NAME

nis_groups, nis_ismember, nis_addmember, nis_removemember, nis_creategroup, nis_destroygroup, nis_verifygroup, nis_print_group_entry, nis_map_group, __nis_map_group – NIS+ group manipulation functions

SYNOPSIS

cc [*flag* ...] *file*... **-lnsl** [*library*...]

#include <rpcsvc/nis.h>

bool_t nis_ismember(const nis_name *principal***, const nis_name** *group***);**

nis_error nis_addmember(const nis_name *member***, const nis_name** *group***);**

nis_error nis_removemember(const nis_name *member***, const nis_name** *group***);**

nis_error nis_creategroup(const nis_name *group***, const u_long** *flags***);**

nis_error nis_destroygroup(const nis_name *group***);**

void nis_print_group_entry(const nis_name *group***);**

nis_error nis_verifygroup(const nis_name *group***);**

MT-LEVEL

MT-Safe

DESCRIPTION

These functions manipulate NIS+ groups. They are used by NIS+ clients and servers, and are the interfaces to the group authorization object.

The names of NIS+ groups are syntactically similar to names of NIS+ objects but they occupy a separate namespace. A group named "a.b.c.d." is represented by a NIS+ group object named "a.groups_dir.b.c.d."; the functions described here all expect the name of the group, not the name of the corresponding group object.

There are three types of group members:

- An *explicit* member is just a NIS+ principal-name, for example "wickedwitch.west.oz."
- An *implicit* ("domain") member, written "∗.west.oz.", means that all principals in the given domain belong to this member. No other forms of wildcarding are allowed: "wickedwitch.∗.oz." is invalid, as is "wickedwitch.west.∗.". Note that principals in subdomains of the given domain are *not* included.
- A *recursive* ("group") member, written "@cowards.oz.", refers to another group; all principals that belong to that group are considered to belong here.

Any member may be made *negative* by prefixing it with a minus sign ('−'). A group may thus contain explicit, implicit, recursive, negative explicit, negative implicit, and negative recursive members.

A principal is considered to belong to a group if it belongs to at least one non-negative group member of the group and belongs to no negative group members.

The **nis_ismember( )** function returns TRUE if it can establish that *principal* belongs to *group*; otherwise it returns FALSE.

The **nis_addmember( )** and **nis_removemember( )** functions add or remove a member. They do not check whether the member is valid. The user must have read and modify rights for the group in question.

The **nis_creategroup( )** and **nis_destroygroup( )** functions create and destroy group objects. The user must have create or destroy rights, respectively, for the *groups_dir* directory in the appropriate domain. The parameter *flags* to **nis_creategroup( )** is currently unused and should be set to zero.

The **nis_print_group_entry( )** function lists a group's members on the standard output.

The **nis_verifygroup( )** function returns **NIS_SUCCESS** if the given group exists, otherwise it returns an error code.

EXAMPLES

Simple Memberships

Given a group **sadsouls.oz.** with members **tinman.oz.**, **lion.oz.**, and **scarecrow.oz.**, the function call

bool_var = nis_ismember("lion.oz.", "sadsouls.oz.");

will return 1 (TRUE) and the function call

bool_var = nis_ismember("toto.oz.", "sadsouls.oz.");

will return 0 (FALSE).

Implicit Memberships

Given a group **baddies.oz.**, with members **wickedwitch.west.oz.** and ∗**.monkeys.west.oz.**, the function call

bool_var = nis_ismember("hogan.monkeys.west.oz.", "baddies.oz.");

will return 1 (TRUE) because any principal from the **monkeys.west.oz.** domain belongs to the implicit group ∗**.monkeys.west.oz.**, but the function call

bool_var = nis_ismember("hogan.big.monkeys.west.oz.", "baddies.oz.");

will return 0 (FALSE).

Recursive Memberships

Given a group **goodandbad.oz.**, with members **toto.kansas**, **@sadsouls.oz.**, and **@baddies.oz.**, and the groups **sadsouls.oz.** and **baddies.oz.** defined above, the function call

bool_var = nis_ismember("wickedwitch.west.oz.", "goodandbad.oz.");

will return 1 (TRUE), because **wickedwitch.west.oz.** is a member of the **baddies.oz.** group which is recursively included in the **goodandbad.oz.** group.

SEE ALSO

nisgrpadm(1), **nis_objects**(3N)

NOTES

These functions only accept fully-qualified NIS+ names.

A group is represented by a NIS+ object (see **nis_objects**(3N)) with a variant part that is defined in the **group_obj** structure. It contains the following fields:

```
u_long          gr_flags;          /* Interpretation Flags
                                      (currently unused) */
struct {
        u_int       gr_members_len;
        nis_name    *gr_members_val;
} gr_members;                      /* Array of members */
```

NIS+ servers and clients maintain a local cache of expanded groups to enhance their performance when checking for group membership. Should the membership of a group change, servers and clients with that group cached will not see the change until either the group cache has expired or it is explicitly flushed. A server's cache may be flushed programmatically by calling the **nis_servstate( )** function with tag **TAG_GCACHE** and a value of 1.

There are currently no known methods for **nis_ismember( )**, **nis_print_group_entry( )**, and **nis_verifygroup( )** to get their answers from only the master server.