NAME

ExtUtils::XSSymSet - keep sets of symbol names palatable to the VMS linker

SYNOPSIS

```
use ExtUtils::XSSymSet;

$set = new ExtUtils::XSSymSet;
while ($sym = make_symbol()) { $set->addsym($sym); }
foreach $safesym ($set->all_trimmed) {
print "Processing $safesym (derived from ",
$self->get_orig($safesym), ")\n";
do_stuff($safesym);
}

$safesym = ExtUtils::XSSymSet->trimsym($onesym);
```

DESCRIPTION

Since the VMS linker distinguishes symbols based only on the first 31 characters of their names, it is occasionally necessary to shorten symbol names in order to avoid collisions. (This is especially true of names generated by xsubpp, since prefixes generated by nested package names can become quite long.) ExtUtils::XSSymSet provides functions to shorten names in a consistent fashion, and to track a set of names to insure that each is unique. While designed with *xsubpp* in mind, it may be used with any set of strings.

This package supplies the following functions, all of which should be called as methods.

new([$maxlen[,$silent]])

Creates an empty ExtUtils::XSSymset set of symbols. This function may be called as a static method or via an existing object. If $maxlen or $silent are specified, they are used as the defaults for maximum name length and warning behavior in future calls to *addsym()* or *trimsym()* via this object. If the compiler has been instructed to do its own symbol shortening via $Config{'useshortenedsymbols'}, a value of 2048 is assumed for $maxlen as a way of bypassing the shortening done by this module.

addsym($name[,$maxlen[,$silent]])

Creates a symbol name from $name, using the methods described under *trimsym()*, which is unique in this set of symbols, and returns the new name. $name and its resultant are added to the set, and any future calls to *addsym()* specifying the same $name will return the same result, regardless of the value of $maxlen specified. Unless $silent is true, warnings are output if $name had to be trimmed or changed in order to avoid collision with an existing symbol name. $maxlen and $silent default to the values specified when this set of symbols was created. This method must be called via an existing object.

trimsym($name[,$maxlen[,$silent]])

Creates a symbol name $maxlen or fewer characters long from $name and returns it. If $name is too long, it first tries to shorten it by removing duplicate characters, then by periodically removing non-underscore characters, and finally, if necessary, by periodically removing characters of any type. $maxlen defaults to 31. Unless $silent is true, a warning is output if $name is altered in any way. This function may be called either as a static method or via an existing object, but in the latter case no check is made to insure that the resulting name is unique in the set of symbols. If the compiler has been instructed to do its own symbol shortening via $Config{'useshortenedsymbols'}, a value of 2048 is assumed for $maxlen as a way of bypassing the shortening done by this module.

delsym($name)

Removes $name from the set of symbols, where $name is the original symbol name passed previously to *addsym()*. If $name existed in the set of symbols, returns its ''trimmed'' equivalent, otherwise returns undef. This method must be called via an existing object.

get_orig($trimmed)
>   Returns the original name which was trimmed to `$trimmed` by a previous call to *addsym()*, or
>   `undef` if `$trimmed` does not correspond to a member of this set of symbols. This method
>   must be called via an existing object.

get_trimmed($name)
>   Returns the trimmed name which was generated from `$name` by a previous call to *addsym()*,
>   or `undef` if `$name` is not a member of this set of symbols. This method must be called via an
>   existing object.

all_orig()
>   Returns a list containing all of the original symbol names from this set.

all_trimmed()
>   Returns a list containing all of the trimmed symbol names from this set.

AUTHOR

Charles Bailey *<bailey@newman.upenn.edu>*

REVISION

Last revised 8-Oct-2010, for Perl 5.13.6.