IXNameAndFileAccess	initWithName:inFile:
	initFromName:inFile:forWriting:
	+ freeFromName:inFile:
	freeFromStore
	getName:andFile:
IXFileFinderConfiguration	setAttributeParsers:
	getAttributeParsers:
	setCrossesDeviceChanges:
	crossesDeviceChanges
	setFollowsSymbolicLinks:
	followsSymbolicLinks
	setGeneratesDescriptions:
	generatesDescriptions
	setIgnoredNames:
	ignoredNames
	setIgnoredTypes:
	ignoredTypes
	setScansForModifiedFiles:
	scansForModifiedFiles
	setUpdatesAutomatically:
	updatesAutomatically
IXFileFinderQueryAndUpdate	rootPath
	recordManager
	performQuery:atPath:forSender:
	stopQueryForSender:
	updateIndexAtPath:
	isUpdating
	suspendUpdating
	resumeUpdating
	clean
	reset
NXReference	references
	addReference
	free

initInStore:atPath:

initFromBlock:inStore:atPath:
initWithName:inFile:atPath:
initWithName:inFile:forWriting:atPath:

initFromBlock:(unsigned int)aHandle inStore:(IXStore *)aStore

Initializes a newly allocated IXFileFinder as initFromBlock:inStore:atPath: with a path argument o
is useful for opening an IXFileFinder whose root directory hasn't changed, which is usually the cas

initFromBlock:inStore:atPath:

This is the designated initializer for opening a pre-existing IXFileFinder with the IXBlockAndStor

initInStore:atPath:, initWithName:inFile:atPath:, IXBlockAndStoreAccess protocol

initFromName:(const char *)aName
inFile:(const char *)filename
forWriting:(BOOL)flag

Initializes a newly allocated IXFileFinder as initFromName:inFile:forWriting:atPath: with a path a
This method is useful for opening an IXFileFinder whose root directory hasn't changed, which is u

initFromName:inFile:forWriting:atPath:

initFromName:(const char *)aName
inFile:(const char *)filename
forWriting:(BOOL)flag
atPath:(const char *)path

Initializes a newly allocated IXFileFinder by opening it from data stored under aName in filename
The IXFileFinder's root path is reset to path it will search for files within the subtree rooted at that
an absolute or relative pathname, or it can be NULL, in which case the IXFileFinder's root path re
root path may not subsequently be changed unless the IXFileFinder is freed and then reopened. If
is opened for reading and writing, and the IXFileFinder is initialized to build and update its index i
is opened for reading only. Returns self if successful, or nil if flag is YES and filename can't be w

An IXFileFinder opened for reading only can be modified however, the changes occur only in mer
written to disk. This can be useful for keeping an index up-to-date until the application terminates,
original file.

This is the designated initializer for opening a pre-existing IXFileFinder with the IXNameAndFile
that the underlying IXStoreFile is opened by the IXFileFinder when this method is used, and that i
the IXFileFinder is freed.

initWithName:inFile:atPath:, IXNameAndFileAccess protocol

initInStore:(IXStore *)aStore

Initializes a newly allocated IXFileFinder as initInStore:atPath: with a path argument of NULL.

initInStore:atPath:

initInStore:(IXStore *)aStore atPath:(const char *)path

Initializes a newly allocated IXFileFinder in aStore, to search for files in the subtree rooted at the d
path is considered the root path for the IXFileFinder, and can be an absolute or relative pathname.
program's working directory is used. The root path may not be changed after initialization. If aSt
IXFileFinder won't attempt to maintain indexes on file attributes using IXRecordManager. This d
semantics in any way an IXFileFinder initialized without an IXStore will return the same query res
initialized with an IXStore. The presence or absence of an IXStore merely affects query performa

initWithName:inFile:atPath:

```
initWithName:(const char *)aName
        inFile:(const char *)filename
        atPath:(const char *)path
```

Initializes the IXFileFinder as a store file client named aName in the store file filename. If filenam
created. The IXFileFinder will search for files in the subtree rooted at the directory named path, w
root path for the IXFileFinder. path can be an absolute or relative pathname If path is NULL, the
directory will be used. The root path may not be changed after initialization. filename is opened f
so that indexes can be created, updated, and cleaned. Returns self.

This is the designated initializer for creating new IXFileFinders with the IXNameAndFileAccess p

initInStore:atPath:, initFromBlock:inStore:atPath:, initWithName:inFile:forWriting:atPath:, IXN
protocol