
Spotlight Reference Collection

Data Management: File Management



2006-07-13



Apple Inc.
© 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, AppleWorks, Carbon, Cocoa, iPod, iTunes, Keynote, Mac, Mac OS, New York, Pages, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

Aperture, Finder, Numbers, and Spotlight are trademarks of Apple Inc.

DEC is a trademark of Digital Equipment Corporation.

Intel and Intel Core are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction **Introduction** 5

Part I **Classes** 7

Chapter 1 **NSMetadataItem Class Reference** 9

Overview 9
Adopted Protocols 9
Tasks 9
Instance Methods 10

Chapter 2 **NSMetadataQuery Class Reference** 13

Overview 13
Tasks 14
Instance Methods 15
Constants 27
Notifications 28

Chapter 3 **NSMetadataQueryAttributeValueTuple Class Reference** 31

Overview 31
Tasks 31
Instance Methods 32

Chapter 4 **NSMetadataQueryResultGroup Class Reference** 33

Overview 33
Tasks 33
Instance Methods 34

Part II **Opaque Types** 37

Chapter 5 **MDItem Reference** 39

Overview 39
Functions by Task 39
Functions 40
Data Types 44
Constants 44

Chapter 6 **MDQuery Reference 65**

- Overview 65
- Functions by Task 65
- Functions 67
- Callbacks 82
- Data Types 84
- Constants 86

Part III **Other References 91**

Chapter 7 **MDLineage Reference 93**

- Overview 93
- Functions by Task 93
- Functions 94

Chapter 8 **MDImporter Reference 97**

- Overview 97
- Callbacks 97
- Constants 98

Chapter 9 **MDSchema Reference 99**

- Overview 99
- Functions 99
- Constants 101

Spotlight Metadata Attributes 103

Document Revision History 131

Introduction

Framework	/System/Library/Frameworks/CoreServices.framework
Header file directories	/System/Library/Frameworks/CoreServices.framework/Headers
Declared in	MDImporter.h MDItem.h MDLineage.h MDQuery.h MDSchema.h NSMetadata.h

This collection of documents provides the API reference for Spotlight. Spotlight is a fast, extensible, desktop search technology that allows users to organize and search for files based on metadata.

Classes

NSMetadataItem Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Declared in	Foundation/NSMetadata.h
Availability	Available in Mac OS X v10.4 and later.
Companion guides	Spotlight Query Programming Guide Spotlight Metadata Attributes Reference
Related sample code	CoreRecipes iSpend PhotoSearch PredicateEditorSample Spotlighter

Overview

The `NSMetadataItem` class represents the metadata associated with a file, providing a simple interface to retrieve the available attribute names and values.

Adopted Protocols

NSCopying
copyWithZone:

Tasks

Getting Item Attributes

- [attributes](#) (page 10)

Returns an array containing the attribute names of the receiver's values.

- [valueForAttribute:](#) (page 10)
Returns the receiver's metadata attribute name specified by a given key.
- [valuesForAttributes:](#) (page 11)
Returns a dictionary containing the key-value pairs for the attribute names specified by a given array of keys.

Instance Methods

attributes

Returns an array containing the attribute names of the receiver's values.

```
- (NSArray *)attributes
```

Return Value

An array containing the attribute names of the receiver's values.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

valueForAttribute:

Returns the receiver's metadata attribute name specified by a given key.

```
- (id)valueForAttribute:(NSString *)key
```

Parameters

key

The name of a metadata attribute.

Return Value

The receiver's metadata attribute name specified by *key*.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

CoreRecipes

PhotoSearch

PredicateEditorSample

Spotlighter

Declared In

NSMetadata.h

valuesForAttributes:

Returns a dictionary containing the key-value pairs for the attribute names specified by a given array of keys.

- (NSDictionary *)valuesForAttributes:(NSArray *)keys

Parameters

keys

An array containing `NSString` objects that specify the names of a metadata attributes.

Return Value

A dictionary containing the key-value pairs for the attribute names specified by *keys*.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSMetadata.h`

NSMetadataQuery Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	Foundation/NSMetadata.h
Companion guide	Spotlight Query Programming Guide
Related sample code	CoreRecipes iSpend PhotoSearch Spotlighter SpotlightFortunes

Overview

The `NSMetadataQuery` class encapsulates the functionality provided by the `MDQuery` opaque type for querying the Spotlight metadata.

`NSMetadataQuery` objects provide metadata query results in several ways:

- As individual attribute values for requested attributes.
- As value lists that contain the distinct values for given attributes in the query results.
- A result array proxy, containing all the query results. This is suitable for use with Cocoa bindings.
- As a hierarchical collection of results, grouping together items with the same values for specified grouping attributes. This is also suitable for use with Cocoa bindings.

Queries have two phases: the initial gathering phase that collects all currently matching results and a second live-update phase.

By default the receiver has no limitation on its search scope. Use `setSearchScopes:` (page 23) to customize.

By default, notification of updated results occurs at 1.0 seconds. Use `setNotificationBatchingInterval:` (page 22) to customize.

You must set a predicate with the `setPredicate:` (page 23) method before starting a query.

Tasks

Creating Metadata Queries

- [init](#) (page 18)
Initializes an allocated `NSMetadataQuery` object.

Configuring Queries

- [searchScopes](#) (page 21)
Returns an array containing the receiver's search scopes.
- [setSearchScopes:](#) (page 23)
Restrict the search scope of the receiver.
- [predicate](#) (page 19)
Returns the predicate the receiver uses to filter query results.
- [setPredicate:](#) (page 23)
Sets the predicate used by the receiver to filter the query results.
- [sortDescriptors](#) (page 24)
Returns an array containing the receiver's sort descriptors.
- [setSortDescriptors:](#) (page 23)
Sets the sort descriptors to be used by the receiver.
- [valueListAttributes](#) (page 26)
Returns an array containing the value list attributes the receiver generates.
- [setValueListAttributes:](#) (page 24)
Sets the value list attributes for the receiver to the specific attribute names.
- [groupingAttributes](#) (page 17)
Returns the receiver's grouping attributes.
- [setGroupingAttributes:](#) (page 22)
Sets the receiver's grouping attributes to specific attribute names.
- [notificationBatchingInterval](#) (page 19)
Returns the interval that the receiver provides notification of updated query results.
- [setNotificationBatchingInterval:](#) (page 22)
Sets the interval between update notifications sent by the receiver.
- [delegate](#) (page 15)
Returns the receiver's delegate.
- [setDelegate:](#) (page 21)
Sets the receiver's delegate

Running Queries

- [isStarted](#) (page 18)
Returns a Boolean value that indicates whether the receiver has started the query.

- [startQuery](#) (page 25)
Attempts to start the query.
- [isGathering](#) (page 18)
Returns a Boolean value that indicates whether the receiver is in the initial gathering phase of the query.
- [isStopped](#) (page 19)
Returns a Boolean value that indicates whether the receiver has stopped the query.
- [stopQuery](#) (page 25)
Stops the receiver's current query from gathering any further results.

Getting Query Results

- [resultCount](#) (page 20)
Returns the number of results returned by the receiver.
- [resultAtIndex:](#) (page 20)
Returns the query result at a specific index.
- [results](#) (page 20)
Returns an array containing the result objects for the receiver.
- [groupedResults](#) (page 16)
Returns an array containing hierarchical groups of query results based on the receiver's grouping attributes.
- [indexOfResult:](#) (page 17)
Returns the index of a query result object in the receiver's results array.
- [valueLists](#) (page 26)
Returns a dictionary containing the value lists generated by the receiver.
- [valueOfAttribute:forResultAtIndex:](#) (page 26)
Returns the value for the attribute name *attrName* at the index in the results specified by *idx*.
- [enableUpdates](#) (page 16)
Enables updates to the query results.
- [disableUpdates](#) (page 16)
Disables updates to the query results.

Instance Methods

delegate

Returns the receiver's delegate.

```
- (id < NSMetadataQueryDelegate >)delegate
```

Return Value

The receiver's delegate, or *nil* if there is none.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDelegate:](#) (page 21)

Declared In

NSMetadata.h

disableUpdates

Disables updates to the query results.

- (void)disableUpdates

Discussion

You should invoke this method before iterating over query results that could change due to live updates.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [enableUpdates](#) (page 16)

Declared In

NSMetadata.h

enableUpdates

Enables updates to the query results.

- (void)enableUpdates

Discussion

You should invoke this method after you're done iterating over the query results.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [disableUpdates](#) (page 16)

Declared In

NSMetadata.h

groupedResults

Returns an array containing hierarchical groups of query results based on the receiver's grouping attributes.

- (NSArray *)groupedResults

Return Value

Array containing hierarchical groups of query results.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [groupingAttributes](#) (page 17)
- [setGroupingAttributes:](#) (page 22)

Declared In

NSMetadata.h

groupingAttributes

Returns the receiver's grouping attributes.

- (NSArray *)groupingAttributes

Return Value

Array containing grouping attributes.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setGroupingAttributes:](#) (page 22)

Declared In

NSMetadata.h

indexOfResult:

Returns the index of a query result object in the receiver's results array.

- (NSUInteger)indexOfResult:(id)result

Parameters

result

Query result object being inquired about.

Return Value

Index of *result* in the query result array.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [resultAtIndex:](#) (page 20)

Declared In

NSMetadata.h

init

Initializes an allocated `NSMetadataQuery` object.

- (id)init

Return Value

An initialized `NSMetadataQuery` object.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSMetadata.h`

isGathering

Returns a Boolean value that indicates whether the receiver is in the initial gathering phase of the query.

- (BOOL)isGathering

Return Value

YES when the query is in the initial gathering phase; NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isStarted](#) (page 18)
- [isStopped](#) (page 19)
- [startQuery](#) (page 25)

Declared In

`NSMetadata.h`

isStarted

Returns a Boolean value that indicates whether the receiver has started the query.

- (BOOL)isStarted

Return Value

YES when the receiver has executed the `startQuery` method; NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isGathering](#) (page 18)
- [isStopped](#) (page 19)
- [startQuery](#) (page 25)

Declared In

NSMetadata.h

isStopped

Returns a Boolean value that indicates whether the receiver has stopped the query.

- (BOOL)isStopped

Return Value

YES when the receiver has stopped the query, NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isGathering](#) (page 18)
- [isStarted](#) (page 18)
- [stopQuery](#) (page 25)

Declared In

NSMetadata.h

notificationBatchingInterval

Returns the interval that the receiver provides notification of updated query results.

- (NSTimeInterval)notificationBatchingInterval

Return Value

The interval at which notification of updated results occurs.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setNotificationBatchingInterval:](#) (page 22)

Declared In

NSMetadata.h

predicate

Returns the predicate the receiver uses to filter query results.

- (NSPredicate *)predicate

Return Value

The predicate used to filter query results.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setPredicate:](#) (page 23)

Declared In

NSMetadata.h

resultAtIndex:

Returns the query result at a specific index.

```
- (id)resultAtIndex:(NSUInteger)index
```

Parameters

index

Index of the desired result in the query result array.

Return Value

Query result at the position specified by *index*.

Discussion

For performance reasons, you should use this method when retrieving a specific result, rather than the array returned by [results](#) (page 20).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [indexOfResult:](#) (page 17)

Declared In

NSMetadata.h

resultCount

Returns the number of results returned by the receiver.

```
- (NSUInteger)resultCount
```

Return Value

The number of objects the query produced.

Discussion

For performance reasons, you should use this method, rather than invoking `count` on [results](#) (page 20).

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

results

Returns an array containing the result objects for the receiver.

- (NSArray *)results

Return Value

Proxy array containing query result objects.

Discussion

The results array is a proxy object that is primarily intended for use with Cocoa bindings. While it is possible to copy the proxy array and receive a “snapshot” of the complete current query results, it is generally not recommended due to performance and memory issues. To access individual result array elements you should instead use the [resultCount](#) (page 20) and [resultAtIndex:](#) (page 20) methods.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [groupedResults](#) (page 16)

Declared In

NSMetadata.h

searchScopes

Returns an array containing the receiver’s search scopes.

- (NSArray *)searchScopes

Return Value

An array containing the receiver’s search scopes.

Discussion

The array can contain `NSString` or `NSURL` objects that represent file system directories or the search scopes specified in “[Constants](#)” (page 27). An empty array indicates that there is no limitation on where the receiver searches.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setSearchScopes:](#) (page 23)

Declared In

NSMetadata.h

setDelegate:

Sets the receiver’s delegate

- (void)setDelegate:(id < NSMetadataQueryDelegate >)delegate

Parameters

delegate

An object to serve as the receiver’s delegate. The delegate must implement the `NSMetadataQueryDelegate` Protocol protocol. Pass `nil` to remove the current delegate.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [delegate](#) (page 15)

Related Sample Code

PhotoSearch

Declared In

NSMetadata.h

setGroupingAttributes:

Sets the receiver's grouping attributes to specific attribute names.

```
- (void)setGroupingAttributes:(NSArray *)attributes
```

Parameters

attributes

Array containing attribute names.

Discussion

Invoking this method on a receiver while it's running a query, stops the query and discards current results, and immediately starts a new query.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [groupingAttributes](#) (page 17)

Declared In

NSMetadata.h

setNotificationBatchingInterval:

Sets the interval between update notifications sent by the receiver.

```
- (void)setNotificationBatchingInterval:(NSTimeInterval)timeInterval
```

Parameters

Term

The Interval at which notification of updated results is to occur.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [notificationBatchingInterval](#) (page 19)

Declared In

NSMetadata.h

setPredicate:

Sets the predicate used by the receiver to filter the query results.

```
- (void)setPredicate:(NSPredicate *)predicate
```

Parameters

predicate

A predicate to be used to filter query results.

Discussion

Invoking this method on a receiver running a query causes the existing query to stop, all current results are discarded, and a new query is started immediately.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [predicate](#) (page 19)

Related Sample Code

iSpend

PhotoSearch

Declared In

NSMetadata.h

setSearchScopes:

Restrict the search scope of the receiver.

```
- (void)setSearchScopes:(NSArray *)scopes
```

Parameters

scopes

Array of `NSString` or `NSURL` objects that specify file system directories. You can also include the predefined search scopes specified in “[Constants](#)” (page 27). An empty array removes search scope limitations.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [searchScopes](#) (page 21)

Declared In

NSMetadata.h

setSortDescriptors:

Sets the sort descriptors to be used by the receiver.

```
- (void)setSortDescriptors:(NSArray *)descriptors
```

Parameters*descriptors*

Array of sort descriptors.

Discussion

Invoking this method on the receiver running a query causes the existing query to stop, all current results are discarded, and a new query is started immediately.

Availability

Available in Mac OS X v10.4 and later.

See Also[- `sortDescriptors`](#) (page 24)**Related Sample Code**

PhotoSearch

Declared In

NSMetadata.h

setValueListAttributes:

Sets the value list attributes for the receiver to the specific attribute names.

```
- (void)setValueListAttributes:(NSArray *)attributes
```

Parameters*attributes*

Array of value list attributes.

Discussion

The query collects the values of these attributes into unique lists that can be used to summarize the results of the query. If *attributes* is `nil`, the query generates no value lists. Note that value list collection increases CPU usage and significantly increases the memory usage of an `NSMetadataQuery` object.

Invoking this method on the receiver while it's running a query, stops the query and discards current results, and immediately starts a new query.

Availability

Available in Mac OS X v10.4 and later.

See Also[- `valueListAttributes`](#) (page 26)**Declared In**

NSMetadata.h

sortDescriptors

Returns an array containing the receiver's sort descriptors.

```
- (NSArray *)sortDescriptors
```


Return Value

An array containing sort descriptors.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setSortDescriptors](#): (page 23)

Declared In

NSMetadata.h

startQuery

Attempts to start the query.

- (BOOL)startQuery

Return Value

YES when successful; NO otherwise.

Discussion

A query can't be started if the receiver is already running a query or no predicate has been specified.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [stopQuery](#) (page 25)

- [isStarted](#) (page 18)

Related Sample Code

iSpend

PhotoSearch

Declared In

NSMetadata.h

stopQuery

Stops the receiver's current query from gathering any further results.

- (void)stopQuery

Discussion

The receiver first completes gathering any unprocessed results. If a query is stopped before the gathering phase finishes, it will not post an `NSMetadataQueryDidStartGatheringNotification` notification.

You would call this function to stop a query that is generating too many results to be useful but still want to access the available results. If the receiver is sent a `startQuery` message after performing this method, the existing results are discarded.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [startQuery](#) (page 25)
- [isStopped](#) (page 19)

Declared In

NSMetadata.h

valueListAttributes

Returns an array containing the value list attributes the receiver generates.

- (NSArray *)valueListAttributes

Return Value

Array containing value list attributes.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setValueListAttributes:](#) (page 24)

Declared In

NSMetadata.h

valueLists

Returns a dictionary containing the value lists generated by the receiver.

- (NSDictionary *)valueLists

Return Value

Dictionary of `NSMetadataQueryAttributeValueTuple` objects.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

valueOfAttribute:forResultAtIndex:

Returns the value for the attribute name `attrName` at the index in the results specified by `idx`.

- (id)valueOfAttribute:(NSString *)attributeName forResultAtIndex:(NSUInteger)index

Parameters*attributeName*

The attribute of the result object at *index* being inquired about. The attribute must be specified in [setValueListAttributes:](#) (page 24), as a sorting key in a specified sort descriptor, or as one of the grouping attributes specified set for the query.

index

Index of the desired return object in the query results array.

Return Value

Value for *attributeName* in the result object at *index* in the query result array.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

Constants

Metadata Query Search Scopes

Constants for the predefined search scopes used by [setSearchScopes:](#) (page 23).

```
NSString * const NSMetadataQueryUserHomeScope;
NSString * const NSMetadataQueryLocalComputerScope;
NSString * const NSMetadataQueryNetworkScope;
```

Constants

NSMetadataQueryUserHomeScope

Search the user's home directory.

Available in Mac OS X v10.4 and later.

Declared in NSMetadata.h.

NSMetadataQueryLocalComputerScope

Search all local mounted volumes, including the user home directory. The user's home directory is searched even if it is a remote volume.

Available in Mac OS X v10.4 and later.

Declared in NSMetadata.h.

NSMetadataQueryNetworkScope

Search all user-mounted remote volumes.

Available in Mac OS X v10.4 and later.

Declared in NSMetadata.h.

Content Relevance

In addition to the requested metadata attributes, a query result also includes content relevance, accessed with the following key.

```
NSString * const NSMetadataQueryResultContentRelevanceAttribute;
```

Constants

`NSMetadataQueryResultContentRelevanceAttribute`

Key used to retrieve an `NSNumber` object with a floating point value between 0.0 and 1.0 inclusive. The relevance value indicates the relevance of the content of a result object. The relevance is computed based on the value of the result itself, not on its relevance to the other results returned by the query. If the value is not computed, it is treated as an attribute on the item that does not exist.

Available in Mac OS X v10.4 and later.

Declared in `NSMetadata.h`.

Notifications

NSMetadataQueryDidFinishGatheringNotification

Posted when the receiver has finished with the initial result-gathering phase of the query.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSMetadata.h`

NSMetadataQueryDidStartGatheringNotification

Posted when the receiver begins with the initial result-gathering phase of the query.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSMetadata.h`

NSMetadataQueryDidUpdateNotification

Posted when the receiver's results have changed during the live-update phase of the query.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSMetadata.h`

NSMetadataQueryGatheringProgressNotification

Posted as the receiver's is collecting results during the initial result-gathering phase of the query.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

NSMetadataQueryAttributeValueTuple Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Declared in	Foundation/NSMetadata.h
Availability	Available in Mac OS X v10.4 and later.
Companion guide	Spotlight Metadata Attributes Reference

Overview

The `NSMetadataQueryAttributeValueTuple` class represents attribute-value tuples, which are objects that contain the attribute name and value of a metadata attribute.

Attribute-value tuples are returned by `NSMetadataQuery` objects as the results in the value lists. Each attribute/value tuple contains the attribute name, the value, and the number of instances of that value that exist for the attribute name.

Tasks

Getting Query Attribute/Value Information

- [attribute](#) (page 32)
Returns the receiver's attribute name.
- [count](#) (page 32)
Returns the number of instances of the value that exist for the attribute name of the receiver.
- [value](#) (page 32)
Returns the receiver's attribute value.

Instance Methods

attribute

Returns the receiver's attribute name.

- (NSString *)attribute

Return Value

The receiver's attribute name.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

count

Returns the number of instances of the value that exist for the attribute name of the receiver.

- (NSUInteger)count

Return Value

The number of instances of the value that exist for the attribute name of the receiver.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

value

Returns the receiver's attribute value.

- (id)value

Return Value

The receiver's attribute value.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

NSMetadataQueryResultGroup Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Declared in	Foundation/NSMetadata.h
Availability	Available in Mac OS X v10.4 and later.
Companion guide	Spotlight Query Programming Guide

Overview

The `NSMetadataQueryResultGroup` class represents a collection of grouped attribute results returned by an `NSMetadataQuery` object.

Tasks

Getting Query Results

- [attribute](#) (page 34)
Returns the attribute name for the receiver's result group.
- [value](#) (page 35)
Returns the value of the attribute name for the receiver.
- [results](#) (page 35)
Returns an array containing the result objects for the receiver.
- [resultCount](#) (page 34)
Returns the number of results returned by the receiver.
- [resultAtIndex:](#) (page 34)
Returns the query result at a specific index.
- [subgroups](#) (page 35)
Returns an array containing the subgroups of the receiver.

Instance Methods

attribute

Returns the attribute name for the receiver's result group.

```
- (NSString *)attribute
```

Return Value

The attribute name for the receiver's result group.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

resultAtIndex:

Returns the query result at a specific index.

```
- (id)resultAtIndex:(NSUInteger)index
```

Parameters

index

The index of the desired result.

Return Value

The query result at a specific index.

Discussion

For performance reasons, you should use this method when retrieving a specific result, rather than the array returned by [results](#) (page 35).

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

resultCount

Returns the number of results returned by the receiver.

```
- (NSUInteger)resultCount
```

Return Value

The number of results returned by the receiver.

Discussion

For performance reasons, you should use this method, rather than invoking `count` on [results](#) (page 35).

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

results

Returns an array containing the result objects for the receiver.

- (NSArray *)results

Return Value

An array containing the result objects for the receiver.

Discussion

The results array is a proxy object that is primarily intended for use with Cocoa bindings. While it is possible to copy the proxy array to get a “snapshot” of the complete current query results, it is generally not recommended due to performance and memory issues. To access individual result array elements you should instead use the `resultCount` and `resultAtIndex:` methods.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [resultCount](#) (page 34)
- [resultAtIndex:](#) (page 34)

Declared In

NSMetadata.h

subgroups

Returns an array containing the subgroups of the receiver.

- (NSArray *)subgroups

Return Value

An array containing the subgroups of the receiver.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

value

Returns the value of the attribute name for the receiver.

- (id)value

Return Value

The value of the attribute name for the receiver.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMetadata.h

Opaque Types

MDItem Reference

Derived From:	CType
Framework:	CoreServices/CoreServices.h
Declared in	MDItem.h
Companion guides	Spotlight Overview Spotlight Query Programming Guide Spotlight Importer Programming Guide Spotlight Metadata Attributes Reference

Overview

MDItem is a CF-compliant object that represents a file and the metadata associated with the file.

For functions that expect an MDItemRef parameter, if this parameter is not a valid MDItemRef, the behavior is undefined. `NULL` is not a valid MDItemRef.

Functions by Task

Creating an MDItem

[MDItemCreate](#) (page 42)

Creates an MDItem object for a file at the specified path.

[MDItemCreateWithURL](#) (page 43)

Creates an MDItem object for a file at the specified file URL.

Getting the Type Identifier

[MDItemGetTypeID](#) (page 43)

Returns the type identifier of all MDItem instances.

Retrieving Metadata Attributes

[MDItemCopyAttribute](#) (page 40)

Returns the value of the specified attribute in the metadata item.

[MDItemCopyAttributes](#) (page 41)

Returns the values of the specified attributes in the metadata item.

[MDItemCopyAttributeList](#) (page 40)

Returns the values of the specified attributes in the metadata item.

[MDItemCopyAttributeNames](#) (page 41)

Returns an array containing the attribute names existing in the metadata item.

Functions

MDItemCopyAttribute

Returns the value of the specified attribute in the metadata item.

```
CTypeRef MDItemCopyAttribute (
    MDItemRef item,
    CFStringRef name
);
```

Parameters

item

The item to be queried.

name

The name of the requested attribute.

Return Value

A `CTypeRef`, or `NULL` if there was a failure reading the attribute or the attribute does not exist.

Availability

Available in Mac OS X version 10.4 and later.

Related Sample Code

SpotlightAPI

UnsharpMask

Declared In

MDItem.h

MDItemCopyAttributeList

Returns the values of the specified attributes in the metadata item.

```
CFDictionaryRef MDItemCopyAttributeList (
    MDItemRef item,
    ...
);
```

Parameters

item

The item to be queried.

...

A comma-separated varargs list of the string attribute names..

Return Value

A CFDictionary containing keys for the requested attribute names, and the corresponding values. If an attribute does not exist, or the attribute is unreadable, there will be no key-value pair for it in the dictionary. Returns NULL on failure.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

MDItem.h

MDItemCopyAttributeNames

Returns an array containing the attribute names existing in the metadata item.

```
CFArrayRef MDItemCopyAttributeNames (
    MDItemRef item
);
```

Parameters

item

The item to be queried.

Return Value

A CFArray of CFString attribute names, or NULL on failure.

Availability

Available in Mac OS X version 10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

MDItem.h

MDItemCopyAttributes

Returns the values of the specified attributes in the metadata item.

```
CFDictionaryRef MDItemCopyAttributes (
    MDItemRef item,
    CFArrayRef names
);
```

Parameters

item

The item to be queried.

names

A CFArray containing the names of the requested attributes.

Return Value

A `CFDictionary` containing keys for the requested attribute names, and the corresponding values. If an attribute does not exist, or the attribute is unreadable, there will be no key-value pair for it in the dictionary. Returns `NULL` on failure.

Availability

Available in Mac OS X version 10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

`MDItem.h`

MDItemCreate

Creates an `MDItem` object for a file at the specified path.

```
MDItemRef MDItemCreate (
    CFAllocatorRef allocator,
    CFStringRef path
);
```

Parameters

allocator

The `CFAllocator` object to be used to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

path

A path to the file from which to create the `MDItem`. The path must exist.

Return Value

An `MDItem` object or `NULL` if there was a problem creating the object.

Discussion

Returns a metadata item for the given URL.

Special Considerations

In Mac OS X v 10.5 and later `MDItemRefs` may or may not be unique. You should always use `CFEqual` for comparison.

Prior to Mac OS X v 10.5 items were guaranteed to be unique and `==` could or `CFEqual` could be used for the comparison.

Availability

Available in Mac OS X version 10.4 and later.

Related Sample Code

SpotlightAPI

UnsharpMask

Declared In

`MDItem.h`

MDItemCreateWithURL

Creates an MDItem object for a file at the specified file URL.

```
MDItemRef MDItemCreateWithURL (
    CFAllocatorRef allocator,
    CFURLRef url
);
```

Parameters

allocator

The `CFAllocator` object to be used to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

url

A file URL to the file from which to create the MDItem. The file must exist.

Return Value

An MDItem object or `NULL` if there was a problem creating the object.

Discussion

Returns a metadata item for the given URL.

Special Considerations

In Mac OS X v 10.5 and later MDItemRefs may or may not be unique. You should always use `CFEqual` for comparison.

Prior to Mac OS X v 10.5 items were guaranteed to be unique and `==` could or `CFEqual` could be used for the comparison.

Availability

Available in Mac OS X version 10.6 and later.

Declared In

MDItem.h

MDItemGetTypeID

Returns the type identifier of all MDItem instances.

```
CTypeID MDItemGetTypeID (
    void
);
```

Return Value

The type identifier for the MDItem opaque type.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

MDItem.h

Data Types

MDItemRef

A reference to a MDItem object.

```
typedef struct __MDItem *MDItemRef;
```

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDItem.h

Constants

Common Metadata Attribute Keys

Metadata attribute keys that are common to many file types.

```

const CFStringRef kMDItemAttributeChangeDate;
const CFStringRef kMDItemAudiences;
const CFStringRef kMDItemAuthors;
const CFStringRef kMDItemAuthorAddresses;
const CFStringRef kMDItemCity;
const CFStringRef kMDItemComment;
const CFStringRef kMDItemContactKeywords;
const CFStringRef kMDItemContentCreationDate;
const CFStringRef kMDItemContentModificationDate;
const CFStringRef kMDItemContentType;
const CFStringRef kMDItemContributors;
const CFStringRef kMDItemCopyright;
const CFStringRef kMDItemCountry;
const CFStringRef kMDItemCoverage;
const CFStringRef kMDItemCreator;
const CFStringRef kMDItemDescription;
const CFStringRef kMDItemDueDate;
const CFStringRef kMDItemDurationSeconds;
const CFStringRef kMDItemEmailAddresses;
const CFStringRef kMDItemEncodingApplications;
const CFStringRef kMDItemFinderComment;
const CFStringRef kMDItemFonts;
const CFStringRef kMDItemHeadline;
const CFStringRef kMDItemIdentifier;
const CFStringRef kMDItemInstantMessageAddresses;
const CFStringRef kMDItemInstructions;
const CFStringRef kMDItemKeywords;
const CFStringRef kMDItemKind;
const CFStringRef kMDItemLanguages;
const CFStringRef kMDItemLastUsedDate;
const CFStringRef kMDItemNumberOfPages;
const CFStringRef kMDItemOrganizations;
const CFStringRef kMDItemPageHeight;
const CFStringRef kMDItemPageWidth;
const CFStringRef kMDItemParticipants;
const CFStringRef kMDItemPhoneNumbers;
const CFStringRef kMDItemProjects;
const CFStringRef kMDItemPublishers;
const CFStringRef kMDItemRecipients;
const CFStringRef kMDItemRecipientAddresses;
const CFStringRef kMDItemRights;
const CFStringRef kMDItemSecurityMethod;
const CFStringRef kMDItemStarRating;
const CFStringRef kMDItemStateOrProvince;
const CFStringRef kMDItemTextContent;
const CFStringRef kMDItemTitle;
const CFStringRef kMDItemVersion;
const CFStringRef kMDItemWhereFroms;
const CFStringRef kMDItemSupportFileType;
const CFStringRef kMDItemAuthorEmailAddresses;
const CFStringRef kMDItemRecipientEmailAddresses;
const CFStringRef kMDItemTheme;
const CFStringRef kMDItemSubject;
const CFStringRef kMDItemCFBundleIdentifier;
const CFStringRef kMDItemFSHasCustomIcon;
const CFStringRef kMDItemFSIsStationery;
const CFStringRef kMDItemInformation;
const CFStringRef kMDItemURL;

```

Constants`kMDItemAttributeChangeDate`

The date and time of the last change made to a metadata attribute. A `CFDate`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAudiences`

The audience for which the file is intended. The audience may be determined by the creator or the publisher or by a third party. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAuthors`

The author, or authors, of the contents of the file. A `CFArray` of `CFStrings`.

The order of the authors is preserved, but does not represent the main author or relative importance of the authors.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAuthorAddresses`

This attribute indicates the author addresses of the document. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.6 and later.

Declared in `MDItem.h`.

`kMDItemCity`

Identifies city of origin according to guidelines established by the provider. A `CFString`.

For example, "New York", "Cupertino", or "Toronto".

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemComment`

A comment related to the file. This differs from the Finder comment, `kMDItemFinderComment`. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemContactKeywords`

A list of contacts that are associated with this document, not including the authors. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemContentCreationDate`

The date that the contents of the file were created. A `CFDate`.

This is different than the file creation date. Its can be used to store when the file contents were first created, or first modified.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemContentModificationDate`

The date and time that the contents of the file were last modified. A `CFDate`.

This is not necessarily the file modification date.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemContentType`

The UTI pedigree of a file. A `CFString`.

For example, a jpeg image file will have a value of `public.jpeg/public.image/public.data`. The value of this attribute is set by the `MDImporter`. Changes to this value are lost when the file attributes are next imported.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemContributors`

The entities responsible for making contributions to the content of the resource. A `CFArray` of `CFStrings`.

Examples of a contributor include a person, an organization or a service.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCopyright`

The copyright owner of the file contents. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCountry`

The full, publishable name of the country or primary location where the intellectual property of the item was created, according to guidelines of the provider. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCoverage`

The extent or scope of the content of the resource. A `CFString`.

Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity).

Recommended best practice is to select a value from a controlled vocabulary, and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of co-ordinates or date ranges.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCreator`

Application used to create the document content (e.g. "Word", "AppleWorks", etc.). A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemDescription`

A description of the content of the resource. The description may include an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemDueDate`

Date this item is due. A `CFDate`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemDurationSeconds`

The duration, in seconds, of the content of file. A value of 10.5 represents media that is 10 and 1/2 seconds long. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemEmailAddresses`

Email addresses related to this item. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemEncodingApplications`

Application used to convert the original content into its current form. For example, a PDF file might have an encoding application set to "Distiller". A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFinderComment`

Finder comments for this file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFonts`

Fonts used in this item. You should store the font's full name, the postscript name, or the font family name, based on the available information. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemHeadline`

A publishable entry providing a synopsis of the contents of the file. For example, "Apple Introduces the iPod Photo". A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemIdentifier`

A formal identifier used to reference the resource within a given context. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemInstantMessageAddresses`

Instant message addresses related to this item. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemInstructions`

Editorial instructions concerning the use of the item, such as embargoes and warnings. For example, "Second of four stories". A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemKeywords`

Keywords associated with this file. For example, "Birthday", "Important", etc. An NSArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemKind`

A description of the kind of item this file represents. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemLanguages`

Indicates the languages of the intellectual content of the resource. Recommended best practice for the values of the Language element is defined by RFC 3066. A NSArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemLastUsedDate`

The date and time that the file was last used. This value is updated automatically by LaunchServices everytime a file is opened by double clicking, or by asking LaunchServices to open a file. A CFDate.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemNumberOfPages`

Number of pages in the document. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemOrganizations`

The company or organization that created the document. A NSArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPageHeight`

Height of the document page, in points (72 points per inch). For PDF files this indicates the height of the first page only. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPageWidth`

Width of the document page, in points (72 points per inch). For PDF files this indicates the width of the first page only. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemParticipants`

The list of people who are visible in an image or movie or written about in a document. A CFArray of CFStrings.

Available in Mac OS X v10.6 and later.

Declared in `MDItem.h`.

`kMDItemPhoneNumbers`

Phone numbers related to this item. A CFArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemProjects`

The list of projects that this file is part of. For example, if you were working on a movie all of the files could be marked as belonging to the project "My Movie". A CFArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPublishers`

The entity responsible for making the resource available. For example, a person, an organization, or a service. Typically, the name of a publisher should be used to indicate the entity. A CFArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemRecipients`

Recipients of this item. A CFArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemRecipientAddresses`

This attribute indicates the recipient addresses of the document. A CFArray of CFStrings.

Available in Mac OS X v10.6 and later.

Declared in `MDItem.h`.

`kMDItemRights`

Provides a link to information about rights held in and over the resource. A CFString.

Contains a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights.

If this attribute is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemSecurityMethod`

The security or encryption method used for the file. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemStarRating`

User rating of this item. For example, the stars rating of an iTunes track. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemStateOrProvince`

Identifies the province or state of origin according to guidelines established by the provider. For example, "CA", "Ontario", or "Sussex". A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemTextContent`

Contains a text representation of the content of the document. Data in multiple fields should be combined using a whitespace character as a separator. A `CFString`.

An application's Spotlight importer provides the content of this attribute. Applications can search for values in this attribute, but are not able to read the content of this attribute directly.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemTitle`

The title of the file. For example, this could be the title of a document, the name of a song, or the subject of an email message. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemVersion`

The version number of this file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemWhereFroms`

Describes where the file was obtained from. A `CFArray` of `CFStrings`.

For example, a downloaded file may refer to the URL, files received by email may indicate the sender's email address, message subject, etc.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemSupportFileType`

A `CFArray` of `CFStrings`.

Available in Mac OS X v10.5 and later.

Deprecated in Mac OS X v10.5.

Declared in `MDItem.h`.

`kMDItemAuthorEmailAddresses`

This attribute indicates the author of the emails message addresses. (This is always the email address, and not the human readable version). A `CFArray` of `CFStrings`.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemRecipientEmailAddresses`

This attribute indicates the recipients email addresses. (This is always the email address, and not the human readable version). A CFArray of CFStrings.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemTheme`

Theme of the this item. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemSubject`

Subject of the this item. Type is a CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCFBundleIdentifier`

If this item is a bundle, then this is the `CFBundleIdentifier`. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemFSHasCustomIcon`

Boolean indicating if this file has a custom icon. Type is a CFBoolean.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemFSIsStationery`

Boolean indicating if this file is stationery. Type is a CFBoolean.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemInformation`

Information about the item. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemURL`

Url of the item. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

Image Metadata Attribute Keys

Metadata attribute keys that are common to image files.

```

const CFStringRef      kMDItemPixelHeight;
const CFStringRef      kMDItemPixelWidth;
const CFStringRef      kMDItemPixelCount;
const CFStringRef      kMDItemColorSpace;
const CFStringRef      kMDItemBitsPerSample;
const CFStringRef      kMDItemFlashOnOff;
const CFStringRef      kMDItemFocalLength;
const CFStringRef      kMDItemAcquisitionMake;
const CFStringRef      kMDItemAcquisitionModel;
const CFStringRef      kMDItemISOSpeed;
const CFStringRef      kMDItemOrientation;
const CFStringRef      kMDItemLayerNames;
const CFStringRef      kMDItemWhiteBalance;
const CFStringRef      kMDItemAperture;
const CFStringRef      kMDItemProfileName;
const CFStringRef      kMDItemResolutionWidthDPI;
const CFStringRef      kMDItemResolutionHeightDPI;
const CFStringRef      kMDItemExposureMode;
const CFStringRef      kMDItemExposureTimeSeconds;
const CFStringRef      kMDItemEXIFVersion;
const CFStringRef      kMDItemAlbum;
const CFStringRef      kMDItemHasAlphaChannel;
const CFStringRef      kMDItemRedEyeOnOff;
const CFStringRef      kMDItemMeteringMode;
const CFStringRef      kMDItemMaxAperture;
const CFStringRef      kMDItemFNumber;
const CFStringRef      kMDItemExposureProgram;
const CFStringRef      kMDItemExposureTimeString;
const CFStringRef      kMDItemEXIFGPSVersion;
const CFStringRef      kMDItemAltitude;
const CFStringRef      kMDItemLatitude;
const CFStringRef      kMDItemLongitude;
const CFStringRef      kMDItemTimestamp;
const CFStringRef      kMDItemSpeed;
const CFStringRef      kMDItemGPSTrack;
const CFStringRef      kMDItemImageDirection;
const CFStringRef      kMDItemNamedLocation;

```

Constants

`kMDItemPixelHeight`

The height, in pixels, of the contents. For example, the image height or the video frame height. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPixelWidth`

The width, in pixels, of the contents. For example, the image width or the video frame width. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPixelCount`

The total number of pixels in the contents. Same as [kMDItemPixelWidth](#) (page 53) x [kMDItemPixelHeight](#) (page 53). A `CFNumber`.

Available in Mac OS X v10.6 and later.

Declared in `MDItem.h`.

`kMDItemColorSpace`

The color space model used by the document contents. For example, “RGB”, “CMYK”, “YUV”, or “YCbCr”. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemBitsPerSample`

The number of bits per sample. For example, the bit depth of an image (8-bit, 16-bit etc...) or the bit depth per audio sample of uncompressed audio data (8, 16, 24, 32, 64, etc..). A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFlashOnOff`

Indicates if a camera flash was used. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFocalLength`

The actual focal length of the lens, in millimeters. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAcquisitionMake`

The manufacturer of the device used to acquire the document contents. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAcquisitionModel`

The model of the device used to acquire the document contents. For example, 100, 200, 400, etc. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemISOSpeed`

The ISO speed used to acquire the document contents. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemOrientation`

The orientation of the document contents. Possible values are 0 (landscape) and 1 (portrait). A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemLayerNames`

The names of the layers in the file. A CFArray of CFStrings.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemWhiteBalance`

The white balance setting used to acquire the document contents. Possible values are 0 (auto white balance) and 1 (manual). A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAperture`

The aperture setting used to acquire the document contents. This unit is the APEX value. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemProfileName`

The name of the color profile used by the document contents. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemResolutionWidthDPI`

Resolution width, in DPI, of this image. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemResolutionHeightDPI`

Resolution height, in DPI, of this image. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemExposureMode`

The exposure mode used to acquire the document contents. A CFNumber.

Possible values are 0 (auto exposure), 1 (manual exposure) and 2 (auto bracket).

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemExposureTimeSeconds`

The exposure time, in seconds, used to acquire the document contents. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemEXIFVersion`

The version of the EXIF header used to generate the metadata. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAlbum`

The title for a collection of media. This is analagous to a record album, or photo album. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemHasAlphaChannel`

Indicates if this image file has an alpha channel. A CFBoolean.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemRedEyeOnOff`

Indicates if red-eye reduction was used to take the picture. A `CFBoolean`.

Possible values are 0 (no red-eye reduction mode or unknown) and 1 (red-eye reduction used).

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMeteringMode`

The metering mode used to take the image. A `CFString`.

Possible values are: `Unknown`, `Average`, `CenterWeightedAverage`, `Spot`, `MultiSpot`, `Pattern`, and `Partial`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMaxAperture`

The smallest f-number of the lens. Ordinarily it is given in the range of 00.00 to 99.99. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFNumber`

The diameter of the diaphragm aperture in terms of the effective focal length of the lens.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemExposureProgram`

The class of the exposure program used by the camera to set exposure when the image is taken.

Possible values include: `Manual`, `Normal`, and `Aperture priority`. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemExposureTimeString`

The time of the exposure. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemEXIFGPSVersion`

The version of `GPSInfoFD` in EXIF used to generate the metadata. A `CFString`.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemAltitude`

The altitude of the item in meters above sea level, expressed using the WGS84 datum. Negative values lie below sea level. A `CFString`.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemLatitude`

The latitude of the item in degrees north of the equator, expressed using the WGS84 datum. Negative values lie south of the equator. A `CFString`.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemLongitude`

The longitude of the item in degrees east of the prime meridian, expressed using the WGS84 datum. Negative values lie west of the prime meridian. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemTimestamp`

The timestamp on the item. This generally is used to indicate the time at which the event captured by the item took place. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemSpeed`

The speed of the item, in kilometers per hour. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemGPSTrack`

The direction of travel of the item, in degrees from true north. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemImageDirection`

The direction of the item's image, in degrees from true north. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemNamedLocation`

The name of the location or point of interest associated with the item. The name may be user provided. A CFString.

Available in Mac OS X v10.6 and later.

Declared in `MDItem.h`.

Video Metadata Attribute Keys

Metadata attribute keys that are common to video files.

```

const CFStringRef kMDItemAudioBitRate;
const CFStringRef kMDItemCodecs;
const CFStringRef kMDItemDeliveryType;
const CFStringRef kMDItemMediaTypes;
const CFStringRef kMDItemStreamable;
const CFStringRef kMDItemTotalBitRate;
const CFStringRef kMDItemVideoBitRate;
const CFStringRef kMDItemDirector;
const CFStringRef kMDItemProducer;
const CFStringRef kMDItemGenre;
const CFStringRef kMDItemPerformers;
const CFStringRef kMDItemOriginalFormat;
const CFStringRef kMDItemOriginalSource;

```

Constants

`kMDItemAudioBitRate`

The audio bit rate. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemCodecs`

The codecs used to encode/decode the media. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemDeliveryType`

The delivery type. Values are “Fast start” or “RTSP”. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMediaTypes`

The media types present in the content. A `CFArray` of `CFStrings`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemStreamable`

Whether the content is prepared for streaming. A `CFBoolean`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemTotalBitRate`

The total bit rate, audio and video combined, of the media. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemVideoBitRate`

The video bit rate. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemDirector`

Directory of the movie. A `CFString`.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemProducer`

Producer of the content. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemGenre`

Genre of the movie. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemPerformers`

Performers in the movie. A CFArray of CFStrings.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemOriginalFormat`

Original format of the movie. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

`kMDItemOriginalSource`

Original source of the movie. A CFString.

Available in Mac OS X v10.5 and later.

Declared in `MDItem.h`.

Audio Metadata Attribute Keys

Metadata attribute keys that describe an audio file.

```

const CFStringRef kMDItemAppleLoopDescriptors;
const CFStringRef kMDItemAppleLoopsKeyFilterType;
const CFStringRef kMDItemAppleLoopsLoopMode;
const CFStringRef kMDItemAppleLoopsRootKey;
const CFStringRef kMDItemAudioChannelCount;
const CFStringRef kMDItemAudioEncodingApplication;
const CFStringRef kMDItemAudioSampleRate;
const CFStringRef kMDItemAudioTrackNumber;
const CFStringRef kMDItemComposer;
const CFStringRef kMDItemIsGeneralMIDISequence;
const CFStringRef kMDItemKeySignature;
const CFStringRef kMDItemLyricist;
const CFStringRef kMDItemMusicalGenre;
const CFStringRef kMDItemMusicalInstrumentCategory;
const CFStringRef kMDItemMusicalInstrumentName;
const CFStringRef kMDItemRecordingDate;
const CFStringRef kMDItemRecordingYear;
const CFStringRef kMDItemTempo;
const CFStringRef kMDItemTimeSignature;

```

Constants

`kMDItemAppleLoopDescriptors`

Specifies multiple pieces of descriptive information about a loop. A CFArray of CFStrings.

Besides genre and instrument, files can contain descriptive information that help users in refining searches.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAppleLoopsKeyFilterType`

Specifies key filtering information about a loop. Loops are matched against projects that often in a major or minor key. A CFString.

To assist users in identifying loops that will "fit" with their compositions, loops can be tagged with one of the following key filters: "AnyKey" "Minor" "Major" "NeitherKey" "BothKeys". "AnyKey" means that it fits with anything (whether in a major key, minor key or neither). "Minor" fits with compositions in a minor key. "NeitherKey" doesn't work well with compositions that are in major or minor key. "BothKeys" means it fits with major or minor key.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAppleLoopsLoopMode`

Specifies how a file should be played. A CFString.

Tagged files can either be loops or non-loops (e.g., a cymbal crash). "Looping" indicates if the file should be treated as a loop. "Non-looping" indicates the file should not be treated as a loop.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAppleLoopsRootKey`

Specifies the loop's original key. The key is the root note or tonic for the loop, and does not include the scale type. A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAudioChannelCount`

Number of channels in the audio data contained in the file. A `CFNumber`.

This integer value only represents the number of discreet channels of audio data found in the file. It does not indicate any configuration of the data in regards to a user's speaker setup.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAudioEncodingApplication`

The name of the application that encoded the data contained in the audio file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAudioSampleRate`

Sample rate of the audio data contained in the file. The sample rate is a float value representing hz (audio_frames/second). For example: 44100.0, 22254.54. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemAudioTrackNumber`

The track number of a song or composition when it is part of an album. A `CFNumber` (integer).

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemComposer`

The composer of the music contained in the audio file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemIsGeneralMIDISequence`

Indicates whether the MIDI sequence contained in the file is setup for use with a General MIDI device. A `CFBoolean`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemKeySignature`

The key of the music contained in the audio file. For example: C, Dm, F#m, Bb. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemLyricist`

The lyricist, or text writer, of the music contained in the audio file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMusicalGenre`

The musical genre of the song or composition contained in the audio file. For example: Jazz, Pop, Rock, Classical. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMusicalInstrumentCategory`

Specifies the category of an instrument. A CFString.

Files should have an instrument associated with them ("Other Instrument" is provided as a catch-all). For some categories, such as "Keyboards", there are instrument names which provide a more detailed instrument definition, for example "Piano" or "Organ".

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemMusicalInstrumentName`

Specifies the name of instrument relative to the instrument category. A CFString.

Files can have an instrument name associated with them if they have certain instrument categories. For example, the "Percussion" category has multiple instruments, including "Conga" and "Bongo".

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemRecordingDate`

The recording date of the song or composition. A CFDate.

This is in contrast to `kMDItemContentCreationDate` which, could indicate the creation date of an edited or 'mastered' version of the original art.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemRecordingYear`

Indicates the year the item was recorded. For example, 1964, 2003, etc. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemTempo`

A float value that specifies the beats per minute of the music contained in the audio file. A CFNumber.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemTimeSignature`

The time signature of the musical composition contained in the audio/MIDI file. For example: "4/4", "7/8". A CFString.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`MDItem.h`

File System Metadata Attribute Keys

Metadata attribute keys that describe the file system attributes for a file.

```

const CFStringRef kMDItemDisplayName;
const CFStringRef kMDItemFSContentChangeDate;
const CFStringRef kMDItemFSCreationDate;
const CFStringRef kMDItemFSExists;
const CFStringRef kMDItemFSInvisible;
const CFStringRef kMDItemFSIsExtensionHidden;
const CFStringRef kMDItemFSIsReadable;
const CFStringRef kMDItemFSIsWritable;
const CFStringRef kMDItemFSLabel;
const CFStringRef kMDItemFSName;
const CFStringRef kMDItemFSNodeCount;
const CFStringRef kMDItemFSOwnerGroupID;
const CFStringRef kMDItemFSOwnerUserID;
const CFStringRef kMDItemFSSize;
const CFStringRef kMDItemPath;

```

Constants

`kMDItemDisplayName`

The localized version of the file name. A CFString.

This is the localized version of the `LaunchServices` call `LSCopyDisplayNameForURL/LSCopyDisplayNameForRef`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSContentChangeDate`

The date the file contents last changed. A CFDate.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSCreationDate`

The date and time that the file was created. A CFDate.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSExists`

This attribute is deprecated and was never implemented.

Deprecated in Mac OS X v10.4.

Declared in `MDItem.h`.

`kMDItemFSInvisible`

Indicates whether the file is invisible. A CFBoolean.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSIsExtensionHidden`

Indicates whether the file extension of the file is hidden. A CFBoolean.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSIsReadable`

This attribute is deprecated and was never implemented.

Deprecated in Mac OS X v10.4.

Declared in `MDItem.h`.

`kMDItemFSIsWriteable`

This attribute is deprecated and was never implemented.

Deprecated in Mac OS X v10.4.

Declared in `MDItem.h`.

`kMDItemFSLabel`

Index of the Finder label of the file. Possible values are 0 through 7. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSName`

The file name of the item. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSNodeCount`

Number of files in a directory. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSOwnerGroupID`

The group ID of the owner of the file. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSOwnerUserID`

The user ID of the owner of the file. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemFSSize`

The size, in bytes, of the file on disk. A `CFNumber`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

`kMDItemPath`

The complete path to the file. A `CFString`.

Available in Mac OS X v10.4 and later.

Declared in `MDItem.h`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`MDItem.h`

MDQuery Reference

Derived From:	CType
Framework:	CoreServices/CoreServices.h
Declared in	MDQuery.h
Companion guides	Spotlight Overview Spotlight Query Programming Guide

Overview

MDQuery is a CF-compliant object, follows the CF conventions, and can be used with the CF polymorphic functions, such as `CFRetain`. MDQuery encapsulates queries against the System store of the file metadata.

An MDQuery normally executes asynchronously and posts progress notifications as the results are collected. During the gathering phase the query results conform to the specified value lists and sorting.

MDQuery gathers results and processes updates only while the current thread's run loop is running.

For functions that take an MDQueryRef parameter, if this parameter is not a valid MDQueryRef, the behavior is undefined. NULL is not a valid MDQueryRef.

For functions that take CF*Ref parameters, such as CFStringRef and CFArrayRef, if this parameter is not a valid CF object of the correct type, the behavior is undefined. NULL is not a valid CF*Ref.

Functions by Task

Creating Queries

[MDQueryCreate](#) (page 69)

Creates a new query instance.

[MDQueryCreateSubset](#) (page 70)

Creates a new query that is a subset of the specified parentquery.

[MDQuerySetSearchScope](#) (page 79) **Available in Mac OS X v10.5 through Mac OS X v10.5**

Sets the search scope for a query instance.

[MDQuerySetDispatchQueue](#) (page 79) **Available in Mac OS X v10.5 through Mac OS X v10.5**

Sets the dispatch queue on which query results will be delivered by MDQueryExecute.

Getting and Setting Query Parameters

[MDQueryGetBatchingParameters](#) (page 72)

Returns the current parameters that control the batching of progress notifications.

[MDQueryCopySortingAttributes](#) (page 67)

Returns the list of attribute names used to sort the results.

[MDQueryCopyQueryString](#) (page 67)

Returns the query string of the query.

[MDQuerySetMaxCount](#) (page 79) **Available in Mac OS X v10.5 through Mac OS X v10.5**

Sets the maximum number of results returned.

[MDQueryCopyValueListAttributes](#) (page 68) **Available in Mac OS X v10.5 through Mac OS X v10.5**

Returns the list of attribute names for which values are being collected by the query.

[MDQuerySetBatchingParameters](#) (page 76) **Available in Mac OS X v10.5 through Mac OS X v10.5**

Set the query batching parameters.

Setting Callback Functions

[MDQuerySetCreateResultFunction](#) (page 76)

Sets the function used to create the result objects of the MDQuery.

[MDQuerySetSortComparator](#) (page 80)

Sets the function used to sort the results of an MDQuery.

[MDQuerySetCreateValueFunction](#) (page 77)

Sets the function used to create the value objects of the MDQuery.

Starting, Stopping and Pausing Queries

[MDQueryExecute](#) (page 71)

Run the query, and populate the query with the results.

[MDQueryStop](#) (page 81)

Stops the query from generating more results.

[MDQueryDisableUpdates](#) (page 70)

Disables updates to the query result list.

[MDQueryEnableUpdates](#) (page 71)

Enables updates to the query result list.

[MDQueryIsGatheringComplete](#) (page 75)

Returns true if the first phase of a query, the initial result gathering, has finished.

Getting Query Result Values

[MDQueryCopyValuesOfAttribute](#) (page 68)

Returns the list of values from the results of the query for the specified attribute.

[MDQueryGetAttributeValueOfResultAtIndex](#) (page 72)

Returns the value of the named attribute for the result at the given index.

[MDQueryGetCountOfResultsWithAttributeValue](#) (page 73)

Returns the number of results which have the given attribute and attribute value.

[MDQueryGetIndexOfResult](#) (page 73)

Returns the current index of the given result.

[MDQueryGetResultAtIndex](#) (page 74)

Returns the current result at the given index.

[MDQueryGetResultCount](#) (page 75)

Returns the number of results currently collected by the query.

[MDQuerySetSortComparatorBlock](#) (page 81)

Sets the block used to sort the results of an MDQuery.

Getting the Type Identifier

[MDQueryGetTypeID](#) (page 75)

Returns the type identifier of all MDQuery instances

Functions

MDQueryCopyQueryString

Returns the query string of the query.

```
CFStringRef MDQueryCopyQueryString (
    MDQueryRef query
);
```

Parameters

query

The query.

Return Value

A CFStringRef containing the query string.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryCopySortingAttributes

Returns the list of attribute names used to sort the results.

```
CFArrayRef MDQueryCopySortingAttributes (
    MDQueryRef query
);
```

Parameters*query*

The query.

Return Value

A CFArrayRef containing the attribute names used to sort the query results.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryCopyValueListAttributes

Returns the list of attribute names for which values are being collected by the query.

```
CFArrayRef MDQueryCopyValueListAttributes (
    MDQueryRef query
);
```

Parameters*query*

The query.

Return Value

A CFArrayRef containing the attribute names of the collected values.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryCopyValuesOfAttribute

Returns the list of values from the results of the query for the specified attribute.

```
CFArrayRef MDQueryCopyValuesOfAttribute (
    MDQueryRef query,
    CFStringRef name
);
```

Parameters*query*

The query.

name

The attribute name to return the value of. If the attribute is not one of those requested when the query was created the behavior is undefined

Return Value

A `CFArrayRef` containing the value objects for the specified attribute. The array contents are not ordered and contain only one occurrence of each value. The array contents may change over time if the query is configured for live-updates.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

MDQueryCreate

Creates a new query instance.

```
MDQueryRef MDQueryCreate (
    CFAllocatorRef allocator,
    CFStringRef queryString,
    CFArrayRef valueListAttrs,
    CFArrayRef sortingAttrs
);
```

Parameters

allocator

The `CFAllocator` object to be used to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

queryString

The query expression string for this query.

valueListAttrs

An optional array of attribute names. The query will collect the values of these attributes into uniqued lists that can be used to summarize the results of the query and allow the user to further qualify the search. This parameter may be `NULL` if no value lists are required. Value list collection increases CPU usage and significantly increases the memory usage of an `MDQuery`. The attribute names are `CFStrings`.

sortingAttrs

An array of attribute names used to sort the results, or `NULL` if no sorting is required. The first name in the array is used as the primary sort key, the second as the secondary key, and so on. The comparison of like-typed values is a simple, literal comparison. Sorting increases memory usage and significantly increases the CPU usage of an `MDQuery`. It is usually more efficient to allow the `MDQuery` to sort the results than retrieving the values and sorting the results yourself. The attribute names are `CFStrings`.

Return Value

An `MDQueryRef`, or `NULL` on failure. If the query string is empty or malformed the function returns `NULL`.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

`MDQuery.h`

MDQueryCreateSubset

Creates a new query that is a subset of the specified parentquery.

```
MDQueryRef MDQueryCreateSubset (
    CFAllocatorRef allocator,
    MDQueryRef query,
    CFStringRef queryString,
    CFArrayRef valueListAttrs,
    CFArrayRef sortingAttrs
);
```

Parameters

allocator

The `CFAllocator` object to be used to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

query

The parent query

queryString

The query expression string for this query.

valueListAttrs

An optional array of attribute names. The query will collect the values of these attributes into uniqued lists that can be used to summarize the results of the query and allow the user to further qualify the search. This parameter may be `NULL` if no value lists are required. Value list collection increases CPU usage and significantly increases the memory usage of an `MDQuery`. The attribute names are `CFStrings`.

sortingAttrs

An array of attribute names used to sort the results, or `NULL` if no sorting is required. The first name in the array is used as the primary sort key, the second as the secondary key, and so on. The comparison of like-typed values is a simple, literal comparison. Sorting increases memory usage and significantly increases the CPU usage of an `MDQuery`. It is usually more efficient to allow the `MDQuery` to sort the results than retrieving the values and sorting the results yourself. The attribute names are `CFStrings`.

Return Value

An `MDQueryRef`, or `NULL` on failure. If the query string is empty or malformed the function returns `NULL`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

MDQueryDisableUpdates

Disables updates to the query result list.

```
void MDQueryDisableUpdates (
    MDQueryRef query
);
```

Parameters

query

The query.

Discussion

This function should be called before iterating over query results that could change due to live-updates. The disabled state is a counter and disabling can be done recursively and from different threads.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryEnableUpdates

Enables updates to the query result list.

```
void MDQueryEnableUpdates (
    MDQueryRef query
);
```

Parameters

query

The query.

Discussion

This function should be called when finished iterating through the list of results. Live-updates to the query results will continue when all the disables have been matched by a corresponding enable.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryExecute

Run the query, and populate the query with the results.

```
Boolean MDQueryExecute (
    MDQueryRef query,
    CFOptionFlags optionFlags
);
```

Parameters

query

The query to execute.

optionFlags

A bitwise OR of the MDQueryOptionFlags to be used by the query.

Return Value

Returns TRUE if the query was started, FALSE otherwise. Queries cannot be executed more than once.

Discussion

Queries only gather results or process updates while the current thread's run loop is running.

Queries have two phases: the initial gathering phase that collects all currently matching results and a second live-update phase. Updates occur during the live-update phase if a change in a file occurs such that it no longer matches the query or if it begins to match the query. Files which begin to match the query are added to the result list, and files which no longer match the query expression are removed from the result list.

Query notifications are posted within the context of the same thread which executes the query.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

MDQuery.h

MDQueryGetAttributeValueOfResultAtIndex

Returns the value of the named attribute for the result at the given index.

```
void * MDQueryGetAttributeValueOfResultAtIndex (
    MDQueryRef query,
    CFStringRef name,
    CFIndex idx
);
```

Parameters

query

The query.

name

The attribute name to return the values of. If the attribute is not one of those requested in the *valueListAttrs* or *sortingAttrs* parameters to one of the query creation functions, the result will be NULL.

idx

The index into the query's result list. If the index is negative or is equal to or larger than the current number of results in the query, the behavior is undefined.

Return Value

The value of the attribute, or NULL if the attribute doesn't exist for the specified result.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryGetBatchingParameters

Returns the current parameters that control the batching of progress notifications.


```
MDQueryBatchingParams MDQueryGetBatchingParameters (
    MDQueryRef query
);
```

Parameters*query*

The query.

Return Value

An MDQueryBatchingParams structure with the current batching parameters.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryGetCountOfResultsWithAttributeValue

Returns the number of results which have the given attribute and attribute value.

```
CFIndex MDQueryGetCountOfResultsWithAttributeValue (
    MDQueryRef query,
    CFStringRef name,
    CTypeRef value
);
```

Parameters*query*

The query.

*name*The attribute name to return the result count of. If the attribute is not one of those requested in the *valueListAttrs* parameter, the behavior is undefined.*value*

The attribute value for which to return the number of results with that value. This parameter may be NULL, in which case the number of results that do not contain the specified attribute is returned.

Return Value

The number of results containing that attribute and value.

Discussion

This count may change over time if the query allows live-updates.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryGetIndexOfResult

Returns the current index of the given result.

```
CFIndex MDQueryGetIndexOfResult (
    MDQueryRef query,
    const void *result
);
```

Parameters*query*

The query.

result

The result object to search for. If a custom create-result function has been set and this parameter is not a valid result object that the provided callbacks can handle, the behavior is undefined. If a custom create-result function has not been set this parameter must be a valid MDItemRef.

Return Value

The index of the given result, or `kCFNotFound` if the value is not one of the query's existing results. If you provided a custom result creation function result, the result will be objects created by that function.

Discussion

If a result-create function has been set, and the equal callback is non-NULL, it will be used to test the query's results against the candidate result.

Note that the index of a result can change over time if the query allows live-updates.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryGetResultAtIndex

Returns the current result at the given index.

```
const void * MDQueryGetResultAtIndex (
    MDQueryRef query,
    CFIndex idx
);
```

Parameters*query*

The query.

idx

The index into the query's result list. If the index is negative, or is equal to or larger than the current number of results in the query, the behavior is undefined.

Return Value

Returns the MDItemRef currently at the given index, or if a result-creation function has been set, returns the result returned by that function.

Discussion

This function causes the result object to be created if it hasn't been created already. For performance reasons you should only request objects that you require. If possible, call this function to fetch only the results you need to display or otherwise process.

Note that the index of a particular result can change over time if the query is configured to allow live-updates.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

MDQuery.h

MDQueryGetResultCount

Returns the number of results currently collected by the query.

```
CFIndex MDQueryGetResultCount (
    MDQueryRef query
);
```

Parameters

query

The query.

Return Value

The number of results in the query.

Discussion

Note that the number of results in a query will change over time as the query's result list is updated.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

MDQuery.h

MDQueryGetTypeID

Returns the type identifier of all MDQuery instances

```
CTypeID MDQueryGetTypeID (
    void
);
```

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryIsGatheringComplete

Returns true if the first phase of a query, the initial result gathering, has finished.

```
Boolean MDQueryIsGatheringComplete (
    MDQueryRef query
);
```

Parameters*query*

The query.

Return ValueReturns `TRUE` if the first phase of a query has completed, otherwise `FALSE`.**Availability**

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQuerySetBatchingParameters

Set the query batching parameters.

```
void MDQuerySetBatchingParameters (
    MDQueryRef query,
    MDQueryBatchingParams params
);
```

Parameters*query*

The query.

params

An MDQueryBatchingParams structure with the batching parameters to set.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQuerySetCreateResultFunction

Sets the function used to create the result objects of the MDQuery.

```
void MDQuerySetCreateResultFunction (
    MDQueryRef query,
    MDQueryCreateResultFunction func,
    void *context,
    const CFArrayCallBacks *cb
);
```

Parameters*query*

The query.

func

The callback function the MDQuery will use to create its results, such as those returned by the function `MDQueryGetResultAtIndex`. This parameter may be `NULL`, in which case any previous result creation settings are cancelled and the MDQuery will subsequently produce `MDItemRefs`. If a function is specified and is not of type `MDQueryCreateResultFunction` or does not behave as a `MDQueryCreateResultFunction` must, the behavior is undefined.

context

A pointer-sized user-defined value, that is passed as the third parameter to the create function. MDQuery does not use this value, does not retain the context in any way, and requires that the context be valid for the lifetime of the query. If the context is not what is expected by the create function, the behavior is undefined.

cb

A pointer to a `CFArrayCallBacks` structure initialized with the callbacks for the query to use to manage the created result objects. A copy of the contents of the callbacks structure is made, so that a pointer to a structure on the stack can be passed in, or can be reused for multiple query creations. Only version 0 of the `CFArrayCallBacks` is supported. The `retain` field may be `NULL`, in which case the MDQuery will not add a retain to the created results for the query. The `release` field may be `NULL`, in which case the MDQuery will not remove the query's retain (such as the one it gets from the create function) on the result objects when the query is destroyed. If the `copyDescription` field is `NULL`, the query will create a simple description for the result objects. If the `equal` field is `NULL`, the query will use pointer equality to test for equality of results. This callbacks parameter itself may be `NULL` in which case it is treated as a valid version 0 structure with all fields `NULL`. Otherwise, if any of the fields are not valid pointers to functions of the correct type, or this parameter is not a valid pointer to a `CFArrayCallBacks` callbacks structure, the behavior is undefined. If any of the value values returned from the create function is not one understood by one or more of the callback functions, the behavior when those callback functions are used is undefined. For example, if the create function can return `NULL`, then `NULL` must be understood by the callback functions as a possible parameter. The `retain` and `release` callbacks must be a matched set, you should not assume that the `retain` function will be unused or that additional reference counts will not be taken on the created results.

Discussion

If no create function is specified for an MDQuery, the default result objects are `MDItemRefs`. Results created after the function `MDQuerySetCreateResultFunction` is called are created through the specified create function, but values created before the function was set, or after it is unset, are not modified. It is not advisable to change this function after the function `MDQueryExecute` has been called. The create-result function is called lazily as results are requested from a query, it is not called on all results, and may not be called at all. This avoids the cost of creating potentially hundreds of thousands of what might be temporary objects.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

MDQuerySetCreateValueFunction

Sets the function used to create the value objects of the MDQuery.

```
void MDQuerySetCreateValueFunction (
    MDQueryRef query,
    MDQueryCreateValueFunction func,
    void *context,
    const CFArrayCallBacks *cb
);
```

Parameters*query*

The query.

func

The callback function the MDQuery should use to create the value list values, such as those returned by the function `MDQueryCopyValuesOfAttribute`. This parameter may be `NULL`, in which case any previous value creation settings are cancelled and the MDQuery will subsequently produce the default `CTypeRefs`. If a function is specified and is not of type `MDQueryCreateValueFunction` or does not behave as a `MDQueryCreateValueFunction` must, the behavior is undefined.

context

A pointer-sized user-defined value, that is passed as the third parameter to the create function. MDQuery does not use this value, does not retain the context in any way, and requires that the context be valid for the lifetime of the query. If the context is not what is expected by the create function, the behavior is undefined.

cb

A pointer to a `CFArrayCallBacks` structure initialized with the callbacks for the query to use to manage the created value objects. A copy of the contents of the callbacks structure is made, so that a pointer to a structure on the stack can be passed in, or can be reused for multiple query creations. Only version 0 of the `CFArrayCallBacks` is supported. The `retain` field may be `NULL`, in which case the MDQuery will not add a retain to the created values. The `release` field may be `NULL`, in which case the MDQuery will do nothing to remove the query's retain (such as the one it gets from the create function) on the value objects when the query is destroyed. If the `copyDescription` field is `NULL`, the query will create a simple description for the value objects. If the `equal` field is `NULL`, the query will use pointer equality to test for equality of values. This callbacks parameter itself may be `NULL` in which case it is treated as a valid version 0 structure with all fields `NULL`. Otherwise, if any of the fields are not valid pointers to functions of the correct type, or this parameter is not a valid pointer to a `CFArrayCallBacks` callbacks structure, the behavior is undefined. If any of the value values returned from the create function is not one understood by one or more of the callback functions, the behavior when those callback functions are used is undefined. For example, if the create function can return `NULL`, then `NULL` must be understood by the callback functions as a possible parameter. The retain and release callbacks must be a matched set, you should not assume that the retain function will be unused or that additional reference counts will not be taken on the created results.

Discussion

Values created after a create function is set will be created using the newly specified function, but existing values are not modified. It is not advisable to change this function after `MDQueryExecute` has been called with the query.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

MDQuerySetDispatchQueue

Sets the dispatch queue on which query results will be delivered by MDQueryExecute.

```
void MDQuerySetDispatchQueue (
    MDQueryRef query,
    dispatch_queue_t queue
);
```

Parameters

query

The query.

queue

The dispatch queue on which results should be delivered.

Discussion

It is not advisable to change set dispatch queue after MDQueryExecute (page 71) has been called with the query.

Setting the dispatch queue for a synchronous query (kMDQuerySynchronous (page 86)) has no effect.

Availability

Available in Mac OS X v10.6 and later.

Declared In

MDQuery.h

MDQuerySetMaxCount

Sets the maximum number of results returned.

```
void MDQuerySetMaxCount (
    MDQueryRef query,
    CFIndex size
);
```

Parameters

query

The query.

size

The maximum number of return results.

Discussion

This must be called before the query is executed.

Availability

Available in Mac OS X v10.5 and later.

Declared In

MDQuery.h

MDQuerySetSearchScope

Sets the search scope for a query instance.

```
void MDQuerySetSearchScope (
    MDQueryRef query,
    CFArrayRef scopeDirectories,
    OptionBits scopeOptions
);
```

Parameters*query*

The query object to modify.

scopeDirectories

A CFArray of CFStringRef or CFURLRef objects which specify where to search. For convenience the `kMDQueryScopeHome`, `kMDQueryScopeComputer` and `kMDQueryScopeNetwork` constants may also be included in the array.

scopeOptions

Additional options for modifying the search. Currently you must pass 0.

Discussion**Availability**

Available in Mac OS X v10.4 and later.

Related Sample Code

SpotlightAPI

Declared In

MDQuery.h

MDQuerySetSortComparator

Sets the function used to sort the results of an MDQuery.

```
void MDQuerySetSortComparator (
    MDQueryRef query,
    MDQuerySortComparatorFunction comparator,
    void *context
);
```

Parameters*query*

The query.

comparator

The callback function the MDQuery uses to sort the results list. This parameter may be NULL which cancels previous sort comparator settings. If a function is specified and is not of type `MDQuerySortComparatorFunction` or does not behave as a `MDQuerySortComparatorFunction` must, the behavior is undefined.

context

A pointer-sized user-defined value, that is passed as the third parameter to the create function. MDQuery does not use this value, does not retain the context in any way, and requires that the context be valid for the lifetime of the query. If the context is not what is expected by the create function, the behavior is undefined.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQuerySetSortComparatorBlock

Sets the block used to sort the results of an MDQuery.

```
void MDQuerySetSortComparatorBlock (
    MDQueryRef query,
    CFComparisonResult (^comparator)(const CTypeRef attrs1[], const CTypeRef
attrs2[])
);
```

Parameters*query*

The query.

comparator

The callback block the MDQuery will use to sort its results. The comparator may be called on multiple threads in parallel, and must be reentrant. To take advantage of parallel sorting, it is best to avoid any locking in the comparator.

The block may be NULL to cancel any custom comparator.

Discussion

You may set the comparator block as many times as you like, even while the query is executing. Whenever the comparator block is set, all results are re-sorted using the new comparator block before the function returns. The block can be NULL to cancel custom sorting and revert to the default sorting.

The default sort provided by [MDQueryCreate](#) (page 69) is an ascending sort. Strings are compared using `CFStringCompare` with the options `kCFCompareNonliteral | kCFCompareLocalized | kCFCompareNumerically`. `CFDataRefs` are compared by using `memcmp()` of the data pointers.

Availability

Available in Mac OS X v10.6 and later.

Declared In

MDQuery.h

MDQueryStop

Stops the query from generating more results.

```
void MDQueryStop (
    MDQueryRef query
);
```

Parameters*query*

The query.

Discussion

Queries may be executed only once and cannot be restarted. The query will first complete processing any unprocessed results.do. That may trigger a progress notification, so be aware of that if you are stopping a query from within your progress note handler; that is, during this function, a recursive progress and/or finished

notification might occur, which might recursively call your notification handler. It is safe to call this function recursively. You would call this function to stop a query that is generating way too many results to be useful, but still want to access the results that have come in so far. If a query is stopped before the gathering phase finishes, it will not report itself as finished, nor will it send out a finished notification.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

Callbacks

MDQueryCreateResultFunction

Callback function used to create the result objects stored and returned by a query.

```
typedef const void * (*MDQueryCreateResultFunction) (
    MDQueryRef query,
    MDItemRef item,
    void *context
);
```

You may implement the function like this:

```
(void) createResult(MDQueryRef query, MDItemRef item, void *context);
```

Parameters

query

The query instance.

item

The default MDItemRef for the result.

context

The user-defined context parameter provided to the MDQuerySetCreateResultFunction function.

Return Value

The function must return a pointer-sized value that can be managed with the callbacks which were set at the same time the create function was given to the query. The value must be returned with a reference (such as if the retain callback had been called on it), as implied by the Create name. If this function doesn't wish to create a new object it can return the given MDItemRef, but must also return it with a new retain, and the callbacks must be able to handle an MDItemRef as an input value. If this function returns NULL, NULL will be stored for the moment in the query, MDQueryGetResultAtIndex() may return NULL for that result, and the next time the query wants the result, it will call this function again.

Discussion

The function may hold onto the given attribute name and/or value in some other data structure, but must retain them for them to remain valid.

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQueryCreateValueFunction

Callback function used to create the value objects stored and returned by a query.

```
typedef const void * (*MDQueryCreateValueFunction) (
    MDQueryRef query,
    CFStringRef attrName,
    CTypeRef attrValue,
    void *context
);
```

You may implement the function like this:

```
(void) createValueFunction(MDQueryRef query, CFStringRef attrName, CTypeRef
attrValue, void *context);
```

Parameters*query*

The query instance.

attrName

The attribute name of the value.

attrValue

The default value of the value.

context

The user-defined context parameter provided in the `MDQuerySetCreateValueFunction` function.

Return Value

The function must return a pointer-sized value that can be managed with the callback which were set at the same time the create function was given to the query. The value must be returned with a reference (such as if the retain callback had been called on it), as implied by the Create name. If this function doesn't wish to create a new object, it can return the given `CTypeRef`, but must also return it with a new retain, and the callbacks must be able to handle a `CTypeRef` as an input value.

Discussion

The function may hold onto the given attribute name and/or value in some other data structure, but must retain them for them to remain valid

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

MDQuerySortComparatorFunction

Callback function used to sort the results of a query.

```
typedef CFComparisonResult (*MDQuerySortComparatorFunction) (
    const CTypeRef attrs1[],
    const CTypeRef attrs2[],
    void *context
);
```

You may implement the function like this:

```
CFStringRef sortFunction(CTypeRef attributes, CTypeRef values);
```

Parameters

query

The query instance.

attrs1

A C array of attribute values for a result. The values occur in the array in the same order and position that the attribute names were passed in the `sortingAttrs` array when the query was created. The values of the attributes will be `NULL` if the attribute doesn't exist for a result or if read access to that attribute is not allowed.

attrs2

A C array of attribute values for a result. The values occur in the array in the same order and position that the attribute names were passed in the `sortingAttrs` array when the query was created. The values of the attributes will be `NULL` if the attribute doesn't exist for a result or if read access to that attribute is not allowed.

context

The user-defined context parameter provided in the function `MDQuerySetSortComparator`.

Return Value

The function must return one of the `CFComparisonResults` `kCFCompareLessThan`, `kCFCompareEqualTo`, or `kCFCompareGreaterThan`. There is no provision for unordered results. The comparison should be a total order relation and produce the same results for the same inputs.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

Data Types

Batching Parameters

MDQueryBatchingParams

Structure containing the progress notification batching parameters of a `MDQuery`.

```
typedef struct {
    size_t first_max_num;
    size_t first_max_ms;
    size_t progress_max_num;
    size_t progress_max_ms;
    size_t update_max_num;
    size_t update_max_ms;
} MDQueryBatchingParams;
```

Fields`first_max_num`

The maximum number of results that can accumulate before the first progress notification is sent. This value is used only during the initial result-gathering phase of a query.

`first_max_ms`

The maximum number of milliseconds that can pass before the first progress notification is sent. This value is advisory, in that the notification will be triggered at some point after `first_max_ms` milliseconds have passed since the query began accumulating results. This value is used only during the initial result-gathering phase of a query.

`progress_max_num`

The maximum number of results that can accumulate before additional progress notifications are sent. This value is used only during the initial result-gathering phase of a query.

`progress_max_ms`

The maximum number of milliseconds that can pass before additional progress notifications are sent. This value is advisory, in that the notification will be triggered at some point after `progress_max_ms` milliseconds have passed since the query began accumulating results. This value is used only during the initial result-gathering phase of a query.

`update_max_num`

The maximum number of results that can accumulate before an update notification is sent. This value is used only during the live-update phase of a query.

`update_max_ms`

The maximum number of milliseconds that can pass before an update notification is sent. This value is advisory, in that the notification will be triggered at some point after `update_max_ms` milliseconds have passed since the query began accumulating results. This value is used only during the live-update phase of a query.

Discussion

The default batching parameters are undefined and subject to change.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`MDQuery.h`

Miscellaneous

MDQueryRef

A reference to a MDQuery object.

```
typedef struct __MDQuery *MDQueryRef;
```

Availability

Available in Mac OS X v10.4 and later.

Declared In

MDQuery.h

Constants

Query Option Flags

MDQueryOptionsFlags

Specify the execution mode for a query.

```
typedef enum {
    kMDQuerySynchronous = 1,
    kMDQueryWantsUpdates = 4,
} MDQueryOptionFlags;
```

Constants

`kMDQuerySynchronous`

Specifies that a query should block during the initial gather phase. The query's run loop will run in the default mode. If this option is not specified the query function returns immediately after starting the query asynchronously.

Available in Mac OS X v10.4 and later.

Declared in MDQuery.h.

`kMDQueryWantsUpdates`

Specifies that a query should provide live-updates to the result list after the initial gathering phase. Updates occur during the live-update phase if a change in a file occurs such that it no longer matches the query or if it begins to match the query. Files which begin to match the query are added to the result list, and files which no longer match the query expression are removed from the result list. Currently, this option is ignored if the `kMDQuerySynchronous` parameter is specified. This is subject to change, and you should always pass the value appropriate to the required behavior.

Available in Mac OS X v10.4 and later.

Declared in MDQuery.h.

Notifications

kMDQueryDidFinishNotification

Indicates that a query has finished with the initial result-gathering phase.

```
const CFStringRef kMDQueryDidFinishNotification;
```

Constants

`kMDQueryDidFinishNotification`

Posted to indicate that the query has finished the initial result-gathering phase.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

Discussion

The query results list is not updated as a result of this notification.

This notification is only sent to the application's notification center.

kMDQueryDidUpdateNotification

Indicates that a query's results list has change during the live-update phase of a query.

```
const CFStringRef kMDQueryDidUpdateNotification;
```

Constants

`kMDQueryDidUpdateNotification`

Notification posted to indicate that a change has occurred to the query's results list during the live-update phase of a query's execution.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

Discussion

The info dictionary of the notification can contain `kMDQueryUpdateAddedItems`, `kMDQueryUpdateChangedItems`, and `kMDQueryUpdateRemovedItems` keys.

This notification is only sent to the application's notification center.

kMDQueryProgressNotification

Indicates that a query's results list has change during the initial result-gathering phase of a query.

```
const CFStringRef kMDQueryProgressNotification;
```

Constants

`kMDQueryProgressNotification`

Notification posted to indicate that a change has occurred to the query's results list during the initial result-gathering phase of execution.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

Discussion

New items are typically added during this phase, however it is possible for items to be removed or updated, if the original file is changed. The info dictionary of the notification can contain `kMDQueryUpdateChangedItems` and `kMDQueryUpdateRemovedItems` keys.

For performance reasons added results are not indicated in progress notifications, to avoid the cost of creating the result objects.

This notification is only sent to the application's notification center.

Notification Info Keys

Query Result Change Keys

Specify the items that have changed in the query results.

```
const CFStringRef kMDQueryUpdateAddedItems;  
const CFStringRef kMDQueryUpdateChangedItems;  
const CFStringRef kMDQueryUpdateRemovedItems;
```

Constants

`kMDQueryUpdateAddedItems`

An array that identifies the items that have been added to the query results. This list only contains result objects that have previously been created, result objects that have not been created are not included.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

`kMDQueryUpdateChangedItems`

An array that identifies the items that have changed in the query results. This list only contains result objects that have previously been created, result objects that have not been created are not included.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

`kMDQueryUpdateRemovedItems`

An array that identifies the items that have been removed from the query results. This list only contains result objects that have previously been created, result objects that have not been created are not included.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

Query Search Scope Keys

Specify the scope of a query's search.


```
const CFStringRef kMDQueryScopeHome;
const CFStringRef kMDQueryScopeComputer;
const CFStringRef kMDQueryScopeNetwork;
const CFStringRef kMDQueryScopeAllIndexed;
const CFStringRef kMDQueryScopeComputerIndexed;
const CFStringRef kMDQueryScopeNetworkIndexed;
```

Constants

`kMDQueryScopeHome`

Specifies that the query should be restricted to the volume and directory that contains the current user's home directory.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

`kMDQueryScopeComputer`

Specifies that the query should be restricted to all locally mounted volumes, plus the user's home directory (which may be on a remote volume).

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

`kMDQueryScopeNetwork`

Specifies that the query should include all user mounted remote volumes.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

`kMDQueryScopeAllIndexed`

Specifies that the search should be restricted to indexed, locally mounted volumes and indexed user mounted remote volumes, plus the user's home directory.

Available in Mac OS X v10.6 and later.

Declared in `MDQuery.h`.

`kMDQueryScopeComputerIndexed`

Specifies that the search should be restricted to indexed, locally mounted volumes, plus the user's home directory (which may be on a remote volume).

Available in Mac OS X v10.6 and later.

Declared in `MDQuery.h`.

`kMDQueryScopeNetworkIndexed`

Specifies that the search should include indexed user mounted remote volumes.

Available in Mac OS X v10.6 and later.

Declared in `MDQuery.h`.

Discussion

These constants can be passed in the `scopeDirectories` array to the function `MDQuerySetSearchScope`.

Result Relevance Sorting Key

Key used in a user notification's description dictionary that indicates the relevance of a result.

```
const CFStringRef kMDQueryResultContentRelevance;
```

Constants

`kMDQueryResultContentRelevance`

A `CFNumberRef` with a floating point value between 0.0 and 1.0 inclusive.

Available in Mac OS X v10.4 and later.

Declared in `MDQuery.h`.

Discussion

The relevance value indicates the relevance of the content of a result object. The relevance is computed based on the value of the result itself, not on its relevance to the other results returned by the query.

The relevance value is for the content of the object only, not on the result item as a whole, and may not be computed if the item matches the query through evaluation of other attributes

If the value is not computed it is treated as an attribute on the item that does not exist.

Other References

MDLineage Reference

Framework:	CoreServices/CoreServices.h
Declared in	MDLineage.h
Companion guide	Spotlight Overview

Overview

The MDLineage functions can be used to set, alter, and store data concerning the relationships between different versions of the same logical file. Lineage information is stored on files using extended attributes. The lineage for a file can be retrieved or searched for using the Spotlight metadata attribute `kMDItemDocumentLineage`.

Functions by Task

Creating an MDLineage

[MDLineageCreate](#) (page 94)

Creates and returns a new, unique lineage.

[MDLineageCreateFromFile](#) (page 95)

Creates and returns a new, unique lineage using data from a given file.

[MDLineageCreateBranchFromFile](#) (page 94)

Creates and returns a new lineage descending from a given file.

Adding to, and removing from, a File

[MDLineageSetOnFile](#) (page 95)

Writes the given lineage data to a given file.

[MDLineageRemoveFromFile](#) (page 95)

Removes any existing lineage from a given file.

Functions

MDLineageCreate

Creates and returns a new, unique lineage. (Available in Mac OS X v10.5 through Mac OS X v10.5.)

```
CFDictionaryRef MDLineageCreate (
    CFAllocatorRef allocator
);
```

Parameters

allocator

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

Return Value

A new, globally unique lineage, or `NULL` if there was an error. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 through Mac OS X v10.5.

Declared In

MDLineage.h

MDLineageCreateBranchFromFile

Creates and returns a new lineage descending from a given file. (Available in Mac OS X v10.5 through Mac OS X v10.5.)

```
CFDictionaryRef MDLineageCreateBranchFromFile (
    CFAllocatorRef allocator,
    CFURLRef fileURL
);
```

Parameters

allocator

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

file

The URL for a file from which to create branched lineage.

Return Value

A new lineage descending from the data in *fileURL*, or `NULL` if there was no lineage data associated with *fileURL* or if there was an error. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 through Mac OS X v10.5.

Declared In

MDLineage.h

MDLineageCreateFromFile

Creates and returns a new, unique lineage using data from a given file. (Available in Mac OS X v10.5 through Mac OS X v10.5.)

```
CFDictionaryRef MDLineageCreateFromFile (
    CFAllocatorRef allocator,
    CFURLRef fileURL
);
```

Parameters

allocator

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

fileURL

The URL for a file from which to copy lineage data.

Return Value

A new, globally unique lineage using the data from *fileURL*, or `NULL` if there was no lineage data associated with *fileURL* or if there was an error. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 through Mac OS X v10.5.

Declared In

MDLineage.h

MDLineageRemoveFromFile

Removes any existing lineage from a given file. (Available in Mac OS X v10.5 through Mac OS X v10.5.)

```
Boolean MDLineageRemoveFromFile (
    CFURLRef fileURL
);
```

Parameters

file

The file from which to remove all lineage data.

Return Value

`true` if all lineage data was removed from the file at *fileURL*, otherwise `false`. If there was no lineage data on the file, returns `true`.

Availability

Available in Mac OS X v10.5 through Mac OS X v10.5.

Declared In

MDLineage.h

MDLineageSetOnFile

Writes the given lineage data to a given file. (Available in Mac OS X v10.5 through Mac OS X v10.5.)

```
MDLineageSetOnFile (  
    CFDictionaryRef lineage,  
    CFURLRef fileURL  
);
```

Parameters

lineage

Lineage data obtained from one of the MDLineageCreate functions.

file

The URL for a file on which to write the lineage.

Return Value

true if the operation succeeded, otherwise false.

Discussion

This function overwrites any lineage data already marked on the file.

Availability

Available in Mac OS X v10.5 through Mac OS X v10.5.

Declared In

MDLineage.h

MDImporter Reference

Framework:	CoreServices/CoreServices.h
Declared in	MDImporter.h
Companion guides	Spotlight Overview Spotlight Importer Programming Guide Plug-ins

Overview

An MDImporter is responsible for returning the metadata contained with a file. The Spotlight server is notified when a file is changed and loads the appropriate importer to extract the metadata. An importer is implemented as a CFPlugin.

Callbacks

ImporterImportData

Defines a pointer to an importer import callback that imports importers.

```
typedef Boolean (* ImporterImportData)
(
    void *thisInterface,
    CFMutableDictionaryRef attributes,
    CFStringRef contentTypeUTI,
    CFStringRef pathToFile
)
```

If you name your function `GetMetadataForFile`, you would declare it like this:

```
Boolean GetMetadataForFile
(
    void *thisInterface,
    CFMutableDictionaryRef attributes,
    CFStringRef contentTypeUTI,
    CFStringRef pathToFile
)
```

Parameters*thisInterface*

The CFPlugin object that is called. This value is passed to the callback function.

attributes

A mutable dictionary that you should populate with the metadata key/value pairs. This dictionary is created and passed to the callback function.

contentTypeUTI

The content type of the file as a uniform type identifier. This value is passed to the callback function.

pathToFile

The full path to the file. This value is passed to the callback function.

Return Value

Your callback function should return `true` if the metadata was successfully returned, `false` if the metadata was not returned.

Constants

kMDImporterTypeID

Type ID of an importer plug-in.

```
#define kMDImporterTypeID
CFUUIDGetConstantUUIDWithBytes(kCFAAllocatorDefault, 0x8B, 0x08, 0xC4, 0xBF, 0x41, 0x5
B, 0x11, 0xD8, 0xB3, 0xF9, 0x00, 0x03, 0x93, 0x67, 0x26, 0xFC);
```

Constants*kMDImporterTypeID*

Only importers with this type ID are loaded.

Available in Mac OS X v10.4 and later.

Declared in `MDImporter.h`.

Discussion

The string representation of this UUID is 8B08C4BF-415B-11D8-B3F9-0003936726FC.

kMDImporterInterfaceID

Interface required by a importer plug-in.

```
#define kMDImporterInterfaceID
CFUUIDGetConstantUUIDWithBytes(kCFAAllocatorDefault, 0x6E, 0xBC, 0x27, 0xC4, 0x89, 0x9
C, 0x11, 0xD8, 0x84, 0xAE, 0x00, 0x03, 0x93, 0x67, 0x26, 0xFC);
```

Constants*kMDImporterInterfaceID*

Importers must implement an interface corresponding to this UUID.

Available in Mac OS X v10.4 and later.

Declared in `MDImporter.h`.

Discussion

The string representation of this UUID is 6EBC27C4-899C-11D8-84A3-0003936726FC.

MDSchema Reference

Framework:	CoreServices/CoreServices.h
Declared in	MDSchema.h
Companion guide	Spotlight Overview

Overview

The MDSchema functions provide information about the metadata returned for an item including the type of metadata provided for a file type, the localized display name for a metadata attribute key, and the schema for a metadata attribute key.

Functions

MDSchemaCopyAllAttributes

Returns an array containing all the metadata attributes defined in the schema.

```
CFArrayRef MDSchemaCopyAllAttributes (
    void
);
```

Availability

Available in Mac OS X version 10.4 or later.

Declared In

MDSchema.h

MDSchemaCopyAttributesForContentType

Returns a dictionary containing the metadata attributes for the specified UTI type.

```
MD_BEGIN_C_DECLS CFDictionaryRef MDSchemaCopyAttributesForContentType (
    CFStringRef contentTypeUTI
);
```

Parameters

utiType

The UTI type.

Return Value

A dictionary containing `kMDAttributeDisplayValues` and `kMDAttributeAllValues` keys. Returns `NULL` if the UTI type is unknown.

Discussion

This function returns the metadata attributes for the specified UTI type only.

Availability

Available in Mac OS X version 10.4 or later.

Declared In

`MDSchema.h`

MDSchemaCopyDisplayDescriptionForAttribute

Returns the localized description of a metadata attribute key.

```
CFStringRef MDSchemaCopyDisplayDescriptionForAttribute (
    CFStringRef name
);
```

Parameters

name

The name of the metadata attribute key.

Return Value

The localized description of the metadata attribute, or `NULL` if no localized description is available.

Availability

Available in Mac OS X version 10.4 or later.

Declared In

`MDSchema.h`

MDSchemaCopyDisplayNameForAttribute

Returns the localized display name of a metadata attribute key.

```
CFStringRef MDSchemaCopyDisplayNameForAttribute (
    CFStringRef name
);
```

Parameters

name

The name of the metadata attribute key.

Return Value

The localized display name of the metadata attribute, or `NULL` if no localized display name is available.

Availability

Available in Mac OS X version 10.4 or later.

Related Sample Code

SpotlightAPI

Declared In

MDSchema.h

MDSchemaCopyMetaAttributesForAttribute

Returns a dictionary describing the values for the specified metadata attribute key.

```
CFDictionaryRef MDSchemaCopyMetaAttributesForAttribute (
    CFStringRef name
);
```

Parameters*name*

The name of the metadata attribute key.

Return Value

A dictionary describing the schema of the metadata attribute key.

Availability

Available in Mac OS X version 10.4 or later.

Declared In

MDSchema.h

Constants

Available Metadata Attribute Keys

Specify the available metadata attribute keys for a content type.

```
const CFStringRef    kMDAttributeDisplayValues;
const CFStringRef    kMDAttributeAllValues;
const CFStringRef    kMDAttributeReadOnlyValues;
const CFStringRef    kMDEXporterAvaliable;
```

Constants*kMDAttributeDisplayValues*

An array of strings containing the available display metadata attribute keys, or NULL if the type is not known by the system.

Available in Mac OS X v10.4 and later.

Declared in MDSchema.h.

kMDAttributeAllValues

An array of strings containing the available metadata attribute keys, or NULL if the type is not known by the system.

Available in Mac OS X v10.4 and later.

Declared in MDSchema.h.

`kMDAttributeReadOnlyValues`

An array of strings containing the read-only metadata attribute keys, or `NULL` if the type is not known by the system.

Available in Mac OS X v10.5 and later.

Declared in `MDSchema.h`.

`kMDExporterAvailable`

A `CFBoolean` that indicates if an exporter is available for this UTI type.

Available in Mac OS X v10.5 and later.

Declared in `MDSchema.h`.

Discussion

These keys are in the dictionary returned by the function `MDSchemaCopyAttributesForContentType`.

Metadata Attribute Schema Description Keys

Specify the schema of a metadata attribute key.

```
const CFStringRef kMDAttributeName;
const CFStringRef kMDAttributeType;
const CFStringRef kMDAttributeMultiValued;
```

Constants

`kMDAttributeName`

A string containing the name of the metadata attribute key.

Available in Mac OS X v10.4 and later.

Declared in `MDSchema.h`.

`kMDAttributeType`

A `CFNumberRef` or `CFTypeID` describing the type of data returned as the value of the metadata attribute key.

Available in Mac OS X v10.4 and later.

Declared in `MDSchema.h`.

`kMDAttributeMultiValued`

A boolean that indicates if the metadata attribute value is multi-valued. If this is `TRUE`, the metadata attribute value is an array of the types specified in `kMDAttributeType`.

Available in Mac OS X v10.4 and later.

Declared in `MDSchema.h`.

Discussion

These keys are in the dictionary returned by the function `MDSchemaCopyMetaAttributesForAttribute`.

Spotlight Metadata Attributes

Common Metadata Attribute Keys

Metadata attribute keys that are common to many file types.

`kMDItemAttributeChangeDate`

Date and time of the last change made to a metadata attribute.

Value Type:	CFDate
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

`kMDItemAudiences`

The audience for which the file is intended. The audience may be determined by the creator or the publisher or by a third party.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

`kMDItemAuthors`

The author, or authors, of the contents of the file. The order of the authors is preserved, but does not represent the main author or relative importance of the authors.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

`kMDItemAuthorAddresses`

This attribute indicates the author addresses of the document.

Value Type:	Array of CFStrings
--------------------	--------------------

Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.6 and later.

kMDItemCity

Identifies city of origin according to guidelines established by the provider. For example, "New York", "Cupertino", or "Toronto".

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemComment

A comment related to the file. This comment is not displayed by the Finder.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContactKeywords

A list of contacts that are associated with this document, not including the authors.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContentCreationDate

The date and time that the content was created.

Value Type: CFDate
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContentModificationDate

Date and time when the content of this item was modified.

Value Type: CFDate
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContentType

Uniform Type Identifier of the file. For example, a jpeg image file will have a value of public.jpeg. The value of this attribute is set by the Spotlight importer. Changes to this value are lost when the file attributes are next imported.

This attribute is marked as `nosearch`. You must specify this attribute key explicitly in a query in order for its contents to be searched.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContentTypeTree

Uniform Type Identifier hierarchy of the file. For example, a jpeg image file will return an array containing "public.jpeg", "public.image", and "public.data". The value of this attribute is set by the Spotlight importer. Changes to this value are lost when the file attributes are next imported.

This attribute is marked as `nosearch`. You must specify this attribute key explicitly in a query in order for its contents to be searched.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemContributors

Entities responsible for making contributions to the content of the resource. Examples of a contributor include a person, an organization or a service.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemCopyright

Copyright owner of the file contents.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemCountry

The full, publishable name of the country or primary location where the intellectual property of the item was created, according to guidelines of the provider.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemCoverage

Extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary, and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of co-ordinates or date ranges.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemCreator

Name of the application used to create the document content. For example, "Pages" or "Keynote".

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemDescription

Description of the kind of item this file represents.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h

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

kMDItemDisplayName

Localized version of the file name. This is the localized version of the LaunchServices call `LSCopyDisplayNameForURL()` / `LSCopyDisplayNameForRef()`.

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemDueDate

Date this item is due.

Value Type: CFDate

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemDurationSeconds

The duration, in seconds, of the content of the item. A value of 10.5 represents media that is 10 and 1/2 seconds long.

Value Type: CFNumber

Units: seconds

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemEmailAddress

Email addresses related to this item.

Value Type: Array of CFStrings

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemEncodingApplications

Applications used to convert the original content into its current form. For example, a PDF file might have an encoding application set to "Distiller".

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemFinderComment

Finder comments for this item.

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemFonts

Fonts used by this item. You should store the font's full name, the postscript name, or the font family name, based on the available information.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemHeadline

Publishable entry providing a synopsis of the contents of the item. For example, "Apple Introduces the iPod Photo".

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemIdentifier

Formal identifier used to reference the resource within a given context. For example, the Message-ID of a mail message.

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h

Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemInstantMessageAddresses

Instant message addresses related to this item.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemInstructions

Instructions concerning the use of the item, such as embargoes and warnings. For example, "Second of four stories".

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemKeywords

Keywords associated with this file. For example, "Birthday", "Important", etc.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemKind

Description of the kind of item this file represents.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemLanguages

Indicates the languages used by the item. The recommended best practice for the values of this attribute are defined by RFC 3066.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemLastUsedDate

Date and time that the file was last used. This value is updated automatically by LaunchServices everytime a file is opened by double clicking, or by asking LaunchServices to open a file.

Value Type: CFDate
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemNumberOfPages

Number of pages in the document.

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemNamedLocation

The name of the location or point of interest associated with the item. The name may be user provided.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.6 and later.

kMDItemOrganizations

Companies or organizations that created the document.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemPageHeight

Height of the document page, in points (72 points per inch). For PDF files this indicates the height of the first page only.

Value Type:	CFNumber
Units:	points
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemPageWidth

Width of the document page, in points (72 points per inch). For PDF files this indicates the width of the first page only.

Value Type:	CFNumber
Units:	points
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemParticipants

The list of people who are visible in an image or movie or written about in a document.

Value Type:	Array of CFStringss
Framework Path:	CoreServices/CoreServices
Header:	MDItem.h
Availability:	Available in Mac OS X v10.6 and later.

kMDItemPhoneNumbers

Phone numbers related to this item.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemProjects

List of projects related to this item. For example, if you were working on a movie, all of the files could be marked as belonging to the project "My Movie".

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemPublishers

Publishers of the item. For example, a person, an organization, or a service.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemRecipients

Recipients of this item.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemRecipientAddresses

This attribute indicates the recipient addresses of the document.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.6 and later.

kMDItemRights

Provides a link to information about rights held on the document. Contains a rights management statement for the document, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), copyright, and various property rights. If this attribute is absent, no assumptions can be made about the status of these and other rights with respect to the document.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemSecurityMethod

Encryption method used to make the item secure. PDF files return "None" or "Password Encrypted".

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemStarRating

User rating of this item. For example, the user rating (number of stars) of an iTunes track.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemStateOrProvince

Province or state of origin according to guidelines established by the provider. For example, "CA", "Ontario", or "Sussex".

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemTextContent

Contains a text representation of the content of the document. Data in multiple fields should be combined using a whitespace character as a separator. An application's Spotlight importer provides the content of this attribute.

Applications can create queries using this attribute, but are not able to read the value of this attribute directly.

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemTitle

Title of the item. For example, this could be the title of a document, the name of a song, or the subject of an email message.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemVersion

Version number of the item.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemWhereFroms

Describes where the item was obtained from. For example, a downloaded file may refer to the URL, files received by email may indicate the sender's email address, message subject, etc.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

Image Metadata Attribute Keys

Metadata attribute keys that are common to image files.

kMDItemAcquisitionMake

Manufacturer of the device used to acquire the document contents.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAcquisitionModel

Model of the device used to acquire the document contents.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h

Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAlbum

Title for the collection containing this item. This is analogous to a record label or photo album.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAperture

Aperture setting used to acquire the document contents. This unit is the APEX value.

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemBitsPerSample

Number of bits per sample. For example, the bit depth of an image (8-bit, 16-bit etc...) or the bit depth per audio sample of uncompressed audio data (8, 16, 24, 32, 64, etc..).

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemColorSpace

Color space model used by the document contents. For example, "RGB", "CMYK", "YUV", or "YCbCr".

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemEXIFVersion

Version of the EXIF header used to generate the metadata.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemExposureMode

Exposure mode used to acquire the document contents.

Value Type: CFNumber
Expected Values: 0 (auto exposure), 1 (manual exposure), 2 (auto bracket)
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemExposureProgram

Type of exposure program used by the camera to acquire the document contents. Possible values include: Manual, Normal, Aperture priority, etc.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemExposureTimeSeconds

Exposure time used to capture the document contents.

Value Type: CFNumber
Units: seconds
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemExposureTimeString

Time when the document contents were captured. Typically this corresponds to when a photograph is exposed.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h

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

kMDItemFNumber

Diameter of the aperture relative to the effective focal length of the lens.

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFlashOnOff

Whether a camera flash was used to capture the document contents.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFocalLength

Actual focal length of the lens, in millimeters.

Value Type: CFNumber
Units: millimeters
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemHasAlphaChannel

Whether the image has an alpha channel.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemISOSpeed

ISO speed used to acquire the document contents. For example, 100, 200, 400, etc.

Value Type: CFNumber

Units:	ISO Speed
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemLayerNames

Names of the layers in the file.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemMaxAperture

Smallest F number of the lens in APEX value units, usually in the range of 00.00 to 99.99.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemMeteringMode

Metering mode used to acquire the image.

Value Type:	CFString
Expected Values:	Unknown, Average, CenterWeightedAverage, Spot, MultiSpot, Pattern, Partial
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemOrientation

Orientation of the document contents.

Value Type:	CFNumber
Expected Values:	0 (landscape), 1 (portrait)
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemPixelHeight

Height, in pixels, of the contents. For example, the image height or the video frame height.

Value Type:	CFNumber
Units:	pixels
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemPixelWidth

Width, in pixels, of the contents. For example, the image width or the video frame width.

Value Type:	CFNumber
Units:	pixels
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemPixelWidth

The total number of pixels in the contents.. Same as [kMDItemPageWidth](#) (page 111) * [kMDItemPageHeight](#) (page 111).

Value Type:	CFNumber
Units:	pixels
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.6 and later.

kMDItemProfileName

Name of the color profile used by the document contents.

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemRedEyeOnOff

Whether red-eye reduction was used to take the picture.

Value Type:	CFBoolean
--------------------	-----------

Expected Values: 0 (no red-eye reduction mode or unknown), 1 (red-eye reduction used)
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemResolutionHeightDPI

Resolution height, in DPI, of the item.

Value Type: CFNumber
Units: dots per inch (DPI)
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemResolutionWidthDPI

Resolution width, in DPI, of the item.

Value Type: CFNumber
Units: dots per inch (DPI)
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemWhiteBalance

White balance setting of the camera when the picture was taken.

Value Type: CFNumber
Expected Values: 0 (auto white balance), 1 (manual)
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

Video Metadata Attribute Keys

Metadata attribute keys that are common to video files.

kMDItemAudioBitRate

Bit rate of the audio in the media.

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemCodecs

Codecs used to encode/decode the media.

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemDeliveryType

Method used to deliver streaming media.

Value Type: CFString
Expected Values: "Fast Start", "RTSP"
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemMediaTypes

Media types present in the content. For example, a QuickTime movie may return:

```
kMDItemMediaTypes = (Sound, Video, "Hinted Video Track", "Hinted Sound Track")
kMDItemMediaTypes = (Sound, Video)
kMDItemMediaTypes = ("MPEG1 Muxed")
```

Value Type: Array of CFStrings
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemStreamable

Whether the content is prepared for streaming.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemTotalBitRate

Total bit rate, audio and video combined, of the media.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemVideoBitRate

Bit rate of the video in the media.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

Audio Metadata Attribute Keys

Metadata attribute keys that describe an audio file.

kMDItemAppleLoopDescriptors

Specifies multiple pieces of descriptive information about a loop. Besides genre and instrument, files can contain descriptive information that help users in refining searches.

Value Type:	Array of CFStrings
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemAppleLoopsKeyFilterType

Specifies key filtering information about a loop. Loops are matched against projects that often differ in a major or minor key. To assist users in identifying loops that will "fit" with their compositions, loops can be tagged with one of the following key filters: "AnyKey", "Minor", "Major", "NeitherKey", or "BothKeys". "AnyKey" means that it fits with anything (whether in a major key, minor key or neither). "Minor" fits with compositions in a minor key. "NeitherKey" doesn't work well with compositions that are in major or minor key. "BothKeys" means it fits with compositions that are in major or minor key.

Value Type:	CFString
Expected Values:	"AnyKey", "Minor", "Major", "NeitherKey", "BothKeys"
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h

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

kMDItemAppleLoopsLoopMode

Specifies how a file should be played. Tagged files can either be loops or non-loops (e.g., a cymbal crash). "Looping" indicates if the file should be treated as a loop. "Non-looping" indicates the file should not be treated as a loop.

Value Type: CFString
Expected Values: "Looping", "Non-looping"
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAppleLoopsRootKey

Specifies the loop's original key. The key is the root note or tonic for the loop, and does not include the scale type

Value Type: CFString
Expected Values: "C", "C#/Db", "D", "D#/Eb", "E", "F", "F#/Gb", "G", "G#/Ab", "A", "A#/Bb", "B", "NoKey"
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAudioChannelCount

Number of channels in the audio data contained in the file. This integer value only represents the number of discrete channels of audio data found in the file. It does not indicate any configuration of the data in regards to a user's speaker setup.

Value Type: CFNumber
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAudioEncodingApplication

Name of the application that encoded the audio of the document.

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemAudioSampleRate

Sample rate of the item's audio data. The sample rate is a float value representing hz (audio_frames/second). For example: 44100.0, 22254.54.

Value Type:	CFNumber
Units:	hz (audio_frames/second)
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemAudioTrackNumber

Track number of a song or composition when it is part of an album.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemComposer

Composer of the song in the audio file.

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemIsGeneralMIDISequence

Whether the MIDI sequence contained in the file is set up for use with a General MIDI device.

Value Type:	CFBoolean
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemKeySignature

Musical key of the song in the audio file. For example: "C", "Dm", "F#m", "Bb".

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h

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

kMDItemLyricist

Lyricist of the song in the audio file.

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemMusicalGenre

Musical genre of the song or composition contained in the audio file. For example: "Jazz", "Pop", "Rock", "Classical".

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemMusicalInstrumentCategory

Specifies the category of an instrument. Files should have an instrument associated with them ("Other Instrument" is provided as a catch-all). For some categories, such as "Keyboards", there are instrument names which provide a more detailed instrument definition, for example "Piano" or "Organ".

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemMusicalInstrumentName

Specifies the name of instrument relative to the instrument category. Files can have an instrument name associated with them if they have certain instrument categories. For example, the "Percussion" category has multiple instruments, including "Conga" and "Bongo".

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemRecordingDate

Recording date of the song or composition. This is in contrast to `kMDItemContentCreationDate` which, could indicate the creation date of an edited or "mastered" version of the original art.

Value Type:	CFDate
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemRecordingYear

Year the item was recorded. For example: 1964, 1995, 1997, or 2003.

Value Type:	CFNumber
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemTempo

Tempo of the music in the audio file. A floating point value.

Value Type:	CFNumber
Units:	Beats per Minute (BPM)
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

kMDItemTimeSignature

Time signature of the musical composition contained in the audio/MIDI file. For example: "4/4", "7/8".

Value Type:	CFString
Framework Path:	CoreServices/CoreServices.h
Header:	MDItem.h
Availability:	Available in Mac OS X v10.4 and later.

File System Metadata Attribute Keys

Metadata attribute keys that describe the file system attributes for a file. These attributes are available for files on any mounted volume.

kMDItemFSContentChangeDate

Date the file contents last changed.

Value Type: CFDate
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFSCreationDate

Date that the contents of the file were created.

Value Type: CFDate
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFSInvisible

Whether the file is invisible.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFSIsExtensionHidden

Whether the file extension of the file is hidden.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

kMDItemFSLabel

Index of the Finder label of the file. Possible values are 0 through 7.

Value Type: CFNumber
Expected Values: 0 through 7
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h

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

kMDItemFSName

File name of the item.

Value Type: CFString

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemFSNodeCount

Number of files in a directory.

Value Type: CFNumber

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemFSOwnerGroupID

Group ID of the owner of the file.

Value Type: CFNumber

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemFSOwnerUserID

User ID of the owner of the file.

Value Type: CFNumber

Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

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

kMDItemFSSize

Size, in bytes, of the file on disk.

Value Type: CFNumber

Units: bytes

Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

`kMDItemPath`

Complete path to the file. This value of this attribute can be retrieved, but can't be used in a query or to sort search results. This attribute can't be used as a member of the `valueListAttrs` array parameter for [MDQueryCreate](#) (page 69) or [MDQueryCreateSubset](#) (page 70).

Value Type: CFString
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Available in Mac OS X v10.4 and later.

Deprecated Metadata Attribute Keys

Metadata attribute keys that have been deprecated.

`kMDItemFSExists`

This attribute is deprecated and was never implemented.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Deprecated in Mac OS X v10.4 and later.

`kMDItemFSIsReadable`

This attribute is deprecated and was never implemented.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h
Header: MDItem.h
Availability: Deprecated in Mac OS X v10.4 and later.

`kMDItemFSIsWritable`

This attribute is deprecated and was never implemented.

Value Type: CFBoolean
Framework Path: CoreServices/CoreServices.h

Header: MDItem.h

Availability: Deprecated in Mac OS X v10.4 and later.

Document Revision History

This table describes the changes to *Spotlight Reference Collection*.

Date	Notes
2006-07-13	Updated for Mac OS X v10.5. Added MDLineage Reference.
2006-05-23	First publication of this content as a collection of separate documents.

REVISION HISTORY

Document Revision History