

MiscFile (Modification)

Inherits From: MiscGraphNode

Declared In: <misckit/MiscFile.h>

Category Description

A category that allow modification to the underlying characteristics of the file that the instance of MiscFile represents. For now, all the methods deal with changing file permissions. I suppose there could be **changeGroup:** and **changeOwner:** methods added, though only root would be able to change owners of files anyways. As with methods in other categories of MiscFile, if a method returns a MISCFILE_ERROR (or NULL or nil for some methods), you can use **-errorCode** (MiscFile+Unix) to find the reason for the failure. Most of the file errors are listed in MiscFileDefs.h

If you have any suggestions of other methods that would fit in this category, let me know...

Method Types

- `addPermissions:for:`
- `removePermissions:for:`
- `setPermissions:for:`

Instance Methods

`addPermissions:for:`

- `(int)`**`addPermissions:(int)`***perms*
`for:(int)`*who*

Adds the given permissions to the file the receiver represents. The *perms* arguments can either one of or an ORed combination of either `MISCFILE_READ`, `MISCFILE_WRITE`, `MISCFILE_EXECUTE`, `MISCFILE_STICKY`, `MISCFILE_SETUID`, or `MISCFILE_SETGID`. The *who* argument can be one of or an ORed combination of `MISCFILE_OWNER`, `MISCFILE_GROUP`, or `MISCFILE_OTHER`. If the change of permissions was successful, `MISCFILE_SUCCESS` will be returned, otherwise `MISCFILE_ERROR`.

Example:

```
[someFile addPermissions: MISCFILE_EXECUTE for: MISCFILE_OTHER]; // equivalent to chmod o+x.
```

See also: `± removePermissions:for:`, `± setPermissions:for:`

`removePermissions:for:`

- `(int)`**`removePermissions:(int)`***perms*
`for:(int)`*who*

Removes the given permissions from the file the receiver represents. The *perms* and *who* arguments are the same as the **`addPermissions:for:`** method accepts. The return value is also the same. If you attempt to remove a permission that wasn't there in the first place, this method will still return `MISCFILE_SUCCESS` if you would have had the permission to remove it.

See also: `± addPermissions:for:`, `± setPermissions:for:`

setPermissions:for:

- (int)**setPermissions:**(int)*perms*
 for:(int)*who*

This method will erase all permissions that the file originally had, and set it according to the arguments *perms* and *who*. The arguments and return value for this method are the same as for the **removePermissions:for:** and **addPermissions:for:** methods.

See also: `± addPermissions:for:`, `± removePermissions:for:`