

## NAME

base - Establish IS-A relationship with base classes at compile time

## SYNOPSIS

```
package Baz;
use base qw(Foo Bar);
```

## DESCRIPTION

Allows you to both load one or more modules, while setting up inheritance from those modules at the same time. Roughly similar in effect to

```
package Baz;
BEGIN {
    require Foo;
    require Bar;
    push @ISA, qw(Foo Bar);
}
```

If any of the listed modules are not loaded yet, *base* silently attempts to *require* them (and silently continues if the *require* failed). Whether to *require* a base class module is determined by the absence of a global variable `$VERSION` in the base package. If `$VERSION` is not detected even after loading it, `<base>` will define `$VERSION` in the base package, setting it to the string `-1`, set by `base.pm`.

Will also initialize the fields if one of the base classes has it. Multiple inheritance of fields is **NOT** supported, if two or more base classes each have inheritable fields the 'base' pragma will croak. See *fields*, *public* and *protected* for a description of this feature.

## DIAGNOSTICS

Base class package "%s" is empty.

base.pm was unable to require the base package, because it was not found in your path.

## HISTORY

This module was introduced with Perl 5.004\_04.

## CAVEATS

Due to the limitations of the implementation, you must use *base* *before* you declare any of your own fields.

## SEE ALSO

*fields*