

Contribute

How to contribute to the mod_perl community

Last modified Sun Feb 16 01:38:09 2014 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. SVN Howto

A short description how to use SVN 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 svn 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 svn repository.

Note that if you for some reason cannot use the SVN 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 SVN Howto

4.1 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.

4.2 Getting SVN

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

4.2.1 *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.

4.2.2 *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.

4.2.3 *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.

4.3 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.

4.4 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.

4.4.1 Getting the Client

4.4.2 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.

4.4.3 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.

4.4.4 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).

4.5 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.

4.6 Maintainers

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

- the documentation mailing list

4.7 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:

Contribute	1
Contribute at the Mailing List	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
Contribute to the Documentation	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
Contribute by Sharing Code	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
SVN Howto	9
4 SVN Howto	9
4.1 Description	10
4.2 Getting SVN	10
4.2.1 Checking Out	10
4.2.2 keeping your copy up to date	10
4.2.3 Sending Patches	10
4.3 Inspecting the SVN server with ViewVC	11
4.4 SVN access for mod_perl committers	11
4.4.1 Getting the Client	11
4.4.2 Working with SVN	12
4.4.3 A Special Note to modperl-docs Committers	13
4.4.4 Avoiding Typing in the Password	13
4.5 mod_perl and Related Projects on svn.apache.org	13
4.6 Maintainers	14
4.7 Authors	14