/AudioDev.git
Link: [Start work : Task 13](#)

Links

Add Related

Parent
12: Audio Body Deve


Change Sets (Remot
Start work : Task 13


Show More

https://rhodium.toyosu.jpapar

- Also make sure that clicking a link to Git Change set shows you the following commit information.

Work with Git >

 Commit 9c375d... Start work : Task 13

SHA1: 9c375d7f64117aae9d8d707bff01d051b7615c3b
 Created on: Mar 5, 2015, 9:58:22 PM
 Author: **Susan**
 Email: susan@mycompany.com
 Comment: Start work : Task 13
 Reasons: [13: Audio Body Plan](#)
 Changes:  [AudioMonitor/src/com/company/monitor/Main.java](#) (Modified)

diff: AudioMonitor/src/com/company/monitor/Main.java

```
@@ -2,10 +2,9 @@ package com.company.monitor;

public class Main {
- private int counter;
-
public Main() {
// TODO Auto-generated constructor stub
+ int counter = 100;
}
}
```