

## NAME

User::grent - by-name interface to Perl's built-in getgr\*() functions

## SYNOPSIS

```
use User::grent;
$gr = getgrgid(0) or die "No group zero";
if ( $gr->name eq 'wheel' && @{$gr->members} > 1 ) {
    print "gid zero name wheel, with other members";
}
```

```
use User::grent qw(:FIELDS);
getgrgid(0) or die "No group zero";
if ( $gr_name eq 'wheel' && @gr_members > 1 ) {
    print "gid zero name wheel, with other members";
}
```

```
$gr = getgr($whoever);
```

## DESCRIPTION

This module's default exports override the core `getgrent()`, `getgruid()`, and `getgrnam()` functions, replacing them with versions that return "User::grent" objects. This object has methods that return the similarly named structure field name from the C's `passwd` structure from *grp.h*; namely `name`, `passwd`, `gid`, and `members` (not `mem`). The first three return scalars, the last an array reference.

You may also import all the structure fields directly into your namespace as regular variables using the `:FIELDS` import tag. (Note that this still overrides your core functions.) Access these fields as variables named with a preceding `gr_`. Thus, `$group_obj->gid()` corresponds to `$gr_gid` if you import the fields. Array references are available as regular array variables, so `@{ $group_obj->members() }` would be simply `@gr_members`.

The `getpw()` function is a simple front-end that forwards a numeric argument to `getpwuid()` and the rest to `getpwnam()`.

To access this functionality without the core overrides, pass the `use` an empty import list, and then access function functions with their full qualified names. On the other hand, the built-ins are still available via the `CORE::` pseudo-package.

## NOTE

While this class is currently implemented using the `Class::Struct` module to build a struct-like class, you shouldn't rely upon this.

## AUTHOR

Tom Christiansen