

AppExceptions

CATEGORY OF

DECLARED IN

Application

AppExceptions.h

CLASS DESCRIPTION

AppExceptions provides for a clearing house of named NeXTSTEP exceptions. Simply link it into your app, and call it using **NXApp**.

AppExceptions is free. Use it and abuse it as you see fit.

Scott Hess
12901 Upton Avenue South, #326
Burnsville, MN 55337
(612) 895-1208
scott@gac.edu
shess@ssesco.com

CATEGORY VARIABLES

<i>Inherited from Application</i>	id	NXApp;
<i>Declared in AppExceptions</i>	HashTable *	exceptionTable;
	int	exceptionBase;
exceptionTable	Maps exception names to numbers.	
exceptionBase	Next exception that -exceptionFor: will pass out.	

METHOD TYPES

Mapping names to numbers - exceptionFor:
Raising exceptions - raiseException:
- raiseException:with:
- raiseException:with:with:

INSTANCE METHODS

exceptionFor:

-(int)**exceptionFor:**(const char *)*name*

Returns the mapping for *name*. If there is no current mapping for *name*, create such a mapping. The mappings are created from NX_APPBASE to NX_APPBASE+999.

raiseException:

- **raiseException:**(const char *)*name*

Calls -**raiseException:with:with:** with NULL for both data parameters.

raiseException:with:

- **raiseException:**(const char *)*name* **with:**(const void *)*data1*

Calls -**raiseException:with:with:** with NULL as the *data2* parameter.

raiseException:with:with:

- **raiseException:**(const char *)*name* **with:**(const void *)*data1* **with:**(const void *)*data2*

Calls on -**exceptionFor:** to get the exception number for *name* and raises the exception with *data1* as the first data parameter and *data2* as the second.

USAGE

```

NX_DURING
    /* Do some work, and end up in a situation which requires an exception. */
    [NXApp raiseException:"Cancel"];
    /* This is the stuff the exception will skip over. */
NX_HANDLER
    /* Perhaps do some recovery here, such as freeing memory or whatever. */
    if( NXLocalHandler.code==[NXApp exceptionFor:"Cancel"]) {
        /* Handle the exception. */
    } else {
        NX_RERAISE();
    }
NX_ENDHANDLER

```