NAME

AutoSplit - split a package for autoloading

SYNOPSIS

```
autosplit($file, $dir, $keep, $check, $modtime);

autosplit_lib_modules(@modules);
```

DESCRIPTION

This function will split up your program into files that the AutoLoader module can handle. It is used by both the standard perl libraries and by the MakeMaker utility, to automatically configure libraries for autoloading.

The `autosplit` interface splits the specified file into a hierarchy rooted at the directory `$dir`. It creates directories as needed to reflect class hierarchy, and creates the file *autosplit.ix*. This file acts as both forward declaration of all package routines, and as timestamp for the last update of the hierarchy.

The remaining three arguments to `autosplit` govern other options to the autosplitter.

$keep

If the third argument, *$keep*, is false, then any pre-existing `*.al` files in the autoload directory are removed if they are no longer part of the module (obsoleted functions). $keep defaults to 0.

$check

The fourth argument, *$check*, instructs `autosplit` to check the module currently being split to ensure that it includes a `use` specification for the AutoLoader module, and skips the module if AutoLoader is not detected. $check defaults to 1.

$modtime

Lastly, the *$modtime* argument specifies that `autosplit` is to check the modification time of the module against that of the `autosplit.ix` file, and only split the module if it is newer. $modtime defaults to 1.

Typical use of AutoSplit in the perl MakeMaker utility is via the command-line with:

```
perl -e 'use AutoSplit; autosplit($ARGV[0], $ARGV[1], 0, 1, 1)'
```

Defined as a Make macro, it is invoked with file and directory arguments; `autosplit` will split the specified file into the specified directory and delete obsolete `.al` files, after checking first that the module does use the AutoLoader, and ensuring that the module is not already currently split in its current form (the modtime test).

The `autosplit_lib_modules` form is used in the building of perl. It takes as input a list of files (modules) that are assumed to reside in a directory **lib** relative to the current directory. Each file is sent to the autosplitter one at a time, to be split into the directory **lib/auto**.

In both usages of the autosplitter, only subroutines defined following the perl *__END__* token are split out into separate files. Some routines may be placed prior to this marker to force their immediate loading and parsing.

Multiple packages

As of version 1.01 of the AutoSplit module it is possible to have multiple packages within a single file. Both of the following cases are supported:

```
package NAME;
__END__
```

```
sub AAA { ... }
package NAME::option1;
sub BBB { ... }
package NAME::option2;
sub BBB { ... }

package NAME;
__END__
sub AAA { ... }
sub NAME::option1::BBB { ... }
sub NAME::option2::BBB { ... }
```

DIAGNOSTICS

`AutoSplit` will inform the user if it is necessary to create the top-level directory specified in the invocation. It is preferred that the script or installation process that invokes `AutoSplit` have created the full directory path ahead of time. This warning may indicate that the module is being split into an incorrect path.

`AutoSplit` will warn the user of all subroutines whose name causes potential file naming conflicts on machines with drastically limited (8 characters or less) file name length. Since the subroutine name is used as the file name, these warnings can aid in portability to such systems.

Warnings are issued and the file skipped if `AutoSplit` cannot locate either the *__END__* marker or a "package Name;"-style specification.

`AutoSplit` will also emit general diagnostics for inability to create directories or files.

AUTHOR

`AutoSplit` is maintained by the perl5-porters. Please direct any questions to the canonical mailing list. Anything that is applicable to the CPAN release can be sent to its maintainer, though.

Author and Maintainer: The Perl5-Porters <perl5-porters@perl.org>

Maintainer of the CPAN release: Steffen Mueller <smueller@cpan.org>

COPYRIGHT AND LICENSE

This package has been part of the perl core since the first release of perl5. It has been released separately to CPAN so older installations can benefit from bug fixes.

This package has the same copyright and license as the perl core:

```
    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.
```