NAME

Module::Load - runtime require of both modules and files

SYNOPSIS

```
use Module::Load;

   my $module = 'Data:Dumper';
   load Data::Dumper;      # loads that module
   load 'Data::Dumper';    # ditto
   load $module            # tritto

   my $script = 'some/script.pl'
   load $script;
   load 'some/script.pl'; # use quotes because of punctuations

   load thing;             # try 'thing' first, then 'thing.pm'

   load CGI, ':standard'   # like 'use CGI qw[:standard]'
```

DESCRIPTION

`load` eliminates the need to know whether you are trying to require either a file or a module.

If you consult `perldoc -f require` you will see that `require` will behave differently when given a bareword or a string.

In the case of a string, `require` assumes you are wanting to load a file. But in the case of a bareword, it assumes you mean a module.

This gives nasty overhead when you are trying to dynamically require modules at runtime, since you will need to change the module notation (`Acme::Comment`) to a file notation fitting the particular platform you are on.

`load` eliminates the need for this overhead and will just DWYM.

Rules

`load` has the following rules to decide what it thinks you want:

- If the argument has any characters in it other than those matching `\w`, `:` or `'`, it must be a file
- If the argument matches only `[\w:']`, it must be a module
- If the argument matches only `\w`, it could either be a module or a file. We will try to find `file` first in `@INC` and if that fails, we will try to find `file.pm` in @INC. If both fail, we die with the respective error messages.

Caveats

Because of a bug in perl (#19213), at least in version 5.6.1, we have to hardcode the path separator for a require on Win32 to be `/`, like on Unix rather than the Win32 `\`. Otherwise perl will not read its own %INC accurately double load files if they are required again, or in the worst case, core dump.

`Module::Load` cannot do implicit imports, only explicit imports. (in other words, you always have to specify explicitly what you wish to import from a module, even if the functions are in that modules' `@EXPORT`)

ACKNOWLEDGEMENTS

Thanks to Jonas B. Nielsen for making explicit imports work.

BUG REPORTS

Please report bugs or other issues to <bug-module-load@rt.cpan.org<gt>.

AUTHOR

This module by Jos Boumans <kane@cpan.org>.

COPYRIGHT