



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'B::Op_private.3perl'

\$ man B::Op_private.3perl

B::Op_private(3perl) Perl Programmers Reference Guide B::Op_private(3perl)

NAME

B::Op_private - OP op_private flag definitions

SYNOPSIS

```
use B::Op_private;

# flag details for bit 7 of OP_AELEM's op_private:
my $name = $B::Op_private::bits{aelem}{7}; # OPpLVAL_INTRO
my $value = $B::Op_private::defines{$name}; # 128
my $label = $B::Op_private::labels{$name}; # LVINTRO

# the bit field at bits 5..6 of OP_AELEM's op_private:
my $bf = $B::Op_private::bits{aelem}{6};
my $mask = $bf->{bitmask}; # etc
```

DESCRIPTION

This module provides four global hashes:

```
%B::Op_private::bits
%B::Op_private::defines
%B::Op_private::labels
```

```
%B::Op_private::ops_using
```

which contain information about the per-op meanings of the bits in the op_private field.

```
%bits
```

This is indexed by op name and then bit number (0..7). For single bit flags, it returns the name of the define (if any) for that bit:

```
$B::Op_private::bits{aelem}{7} eq 'OPpLVAL_INTRO';
```

For bit fields, it returns a hash ref containing details about the field. The same reference will be returned for all bit positions that make up the bit field; so for example these both return the same hash ref:

```
$bitfield = $B::Op_private::bits{aelem}{5};
```

```
$bitfield = $B::Op_private::bits{aelem}{6};
```

The general format of this hash ref is

```
{  
  # The bit range and mask; these are always present.  
  bitmin    => 5,  
  bitmax    => 6,  
  bitmask   => 0x60,  
  
  # (The remaining keys are optional)  
  
  # The names of any defines that were requested:  
  mask_def  => 'OPpFOO_MASK',  
  baseshift_def => 'OPpFOO_SHIFT',  
  bitcount_def => 'OPpFOO_BITS',  
  
  # If present, Concise etc will display the value with a 'FOO='
```

prefix. If it equals '-', then Concise will treat the bit

field as raw bits and not try to interpret it.

```
label    => 'FOO',
```

If present, specifies the names of some defines and the

display labels that are used to assign meaning to particu-

lar integer values within the bit field; e.g. 3 is dis-

played as 'C'.

```
enum     => [ qw(
```

```
    1  OPpFOO_A  A
```

```
    2  OPpFOO_B  B
```

```
    3  OPpFOO_C  C
```

```
    )],
```

```
};
```

%defines

This gives the value of every "OPp" define, e.g.

```
$B::Op_private::defines{OPpLVAL_INTRO} == 128;
```

%labels

This gives the short display label for each define, as used by "B::Concise" and "perl

-Dx", e.g.

```
$B::Op_private::labels{OPpLVAL_INTRO} eq 'LVINTRO';
```

If the label equals '-', then Concise will treat the bit as a raw bit and not try to display it symbolically.

%ops_using

For each define, this gives a reference to an array of op names that use the flag.

```
@ops_using_lvintro = @{$B::Op_private::ops_using{OPp_LVAL_INTRO}};
```

perl v5.34.0

2023-11-23

B::Op_private(3perl)