

**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

Copyright (C) 2003–2011 by the gtk2–perl team.

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