Foundation Constants Reference

Cocoa > Data Management

2009-05-06

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

Foundation Constants Reference

Framework:	Foundation/Foundation.h
Declared in	FoundationErrors.h NSDecimal.h NSError.h NSException.h NSGeometry.h NSHashTable.h NSJavaSetup.h NSMapTable.h NSObjCRuntime.h NSURLError.h NSZone.h QTKitDefines.h

Overview

This document defines constants in the Foundation framework that are not associated with a particular class.

Constants

Enumerations

NSNotFound

Defines a value that indicates that an item requested couldn't be found or doesn't exist.

```
enum {
   NSNotFound = 0x7fffffff
};
```

Constants

`NSNotFound`

A value that indicates that an item requested couldn't be found or doesn't exist.

Available in Mac OS X v10.0 and later.

Declared in `NSObjCRuntime.h`.

Discussion
`NSNotFound` is typically used by various methods and functions that search for items in serial data and return indices, such as characters in a string object or `id`s in an `NSArray` object.

Declared In
`NSObjCRuntime.h`

Memory Allocation

These constants are used as components in a bitfield to specify the behavior of `NSAllocateCollectable` and `NSReallocateCollectable`.

```
enum {
    NSScannedOption = (1<<0),
    NSCollectorDisabledOption = (2<<0),
};
```

Constants

`NSScannedOption`

Specifies allocation of scanned memory.

Available in Mac OS X v10.4 and later.

Declared in `NSZone.h`.

`NSCollectorDisabledOption`

Specifies that the block is retained, and therefore ineligible for collection.

Available in Mac OS X v10.5 and later.

Declared in `NSZone.h`.

Declared In
`NSGarbageCollector.h`

NSError Codes

`NSError` codes in the Cocoa error domain.

```
enum {
    NSFileNoSuchFileError = 4,
    NSFileLockingError = 255,
    NSFileReadUnknownError = 256,
    NSFileReadNoPermissionError = 257,
    NSFileReadInvalidFileNameError = 258,
    NSFileReadCorruptFileError = 259,
    NSFileReadNoSuchFileError = 260,
    NSFileReadInapplicableStringEncodingError = 261,
    NSFileReadUnsupportedSchemeError = 262,
    NSFileWriteUnknownError = 512,
    NSFileWriteNoPermissionError = 513,
    NSFileWriteInvalidFileNameError = 514,
    NSFileWriteInapplicableStringEncodingError = 517,
    NSFileWriteUnsupportedSchemeError = 518,
    NSFileWriteOutOfSpaceError = 640,
    NSKeyValueValidationError = 1024,
    NSFormattingError = 2048,
    NSUserCancelledError = 3072,

    NSFileErrorMinimum = 0,
    NSFileErrorMaximum = 1023,
    NSValidationErrorMinimum = 1024,
    NSValidationErrorMaximum = 2047,
    NSFormattingErrorMinimum = 2048,
    NSFormattingErrorMaximum = 2559,

    NSExecutableErrorMinimum = 3584,
    NSExecutableNotLoadableError = 3584,
    NSExecutableArchitectureMismatchError = 3585,
    NSExecutableRuntimeMismatchError = 3586,
    NSExecutableLoadError = 3587,
    NSExecutableLinkError = 3588,
    NSExecutableErrorMaximum = 3839
}
```

Constants

`NSFileNoSuchFileError`

File-system operation attempted on non-existent file

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileLockingError`

Failure to get a lock on file

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadUnknownError`

Read error, reason unknown

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadNoPermissionError`

Read error because of a permission problem

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadInvalidFileNameError`
Read error because of an invalid file name

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadCorruptFileError`
Read error because of a corrupted file, bad format, or similar reason

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadNoSuchFileError`
Read error because no such file was found

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadInapplicableStringEncodingError`
Read error because the string encoding was not applicable.

Access the bad encoding from the `userInfo` dictionary using the `NSStringEncodingErrorKey` key.

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileReadUnsupportedSchemeError`
Read error because the specified URL scheme is unsupported

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileWriteUnknownError`
Write error, reason unknown

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileWriteNoPermissionError`
Write error because of a permission problem

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileWriteInvalidFileNameError`
Write error because of an invalid file name

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileWriteInapplicableStringEncodingError`
Write error because the string encoding was not applicable.

Access the bad encoding from the `userInfo` dictionary using the `NSStringEncodingErrorKey` key.

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSFileWriteUnsupportedSchemeError`
: Write error because the specified URL scheme is unsupported

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFileWriteOutOfSpaceError`
: Write error because of a lack of disk space

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSKeyValueValidationError`
: Key-value coding validation error

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFormattingError`
: Formatting error (related to display of data)

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSUserCancelledError`
: The user cancelled the operation (for example, by pressing Command-period).

: This code is for errors that do not require a dialog displayed and might be candidates for special-casing.

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFileErrorMinimum`
: Marks the start of the range of error codes reserved for file errors

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFileErrorMaximum`
: Marks the end of the range of error codes reserved for file errors

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSValidationErrorMinimum`
: Marks the start of the range of error codes reserved for validation errors.

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSValidationErrorMaximum`
: Marks the start and end of the range of error codes reserved for validation errors.

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFormattingErrorMinimum`
: Marks the start of the range of error codes reserved for formatting errors.

: Available in Mac OS X v10.4 and later.

: Declared in `FoundationErrors.h`.

`NSFormattingErrorMaximum`
Marks end of the range of error codes reserved for formatting errors.

Available in Mac OS X v10.4 and later.

Declared in `FoundationErrors.h`.

`NSExecutableErrorMinimum`
Marks beginning of the range of error codes reserved for errors related to executable files.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableNotLoadableError`
Executable is of a type that is not loadable in the current process.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableArchitectureMismatchError`
Executable does not provide an architecture compatible with the current process.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableRuntimeMismatchError`
Executable has Objective C runtime information incompatible with the current process.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableLoadError`
Executable cannot be loaded for some other reason, such as a problem with a library it depends on.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableLinkError`
Executable fails due to linking issues.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

`NSExecutableErrorMaximum`
Marks end of the range of error codes reserved for errors related to executable files.

Available in Mac OS X v10.5 and later.

Declared in `FoundationErrors.h`.

Discussion
The constants in this enumeration are `NSError` code numbers in the Cocoa error domain (`NSCocoaErrorDomain`). Other frameworks, most notably the Application Kit, provide their own `NSCocoaErrorDomain` error codes.

The enumeration constants beginning with `NSFile` indicate file-system errors or errors related to file I/O operations. Use the key `NSFilePathErrorKey` or the `NSURLErrorKey` (whichever is appropriate) to access the file-system path or URL in the `userInfo` dictionary of the `NSError` object.

Declared In
`FoundationErrors.h`

URL Loading System Error Codes

These values are returned as the error code property of an `NSError` object with the domain "NSURLErrorDomain".

```
typedef enum
{
    NSURLErrorUnknown = -1,
    NSURLErrorCancelled = -999,
    NSURLErrorBadURL = -1000,
    NSURLErrorTimedOut = -1001,
    NSURLErrorUnsupportedURL = -1002,
    NSURLErrorCannotFindHost = -1003,
    NSURLErrorCannotConnectToHost = -1004,
    NSURLErrorDataLengthExceedsMaximum = -1103,
    NSURLErrorNetworkConnectionLost = -1005,
    NSURLErrorDNSLookupFailed = -1006,
    NSURLErrorHTTPTooManyRedirects = -1007,
    NSURLErrorResourceUnavailable = -1008,
    NSURLErrorNotConnectedToInternet = -1009,
    NSURLErrorRedirectToNonExistentLocation = -1010,
    NSURLErrorBadServerResponse = -1011,
    NSURLErrorUserCancelledAuthentication = -1012,
    NSURLErrorUserAuthenticationRequired = -1013,
    NSURLErrorZeroByteResource = -1014,
    NSURLErrorFileDoesNotExist = -1100,
    NSURLErrorFileIsDirectory = -1101,
    NSURLErrorNoPermissionsToReadFile = -1102,
    NSURLErrorSecureConnectionFailed = -1200,
    NSURLErrorServerCertificateHasBadDate = -1201,
    NSURLErrorServerCertificateUntrusted = -1202,
    NSURLErrorServerCertificateHasUnknownRoot = -1203,
    NSURLErrorServerCertificateNotYetValid = -1204,
    NSURLErrorClientCertificateRejected = -1205,
    NSURLErrorCannotLoadFromNetwork = -2000,
    NSURLErrorCannotCreateFile = -3000,
    NSURLErrorCannotOpenFile = -3001,
    NSURLErrorCannotCloseFile = -3002,
    NSURLErrorCannotWriteToFile = -3003,
    NSURLErrorCannotRemoveFile = -3004,
    NSURLErrorCannotMoveFile = -3005,
    NSURLErrorDownloadDecodingFailedMidStream = -3006,
    NSURLErrorDownloadDecodingFailedToComplete = -3007
}
```

Constants

`NSURLErrorUnknown`

Returned when the URL Loading system encounters an error that it cannot interpret.

This can occur when an error originates from a lower level framework or library. Whenever this error code is received, it is a bug, and should be reported to Apple.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCancelled`

Returned when an asynchronous load is canceled.

A Web Kit framework delegate will receive this error when it performs a cancel operation on a loading resource. Note that an `NSURLConnection` or `NSURLDownload` delegate will not receive this error if the download is canceled.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorBadURL`

Returned when a URL is sufficiently malformed that a URL request cannot be initiated

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorTimedOut`

Returned when an asynchronous operation times out.

`NSURLConnection` will send this error to its delegate when the timeoutInterval in NSURLRequest expires before a load can complete.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorUnsupportedURL`

Returned when a properly formed URL cannot be handled by the framework.

The most likely cause is that there is no available protocol handler for the URL.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotFindHost`

Returned when the host name for a URL cannot be resolved.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotConnectToHost`

Returned when an attempt to connect to a host has failed.

This can occur when a host name resolves, but the host is down or may not be accepting connections on a certain port.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorDataLengthExceedsMaximum`

Returned when the length of the resource data exceeds the maximum allowed.

Available in Mac OS X v10.5 and later.

Declared in `NSURLError.h`.

`NSURLErrorNetworkConnectionLost`

Returned when a client or server connection is severed in the middle of an in-progress load.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorDNSLookupFailed`

See `NSURLErrorCannotFindHost`

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorHTTPTooManyRedirects`

Returned when a redirect loop is detected or when the threshold for number of allowable redirects has been exceeded (currently 16).

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorResourceUnavailable`

Returned when a requested resource cannot be retrieved.

Examples are “file not found”, and data decoding problems that prevent data from being processed correctly.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorNotConnectedToInternet`

Returned when a network resource was requested, but an internet connection is not established and cannot be established automatically, either through a lack of connectivity, or by the user's choice not to make a network connection automatically.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorRedirectToNonExistentLocation`

Returned when a redirect is specified by way of server response code, but the server does not accompany this code with a redirect URL.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorBadServerResponse`

Returned when the URL Loading system receives bad data from the server.

This is equivalent to the “500 Server Error” message sent by HTTP servers.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorUserCancelledAuthentication`

Returned when an asynchronous request for authentication is cancelled by the user.

This is typically incurred by clicking a “Cancel” button in a username/password dialog, rather than the user making an attempt to authenticate.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorUserAuthenticationRequired`

Returned when authentication is required to access a resource.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorZeroByteResource`
Returned when a server reports that a URL has a non-zero content length, but terminates the network connection “gracefully” without sending any data.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorFileDoesNotExist`
Returned when a file does not exist.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorFileIsDirectory`
Returned when a request for an FTP file results in the server responding that the file is not a plain file, but a directory.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorNoPermissionsToReadFile`
Returned when a resource cannot be read due to insufficient permissions.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorSecureConnectionFailed`
Returned when an attempt to establish a secure connection fails for reasons which cannot be expressed more specifically.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorServerCertificateHasBadDate`
Returned when a server certificate has a date which indicates it has expired, or is not yet valid.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorServerCertificateUntrusted`
Returned when a server certificate is signed by a root server which is not trusted.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorServerCertificateHasUnknownRoot`
Returned when a server certificate is not signed by any root server.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorServerCertificateNotYetValid`
Returned when a server certificate is not yet valid.

Available in Mac OS X v10.4 and later.

Declared in `NSURLError.h`.

`NSURLErrorClientCertificateRejected`
Returned when a server certificate is rejected.

Available in Mac OS X v10.4 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotLoadFromNetwork`
Returned when a specific request to load an item only from the cache cannot be satisfied.

This error is sent at the point when the library would go to the network accept for the fact that is has been blocked from doing so by the "load only from cache" directive.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotCreateFile`
Returned when `NSURLDownload` object was unable to create the downloaded file on disk due to a I/O failure.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotOpenFile`
Returned when `NSURLDownload` was unable to open the downloaded file on disk.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotCloseFile`
Returned when `NSURLDownload` was unable to close the downloaded file on disk.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotWriteToFile`
Returned when `NSURLDownload` was unable to write to the downloaded file on disk.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotRemoveFile`
Returned when `NSURLDownload` was unable to remove a downloaded file from disk.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorCannotMoveFile`
Returned when `NSURLDownload` was unable to move a downloaded file on disk.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorDownloadDecodingFailedMidStream`
Returned when `NSURLDownload` failed to decode an encoded file during the download.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

`NSURLErrorDownloadDecodingFailedToComplete`
Returned when `NSURLDownload` failed to decode an encoded file after downloading.

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

Availability
Available in Mac OS X v10.2 with Safari 1.0 installed.
Available in Mac OS X v10.2.7 and later.

Declared In
NSURLError.h

Global Variables

Cocoa Error Domain

This constant defines the Cocoa error domain.

```
NSString *const NSCocoaErrorDomain;
```

Constants
NSCocoaErrorDomain
Application Kit and Foundation Kit errors.
Available in Mac OS X v10.4 and later.
Declared in NSError.h.

Declared In
FoundationErrors.h

NSJavaSetup Information

Keys into a dictionary providing information about the way to set up the Java virtual machine.

```
extern NSString *NSJavaClasses;
extern NSString *NSJavaRoot;
extern NSString *NSJavaPath;
extern NSString *NSJavaUserPath;
extern NSString *NSJavaLibraryPath;
extern NSString *NSJavaOwnVirtualMachine;
extern NSString *NSJavaPathSeparator;
```

Constants
NSJavaClasses
The classes that the virtual machine should load so that their associated frameworks will be loaded.
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.
Declared in NSJavaSetup.h.

NSJavaRoot
The root of the location where the application's classes are.
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.
Declared in NSJavaSetup.h.

`NSJavaPath`
: A class path whose components will be prepended by `NSJavaRoot` if they are not absolute locations. This entry is an array.

  Available in Mac OS X v10.0 and later.

  Deprecated in Mac OS X v10.5.

  Declared in `NSJavaSetup.h`.

`NSJavaUserPath`
: Another segment of the class path so that the application developer can customize where classes will be looked for.

  This path goes after the application path so that one cannot replace the classes used by the application.

  Available in Mac OS X v10.0 and later.

  Deprecated in Mac OS X v10.5.

  Declared in `NSJavaSetup.h`.

`NSJavaLibraryPath`
: The path where the runtime should look for dynamic libraries needed by Java wrappers.

  This path is an `NSArray` object.

  Available in Mac OS X v10.0 and later.

  Deprecated in Mac OS X v10.5.

  Declared in `NSJavaSetup.h`.

`NSJavaOwnVirtualMachine`
: An `NSString` object. If this string exists in the dictionary, `NSJavaSetup` attempts to create a new Java virtual machine rather than reusing the existing one. Set the value of this string to `"YES"`.

  Available in Mac OS X v10.0 and later.

  Deprecated in Mac OS X v10.5.

  Declared in `NSJavaSetup.h`.

`NSJavaPathSeparator`
: This path is not a dictionary key—it is a value indicating the separator placed between components of a pathname passed to `NSJavaSetup`.

  Available in Mac OS X v10.0 and later.

  Deprecated in Mac OS X v10.5.

  Declared in `NSJavaSetup.h`.

Declared In
`NSJavaSetup.h`

NSHashTable Callbacks

Predefined sets of callbacks for `NSHashTable`.

```
extern const NSHashTableCallBacks NSIntegerHashCallBacks;
extern const NSHashTableCallBacks NSIntHashCallBacks;
extern const NSHashTableCallBacks NSNonOwnedPointerHashCallBacks;
extern const NSHashTableCallBacks NSNonRetainedObjectHashCallBacks;
extern const NSHashTableCallBacks NSObjectHashCallBacks;
extern const NSHashTableCallBacks NSOwnedObjectIdentityHashCallBacks;
extern const NSHashTableCallBacks NSOwnedPointerHashCallBacks;
extern const NSHashTableCallBacks NSPointerToStructHashCallBacks;
```

Constants

`NSIntegerHashCallBacks`

For sets of `NSInteger`-sized quantities or smaller (for example, `int`, `long`, or `unichar`).

Available in Mac OS X v10.5 and later.

Declared in `NSHashTable.h`.

`NSIntHashCallBacks`

For sets of pointer-sized quantities or smaller (for example, `int`, `long`, or `unichar`). (**Deprecated.** Use `NSIntegerHashCallBacks` instead.)

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSHashTable.h`.

`NSNonOwnedPointerHashCallBacks`

For sets of pointers, hashed by address.

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

`NSNonRetainedObjectHashCallBacks`

For sets of objects, but without retaining/releasing.

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

`NSObjectHashCallBacks`

For sets of objects (similar to `NSSet`).

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

`NSOwnedObjectIdentityHashCallBacks`

For sets of objects, with transfer of ownership upon insertion, using pointer equality.

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

`NSOwnedPointerHashCallBacks`

For sets of pointers, with transfer of ownership upon insertion.

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

`NSPointerToStructHashCallBacks`

For sets of pointers to structs, when the first field of the struct is `int`-sized.

Available in Mac OS X v10.0 and later.

Declared in `NSHashTable.h`.

Discussion
On Mac OS X v10.5 and later, see also the `NSHashTable` class.

Note that you can make your own callback by picking fields among the above callbacks.

Declared In
`NSHashTable.h`

NSMapTable Key Call Backs

Predefined sets of callbacks for `NSMapTable` keys.

```
extern const NSMapTableKeyCallBacks NSIntegerMapKeyCallBacks;
extern const NSMapTableKeyCallBacks NSIntMapKeyCallBacks;
extern const NSMapTableKeyCallBacks NSNonOwnedPointerMapKeyCallBacks;
extern const NSMapTableKeyCallBacks  NSNonOwnedPointerOrNullMapKeyCallBacks;
extern const NSMapTableKeyCallBacks NSNonRetainedObjectMapKeyCallBacks;
extern const NSMapTableKeyCallBacks NSObjectMapKeyCallBacks;
extern const NSMapTableKeyCallBacks NSOwnedPointerMapKeyCallBacks;
```

Constants

`NSIntegerMapKeyCallBacks`

For keys that are pointer-sized quantities or smaller (for example, `int`, `long`, or `unichar`).

Available in Mac OS X v10.5 and later.

Declared in `NSMapTable.h`.

`NSIntMapKeyCallBacks`

For keys that are pointer-sized quantities or smaller (for example, `int`, `long`, or `unichar`). (**Deprecated.** Use `NSIntegerMapKeyCallBacks` instead.)

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSMapTable.h`.

`NSNonOwnedPointerMapKeyCallBacks`

For keys that are pointers not freed.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSNonOwnedPointerOrNullMapKeyCallBacks`

For keys that are pointers not freed, or `NULL`.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSNonRetainedObjectMapKeyCallBacks`

For sets of objects, but without retaining/releasing.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSObjectMapKeyCallBacks`

For keys that are objects.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSOwnedPointerMapKeyCallBacks`
For keys that are pointers, with transfer of ownership upon insertion.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

Discussion
On Mac OS X v10.5 and later, see also the `NSMapTable` class.

Note that you can make your own callback by picking fields among the above callbacks.

Declared In
`NSMapTable.h`

NSMapTable Value Callbacks

These are predefined sets of callbacks for `NSMapTable` values.

```
extern const NSMapTableValueCallBacks NSIntegerMapValueCallBacks;
extern const NSMapTableValueCallBacks NSIntMapValueCallBacks;
extern const NSMapTableValueCallBacks NSNonOwnedPointerMapValueCallBacks;
extern const NSMapTableValueCallBacks NSObjectMapValueCallBacks;
extern const NSMapTableValueCallBacks NSNonRetainedObjectMapValueCallBacks;
extern const NSMapTableValueCallBacks NSOwnedPointerMapValueCallBacks;
```

Constants
`NSIntegerMapValueCallBacks`
For values that are pointer-sized quantities, (for example, `int`, `long`, or `unichar`).

Available in Mac OS X v10.5 and later.

Declared in `NSMapTable.h`.

`NSIntMapValueCallBacks`

For values that are pointer-sized quantities, (for example, `int`, `long`, or `unichar`). (**Deprecated.** Use `NSIntegerMapValueCallBacks` instead.)

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSMapTable.h`.

`NSNonOwnedPointerMapValueCallBacks`
For values that are not owned pointers.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSOwnedPointerMapValueCallBacks`
For values that are owned pointers.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSNonRetainedObjectMapValueCallBacks`
For sets of objects, but without retaining/releasing.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

`NSObjectMapValueCallBacks`
For values that are objects.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

Discussion
On Mac OS X v10.5 and later, see also the `NSMapTable` class.

Note that you can make your own callback by picking fields among the above callbacks.

Declared In
`NSMapTable.h`

NSURL Domain

This error domain is defined for `NSURL`.

```
extern NSString * const NSURLErrorDomain;
```

Constants
`NSURLErrorDomain`
URL loading system errors

Available in Mac OS X v10.2 and later.

Declared in `NSURLError.h`.

Declared In
`NSURLError.h`

Zero Constants

These constants are defined as conveniences and can be used to compare with return values from functions.

```
extern const NSPoint NSZeroPoint;
extern const NSSize NSZeroSize;
extern const NSRect NSZeroRect;
```

Constants
`NSZeroPoint`
An `NSPoint` structure with both x and y coordinates set to `0`.

Available in Mac OS X v10.0 and later.

Declared in `NSGeometry.h`.

`NSZeroSize`
An `NSSize` structure set to `0` in both dimensions.

Available in Mac OS X v10.0 and later.

Declared in `NSGeometry.h`.

`NSZeroRect`
An `NSRect` structure set to `0` in width and height.

Available in Mac OS X v10.0 and later.

Declared in `NSGeometry.h`.

Declared In
NSGeometry.h

Numeric Constants

NSDecimal Constants

Constants used by NSDecimal.

```
#define NSDecimalMaxSize (8)
#define NSDecimalNoScale SHRT_MAX
```

Constants
NSDecimalMaxSize
The maximum size of NSDecimal.

Gives a precision of at least 38 decimal digits, 128 binary positions.

Available in Mac OS X v10.0 and later.

Declared in NSDecimal.h.

NSDecimalNoScale
Specifies that the number of digits allowed after the decimal separator in a decimal number should not be limited.

Available in Mac OS X v10.0 and later.

Declared in NSDecimal.h.

Declared In
NSDecimal.h

NSMapTable Constants

Constants used by NSMapTable.

```
#define NSNotAnIntMapKey ((const void *)0x80000000)
#define NSNotAnIntegerMapKey    ((const void *)NSIntegerMin)
#define NSNotAPointerMapKey ((const void *)0xffffffff)
```

Constants
NSNotAnIntMapKey
Predefined notAKeyMarker for use with NSMapTableKeyCallBacks. (**Deprecated.** Use NSNotAnIntegerMapKey instead.)

Available in Mac OS X v10.0 and later.

Declared in NSMapTable.h.

NSNotAnIntegerMapKey
Predefined notAKeyMarker for use with NSMapTableKeyCallBacks.

Available in Mac OS X v10.5 and later.

Declared in NSMapTable.h.

`NSNotAPointerMapKey`
Predefined `notAKeyMarker` for use with NSMapTableKeyCallBacks.

Available in Mac OS X v10.0 and later.

Declared in `NSMapTable.h`.

Discussion
On Mac OS X v10.5 and later, see also the `NSMapTable` class.

Declared In
`NSMapTable.h`

NSInteger and NSUInteger Maximum and Minimum Values

Constants representing the maximum and minimum values of `NSInteger` and `NSUInteger`.

```
#define NSIntegerMax    LONG_MAX
#define NSIntegerMin    LONG_MIN
#define NSUIntegerMax   ULONG_MAX
```

Constants
`NSIntegerMax`
The maximum value for an `NSInteger`.

Available in Mac OS X v10.5 and later.

Declared in `QTKitDefines.h`.

`NSIntegerMin`
The minimum value for an `NSInteger`.

Available in Mac OS X v10.5 and later.

Declared in `QTKitDefines.h`.

`NSUIntegerMax`
The maximum value for an `NSUInteger`.

Available in Mac OS X v10.5 and later.

Declared in `QTKitDefines.h`.

Declared In
`NSObjCRuntime.h`

Notifications

Java Setup Notification Names

Notifications sent by the Java bridge to registered observers when a virtual machine is created and initialized.

```
extern NSString *NSJavaWillSetupVirtualMachineNotification;
extern NSString *NSJavaDidSetupVirtualMachineNotification;
extern NSString *NSJavaWillCreateVirtualMachineNotification;
extern NSString *NSJavaDidCreateVirtualMachineNotification;
```

Constants

`NSJavaWillSetupVirtualMachineNotification`

Notification sent before the Java virtual machine is set up.

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSJavaSetup.h`.

`NSJavaDidSetupVirtualMachineNotification`

Notification sent after the Java virtual machine is set up.

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSJavaSetup.h`.

`NSJavaWillCreateVirtualMachineNotification`

Notification sent before the Java virtual machine is created.

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSJavaSetup.h`.

`NSJavaDidCreateVirtualMachineNotification`

Notification sent after the Java virtual machine is created.

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSJavaSetup.h`.

Declared In

`NSJavaSetup.h`

Exceptions

General Exception Names

Exceptions defined by `NSException`.

```
extern NSString *NSGenericException;
extern NSString *NSRangeException;
extern NSString *NSInvalidArgumentException;
extern NSString *NSInternalInconsistencyException;
extern NSString *NSMallocException;
extern NSString *NSObjectInaccessibleException;
extern NSString *NSObjectNotAvailableException;
extern NSString *NSDestinationInvalidException;
extern NSString *NSPortTimeoutException;
extern NSString *NSInvalidSendPortException;
extern NSString *NSInvalidReceivePortException;
extern NSString *NSPortSendException;
extern NSString *NSPortReceiveException;
extern NSString *NSOldStyleException;
```

Constants

`NSGenericException`

A generic name for an exception.

You should typically use a more specific exception name.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSRangeException`

Name of an exception that occurs when attempting to access outside the bounds of some data, such as beyond the end of a string.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSInvalidArgumentException`

Name of an exception that occurs when you pass an invalid argument to a method, such as a `nil` pointer where a non-`nil` object is required.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSInternalInconsistencyException`

Name of an exception that occurs when an internal assertion fails and implies an unexpected condition within the called code.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSMallocException`

Obsolete; not currently used.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSObjectInaccessibleException`

Name of an exception that occurs when a remote object is accessed from a thread that should not access it.

See `NSConnection`'s `enableMultipleThreads`.

Available in Mac OS X v10.0 and later.

Declared in `NSException.h`.

`NSObjectNotAvailableException`
: Name of an exception that occurs when the remote side of the `NSConnection` refused to send the message to the object because the object has never been vended.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSDestinationInvalidException`
: Name of an exception that occurs when an internal assertion fails and implies an unexpected condition within the distributed objects.

  This is a distributed objects–specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSPortTimeoutException`
: Name of an exception that occurs when a timeout set on a port expires during a send or receive operation.

  This is a distributed objects–specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSInvalidSendPortException`
: Name of an exception that occurs when the send port of an `NSConnection` has become invalid.

  This is a distributed objects–specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSInvalidReceivePortException`
: Name of an exception that occurs when the receive port of an `NSConnection` has become invalid.

  This is a distributed objects–specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSPortSendException`
: Generic error occurred on send.

  This is an `NSPort`-specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSPortReceiveException`
: Generic error occurred on receive.

  This is an `NSPort`-specific exception.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

`NSOldStyleException`
: No longer used.

  Available in Mac OS X v10.0 and later.

  Declared in `NSException.h`.

Declared In
`NSException.h`

Version Numbers

Foundation Version Number

Version of the Foundation framework in the current environment.

```
double NSFoundationVersionNumber;
```

Constants

`NSFoundationVersionNumber`

The version of the Foundation framework in the current environment.

Available in Mac OS X v10.1 and later.

Declared in `NSObjCRuntime.h`.

Declared In

`NSObjCRuntime.h`

Foundation Framework Version Numbers

Constants to define Foundation Framework version numbers.

```
#define NSFoundationVersionNumber10_0    397.40
#define NSFoundationVersionNumber10_1    425.00
#define NSFoundationVersionNumber10_1_1    425.00
#define NSFoundationVersionNumber10_1_2    425.00
#define NSFoundationVersionNumber10_1_3    425.00
#define NSFoundationVersionNumber10_1_4    425.00
#define NSFoundationVersionNumber10_2    462.00
#define NSFoundationVersionNumber10_2_1    462.00
#define NSFoundationVersionNumber10_2_2    462.00
#define NSFoundationVersionNumber10_2_3    462.00
#define NSFoundationVersionNumber10_2_4    462.00
#define NSFoundationVersionNumber10_2_5    462.00
#define NSFoundationVersionNumber10_2_6    462.00
#define NSFoundationVersionNumber10_2_7    462.70
#define NSFoundationVersionNumber10_2_8    462.70
#define NSFoundationVersionNumber10_3    500.00
#define NSFoundationVersionNumber10_3_1    500.00
#define NSFoundationVersionNumber10_3_2    500.30
#define NSFoundationVersionNumber10_3_3    500.54
#define NSFoundationVersionNumber10_3_4    500.56
#define NSFoundationVersionNumber10_3_5    500.56
#define NSFoundationVersionNumber10_3_6    500.56
#define NSFoundationVersionNumber10_3_7    500.56
#define NSFoundationVersionNumber10_3_8    500.56
#define NSFoundationVersionNumber10_3_9    500.58
#define NSFoundationVersionNumber10_4    567.00
#define NSFoundationVersionNumber10_4_1    567.00
#define NSFoundationVersionNumber10_4_2    567.12
#define NSFoundationVersionNumber10_4_3    567.21
#define NSFoundationVersionNumber10_4_4_Intel    567.23
#define NSFoundationVersionNumber10_4_4_PowerPC    567.21
#define NSFoundationVersionNumber10_4_5    567.25
#define NSFoundationVersionNumber10_4_6    567.26
#define NSFoundationVersionNumber10_4_7    567.27
#define NSFoundationVersionNumber10_4_8    567.28
#define NSFoundationVersionNumber10_4_9    567.29
#define NSFoundationVersionNumber10_4_10    567.29
#define NSFoundationVersionNumber10_4_11    567.36
```

Constants

`NSFoundationVersionNumber10_0`

Foundation version released in Mac OS X version 10.0.

Available in Mac OS X v10.1 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_1`

Foundation version released in Mac OS X version 10.1.

Available in Mac OS X v10.2 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_1_1`

Foundation version released in Mac OS X version 10.1.1.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_1_2`
Foundation version released in Mac OS X version 10.1.2.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_1_3`
Foundation version released in Mac OS X version 10.1.3.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_1_4`
Foundation version released in Mac OS X version 10.1.4.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2`
Foundation version released in Mac OS X version 10.2.
Available in Mac OS X v10.3 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_1`
Foundation version released in Mac OS X version 10.2.1.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_2`
Foundation version released in Mac OS X version 10.2.2.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_3`
Foundation version released in Mac OS X version 10.2.3.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_4`
Foundation version released in Mac OS X version 10.2.4.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_5`
Foundation version released in Mac OS X version 10.2.5.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_6`
Foundation version released in Mac OS X version 10.2.6.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_7`
Foundation version released in Mac OS X version 10.2.7.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_2_8`
Foundation version released in Mac OS X version 10.2.8.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3`
Foundation version released in Mac OS X version 10.3.
Available in Mac OS X v10.4 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_1`
Foundation version released in Mac OS X version 10.3.1.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_2`
Foundation version released in Mac OS X version 10.3.2.
Available in Mac OS X v10.4 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_3`
Foundation version released in Mac OS X version 10.3.3.
Available in Mac OS X v10.4 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_4`
Foundation version released in Mac OS X version 10.3.4.
Available in Mac OS X v10.4 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_5`
Foundation version released in Mac OS X version 10.3.5.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_6`
Foundation version released in Mac OS X version 10.3.6.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_7`
Foundation version released in Mac OS X version 10.3.7.
Available in Mac OS X v10.5 and later.
Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_8`
: Foundation version released in Mac OS X version 10.3.8.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_3_9`
: Foundation version released in Mac OS X version 10.3.9.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4`
: Foundation version released in Mac OS X version 10.4.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_1`
: Foundation version released in Mac OS X version 10.4.1.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_2`
: Foundation version released in Mac OS X version 10.4.2.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_3`
: Foundation version released in Mac OS X version 10.4.3.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_4_Intel`
: Foundation version released in Mac OS X version 10.4.4 for Intel.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_4_PowerPC`
: Foundation version released in Mac OS X version 10.4.4 for PowerPC.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_5`
: Foundation version released in Mac OS X version 10.4.5.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_6`
: Foundation version released in Mac OS X version 10.4.6.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_7`
Foundation version released in Mac OS X version 10.4.7.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_8`
Foundation version released in Mac OS X version 10.4.8.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_9`
Foundation version released in Mac OS X version 10.4.9.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_10`
Foundation version released in Mac OS X version 10.4.10.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

`NSFoundationVersionNumber10_4_11`
Foundation version released in Mac OS X version 10.4.11.

Available in Mac OS X v10.5 and later.

Declared in `NSObjCRuntime.h`.

Declared In
`NSObjCRuntime.h`

Document Revision History

This table describes the changes to *Foundation Constants Reference*.

Date	Notes
2009-05-06	Corrected the definition of NSDecimalNoScale.
2008-11-21	Added definitions of Foundation version numbers for iPhone OS.
2008-10-15	Added definitions of NSScannedOption and NSCollectorDisabledOption.
2007-04-03	Included new constants introduced in Mac OS X v10.5.
2006-06-28	Removed duplicate constant definitions.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index

U

Z