

## 1 NAME

perlutil – Perl 发行版中的实用程序

## 2 DESCRIPTION

Perl 发行版除了 Perl 解释器外还在你的系统上安装了一系列的实用程序。其中一些程序也是被 Perl 发行版本身用于安装过程。本文档将列出所有的这些程序，解释它们的用途，尽量提供相关模块文档的链接。

### 文档

#### perldoc

Perl 文档的主要界面是 `perldoc`，你可能已经知道了。`perldoc` 可以从当前目录下的任何文件、系统中安装的模块、或者标准文档（比如本文）中提取并格式化里面的文档。使用 `perldoc <name>` 可以得到本文中提到的任何程序的信息。

#### pod2man 和 pod2text

如果在终端中运行 `perldoc`，通常它是调用 `pod2man` 来转换 POD（Plain Old Documentation —— 参见 `perlpod`）为 `manpage`，然后运行 `man` 来显示。如果不能使用 `man` 命令，就使用 `pod2text`，并输出到你喜欢的 `pager`。

#### pod2html 和 pod2latex

除了这两个，还有两个转换程序：`pod2html` 能从 POD 中产生 HTML 页面；`pod2latex` 能够产生 LaTeX 文件。

#### pod2usage

如果你想知道怎样使用这里描述的程序，`pod2usage` 可以只抽提出“USAGE”部分；一些程序会在你用 `-help` 运行程序时自动对自身调用 `pod2usage`。

#### podselect

`pod2usage` 是 `podselect` 的一个特例。它是一个从文档的 POD 里抽提出某个名字的部分。例如，就像程序有“USAGE”部分，Perl 的模块通常有“SYNOPSIS”部分：`podselect -s "SYNOPSIS" ...` 将能从一个给定文件中抽提出这一部分。

#### podchecker

如果你要用 POD 写你自己的文档，`podchecker` 程序可以为你的标记检查错误。

#### splain

`splain` 是 `perldiag` 的接口，把你的错误信息粘贴给它，它就会向你解释。

#### roffitall

`roffitall` 程序不是安装在你的系统中，但是在你 Perl 源文件的 `pod/` 目录下。它能把发行版中的文档转换成 `*roff` 格式，然后产生 PostScript 或者文本文件。

## 转换器

为了帮助你将遗留程序转换成 Perl，我们三个转换过滤器：

### a2p

a2p 将 awk 脚本转换成 Perl 程序。例如，a2p -F：对简单的 awk 脚本 {print \$2 将产生一个类似于这样的 Perl 程序：

```
while (<>) {
    ($Fld1,$Fld2) = split(/[:\n]/, $_, 9999);
    print $Fld2;
}
```

### s2p

同样，s2p 将 sed 脚本转换成 Perl 程序。对 s/foo/bar 使用 s2p 将产生类似这样的代码：

```
while (<>) {
    chomp;
    s/foo/bar/g;
    print if $printit;
}
```

### find2perl

最后，find2perl 用 File::Find 模块将 find 命令转换成等价的 Perl 命令。例如，find2perl . -user root -perm 4000 -print 为 File::Find 产生这样的调用函数：

```
sub wanted {
    my ($dev,$ino,$mode,$nlink,$uid,$gid);
    (($dev,$ino,$mode,$nlink,$uid,$gid) = lstat($_)) &&
    $uid == $uid{'root'}) &&
    (($mode & 0777) == 04000);
    print("$name\n");
}
```

除了这些转换其它语言的过滤器，pl2pm 程序能帮你将 Perl 4 旧式的库转换成新式的 Perl5 模块。

## 系统管理员

### libnetcfg

运行 libnetcfg 命令来显示或者更改 libnet 的配置。

## 开发

这里有一系列的程序帮助你开发 Perl 程序，特别是用 C 扩展 Perl。

### perlbug

推荐使用 perlbug 向开发人员提交 perl 解释器本身或者任何标准模块中的 bug。在使用它提交一个 bug 前，请先通读一遍 perlbug 的文档。

## h2ph

在存在 XS 系统连接 C 库之前，程序员曾经使用读取 C 头文件的方法得到库中的常量。你可能还能看到 `require 'syscall.ph'` 或者相似的代码，这个 .ph 文件是对相应 .h 文件使用 h2ph 程序来创建的。参考 h2ph 的文档来获得如何一次性转化多个头文件的方法。

## c2ph 和 pstruct

c2ph 和 pstruct 都用于将 C 结构和联合声明转换成 Perl 代码，为在 Perl 中使用 C 提供另一种方法。它们事实上是相同的程序，但是调用的方法不同会有不同的结果。现在，h2xs 逐渐取代了它们。

## h2xs

h2xs 将 C 头文件转换成 XS 模块，然后尽可能的为 C 库和 Perl 模块提供 粘合代码。它对于创建纯 Perl 模块也是很有用的。

## dprofpp

Perl 提供了一个 profiler: Devel::DProf 模块。dprofpp 程序分析这个 profiler 的输出，然后告诉你哪个函数花费了最多的运行时间。参见 Devel::DProf。

## perlcc

perlcc 是实验中的 Perl 编译器的接口。

## SEE ALSO

perldoc, pod2man, perlpod, pod2html, pod2usage, podselect, podchecker, splain, perldiag, roffitall, a2p, s2p, find2perl, File::Find, pl2pm, perlbug, h2ph, c2ph, h2xs, dprofpp, Devel::DProf, perlcc

## 3 TRANSLATORS

YE Wenbin