

# codeBeamer

## SCMLoop Installation manual

Release	7.7.1
Date	28-th July 2015

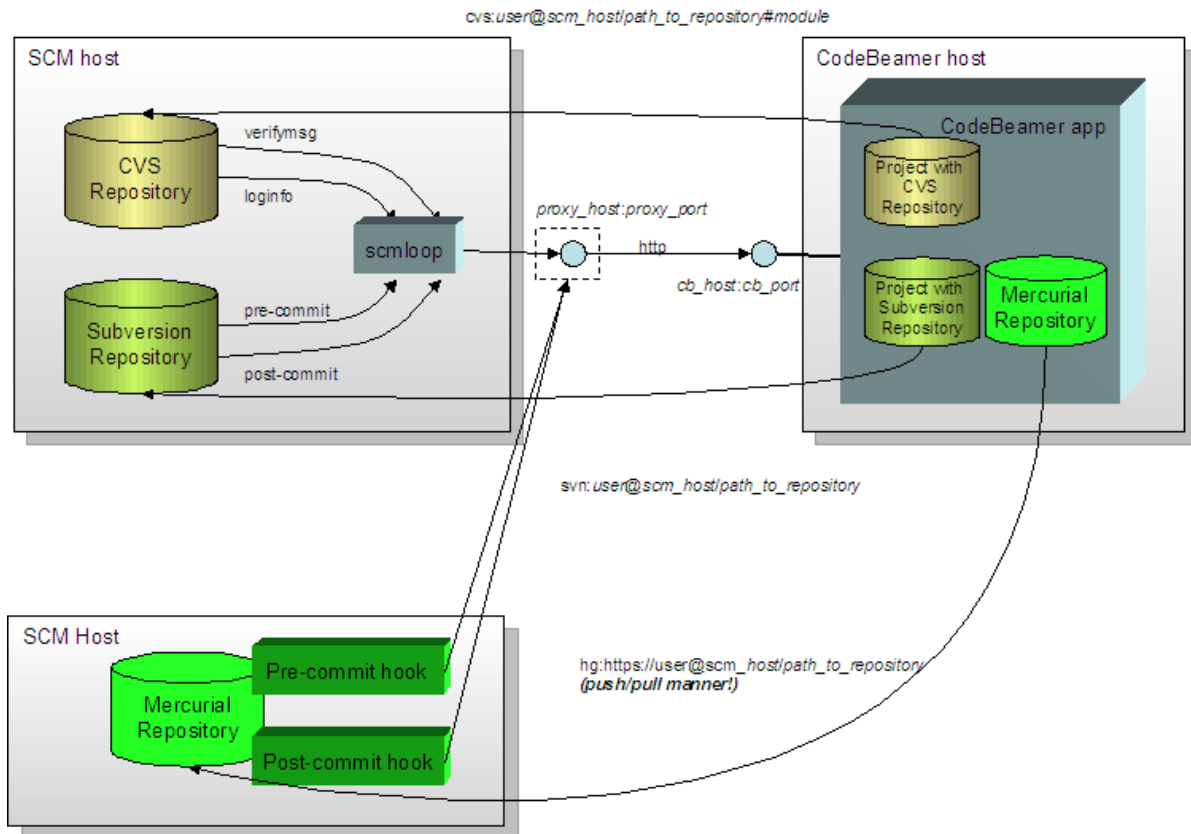
## Content

<b>1</b>	<b><i>Introduction .....</i></b>	<b>3</b>
<b>2</b>	<b><i>Installing SCMLoop.....</i></b>	<b>4</b>
2.1	Installation for Mercurial .....	4
<b>3</b>	<b><i>Instrumenting SCM repositories.....</i></b>	<b>5</b>
3.1	Instrumenting a CVS repository .....	5
3.2	Instrumenting a Subversion repository .....	6
3.3	Instrumenting a Mercurial repository .....	7
3.4	Instrumenting a Git repository .....	8

## 1 Introduction

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

SCMLoop is a notification mechanism, which allows associating commits to a Source Code Management (SCM) repository with Work Items 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 Work Items.

## 2 Installing SCMLoop

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 1.6 or 1.7) is required.

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 Installation for Mercurial

Install Mercurial 1.8 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>2scmloop -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>2scmloop -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.

---

<sup>1</sup> Only on Windows systems.

<sup>2</sup> 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 `scmlloop.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 `scmlloop.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 `${scmlloop}` with the actual path of the `scmlloop` script.
5. **(Required)** In `hgrc-codebeamer` replace all occurrences of the string `${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 *scmlloop.zip* (where *<operating\_system>* is either *windows* or *unix*).

1. Copy the files from *git/<operating\_system>* to the *hooks* folder of the git repository.
2. Edit the scripts: replace the string `${scmlloop}` with the actual path of the scmlloop 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.