MiscUserGroup

Inherits From:	Object
Declared In:	<misckit/MiscUserGroup.h>

Class Description

This class is basically a wrapper around the notion of a UNIX system group. You can initialize a group either by groupId (gid) or by group name, then ask the object for the other. You can ask an instance to give you a List of all it's members, or ask the MiscUserGroup object for all the groups on your system. It is basically a very simple class that is supposed to fit well with the MiscFile and MiscUser classes.

Since the underlying Unix functions used to retrieve the group information will climb the domain hierarchy, the information returned by either the class method +**allGroups** or the instance method -**members** is representative of all the information in all the domains (with duplicate entries removed).

For instance, if your groups were setup like:

```
  Local domain:
    wheel - members = root, me
```

```
   other - members = me
   uucp - members = uucp
Root domain:
   wheel - members = root, joe, fred
   other - members = joe, fred, ed, ted
   uucp - members = uucp
   games - members = <none>

+ allGroups would return : wheel, other, uucp,  games.
- members (for instance representing wheel) would return: root, me, joe, fred.
```

Instance Variables

int **gid**;
char ***groupname**;

gid	The unique group id.
groupname	The character description of the group.

Method Types

+ allGroups

- initWithGroupId:
- initWithGroupName:
- free

- groupId
- groupName
- members

Class Methods

allGroups

+ (List *)**allGroups**

Returns a List containing MiscUserGroup objects representing all the groups on the system. If there are no groups on your system (highly unlikely I'd suspect) then **nil** is returned. You are responsible for freeing both the returned List and it's contents. This list will contain groups from all the domains from the local to root. Duplicate entries are removed.

Instance Methods

free

- **free**

Frees the groupname.

groupId

- (int)**groupId**

Returns the unique gid that the receiver represents.

See also: ± groupName

groupName

- (const char *)**groupName**

Returns the group name that the receiver represents.

See also: ± groupId

initWithGroupId:

- **initWithGroupId:**(int)*groupId*

Initializes a MiscUserGroup class, given it's *groupId*. If there is no such *groupId* on the system, then the receiver is freed and **nil** is returned. This method is the designated initializer.

See also: ± initWithGroupName:

initWithGroupName:

- **initWithGroupName:**(const char *)*gname*

Initializes a MiscUserGroup class, given it's name. If there is no such group on the system, the receiver is freed and **nil** is returned. If *gname* is NULL, **nil** is returned.

Example:

```
MiscUserGroup *wheel = [ [MiscUserGroup alloc] initWithGroupName: "wheel"];
```

See also: ± initWithGroupId:

members

- (List *)**members**

Returns a list of MiscUsers, representing all the members of the group the receiver represents. If the group has no members, **nil** will be returned. You are responsible for freeing both the returned List and it's contents. All the members in all the occurrences of the given group throughout the domain hierarchy will be merged together and returned. Duplicates are removed.