|  |  |  |
| --- | --- | --- |
| Bochs Developers Guide | | |
| [Prev](http://docs.google.com/resources.html) | Chapter 1. Resources for developers | [Next](http://docs.google.com/other.html) |

1.2. Using SVN write access

1.2.1. Checking in files

Once you have a Bochs directory, you can compile the files, edit them, test them, etc. See the documentation section, [Tracking the source code with SVN](http://docs.google.com/user/get-src-svn.html) for more info on SVN, in the User Manual. But what's new and different is that you can now do SVN commits. When a file is all fixed and ready to share with the rest of the world, you run a commit command to upload your version to the server. First, it's good to do a SVN update to make sure nobody else has changed it since you downloaded it last. At the first commit you'll always have to specify your SF username and type your password.

$ svn update file.cc  
 $ svn commit --username *sfusername* file.cc  
 [editor opens. type log message, save, and exit.]  
 Login area: <https://svn.code.sf.net:443> SourceForge Subversion area  
 Username: *sfusername*  
 Password for '*sfusername*': *<--type your password*  
 Sending file.cc  
 Transmitting file data .  
 Committed revision 10.

When SVN starts an editor, The default is usually vi. If you want a different editor, set the EDITOR environment variable to the name of your preferred editor. When you're done, just save the file and quit the editor. Unless there's some problem, you will see a message that says what the new SVN revision number is, and then "done". If while you're editing the log message, you decide that you don't want to commit after all, don't save the file. Quit the editor, and when it asks where the log message went, tell it to abort.

Here is an example of a successful checkin:

$ svn commit misc.txt  
 [edit log msg]  
 Sending misc.txt  
 Transmitting file data .  
 Committed revision 6.

And here is an aborted one:

$ svn commit misc.txt  
 [quit editor without saving]  
 Log message unchanged or not specified  
 a)bort, c)ontinue, e)dit:  
 a

1.2.2. Creating a backup of the SVN repository

Backups of the SVN repository can be made with the **rsync** utility. In case of data corruption or other problems on the server, the repository with all revisions, branches and tags can be restored easily. It is recommended to update this backup frequently. The following example creates a folder called bochs-svn-rsync that contains the repository.

rsync -av svn.code.sf.net::p/bochs/code bochs-svn-rsync

1.2.3. Setting SVN commit notifications

The Bochs SVN repository is set up to send a notification email to the "bochs-cvs" mailing list after each successful commit. This email contains the log message, a list of the modified files and a diff against the previous revision. The diff of large commits will be truncated at 96 kByte.

After each commit the SVN server runs the script **post-commit** located in the hooks folder. On SourceForge, this script forces a refresh of the Allura code browser and it can call a script **post-commit-user** for addition operations if it exists. For Bochs we have set up this script and call **svnnotify** from it to create the notification email.

#!/bin/sh  
  
svnnotify --repos-path $1 --revision $2 -O -C -d -e 98304 -t bochs-cvs@lists.sourceforge.net

|  |  |  |
| --- | --- | --- |
| [Prev](http://docs.google.com/resources.html) | [Home](http://docs.google.com/index.html) | [Next](http://docs.google.com/other.html) |
| Resources for developers | [Up](http://docs.google.com/resources.html) | Ideas for other sections |