
Sync Services Reference Update

Data Management: Syncing



2008-04-29



Apple Inc.
© 2008 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, iPod, Mac, Mac OS, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

MobileMe is a trademark of Apple Inc.

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 to Sync Services Reference Update 5

Organization of This Document 5

See Also 5

10.6 Symbol Changes 7

Classes 7

ISyncClient 7

ISyncManager 7

ISyncSession 7

ISyncSessionDriver 8

C Symbols 8

ISyncChange.h 8

SyncServicesErrors.h 9

10.5 Symbol Changes 11

Classes 11

ISyncManager 11

ISyncRecordReference (New) 11

ISyncRecordSnapshot 11

ISyncSession 12

ISyncSessionDriver (New) 12

NSObject 13

NSPersistentStoreCoordinator 14

Protocols 15

ISyncSessionDriverDataSource (New) 15

NSPersistentStoreCoordinatorSyncing (New) 15

C Symbols 17

ISyncCommon.h 17

ISyncSessionDriver.h 17

SyncServicesErrors.h 18

10.4 Symbol Changes 19

Classes 19

ISyncChange (New) 19

ISyncClient (New) 19

ISyncFilter (New) 21

ISyncManager (New) 22

ISyncRecordSnapshot (New) 22

- ISyncSession (New) 23
- Protocols 25
 - ISyncFiltering (New) 25
- C Symbols 25
 - ISyncChange.h 26
 - ISyncClient.h 26
 - ISyncCommon.h 27
 - ISyncManager.h 27
 - ISyncSession.h 27

Document Revision History 29

Introduction to Sync Services Reference Update

This document summarizes the symbols that have been added to the Sync Services framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

See Also

For reference documentation on this framework, see *Sync Services Framework Reference*

10.6 Symbol Changes

This article lists the symbols added to `SyncServices.framework` in Mac OS X v10.6.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

ISyncClient

Complete reference information is available in the `ISyncClient` reference.

Instance Methods

<code>formatsRelationships</code>	Returns a Boolean value indicating whether the client may reformat relationships.
<code>setFormatsRelationships:</code>	Sets whether the client may reformat relationships.

ISyncManager

Complete reference information is available in the `ISyncManager` reference.

Instance Methods

<code>clientWithIdentifier:needsSyncing:</code>	Notifies the sync engine that a client needs to sync the next time another client is syncing the same data classes.
<code>syncDisabledReason</code>	Returns the reason the sync engine may be disabled.

ISyncSession

Complete reference information is available in the `ISyncSession` reference.

Instance Methods

<code>ping</code>	Notifies the sync engine that the client is alive but busy.
-------------------	---

ISyncSessionDriver

Complete reference information is available in the `ISyncSessionDriver` reference.

Delegate Methods

<code>sessionDriver:didNegotiateAndReturnError:</code>	Informs the receiver that the client did negotiate with the sync engine. (required)
<code>sessionDriver:didReceiveSyncAlertAndReturnError:</code>	Informs the receiver that the client received a sync alert. (required)
<code>sessionDriver:willNegotiateAndReturnError:</code>	Informs the receiver that the client will negotiate with the sync engine. (required)

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

ISyncChange.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>ISyncChangePropertyValueIsDefaultKey</code>	Key for the default value of the property.
<code>ISyncChangeTypeNone</code>	Indicates a record was modified.

SyncServicesErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>ISyncInvalidArgumentsException</code>	Occurs when a Sync Services method is passed bad or inconsistent arguments.
<code>ISyncInvalidSchemaException</code>	Occurs when a client tries to register an invalid schema or schema extension.
<code>ISyncServerDisabledReason</code>	Indicates the reason the sync engine may be unavailable.
<code>ISyncServerDisabledReasonByPreference</code>	The sync engine is disabled because of a preference setting.
<code>ISyncServerDisabledReasonNone</code>	The sync engine is enabled.
<code>ISyncServerDisabledReasonSharedNetworkHome</code>	The sync engine is busy syncing a network home directory.
<code>ISyncServerDisabledReasonUnknown</code>	The sync engine fails to respond due to an unexpected error.
<code>ISyncServerDisabledReasonUnresponsive</code>	Sending the <code>isEnabled</code> message to the sync engine timed out.

10.5 Symbol Changes

This article lists the symbols added to `SyncServices.framework` in Mac OS X v10.5.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

ISyncManager

Complete reference information is available in the [ISyncManager reference](#).

Instance Methods

<code>addRequestMode:</code>	Adds a mode to the set of run-loop input modes that the receiver uses for connection requests.
<code>removeRequestMode:</code>	Removes a mode from the set of run-loop input modes the receiver uses for connection requests.
<code>requestModes</code>	Returns the set of request modes the receiver registers with its <code>NSRunLoop</code> object.

ISyncRecordReference (New)

Complete reference information is available in the [ISyncRecordReference reference](#).

ISyncRecordSnapshot

Complete reference information is available in the [ISyncRecordSnapshot reference](#).

Instance Methods

<code>recordIdentifierForReference:isModified:</code>	Returns the record identifier that corresponds to the given record reference object.
---	--

<code>recordReferenceForRecordWithIdentifier:</code>	Returns a record reference that corresponds to the given record identifier.
--	---

ISyncSession

Complete reference information is available in the `ISyncSession` reference.

Class Methods

<code>beginSessionInBackgroundWithClient:entityNames:target:selector:lastAnchors:</code>	Creates a new sync session for the specified client asynchronously using sync anchors.
<code>beginSessionWithClient:entityNames:beforeDate:lastAnchors:</code>	Creates and returns a new sync session for the specified client using sync anchors.

Instance Methods

<code>clientCommittedAcceptedChangesWithNextAnchors:</code>	Sets the sync anchors for each entity whose records were successfully updated during the pulling phase.
<code>clientFinishedPushingChangesWithNextAnchors:</code>	Sets the sync anchors for each entity whose records were successfully pushed by the client.

ISyncSessionDriver (New)

Complete reference information is available in the `ISyncSessionDriver` reference.

Class Methods

<code>sessionDriverWithDataSource:</code>	Creates and returns a new driver object with the specified data source object.
---	--

Instance Methods

<code>client</code>	Returns the client object used by the receiver to perform the sync operation.
<code>dataSource</code>	Returns the data source object for the receiver.
<code>delegate</code>	Returns the receiver's delegate.

<code>finishSyncing</code>	Notifies the sync engine that the client is done syncing.
<code>handlesSyncAlerts</code>	Returns a Boolean value indicating whether the receiver handles sync alerts.
<code>lastError</code>	Returns the error that occurred during the last sync session.
<code>session</code>	Returns the session object used to manage the sync session.
<code>setDelegate:</code>	Sets the receiver's delegate to the specified object.
<code>setHandlesSyncAlerts:</code>	Specifies whether the receiver should handle sync alerts.
<code>startAsynchronousSync:</code>	Syncs client records, specified by the data source, in a separate thread.
<code>sync</code>	Syncs client records, specified by the data source, with the sync engine.

Delegate Methods

<code>sessionDriver:didPullAndReturnError:</code>	Informs the receiver that changes were pulled from the sync engine. (required)
<code>sessionDriver:didPushAndReturnError:</code>	Informs the receiver that client changes were pushed to the sync engine. (required)
<code>sessionDriver:didRegisterClientAndReturnError:</code>	Informs the receiver that a client was registered. (required)
<code>sessionDriver:willFinishSessionAndReturnError:</code>	Informs the receiver that a session will be finished. (required)
<code>sessionDriver:willPullAndReturnError:</code>	Informs the receiver that changes will be pulled from the sync engine. (required)
<code>sessionDriver:willPushAndReturnError:</code>	Informs the receiver that client changes will be pushed to the sync engine. (required)
<code>sessionDriverDidCancelSession:</code>	Informs the receiver that a session was cancelled. (required)
<code>sessionDriverDidFinishSession:</code>	Informs the receiver that a session was finished. (required)
<code>sessionDriverWillCancelSession:</code>	Informs the receiver that a session will be cancelled. (required)

NSObject

Complete reference information is available in the `NSObject` reference.

Instance Methods

<code>attributedStringForIdentityPropertiesWithNames: inRecord:comparisonRecords:firstLineAttributes: secondLineAttributes:</code>	Provides a custom presentation of a schema to the data change alert and conflict resolver user interfaces. (required)
<code>attributedStringForPropertiesWithNames:inRecord: comparisonRecords:defaultAttributes:</code>	Provides a custom presentation of a record to the data change alert and conflict resolver user interfaces. (required)
<code>changedRecordsForEntityName:moreComing:error:</code>	Returns changed records for the given entity name that should be pushed to the sync engine during a fast sync. (required)
<code>changesForEntityName:moreComing:error:</code>	Returns the changes to records that should be pushed to the sync engine during a fast sync. (required)
<code>entityNamesToPull</code>	Returns an array of NSString objects representing the names of entities this client wants to pull. (required)
<code>entityNamesToSync</code>	Returns an array of NSString objects representing the names of entities this client wants to sync. (required)
<code>identifiersForRecordsToDeleteForEntityName: moreComing:error:</code>	Returns the record identifiers for deleted records that should be pushed to the sync engine during a fast sync. (required)
<code>lastAnchorForEntityName:</code>	Returns the last sync anchor for the specified entity name. (required)
<code>nextAnchorForEntityName:</code>	Returns the next sync anchor for the specified entity name. (required)
<code>sessionBeginTimeout</code>	Returns the time, in seconds, that the client is willing to wait for a sync session to begin. (required)
<code>sessionPullChangesTimeout</code>	Returns the time, in seconds, that the client is willing to wait for a sync session to mingle—that is, prepare to pull changes. (required)

NSPersistentStoreCoordinator

Complete reference information is available in the [NSPersistentStoreCoordinator](#) reference.

Instance Methods

<code>setStoresFastSyncDetailsAtURL:forPersistentStore:</code>	Specifies where to save details about fast syncing for a persistent store.
<code>syncWithClient:inBackground:handler:error:</code>	Syncs managed objects, stored by the receiver, with the sync engine.

Protocols

All of the protocols with new symbols are listed alphabetically, with their new methods described.

ISyncSessionDriverDataSource (New)

Complete reference information is available in the [ISyncSessionDriverDataSource](#) reference.

Instance Methods

<code>applyChange:forEntityName: remappedRecordIdentifier:formattedRecord: error:</code>	Applies the given changes to a client's record during the pulling phase of a sync session.
<code>clientDescriptionURL</code>	Returns an NSURL object representing the path to the client description property list.
<code>clientIdentifier</code>	Returns the client's unique identifier specified when registering the client.
<code>deleteAllRecordsForEntityName:error:</code>	Deletes all records for the specified entity.
<code>preferredSyncModeForEntityName:</code>	Returns the client's preferred sync mode for the session.
<code>recordsForEntityName:moreComing:error:</code>	Returns records for the given entity name that should be pushed to the sync engine during a slow sync.
<code>schemaBundleURLs</code>	Returns an array containing NSURL objects representing the path to schemas this client uses.

NSPersistentStoreCoordinatorSyncing (New)

Complete reference information is available in the [NSPersistentStoreCoordinatorSyncing](#) reference.

Instance Methods

<code>managedObjectContextsToMonitorWhenSyncing-PersistentStoreCoordinator:</code>	Returns the managed object contexts that the receiver wants to monitor during the next sync session. (required)
<code>managedObjectContextsToReloadAfterSyncing-PersistentStoreCoordinator:</code>	Returns the managed object contexts that should be reloaded after the persistent store coordinator syncs. (required)
<code>persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession:</code>	Informs the receiver that pulled changes were applied to a specific record during a sync session. (required)
<code>persistentStoreCoordinator:didCancelSyncSession:error:</code>	Informs the receiver that a session was cancelled. (required)
<code>persistentStoreCoordinator:didCommitChanges:inSyncSession:</code>	Informs the receiver that all applied changes were committed during a sync session. (required)
<code>persistentStoreCoordinator:didFinishSyncSession:</code>	Informs the receiver that a session was finished. (required)
<code>persistentStoreCoordinator:didPullChangesInSyncSession:</code>	Informs the receiver that changes were pulled from the sync engine. (required)
<code>persistentStoreCoordinator:didPushChangesInSyncSession:</code>	Informs the receiver that client changes were pushed to the sync engine. (required)
<code>persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession:</code>	Informs the receiver that pulled changes will be applied to a specific record during a sync session. (required)
<code>persistentStoreCoordinator:willDeleteRecordWithIdentifier:inSyncSession:</code>	Informs the receiver that a specific record will be deleted during the pushing phase of a sync session. (required)
<code>persistentStoreCoordinator:willPullChangesInSyncSession:</code>	Informs the receiver that changes will be pulled from the sync engine. (required)
<code>persistentStoreCoordinator:willPushChangesInSyncSession:</code>	Informs the receiver that client changes will be pushed to the sync engine. (required)
<code>persistentStoreCoordinator:willPushRecord:forManagedObject:inSyncSession:</code>	Informs the receiver that client changes to a specific record will be pushed to the sync engine. (required)
<code>persistentStoreCoordinatorShouldStartSyncing:</code>	Returns whether or not the persistent store coordinator should start syncing. (required)

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

ISyncCommon.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

H_ISYNCCOMMON

ISyncSessionDriver.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ISyncSessionDriverChangeAccepted	Indicates the client accepted the change.
ISyncSessionDriverChangeError	Indicates an error occurred while applying the change.
ISyncSessionDriverChangeIgnored	Indicates the client ignored the change.
ISyncSessionDriverChangeRefused	Indicates the client refused the change.
ISyncSessionDriverChangeResult	Specifies whether a change was applied.
ISyncSessionDriverMode	Specifies the preferred sync mode for a client.
ISyncSessionDriverModeFast	Indicates that the client wants to fast sync.
ISyncSessionDriverModeRefresh	Indicates that the client wants to refresh sync. If the preferredSyncModeForEntityName: method returns this constant, the ISyncSessionDriver object sends clientDidResetEntityNames: to the ISyncSession object.
ISyncSessionDriverModeSlow	Indicates that the client wants to slow sync. If the client slow syncs, it needs to push every record.

SyncServicesErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>ISyncErrorDomain</code>	Domain for errors that occurred using Sync Services classes and methods.
<code>ISyncSessionClientAlreadySyncingError</code>	Error code that indicates the client is already syncing.
<code>ISyncSessionDriverFatalError</code>	Error code that indicates an <code>ISyncSessionDriver</code> received a fatal error.
<code>ISyncSessionDriverPullFailureError</code>	Error code that indicates the session driver failed to pull records.
<code>ISyncSessionDriverRegistrationError</code>	Error code that indicates the session driver failed to register the client.
<code>ISyncSessionUserCanceledSessionError</code>	Error code that indicates the user canceled the sync session.

10.4 Symbol Changes

This article lists the symbols added to `SyncServices.framework` in Mac OS X v10.4.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

ISyncChange (New)

Complete reference information is available in the [ISyncChange](#) reference.

Class Methods

<code>changeWithType:recordIdentifier:changes:</code>	Creates an <code>ISyncChange</code> object.
---	---

Instance Methods

<code>changes</code>	Returns an array of changes made to the record.
<code>initWithChangeType:recordIdentifier:changes:</code>	Initializes an <code>ISyncChange</code> object.
<code>record</code>	Returns a dictionary representation of the record that changed.
<code>recordIdentifier</code>	Returns the unique record identifier for the record that changed.
<code>type</code>	Returns the type of change the receiver represents.

ISyncClient (New)

Complete reference information is available in the [ISyncClient](#) reference.

Instance Methods

<code>canPullChangesForEntityName:</code>	Returns YES if the client supports pulling changes to entity records specified by <code>entityName</code> , NO otherwise.
<code>canPushChangesForEntityName:</code>	Returns YES if the client supports pushing changes to entity records specified by <code>entityName</code> , NO otherwise.
<code>clientIdentifier</code>	Returns the client's identifier specified when registering the client.
<code>clientType</code>	Returns the receiver's client type.
<code>displayName</code>	Returns the receiver's display name specified in the client description file when registering the client or by sending <code>setDisplayName:</code> to the receiver.
<code>enabledEntityNames</code>	Returns an array of NSString objects containing the names of the entities that are enabled.
<code>filters</code>	Returns an array of filters that define a subset of the records the client syncs.
<code>imagePath</code>	Returns the absolute path to the image representation of the client.
<code>isEnabledForEntityName:</code>	Returns YES if the entity specified by <code>entityName</code> is enabled, NO otherwise.
<code>lastSyncDateForEntityName:</code>	Returns the start date of the last time an entity, specified by <code>entityName</code> , was synced.
<code>lastSyncStatusForEntityName:</code>	Returns the status of the last time an entity, specified by <code>entityName</code> was synced.
<code>objectForKey:</code>	Returns the object for key that was specified using the <code>setObject:forKey:</code> method.
<code>setDisplayName:</code>	Sets the display name for the receiver to <code>displayName</code> .
<code>setEnabled:forEntityNames:</code>	If <code>flag</code> is YES, enables the entities specified by <code>entityNames</code> , otherwise disables them.
<code>setFilters:</code>	Sets the receiver's filters used to control the records pulled from the sync engine to <code>filters</code> , an array of objects conforming to the <code>ISyncFiltering</code> protocol.

<code>setImagePath:</code>	Sets the receiver's absolute image path to path.
<code>setObject:forKey:</code>	Associates arbitrary information specified by a key-value pair to the receiver.
<code>setShouldReplaceClientRecords:forEntityNames:</code>	Sets whether or not a client should pull the truth—replace all its records for the specified entities on the next sync.
<code>setShouldSynchronize:withClientsOfType:</code>	Adds the receiver as an observer of alerts when clients of the specified type sync.
<code>setSyncAlertHandler:selector:</code>	Sets the target and action to be invoked when an observed client creates a session and begins syncing.
<code>setSyncAlertToolPath:</code>	Specifies the absolute path to a tool that is launched when an observed client creates a session and begins syncing.
<code>shouldReplaceClientRecordsForEntityName:</code>	Returns YES if the client should replace all records for the entity specified by <code>entityName</code> during the next sync, NO otherwise.
<code>shouldSynchronizeWithClientsOfType:</code>	Returns YES if the client is registered to receive alerts when clients of <code>clientType</code> sync, NO otherwise.
<code>supportedEntityNames</code>	Returns an array of NSString objects containing the names of the entities the client supports.
<code>syncAlertToolPath</code>	Returns the path to the tool that is launched when an observed client begins syncing.

ISyncFilter (New)

Complete reference information is available in the `ISyncFilter` reference.

Class Methods

<code>filterMatchingAllFilters:</code>	Returns a filter that is the logical AND of the filters specified in the filters array.
<code>filterMatchingAtLeastOneFilter:</code>	Returns a filter that is the logical OR of the filters specified in the filters array.

ISyncManager (New)

Complete reference information is available in the [ISyncManager](#) reference.

Class Methods

<code>sharedManager</code>	Returns a shared <code>ISyncManager</code> object.
----------------------------	--

Instance Methods

<code>clientWithIdentifier:</code>	Returns the sync client identified by <code>clientIdentifier</code> , or nil if not found.
<code>isEnabled</code>	Returns NO if the sync engine is disabled, YES otherwise.
<code>registerClientWithIdentifier:descriptionFilePath:</code>	Returns an existing or new sync client uniquely identified by <code>clientIdentifier</code> .
<code>registerSchemaWithBundlePath:</code>	Registers a schema property list located in a bundle at <code>bundlePath</code> .
<code>snapshotOfRecordsInTruthWithEntityNames:usingIdentifiersForClient:</code>	Returns an immutable snapshot of the records for <code>entityNames</code> from the truth database.
<code>unregisterClient:</code>	Unregisters a sync client represented by <code>client</code> .
<code>unregisterSchemaWithName:</code>	Unregisters a schema uniquely identified by <code>schemaName</code> , and removes all associated records.

ISyncRecordSnapshot (New)

Complete reference information is available in the [ISyncRecordSnapshot](#) reference.

Instance Methods

<code>recordsWithIdentifiers:</code>	Returns a dictionary containing the records.
<code>recordsWithMatchingAttributes:</code>	Returns a dictionary containing all the records that match a query.
<code>sourceIdentifiersForRelationshipName:withTargetIdentifier:</code>	Returns an array of the record identifiers belonging to the source objects of a relationship.

<code>targetIdentifiersForRelationshipName:withSourceIdentifier:</code>	Returns an array of record identifiers belonging to the target objects of a relationship.
---	---

ISyncSession (New)

Complete reference information is available in the `ISyncSession` reference.

Class Methods

<code>beginTransactionInBackgroundWithClient:entityNames:target:selector:</code>	Creates a new sync session for the specified client asynchronously.
<code>beginTransactionWithClient:entityNames:beforeDate:</code>	Creates and returns a new sync session for the specified client.
<code>cancelPreviousBeginSessionWithClient:</code>	Cancels a previous request to create a session using <code>beginTransactionInBackgroundWithClient:entityNames:target:selector:</code> for client.

Instance Methods

<code>cancelSyncing</code>	Cancels the current session.
<code>changeEnumeratorForEntityNames:</code>	Returns the object enumerator for the <code>ISyncChange</code> objects which contain all the changes the client should apply to its local data.
<code>clientAcceptedChangesForRecordWithIdentifier:formattedRecord:newRecordIdentifier:</code>	Informs the sync engine that the client has accepted the changes to the record identified by <code>recordIdentifier</code> during the pulling state.
<code>clientChangedRecordIdentifiers:</code>	Changes the record identifiers of the given records.
<code>clientCommittedAcceptedChanges</code>	Informs the sync engine that all accepted and rejected changes in the current transaction during the pulling state should be committed.
<code>clientDidResetEntityNames:</code>	Tells the sync engine to perform a refresh sync of all the records for the specified entities.

<code>clientInfoForRecordWithIdentifier:</code>	Returns a client-specific, nonprepsynchronized object that stores additional information about a record specified by <code>recordIdentifier</code> .
<code>clientLostRecordWithIdentifier: shouldReplaceOnNextSync:</code>	Tells the sync engine that a record identified by <code>recordIdentifier</code> , no longer exists on the client, and indicates whether or not it should be replaced.
<code>clientRefusedChangesForRecordWithIdentifier:</code>	Informs the sync engine during the pulling state that the client has refused to apply the changes for the record specified by <code>recordIdentifier</code> .
<code>clientWantsToPushAllRecordsForEntityNames:</code>	Forces a slow sync of all the records for the specified entities.
<code>deleteRecordWithIdentifier:</code>	Creates a delete change for the record specified by <code>recordIdentifier</code> and pushes the change to the sync engine.
<code>finishSyncing</code>	Tells the sync engine that the client is done syncing. Invoking this method closes any open transactions in the pushing or pulling states.
<code>isCancelled</code>	Returns YES if the receiver was canceled, NO otherwise.
<code>prepareToPullChangesForEntityNames:beforeDate:</code>	Moves the receiver to the mingling state and returns when the sync engine is ready for the client to begin pulling changes to the specified entities.
<code>prepareToPullChangesInBackgroundForEntityNames:target:selector:</code>	Moves the receiver to the mingling state and sends a message to a specified target when the sync engine is ready for the client to begin pulling changes to the specified entities.
<code>pushChange:</code>	Pushes changes made to a single record, specified by <code>change</code> , to the sync engine.
<code>pushChangesFromRecord:withIdentifier:</code>	Compares record to the client's previous known state of the record, identified by <code>recordIdentifier</code> , and pushes the changes to the sync engine.
<code>setClientInfo:forRecordWithIdentifier:</code>	Associates a client-specific, non synchronized object, <code>clientInfo</code> , to a record specified by <code>recordIdentifier</code> .

<code>shouldPullChangesForEntityName:</code>	Returns YES if the client should pull changes to records for <code>entityName</code> , NO otherwise.
<code>shouldPushAllRecordsForEntityName:</code>	Returns YES if the client should push all the records for <code>entityName</code> to the sync engine; otherwise, NO.
<code>shouldPushChangesForEntityName:</code>	Returns YES if the client should push changes to records for <code>entityName</code> since the last sync, NO otherwise.
<code>shouldReplaceAllRecordsOnClientForEntityName:</code>	Returns YES if the client should delete all the records for the entity, specified by <code>entityName</code> , and replace them with records pulled from the sync engine, NO otherwise.
<code>snapshotOfRecordsInTruth</code>	Returns an immutable snapshot of the records in the truth database.

Protocols

All of the protocols with new symbols are listed alphabetically, with their new methods described.

ISyncFiltering (New)

Complete reference information is available in the `ISyncFiltering` reference.

Instance Methods

<code>isEqual:</code>	Returns YES if the receiver and another filter are equal, NO otherwise. (required)
<code>shouldApplyRecord:withRecordIdentifier:</code>	Returns YES if the client should pull record uniquely identified by <code>recordIdentifier</code> , NO otherwise. (required)
<code>supportedEntityNames</code>	Returns an array of entity names that this filter supports. (required)

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

ISyncChange.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ISyncChangePropertyActionKey	Specifies whether or not the property is being set or deleted. The value for this key should be either ISyncChangePropertySet or ISyncChangePropertyClear described below.
ISyncChangePropertyClear	Indicates the property was deleted.
ISyncChangePropertyNameKey	Key for the name of the property.
ISyncChangePropertySet	Indicates the property was modified.
ISyncChangePropertyValueKey	Key for the new value of the property.
ISyncChangeType	The type of change.
ISyncChangeTypeAdd	Indicates a record was added.
ISyncChangeTypeDelete	Indicates a record was deleted.
ISyncChangeTypeModify	Indicates a record was modified.

ISyncClient.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ISyncClientTypeApplication	Indicates the client is an application, such as Mail or iCal.
ISyncClientTypeDevice	Indicates the client is used to sync a device such as a phone or an iPod.
ISyncClientTypePeer	Indicates the client is a peer, such as another computer.
ISyncClientTypeServer	Indicates the client is used to sync a remote server such as MobileMe.
ISyncStatus	The following constants are returned by the lastSyncStatusForEntityName: method to indicate the state of the last sync session.
ISyncStatusCancelled	Indicates the last sync was canceled.

ISyncStatusErrors	Indicates the last sync resulted in errors.
ISyncStatusFailed	Indicates the last sync failed to complete (for example, the client crashed).
ISyncStatusNever	Indicates the client has never synced.
ISyncStatusRunning	Indicates the client is syncing.
ISyncStatusSuccess	Indicates the last sync was successful.
ISyncStatusWarnings	Indicates the last sync resulted in warnings.

ISyncCommon.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

SYNCSERVICES_EXPORT

ISyncManager.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

ISyncAvailabilityChangedNotification	Posted by the distributed notification center when syncing is enabled or disabled.
ISyncServerUnavailableException	A string aggregating the name, reason, and user info from the originating exception. Thrown by any ISyncManager method when communication to the server is lost.

ISyncSession.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

10.4 Symbol Changes

<code>ISyncInvalidEntityException</code>	Thrown if a client tries creating a session with an entity that does not exist.
<code>ISyncInvalidRecordException</code>	Thrown if a client pushes a malformed record.
<code>ISyncInvalidRecordIdentifiersKey</code>	An array of the record identifiers that raised the exception.
<code>ISyncInvalidRecordReasonsKey</code>	A dictionary where keys are the invalid record identifies and the values are the reasons for the exception.
<code>ISyncInvalidRecordsKey</code>	A dictionary where the keys are the invalid record identifiers and the values are the property keys that raised the exception.
<code>ISyncRecordEntityNameKey</code>	Used in a record pushed to the sync engine.
<code>ISyncSessionCancelledException</code>	Thrown by any method if invoked after the session was canceled.
<code>ISyncSessionUnavailableException</code>	Thrown if a session cannot be created, for example, if a client is already syncing.
<code>ISyncUnsupportedEntityException</code>	Thrown if a client tries creating a session with an entity that exists but the client does not support.

Document Revision History

This table describes the changes to *Sync Services Reference Update*.

Date	Notes
2008-04-29	Updated with the symbols added to the Sync Services framework in Mac OS X v10.6.
2007-07-18	Updated with the symbols added to the Sync Services framework in Mac OS X v10.5.
2005-04-29	Minor editorial corrections throughout.
	New document that summarizes the symbols added to the Sync Services framework in Mac OS X v10.4.

