INTLAND SOFTWARE

Intland Software GmbH
Gropiusplatz 10
D-70563 Stuttgart,
Germany

T +49 (711) 6740 0677
F +49 (711) 6740 0686
info@intland.com
www.intland.com

codeBeamer

SCMLoop Installation manual

Release	6.0.5
Date	1-st July 2013

Content

1 Introduction *3*

2 Installing SCMLoop *4*

2.1 Installation for Subversion and CVS **4**

2.1.1 Special SCMLoop settings 4

2.1.2 Accessing CodeBeamer via a Proxy 4

2.1.3 Allowing the SCM to commit changes even if CodeBeamer is not running 5

2.1.4 Host name problems when SCM system and CodeBeamer run on different hosts 5

2.2 Installation for Mercurial **5**

3 Instrumenting SCM repositories *6*

3.1 Instrumenting a CVS repository **6**

3.2 Instrumenting a Subversion repository **7**

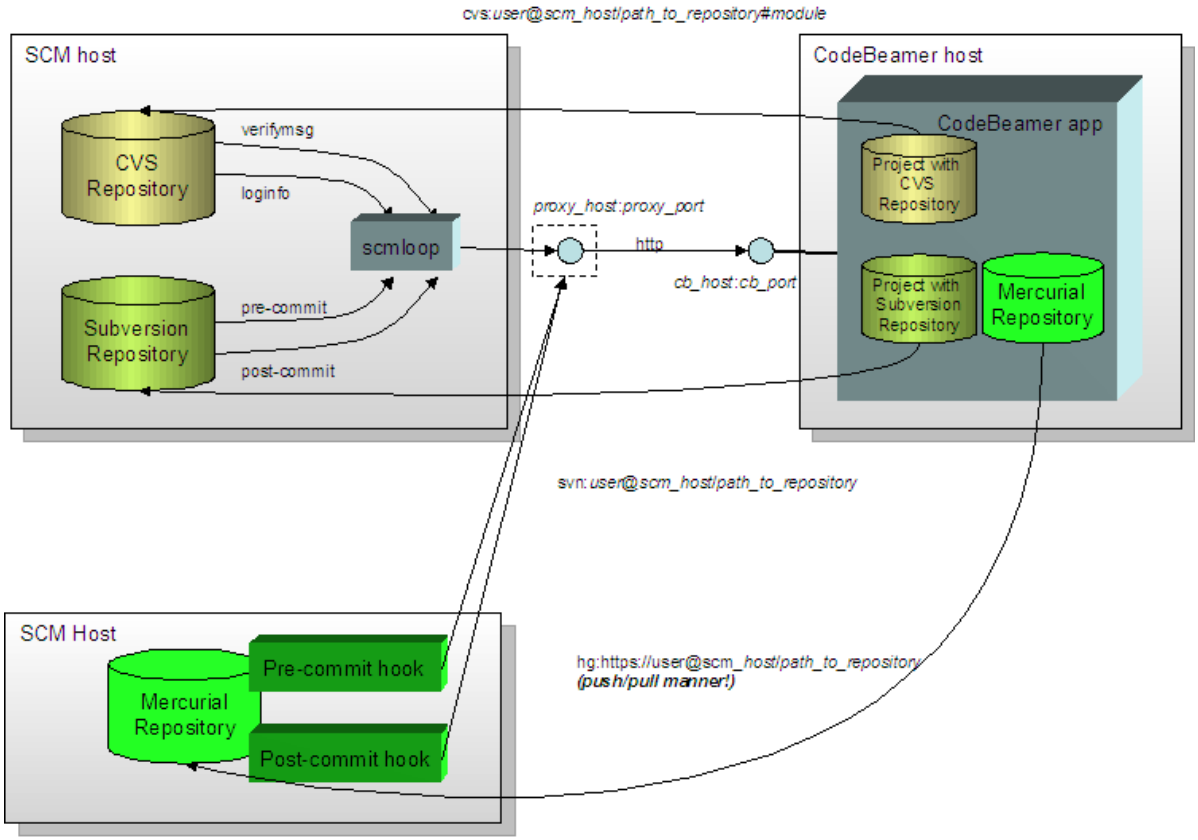
3.3 Instrumenting a Mercurial repository **8**

3.4 Instrumenting a Git repository **9**

1 Introduction

This document describes how to install the Intland codeBeamer SCMLoop component.

SCMLoop is a notification mechanism, which allows to associate commits to a Source Code Management (SCM) repository with tasks (issues) in codeBeamer.



Currently, SCMLoop is available for the following SCM systems:

- **Git (only on Unix)**
- **Subversion**
- **Mercurial (Hg)**
- **CVS**

CodeBeamer has built in support for so called “Managed Repositories”. Managed repositories are SCM repositories created and maintained by codeBeamer itself.

An exclusive feature of Managed (Git, Mercurial and Subversion) Repositories is the seamless integration to create and administrate (including access control) of repository with codeBeamer.

You only need to setup the extra SCMLoop component manually if you want to connect already existing (external) Subversion, Git, Mercurial,and CVS repositories to codeBeamer to for example associate SCM changesets (commits) to codeBeamer tasks (issues).

2 Installing SCMLoop

2.1 Installation for Subversion and CVS

You need to install the SCMLoop component on every server that hosts SCM repositories you want to connect.

Unpack the SCMLoop distribution archive `scmloop.zip` to a publicly accessible directory/folder. Either on each server that hosts SCM repositories, or on a network folder that is accessible from each SCM host.

Please note, that also a Java Runtime Environment (JRE 1.5 or higher) is required.

If you don't want to install a standard JRE, you can simply copy the `jre` directory/folder from the codeBeamer distribution to the same location where `scmloop` resides.

In order for SCMLoop to work, its configuration must be adjusted to the current environment.

On Windows systems you have to edit the script `scmloop/scmloop.bat`, on Unix systems `scmloop/scmloop`, and modify the values of the following variables:

- `SCM_DIR` is the path to the `scmloop[.bat]` script itself
- `CB_BASE_URL` the base URL of the codeBeamer webapplication. For example: http://localhost:8080/cb
- `JAVA` must point to the `java` executable of the JRE to use.

Make sure, that all SCM users are allowed to access and execute the `scmloop[.bat]` script. On Linux/Unix systems it must be ensured (eg. using command *chmod*) that the script `scmloop` is executable.

After you have successfully installed and configured SCMLoop, you can now start to instrument your SCM repositories.

2.1.1 Special SCMLoop settings

Under rare conditions, you may have to adjust other special SCMLoop settings.

2.1.2 Accessing codeBeamer via a Proxy

If `scmloop` cannot access codeBeamer directly, but only via an intermediate proxy, you have to set additional script variables:

```
PROXY_OPTIONS= -Dhttp.proxyHost=<hostname> -Dhttp.proxyPort=<portno>
```

If the proxy server also requires authentication, you have to provide the proxy authentication information via the options `-proxyuser` and `-proxypassword` in the `SCM_OPTIONS` variable:

```
SCM_OPTIONS= … -proxyuser <username> -proxypassword <password> …
```

2.1.3 Allowing changes even if codeBeamer is not running

After a SCM repository has been instrumented for `scmloop` (see 3), it will not longer be possible to commit changes to the repository while codeBeamer is not running or reachable.

This mode enables codeBeamer to control, whether commits without a valid tracker reference are allowed or not.

If for some reason, this constraint is not adequate for your installation, you can tell `scmloop` via the "`-catch`" option in the `SCM_OPTIONS` variable, to allow commits without codeBeamer confirmation:

```
SCM_OPTIONS= … -catch …
```

Commits executed while codeBeamer is unreachable will not get lost, because codeBeamer regularly synchronizes with all SCM repositories and will pick up these commits during this process.

2.1.4 Host name problems when SCM system and codeBeamer run on different hosts

Some common installation problem is with host names. By default, the scmloop script sends the official TCP/IP host name as the SCM system host name. When in codeBeamer an alias host name is configured for the SCM system, codeBeamer can't find the appropriate repository for the commit because codeBeamer compares host names and not IP addresses. The **"`-host`"** option can be used in `scmloop` script to send a host name instead of the official TCP/IP host name, that should match the SCM system host name configured in codeBeamer.

```
SCM_OPTIONS= … -host <hostname> …
```

Please consider the following example: your CVS or Subversion server has the TCP/IP name "**pluto**" and it has also an alias "**cvs-host**" and this name is configured as SCM host in a codeBeamer repository.
In this case as default, the scmloop script sends "**pluto**" as the host name of the SCM server and codeBeamer won't find the repository because this name doesn't match "**cvs-host**" .
But when you add " `-host cvs-host`" to the scmloop `SCM_OPTIONS`, codeBeamer will find the appropriate repository.

2.2 Installation for Mercurial

Install Mercurial 1.7.2 or later (http://mercurial.selenic.com/wiki/Download) and add `hg` to the `PATH`. On Windows use:

```
setx PATH "%PATH%;MERCURIAL_HOME"
```

on linux use

```
export PATH=$PATH:MERCURIAL_HOME
```

where `MERCURIAL_HOME` is the directory where you installed Mercurial.

3 Instrumenting SCM repositories

Instrumenting a SCM repository means to add appropriate SCMLoop callbacks/hooks to every SCM repository you want to connect to Codebeamer (or configuration file for Mercurial).

The SCMLoop distribution contains pre-instrumented examples for all files named in this chapter.
When copying the example files, please take care to adjust the path to the `scmloop` directory accordingly.

3.1 Instrumenting a CVS repository

Setting up a CVS repository and access to it, is not part of this document. For more information about CVS see http://www.cvshome.org.

We assume that the repository already exists and is accessible.

1. Check out the CVS repository administrative files (CVSROOT) to some working directory.
2. Edit the file **`verifymsg`** in the **`CVSROOT`** folder in the working directory and add the following line:

   `ALL` `[`**`cmd /c`**`]`[1] *`<path_to_scmloop>`*[2]**`scmloop -mode verifymsg`**

3. Edit the file **`loginfo`** in the **`CVSROOT`** folder in the working directory and add the following line:

   `ALL` `[`**`cmd /c`**`]`[1] *`<path_to_scmloop>`*[2]**`scmloop -mode loginfo %{sVv}`**

4. On some CVS servers you might have to add "`LogHistory=all`" to **`CVSROOT/config`**
5. Commit your changes to the repository.

[1] Only on Windows systems.

[2] Replace with the absolute path to the directory, where the `scmloop[.bat]` script resides (see chapter 2).

3.2 Instrumenting a Subversion repository

We assume that the repository already exists and is accessible. The templates of the Subversion hook scripts can be found under *svn/<operating_system>* directory in `scmloop.zip` (where *<operating_system>* is either *windows* or *unix*).

1. Check out the **`hooks`** directory of the Subversion repository to some working directory.
2. Copy the files from *svn/<operating_system>* to the `hooks` folder of the working directory.
3. Edit the scripts: replace the string `${scmloop}` with the actual path of the scmloop script.
4. On Linux/Unix systems it must be ensured (eg. using command *chmod*) that the scripts are executable.

Create a file codebeamer.properties under `conf` in your repository and add the following line to it:

`repositoryId=` *repository-ID*

where *repository-ID* is the id of the codeBeamer repository that you want to associate the repository with.

5. Commit your changes to the repository.

3.3 Instrumenting a Mercurial repository

We assume that the repository already exists and is accessible. The mercurial example configuration files can be found under the `hg` directory in scmloop.zip.

1. Check `hgrc` file under **`.hg`** directory (hidden directory at the root of repository) of the Mercurial repository. If there is an already existing one, append content of the provided **`hgrc`** file to the original one (with a preferred editor), otherwise copy it to **`.hg`** dir. Also copy `hgrc-codebeamer` to the `.hg` dir.
2. Copy the `commit-hook(.bat)` and the `changegroup-hook(.bat) scripts from hg/<operating_system>` (in scmloop.zip) to a directory of your choice.
3. (**Required**) Set your repositoryId property for the project where this repository used:

```
[codebeamer]
repositoryId = 266 # (number!)
...
```

4. **(Required)** In `hgrc-codebeamer` replace the string `${scmloop}` with the actual path of the scmloop script.
5. **(Required)** In `hgrc-codebeamer` replace all occurrences of the string `${hookDirectory}` with the absolute path of the directory where you copied the scripts to (step 2).
6. (**Optional**) In `hgrc-codebeamer` set your host name as it was registered in codebeamer. This value will be the hostname of the machine where the Mercurial run by default.

```
[codebeamer]
host = myscmhost
...
```

7. (**Optional**) In `hgrc-codebeamer` set cburl property in the `hgrc` file according to the settings of your codebeamer commit url (this will be set to `http://localhost:8080/cb/sccCommitInfo` if missing):

```
[codebeamer]
cburl = http://mycbhost
...
```

8. (**Optional**) Check and set proxyhost, proxyport, proxyuser, password properties in the `hgrc` file according to the settings of your network like:

```
[codebeamer]
proxyhost = myproxy
proxyport = 81
proxyuser = myproxyuser
password = v3ght # put password here in a base64 encoded form
...
```

   Online tool can be used for base64 encoding
   (like: http://www.motobit.com/util/base64-decoder-encoder.asp)
9. (**Optional**) If you would like to ignore codebeamer accessibility errors set the ignore flag (false by default):

```
[codebeamer]
...
ignore = true
```

3.4 Instrumenting a Git repository

We assume that the repository already exists and is accessible. The templates of the Git hook scripts can be found under *git/<operating_system>* directory in `scmloop.zip` (where *<operating_system>* is either *windows* or *unix*).

1. Copy the files from *git/<operating_system>* to the `hooks` folder of the working.
2. Edit the scripts: replace the string `${scmloop}` with the actual path of the scmloop script.
3. On Linux/Unix systems it must be ensured (eg. using command *chmod*) that the scripts are executable.
4. Create a file called `codebeamer.properties` under the `.git` directory of your repository or under the top directory in case of bare repository and add the following line to it:
   `repositoryId=` *repository-ID*
   where *repository-ID* is the id of the codeBeamer repository that you want to associate the repository with.