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 Sun Feb 16 01:37:24 2014 GMT
- 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 consistent about our use of different fonts and faces, so that you’ll recognize special words more easily.

- Use F<filename> for filenames, directories/paths, URIs, and the like.
- Use I<italics> for emphasizing things. But use them with care, when things really need to be emphasized.
- Use B<stress> for stressing things more strongly than I<italics> does. But avoid using this tag unless you think things are very important. Defer to I<italics> for emphasis, instead. Over-use of bold text reduces its original intention -- so use it with care to really make things stand out when they need to stand out.
• Use **Constant width** 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:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Conventions used in the mod_perl Documentation</strong></td>
<td>3</td>
</tr>
<tr>
<td>1 Conventions used in the mod_perl Documentation</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Description</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Prompts</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Typographical conventions</td>
<td>4</td>
</tr>
</tbody>
</table>