

# Contribute

How to contribute to the mod\_perl community

Last modified Fri Jul 30 11:02:37 2004 GMT

## **Part I: Contribute at the Mailing List**

### **- 1. Contribute at the Mailing List**

The first way to contribute to the mod\_perl community is by following the mailing list and helping out anyone else having problems.

## **Part II: Contribute to the Documentation**

### **- 2. Contribute to the Documentation**

Once you have read the documentation provided for mod\_perl, you will probably have some additions of your own, which are always welcome.

### **- 3. Download the Documentation**

How to download the documentation to work on it locally

### **- 4. How to Contribute to the Documentation**

How to contribute to the mod\_perl documentation: style, tools, etc.

## **Part III: Contribute by Sharing Code**

### **- 5. Contribute by Sharing Code**

Your Perl and mod\_perl know-how is more than welcome as contributions to Apache modules or to mod\_perl itself.

### **- 6. CVS Howto**

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

### **- 7. mod\_perl 2.0 Developer's guide**

If you want to contribute, you should read this documentation to learn mod\_perl 2 coding

# **1 Contribute at the Mailing List**

## 1.1 Description

The first way to contribute to the mod\_perl community is by following the mailing list and helping out anyone else having problems.

## 1.2 Answering Questions

Since mod\_perl is an open source technology, its main support facility are the mailing lists. Therefore it's very essential to support users and developers by sharing your mod\_perl and related knowledge through answering questions and submitting interesting posts that enrich the community's knowledge in the related topics.

You don't have to be a guru to be a very valuable person to the mod\_perl community. Through answering questions you learn a lot by yourself and eventually become a guru too. Actually, by doing a little research for people (which they should have been doing themselves), you'll learn of things you had never even thought of!

## 1.3 Helping Navigating the Documentation

mod\_perl is one of the few open source projects that is bundled with a lot of comprehensive documentation. And although the documentation is well indexed, it can be hard to find the right piece of information, especially for a novice mod\_perl user. If you point someone to the existing documentation, please try to point her to the specific URL containing the information, and not just to "the mod\_perl documentation" in general.

## 1.4 Pointing to Past Discussions

Very often there will come up questions that have been answered before. Instead of replicating answers, do everyone a favor and retrieve the URL of the discussion from the archives, replying with that if it answers the question or at least helps along the way.

## 1.5 Obeying the Email Etiquette

Do not forget the important mailing list guidelines. Whenever posting to the list, you should be aware of these and follow them as best as possible. Furthermore, if some users seem to not be aware of it, don't hesitate about pointing them to that place.

## **2 Contribute to the Documentation**

## 2.1 Description

Once you have read the documentation provided for mod\_perl, you will probably have some additions of your own, which are always welcome.

## 2.2 Submit Corrections

There is a lot of mod\_perl documentation. And we are sure it includes documentation bugs. So if you spot any make sure to report them.

The documentation includes code examples, which sometimes were written without testing, so it's possible that they are broken. If some example doesn't work for you, please let us know.

If you notice bad grammar of English and spelling, don't hesitate to tell us where and what you think is the best way to correct it. Da!

## 2.3 Submit New Articles

If you have some mod\_perl knowledge that is not documented please share it and send us the documentation patch, or a complete new article. This can include installation and configuration scenarios, performance improving tips, debugging techniques and what not.

Developers using mod\_perl constantly work with other technologies related to mod\_perl. If you have spent some time lurking on the mod\_perl lists, you have probably discovered that mod\_perl developers are pros in many other related fields, such as Apache, Perl, relational databases, etc. If you think that the mod\_perl community will find an information about a related technology useful, please submit this information for the inclusion in the documentation.

## 2.4 How to Send Information and Where

The documentation contributions and corrections are to be sent to the mod\_perl documentation mailing list, docs-dev

If the correction is minor, please specify the full URL you are talking about.

If the correction is medium to large use the cvs repository and submit the patch against the latest version, or even better, just send us the modified document. This document explains how to use the cvs repository.

Note that if you for some reason cannot use the CVS repository, each page contains a *src* link in the upper-right corner. From there you can access the POD/HTML source of documents, which you can then modify and send to us.

## **3 Contribute by Sharing Code**

## 3.1 Description

Your Perl and mod\_perl know-how is more than welcome as contributions to Apache modules or to mod\_perl itself.

## 3.2 mod\_perl Patches

If you have modified the mod\_perl source code, improved and extended it, please submit your modifications back to the mod\_perl community so others will be able to enjoy your efforts.

If you are trying to extend mod\_perl, but don't have enough tuits/knowledge to accomplish this by yourself you can ask the mod\_perl developers to help you. But make sure that you clearly explain what you want to do and provide code samples if possible.

## 3.3 Complete Modules

If you have developed a third-party mod\_perl module, consider submitting it to CPAN (<http://cpan.org/>). Other than sharing your works, chances are that the code will receive a lot of attention from other users. This usually leads to more robust code and interface, as it gets reviewed and improved by its users. In some cases new features are submitted as patches, so you benefit from others sharing their works with you.

Before submitting a new `Apache::` module to CPAN, please discuss it first on the mod\_perl list. It's important to choose a good intuitive name for your module. Also it's possible that a module with similar functionality already exists and most likely that someone will point it out. In which case you may want to help developing this existing module instead and integrating your features into it.

## 3.4 Code Snippets

If you have some neat code snippets, please send them to the mailing list. If you are kind enough to annotate these, they could probably fit into mod\_perl documentation. At least they will be archived in the mailing list archive and in the future users will be able to find your code and reuse it.



## **4 CVS Howto**

## 4.1 Description

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

Just as cvs access to the Apache development tree, the mod\_perl code pulled from cvs is not guaranteed to do anything, especially not compile or work. But, that's exactly why we are using cvs, 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 and simply testing the latest snapshots is just as, if not more helpful.

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

## 4.2 Anonymous CVS

You can get cvs here: <http://www.cvshome.org/>

### 4.2.1 Checking Out

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

```
% cvs -d ":pserver:anoncvs@cvs.apache.org:/home/cvspublic" login
```

(use the password "anoncvs")

Next you need to checkout the cvs module you want. e.g. one of modperl-2.0, modperl-docs or modperl. For example modperl:

```
% cvs -d ":pserver:anoncvs@cvs.apache.org:/home/cvspublic" co modperl
```

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

```
% cd modperl
```

before you can start changing things.

### 4.2.2 keeping your copy up to date

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

```
% cvs update -dP
```

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

### 4.2.3 Sending Patches

To send a patch, first run:

```
% cvs diff -u
```

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:

```
% cvs diff -u > 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.

### 4.2.4 Useful Default Parameters

Here are some recommended parameters. Save them in *~/.cvsrc*, so you don't have to type them everytime you use cvs.

```
cvs -z9
update -dP
diff -u
```

- *cvs -z9* sets the compression level to 9 (the highest value) to speed up downloading.
- *update -d* automatically creates directories that are missing in your local copy because they were added to the repository after your initial checkout or your last update.
- *update -P* automatically deletes empty directories.
- *diff -u* to use the unified output format so that your changes can be easily merged back into the repository.

## 4.3 cvsup

cvsup has come out of the FreeBSD group. It's a client/server beast that offers an efficient way to sync collections of files over the net, and it is very CVS aware, allowing synchronisation of repositories or checked out files using the cvs deltas to bring the client side files up to date with minimal data transfer.

For a FreeBSD cvsup client see <http://www.freebsd.org/cgi/ports.cgi?query=cvsup&stype=all>

Others (SunOS, alpha.osf, linux, Solaris2.4, HPAA 10.2, irix): <ftp://ftp.postgresql.org/pub/CVSup/>

Here's a config file for the client (cvsup) to sync modperl sources.

```
*default tag=.
# comment out the above if you want the raw cvs files

*default host=cvs.apache.org
*default prefix=/path/on/this/machine/to/install/
# a subdir for modperl will appear here ^^^

*default base=/path/on/this/machine/where/cvsup/will/keep/status/info
# you'll never need to look in the 'base' dir.

*default release=cvs delete use-rel-suffix compress

modperl
#apache-1.3
#apache-docs
#modperl-2.0
#httpd-2.0
#modperl-docs
#make your picks above by uncommenting the entries
```

## 4.4 Getting CVS snapshots

In case you can't get anonymous CVS access to work (or don't want to), there is another possibility: at <http://cvs.apache.org/snapshots/>, there are snapshots of the interesting CVS repositories you might want to download. These snapshots are extracted from CVS every 6 hours, so might not contain the *latest* changes, but you'll get pretty close. The file names contain the date and time, which you can also see in the directory listing. So just grab the latest one by date, which will get you the latest version.

See the list of interesting repositories below to find out which snapshots you might want to download.

## 4.5 Inspecting the CVS server with ViewCVS

ViewCVS is installed on the Apache CVS server. You can reach it at <http://cvs.apache.org/viewcvs.cgi/>.

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 CVS client (through `cvs log`, `cvs diff` and friends), but the web interface makes it much easier to get a good overview of the different files.

## 4.6 CVS+SSH access for mod\_perl committers

mod\_perl committers need to use CVS over SSH. Normal SSH authentication mechanisms apply; you can use public key, password, etc. Refer to your ssh client's manpage.

### 4.6.1 Getting the Client

If you don't have SSH already installed, you can get it from <http://www.openssh.org/>. If your platform is not supported, this site provides pointers to other implementations. For example for Windows, you might want to look at Putty, <http://www.chiark.greenend.org.uk/~sgtatham/putty/>, especially the plink client which runs from the command line. Also see <http://www.ssh.com/>.

### 4.6.2 Setting Up the Environment

Set your CVS\_RSH environment variable to ssh (if your SSH program has a different name, such as *plink*, use that instead). For example if you are using Bourne-style shell:

```
% export CVS_RSH=ssh
```

You should add this your *.bashrc* or similar file, so that it's set on startup. For Windows, add the line

```
set CVS_RSH=ssh
```

to *autoexec.bat*, or set the environment variable through the System section of the *Control Panel* on NT systems.

The CVS Root is different when using SSH. It's:

```
username@cvs.apache.org:/home/cvs
```

where *username* is your username on the *cvs.apache.org* machine.

### 4.6.3 Working with CVS

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

To check out the repository do:

```
% cvs -d username@cvs.apache.org:/home/cvs checkout modperl-docs
```

or the shortcut:

```
% cvs -d username@cvs.apache.org:/home/cvs co modperl-docs
```

If it's a first time you ssh to *cvs.apache.org*, it will ask if you want to allow the host *cvs.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 cvs access.

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

```
% cvs update -dP
```

or the shortcut:

```
% cvs up -dP
```

The `-d` option builds directories (if any were added since last update), like checkout does. The `-P` option prunes empty directories if any. You can put these and other handy options into the rc file. For example with openssh and ssh clients, this is `~/.cvsrc`. Usually we have the following in this file.

```
cvs -z9
update -dP
diff -u
```

The first line tells to use the best compression level when communicating with the server. The last line will do a unified *diff* when `cvs diff` is used.

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:

```
% cvs commit filename(s)
```

or the shortcut:

```
% cvs ci filename(s)
```

But first run `cvs 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 `cvs commit`, which will incorporate any changes into the repository. To commit only a single file, do:

```
% cvs commit path/to/file
```

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

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

Usually cvs recognizes binary files by their extensions (e.g. images), but if you commit a file with some unusual extension tell your cvs client that it's a binary file with `-kb` option:

```
% cvs add -kb path/to/binary/file
```

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

#### ***4.6.4 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 `cvs 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.

#### ***4.6.5 Avoiding Typing in the Password***

After awhile you will get tired of typing the password for every cvs 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 cvs over ssh (as long as `ssh-agent` is running). Refer to the relevant manpage for more info (*ssh-agent* in this case).

### **4.7 mod\_perl and Related Projects on cvs.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**

#### 4.8 See also

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://cvs.apache.org/viewcvs.cgi/> for a list of all projects.

## 4.8 See also

- <http://httpd.apache.org/dev/anoncvstxt>

For a basic introduction to Anonymous CVS on the Apache CVS server.

- <http://cvsbook.red-bean.com/>

Open Source Development with CVS is a book published by Coriolis Inc. as part of the Coriolis OpenPress series. Chapters 2, 4, 6, 8, 9, and 10 -- comprising a complete introduction, tutorial and reference to CVS -- are being released free under the terms of the GNU General Public License.

- <http://www.cvshome.org/docs/manual/>

Version Management with CVS by Per Cederqvist et al is the "official" manual for CVS. Commonly known as "the Cederqvist," the manual covers repositories, branches, and file maintenance, and includes reference material for both CVS users and CVS repository administrators.

## 4.9 Maintainers

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

- the documentation mailing list

## 4.10 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:

<b>Contribute</b>	1
<b>Contribute at the Mailing List</b>	3
1 Contribute at the Mailing List	3
1.1 Description	4
1.2 Answering Questions	4
1.3 Helping Navigating the Documentation	4
1.4 Pointing to Past Discussions	4
1.5 Obeying the Email Etiquette	4
<b>Contribute to the Documentation</b>	5
2 Contribute to the Documentation	5
2.1 Description	6
2.2 Submit Corrections	6
2.3 Submit New Articles	6
2.4 How to Send Information and Where	6
<b>Contribute by Sharing Code</b>	7
3 Contribute by Sharing Code	7
3.1 Description	8
3.2 mod_perl Patches	8
3.3 Complete Modules	8
3.4 Code Snippets	8
<b>CVS Howto</b>	9
4 CVS Howto	9
4.1 Description	10
4.2 Anonymous CVS	10
4.2.1 Checking Out	10
4.2.2 keeping your copy up to date	10
4.2.3 Sending Patches	11
4.2.4 Useful Default Parameters	11
4.3 cvsup	11
4.4 Getting CVS snapshots	12
4.5 Inspecting the CVS server with ViewCVS	12
4.6 CVS+SSH access for mod_perl committers	13
4.6.1 Getting the Client	13
4.6.2 Setting Up the Environment	13
4.6.3 Working with CVS	13
4.6.4 A Special Note to modperl-docs Committers	15
4.6.5 Avoiding Typing in the Password	15
4.7 mod_perl and Related Projects on cvs.apache.org	15
4.8 See also	16
4.9 Maintainers	16
4.10 Authors	16