

NAMF

attrs - set/get attributes of a subroutine (deprecated)

SYNOPSIS

```
sub foo {
    use attrs qw(locked method);
    ...
}

@a = attrs::get(\&foo);
```

DESCRIPTION

NOTE: Use of this pragma is deprecated. Use the syntax

```
sub foo : locked method { }
```

to declare attributes instead. See also attributes.

This pragma lets you set and get attributes for subroutines. Setting attributes takes place at compile time; trying to set invalid attribute names causes a compile-time error. Calling attrs::get on a subroutine reference or name returns its list of attribute names. Notice that attrs::get is not exported. Valid attributes are as follows.

method

Indicates that the invoking subroutine is a method.

locked

Setting this attribute is only meaningful when the subroutine or method is to be called by multiple threads. When set on a method subroutine (i.e. one marked with the **method** attribute above), perl ensures that any invocation of it implicitly locks its first argument before execution. When set on a non-method subroutine, perl ensures that a lock is taken on the subroutine itself before execution. The semantics of the lock are exactly those of one explicitly taken with the lock operator immediately after the subroutine is entered.