



CPAN Frequently Asked Questions

For general information about Perl you should see the [Perl Home Page](http://www.perl.com) (www.perl.com). See the Perl FAQ (especially for any Perl programming questions, but also for other resources), available at <http://perlfaq.cpan.org/>.

© 1998-2001 Elaine Ashton <hfb@chaos.wustl.edu> and Jarkko Hietaniemi <jhi@iki.fi> All Rights Reserved.

I. - General Questions.

- [What is Perl?](#)
 - [What is Perl6?](#)
 - [What is the CPAN?](#)
 - [Where did CPAN get the logo and may I use it in my documentation, etc.?](#)
 - [What is PAUSE?](#)
 - [How does the CPAN work?](#)
 - [How does the CPAN multiplexer work?](#)
-

II. - The Quest for Perl source, modules and scripts.

- [Where can I find the current release of the Perl source code?](#)
- [Where can I find older/obsolete versions of Perl or Perl Modules?](#)
- [How do I interpret the Perl version numbers?](#)
- [How do I install Perl from the source code?](#)
- [Where can I find Perl modules?](#)
- [How do I install Perl modules?](#)
- [How do I remove installed Perl modules?](#)
- [How do I find out what modules are already installed on my system?](#)
- [What is a bundle and how do I make one for my own use or distribution on CPAN?](#)
- [How do I check to see if certain modules run on my operating system?](#)
- [How do I manually install a module in a private/non-standard directory?](#)
- [How do I use a module installed in a private/non-standard directory?](#)
- [Where can I find the most recently uploaded Perl modules?](#)
- [Where can I find Perl modules for Windows?](#)
- [Where can I find Perl binaries/packages or Perl module binaries?](#)
- [How are Perl and the CPAN modules licensed?](#)
- [Does the Perl source distribution include any modules?](#)
- [What modules are platform dependent?](#)
- [Where can I find Perl scripts?](#)

III. - RIF [Reading is Fun-duh-mental]

- [Where can I find the Perl FAQs?](#)
 - [Where can I find Perl documentation?](#)
 - [Where can I find Perl module documentation?](#)
 - [Where can I find Perl DBI/DBD/database documentation?](#)
 - [Where can I find/join/create Perl mailing lists?](#)
 - [Where can I find Perl journals/magazines?](#)
 - [Where can I find Perl courses/training/on-line tutorials?](#)
 - [Where can I find CPAN on a CD-ROM?](#)
 - [How do I find/join/organise a Perl User Group?](#)
 - [Where can I find a history of Perl releases or a general history of the Perl community?](#)
-

IV. - Danger Will Robinson! Danger! Danger!

- [I got an error downloading a module, what should I do?](#)
 - [I downloaded a module/script/file but it was corrupt, what should I do?](#)
 - [How do I use module Foo::Bar, can you help? \(a.k.a. Are you a helpdesk?\)](#)
 - [When downloading a module a strange VRML viewer started up and I got an error, what should I do?](#)
 - [Where can I find the GDBM_File/DB_File module?](#)
 - [I'm having trouble with search.cpan.org, whom do I need to contact?](#)
-

V. - Searching CPAN, CSPAN and the rest of the known universe.

- [How do I search CPAN?](#)
 - [How do I search for module/script documentation? \(a.k.a. "How do I use ..."?\)](#)
 - [How do I search for really ANYTHING?](#)
 - [How do I find Ralph Nader? \(a.k.a. We're C-P-A-N not C-S-P-A-N!\)](#)
-

VI. - Contributing modules, patches, and bug reports.

- [How do I contribute modules to CPAN?](#)
 - [Does CPAN allow contributions of shareware or code that requests a fee of any kind?](#)
 - [How do I contribute scripts to CPAN?](#)
 - [How do I contribute documentation?](#)
 - [How do I report/fix a bug in Perl and/or its documentation?](#)
 - [How do I report/fix a bug in a module/script?](#)
 - [How do I go about maintaining a module when the author is unresponsive?](#)
 - [Is there a site for module bug reports/tests?](#)
 - [Does CPAN provide download statistics for authors?](#)
-

VII. - How to mirror CPAN.

- [How do I mirror CPAN?](#)
 - [What do I need to mirror CPAN?](#)
 - [I have Windows 2000/NT/98/95, how can I mirror CPAN?](#)
 - [Which CPAN site should I mirror?](#)
 - [How do I register my mirror to make it a public resource?](#)
-

What is Perl?

From the Perl documentation:

Perl is a high-level programming language with an eclectic heritage written by Larry Wall and a cast of thousands. It derives from the ubiquitous C programming language and to a lesser extent from sed, awk, the Unix shell, and at least a dozen other tools and languages. Perl's process, file, and text manipulation facilities make it particularly well-suited for tasks involving quick prototyping, system utilities, software tools, system management tasks, database access, graphical programming, networking, and world wide web programming. These strengths make it especially popular with system administrators and CGI script authors, but mathematicians, geneticists, journalists, and even managers also use Perl. Maybe you should, too.

See also <http://www.sysadminmag.com/tpj/whatisperl.html>

Two good starting points for Perl information are <http://www.perl.com/> and <http://www.perl.org/>

What is Perl6?

At The Second O'Reilly Open Source Software Convention, Larry Wall announced Perl6 development would begin in earnest. Perl6 was an oft used term for Chip Salzenberg's project to rewrite Perl in C++ named Topaz. However, Topaz should not be confused with the nusus to rewrite Perl while keeping the lessons learned from other software, as well as Perl5, in mind.

If you have a desire to help in the crusade to make Perl a better place then peruse the Perl6 developers page at <http://www.perl.org/perl6> and get involved.

"We're really serious about reinventing everything that needs reinventing." --Larry Wall

What is CPAN?

CPAN is the Comprehensive Perl Archive Network, a large collection of Perl software and documentation. You can begin exploring from either <http://www.cpan.org/>, <http://www.perl.com/CPAN/> or any of the mirrors listed at <http://www.cpan.org/SITES.html> and <http://mirror.cpan.org/>.

Note that CPAN is also the name of a Perl module, CPAN.pm, which is used to download and install Perl software from the CPAN archive. This FAQ covers only a little about the CPAN module and you may find the documentation for it by using `perldoc CPAN` via the command line or on the web at <http://search.cpan.org/author/JHI/perl-5.8.0/lib/CPAN.pm>.

Where did CPAN get the logo and may I use it in my documentation, etc.?

The [CPAN logo](#) was designed by J.C. Thorpe who donated his time and creativity at our behest. The "A" for Archive is formed by 3 books and is, we think, a nice addition to the front page.

We allow the use of the logo for purposes relating directly to CPAN, e.g. you would like to make a link to CPAN on your web page or use the logo in documentation discussing CPAN. For any other purpose, please email us and tell us how you intend to use it.

What is PAUSE?

[PAUSE](#) is the Perl Authors Upload SErver, a registry for Perl module, script and documentation authors to upload their work to the CPAN. CPAN and PAUSE are often used interchangeably but they are distinct from each other. The [CPAN.pm](#) documentation explains it rather simply;

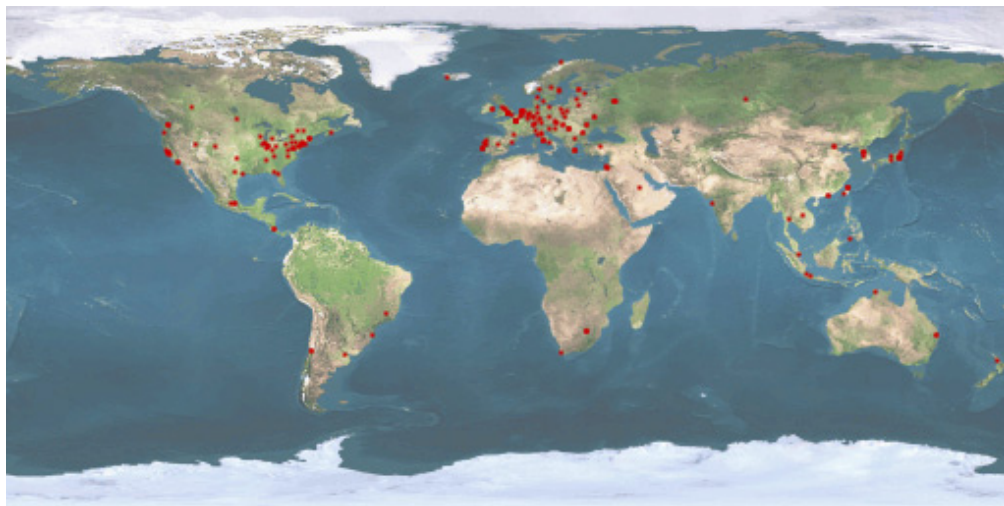
In this discussion CPAN and PAUSE have become equal -- but they are not. PAUSE is authors/, modules/ and scripts/. CPAN is PAUSE plus the cpa/, doc/, misc/, ports/ and src/.

See the question ['How do I contribute modules?'](#) below if you want to become a registered PAUSE user.

How does the CPAN work?

With dark magic, evil-looking sacrificial knives and scantily clad virgins under pale moonlight.

It actually works with the generosity and cooperation of hundreds of developers, over 175 participating mirrors, funet.fi donating the network bandwidth, storage space and computing power, volunteers who help keep everything together and users whose interest in Perl keep the archive alive and growing.



After an author uploads their module to [PAUSE](#), it will be mirrored to CPAN once an hour and from there, to the rest of the mirrors around the world. Various scripts run on CPAN daily to make sure that mirrors are up and running and that the mirror to PAUSE is functional, etc. There are people who advise authors on their choice of name and namespace for their modules and a few others who answer questions and investigate issues sent to cpan@perl.org.

How does the CPAN multiplexer work?

Not nearly as much dark magic and certainly no virgins, scantily clad or not...er...I mean...

The CPAN Multiplexer is the creation of Tom Christiansen and it lives at <http://www.perl.com/CPAN> which will present you with a menu of available mirrors and links to the source code if you wish to see how it really works. You may use a trailing slash on the above URL to have it try and direct you to the nearest mirror. The multiplexer attempts to map the tail part of the resolved DNS name of the client to the closest possible CPAN site, e.g. An *.ac.uk address will be directed to a mirror in the .uk domain, but an unresolvable numeric address or a *.com will be directed to the perl.com mirror.

Where can I find the current release of the Perl source code?

- <http://www.cpan.org/src/latest.tar.gz> - (UNIX-style tar.gz)
 - <http://www.cpan.org/src/latest.zip> - (PC-style ZIP, with CR LF line endings).
 - <http://www.cpan.org/src/index.html> - A more detailed list of source code offerings.
-

Where can I find older/obsolete versions of Perl or Perl Modules?

Unless you have A Very Good Reason™ you shouldn't be installing obsolete versions because they might contain bugs, possibly even security bugs.

Good Reasons may include having to support Perl 4 programs, trying to replicate a bug that requires an old Perl release, or pure joy of software archaeology. (Are you Perl 1 compliant?)

CPAN does not carry all ancient releases and patchlevels of Perl (because of the bugs we mentioned above and because they would take quite a lot of storage space). However, there are a few archives that carry an assortment of old Perl releases:

- <http://www.etla.org/retroperl/>
 - <http://mirrors.valueclick.com/perl/really-ancient-perls/>
 - <http://history.perl.org/src/>
 - <http://mirrors.valueclick.com/perl/backup.pause/> - for old versions of modules
-

How do I interpret the Perl version numbers?

Perl changed the version numbering system with v5.6.0 as was indicated in the release announcement:

Perl v5.6.0 is a major release that incorporates all maintenance and development changes since the last major release, 5.005. As you may have noticed, the version numbering has changed. Releases will henceforth be numbered as revision.version.subversion triples. Maintenance releases will have an even version component, while the version component for development releases will be odd. For example, the next maintenance update of Perl 5.6.0 will be v5.6.1, and the development series will begin life at v5.7.0.

You may also peruse the `perlhists` manpage for a complete list of versions and their release dates.

How do I install Perl using the source code?

To build Perl you need a C compilation environment. After downloading the source code and opening it up, you should first read the `INSTALL` document which will detail how to build Perl on most systems. There are a number of `README.[platform]` for platforms where special care is needed in building Perl. As always, reading the documentation is a Good Thing™.

Perl can be installed using the standard source code distribution on almost all platforms Perl runs on. This includes all the UNIXes (and good

lookalikes, meaning POSIX environments like OS/2, Plan 9, QNX, Amiga, MPE/iX, VMS, OS390, Stratus VOS), and Microsoft platforms. The most notable exceptions are (as of 1999-Mar-24);

- MacOS (1)
- AS/400 (2)
- Novell Netware (2)

For these platforms a binary release may be the easiest path.

1. The source code to compile MacPerl is available at <http://www.cpan.org/ports/mac/>.
2. The source code for AS/400 and Netware Perls have not been merged to the main Perl source code distribution. If you want to try compiling them yourself, get the sources from <http://www.cpan.org/ports/As400/> or <http://www.cpan.org/ports/netware/> and then continue at <http://www.cpan.org/src/README>

Where can I find Perl modules?

- <http://www.cpan.org/modules/00modlist.long.html>
- <http://search.cpan.org/>
- <http://kobesearch.cpan.org/>

Due to the ever increasing number of modules on CPAN, the CPAN search engine is possibly a better starting point in your quest for code, especially if you already know exactly what you are looking for.

How do I install Perl modules?

Installing a new module can be as simple as typing `perl -MCPAN -e 'install Chocolate::Belgian'`. The [CPAN.pm documentation](#) has more complete instructions on how to use this convenient tool. If you are uncomfortable with having something take that much control over your software installation, or it otherwise doesn't work for you, the [perlmodinstall](#) documentation covers module installation for UNIX, Windows and Macintosh in more familiar terms.

Finally, if you're using ActivePerl on Windows, the [PPM](#) (Perl Package Manager) has much of the same functionality as CPAN.pm.

How do I remove installed Perl modules?

By using the [ExtUtils::Installed](#) and [ExtUtils::Packlist](#) modules that come with Perl as in the example below. There is also a more elaborate example in the [ExtUtils::Packlist](#) man page.

```
#!/usr/local/bin/perl -w

use ExtUtils::Packlist;
use ExtUtils::Installed;

$ARGV[0] or die "Usage: $0 Module::Name\n";

my $mod = $ARGV[0];

my $inst = ExtUtils::Installed->new();

    foreach my $item (sort($inst->files($mod))) {
        print "removing $item\n";
    }
```

```

        unlink $item;
    }

my $packfile = $inst->packlist($mod)->packlist_file();
print "removing $packfile\n";
unlink $packfile;

```

How do I find out what modules are already installed on my system?

- **perldoc perllocal**

Each time a module is installed on your system, it appends information like the following to a file called `perllocal.pod` which can be found in `/usr/local/lib/perl5/version number/architecture/` or something akin to that. The path for your specific installation is in your `@INC` which you can divine with `perl -V`.

```

=head2 Wed May 12 13:42:53 1999: C<Module> L<Data::Dumper>

=over 4

=item *

C<installed into: /usr/local/lib/perl5/5.00503>

=item *

C<LINKTYPE: dynamic>

=item *

C<VERSION: 2.101>

=item *

C<EXE_FILES: >

=back

```

Each entry includes the Module name, date and time it was installed, where it was installed, linktype [static or dynamic], version and executables, if any, included with the module.

- **Use the ExtUtils::Installed module**

[ExtUtils::Installed](#) provides a standard way to find out what core and module files have been installed. It uses the information stored in `.packlist` files created during installation to provide this information. In addition it provides facilities to classify the installed files and to extract directory information from the `.packlist` files.

```

#!/usr/local/bin/perl

use ExtUtils::Installed;
my $instmod = ExtUtils::Installed->new();
foreach my $module ($instmod->modules()) {
my $version = $instmod->version($module) || "???";
    print "$module -- $version\n";
}

```

produces the following list of modules and their version

```
Apache::DBI -- 0.87
Apache::DBILogConfig -- 0.01
Apache::DBILogger -- 0.93
AppConfig -- 1.52
Archive::Tar -- 0.22
BerkeleyDB -- 0.06
CGI -- 2.74
CPAN -- 1.59
CPAN::WAIT -- 0.27
Catalog -- 1.00
Compress::Zlib -- 1.11
Config::IniFiles -- 2.14
Convert::BER -- 1.26
Coy -- ???
Crypt::Rot13 -- 0.04
Crypt::SSLeay -- 0.16
DBI -- 1.14
[.....]
```

- **ActivePerl has the PPM utility.**

```
C:\>ppm query
Archive-Tar    [0.072 ] module for manipulation of tar archives.
Compress-Zlib [1.03  ] Interface to zlib compression library
DBI            [1.13  ] Database independent interface for Perl
GD            [1.25  ] Interface to Gd Graphics Library
HTML-Parser   [2.23  ] SGML parser class
MIME-Base64   [2.11  ] Encoding and decoding of base64 strings
PPM           [1.1.4 ] Perl Package Manager: locate, install, upgrade software
```

- **The pmtools suite** written by Tom Christiansen to help navigate and manage Perl module installations. You can obtain them at <http://language.perl.com/misc/pmtools-1.00.tar.gz>. From the README:

This is pmtools -- a suite of small programs to help manage modules. The names are totally preliminary, and in fact, so is the code. We follow the "keep it small" notion of many tiny tools each doing one thing well, eschewing giant megatools with millions of options.

- pmpath - show the module's full path
- pmvers - get a module version number
- pmdesc - get a module description
- pmall - get all installed modules pmdesc descriptions
- pmdir - print the perl module path, newline separated
- plxload - show what files a given program loads at compile time
- pmload - show what files a given module loads at compile time
- pmexp - show a module's exports
- pminst - find what's installed
- pmeth - list a class's methods, recursively
- pmls - long list the module path
- pmcat - cat the module source through your pager
- pman - show the module's pod docs
- pmfunc - show a function source code from a module
- podgrep - grep in pods of a file
- pfcats - show pods from perlfunc
- podtoc - list table of contents of a podpage
- podpath - show full path of pod file
- pods - list all standard pods and module pods
- sitepods - list only pods in site_perl directories

- o basepods - list only normal "man-page" style pods
- o faqpods - list only faq pods
- o modpods - all module pods, including site_perl ones
- o stdpods - list standard pods, not site_perl ones

What is a bundle and how do I make one for my own use or distribution on CPAN?

The 3rd edition of *Programming Perl* defines a bundle as "A group of related modules on CPAN" which is reasonably straightforward but it doesn't tell you what it is or how to make one yourself. A bundle can be any collection of modules, related or not. Bundles are used by CPAN.pm to install a group of modules quickly and easily.

A bundle is essentially a module in the `Bundle::` namespace that has all the look and feel of a module but really isn't since it contains no code instead having a list of modules to be installed. Think of it as a list of ingredients for a CPAN.pm recipe. There are a number of existing module bundles that you can view as examples to help in building your own.

You can get a [list of bundle](#) distributions on search.cpan.org and CPAN.pm also lists all currently available bundles when you type 'b' at the prompt.

```
cpan> b
Bundle      Bundle::ABH      (A/AB/ABH/Bundle-ABH-1.04.tar.gz)
Bundle      Bundle::AO::Base (I/IX/IX/AO-0.32.tar.gz)
Bundle      Bundle::AO::Standard (I/IX/IX/AO-0.32.tar.gz)
Bundle      Bundle::Apache   (D/DO/DOUGM/mod_perl-1.25.tar.gz)
Bundle      Bundle::Apache::ASP (C/CH/CHAMAS/Apache-ASP-2.09.tar.gz)
Bundle      Bundle::Apache::Roaming (J/JW/JWIED/Apache-Roaming-0.1003.tar.gz)
Bundle      Bundle::BioPerl  (C/CR/CRAFFI/Bundle-BioPerl-1.00.tar.gz)
Bundle      Bundle::Bonsai   (Z/ZL/ZLIPTON/Bundle-Bonsai-0.02.tar.gz)
[....]
```

If you give 'b' a module argument it will list extended information about the bundle.

```
cpan> b Bundle::DBI
Bundle id = Bundle::DBI
CPAN_USERID  TIMB (Tim Bunce )
CPAN_VERSION 1.03
CPAN_FILE    T/TI/TIMB/DBI-1.15.tar.gz
MANPAGE      Bundle::DBI - A bundle to install DBI and required modules.
CONTAINS     Storable Net::Daemon RPC::PlServer Getopt::Long DBI
INST_FILE    /usr/local/lib/perl5/site_perl/5.005/sun4-solaris/Bundle/DBI.pm
INST_VERSION 1.03
```

There are 2 Ways to make a bundle:

1. Use the `autobundle` feature of CPAN.pm

`autobundle` will, if no arguments are given, inventory all modules installed on the system and make a systemwide bundle. If you only want a few modules in the bundle you can simply give it a list of modules and it will do the rest.

```
foo@bar /home/chaos/monkey> perl -MCPAN -eshell

cpan> autobundle CGI Crypt::Rot13 Date::Christmas Date::Manip

Going to read /home/chaos/monkey/.cpan/sources/authors/01mailrc.txt.gz
CPAN: Compress::Zlib loaded ok
Going to read /home/chaos/monkey/.cpan/sources/modules/02packages.details.txt.gz
```

```
Going to read /home/chaos/monkey/.cpan/sources/modules/03modlist.data.gz
```

Package namespace	installed	latest	in CPAN file
CGI	2.74	2.753	L/LD/LDS/CGI.pm-2.753.tar.gz
Crypt::Rot13	0.04	0.04	A/AY/AYRNIEU/Crypt-Rot13-0.04.tar.gz
Date::Christmas	1.01	1.02	H/HF/HFB/Date-Christmas-1.02.tar.gz
Date::Manip	5.39	5.39	S/SB/SBECK/DateManip-5.39.tar.gz

```
Wrote bundle file
/home/chaos/monkey/.cpan/Bundle/Snapshot_2001_05_09_01.pm
```

The "Snapshot_2001_05_09_01.pm" file may then be used with the CPAN.pm

```
perl -MCPAN -e 'install Bundle::Snapshot_2001_05_09_01'
```

If it has trouble finding the snapshot check your CPAN.pm configuration.

```
cpan> o conf cpan_home
cpan_home /home/chaos/monkey/.cpan
```

This particular feature comes in very handy if you are a Systems Administrator or a user who have a number of modules you like to keep up to date on without much hassle or if you need to distribute modules to a large number of machines and want to keep it consistent. There are a lot of excellent uses for this convenient little tool.

2. Make Bundle distribution for CPAN

- **Create framework for Bundle like so:**

```
foo@bar /home/chaos/monkey> h2xs -AXcfn Bundle::MyBundle
Writing Bundle/MyBundle/MyBundle.pm
Writing Bundle/MyBundle/Makefile.PL
Writing Bundle/MyBundle/test.pl
Writing Bundle/MyBundle/Changes
Writing Bundle/MyBundle/MANIFEST
```

- **Edit the MyBundle.pm**

- Remove all but the package declaration, the version and the 1; in the body of the module
- Edit the name and add a brief ' - descriptive info' in the name
- Add `C<perl -MCPAN -e 'install Bundle::MyBundle'>` to the synopsis.
- Add a `=head1 CONTENTS` header and under it add modules in the bundle in the format:


```
ModuleName [ optional version ] [ - comment or description ]
```

 The '-' in front of the comment is required.
- Add a brief description.
- Add author information and license.

```
package Bundle::MyBundle;

$VERSION = '0.01';

1;
__END__

=head1 NAME

Bundle::MyBundle - Bundle of my very favourite modules

=head1 SYNOPSIS
```

```

C<perl -MCPAN -e 'install Bundle::MyBundle'>

=head1 CONTENTS

CGI 2.74          - CGI is groovy baby
Crypt::Rot13     - This would make for some interesting web pages
Date::Christmas - defies description
Date::Manip

=head1 DESCRIPTION

These modules rock, especially when installed together!

=head1 AUTHOR

Ima Bundle ima@bundle.com

=head1 LINCENSE

Steal this code!

=head1 SEE ALSO

perl(1).

=cut

```

- o **Remove files you don't need** such as 'Changes' and 'test.pl' from the directory and add a README file with a short description of the Bundle and a list of modules it will install. Update the MANIFEST file to accurately list the contents of the Bundle directory.
- o **Make the distribution** by first issuing 'perl Makefile.pl' then 'make dist' which will do the following leaving you with Bundle-MyBundle-0.01.tar.gz which is ready to be uploaded to CPAN.

```

foo@bar monkey/Bundle/MyBundle> perl Makefile.PL

Checking if your kit is complete...
Looks good
Writing Makefile for Bundle::MyBundle

foo@bar monkey/Bundle/MyBundle> make dist
rm -rf Bundle-MyBundle-0.01
/usr/bin/perl -I/usr/local/lib/perl5/5.00503/sun4-solaris -
I/usr/local/lib/perl5/5.00503 \
-MExtUtils::Manifest=manicopy,maniread -e "manicopy(maniread(),'Bundle-MyBu
0.01', 'best');"
mkdir Bundle-MyBundle-0.01
tar cvf Bundle-MyBundle-0.01.tar Bundle-MyBundle-0.01
a Bundle-MyBundle-0.01/ OK
a Bundle-MyBundle-0.01/Makefile.PL 1K
a Bundle-MyBundle-0.01/MyBundle.pm 1K
a Bundle-MyBundle-0.01/README 1K
a Bundle-MyBundle-0.01/MANIFEST 1K
rm -rf Bundle-MyBundle-0.01
gzip --best Bundle-MyBundle-0.01.tar

```

That's all there is to it. Developers may find the bundle feature convenient if there is an optimal installation of modules that work best with their distribution.

How do I check to see if certain modules run on my operating system?

Browse through <http://testers.cpan.org/> which has a listing of modules and whether or not the module succeeded or failed on a number of platforms.

The CPAN testers is a group initially organised by Graham Barr and Chris Nandor to test modules on as many platforms as possible to improve portability and generate useful feedback for module authors. There is always need for more people with esoteric platforms and who are willing to install modules and send in reports. See the testers web page for more details on how you can help.

How do I manually install a module in a private/non-standard directory?

You need to set `PREFIX` and `LIB` when you run the `Makefile.PL`. `LIB` is where the module files will go and `PREFIX` is the stub directory for everything else. For example:

```
foo@barbell$ perl Makefile.PL LIB=/home/foobar/mylib PREFIX=/home/foobar/mylib
```

Read more about this in the [ExtUtils::MakeMaker](#) documentation.

How do I use a module installed in a private/non-standard directory?

There are several ways to use modules installed in private directories:

- `foo@barbell$ setenv PERL5LIB /path/to/module` sets the environment variable `PERL5LIB`.
- use `lib qw(/path/to/module)`; used at the top of your script tells perl where to find your module.
- `foo@barbell$ perl -I/path/to/module`

All of these will append `/path/to/module` to `@INC`. You should keep in mind that having private/non-standard libraries may cause problems if you wish to have portable code.

Where can I find the most recently uploaded Perl modules?

There are so many new and updated modules that it is hard to keep up with the deluge, but there are ways to stay abreast of the tide.

- <http://search.cpan.org/recent>
- http://theoryx5.uwinnipeg.ca/mod_perl/cpan-search?request=recent
- <http://use.perl.org/>
- <http://search.cpan.org/recent.rdf> - an RSS format of /recent

Any of these should be good for your daily feed of new modules.

Where can I find Perl modules for Windows?

www.activestate.com has a FAQ for their Package Manager or <http://www.northbound-train.com/perlwin32.html> which has a nice listing of Win32 resources including modules.

Where can I find Perl binaries/packages or Perl module binaries?

<http://www.cpan.org/ports/index.html> is a current list of Perl binaries that we are aware of at this time. If you have a package for a platform, send us a URL. We do not endorse nor guarantee these packages nor do we store them locally on CPAN due to the potential size of the archive if we did.

Perl module binaries for use with ActivePerl's PPM can be found at <http://www.activestate.com/PPMPackages/>.

How are Perl and the CPAN modules licensed?

Most, though not all, modules on CPAN are licensed under the GNU Public License (GPL) or the Artistic license and should be stated in the documentation that accompanies the module itself. If the license is not specifically stated in the module, you can always write the author to clarify the issue for you.

The Open Source Initiative has a nice page of 'approved' licenses and license mailing list information at <http://www.opensource.org/licenses/index.html>.

CPAN and PAUSE are not responsible for any licenses or lack thereof contained in the contents of the archive. We do recommend that authors license their modules to avoid legal ambiguity and so that people may use the code in good conscience. If you require help with a license, we urge you to consult legal counsel who can give you sound advice.

The text of the Artistic license and the GNU Public License are included in the root directory of the source distribution. From the 'README' file that comes with Perl:

```
Perl Kit, Version 5.0
```

```
Copyright 1989-1999, Larry Wall  
All rights reserved.
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of either:
```

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License" which comes with this Kit.

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See either  
the GNU General Public License or the Artistic License for more details.
```

```
You should have received a copy of the Artistic License with this  
Kit, in the file named "Artistic". If not, I'll be glad to provide one.
```

```
You should also have received a copy of the GNU General Public License  
along with this program in the file named "Copying". If not, write to the  
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  
02111-1307, USA or visit their web page on the internet at  
http://www.gnu.org/copyleft/gpl.html.
```

```
For those of you that choose to use the GNU General Public License,  
my interpretation of the GNU General Public License is that no Perl  
script falls under the terms of the GPL unless you explicitly put  
said script under the terms of the GPL yourself. Furthermore, any
```

object code linked with perl does not automatically fall under the terms of the GPL, provided such object code only adds definitions of subroutines and variables, and does not otherwise impair the resulting interpreter from executing any standard Perl script. I consider linking in C subroutines in this manner to be the moral equivalent of defining subroutines in the Perl language itself. You may sell such an object file as proprietary provided that you provide or offer to provide the Perl source, as specified by the GNU General Public License. (This is merely an alternate way of specifying input to the program.) You may also sell a binary produced by the dumping of a running Perl script that belongs to you, provided that you provide or offer to provide the Perl source as specified by the GPL. (The fact that a Perl interpreter and your code are in the same binary file is, in this case, a form of mere aggregation.) This is my interpretation of the GPL. If you still have concerns or difficulties understanding my intent, feel free to contact me. Of course, the Artistic License spells all this out for your protection, so you may prefer to use that.

Does the Perl source include any modules?

Yes, Perl comes with a number of useful modules and are listed in the perlmodlib pod:

- <http://search.cpan.org/author/JHI/perl-5.8.0/pod/perlmodlib.pod>
 - perldoc perlmodlib
-

What modules are platform dependent?

Several implementations of Perl on specific platforms come bundled with a collection of platform specific modules.

Additional information is available for:

IBM OS/2

The following modules come with standard Perl.

[OS2::DLL](#)

Access to DLLs with REXX calling convention and REXX runtime.

[OS2::ExtAttr](#)

Access to extended attributes.

[OS2::PrfDB](#)

Access to the OS/2 setting database.

[OS2::Process](#)

Constants for `\tool{system}(2)` call on OS/2.

[OS2::REXX](#)

Access to DLLs with REXX calling convention and REXX runtime.

DEC (Open)VMS

The following modules come with standard Perl.

[VMS::DCLsym](#)

Perl extension to manipulate DCL symbols.

[VMS::Filespec](#)

Converts between VMS and Unix file specification syntax.

[VMS::Stdio](#)

Standard I/O functions via VMS extensions.

[VMS::XSSymSet](#)

Keeps sets of symbol names palatable to the VMS linker.

Microsoft Windows (32 bit)

The following modules come with [ActiveState Perl](#).

[Win32::ChangeNotify](#)

Monitors events related to files and directories.

[Win32::Console](#)

Uses Win32 Console and Character Mode Functions.

[Win32::Event](#)

Uses Win32 event objects.

[Win32::EventLog](#)

Processes Win32 Event Logs.

[Win32::File](#)

Manages file attributes.

[Win32::FileSecurity](#)

Manages FileSecurity Discretionary Access Control Lists.

[Win32::IPC](#)

Loads base class for Win32 synchronization objects.

[Win32::Internet](#)

Accesses WININET .DLL functions.

[Win32::Mutex](#)

Uses Win32 mutex objects.

[Win32::NetAdmin](#)

Manages network groups and users.

[Win32::NetResource](#)

Manages network resources.

[Win32::ODBC](#)

Uses ODBC Extension for Win32.

[Win32::OLE](#)

Uses OLE Automation extensions.

[Win32::OLE::Const](#)

Extracts constant definitions from TypeLib.

[Win32::OLE::Enum](#)

Uses OLE Automation Collection Objects.

[Win32::OLE::NLS](#)

Uses OLE National Language Support.

[Win32::OLE::Variant](#)

Creates and modifies OLE VARIANT variables.

[Win32::PerfLib](#)

Accesses the Windows NT Performance Counter.

[Win32::Process](#)

Creates and manipulates processes.

[Win32::Semaphore](#)

Uses Win32 semaphore objects.

[Win32::Service](#)

Manages system services.

[Win32::Sound](#)

Plays with Windows sounds.

[Win32::TieRegistry](#)

Mungs the registry.

[Win32API::File](#)

Accesses low-level Win32 system API calls for files and directories.

[Win32API::Net](#)

Manages Windows NT LanManager accounts.

[Win32API::Registry](#)

Accesses low-level Win32 system API calls from WINREG.H.

Where can I find Perl scripts?

- <http://www.cpan.org/scripts/index.html>
 - <http://www.perlarchive.com/>
 - <http://freshmeat.net/>
 - and many, many other places on the net have Perl programs....
-

Where can I find the Perl FAQs?

The Perl FAQ is included with the Perl source code distribution.

- `perldoc perlfaq`
 - <http://perlfaq.cpan.org/>
-

Where can I find Perl documentation?

- <http://www.cpan.org/doc/index.html>
 - <http://perldoc.cpan.org/>
 - `perldoc perldoc` to use the documentation included with your Perl distribution.
 - http://search.cpan.org/author/JHI/perl-5.8.0/pod/perlfaq2.pod#Perl_Books for those who have read all of the free documentation and want something they can read in the loo by candlelight.
-

Where can I find Perl module documentation?

- <http://search.cpan.org/>,
 - <http://wait.cpan.org/>
 - `perldoc Foo::Bar` if the module is installed locally.
-

Where do I find Perl DBI/DBD/database documentation?

- <http://dbi.perl.org/> - Alligator Descartes Definitive DBI page.
- http://www.savebaseball.com/mysql/DBD_3.21.X.php3 - A mySQL Perl DBI/DBD Manual.
- <http://search.cpan.org/search?module=DBI> - The DBI man page.
- `perldoc DBI`

Where can I find Perl mailing lists?

There are quite a few mailing lists with a broad range of topics.

- <http://lists.perl.org/>
- <http://www.perl.com/pub/language/info/mailing-lists.html>,
- http://www.perl.org/support/mailing_lists.html
- <http://www.mail-archive.com/> - browseable web archives of many of the Perl lists.

Many of the lists are open for general subscription. If you don't see a list that interests you and would like to start your own you may ask lists@perl.org to set one up for you if you cannot host it yourself.

Where can I find Perl journals/magazines?

- <http://www.tpj.com/> - **The** journal in print for Perl.
 - <http://www.perlmonth.com/> - a terrific on-line only journal.
 - <http://www.unixreview.com/>
 - <http://www.stonehenge.com/merlyn/WebTechniques/> - Randal Schwartz' Web Techniques Columns.
 - <http://www.usenix.org/publications/login/login.html> - ;login, the publication of USENIX.
 - <http://www.samag.com/> - System Admin Magazine has occasional Perl articles.
-

Where can I find Perl courses/training/on-line tutorials?

Training

- <http://www.perltraining.org/>
- <http://www.perl.org/phbs/training.html>

On-line Tutorials

- <http://www.ebb.org/PickingUpPerl/>
 - <http://www.speech.cs.cmu.edu/~sburke/pub/perl.html>
 - <http://www.comp.leeds.ac.uk/Perl/start.html>
 - <http://www.cpan.org/doc/FMTEYEWTK/index.html>
 - <http://www.cclabs.missouri.edu/things/instruction/perl/perlcourse.html>
-

Where can I find CPAN on a CD-ROM?

CPAN (cpan.org) does not produce CD-ROMs so we cannot guarantee the freshness or quality of CD-ROMs produced by others or contained in the products below.

Red Hat offers a CPAN CD-ROM in its Deluxe Edition of Red Hat Linux. See <http://www.redhat.com/products/> for more details.

O'Reilly and Associates produce The Perl Resource Kit, the UNIX and Win32 Editions. The UNIX version is from November 1997 and no longer in print. The Win32 version is from August 1998. See <http://www.oreilly.com/catalog/> for more information and possible updates to the PRK.

Last but not least, the above list is certainly not comprehensive nor is it to be considered as an endorsement. If you know of other CPAN-on-a-CD products, please let us know and we will include them here.

How do I find/join/organise a Perl User Group?

The Perl User Groups are known as "Perl Mongers" and have active groups all over the world. You can find an established group at <http://www.pm.org/groups/index.html> or start a new group if one isn't near you via <http://www.pm.org/start/index.html>

Where can I find a history of Perl releases or a general history of the Perl community?

A history of Perl releases can be found in your Perl distribution via `perldoc perlhist` or via the web at <http://search.cpan.org/author/JHI/perl-5.8.0/pod/perlhist.pod>.

A more general history of the Perl community, CFAST, can be found at <http://history.perl.org/>.

I got an error downloading a module, what should I do?

If you feel that you have experienced unreasonable difficulty reaching a particular mirror and wish to inform us of this, please check the points below before contacting us. We do poll mirrors daily to be sure they are on-line and available, but there are times when servers or their networks go down for brief periods of time that we don't always see and have no control over.

- **What program were you using to download?**

A Web browser?
An FTP client?

- **Which server were you using?**

Note that many Perl software servers redirect your WWW requests to a site (hopefully) nearer to you. For example, the perl.com multiplexer does that so you often aren't downloading from perl.com itself. If you are using a web browser, take a close look at the URL/Location. Also note that we cannot debug your network connectivity and if you have problems connecting to anywhere other than the CPAN master site, ftp.funet.fi, we probably cannot be of service.

You can try changing the server to <http://www.perl.com/CPAN>, (note: NO final slash) and pick a new server. Note that the selection is 'sticky' and your browser will use that server from then on. Using <http://www.perl.com/CPAN/> (note the final slash) will use the multiplexer to pick a new mirror for you.

- **What was the exact error message?**

We are not clairvoyant so please include the exact error message, cut and paste if you must.

- **Did you retry later?**

The server might be temporarily busy or be offline and refuse connections for a while. Retry later or try another server.

I downloaded a module/script/file but it was corrupt, what should I do?

Many CPAN filenames end in `.tar.gz`. Unfortunately some programs mutilate such names (e.g., rename them with `_tar.tar`) so that unpacking programs don't recognise them and refuse to unpack them. Try saving the file using the `.tgz` suffix or try changing your web client. Also, you could try a plain FTP client as almost all the CPAN sites are ftp-reachable. You can find the full list of mirrors <http://www.cpan.org/SITES.html> or at <http://mirror.cpan.org/>.

If you use FTP remember to download in binary format, not text format.

Please read <http://www.cpan.org/ENDINGS> if you aren't sure what the files should be unpacked with and want to know if you are using the right program.

If you still think you have a corrupt file, try downloading the file from another site. If you still have no satisfaction, then please let us know the exact file name and URL/FTP site and path.

How do I use module `Foo::Bar`, can you help? (a.k.a. Are you a helpdesk?)

We at CPAN are not a helpdesk. We may point you towards a plethora of documentation to help you in your quest for knowledge but we cannot debug your code or read for you. We exist specifically to answer questions and solve problems relating directly to the functioning of the CPAN itself.

In addition to the on-line documentation you might try the newsgroup `comp.lang.perl.modules` for help with a particular module. Also, looking at other code using the same module might prove enlightening.

When downloading a module a strange VRML viewer started up and I got an error, what should I do?

You saw an error window "VRML Console" saying "Compilation error: Unrecognized header string". The reason the VRML viewer starts is that there is a type of compressed VRML file with a `.gz` extension, using (theoretically) the standard gzip compression. The VRML folk used `.gz` as a means of cutting down the download time on the old `wr1` files. Most VRML browsers come with a mini gunzip program. See also the VRML [FAQ](#) which also covers this problem.

You can fix this with the following steps;

1. Open "My Computer" or Windows NT Explorer.
2. Select the "View" menu.
3. Select "Options".
4. Click the "File Types" tab.
5. Scroll through the "Registered file types" until you reach "WorldView VRMLViewer Object" and select it.
6. Click "Remove".
7. Answer "Yes". (I've never seen these VRML files WorldView claims to be caring about: if you have and you do care, please tell me how to make WorldView stop caring about the `.gz` and `application/x-gzip`)
8. Quit and restart your browser for the change to take effect.
9. The next time you try open a file ending in `.gz` Windows will ask you which application to use to open that file. Scroll down to WinZip (`winzip32`), and remember to check the box asking whether to always use this application to open this type of file unless, of course, you plan on regularly using the VRML viewer.

Where can I find the `GDBM_File/DB_File` module?

The `GDBM_File` module comes standard with Perl 5.

The problem you are most likely to be having is that your system (or the system your binary distribution was built on in the case you are not using the source code distribution but instead relying on a prebuilt binary installation kit) does not have the **external library** called libgdbm, or GNU DBM. The GDBM_File module needs that to be built, installed, and used. The library has nothing to do with Perl as such. You can try hunting for it using the standard software repositories for your platform or <http://www.gnu.org/>

If you are looking for DB_File, every time it says GDBM think of DB, GDBM_File needs an external library called libdb, or Berkeley DB. You might try <http://www.sleepycat.com> who distributes the Berkeley DB source code.

I'm having trouble with search.cpan.org, whom do I need to contact?

If you are experiencing difficulty using search.cpan.org due to network or server errors, you need to contact webmaster@search.cpan.org.

How do I search for anything on CPAN?

By using a CPAN search engine.

- <http://www.perldoc.com/>
 - <http://search.cpan.org/> - Graham Barr's search engine which can search for modules, distributions or authors in all of CPAN.
 - <http://kobesearch.cpan.org/> - Randy Kobes' search engine that can also search all of CPAN for modules, documentation, etc.
 - <http://wait.cpan.org/> - full text search of pods on PAUSE.
-

How do I search for module/script documentation? (also known as "How do I use ..."?)

In general modules and scripts come with their own documentation which should have been installed along with your module/script. (Thanks to Perl's pod-style documentation, "it is very hard to misplace your documentation".)

- `perldoc Module::Name`
 - `perldoc scriptname`
 - <http://www.perldoc.com/>
 - <http://search.cpan.org/>
 - <http://wait.cpan.org/>
 - <http://theoryx5.uwinnipeg.ca/CPAN/>
-

How do I search for ANYTHING, really?

- <http://www.google.com/>
- <http://ragingsearch.altavista.com/>
- <http://www.altavista.com/>
- <http://www.hotbot.com/>
- <http://www.lycos.com/>
- <http://www.yahoo.com/>

(too many to list)

How do I find Ralph Nader? (a.k.a. We're C-P-A-N not C-S-P-A-N!)

You dialed the wrong number. The place you are looking for is <http://www.cspan.org/>.

How do I contribute modules to CPAN?

If you would like to learn more about PAUSE and how to go about contributing your module to CPAN please read the PAUSE FAQ at <http://www.cpan.org/modules/04pause.html> which will tell you how to go about getting a PAUSE ID and the steps needed to upload your code. Also, [perldoc perlmodlib](#) and [perldoc perlmod](#) are a good introduction to Perl modules.

Does CPAN allow contributions of shareware or code that requests a fee of any kind?

No. Everything on CPAN is free of charge. The reason for this is that CPAN is the product of hundreds of people donating their time and resources for the common good of the Perl community. There are places on the net where one can offer shareware without treading on the generosity of others and this is not that place.

How do I contribute scripts to CPAN?

CPAN has a scripts repository at <http://www.cpan.org/scripts/> and <http://www.cpan.org/scripts/submitting.html> will instruct you on how to go about contributing your scripts.

How do I contribute documentation to CPAN?

If the documentation is for a particular module that isn't a core distribution module, then please send it to the module author. If the module is a core module the most appropriate place to send doc patches and enhancements is the Perl5Porters mailing list.

How do I report/fix a bug in Perl and/or its documentation?

Always remember to make your bug reports as detailed as possible. "Perl doesn't work." is not a bug report.

Please note that problems concerning modules that are installed separately from the Perl distribution (such as Tk) are reported differently.

Here is a checklist from [perlbug](#), a bug reporting tool included in your Perl distribution. It is a bit on the long side, but please read it carefully as the better your bug report is, the more likely the issue will be addressed.

- **What version of Perl you are running?**

Try `perl -v` at the command line to find out.

- **Are you running the latest released version of perl?**

Look at <http://www.perl.com/> to find out. If it is not the latest released version, get that one and see whether your bug has been fixed. Note that bug reports about old versions of Perl, especially those prior to the 5.0 release, are less likely to be incorporated into the source as Perl1 through Perl4 are largely unmaintained.

- **Are you sure what you have is a bug?**

A significant number of the bug reports we get turn out to be documented features in Perl. Make sure the behavior you are witnessing doesn't fall under that category, by glancing through the [documentation](#) that comes with Perl (we'll admit this is no small task, given the sheer volume of it all, but at least have a look at the sections that seem relevant).

Be aware of the familiar traps that perl programmers of various hues fall into. See the [perltrap](#) documentation.

Check in [perldiag](#) to see what any Perl error message(s) mean. If message isn't in perldiag, it probably isn't generated by Perl. Consult your operating system documentation instead.

If you are on a non-UNIX platform check also [perlport](#) documentation, some features may not be implemented or work differently.

Try to study the problem under the Perl debugger, if necessary. See the [perldebug](#) documentation.

- **Do you have a proper test case?**

The easier it is to reproduce your bug, the more likely it will be fixed, because if no one can duplicate the problem, no one can fix it. A good test case has most of these attributes: fewest possible number of lines; few dependencies on external commands, modules, or libraries; runs on most platforms unimpeded; and is self-documenting.

A good test case is almost always a good candidate to be on the perl test suite. If you have the time, consider making your test case so that it will readily fit into the standard test suite.

Remember also to include all the exact error messages, if any. "Perl complained something" is not an exact error message.

If you get a core dump (or equivalent), you may use a debugger (dbx, gdb, etc) to produce a stack trace to include in the bug report. NOTE: unless your Perl has been compiled with debug info (often -g), the stack trace is likely to be somewhat hard to use because it will most probably contain only the function names and not their arguments. If possible, recompile your Perl with debug info and reproduce the dump and the stack trace.

- **Can you describe the bug in plain English?**

The easier it is to understand a reproducible bug, the more likely it will be fixed. Anything you can provide by way of insight into the problem helps a great deal. In other words, try to analyze the problem (to the extent you can) and report your discoveries.

- **Can you fix the bug yourself?**

A bug report which includes a patch to fix it will almost definitely be fixed. Use the diff program to generate your patches (diff is being maintained by the GNU folks as part of the diffutils package, so you should be able to get it from any of the GNU software repositories). If you do submit a patch, the cool-dude counter at perlbug@perl.com will register you as a savior of the world. Your patch may be returned with requests for changes, or requests for more detailed explanations about your fix.

Here are some clues for creating quality patches:

Use the -c or -u switches to the diff program (to create a so-called context or unified diff). Make sure the patch is not reversed (the first argument to diff is typically the original file, the second argument your changed file). Make sure you test your patch by applying it with the patch program before you send it on its way. Try to follow the same style as the code you are trying to patch. Make sure your patch really does work (make test, if the thing you're patching supports it).

- **Can you use perlbug to submit the report?**

perlbug will, amongst other things, ensure your report includes crucial information about your version of perl. If perlbug is unable to mail your report after you have typed it in, you may have to compose the message yourself, add the output produced by perlbug -d and email it to perlbug@perl.com. If, for some reason, you cannot run perlbug at all on your system, be sure to include the entire output produced by running perl -V (note the uppercase V).

Whether you use perlbug or send the email manually, please make your Subject line informative. "a bug" is not informative. Neither is "perl crashes" nor "HELP!!!" These don't help. A compact description of what's wrong is fine.

Having done your bit, please be prepared to wait, to be told the bug is in your code, or even to get no reply at all. The Perl maintainers are busy folks, so if your problem is a small one or if it is difficult to understand or already known, they may not respond with a personal reply. If it is important to you that your bug be fixed, do monitor the Changes file in any development releases since the time you submitted the bug, and encourage the maintainers with kind words (but never any flames!). Feel free to resend your bug report if the next released version of perl comes out and your bug is still present.

How do I report/fix a bug in a module/script?

Use <http://rt.cpan.org/> to open a bug ticket.

Please contact the author of the module/script. The documentation of the module/script should contain a contact address or you can try `CPANID@perl.org` where CPANID is the authors CPANID.

Most of the checklist in [reporting bugs in Perl](#) above applies for modules as well. Make your bug report as good as possible if you really want the bug fixed. If the module is included with the Perl distribution you should also follow the Perl bug reporting tips.

How do I go about maintaining a module when the author is unresponsive?

Sometimes a module goes unmaintained for a while due to the author pursuing other interests, being busy, etc. and another person needs changes applied to that module and may become frustrated when their email goes unanswered. CPAN does not mediate or dictate a policy in this situation and rely on the respective authors to work out the details. If you treat other authors as you would like to be treated in the same situation the manner in which you go about dealing with such problems should be obvious.

- Be courteous.
- Be considerate.
- Make an earnest attempt to contact the author.
- Give it time. If you need changes made immediately, consider applying your patches to the current module, changing the version and requiring that version for your application. Eventually the author will turn up and apply your patches, offer you maintenance of the module or, if the author doesn't respond in a year, you may get maintenance by having interest.
- If you need changes in order for another module or application to work, consider making the needed changes and bundling the new version with your own distribution and noting the change well in the documentation. Do not upload the new version under the same namespace to CPAN until the matter has been resolved with the author or CPAN.

Simply keep in mind that you are dealing with a person who invested time and care into something. A little respect and courtesy go a long way.

Is there a site for module bug reports/tests?

Yes, through the diligence of Paul Schinder and a few others, we have [CPAN Testers](#) which is a collection of test results for modules on a number of different platforms. This information is also available when viewing module information on [CPAN search](#).

There is also <http://rt.cpan.org/> where you might search for a module bug already reported and/or report a bug.

Does CPAN provide download statistics for authors?

No we don't. <http://xxx.lanl.gov/help/faq/statfaq> sums up our thoughts on the matter quite well.

How do I mirror CPAN?

Either an FTP or rsync client will do. Scantly clad virgins and pale moonlight are optional and are not included in the sales price.

What do I need to mirror CPAN?

- "Good" Internet connectivity, e.g. better than a 14.4 modem but not so much as an OC3.
- Around 1GB of storage space. <http://www.cpan.org/indices/du-k.gz> tells you the current size of the CPAN.
- An FTP or rsync client.

- For FTP there is a Perl script named mirror (which assumes a command line FTP client):

<http://sunsite.org.uk/packages/mirror/>

The FTP address for the CPAN master site is:

<ftp://ftp.funet.fi/pub/languages/perl/CPAN/>

- and for rsync:

<http://rsync.samba.org/>

A few mirrors who have rsync addresses available for CPAN are:

```
rsync.nic.funet.fi::CPAN
cpan.mirror.smartworker.org::CPAN
theoryx5.uwinnipeg.ca::CPAN
ftp.shellhung.org::CPAN
ftp.gwdg.de::FTP/languages/perl/CPAN/
ftp.leo.org::CPAN
CPAN.piksi.itb.ac.id::CPAN
ftp.iglu.org.il::CPAN
gusp.dyndns.org::cpan
ftp.kddlabs.co.jp::cpan
cpan.teleglobe.net::CPAN
ftp.sedl.org::cpan
archive.progeny.com::CPAN
mirrors.kernel.org::mirrors/CPAN
mirrors.phenominet.com::CPAN
mirror.csit.fsu.edu::CPAN
csociety-ftp.ecn.purdue.edu::CPAN
```

Once you have the rsync client installed on your system and the disk space mapped out, you may then add an entry to your crontab [if using UNIX, AT if using Windows NT] like the following example:

```
0 20 * * * /usr/local/bin/rsync -av --delete rsync.nic.funet.fi::CPAN
/project/CPAN/ >/dev/null 2>&1
```

```
AT 20:00 /every:M,T,W,Th,F,S,Su "C:\Program Files\Rsync\rsync -av --delete
rsync.nic.funet.fi::CPAN /project/CPAN/ >/dev/null 2>&1"
```

I have Windows 2000/NT/98/95, how can I mirror CPAN?

By using [rsync](#) for [NT](#).

Which CPAN site should I mirror?

The one that gives the best bandwidth (where your mirror finishes quickest) and which is most up-to-date. Most up-to-date is, by definition, the CPAN master site `ftp.funet.fi`. Note that it lives in the GMT+2 time zone so please try not to mirror during working hours: 0600 to 1400 GMT/UTC. If you want to mirror from somewhere else, check the list at <http://www.cpan.org/SITES.html> or at <http://mirror.cpan.org/>.

Please organize the mirroring with the corresponding FTP maintainer (their email addresses from the file <http://www.cpan.org/MIRRORED.BY> or <http://mirror.cpan.org/>) so that you will not overload their site and that your mirror starts just after theirs has finished.

A few tips to keep in mind:

- For your mirror site to be useful to your users you should mirror daily.
 - You can also provide a HTTP interface in addition to an FTP interface to CPAN if you wish to do so.
 - Consider also giving rsync access to your mirror. Many people like rsync because it's very bandwidth-friendly.
 - Remember to tell `cpan@perl.org` about all the access methods you provide to your mirror: ftp, http, or rsync (or any other methods).
-

How do I register my mirror to make it a public resource?

If you want to publicize your CPAN mirror (meaning that your mirror is publicly available), please use the on-line form at <http://mirror.cpan.org/mirror.cgi> or fill in the following template which can be found from the top of the file <http://www.cpan.org/MIRRORED.BY>:

```
hostname.of.the.CPAN.mirroring.site:
frequency          = "daily/bidaily/.../weekly"
dst_ftp            = "ftp://the.same.host.name:/CPAN/mirror/directory/"
dst_http           = "http://the.same.host.name:/CPAN/mirror/directory/"
dst_rsync          = "the.same.host.name::CPAN"
dst_location       = "city, (area?, )country, continent (lat long)"
dst_organisation  = "full organisation name"
dst_timezone       = "GMT[+-]n"
dst_bandwidth      = "Approximate connection speed,e.g. T1, E3, etc."
dst_contact        = "email.address.to.contact@for.this.mirror"
dst_src            = "host.that.you.mirror.from"
dst_loadbal        = "Y" or "N" Join the load balancing pool for ftp.cpan.org
dst_notes          = "(optional field) access restrictions, for example?"
```

There are plenty of examples of how to fill in the template file. When your template is ready, please send it to cpan@perl.org and you will be registered.

