

## CHAPTER 23

# Getting Help and Online Resources

In this chapter, we propose a way to solve your `mod_perl`-related problems and provide starting points for information resources related to `mod_perl`.

If you have any problem with `mod_perl` itself, be it a build problem or a runtime problem, you should follow the steps below. But before you follow them, think carefully about whether the problem you are experiencing is `mod_perl`-related. It's quite possible that the problem is in the Perl code, SQL code, Apache itself, or something else entirely. In such cases, you should refer to other resources presented later in this chapter. Remember that although `mod_perl` resources might help you with many related things, they will never be as detailed as resources devoted to the topic at hand.

If you still think that the problem has something to do with `mod_perl`, these are the steps to follow:

1. Try to tackle the problem by yourself for a while. Check that you have the right permissions, that there is enough disk space, etc. Do sanity checks: try to remove the `mod_perl` source tree, unpack it again, and build from fresh.

When trying to figure out what the problem is, always run under single-server mode (`httpd -X`) and always check the `error_log` file.

If you still have problems, proceed to step 2.

2. Reread the documentation (or if you didn't read it yet, do it now). Try to follow the build and usage steps as explained there. This book, *Writing Apache Modules with Perl and C* (O'Reilly), and the documentation distributed with the `mod_perl` sources provide in-depth details on this topic. Also, make sure to read Chapter 22 thoroughly. If you are still in trouble, proceed to step 3.
3. Go to the `mod_perl` list archives (at <http://perl.apache.org/maillist/>) and see whether someone has already reported the same problem. If someone did, chances are that a cure to the problem has been posted to the list, be it a source

patch or a workaround. If after doing an exhaustive search you haven't come up with any solution, proceed to step 4.

Notice that sometimes doing this step before step 2 can be a good idea as well—you may happen to have encountered a well-known bug, and if that's the case doing a quick lookup in the mailing-list archives will save you time and frustration.

4. This step is the last resort. Contact the `mod_perl` mailing list. You should never abuse this step, and use it only when you have already been through the previous three steps. If you ask FAQ questions unnecessarily, chances are that people will not reply to you. And if you ask more FAQ questions, you might get onto people's blacklists and they will not answer your future questions even if they are relevant. Remember that all the answers that you get are coming from volunteers who, instead of having fun outdoors, try to have fun answering challenging questions. FAQ questions aren't challenging, and few people have fun answering them. See more details about `mod_perl` list etiquette in the next section.

It's not enough to just contact the list and ask for help. You have to provide as many details as possible. The next section covers the details you have to provide.

However, don't be afraid. The `mod_perl` mailing list is filled with only nice people who can provide much help and guidance, so if you can't figure something out after having followed the above steps, your question is welcome.

You cannot post to the list without first subscribing to it. To subscribe, send an email to `modperl-subscribe@perl.apache.org`. After you receive a confirmation email, you can start posting to the list. Send your emails to `modperl@perl.apache.org`.

There are other related mailing lists you might want to be on too. See the list of these and subscription instructions in the Resources section of this chapter.

## How to Report Problems

When reporting a problem to the `mod_perl` mailing list, always send these details:

- Anything in the `error_log` file that looks suspicious and possibly related to the problem
- Output of `perl -V`
- Version of `mod_perl`
- Version of Apache
- Options given to `mod_perl`'s `Makefile.PL` file
- Server configuration details
- If `make test` fails, the output of `make test TEST_VERBOSE=1`

Also check whether:

- *make test* passes 100%
- The script works under *mod\_cgi*, if applicable

You should try to isolate the problem and send the smallest possible code snippet that reproduces the problem.

## Getting the Backtrace from Core Dumps

If you get a *core* dump (segmentation fault), send a backtrace if possible. Before you try to produce it, rebuild *mod\_perl* with:

```
panic% perl Makefile.PL PERL_DEBUG=1
```

which will:

- Add *-g* to *EXTRA\_CFLAGS*
- Turn on *PERL\_TRACE*
- Set *PERL\_DESTRUCT\_LEVEL=2* (additional checks during Perl cleanup)
- Link against *libperl*, if it exists

You can read a full explanation in Chapter 21, but here is a summary of how to get a backtrace:

```
panic% cd mod_perl-1.xx
panic% gdb ../apache_1.3.xx/src/httpd
(gdb) run -X -f `pwd`/t/conf/httpd.conf -d `pwd`/t
[now make request that causes core dump]
(gdb) bt
```

In English: *cd* to the *mod\_perl* source directory and start *gdb* with a path to the *httpd* binary, which is located in the Apache source tree. (Of course, replace *x* with real version numbers.) Next, start the *httpd* process from within *gdb* and issue a request that causes a core dump. When the code has died with the *SIGSEGV* signal, run *bt* to get the backtrace.

Alternatively, you can also attach to an already running process like so:

```
panic% gdb httpd <process id number>
```

If the dump is happening in *libperl*, you have to rebuild Perl with *-DDEBUGGING* enabled during the *./Configure* stage. A quick way to this is to go to your Perl source tree and run these commands:

```
panic% rm *.oa]
panic% make LIBPERL=libperl.a
panic% cp libperl.a $Config{archlibexp}/CORE
```

where *\$Config{archlibexp}* is:

```
% perl -V:archlibexp
```

## Spinning Processes

The *gdb* attaching to the live process approach is helpful when debugging a *spinning* process. You can also get a Perl stack trace of a spinning process by installing a `$SIG{USR1}` handler in your code:

```
use Carp ();
$SIG{USR1} = \&Carp::confess;
```

While the process is spinning, send it a USR1 signal:

```
panic% kill -USR1 <process id number>
```

and the Perl stack trace will be printed.

Alternatively, you can use *gdb* to find which Perl code is causing the spin:

```
panic% gdb httpd <pid of spinning process>
(gdb) where
(gdb) source mod_perl-x.xx/.gdbinit
(gdb) curinfo
```

After loading the special macros file (*.gdbinit*), you can use the *curinfo* *gdb* macro to figure out the file and line number in which the code stuck. Chapter 21 talks in more detail about tracing techniques.

Finally, send all these details to [modperl@perl.apache.org](mailto:modperl@perl.apache.org).

## Mailing List Etiquette

Like any community, the `mod_perl` mailing list has its own rules of etiquette that you would be wise to avoid violating:

- Never contact people in person to ask a question unless they have explicitly given you permission. Even if someone was kind enough to reply to a previous question, this doesn't mean he wants to be your go-to person for every subsequent problem as well. If you do this, don't be surprised if your question is ignored. Just think about how many emails these people receive daily, and you will understand the reason. Remember that this is a voluntary effort, not a technical support service.
- If a reply to your question is posted to the list and you want to follow up on it, in most cases you should keep posting to the list, so the conversation will be saved in the mailing-list archives and can later be reused by other users who seek help in the archives.
- However, if you receive a private email reply to the question, keep the conversation private, because the person who has answered you might not have wanted his answer to be seen in public. You have to respect that and not resend the reply to the list without this person's permission.

- When posting to the list, always use relevant subject lines. Don't just say "help" in the subject field of your post. Chances are that these messages will be ignored. Most of the people are interested in only specific topics, and therefore they will delete messages with unspecific subject lines without even reading them. To catch their attention, you should provide a concise, meaningful subject line.
- When replying to a message, please try to quote only relevant parts of the original post: don't overquote and don't overtrim. Refrain from replying on the top of the original message, since it makes it hard for other users to understand the conversation. Please use proper quoting delimiters, so users can easily tell your reply from the original message.
- If your English is not fluent, do not feel frightened to post. The `mod_perl` community includes many people for whom English is not their primary language. But please run a spell-checker before posting if you know that you tend to make many mistakes. Sometimes people post questions that are never answered simply because nobody understands the question.
- Avoid posting off-topic (not `mod_perl`-related) questions. If you really feel that you have to, at least let others know that the post is off-topic. The correct way to do that is to start your post's subject field with the `[OT]` tag.
- Avoid flaming. At least, don't flame in public—contact others in person if you really want to. Flaming people in public may hurt their feelings. They might leave the list, and all of us will lose an active (or potentially active) contributor. We try hard to make the `mod_perl` list a fun place to be.
- Remember that sometimes it might take days or even weeks before your question is answered, although during the working week most questions are answered within a few hours. Occasionally, questions aren't answered at all. If this is the case, you might want to post again some time later (at least one week), maybe with more information.
- Finally, use common sense when posting, and you will be fine. Online conversations needn't be any different than real-life ones; be polite and precise and everybody will be happy. Subscribing to the list and spending some time reading the posts will give you an idea of how things are done.

## Resources

This section includes centralized resources that you should find useful when you work with `mod_perl` and related technologies, such as Apache, Perl, CGI, CVS, Squid, DBI, SQL, Security, etc.

### `mod_perl`

- `mod_perl` home page: <http://perl.apache.org/>
- `mod_perl` documentation: <http://perl.apache.org/docs/>

- mod\_perl books

- Writing Apache Modules with Perl and C, by Lincoln Stein and Doug MacEachern (O'Reilly)

<http://www.modperl.com> is the home site for this book, which is about creating web server modules using the Apache API. You absolutely must have this book if you plan to use mod\_perl for anything other than speeding up plain CGI scripts. It will teach you the mod\_perl API and provide lots of examples to learn from. This book is also very useful for developers who write Apache modules in C.

- *The mod\_perl Developer's Cookbook*, by Geoffrey Young, Paul Lindner, and Randy Kobes (Sams)

<http://www.modperlcookbook.org/> is the home site of this book, which will save you a lot of precious development time. It provides out-of-box solutions to pretty much any problem or challenge you may encounter while developing mod\_perl applications. Every solution is followed by an in-depth discussion, helping you understand how the solution works and making it easy to adjust the provided code to your particular situation.

- *mod\_perl Pocket Reference*, by Andrew Ford (O'Reilly)

<http://www.oreilly.com/catalog/modperlpr/> is the home site of this book.

You should probably also get the *Apache Pocket Reference*, by the same author and the same publisher: <http://www.oreilly.com/catalog/apachepr/>.

See also Andrew's collection of reference cards for Apache and other programs: <http://www.refcards.com/>.

- There are a few good books that cover technologies that deploy mod\_perl. Among them are *Embedding Perl in HTML with Mason*, by Dave Rolsky and Ken Williams (O'Reilly), available from <http://www.masonbook.com/>; and *Running Weblogs with Slash*, by chromatic, Brian Aker, and David Krieger (O'Reilly). To see an updated list of books, please refer to <http://perl.apache.org/docs/offsite/books.html>.

## mod\_perl Mailing Lists

- The mod\_perl mailing list

The Apache/Perl mailing list is available for mod\_perl users and developers to share ideas, solve problems, and discuss things related to mod\_perl and the Apache::\* modules. To subscribe to this list, send an empty email to [modperl-subscribe@perl.apache.org](mailto:modperl-subscribe@perl.apache.org). To unsubscribe, send email to [modperl-unsubscribe@perl.apache.org](mailto:modperl-unsubscribe@perl.apache.org). Send email to [modperl@perl.apache.org](mailto:modperl@perl.apache.org) to post to the list.

To subscribe to the digest list, send email to [modperl-digest-subscribe@perl.apache.org](mailto:modperl-digest-subscribe@perl.apache.org).

The searchable mod\_perl mailing-list archives are available at <http://mathforum.org/epigone/modperl/>. Thanks to Ken Williams for this.

The following archives are also available:

<http://www.geocrawler.com/lists/3/web/182/0/>  
<http://www.mail-archive.com/modperl%40apache.org/>  
<http://www.davin.ottawa.on.ca/archive/modperl/>  
<http://marc.theaimsgroup.com/?l=apache-modperl>  
<http://www.egroups.com/group/modperl/>

- The mod\_perl development mailing list

This list is for discussions about the development of the core mod\_perl. To subscribe, send an empty email to [dev-subscribe@perl.apache.org](mailto:dev-subscribe@perl.apache.org). To unsubscribe from the list, send an empty email to [dev-unsubscribe@perl.apache.org](mailto:dev-unsubscribe@perl.apache.org). To get help with the list, send an empty email to [dev-help@perl.apache.org](mailto:dev-help@perl.apache.org).

The list's searchable archives are:

<http://mathforum.org/epigone/modperl-dev/>  
<http://marc.theaimsgroup.com/?l=apache-modperl-dev&r=1&w=2#apache-modperl-dev>  
<http://www.mail-archive.com/dev%40perl.apache.org/>

- The mod\_perl documentation mailing list

This mailing list is for discussing the development of the mod\_perl documentation and site. To subscribe, send an empty email to [docs-dev-subscribe@perl.apache.org](mailto:docs-dev-subscribe@perl.apache.org). To unsubscribe from the list, send an empty email to [docs-dev-unsubscribe@perl.apache.org](mailto:docs-dev-unsubscribe@perl.apache.org). To get help with the list, send an empty email to [docs-dev-help@perl.apache.org](mailto:docs-dev-help@perl.apache.org).

The list has a searchable archive at <http://mathforum.org/epigone/modperl-docs-dev/>.

- The Apache test framework development mailing list

The *test-dev* list is the list where the Apache HTTP Test project is discussed.

To subscribe, send an empty email to [test-dev-subscribe@httpd.apache.org](mailto:test-dev-subscribe@httpd.apache.org). To unsubscribe from the list, send an empty email to [test-dev-unsubscribe@httpd.apache.org](mailto:test-dev-unsubscribe@httpd.apache.org). To get help with the list, send an empty email to [test-dev-help@httpd.apache.org](mailto:test-dev-help@httpd.apache.org).

The list has a searchable archive at <http://www.apachelabs.org/test-dev/>.

- The advocacy mailing list

The list for mod\_perl advocacy issues, discussions about sites, etc.

To subscribe send an empty email to [advocacy-subscribe@perl.apache.org](mailto:advocacy-subscribe@perl.apache.org). To unsubscribe from the list, send an empty email to [advocacy-unsubscribe@perl.apache.org](mailto:advocacy-unsubscribe@perl.apache.org). To get help with the list, send an empty email to [advocacy@perl.apache.org](mailto:advocacy@perl.apache.org).

The list has a searchable archive at <http://www.mail-archive.com/advocacy@perl.apache.org/>.

- The *modperl-cvs* mailing list

The `mod_perl` CVS list is the list where you can watch `mod_perl` getting patched. No real discussions happen on this list, but if you want to know about the latest changes in the `mod_perl` core before everyone else, this is the list to be on.

To subscribe, send email to [modperl-cvs-subscribe@perl.apache.org](mailto:modperl-cvs-subscribe@perl.apache.org). To unsubscribe send email to [modperl-cvs-unsubscribe@perl.apache.org](mailto:modperl-cvs-unsubscribe@perl.apache.org). Send email to [modperl-cvs@perl.apache.org](mailto:modperl-cvs@perl.apache.org) to post to the list.

The list is archived at <http://marc.theaimsgroup.com/?l=apache-modperl-cvs&r=1&w=2#apache-modperl-cvs>.

## Perl

The following resources are available for Perl:

- Books:
  - *Programming Perl*, Third Edition, by Larry Wall, Tom Christiansen, and Jon Orwant (O'Reilly)
  - *The Perl Cookbook*, by Tom Christiansen and Nathan Torkington (O'Reilly)
  - *Effective Perl Programming*, by Joseph Hall (Addison Wesley)
  - *Web Client Programming with Perl*, by Clinton Wong (O'Reilly)
- The Perl FAQ: <http://www.perl.com/language/faq/>
- The Perl home pages: <http://www.perl.com/> and <http://www.perl.org/>
- The Perl Journal: <http://www.tpj.com/>
- The Perl Review: <http://www.theperlreview.com/>
- Perl Monks: <http://www.perlmonks.org/>
- Searchable Perl documentation: <http://www.perldoc.com/>
- Perl Module Mechanics: [http://world.std.com/~swmcd/steven/perl/module\\_mechanics.html](http://world.std.com/~swmcd/steven/perl/module_mechanics.html)

This page describes the mechanics of creating, compiling, releasing, and maintaining Perl modules
- Perl news: <http://use.perl.org/>
- Searchable CPAN: <http://search.cpan.org/>
- Perl mailing lists: <http://lists.perl.org/>



## Perl/CGI

The following resources are valuable for learning more about writing CGI scripts with Perl:

- *The Official Guide to CGI.pm*, by Lincoln Stein (John Wiley & Sons)
- *CGI/Perl Cookbook*, by Craig Patchett and Matthew Wright (John Wiley & Sons)
- *CGI Programming with Perl*, Second Edition, by Scott Guelich, Shishir Gundavaram, and Gunther Birznieks (O'Reilly)

Here are some resources on the Web you might find useful:

*Answers to Some Troublesome Perl and Perl/CGI Questions*

<http://stason.org/TULARC/webmaster/myfaq.html>

*Idiot's Guide to CGI Programming*

[http://www.webdeveloper.com/cgi-perl/cgi\\_idiots\\_guide\\_to\\_perl.html](http://www.webdeveloper.com/cgi-perl/cgi_idiots_guide_to_perl.html)

*WWW Security FAQ*

<http://www.w3.org/Security/Faq/www-security-faq.html>

*CGI/Perl Taint Mode FAQ*

<http://www.gunther.web66.com/FAQS/taintmode.html> (by Gunther Birznieks)

*cgi-list Mailing List*

Send email to [majordomo@jann.com](mailto:majordomo@jann.com) with body:

subscribe cgi-list

*CGI Newsgroup*

[comp.infosystems.www.authoring.cgi](http://comp.infosystems.www.authoring.cgi)

## Apache

The following resources are useful for learning more about Apache:

- Apache Software Foundation home: <http://www.apache.org/>
- Apache *httpd* server: <http://httpd.apache.org/>
- Apache mailing lists: <http://www.apache.org/foundation/maillinglists.html> contains a comprehensive list of all Apache projects' mailing lists
- Apache quick reference card: <http://www.refcards.com/> (other reference cards are also available from this link)
- The Apache FAQ: <http://httpd.apache.org/docs/misc/FAQ.html>
- Apache server documentation: <http://httpd.apache.org/docs/> for 1.3.xx, <http://httpd.apache.org/docs-2.0/> for 2.0
- Apache handlers in C: <http://httpd.apache.org/docs/handler.html>
- mod\_rewrite Guide: <http://www.engelschall.com/pw/apache/rewriteguide/>
- Security

- “Security and Apache: An Essential Primer,” by Ken Coar: <http://linuxplanet.com/linuxplanet/print/1527/>
- “Using Apache with Suexec on Linux,” by Ken Coar: <http://linuxplanet.com/linuxplanet/print/1445/>
- The Unix *chroot* jail facility
  - “How to ‘chroot’ an Apache tree with Linux and Solaris”: <http://penguin.epfl.ch/chroot.html>
  - “Installing and Securing the Apache Webserver with SSL,” by Dale Coddington: <http://online.securityfocus.com/infocus/1356/>
  - “How to break out of a chroot() jail”: <http://www.bpfh.net/simes/computing/chroot-break.html>
- The FreeBSD jail facility:
  - Jails: Confining the omnipotent root,” by Paul-Henning Kamp and Robert N. M. Watson: <http://docs.freebsd.org/44doc/papers/jail/jail.html>
  - Chapter 12 of FreeBSD Developers' Handbook, by Evan Sarmiento: [http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/developers-handbook/jail.html](http://www.freebsd.org/doc/en_US.ISO8859-1/books/developers-handbook/jail.html)
- `mod_throttle_access`: <http://www.fremen.org/apache/>
- Books:
  - *How to Set Up and Maintain a Web Site: The Guide for Information Providers*, Second Edition, by Lincoln Stein (Addison Wesley)
  - *Apache: The Definitive Guide*, Second Edition, by Ben Laurie and Peter Laurie (O'Reilly)
  - *Apache Server for Dummies*, by Ken Coar (IDE)

## DBI and SQL

The following resources are useful for questions on DBI and SQL:

- Introduction to Structured Query Language: <http://www.dbbm.fiocruz.br/class/Lecture/d17/sql/jhoffman/sqltut.html>
- “SQL for Web Nerds,” by Philip Greenspun: <http://www.arsdigita.com/books/sql/>
- DBI Examples and Performance Tuning, by Jeffery Baker: <http://www.saturn5.com/~jwb/dbi-examples.html>
- DBI home page: <http://dbi.perl.org/>
- DBI mailing-list information: <http://www.fugue.com/dbi/>
- DBI mailing-list archives: <http://www.bitmechanic.com/mail-archives/dbi-users/> and <http://www.xray.mpe.mpg.de/mailling-lists/dbi/>

## Squid

- Home page: <http://www.squid-cache.org/>
- FAQ: <http://www.squid-cache.org/Doc/FAQ/FAQ.html>
- Users guide: <http://www.squid-cache.org/Doc/Users-Guide/>
- Mailing lists: <http://www.squid-cache.org/mailling-lists.html>

## CVS

- CVS instructions for access to the mod\_perl repository: [http://perl.apache.org/contribute/cvs\\_howto.html](http://perl.apache.org/contribute/cvs_howto.html)
- Open source development with CVS: <http://cvsbook.red-bean.com/>
- Online documents: <http://www.cvshome.org/docs/>
- CVS quick reference card: <http://www.refcards.com/about/cvs.html>

## Performance and Scalability

- “Techniques and Technologies for Scaling Internet Services” mailing list: [scalable@arctic.org](mailto:scalable@arctic.org). Subscribe by sending a message to [scalable-subscribe@arctic.org](mailto:scalable-subscribe@arctic.org).
- “Solaris 2.x—Tuning Your TCP/IP Stack and More”: <http://www.sean.de/Solaris/tune.html>

This page talks about the TCP/IP stack and various tricks of tuning your system to get the most out of it as a web server. While the information is for the Solaris 2.x OS, most of it is relevant to other Unix flavors. At the end, an extensive list of related literature is presented.

## Web Security

- *Web Security: A Step-by-Step Reference Guide*, by Lincoln Stein (Addison Wesley)
- *Web Security and Electronic Commerce*, by Simpson Garfinkle with Gene Spafford (O’Reilly)
- Chapter 13 of *Apache: The Definitive Guide*, Second Edition, by Ben Laurie and Peter Laurie (O’Reilly) talks extensively about the Apache configuration process