NAME

ExtUtils::Mksymlists – write linker options files for dynamic extension

SYNOPSIS

```
use ExtUtils::Mksymlists;
Mksymlists(  NAME     => $name ,
             DL_VARS  => [ $var1, $var2, $var3 ],
             DL_FUNCS => { $pkg1 => [ $func1, $func2 ],
                           $pkg2 => [ $func3 ] );
```

DESCRIPTION

`ExtUtils::Mksymlists` produces files used by the linker under some OSs during the creation of shared libraries for dynamic extensions. It is normally called from a MakeMaker-generated Makefile when the extension is built. The linker option file is generated by calling the function `Mksymlists`, which is exported by default from `ExtUtils::Mksymlists`. It takes one argument, a list of key-value pairs, in which the following keys are recognized:

DLBASE

This item specifies the name by which the linker knows the extension, which may be different from the name of the extension itself (for instance, some linkers add an '_' to the name of the extension). If it is not specified, it is derived from the NAME attribute. It is presently used only by OS2 and Win32.

DL_FUNCS

This is identical to the DL_FUNCS attribute available via MakeMaker, from which it is usually taken. Its value is a reference to an associative array, in which each key is the name of a package, and each value is an a reference to an array of function names which should be exported by the extension. For instance, one might say `DL_FUNCS => { Homer::Iliad => [ qw(trojans greeks) ], Homer::Odyssey => [ qw(travellers family suitors) ] }`. The function names should be identical to those in the XSUB code; `Mksymlists` will alter the names written to the linker option file to match the changes made by *xsubpp*. In addition, if none of the functions in a list begin with the string **boot_**, `Mksymlists` will add a bootstrap function for that package, just as xsubpp does. (If a **boot_<pkg>** function is present in the list, it is passed through unchanged.) If DL_FUNCS is not specified, it defaults to the bootstrap function for the extension specified in NAME.

DL_VARS

This is identical to the DL_VARS attribute available via MakeMaker, and, like DL_FUNCS, it is usually specified via MakeMaker. Its value is a reference to an array of variable names which should be exported by the extension.

FILE

This key can be used to specify the name of the linker option file (minus the OS-specific extension), if for some reason you do not want to use the default value, which is the last word of the NAME attribute (*e.g.* for `Tk::Canvas`, FILE defaults to `Canvas`).

FUNCLIST

This provides an alternate means to specify function names to be exported from the extension. Its value is a reference to an array of function names to be exported by the extension. These names are passed through unaltered to the linker options file. Specifying a value for the FUNCLIST attribute suppresses automatic generation of the bootstrap function for the package. To still create the bootstrap name you have to specify the package name in the DL_FUNCS hash:

```
Mksymlists(  NAME     => $name ,
             FUNCLIST => [ $func1, $func2 ],
             DL_FUNCS => { $pkg => [] } );
```

IMPORTS

This attribute is used to specify names to be imported into the extension. It is currently only used by OS/2 and Win32.

NAME

This gives the name of the extension (*e.g.* `Tk::Canvas`) for which the linker option file will be produced.

When calling `Mksymlists`, one should always specify the NAME attribute. In most cases, this is all that's necessary. In the case of unusual extensions, however, the other attributes can be used to provide additional information to the linker.

AUTHOR
Charles Bailey *<bailey AT newman DOT upenn DOT edu>*

REVISION
Last revised 14–Feb–1996, for Perl 5.002.