ISyncManager Class Reference

Cocoa > Syncing

2007-07-11

Contents

ISyncManager Class Reference

Inherits from   NSObject

Conforms to   NSObject (NSObject)

Framework   /System/Library/Frameworks/SyncServices.framework

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

Companion guide   Sync Services Programming Guide

Declared in   ISyncManager.h

Related sample code   People
                                   SeeMyFriends
                                   StickiesExample

Overview

You use an ISyncManager object to communicate directly with the sync engine to perform administrative operations. A client must register itself with an ISyncManager object before it can sync its data. If a client is not using an existing schema, it must register the schema before it registers itself. You also use an ISyncManager to look up an existing client and unregister a client.

There's only one ISyncManager instance per client process you obtain using the `sharedManager` (page 7) class method. You should never instantiate or subclass ISyncManager directly.

Sync Services provides three canonical schemas: `Bookmarks.syncschema`, `Contacts.syncschema`, and `Calendars.syncschema`. If you want to extend one of these existing schemas or define your own schema, then you need to register that schema with the shared ISyncManager object. You use the `registerSchemaWithBundlePath:` (page 9) method to register a schema with the sync engine or update an existing schema. Occasionally, you might use `unregisterSchemaWithName:` (page 11) to remove a schema and all associated records. Removing a schema impacts every client that uses that schema. Typically, you just register a schema once and reregister it when it changes.

You also use the shared ISyncManager object to create and register a sync client—that is, an instance of ISyncClient—with a unique identifier that you specify. Use `registerClientWithIdentifier:descriptionFilePath:` (page 8) to register your client—this method returns either a new client object or an existing client. You also use this method to describe the capabilities of the client—for example, describe what entities and properties the client supports. You use the `unregisterClient:` (page 11) method to unregister a client. See *Sync Services Programming Guide* for more information on registering and unregistering clients.

Tasks

Getting the Default Manager

+ sharedManager (page 7)

> Returns a shared ISyncManager object.

Getting a Manager's State

– isEnabled (page 8)

> Returns NO if the sync engine is disabled, YES otherwise.

Registering Schemas

– registerSchemaWithBundlePath: (page 9)

> Registers a schema property list located in a bundle at *bundlePath*.

– unregisterSchemaWithName: (page 11)

> Unregisters a schema uniquely identified by *schemaName*, and removes all associated records.

Registering Clients

– clientWithIdentifier: (page 8)

> Returns the sync client identified by *clientIdentifier*, or nil if not found.

– registerClientWithIdentifier:descriptionFilePath: (page 8)

> Returns an existing or new sync client uniquely identified by *clientIdentifier*.

– unregisterClient: (page 11)

> Unregisters a sync client represented by *client*.

Getting Snapshots

– snapshotOfRecordsInTruthWithEntityNames:usingIdentifiersForClient: (page 10)

> Returns an immutable snapshot of the records for *entityNames* from the truth database.

Using Sync Alert Handlers

– addRequestMode: (page 7)

> Adds a mode to the set of run-loop input modes that the receiver uses for connection requests.

– removeRequestMode: (page 10)

> Removes a mode from the set of run-loop input modes the receiver uses for connection requests.

– requestModes (page 10)

> Returns the set of request modes the receiver registers with its NSRunLoop object.

Class Methods

sharedManager

Returns a shared ISyncManager object.

```
+ (ISyncManager *)sharedManager
```

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

Related Sample Code
People
SeeMyFriends
StickiesExample

Declared In
`ISyncManager.h`

Instance Methods

addRequestMode:

Adds a mode to the set of run-loop input modes that the receiver uses for connection requests.

```
- (void)addRequestMode:(NSString *)mode
```

Parameters
mode

    The mode to add to the receiver. See *NSRunLoop Class Reference* for more information on input modes.

Discussion
Clients that register sync alert handlers may use this method to manage the request modes of connections that are sent alerts by the sync engine. This method is similar to the `addRequestMode:` method of NSConnection. For example, a client that registers a sync alert handler in a process that might present a modal dialog to the user, should add the appropriate request mode to the run-loop, so alerts can be handled in a timely manner even when the application is blocked for user input.

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

See Also
- `removeRequestMode:` (page 10)
- `requestModes` (page 10)
- `addRequestMode:` (NSConnection)

Declared In
`ISyncManager.h`

clientWithIdentifier:

Returns the sync client identified by *clientIdentifier*, or `nil` if not found.

    - (ISyncClient *)clientWithIdentifier:(NSString *)clientIdentifier

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

See Also
- registerClientWithIdentifier:descriptionFilePath: (page 8)
- unregisterClient: (page 11)

Related Sample Code
SeeMyFriends
StickiesExample

Declared In
ISyncManager.h

isEnabled

Returns `NO` if the sync engine is disabled, `YES` otherwise.

    - (BOOL)isEnabled

Discussion
You should not begin a sync session when this method returns `NO`. However, you can register for the ISyncAvailabilityChangedNotification (page 12) notification, which is sent when the sync engine state changes.

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

Declared In
ISyncManager.h

registerClientWithIdentifier:descriptionFilePath:

Returns an existing or new sync client uniquely identified by *clientIdentifier*.

    - (ISyncClient *)registerClientWithIdentifier:(NSString *)clientIdentifier
        descriptionFilePath:(NSString *)descriptionFilePath

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

The client description file located at *descriptionFilePath* is a property list that specifies client information that the sync engine needs to know to sync its records. For example, the client description file a list of the client supported entities and properties. See *Sync Services Programming Guide* for a complete description of the client description file.

If the client already exists, then invoking this method updates the client description. If the set of supported entities and properties changes, the sync engine may force the client to slow sync the next time it syncs. This can be expensive, so only reregister a client if necessary.

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

See Also
- `clientWithIdentifier:` (page 8)
- `unregisterClient:` (page 11)
- `canPullChangesForEntityName:` (ISyncClient)
- `canPushChangesForEntityName:` (ISyncClient)
- `displayName` (ISyncClient)
- `imagePath` (ISyncClient)

Related Sample Code
People
SeeMyFriends
StickiesExample

Declared In
`ISyncManager.h`

registerSchemaWithBundlePath:

Registers a schema property list located in a bundle at *bundlePath*.

- `(BOOL)registerSchemaWithBundlePath:(NSString *)bundlePath`

Discussion
The schema can define new entities and properties, and extend existing entities. The schema bundle may contain other files, such as images and localization files. See *Sync Services Programming Guide* for more details on the schema format and contents of the schema bundle.

If a schema of the same name exists, invoking this method updates that schema. Consequently, records and properties of records may be removed if an entity or property is removed from the schema. This action may cause clients that use this schema to slow sync the next time they sync. This process can be expensive, so reregister a schema only if necessary.

Returns `YES` if successful, `NO` otherwise.

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

See Also
- `unregisterSchemaWithName:` (page 11)

Related Sample Code
StickiesExample

Declared In
`ISyncManager.h`

removeRequestMode:

Removes a mode from the set of run-loop input modes the receiver uses for connection requests.

```
(void)removeRequestMode:(NSString *)mode
```

Parameters

mode

> The mode to remove. See *NSRunLoop Class Reference* for more information on input modes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- addRequestMode: (page 7)
- requestModes (page 10)
- removeRequestMode: (NSConnection)

Declared In

ISyncManager.h

requestModes

Returns the set of request modes the receiver registers with its NSRunLoop object.

```
(NSArray *)requestModes
```

Return Value

An array of `NSString` objects that represents the set of request modes that the receiver registers. See *NSRunLoop Class Reference* for more information on input modes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- addRequestMode: (page 7)
- removeRequestMode: (page 10)
- requestModes (NSConnection)

Declared In

ISyncManager.h

snapshotOfRecordsInTruthWithEntityNames:usingIdentifiersForClient:

Returns an immutable snapshot of the records for *entityNames* from the truth database.

```
- (ISyncRecordSnapshot *)snapshotOfRecordsInTruthWithEntityNames:(NSArray
    *)entityNames usingIdentifiersForClient:(ISyncClient *)client
```

Discussion
The truth database stores a copy of all the synced records and contains the amalgamation of all entities and properties from all clients. The snapshot is made of the records for entities specified by the *entityNames* argument, an array of NSString objects containing the names of entities. You access the records by sending messages to the returned ISyncRecordSnapshot object.

Each client has its own name space for record identifiers. The *client* argument specifies the name space you want to use. If *client* is `nil` or invalid, the record identifiers from the sync engine's global name space are used.

The snapshot is an immutable copy of the records taken at the time returned object is created. If the truth database is subsequently modified, the changes are not be reflected in the snapshot. You should create a new snapshot if you want up-to-date records.

Do not use this method if you are syncing and want a snapshot that is consistent with the sync session. Another client may be pushing changes that you have not pulled yet. Instead, you can use the ISyncSession `snapshotOfRecordsInTruth` method to get the state of a session.

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

See Also
– clientWithIdentifier: (page 8)

Related Sample Code
SeeMyFriends

Declared In
ISyncManager.h

unregisterClient:

Unregisters a sync client represented by *client*.

- (void)**unregisterClient:**(ISyncClient *)*client*

Discussion
Does nothing if *client* is not registered.

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

See Also
– clientWithIdentifier: (page 8)
– registerClientWithIdentifier:descriptionFilePath: (page 8)

Declared In
ISyncManager.h

unregisterSchemaWithName:

Unregisters a schema uniquely identified by *schemaName*, and removes all associated records.

```
- (void)unregisterSchemaWithName:(NSString *)schemaName
```

Discussion
This action causes clients that use this schema to slow sync the next time they sync. This can be expensive and results in the loss of data, so only unregister a schema if necessary. This method does nothing if the schema is not registered.

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

See Also
- registerSchemaWithBundlePath: (page 9)

Declared In
ISyncManager.h

Constants

The following constant is thrown whenever the connection with the sync engine is lost.

Constant	Description
ISyncServer-UnavailableException	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.

Notifications

ISyncAvailabilityChangedNotification

Posted by the distributed notification center when syncing is enabled or disabled. The notification object is an NSString equal to "YES" if enabled and "NO" if disabled. The receiver should still invoke isEnabled (page 8) before beginning a sync session. This notification does not contain a userInfo dictionary.

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

Declared In
ISyncManager.h

Document Revision History

This table describes the changes to *ISyncManager Class Reference*.

Date	Notes
2007-07-11	First publication of this content as a separate document.

Index

A

`addRequestMode:` instance method 7

C

`clientWithIdentifier:` instance method 8

I

`isEnabled` instance method 8
`ISyncAvailabilityChangedNotification` notification 12
`ISyncServerUnavailableException` constant 12

R

`registerClientWithIdentifier:descriptionFilePath:` instance method 8
`registerSchemaWithBundlePath:` instance method 9
`removeRequestMode:` instance method 10
`requestModes` instance method 10

S

`sharedManager` class method 7
`snapshotOfRecordsInTruthWithEntityNames: usingIdentifiersForClient:` instance method 10

U

`unregisterClient:` instance method 11
`unregisterSchemaWithName:` instance method 11