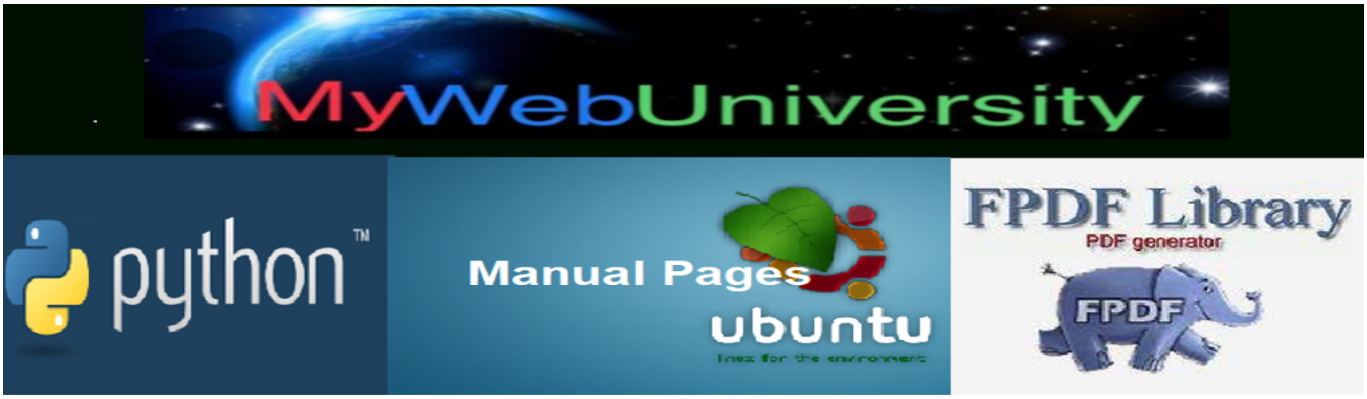

Rocky Enterprise Linux 9.2 Manual Pages on command 'Glib::Flags.3pm'

$ man Glib::Flags.3pm

Glib::Flags(3pm) User Contributed Perl Documentation Glib::Flags(3pm)

NAME

Glib::Flags - methods and overloaded operators for flags

HIERARCHY

Glib::Flags

DESCRIPTION

Glib maps flag and enum values to the nicknames strings provided by the underlying C libraries. Representing flags this way in Perl is an interesting problem, which Glib solves by using some cool overloaded operators.

The functions described here actually do the work of those overloaded operators. See the description of the flags operators in the "This Is Now That" section of Glib for more info.

METHODS

scalar = $class->new ($a)

? $a (scalar)

Create a new flags object with given bits. This is for use from a subclass, it's not possible to create a "Glib::Flags" object as such. For example,

```
my $f1 = Glib::ParamFlags->new ('readable');
my $f2 = Glib::ParamFlags->new (['readable','writable']);
```

An object like this can then be used with the overloaded operators.

scalar = $a->all ($b, $swap)

? $b (scalar)

? $swap (scalar)

aref = $f->as_arrayref

Return the bits of $f as a reference to an array of strings, like ['flagbit1','flagbit2'].

This is the overload function for "@{}", ie. arrayizing $f. You can call it directly as a method too.

Note that @$f gives the bits as a list, but as_arrayref gives an arrayref. If an arrayref is what you want then the method style somefunc()->as_arrayref can be more readable than [@{somefunc()}].

bool = $f->bool

Return 1 if any bits are set in $f, or 0 if none are set. This is the overload for $f in boolean context (like "if", etc). You can call it as a method to get a true/false directly too.

integer = $a->eq ($b, $swap)

? $b (scalar)

? $swap (integer)

integer = $a->ge ($b, $swap)

? $b (scalar)

? $swap (integer)

scalar = $a->intersect ($b, $swap)

? $b (scalar)

? $swap (scalar)

integer = $a->ne ($b, $swap)

? $b (scalar)

? $swap (integer)

scalar = $a->sub ($b, $swap)

? $b (scalar)

? $swap (scalar)

scalar = $a->union ($b, $swap)

? $b (scalar)

? $swap (scalar)

scalar = $a->xor ($b, $swap)

? $b (scalar)

? $swap (scalar)

SEE ALSO

Glib

COPYRIGHT

This software is licensed under the LGPL. See Glib for a full notice.