

# Application(MiscAppExtensions)

**Declared In:**           misckit/MiscAppExtensions.h

## Category Description

The MiscAppExtensions category provides extra information about an application, making it easy to find out more about the application's environment. It provides things such as the directory of the application, the user and group ID's that are applicable for the application, the user's home directory, the full name of the user, the host ID of the machine, and so on. Basically, it provides a convenient wrapper around several typical UNIX functions.

If you used the NEXTSTEP 2.x version of this object, note that with the advent of NEXTSTEP 3.0, the **ExtendedApp.rtf;appDirectory;-+appDirectory** method became redundant, and has therefore become a wrapper around the appropriate NXBundle function. (This means±or at least implies±that it no longer provides a trailing <sup>a/o</sup> on the directory name, too. If you wish to retain that behavior, you could do it with a subclass of Application which overrides this method, or use a category to add a new method with the desired behavior.)

## Method Types

Getting information

```
;ExtendedApp.rtf;appDirectory;± appDirectory  
;MiscAppExtensions.rtf;didAutolaunch;± didAutolaunch  
;ExtendedApp.rtf;domainName;± domainName  
;ExtendedApp.rtf;effectiveGroupIDNum;± effectiveGroupIDNum  
;ExtendedApp.rtf;effectiveUserIDNum;± effectiveUserIDNum  
;ExtendedApp.rtf;groupIDNum;± groupIDNum  
;ExtendedApp.rtf;groupMemberForID;± groupMemberForID:  
;ExtendedApp.rtf;groupMemberForString;± groupMemberForString:  
;ExtendedApp.rtf;hostID;± hostID  
;MiscAppExtensions.rtf;knowsDefaultValue;± knowsDefaultValue:  
;ExtendedApp.rtf;realDomainName;± realDomainName  
;ExtendedApp.rtf;realHostName;± realHostName  
;ExtendedApp.rtf;realNameFor;± realNameFor:  
;ExtendedApp.rtf;userHomeDirectory;± userHomeDirectory  
;ExtendedApp.rtf;userIDNum;± userIDNum  
;ExtendedApp.rtf;userLoginName;± userLoginName  
;ExtendedApp.rtf;userRealName;± userRealName
```

## Instance Methods

**appDirectory;**±**appDirectory**  
- (const char \*)**appDirectory**

Returns the directory in which the application's executable resides (the app wrapper). There is no trailing <sup>a/o</sup> on the path name, so if you plan to concatenate a file name to the path, don't forget to first append the <sup>a/o</sup>.

See also: **-directory** (NXBundle class)

**didAutolaunch;¬didAutolaunch**

- (BOOL)**didAutolaunch**

Returns whether the application was launched by the user or by some autolaunch mechanism (e.g. Service, Workspace autolaunch path,...).

**domainName;¬domainName**

- (NXAtom)**domainName**

Returns the name of the domain the machine is on. See the man page for getdomainname() for more information. This is identical to the **±realDomainName** method, but exists for completeness and orthogonality of the API. **Future implementations should take care of the NXHost mechanism and take is domain info into account. Then the **hostName** method should change to really return the host name only.**

See also: **;ExtendedApp.rtf;realDomainName;¬-realDomainName**

**effectiveGroupIDNum;¬effectiveGroupIDNum**

- (int)**effectiveGroupIDNum**

Returns the results of the getegid() function.

See also: **;ExtendedApp.rtf;effectiveUserIDNum;¬-effectiveUserIDNum, ;ExtendedApp.rtf;groupIDNum;¬-groupIDNum, and ;ExtendedApp.rtf;userIDNum;¬-userIDNum.**

**effectiveUserIDNum;¬effectiveUserIDNum**

- (int)**effectiveUserIDNum**

Returns the results of the `geteuid()` function.

See also: **;ExtendedApp.rtf;effectiveGroupIDNum;¬-effectiveGroupIDNum,**  
**;ExtendedApp.rtf;groupIDNum;¬-groupIDNum,** and **;ExtendedApp.rtf;userIDNum;¬-userIDNum.**

**groupIDNum;¬groupIDNum**

- (int)**groupIDNum**

Returns the results of the `getgid()` function.

See also: **;ExtendedApp.rtf;effectiveGroupIDNum;¬-effectiveGroupIDNum,**  
**;ExtendedApp.rtf;effectiveUserIDNum;¬-effectiveUserIDNum,** and  
**;ExtendedApp.rtf;userIDNum;¬-userIDNum.**

**groupMemberForID;;¬groupMemberForID:**

- (BOOL)**groupMemberForID:(gid\_t)anId**

Returns YES if the user running the application is a member of the group specified by *anId*. Returns NO otherwise.

See also: **;ExtendedApp.rtf;groupMemberForString;;¬groupMemberForString:,**  
**;ExtendedApp.rtf;effectiveGroupIDNum;¬-effectiveGroupIDNum,** and  
**;ExtendedApp.rtf;groupIDNum;¬-groupIDNum.**

**groupMemberForString;;¬groupMemberForString:**

- (BOOL)**groupMemberForString:(const char \*)aString**

Returns YES if the user running the application is a member of the group specified by *aString*. Returns NO otherwise.

See also: **;ExtendedApp.rtf;groupMemberForID;¬-groupMemberForID;**  
**;ExtendedApp.rtf;effectiveGroupIDNum;¬-effectiveGroupIDNum,** and  
**;ExtendedApp.rtf;groupIDNum;¬-groupIDNum.**

**hostID;¬hostID**

- (unsigned long int)**hostID**

Returns the results of the gethostid() function. This value is unique to the hardware.

See also: **;ExtendedApp.rtf;realHostName;¬-realHostName**

**knowsDefaultValue;¬knowsDefaultValue:**

- (BOOL)**knowsDefaultValue:(const char \*)defName**

Returns YES if the defaults system has a valid value for the specified defaultName. Use this method before reading any value. This gives you the chance to use app internal defaults.

**realDomainName;¬realDomainName**

- (NXAtom)**realDomainName**

Returns the name of the domain the machine is on. See the man page for getdomainname() for more information.

See also: **;ExtendedApp.rtf;domainName;¬-domainName**

**realHostName;¬realHostName**

- (NXAtom)**realHostName**

Returns the host name as returned by the gethostname() function.

See also: **;ExtendedApp.rtf;hostID;¬-hostID**

**realNameFor;;¬realNameFor:**

- (NXAtom)**realNameFor:(NXAtom)userId**

Returns the full name for the user with login name *userId*. This method is used to find the name of the user running the application for the **;ExtendedApp.rtf;userRealName;¬±userRealName** method below.

See also: **;ExtendedApp.rtf;userLoginName;¬-userLoginName** and **;ExtendedApp.rtf;userRealName;¬-userRealName**

**userHomeDirectory;¬userHomeDirectory**

- (NXAtom)**userHomeDirectory**

Returns the home directory of the user who ran the application as found by NXHomeDirectory().

**userIDNum;¬userIDNum**

- (int)**userIDNum**

Returns the results of the getuid() function.

See also: **;ExtendedApp.rtf;effectiveGroupIDNum;¬-effectiveGroupIDNum,**  
**;ExtendedApp.rtf;effectiveUserIDNum;¬-effectiveUserIDNum,** and

**;ExtendedApp.rtf;groupIDNum;¬-groupIDNum.**

**userLoginName;¬-userLoginName**

- (NXAtom)**userLoginName**

Returns the results of the NXUserName() function, which returns the login name of the user who ran the application.

See also: **;ExtendedApp.rtf;realNameFor;;¬-realNameFor:** and **;ExtendedApp.rtf;userRealName;¬-userRealName**

**userRealName;¬-userRealName**

- (NXAtom)**userRealName**

Returns the full name of the user who ran the application.

See also: **;ExtendedApp.rtf;realNameFor;;¬-realNameFor:** and **;ExtendedApp.rtf;userLoginName;¬-userLoginName**