

以非超级用户身份安装mod\_perl PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/167/2021\\_2022\\_\\_E4\\_BB\\_A5\\_E9\\_9D\\_9E\\_E8\\_B6\\_85\\_E7\\_c103\\_167238.htm](https://www.100test.com/kao_ti2020/167/2021_2022__E4_BB_A5_E9_9D_9E_E8_B6_85_E7_c103_167238.htm) 就像大家从以前的文章里看到的，mod\_perl 使Apache由两个组件组成：  
Perl 模块和 Apache 本身。虽然以非超级用户权限安装Apache很简单，但是应该知道如何安装在非系统位置的Perl模块。本文将论证几种方法来实现这个任务。在这些例子里，我将用stas作为用户名，/home/stas作为该用户的私人目录。安装Perl模块到选择的目录 既然不用超级用户权限，那么你就不允许安装到系统目录，象/usr/lib/perl5。你需要找到如何在你的私人目录安装这些模块的方法。这很容易。  
首先，你需要决定安装这些模块到什么地方。最简单的途径就是在你的私人目录下模仿与Perl有关的根目录文件系统的移植。实际上，我们只需要两个目录：  
/home/stas/bin /home/stas/lib 既然他们会在第一个模块被安装时自动创建，那么我们就没有必要创建它们。99%的文件将被拷贝到lib目录下。有时候，当一些模块随着Perl脚本发布的时候会被拷贝到bin目录。如果目录不存在会被创建。让我们安装CGI.pm包，该包包含了一些其它CGI::\*模块。通常，从CPAN库下载该包，解开该包并进入新建的目录里。现在做一个标准的perl文件Makefile.PL去准备Makefile.但是这个时候让 MakeMaker 用你的Perl安装目录，而不是缺省的。  
% perl Makefile.PL PREFIX=/home/stas  
PREFIX=/home/stas 只是安装过程不同于通常的一部分。  
注意如果你不在意MakeMaker如何选择剩下的目录，或

者如果你正在用一个需要显式指明所有目的目录的老版本，那么这样做：

```
nbsp. % nbsp.perl nbsp.Makefile.PL
nbsp.PREFIX=/home/stas nbsp.
INSTALLPRIVLIB=/home/stas/lib/perl5 nbsp.
INSTALLSCRIPT=/home/stas/bin nbsp.
INSTALLSITELIB=/home/stas/lib/perl5/site_perl nbsp.
INSTALLBIN=/home/stas/bin nbsp.
INSTALLMAN1DIR=/home/stas/lib/perl5/man nbsp.
INSTALLMAN3DIR=/home/stas/lib/perl5/man3
```

剩下的跟通常一样：

```
nbsp. % nbsp.make % nbsp.make nbsp.test % nbsp.make
nbsp.install make nbsp.install nbsp.
```

安装所有的文件到私人目录。注意所有丢失的目录会被自动创建，因此没有必要创建它们。以下就是这么做的（稍微改动过）：

```
nbsp.正在安装
nbsp./home/stas/lib/perl5/CGI/Cookie.pm 正在安装
nbsp./home/stas/lib/perl5/CGI.pm 正在安装
nbsp./home/stas/lib/perl5/man3/CGI.3 正在安装
nbsp./home/stas/lib/perl5/man3/CGI::Cookie.3 正在写入
nbsp./home/stas/lib/perl5/auto/CGI/.packlist 正在追加安装信息
到 nbsp./home/stas/lib/perl5/perllocal.pod
```

如果你不得不用显式目的参数，那么不用单个PREFIX参数，你会发现建立一个被调用的文件很有用。例如，

```
nbsp./perl_dirs nbsp.
```

(这里 nbsp.在我们的例子中是 nbsp./home/stas) nbsp.的内容包含：

```
nbsp.
PREFIX=/home/stas nbsp.
INSTALLPRIVLIB=/home/stas/lib/perl5
nbsp.
INSTALLSCRIPT=/home/stas/bin nbsp.
INSTALLSITELIB=/home/stas/lib/perl5/site_perl nbsp.
INSTALLBIN=/home/stas/bin nbsp.
```

INSTALLMAN1DIR=/home/stas/lib/perl5/man

INSTALLMAN3DIR=/home/stas/lib/perl5/man3 从现在起，任何

时候你想本地化安装Perl模块就可以这样执行：

`perl Makefile.PL`cat ./perl_dirs``

`make`

`make test`

`make install` 用这个方法，你

可以很容易地保持不同Perl模块的存储。例如，你可以一个用

作production Perl，另一个用作开发。

`perl Makefile.PL`cat ./perl_dirs.production`` 或者

`perl Makefile.PL`cat ~/.perl_dirs.develop``

编写你自己的脚本找到本地已安装模块 这些Perl模块通常都

放在四个主要目录。为了找到这些目录，执行：

`perl -V` 输出包含重要的perl安装信息。最后你将看到：

Characteristics of this binary (from

libperl): Built under linux Compiled at

Apr 6 1999 23:34:07 @INC:

/usr/lib/perl5/5.00503/i386-linux /usr/lib/perl5/5.00503

/usr/lib/perl5/site\_perl/5.005/i386-linux

/usr/lib/perl5/site\_perl/5.005 . 这向我们表明Perl特殊变量@INC

的内容，这个特殊变量通常被Perl用来查找模块的。它等价

于Unix shells里的PATH环境变量，是用来查找可执行程序

的。注意Perl也在.目录（代表当前目录）查找模块，这是

在以上输出目录里的最后的入口路径。当然，这个例子来

自5.00503版本的perl，它安装在我的x86结构PC的Linux中

。这就是为什么你看到了i386-linux和5.00503。

如果你的系统运行着不同版本的perl、操作系统、处理器或者芯片结构，

那么一些目录会有不同的名字。我也安装了perl 5.6.1版

本到 `/usr/local/lib/`，所以当我这么做：`%`  
`/usr/local/bin/perl5.6.1 -V` 你会看到：`@INC:`  
`/usr/local/lib/perl5/5.6.1/i586-linux` `/usr/local/lib/perl5/5.6.1`  
`/usr/local/lib/site_perl/5.6.1/i586-linux` `/usr/local/lib/site_perl` 注意  
这还是Linux, 但是较新的perl版本用了奔腾的处理器（所以是i586而不是i386）。这对奔腾处理器的编译优化有用，当  
二进制Perl扩展被创建时。所有平台指定的文件，象已编译  
的C文件用XS或者SWIG去粘合Perl, 应该进入类似i386-linux目  
录。重点：当我们已经安装了Perl模块进入非标准目录，我  
们不得不让Perl知道去哪里搜索这四个目录。有两个实现方法  
：你可以设置PERL5LIB 环境变量或者你可以在你的脚本  
中修改@INC变量。假设我们用Perl 5.00503版本，在我们的  
例子中目录如下：

`/home/sbekman/lib/perl5/5.00503/i386-linux`

`/home/sbekman/lib/perl5/5.00503`

`/home/sbekman/lib/perl5/site_perl/5.005/i386-linux`

`/home/sbekman/lib/perl5/site_perl/5.005` 就象前面提到的，你能  
通过`perl -V`找到确切的目录并用你自己的私人目录代替  
全局的perl安装基目录。修改@INC 是很容易的。  
最好的途径是用lib模块(编译指示), 通过在你的脚本最上  
方加入以下代码片段，这些代码需要本地化已经安装的模块

：`use lib qw(/home/stas/lib/perl5/5.00503/`

`/home/stas/lib/perl5/site_perl/5.005)`. 另一个途径是写代码显式

修改@INC：`BEGIN { unshift @INC,`

`qw(/home/stas/lib/perl5/5.00503`

`/home/stas/lib/perl5/5.00503/i386-linux`

/home/stas/lib/perl5/site\_perl/5.005

/home/stas/lib/perl5/site\_perl/5.005/i386-linux). } 注意既然它们如果存在会自动加入（确切地说，当\$dir/\$archname/auto存在的时候），那么用lib模块我们不必列出一致的结构指定目录。

而且，注意到两种途径都预先考虑到@INC里被搜索的目录。

这就允许你安装一个更新的模块加入你的本地库。Perl会用这个模块取代已经安装在系统库里的旧模块。两种途径都是在

编译期间修改 @INC 的值。这个lib模块也用BEGIN块，但是是内部地。现在，让我假设如下情景。我已经在我的本地库中安装了

perl.LWP 包。现在我想安装另一个模块(例如 perl.mod\_perl) ，这个模块有在它的首选列表中列出的

perl.LWP。我知道我已经安装 perl.LWP ，但是当我为了准备安装的模块运行perl Makefile.pl，我被告知我没有安装LWP。

Perl没有办法知道我们有本地已经安装的模块。所有这些都在@INC里列出的目录搜索中。既然后者包含只有4个缺省目录（加上 perl..目录），那么它不可能找到本地安装的

perl.LWP 包。我们不能通过加入代码修改@INC来解决这个问题，而要改变PERL5LIB环境变量才能解决。

如果你正在用tcsh解决交互工作，那么这样做：

```
perl. setenv perl.PERL5LIB perl./home/stas/lib/perl5/5.00503:
```

```
/home/stas/lib/perl5/site_perl/5.005 它应该是带目录的单行，并且目录是被冒号（：）和非空格所分开。如果你是bash用户
```

```
，那么这么做： perl. export
```

```
perl.PERL5LIB=/home/stas/lib/perl5/5.00503:
```

```
/home/stas/lib/perl5/site_perl/5.005 同样，写成单行。如果你
```

```
用bash，那么你能通过反斜杆（\）来输入多行命令，象这样
```

: export PERL5LIB=/home/stas/lib/perl5/5.00503:  
/home/stas/lib/perl5/site\_perl/5.005 由于用lib，所以如果存在  
，Perl会自动预先考虑结构指定目录到 @INC。当我们这么做的时候，我们就要象以前那样通过执行 perl -V 来确认 @INC 新配置的值。你应该看到 @INC 修改的值：  
% perl -V Characteristics of this binary (from libperl): Built under linux  
Compiled at Apr 6 1999 23:34:07  
%ENV: PERL5LIB= "/home/stas/lib/perl5/5.00503:/home/stas/lib/perl5/site\_perl/5.005" @INC:  
/home/stas/lib/perl5/5.00503/i386-linux  
/home/stas/lib/perl5/5.00503  
/home/stas/lib/perl5/site\_perl/5.005/i386-linux  
/home/stas/lib/perl5/site\_perl/5.005  
/usr/lib/perl5/5.00503/i386-linux /usr/lib/perl5/5.00503  
/usr/lib/perl5/site\_perl/5.005/i386-linux  
/usr/lib/perl5/site\_perl/5.005 . 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)