

rootPath

Getting the record manager recordManager

Performing queries performQuery:atPath:forSender:
stopQueryForSender:

Updating indexes updateIndexAtPath:forSender:
isUpdating
suspendUpdating
resumeUpdating
clean
reset

clean

Removes all inaccurate or out of date information from the file finder's index, leaving the index in an incomplete, but otherwise accurate, state. Returns self.

reset

(BOOL)isUpdating

Returns YES if the file finder is updating its index in the background. That is, if each time it performs a query, it records those files that are out of date with respect to its index, and updates them in the background.

suspendUpdating, resumeUpdating, updateIndexAtPath:forSender:, setUpdatesAutomatically:
(IXFileFinderConfiguration)

(IXPostingList *)performQuery:(const char *)aQuery
atPath:(const char *)path
forSender:sender

Performs a query within the file finder's directory for all files within the subdirectory named path relative to the working directory of the process (note: previous versions of the documentation incorrectly described path as relative to the file finder's root path). aQuery is a string defining a request in the file finder's query language (described under "The Indexing Kit Query Language" in the "Other Features" section of this chapter). If there's an index, it's checked first, and

recordManager

Returns the record manager (usually of class `IXRecordManager`) used by the file finder to maintain

reset

Removes all information from the file finder's index, but doesn't remove the indexes themselves. rebuilding a file finder's index from scratch: to do so, send this message followed by an `updateIndex` message. Returns self.

clean

resumeUpdating

Resumes automatic background updating of the file finder's index after a `suspendUpdating` message. If updating hasn't been suspended, this method has no effect. Returns self.

`suspendUpdating`, `updateIndexAtPath:forSender:`, `isUpdating`, `setUpdatesAutomatically:` (`IXFileFinderConfiguration`)

(const char *)rootPath

Returns the full pathname of file finder's primary directory—the top level directory that the indexed directory is set when the file finder is initialized, and can't be changed.

stopQueryForSender:sender

Stops background searching for the query last requested by `sender`. This is useful if `sender` doesn't want more `fileFinder:didFindFile:` messages after performing a query (for example, if it was looking for a file and found it). This message doesn't need to be sent if the `sender` passed to `performQueryAtPath:forSender:` is `fileFinder:didFindFile:`, or if the `sender` was passed as `nil`. Returns self.

`performQueryAtPath:usingIndexes:`, `fileFinder:didFindFile:` (°Methods Implemented by the Sender of Update°)

suspendUpdating

Suspends automatic background updating, if it's being done. Automatic updating may be reestablished by `resumeUpdating`. Returns self.

`resumeUpdating`, `updateIndexAtPath:forSender:`, `isUpdating`, `setUpdatesAutomatically:` (`IXFileFinderConfiguration`)

fileFinder:(IXFileFinder *)aFinder didFindFile:(IXFileRecord *)aRecord

Sent during a background search by the file finder to the sender of a performQuery:atPath:forSender: message. Returns self if the file finder finds a file matching the sender's query. Returns self.

aRecord can be used to access information about the file that matched the query, which can be displayed to the user or processed.

performQuery:atPath:forSender:, fileFinder:didFindList:

fileFinder:(IXFileFinder *)aFinder didFindList:(IXPostingList *)aList

Sent during a background search by the file finder to the sender of a performQuery:atPath:forSender: message. Returns self if the file finder finds a set of files matching the sender's query. Returns self.

aList contains IXFileRecord objects, which can be used to access information about the files that matched the query, which can be displayed to the user or processed.

performQuery:atPath:forSender:, fileFinder:didFindFile:

fileFinder:(IXFileFinder *)aFinder willAddFile:(IXFileRecord *)aRecord

Sent by the file finder to the sender of an updateIndexAtPath:forSender: message before it adds aRecord to the index. The file finder will add the record returned by this method (which should be an IXFileRecord or subclass). The sender may alter, replace, or even free aRecord, and return the record as the receiver wants it added, or nil if the sender does not want it added. If it turns out that no changes need to be made, your implementation of this method should return aRecord.

updateIndexAtPath:forSender: