

第四部分

附录

附录 安装模块

附录 安装模块

在Perl中安装模块并不困难，如果你想真正掌握 Perl，那么学会如何安装这些模块是非常重要的。本附录包含了关于如何安装你需要的模块的信息。



在Perl的文档资料中，你可以得到在各种操作系统下安装模块的详细说明。名叫“Perlmodinstall”的文档甚至包含了在OS/2和VMS之类的操作系统下安装模块的说明。

A.1 选择正确的模块

首先，必须选择正确的模块。可以通过站点 <http://www.perl.com> / CPAN上的CPAN寻找你要的模块。你必须确定对哪个模块感兴趣。

CPAN模块大体上是按它们的功能来命名的。例如，Image::size带有一个图形，并且能够报告该图形的大小，该模块可用来与 Web页一道运行。不过，有些模块使用一些特殊的名字。LWP是根据Perl库libwww-perl而得名的。

还可以在CPAN上找到模块包。这些模块包含有若干相关的模块，这些模块通常是一些必须要有的模块，它们全部放在一个大模块包中。例如，libnet模块包可以像一个模块那样来安装，不过在安装过程中，你会得到若干个与网络相关的模块。LWP就是libnet模块包中的一部分。



当你安装一个模块时，还会自动获得该模块需要的所有文档。

A.2 在何种操作系统下安装

在下面各节中的每个安装模块的例子中，你将安装来自 CPAN的Date::Manip模块。若要安装你自己的模块或模块包，只要用你的模块包取代 Date::Manip即可。

A.2.1 在Windows95 / 98 / NT下安装

在Windows下，假定已经安装了来自 ActiveState Tool 公司的Perl，安装模块的最容易的方法是使用ActiveState Tool公司已打包的模块。

若要在Windows下安装预装模块，首先必须启动 Perl Package Manager (PPM)。该实用程序通过提供一个用于模块安装的交互式界面，从而简化了模块安装进程。为了启动 PPM，你必须显示一个DOS命令提示符，如图A-1所示，应该连接到Internet。

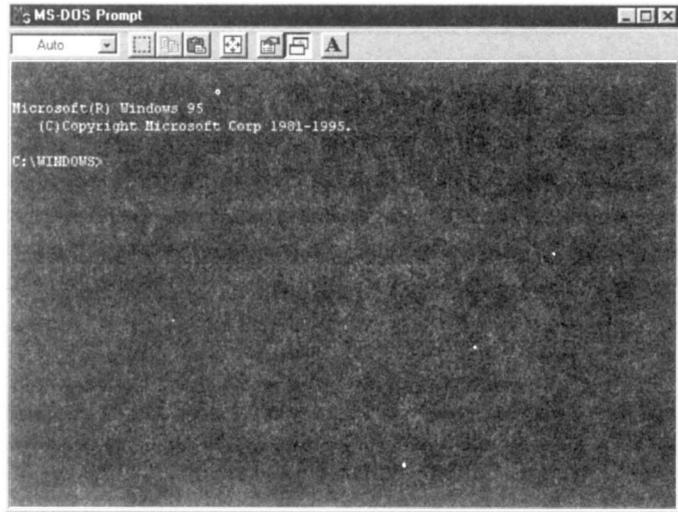
在命令提示符处，键入如下所示的 PPM。这时PPM实用程序应该启动运行。如果它没有启动运行，你必须查找与 ActiveState Perl一道安装的ppm.bat文件，并用全路径名运行它：

```
C:\Windows>ppm
PPM interactive shell (1.0.0) - type 'help' for available commands
PPM>
```

若要搜索某个模块，请使用下面所示的 search 命令。之所以你必须使用该命令，原因是 ActiveState 并没有 CPAN 中的所有模块的预装模块包，它只有比较常用的模块。另外，为了进行安装，必须正确拼写模块名。

```
PPM> search Date
Packages available from http://www.ActiveState.com/packages:
    Date-Calc
    Date-Manip
    TimeDate
PPM>
```

图A-1 可以从 DOS 命令提示符处开始安装模块的操作



当你找到你想要的模块（比如这个例子中的 Date-Manip）后，就可以使用 install 命令，对该模块进行安装，如下所示：

```
PPM> install Date-Manip
Install package 'Date-Manip?' (y/N): y
Installing C:\Perl\html\lib\Date\Manip.html
Installing C:\Perl\htmlhelp\pkg-Date-Manip.chm
Installing C:\Perl\htmlhelp\pkg-Date-Manip.hhc
Installing C:\Perl\site\lib\Date\Manip.pm
Writing C:\Perl\site\lib\auto\Date-Manip/.packlist
PPM>
```

这时 Date::Manip 模块就安装好了。

如果你想下载模块包并用人工进行安装（也许 PC 没有与 Internet 连网或者它位于防火墙的后面），可以在 ActiveState 的 Web 站点（<http://www.ActiveState.com>）上找到下载和人工安装模块的说明。ActiveState 维护了一个关于它销售的 Perl 产品的特定 FAQ，你可以在那里找到必要的说明。



不使用 PPM 来安装模块，比如使用 Windows 下你自己的 C 编译器来进行安装，这不是本书要讲解的内容。Perl 的原始产品中包含了在 Windows 下你自己安装 Perl 的说明，但这不是初学者能够做的工作。如果你能够进行这项操作，那么自己来安装模块就不会太难，因为安装过程是大致相同的。

A.2.2 在UNIX下使用CPAN来安装模块

在UNIX下安装模块是很有趣并且会遇到许多问题的，但是它也可能是非常容易的。你需要一个ANSI C编译器（用于安装Perl的编译器就很好），如果供应商要求的话，你还必须拥有编译器许可证。你不需要GNU压缩程序gzip / gunzip的拷贝，有些UNIX供应商将它作为一个标准实用程序提供给用户使用。如果你没有这个拷贝，可以从网址 <http://www.fsf.org> 下载一个拷贝。



有些UNIX供应商（比如HP公司）在它们的操作系统中配备了一个C编译器，但是它不是ANSI C编译器，这是C编译器的一个非常简化的版本，因此你必须花钱购买实际的C编译器，或者交费下载和安装GNU C编译器。

最后一个是：你在安装模块的计算机上必须拥有根（管理员）访问权限。通常情况下，Perl是作为整个系统范围的实用程序来安装的。将模块安装到系统目录中，你必须拥有足够的访问权限（即根权限）才能进行这种操作。

Perl产品配有一个称为CPAN的模块，用来帮助你安装其他的模块。若要开始安装操作，你必须使用CPAN模块的shell命令来启动Perl，如下所示：

```
$ perl -MCPAN -e shell
```

如果你是初次运行该命令，CPAN模块就会要求你确定从何处取得Perl的模块以及你想要如何安装这些模块。大多数情况下，默认答案就足以满足你的要求。然后它会问你临时目录的位置在什么地方（这是CPAN对你想使用的目录进行镜像的目录），并且问你是否通过代理程序来访问Internet。

当CPAN结束对你的提问后，你会看到下面这个提示：

```
cpan shell -- CPAN exploration and modules installation (v1.3901)
ReadLine support available (try `install Bundle::CPAN`)
```

```
cpan>
```

在这个提示后面，你可以使用命令 `i / pat /`，搜索关于模块包的信息，其中 `pat` 用于说明你要搜索的模式。例如，若要查找 `Date::Manip` 模块，请输入下面这个命令：

```
cpan> i /Manip/
```

CPAN模块必须与一个CPAN服务器取得联系，以使获取该索引的新拷贝。这种情况只有在需要时才会出现，并且这个进程只需很短时间就能完成。当查询结束时，CPAN就会答复下面这样的信息：

```
Distribution  SBECK/DateManip-5.35.tar.gz
Module       Date::Manip    (SBECK/DateManip-5.35.tar.gz)
```

若要安装该模块，请键入下面的命令：

```
cpan> install Date::Manip
```

这时，CPAN模块开始按步骤执行索取、编译、测试和安装模块的各个进程。它显示的信息相当零乱，不过它类似下面这个大大简化了的例子（#后面的注释通常并不出现，这里增加了注释，目的是使它更加清楚）：

```
Running make for SBECK/DateManip-5.35.tar.gz
Fetching with LWP:          # Fetching the module

ftp://ftp.perl.org/pub/perl/CPAN/authors/id/SBECK/DateManip-5.35.t
ar.gz
Writing Makefile for Date::Manip
mkdir blib                  # Building the module
mkdir blib/lib
Target "makemakerdflt" is up to date.
/usr/bin/make -- OK
Running make test          # Testing to ensure it works
    PERL_DL_NONLAZY=1 /usr/bin/perl -Iblib/arch -Iblib/lib
-I/usr/local/lib/
perl5/5.00502/aix -I/usr/local/lib/perl5/5.00502 -e 'use
Test::Harness qw(&runte
sts $verbose); $verbose=0; runtests @ARGV;' t/*.t
t/settime.....ok
t/unixdate.....ok
All tests successful.
Files=30, Tests=826, 178 wallclock secs (168.85 cusr + 5.23
csys = 174.08 CPU)
Target "test" is up to date.
/usr/bin/make test -- OK
Running make install      # Installing the module
Target "install" is up to date.
/usr/bin/make install -- OK
```

你得到的输出可能与上面的情况有很大的不同。现在该模块已经测试和安装好了。

A.2.3 在UNIX下用另一种方法安装模块

虽然你可以不使用 CPAN 模块在 UNIX 下安装各个模块，但是大多数情况下不需要用下面这种方法来安装模块。我们只是为了完整起见才介绍这种安装方法，但是只要可能，都应该使用 CPAN 模块来安装各个模块。

首先，必须从 CPAN 下载你要安装的模块。它是个压缩了的综合模块包。例如，如果要安装的模块是 Date::Calc，你必须得到它的新版本，它的名字类似 Date-Calc-X.Y.tar.gz。当你下载了该模块包后，进入该目录，对该模块包进行拆包操作，如下所示：

```
$ gunzip Date-Calc-4.2.tar.gz
$ tar xf Date-Calc-4.2.tar
```

拆包后便产生一个子目录，称为 Date-Calc-4.2。若要转入该子目录，请使用 cd，并键入下面的命令：

```
$ perl Makefile.PL
Checking if your kit is complete...
Looks good
Writing Makefile for Date::Calc
```

现在你就拥有一个 make 程序的描述文件，这对于安装进程来说是个必不可少的文件。接着，使用下面这样的 make 命令，安装该模块：

```
$ make
mkdir blib
mkdir blib/lib
:
Manifesting blib/man3/Date::Calc.3
Target "makemakerdflt" is up to date.
```

这个进程的运行需要花费一定的时间。

在下一个提示符后面，你必须测试该模块，以了解它的安装是否正确。请键入下面这个 make test 命令：

```
$ make test
PERL_DL_NONLAZY=1 /usr/bin/perl -Ilib/arch -Ilib/lib
-I/usr/local/lib/
perl5/5.00502/aix -I/usr/local/lib/perl5/5.00502 -e 'use
Test::Harness qw(&runte
sts $verbose); $verbose=0; runtests @ARGV;' t/*.t
t/f000.....ok
t/f001.....ok
:
t/f032.....ok
t/f033.....ok
All tests successful.
Files=34, Tests=1823, 14 wallclock secs ( 9.81 cusr + 1.10 csys
= 10.91 CPU)
Target "test" is up to date.
```

你始终都应该运行 make test 命令，以确保模块安装正确。它能省去你以后好几个小时的调试时间。当测试完成后，必须像下面这样安装该模块。这个操作步骤通常是以根用户身份来进行的，因为安装时必须写入系统目录：

```
$ su
Password: *****
# make install
Installing
/usr/local/lib/perl5/site_perl/5.005/aix/auto/Date/Calc/Calc.so
:
Appending installation info to
/usr/local/lib/perl5/5.00502/aix/perllocal.pod
Target "install" is up to date.
#
```

这样，你的安装操作就完成了。

A.2.4 在Macintosh系统上安装模块

在Macintosh系统上安装模块是比较困难的。你应该查看 MacPerl的FAQ，了解关于可以用来安装模块的方法的信息。MacPerl FAQ可以在网址<http://WWW.macperl.com>上找到。

A.3 当不允许你安装模块时该怎么办

如果你能够在系统上安装程序，你就能够安装模块。你能够这样做，取决于模块的复杂程度和你会遇到何种困难。有时系统管理员不允许你安装某个模块，因为他不想让其他人使用该模块。在某些情况下，只有你或者一组人才想使用某些特定模块，在整个系统范围内安装这些模块太复杂了。

无论哪种情况，在你自己的目录中安装 Perl 模块的专用拷贝并不难。

首先，必须使用前面给出的说明（只有一些小的例外）安装模块。你可以指定安装程序，将模块安装到特写的目录中。如果在 Windows 下使用 PPM，在你安装模块前，必须告诉 PPM，你想将模块安装到另一个目录。为此可以使用下面这样的 set 命令：

```
PPM> set root c:\myperl
PPM> set build c:\myperl
```

然后该模块被组装在目录 C:\myperl 中。

在UNIX下，当你使用CPAN模块时，可以使用下面的 `makepl_arg` 设置项来设定安装目录：

```
cpan> o conf makepl_arg PREFIX="/home/clintp/perl/lib"
```

或者，如果你使用 `make` 实用程序人工安装模块，你可以在第一个代码行上使用 `PREFIX` 参数，设定安装目录：

```
$ perl Makefile.PL PREFIX="/home/clintp/perl/lib"
```

无论使用哪种方法，你要安装的模块将被安装到 `/home/clintp/perl/lib` 目录中。如果需要的话，你可以再将该模块移到另一个目录中。

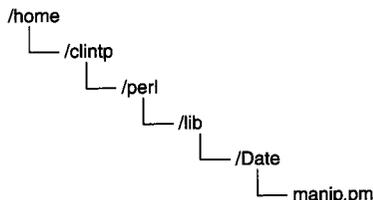


你应该注意，不要将模块在不同操作系统的计算机之间移动。经过编译的模块只能在一种类型的操作系统上运行，这与 Perl 本身的情况是一样的。另外，不要试图在不同版本的 Perl 之间移动模块，有时它不能运行。在这种情况下，你必须重新安装该模块。

使用安装在特殊位置中的模块

若要使用安装在非标准目录中的模块，必须使用命令 `use lib`。例如，如果你使用上一节中的说明将模块 `Date::Manip` 安装在目录 `/home/clintp/perl/lib` 中，就会得到一个图 A-2 所示的文件树。

图A-2 安装 `Date::Manip` 模块后形成的文件树



在你的程序开始处，只需要使用下面的代码：

```
use lib '/home/clintp/perl/lib'; # Look for module here
else
use Date::Manip;
```

这时 Perl 在搜索它自己的目录之前，首先搜索该目录，找出它要的模块。还可以使用这种方法将模块的新版本安装在系统上（以便达到测试目的），但不会改写老的版本，也不会带来不兼容的问题。