Documentation

The mod_perl project features a lot of documentation, both for mod_perl 1.0 and 2.0. If there is anything you need to learn about mod_perl, you'll learn it here.

Last modified Sat Dec 6 12:18:53 2003 GMT

Table of Contents:

- 1. Conventions used in the mod_perl Documentation
  We use a number of conventions in this documentation, that are mostly easy to understand; if you're in doubt, look here for the explanation.
- 2. mod_perl 1.0 Documentation
  A collection of the documents specific to the mod_perl 1.0 generation.
- 3. mod_perl 2.0 Documentation
  A collection of the documents specific to the mod_perl 2.0 generation.
- 4. General Documentation
  Here you can find documentation concerning mod_perl in general, but also not strictly mod_perl related information that is still very useful for working with mod_perl. Most of the information here applies to mod_perl 1.0 and 2.0.
- 5. Tutorials
  mod_perl related tutorials, teaching you things not only about mod_perl, but also about any related topics of great interest to mod_perl programmers.
- 6. Offsite Resources
  mod_perl Books, Articles, Presentations, and links to sites covering other relevant topics.

1 Conventions used in the mod_perl Documentation

1.1 Description

We use a number of conventions in this documentation, that are mostly easy to understand; if you're in doubt, look here for the explanation.

1.2 Prompts

For commands that should be typed in your shell (or your *Command prompt*, or whatever your OS calls it), we use the following prompts:

```
% ls -l
```

for the user-mode prompt (ie. a normal user account, with no special privileges).

```
# ls -l
```

for the superuser prompt: this means you'll have to change users to become the super user on your platform. On Unix you can use the `su` or `sudo` utilities to gain superuser privileges (provided you know the *root* password); on other platforms you might have to change the user -- to *Administrator* for example on Windows.

If you cannot obtain super user privileges, there will often be explanations about how doing the selected task without those privileges; in any other case, contact your system administrator.

On documents specific to a certain Operating System, the prompt might change. For example, in Windows documentation, we might use:

```
C:\> nmake
```

instead of any other prompt.

1.3 Typographical conventions

We try to be consisten about our use of different fonts and faces, so that you'll recognize special words more easily.

- *Italic*

  is used for file names or things you have to replace.

- `Constant width`

  is used for commands/program names, configuration items or Perl code/function names, and manpage references.

- **Bold**

  is used for things we want to stress.

Table of Contents:

Documentation 1
Conventions used in the mod_perl Documentation 3
1 Conventions used in the mod_perl Documentation 3
1.1 Description 4
1.2 Prompts 4
1.3 Typographical conventions 4