

# 1 SVN Howto

## 1.1 Description

A short description how to use SVN to access the mod\_perl and related projects source distributions (also applies to the documentation project).

Just as svn access to the Apache development tree, the mod\_perl code pulled from svn is not guaranteed to do anything, especially not compile or work. But, that's exactly why we are using svn, so everyone has access the latest version and can help see to it that mod\_perl does compile and work on all platforms, with the various versions and configurations of Perl and Apache. Patches are always welcome.

It's recommended to subscribe to the modperl-cvs list, which is the place svn commit logs and diffs are mailed to; at least if you're going to work on the code.

## 1.2 Getting SVN

<http://subversion.tigris.org/>

### 1.2.1 *Checking Out*

To checkout a fresh copy of modperl-2.0 run the following commands from the directory you want the sources to stay in:

```
% svn co https://svn.apache.org/repos/asf/perl/modperl/trunk/ mod_perl-2.0
```

After svn has finished downloading the files you will find a new directory called *modperl-2.0* in the current working directory. Now you need to chdir into that directory:

```
% cd modperl-2.0
```

before you can start changing things.

### 1.2.2 *keeping your copy up to date*

To keep your local copy in sync with the repository, do

```
% svn update
```

in the top directory of the project. You should run update every time before you start working on the project.

### 1.2.3 *Sending Patches*

To send a patch, first run:

```
% svn diff
```

in the top directory of the project. The output of diff will be sent to STDOUT, so it might be better to redirect the output to a file:

```
% svn diff > patch
```

If you added files or directories to the project, do a diff against */dev/null*:

```
% diff -u /dev/null newdir/newfilename
```

When this patch is applied, the new dir and the new file will be automatically created.

On Windows-based systems, you can do

```
% diff -u NUL newdir/newfilename
```

instead.

Then send your patch to the maintainer of the project, or the appropriate mailing list.

## 1.3 Inspecting the SVN server with ViewVC

ViewVC is installed on the Apache SVN server. You can reach it at <http://svn.apache.org/viewvc/>.

From there you can browse the list of available projects, look at the files contained in those projects, their logs, and do colored diffs between versions. This is information you can get from your SVN client (through `svn log`, `svn diff` and friends), but the web interface makes it much easier to get a good overview of the different files.

## 1.4 SVN access for mod\_perl committers

mod\_perl committers need to first set it via

<https://svn.apache.org/change-password>

Then the first time you want to commit you will be asked for a password, which is then will be stored in `~/.subversion`. Since all the commits require `https://` checkouts, no password can be sniffed during the commit when it's sent to the server via an encrypted channel.

Before you make your first commit, you should make sure that your `@apache.org` address is subscribed to the SVN commit mailing lists or that its allowed to post to them.

### 1.4.1 Getting the Client

## 1.4.2 Working with SVN

For example let's say you want to work with the *modperl-docs* svn repository.

To check out the repository do:

```
% svn co https://svn.apache.org/repos/asf/perl/modperl/docs/trunk modperl-docs
```

If it's a first time you ssh to *svn.apache.org*, it will ask if you want to allow the host *svn.apache.org*. Answer *yes*. Then you are asked for your password; type it in. Now you will get a freshly checked out copy of the *modperl-docs* repository.

If you get permission problems, most likely your Unix group wasn't adjusted. Contact the person who gave you the svn access.

To bring your repository's copy up to date, run:

```
% svn update
```

or the shortcut:

```
% svn up
```

If you have done some changes, which weren't committed, it's possible that while trying to merge the differences the client will report about collisions which happens when you've happened to change something that was changed and committed by somebody else. You will have to resolve the conflicts by manual editing of the files in question.

Normally, most changes should go through peer review first. It might be a good idea to discuss the intricacies of a change on the appropriate mailing list before committing anything. Then, to commit:

```
% svn commit filename(s)
```

or the shortcut:

```
% svn ci filename(s)
```

But first run `svn update` to avoid any problems with out of date versions of files. If you get any conflicts because of it, these must be changed before doing `svn commit`, which will incorporate any changes into the repository. To commit only a single file, do:

```
% svn commit path/to/file
```

If a file or a directory is not under svn control you have to add it to the svn first and then commit it:

```
% svn add path/to/file
% svn ci path/to/file
```

Unlike CVS, SVN doesn't need the `-kb` option to recognize binary files. So instead of doing:

```
% cvs add -k /path/to/file.jpg
```

just do

```
% svn add /path/to/file.jpg
```

Then, to add it permanently, you will have to commit it.

### ***1.4.3 A Special Note to modperl-docs Committers***

One **very** important note before (ab)using your powers: the `mod_perl` documentation project makes use of an automatic build system. This means that any changes committed will be periodically rebuilt to create the new site (so you don't have to login and do the manual update/rebuild). This is great, but a side-effect of this is that if someone commits anything that doesn't work, the build will fail, and might break parts of the site.

To avoid these problems, please make sure to run `bin/build` on your working copy to test *before* committing. Also, make sure to run `svn update` to check that you have added all files to the repository; it's easy to forget adding the files you have created, and `bin/build` will work fine your side, but will fail for others because of the missing files the build depends on.

### ***1.4.4 Avoiding Typing in the Password***

After awhile you will get tired of typing the password for every `svn` operation that you do. You can avoid that using the public key authentication and the `ssh` authentication agent. Refer to your `ssh` client's manpage for more information. For example for the `openssh` and `ssh` clients, the `ssh-add` utility can be used to enter the password once for your private key and then it'll do the public key authentication for you every time you work with `svn` over `ssh` (as long as `ssh-agent` is running). Refer to the relevant manpage for more info (*ssh-agent* in this case).

## **1.5 mod\_perl and Related Projects on svn.apache.org**

- **modperl**

sources for `mod_perl` 1.0, for use with `apache-1.3`

- **apache-1.3**

the Apache 1.3 HTTP Server

- **modperl-2.0**

the new version of `mod_perl`, for use with `httpd-2.0`. See the install docs for more information about downloading the 2.0 components and installing them.

- **httpd-2.0**

the new Apache 2.0 HTTP Server

- **apr**

needed for modperl-2.0

- **apr-util**

needed for modperl-2.0

- **modperl-docs**

the mod\_perl documentation (i.e. this site). See the documentation download for information on how to download, build and submit patches to the documentation.

Or see <http://svn.apache.org/viewvc/> for a list of all projects.

## 1.6 Maintainers

Maintainer is the person(s) you should contact with updates, corrections and patches.

- the documentation mailing list

## 1.7 Authors

- Thomas Klausner <domm (at) zsi.at>
- Doug MacEachern
- Per Einar Ellefsen <per.einar (at) skynet.be>

Only the major authors are listed above. For contributors see the Changes file.

## Table of Contents:

1	SVN Howto	1
1.1	Description	2
1.2	Getting SVN	2
1.2.1	Checking Out	2
1.2.2	keeping your copy up to date	2
1.2.3	Sending Patches	2
1.3	Inspecting the SVN server with ViewVC	3
1.4	SVN access for mod_perl committers	3
1.4.1	Getting the Client	3
1.4.2	Working with SVN	4
1.4.3	A Special Note to modperl-docs Committers	5
1.4.4	Avoiding Typing in the Password	5
1.5	mod_perl and Related Projects on svn.apache.org	5
1.6	Maintainers	6
1.7	Authors	6