NSPersistentStoreCoordinatorSyncing Protocol Reference

Cocoa > Syncing

2007-07-11

Apple Inc.
© 2007 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, Cocoa, 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

NSPersistentStoreCoordinatorSyncing Protocol Reference 5

Overview 5
Tasks 5
  Getting Managed Contexts 5
  Controlling Sync Behavior 5
Instance Methods 6
  managedObjectContextsToMonitorWhenSyncingPersistentStoreCoordinator: 6
  managedObjectContextsToReloadAfterSyncingPersistentStoreCoordinator: 7
  persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession: 7
  persistentStoreCoordinator:didCancelSyncSession:error: 8
  persistentStoreCoordinator:didCommitChanges:inSyncSession: 8
  persistentStoreCoordinator:didFinishSyncSession: 9
  persistentStoreCoordinator:didPullChangesInSyncSession: 9
  persistentStoreCoordinator:didPushChangesInSyncSession: 10
  persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession: 10
  persistentStoreCoordinator:willDeleteRecordWithIdentifier:inSyncSession: 11
  persistentStoreCoordinator:willPullChangesInSyncSession: 11
  persistentStoreCoordinator:willPushChangesInSyncSession: 12
  persistentStoreCoordinator:willPushRecord:forManagedObject:inSyncSession: 12
  persistentStoreCoordinatorShouldStartSyncing: 13

Document Revision History 15

Index 17

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

NSPersistentStoreCoordinatorSyncing Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/SyncServices.framework
Availability	Available in Mac OS X v10.5 and later.
Companion guide	Sync Services Programming Guide
Declared in	ISyncCoreData.h

Overview

The `NSPersistentStoreCoordinatorSyncing` protocol defines callback messages that are sent to a sync handler while a Core Data application syncs. You set a sync handler when you start a sync session using the `syncWithClient:inBackground:handler:error:` `NSPersistentStoreCoordinator` method. The callback messages defined in this protocol are sent before and after most phases of a sync session. A sync handler may implement these optional methods to customize the behavior of sync sessions. For example, a sync handler might change records before their pushed to the sync engine or verify changes pulled from the sync engine before they are applied to managed objects.

Read Syncing Core Data Applications in *Sync Services Programming Guide* for more details on using Core Data sync.

Tasks

Getting Managed Contexts

`- managedObjectContextsToMonitorWhenSyncingPersistentStoreCoordinator:` (page 6)
Returns the managed object contexts that the receiver wants to monitor during the next sync session.

`- managedObjectContextsToReloadAfterSyncingPersistentStoreCoordinator:` (page 7)
Returns the managed object contexts that should be reloaded after the persistent store coordinator syncs.

Controlling Sync Behavior

`- persistentStoreCoordinatorShouldStartSyncing:` (page 13)
Returns whether or not the persistent store coordinator should start syncing.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

- `persistentStoreCoordinator:willPushChangesInSyncSession:` (page 12)
  Informs the receiver that client changes will be pushed to the sync engine.
- `persistentStoreCoordinator:didPushChangesInSyncSession:` (page 10)
  Informs the receiver that client changes were pushed to the sync engine.
- `persistentStoreCoordinator:willPullChangesInSyncSession:` (page 11)
  Informs the receiver that changes will be pulled from the sync engine.
- `persistentStoreCoordinator:didPullChangesInSyncSession:` (page 9)
  Informs the receiver that changes were pulled from the sync engine.
- `persistentStoreCoordinator:didFinishSyncSession:` (page 9)
  Informs the receiver that a session was finished.
- `persistentStoreCoordinator:didCancelSyncSession:error:` (page 8)
  Informs the receiver that a session was cancelled.
- `persistentStoreCoordinator:willPushRecord:forManagedObject:inSyncSession:` (page 12)
  Informs the receiver that client changes to a specific record will be pushed to the sync engine.
- `persistentStoreCoordinator:willDeleteRecordWithIdentifier:inSyncSession:` (page 11)
  Informs the receiver that a specific record will be deleted during the pushing phase of a sync session.
- `persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession:` (page 10)
  Informs the receiver that pulled changes will be applied to a specific record during a sync session.
- `persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession:` (page 7)
  Informs the receiver that pulled changes were applied to a specific record during a sync session.
- `persistentStoreCoordinator:didCommitChanges:inSyncSession:` (page 8)
  Informs the receiver that all applied changes were committed during a sync session.

Instance Methods

managedObjectContextsToMonitorWhenSyncingPersistentStoreCoordinator:

Returns the managed object contexts that the receiver wants to monitor during the next sync session.

```
- (NSArray
    *)managedObjectContextsToMonitorWhenSyncingPersistentStoreCoordinator:(NSPersistentStoreCoordinator
    *)coordinator
```

Parameters

coordinator
  The persistent store coordinator being synced.

Return Value
An array containing the managed object contexts to monitor.

Discussion
The sync session uses this method to determine if pulled changes should be applied. Pulled changes are ignored—not applied to a record—if any of the managed contexts, returned by this method, changed the same record. In this case, the local changes are pushed in the next sync session and the sync engine is responsible for resolving any conflicts.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Conflicts can result if clients are allowed to change managed objects during a sync session or editing is not disabled during syncing. For example, conflicts can result if the user changes a managed object while the sync session is pulling changes to the same managed object. However, implementing this method does not handle all types of conflicts. The user may still modify a managed object after a sync session applies changes and before a sync session finishes. To avoid this, you should either not allow editing during a sync session, or be prepared to merge local changes with pulled changes after a sync session.

Therefore, although this method is optional, not implementing this method increases the risk of conflicts unless you sync synchronously or disable editing when syncing.

Availability
Available in Mac OS X v10.5 and later.

See Also
- managedObjectContextsToReloadAfterSyncingPersistentStoreCoordinator: (page 7)

Declared In
ISyncCoreData.h

managedObjectContextsToReloadAfterSyncingPersistentStoreCoordinator:

Returns the managed object contexts that should be reloaded after the persistent store coordinator syncs.

```
- (NSArray
    *)managedObjectContextsToReloadAfterSyncingPersistentStoreCoordinator:(NSPersistentStoreCoordinator
     *)coordinator
```

Parameters

coordinator
The persistent store coordinator being synced.

Return Value
An array containing the managed object contexts to reload.

Discussion
If you do not implement this method, it is your responsibility to reload managed object contexts after a sync.

Availability
Available in Mac OS X v10.5 and later.

See Also
- managedObjectContextsToMonitorWhenSyncingPersistentStoreCoordinator: (page 6)

Declared In
ISyncCoreData.h

persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession:

Informs the receiver that pulled changes were applied to a specific record during a sync session.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didApplyChange:(ISyncChange *)change toManagedObject:(NSManagedObject
    *)managedObject inSyncSession:(ISyncSession *)session
```


2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Parameters

coordinator
The persistent store coordinator being synced.

change
The changes that was applied.

managedObject
The managed object that corresponds to the changes.

session
The sync session that applied the changes.

Availability
Available in Mac OS X v10.5 and later.

See Also
- persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession: (page 10)

Declared In
ISyncCoreData.h

persistentStoreCoordinator:didCancelSyncSession:error:

Informs the receiver that a session was cancelled.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didCancelSyncSession:(ISyncSession *)session error:(NSError *)error
```

Parameters

coordinator
The persistent store coordinator being synced.

session
The sync session that was cancelled.

error
Describes the error that caused the cancellation.

Availability
Available in Mac OS X v10.5 and later.

See Also
- persistentStoreCoordinator:didFinishSyncSession: (page 9)

Declared In
ISyncCoreData.h

persistentStoreCoordinator:didCommitChanges:inSyncSession:

Informs the receiver that all applied changes were committed during a sync session.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didCommitChanges:(NSDictionary *)changes inSyncSession:(ISyncSession *)session
```


2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Parameters

coordinator

The persistent store coordinator being synced.

changes

A dictionary containing the changes to possibly multiple objects that were applied and committed. The dictionary contains the following keys: `NSInsertedObjectsKey`, `NSUpdatedObjectsKey`, and `NSDeletedObjectsKey`.

session

The sync session that committed the changes.

Discussion

Typically, this method is invoked after the persistent store changes are saved and the sync session receives the `clientCommittedAcceptedChanges` message. This method can be invoked multiple times during a sync session.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`ISyncCoreData.h`

persistentStoreCoordinator:didFinishSyncSession:

Informs the receiver that a session was finished.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didFinishSyncSession:(ISyncSession *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

session

The sync session that finished.

Availability

Available in Mac OS X v10.5 and later.

See Also

`- persistentStoreCoordinator:didCancelSyncSession:error:` (page 8)

Declared In

`ISyncCoreData.h`

persistentStoreCoordinator:didPullChangesInSyncSession:

Informs the receiver that changes were pulled from the sync engine.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didPullChangesInSyncSession:(ISyncSession *)session
```


2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Parameters

coordinator

The persistent store coordinator being synced.

session

The sync session that pulled the changes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `persistentStoreCoordinator:willPullChangesInSyncSession:` (page 11)

Declared In

`ISyncCoreData.h`

persistentStoreCoordinator:didPushChangesInSyncSession:

Informs the receiver that client changes were pushed to the sync engine.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    didPushChangesInSyncSession:(ISyncSession *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

session

The sync session that pushed the changes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `persistentStoreCoordinator:willPushChangesInSyncSession:` (page 12)

Declared In

`ISyncCoreData.h`

persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession:

Informs the receiver that pulled changes will be applied to a specific record during a sync session.

```
- (ISyncChange *)persistentStoreCoordinator:(NSPersistentStoreCoordinator
    *)coordinator willApplyChange:(ISyncChange *)change
    toManagedObject:(NSManagedObject *)managedObject inSyncSession:(ISyncSession
    *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

change

The changes that will be applied. An `ISyncChange` object can represent a delete record change, as well as an insert and update record change.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

managedObject
The managed object that corresponds to the changes.

session
The sync session that is applying the changes.

Return Value
The change to apply. `nil` if you do not want to apply this change.

Discussion
Implement this method if you want to modify a change before it is applied.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession:` (page 7)

Declared In
`ISyncCoreData.h`

persistentStoreCoordinator:willDeleteRecordWithIdentifier:inSyncSession:

Informs the receiver that a specific record will be deleted during the pushing phase of a sync session.

```
- (BOOL)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    willDeleteRecordWithIdentifier:(NSString *)identifier inSyncSession:(ISyncSession
     *)session
```

Parameters
coordinator
The persistent store coordinator being synced.

identifier
The identifier for the record that will be deleted.

session
The sync session that is pushing records.

Return Value
`YES` to delete the record; otherwise, `NO`.

Discussion
Implement this method if you want to verify if a record should be deleted before it is deleted.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`ISyncCoreData.h`

persistentStoreCoordinator:willPullChangesInSyncSession:

Informs the receiver that changes will be pulled from the sync engine.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    willPullChangesInSyncSession:(ISyncSession *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

session

The sync session that is pulling the changes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- persistentStoreCoordinator:didPullChangesInSyncSession: (page 9)

Declared In

ISyncCoreData.h

persistentStoreCoordinator:willPushChangesInSyncSession:

Informs the receiver that client changes will be pushed to the sync engine.

```
- (void)persistentStoreCoordinator:(NSPersistentStoreCoordinator *)coordinator
    willPushChangesInSyncSession:(ISyncSession *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

session

The sync session that is pushing the changes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- persistentStoreCoordinator:didPushChangesInSyncSession: (page 10)

Declared In

ISyncCoreData.h

persistentStoreCoordinator:willPushRecord:forManagedObject:inSyncSession:

Informs the receiver that client changes to a specific record will be pushed to the sync engine.

```
- (NSDictionary *)persistentStoreCoordinator:(NSPersistentStoreCoordinator
    *)coordinator willPushRecord:(NSDictionary *)record
    forManagedObject:(NSManagedObject *)managedObject inSyncSession:(ISyncSession
     *)session
```

Parameters

coordinator

The persistent store coordinator being synced.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

record
The record that will be pushed.

managedObject
The managed object that corresponds to the record.

session
The sync session that is pushing records.

Return Value
The record to push. `nil` if you do not want to push the record.

Discussion
Implement this method if you want to modify a record before pushing it.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`ISyncCoreData.h`

persistentStoreCoordinatorShouldStartSyncing:

Returns whether or not the persistent store coordinator should start syncing.

```
- (BOOL)persistentStoreCoordinatorShouldStartSyncing:(NSPersistentStoreCoordinator
    *)coordinator
```

Parameters

coordinator
The persistent store coordinator being synced.

Return Value
`YES` if the persistent store coordinator can start syncing; otherwise, `NO`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`ISyncCoreData.h`

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Document Revision History

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

Date	Notes
2007-07-11	New document that describes a protocol that you use to change the behavior of a Core Data application sync session.

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Document Revision History

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.

Index

M

managedObjectContextsToMonitorWhenSyncing-PersistentStoreCoordinator: protocol instance method 6
managedObjectContextsToReloadAfterSyncing-PersistentStoreCoordinator: protocol instance method 7

P

persistentStoreCoordinator:didApplyChange:toManagedObject:inSyncSession: protocol instance method 7
persistentStoreCoordinator:didCancelSyncSession:error: protocol instance method 8
persistentStoreCoordinator:didCommitChanges:inSyncSession: protocol instance method 8
persistentStoreCoordinator:didFinishSyncSession: protocol instance method 9
persistentStoreCoordinator:didPullChangesInSyncSession: protocol instance method 9
persistentStoreCoordinator:didPushChangesInSyncSession: protocol instance method 10
persistentStoreCoordinator:willApplyChange:toManagedObject:inSyncSession: protocol instance method 10
persistentStoreCoordinator:willDeleteRecordWithIdentifier:inSyncSession: protocol instance method 11
persistentStoreCoordinator:willPullChangesInSyncSession: protocol instance method 11
persistentStoreCoordinator:willPushChangesInSyncSession: protocol instance method 12
persistentStoreCoordinator:willPushRecord:forManagedObject:inSyncSession: protocol instance method 12
persistentStoreCoordinatorShouldStartSyncing: protocol instance method 13

2007-07-11 | © 2007 Apple Inc. All Rights Reserved.