

ExtendedApp

Inherits From:	Application
Declared In:	daymisckit/ExtendedApp.h ExtendedApp.h (a link; it is the required location for use in the IB/PB generated main())

Class Description

The ExtendedApp 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 UNIX functions. Because some queries are made via NetInfo, to use the ExtendedApp in your project you must include the NeXT shared library **ni_s** when linking.

If you used the 2.x version of this object, note that with the advent of NeXTSTEP 3.0, the `±appDirectory` method became redundant, and has therefore become a wrapper around the appropriate `NXBundle` function. (This means±or at least implies±that it lo longer provides a trailing "/" on the directory name, too. If you wish to retain that behavior, you could do it with a simple subclass.)

Instance Variables

```
char realHostName[MAXHOSTNAMELEN];  
NXAtom userRealName;
```

`realHostName` The name of the host the application is running on.

`userRealName` The full name of the user running the program as stored in netinfo. (The 'real' name versus the 'login' name.)

Method Types

Initializing an ExtendedApp + new

Getting information ± appDirectory
 ± effectiveGroupIDNum
 ± effectiveUserIDNum
 ± groupIDNum
 ± hostID
 ± realHostName
 ± realNameFor:
 ± userHomeDirectory
 ± userIDNum
 ± userLoginName
 ± userRealName

Class Methods

new

+ **new**

Sets up the instance variables of the ExtendedApp so that subsequent call sot the instance methods will return the correct values. Returns *self*, the Application object, or a new instance if no Application object already exists.

See also: **-new** (Application class)

Instance Methods

appDirectory

- (const char *)**appDirectory**

Returns the directory in which the application's executable resides (the app wrapper).

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

effectiveGroupIDNum

- (int)**effectiveGroupIDNum**

Returns the results of the getegid() function.

See also: **-effectiveUserIDNum**, **-groupIDNum**, and **-userIDNum**.

effectiveUserIDNum

- (int)**effectiveUserIDNum**

Returns the results of the geteuid() function.

See also: **-effectiveGroupIDNum**, **-groupIDNum**, and **-userIDNum**.

groupIDNum

- (int)**groupIDNum**

Returns the results of the getgid() function.

See also: **-effectiveGroupIDNum**, **-effectiveUserIDNum**, and **-userIDNum**.

hostID

- (unsigned long int)**hostID**

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

See also: **-realHostName**

realHostName

- (NXAtom)**realHostName**

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

See also: **-hostID**

realNameFor:

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

Returns the full name, as found in NetInfo, for the user with login name *userId*. This method works correctly for both network (root domain) and local users; it searches the local domain first and then the root domain. It has not been tested on networks with more than two levels, and it is expected to fail on sub-domain users of such nets. *(Author's note: If you know of a better way to obtain this information which is more accurate, or doesn't require NetInfo, please let me know so that I can use it instead.)* This method is used to find the name of the user running the application for the **±userRealName** method below.

See also: **-userLoginName** and **-userRealName**

userHomeDirectory

- (NXAtom)**userHomeDirectory**

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

userIDNum

- (int)**userIDNum**

Returns the results of the getuid() function.

See also: **-effectiveGroupIDNum**, **-effectiveUserIDNum**, and **-groupIDNum**.

userLoginName

- (NXAtom)**userLoginName**

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

See also: **-realNameFor:** and **-userRealName**

userRealName

- (NXAtom)**userRealName**

Returns the full name of the user who ran the application, as stored in NetInfo.

See also: **-realNameFor:** and **-userLoginName**