
ISyncSessionDriverDataSource Protocol Reference

Data Management: Syncing





Apple Inc.
© 2009 Apple 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, iCal, Mac, and Mac OS are trademarks of Apple Inc., registered 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

ISyncSessionDriverDataSource Protocol Reference 5

Overview	5
Tasks	6
Getting Client Information	6
Negotiating	6
Pulling	6
Pushing	6
Using Sync Anchors	7
Instance Methods	7
applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:	7
changedRecordsForEntityName:moreComing:error:	8
changesForEntityName:moreComing:error:	9
clientDescriptionURL	10
clientIdentifier	10
deleteAllRecordsForEntityName:error:	10
entityNamesToPull	11
entityNamesToSync	11
identifiersForRecordsToDeleteForEntityName:moreComing:error:	11
lastAnchorForEntityName:	12
nextAnchorForEntityName:	13
preferredSyncModeForEntityName:	13
recordsForEntityName:moreComing:error:	14
schemaBundleURLs	15
sessionBeginTimeout	15
sessionPullChangesTimeout	15
Constants	16
ISyncSessionDriverMode	16
ISyncSessionDriverChangeResult	17

Document Revision History 19

Index 21

ISyncSessionDriverDataSource Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/SyncServices.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	ISyncSessionDriver.h

Overview

The `ISyncSessionDriverDataSource` protocol defines a set of methods that the data source of an `ISyncSessionDriver` object must implement. This document also includes optional informal protocol methods that a data source can implement.

A data source must implement the `clientIdentifier` (page 10) and `clientDescriptionURL` (page 10) methods in order for a driver to create an `ISyncClient` object. A data source must implement `preferredSyncModeForEntityName:` (page 13) to request a sync mode—for example, a slow sync when an application doesn't have information on changes since the last sync. However, as with any sync session, the sync engine decides on the actual sync mode used, which depends on many other factors.

A data source also needs to implement the `recordsForEntityName:moreComing:error:` (page 14) method to support slow syncing, and optionally, implement the `changedRecordsForEntityName:moreComing:error:` (page 8) or `changesForEntityName:moreComing:error:` (page 9) methods to support fast syncing. All of these pushing methods allow you to batch records and changes.

Similarly, the data source must implement the `applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:` (page 7) and `deleteAllRecordsForEntityName:error:` (page 10) methods to apply changes during the pulling phase of a sync session.

Optional methods include `entityNamesToSync` (page 11) and `entityNamesToPull` (page 11) which can return a subset of the entities used by the client. The default is to push and pull records for all the entities provided in the client description.

You should use sync anchors to improve performance and avoid serious errors. A sync anchor is an object that is unique per client and per entity, that is saved periodically throughout a sync session. The sync engine compares the clients locally stored sync anchors with its copies to determine the next sync mode. For example, if there is a discrepancy in sync anchors, the client must slow sync. Read *ISyncSession Class Reference* for details.

To use sync anchors, implement the `lastAnchorForEntityName:` (page 12) method to return the previous sync anchor for the specified entity, and implement the `nextAnchorForEntityName:` (page 13) method to return the new sync anchor for the specified entity. It is your responsibility to save the sync anchors

returned by the [nextAnchorForEntityName:](#) (page 13) method locally, and return them in a subsequent call to the [lastAnchorForEntityName:](#) (page 12) method during the next sync session. Although these methods are optional, if you implement one, you must implement the other.

Tasks

Getting Client Information

- [clientIdentifier](#) (page 10) *required method*
Returns the client's unique identifier specified when registering the client. (required)
- [clientDescriptionURL](#) (page 10) *required method*
Returns an NSURL object representing the path to the client description property list. (required)
- [schemaBundleURLs](#) (page 15) *required method*
Returns an array containing NSURL objects representing the path to schemas this client uses. (required)
- [entityNamesToSync](#) (page 11)
Returns an array of NSString objects representing the names of entities this client wants to sync.
- [entityNamesToPull](#) (page 11)
Returns an array of NSString objects representing the names of entities this client wants to pull.
- [sessionBeginTimeout](#) (page 15)
Returns the time, in seconds, that the client is willing to wait for a sync session to begin.
- [sessionPullChangesTimeout](#) (page 15)
Returns the time, in seconds, that the client is willing to wait for a sync session to mingle—that is, prepare to pull changes.

Negotiating

- [preferredSyncModeForEntityName:](#) (page 13) *required method*
Returns the client's preferred sync mode for the session. (required)

Pulling

- [applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:](#) (page 7) *required method*
Applies the given changes to a client's record during the pulling phase of a sync session. (required)
- [deleteAllRecordsForEntityName:error:](#) (page 10) *required method*
Deletes all records for the specified entity. (required)

Pushing

- [recordsForEntityName:moreComing:error:](#) (page 14) *required method*
Returns records for the given entity name that should be pushed to the sync engine during a slow sync. (required)

- [changedRecordsForEntityName:moreComing:error:](#) (page 8)
Returns changed records for the given entity name that should be pushed to the sync engine during a fast sync.
- [changesForEntityName:moreComing:error:](#) (page 9)
Returns the changes to records that should be pushed to the sync engine during a fast sync.
- [identifiersForRecordsToDeleteForEntityName:moreComing:error:](#) (page 11)
Returns the record identifiers for deleted records that should be pushed to the sync engine during a fast sync.

Using Sync Anchors

- [lastAnchorForEntityName:](#) (page 12)
Returns the last sync anchor for the specified entity name.
- [nextAnchorForEntityName:](#) (page 13)
Returns the next sync anchor for the specified entity name.

Instance Methods

applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:

Applies the given changes to a client's record during the pulling phase of a sync session. (required)

```
(ISyncSessionDriverChangeResult)applyChange:(ISyncChange *)change
forEntityName:(NSString *)entityName remappedRecordIdentifier:(NSString
**)outRecordIdentifier formattedRecord:(NSDictionary **)outRecord error:(NSError
**)outError
```

Discussion

This method applies the changes from the truth database to the local copy of the record. The *change* parameter is an *ISyncChange* object that describes the changes to a record since the last sync.

If the change is of type *ISyncChangeTypeDelete*, then the *outRecordIdentifier* and *outRecord* parameters are ignored. Otherwise, they may be used to pass back additional information to the driver.

If the change is of type *ISyncChangeTypeAdd* or *ISyncChangeTypeModify* and this method accepts the change, then it may set the value referenced by *outRecordIdentifier* to an alternate local record identifier. The sync engine uses the returned local record identifier when communicating future changes.

If the change is of type *ISyncChangeTypeAdd* or *ISyncChangeTypeModify* and this method accepts the change, then it may specify an alternate format by setting the value referenced by *outRecord* to the new format. See

[clientAcceptedChangesForRecordWithIdentifier:formattedRecord:newRecordIdentifier:](#) for more details on formatting records.

This method returns [schemaBundleURLs](#) (page 15) if the change is accepted (successfully applied), [sessionBeginTimeout](#) (page 15) if it is refused, and “Pulling” (page 6) if it is neither accepted nor refused. If a client refuses a change, the sync engine does not send the same change during any subsequent syncs unless the record is modified.

If an error occurs, this method returns [“Getting Client Information”](#) (page 6) and sets *outError* to an NSError object that describes the error. If this method returns [“Getting Client Information”](#) (page 6), the ISyncSessionDriver object that invoked this method cancels the sync session.

This method is invoked by the driver during the pulling phase of a sync session, after pushing records.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [clientAcceptedChangesForRecordWithIdentifier:formattedRecord:newRecordIdentifier:](#) (ISyncSession)
- [clientRefusedChangesForRecordWithIdentifier:](#) (ISyncSession)

changedRecordsForEntityName:moreComing:error:

Returns changed records for the given entity name that should be pushed to the sync engine during a fast sync.

```
- (NSDictionary *)changedRecordsForEntityName:(NSString *)entityName moreComing:(BOOL *)moreComing error:(NSError **)outError
```

Discussion

Returns a dictionary where the keys are record identifiers and the values are dictionary records that you want to push during a fast sync. The dictionary records must be suitable for pushing to the sync engine and belong to the entity specified by *entityName*. A dictionary record contains the properties of a record that you want to sync. All dictionary records must contain a value for the `ISyncRecordEntityNameKey` key that identifies the record’s entity. This method returns an empty dictionary if there are no records to push for this entity.

The *moreComing* parameter is used to batch records. If this method sets the value referenced by *moreComing* to YES, then this method is invoked repeatedly during the pushing phase of a sync session until *moreComing* is set to NO.

Because this method is invoked during a fast sync, it returns only records that changed since the last sync session. If this method batches the records, all changes should be returned by multiple invocations of this method before *moreComing* is set to NO. The sync engine compares each dictionary record with the previous version to determine which properties changed. Use the [changesForEntityName:moreComing:error:](#) (page 9) method instead if you know which properties changed.

If an error occurs, this method returns *nil* and sets *outError* to an NSError object that describes the error. If this method returns *nil*, the ISyncSessionDriver object that invoked this method cancels the sync session.

This method is invoked by the driver during the pushing phase of a sync session, before pulling records. A data source of an ISyncSessionDriver object is required to implement this method or the [changesForEntityName:moreComing:error:](#) (page 9) method if it requests a fast sync—that is, if the [preferredSyncModeForEntityName:](#) (page 13) method may return `ISyncSessionDriverDataSource` (page 5).

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [changesForEntityName:moreComing:error:](#) (page 9)
- [preferredSyncModeForEntityName:](#) (page 13)

Declared In

ISyncSessionDriver.h

changesForEntityName:moreComing:error:

Returns the changes to records that should be pushed to the sync engine during a fast sync.

```
- (NSArray *)changesForEntityName:(NSString *)entityName moreComing:(BOOL *)moreComing error:(NSError **)outError
```

Discussion

Returns an array of `ISyncChange` objects describing the changes made to records since the last sync session. The array should only contain changes to records belonging to the entity specified by `entityName`. Returns an empty array if there are no changes to push for this entity.

The `moreComing` parameter is used to batch changes. If this method sets the value referenced by `moreComing` to YES, then this method is invoked repeatedly during the pushing phase of a sync session until `moreComing` is set to NO.

Because this method is invoked during a fast sync, it should return all changes since the last sync session. If this method batches the records, all changed records should be returned by multiple invocations of this method before `moreComing` is set to NO. Use the [changedRecordsForEntityName:moreComing:error:](#) (page 8) method instead if you don't know which properties changed.

If an error occurs, this method returns `nil` and sets `outError` to an `NSError` object that describes the error. If this method returns `nil`, the `ISyncSessionDriver` object that invoked this method cancels the sync session.

This method is invoked by the driver during the pushing phase of a sync session, before pulling records. A data source of an `ISyncSessionDriver` object is required to implement this method or the [changedRecordsForEntityName:moreComing:error:](#) (page 8) method if it requests a fast sync—that is, if the [preferredSyncModeForEntityName:](#) (page 13) method may return `ISyncSessionDriverDataSource` (page 5).

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [changedRecordsForEntityName:moreComing:error:](#) (page 8)
- [preferredSyncModeForEntityName:](#) (page 13)

Declared In

ISyncSessionDriver.h

clientDescriptionURL

Returns an NSURL object representing the path to the client description property list. (required)

```
- (NSURL *)clientDescriptionURL
```

Discussion

The client description property list specifies client information that the sync engine needs to know to sync its records. See *Sync Services Programming Guide* for a complete description of the client description file.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

See Also

```
- registerClientWithIdentifier:descriptionFilePath: (ISyncManager)
```

clientIdentifier

Returns the client's unique identifier specified when registering the client. (required)

```
- (NSString *)clientIdentifier
```

Discussion

There are no restrictions on the content or length of the client identifier, but it must be unique across all clients. Typically, it's a DNS-style name such as `com.apple.iCal`.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

See Also

```
- registerClientWithIdentifier:descriptionFilePath: (ISyncManager)
```

deleteAllRecordsForEntityName:error:

Deletes all records for the specified entity. (required)

```
- (BOOL)deleteAllRecordsForEntityName:(NSString *)entityName error:(NSError **)outError
```

Discussion

Returns YES if the request to delete all records belonging to the entity specified by *entityName* is accepted, otherwise NO. This method only returns NO and sets *outError* to an NSError object that describes the error if a serious error occurred deleting all records.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

entityNamesToPull

Returns an array of NSString objects representing the names of entities this client wants to pull.

- (NSArray *)entityNamesToPull

Discussion

Optionally implement this method to return the names of the entities to pull that must be a subset of the entity names returned by the [entityNamesToSync](#) (page 11) method. If this method is not implemented, the sync session pulls the entities returned by the [entityNamesToSync](#) (page 11) method. Returns an empty array if this client doesn't want to pull any entities.

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [entityNamesToSync](#) (page 11)

Declared In

ISyncSessionDriver.h

entityNamesToSync

Returns an array of NSString objects representing the names of entities this client wants to sync.

- (NSArray *)entityNamesToSync

Discussion

Returns an empty array if this client doesn't want to sync any entities. The sync session pushes and pulls the entities returned by this method unless an alternate set of entities is specified by the optional [entityNamesToPull](#) (page 11) method. If this method is not implemented, the driver syncs all enabled entities contained in the client description property list.

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [entityNamesToPull](#) (page 11)

Declared In

ISyncSessionDriver.h

identifiersForRecordsToDeleteForEntityName:moreComing:error:

Returns the record identifiers for deleted records that should be pushed to the sync engine during a fast sync.

- (NSArray *)identifiersForRecordsToDeleteForEntityName:(NSString *)entityName
moreComing:(BOOL *)moreComing error:(NSError **)outError

Discussion

Returns an array of NSString objects representing the record identifiers of records that the client deleted since the last sync. The array should contain only identifiers of deleted records belonging to the entity specified by *entityName*. This method is invoked during a fast sync only. Returns an empty array if there are no deleted records to push for this entity.

The *moreComing* parameter is used to batch changes. If this method sets the value referenced by *moreComing* to YES, then this method is invoked repeatedly during the pushing phase of a sync session until *moreComing* is set to NO.

Alternatively, you can implement the [changedRecordsForEntityName:moreComing:error:](#) (page 8) method using the `ISyncChangeTypeDelete` constant to denote a deleted record.

If an error occurs, this method returns `nil` and sets *outError* to an NSError object that describes the error. If this method returns `nil`, the `ISyncSessionDriver` object that invoked this method cancels the sync session.

This method is invoked by the driver during the pushing phase of a sync session, before pulling records. A data source of an `ISyncSessionDriver` object can optionally implement this method or the [changesForEntityName:moreComing:error:](#) (page 9) method using the `ISyncChangeTypeDelete` constant to denote a deleted record.

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [changesForEntityName:moreComing:error:](#) (page 9)
- [preferredSyncModeForEntityName:](#) (page 13)

Declared In

`ISyncSessionDriver.h`

lastAnchorForEntityName:

Returns the last sync anchor for the specified entity name.

```
- (NSString *)lastAnchorForEntityName:(NSString *)entityName
```

Parameters

entityName

An entity name.

Return Value

A sync anchor corresponding to the entity name that was saved locally and returned by the [nextAnchorForEntityName:](#) (page 13) method in the previous sync session.

Discussion

This method is invoked immediately after a sync session is created and before pushing records.

Note: This method is optional.

However, if you implement this method you must also implement the [nextAnchorForEntityName:](#) (page 13) method.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [nextAnchorForEntityName:](#) (page 13)

Declared In

ISyncSessionDriver.h

nextAnchorForEntityName:

Returns the next sync anchor for the specified entity name.

```
- (NSString *)nextAnchorForEntityName:(NSString *)entityName
```

Parameters

entityName

An entity name.

Return Value

A new sync anchor corresponding to the entity name that is saved locally.

Sync anchors must be globally unique `NSString` objects. Typically, sync anchors contain a UUID or date.

Discussion

This method is invoked once per entity name after pushing records just before mingling, and once per entity name after pulling records just before accepting changes.

Note: This method is optional.

However, if you implement this method you must also implement the [lastAnchorForEntityName:](#) (page 12) method.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [lastAnchorForEntityName:](#) (page 12)

Declared In

ISyncSessionDriver.h

preferredSyncModeForEntityName:

Returns the client's preferred sync mode for the session. (required)

```
- (ISyncSessionDriverMode)preferredSyncModeForEntityName:(NSString *)entity
```

Discussion

Returns one of these constants that specifies the preferred sync mode for this client: [ISyncSessionDriverDataSource](#) (page 5), [sessionPullChangesTimeout](#) (page 15), and “[Negotiating](#)” (page 6). This method is invoked by the driver during the negotiation phase of a sync session, before pushing records.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `clientDidResetEntityNames:` (ISyncSession)
- `clientWantsToPushAllRecordsForEntityNames:` (ISyncSession)
- `shouldPushChangesForEntityName:` (ISyncSession)
- `shouldPushAllRecordsForEntityName:` (ISyncSession)
- `shouldPullChangesForEntityName:` (ISyncSession)
- `shouldReplaceAllRecordsOnClientForEntityName:` (ISyncSession)

recordsForEntityName:moreComing:error:

Returns records for the given entity name that should be pushed to the sync engine during a slow sync. (required)

```
- (NSDictionary *)recordsForEntityName:(NSString *)entityName moreComing:(BOOL *)moreComing error:(NSError **)outError
```

Discussion

Returns a dictionary where the keys are record identifiers and the values are dictionary records that you want to push. The dictionary records must be suitable for pushing to the sync engine and belong to the entity specified by `entityName`. A dictionary record contains the properties of a record that you want to sync. All dictionary records must contain a value for the `ISyncRecordEntityNameKey` key that identifies the record's entity. This method returns an empty dictionary if there are no records to push for this entity.

The `moreComing` parameter is used to batch records. If this method sets the value referenced by `moreComing` to YES, then this method is invoked repeatedly during the pushing phase of a sync session until `moreComing` is set to NO.

Because this method is invoked during a slow sync, it should return all records for the given entity. If this method batches the records, all records should be returned by multiple invocations of this method before `moreComing` is set to NO. Otherwise, the sync engine assumes you deleted the records that you did not push and data loss may result.

If an error occurs, this method returns `nil` and sets `outError` to an NSError object that describes the error. If this method returns `nil`, the `ISyncSessionDriver` object that invoked this method cancels the sync session.

This method is invoked by the driver during the pushing phase of a sync session, before pulling records.

This method is required.



Warning: If this method does not return all the records that the client was known to have on the last sync, the sync engine assumes the record was deleted and deletes it from the truth database.

Availability

Available in Mac OS X v10.5 and later.

schemaBundleURLs

Returns an array containing NSURL objects representing the path to schemas this client uses. (required)

- (NSArray *)schemaBundleURLs

Discussion

A schema can define new entities and properties, and extend existing entities. A schema bundle may contain other files, such as images and localization files. The returned array should contain URLs for all the schemas—including the public schemas—that this client intends to use. See *Sync Services Programming Guide* for more details on the schema format and contents of a schema bundle.

This method is required.

Availability

Available in Mac OS X v10.5 and later.

See Also

- registerSchemaWithBundlePath: (ISyncManager)

sessionBeginTimeout

Returns the time, in seconds, that the client is willing to wait for a sync session to begin.

- (NSTimeInterval)sessionBeginTimeout

Discussion

The default value is 60.0 seconds.

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

+ beginSessionWithClient:entityNames:beforeDate: (ISyncSession)

Declared In

ISyncSessionDriver.h

sessionPullChangesTimeout

Returns the time, in seconds, that the client is willing to wait for a sync session to mingle—that is, prepare to pull changes.

- (NSTimeInterval)sessionPullChangesTimeout

Discussion

The default value is 600.0 seconds.

This method is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- prepareToPullChangesForEntityNames:beforeDate: (ISyncSession)

Declared In

ISyncSessionDriver.h

Constants

ISyncSessionDriverMode

Specifies the preferred sync mode for a client.

```
typedef enum {
    ISyncSessionDriverModeFast = 1,
    ISyncSessionDriverModeSlow,
    ISyncSessionDriverModeRefresh,
} ISyncSessionDriverMode;
```

Constants

ISyncSessionDriverModeFast

Indicates that the client wants to fast sync.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

ISyncSessionDriverModeRefresh

Indicates that the client wants to refresh sync. If the [preferredSyncModeForEntityName:](#) (page 13) method returns this constant, the ISyncSessionDriver object sends `clientDidResetEntityNames:` to the ISyncSession object.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

ISyncSessionDriverModeSlow

Indicates that the client wants to slow sync. If the client slow syncs, it needs to push every record.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

Discussion

Use these constants as possible return values for the [preferredSyncModeForEntityName:](#) (page 13) method. Read *Managing Your Sync Session* in *Sync Services Programming Guide* for a description of the different sync modes.

Availability

Available in Mac OS X v10.5 and later.

ISyncSessionDriverChangeResult

Specifies whether a change was applied.

```
typedef enum {
    ISyncSessionDriverChangeRefused = 0,
    ISyncSessionDriverChangeAccepted,
    ISyncSessionDriverChangeIgnored,
    ISyncSessionDriverChangeError
} ISyncSessionDriverChangeResult;
```

Constants

ISyncSessionDriverChangeAccepted

Indicates the client accepted the change.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

ISyncSessionDriverChangeError

Indicates an error occurred while applying the change.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

ISyncSessionDriverChangeIgnored

Indicates the client ignored the change.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

ISyncSessionDriverChangeRefused

Indicates the client refused the change.

Available in Mac OS X v10.5 and later.

Declared in ISyncSessionDriver.h.

Discussion

Use these constants as possible return values for the

[applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:](#) (page 7) method.

Availability

Available in Mac OS X v10.5 and later.

Document Revision History

This table describes the changes to *ISyncSessionDriverDataSource Protocol Reference*.

Date	Notes
2009-11-17	Updated optional status of protocol methods.
2009-04-02	Updated per Mac OS X v10.6 changes.
2007-07-11	New document that describes the ISyncSessionDriverDataSource protocol.

REVISION HISTORY

Document Revision History

Index

A

`applyChange:forEntityName:remappedRecordIdentifier:formattedRecord:error:`
protocol instance method 7

C

`changedRecordsForEntityName:moreComing:error:`
protocol instance method 8
`changesForEntityName:moreComing:error:` **protocol instance method 9**
`clientDescriptionURL` **protocol instance method 10**
`clientIdentifier` **protocol instance method 10**

D

`deleteAllRecordsForEntityName:error:` **protocol instance method 10**

E

`entityNamesToPull` **protocol instance method 11**
`entityNamesToSync` **protocol instance method 11**

I

`identifiersForRecordsToDeleteForEntityName:moreComing:error:` **protocol instance method 11**
`ISyncSessionDriverChangeAccepted` **constant 17**
`ISyncSessionDriverChangeError` **constant 17**
`ISyncSessionDriverChangeIgnored` **constant 17**
`ISyncSessionDriverChangeRefused` **constant 17**
`ISyncSessionDriverChangeResult` **17**
`ISyncSessionDriverMode` **16**

`ISyncSessionDriverModeFast` **constant 16**
`ISyncSessionDriverModeRefresh` **constant 16**
`ISyncSessionDriverModeSlow` **constant 16**

L

`lastAnchorForEntityName:` **protocol instance method 12**

N

`nextAnchorForEntityName:` **protocol instance method 13**

P

`preferredSyncModeForEntityName:` **protocol instance method 13**

R

`recordsForEntityName:moreComing:error:` **protocol instance method 14**

S

`schemaBundleURLs` **protocol instance method 15**
`sessionBeginTimeout` **protocol instance method 15**
`sessionPullChangesTimeout` **protocol instance method 15**