

**NAME**

**cap\_getgrent, cap\_getgrnam, cap\_getgrgid, cap\_getgrent\_r, cap\_getgrnam\_r, cap\_getgrgid\_r, cap\_setgroupent, cap\_setgrent, cap\_endgrent, cap\_grp\_limit\_cmds, cap\_grp\_limit\_fields, cap\_grp\_limit\_groups** - library for group database operations in capability mode

**LIBRARY**

library "libcap\_grp"

**SYNOPSIS**

```
#include <sys/nv.h>
```

```
#include <libcasper.h>
```

```
#include <casper/cap_grp.h>
```

```
struct group *
```

```
cap_getgrent(cap_channel_t *chan);
```

```
struct group *
```

```
cap_getgrnam(cap_channel_t *chan, const char *name);
```

```
struct group *
```

```
cap_getgrgid(cap_channel_t *chan, gid_t gid);
```

```
int
```

```
cap_getgrent_r(cap_channel_t *chan, struct group *grp, char *buffer, size_t bufsize,  
  struct group **result);
```

```
int
```

```
cap_getgrnam_r(cap_channel_t *chan, const char *name, struct group *grp, char *buffer, size_t bufsize,  
  struct group **result);
```

```
int
```

```
cap_getgrgid_r(cap_channel_t *chan, gid_t gid, struct group *grp, char *buffer, size_t bufsize,  
  struct group **result);
```

```
int
```

```
cap_setgroupent(cap_channel_t *chan, int stayopen);
```

```
int
```

```
cap_setgrent(cap_channel_t *chan);
```

*void*

```
cap_endgrent(cap_channel_t *chan);
```

*int*

```
cap_grp_limit_cmds(cap_channel_t *chan, const char * const *cmds, size_t ncmds);
```

*int*

```
cap_grp_limit_fields(cap_channel_t *chan, const char * const *fields, size_t nfields);
```

*int*

```
cap_grp_limit_groups(cap_channel_t *chan, const char * const *names, size_t nnames,
    const gid_t *gids, size_t ngids);
```

## DESCRIPTION

The functions **cap\_getgrent()**, **cap\_getgrnam()**, **cap\_getgrgid()**, **cap\_getgrent\_r()**, **cap\_getgrnam\_r()**, **cap\_getgrgid\_r()**, **cap\_setgroupernt()**, **cap\_setgrent()**, and **cap\_endgrent()** are respectively equivalent to **getgrent(3)**, **getgrnam(3)**, **getgrgid(3)**, **getgrent\_r(3)**, **getgrnam\_r(3)**, **getgrgid\_r(3)**, **setgroupernt(3)**, **setgrent(3)**, and **endgrent(3)** except that the connection to the **system.grp** service needs to be provided.

The **cap\_grp\_limit\_cmds()** function limits the functions allowed in the service. The *cmds* variable can be set to **getgrent**, **getgrnam**, **getgrgid**, **getgrent\_r**, **getgrnam\_r**, **getgrgid\_r**, **setgroupernt**, **setgrent**, or **endgrent** which will allow to use the function associated with the name. The *ncmds* variable contains the number of *cmds* provided.

The **cap\_grp\_limit\_fields()** function allows limit fields returned in the structure *group*. The *fields* variable can be set to **gr\_name** **gr\_passwd** **gr\_gid** or **gr\_mem**. The field which was set as the limit will be returned, while the rest of the values not set this way will have default values. The *nfields* variable contains the number of *fields* provided.

The **cap\_grp\_limit\_groups()** function allows to limit access to groups. The *names* variable allows to limit groups by name and the *gids* variable by the group number. The *nnames* and *ngids* variables provide numbers of limited names and gids.

## EXAMPLES

The following example first opens a capability to casper and then uses this capability to create the **system.grp** casper service and uses it to get a group name.

```
cap_channel_t *capcas, *capgrp;
const char *cmds[] = { "getgrgid" };
const char *fields[] = { "gr_name" };
```

```
const gid_t gid[] = { 1 };
struct group *group;

/* Open capability to Casper. */
capcas = cap_init();
if (capcas == NULL)
    err(1, "Unable to contact Casper");

/* Enter capability mode sandbox. */
if (cap_enter() < 0 && errno != ENOSYS)
    err(1, "Unable to enter capability mode");

/* Use Casper capability to create capability to the system.grp service. */
capgrp = cap_service_open(capcas, "system.grp");
if (capgrp == NULL)
    err(1, "Unable to open system.grp service");

/* Close Casper capability, we don't need it anymore. */
cap_close(capcas);

/* Limit service to one single function. */
if (cap_grp_limit_cmds(capgrp, cmds, nitems(cmds))
    err(1, "Unable to limit access to system.grp service");

/* Limit service to one field as we only need name of the group. */
if (cap_grp_limit_fields(capgrp, fields, nitems(fields))
    err(1, "Unable to limit access to system.grp service");

/* Limit service to one gid. */
if (cap_grp_limit_groups(capgrp, NULL, 0, gid, nitems(gid))
    err(1, "Unable to limit access to system.grp service");

group = cap_getgrgid(capgrp, gid[0]);
if (group == NULL)
    err(1, "Unable to get name of group");

printf("GID %d is associated with name %s.\n", gid[0], group->gr_name);

cap_close(capgrp);
```

**SEE ALSO**

cap\_enter(2), endgrent(3), err(3), getgrent(3), getgrent\_r(3), getgrgid(3), getgrgid\_r(3), getgrnam(3), getgrnam\_r(3), setgrent(3), setgroupent(3), capsicum(4), nv(9)

**HISTORY**

The **cap\_grp** service first appeared in FreeBSD 10.3.

**AUTHORS**

The **cap\_grp** service was implemented by Pawel Jakub Dawidek <[pawel@dawidek.net](mailto:pawel@dawidek.net)> under sponsorship from the FreeBSD Foundation.

This manual page was written by  
Mariusz Zaborski <[oshogbo@FreeBSD.org](mailto:oshogbo@FreeBSD.org)>.