NSPersistentDocument Class Reference

Cocoa > Data Management

2008-02-08

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

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

NSPersistentDocument Class Reference

Inherits from	NSDocument : NSObject
Conforms to	NSUserInterfaceValidations (NSDocument) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	NSPersistentDocument.h
Companion guides	NSPersistentDocument Core Data Tutorial Document-Based Applications Overview Core Data Programming Guide
Related sample code	Departments and Employees DerivedProperty File Wrappers with Core Data Documents QTMetadataEditor TwoManyControllers

Overview

The `NSPersistentDocument` class is a subclass of `NSDocument` that is designed to easily integrate into the Core Data framework. It provides methods to access a document-wide `NSManagedObjectContext` object, and provides default implementations of methods to read and write files using the persistence framework. In a persistent document, the undo manager functionality is taken over by managed object context.

Standard document behavior is implemented as follows:

- Opening a document invokes `configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7) with the new URL, and adds a store of the default type (XML). Objects are loaded from the persistent store on demand through the document's context.
- Saving a new document adds a store of the default type with the chosen URL and invokes save: on the context. For an existing document, a save just invokes `save:` on the context.
- Save As for a new document, simply invokes save. For an opened document, migrates the persistent store to the new URL, and invokes `save:` on the context.
- Revert resets the document's managed object context. Objects are subsequently loaded from the persistent store on demand, as with opening a new document.

Note that NSPersistentDocument does not support some standard document behavior, in particular NSPersistentDocument does not support file wrappers. “Save To…” and Autosave are not directly supported—Core Data cannot save to a store and maintain the same changed state in the managed object context, all the while maintaining an unsaved stack as the current document.

By default an NSPersistentDocument instance creates its own ready-to-use persistence stack including managed object context, persistent object store coordinator and persistent store. There is a one-to-one mapping between the document and the backing object store.

You can customize the architecture of the persistence stack by overriding the methods managedObjectModel (page 9) and configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error: (page 7). You might wish to do this, for example, to specify a particular managed object model.

Tasks

Managing the Persistence Objects

– managedObjectContext (page 8)
    Returns the managed object context for the receiver.

– managedObjectModel (page 9)
    Returns the receiver’s managed object model.

– setManagedObjectContext: (page 12)
    Sets the receiver’s managed object context.

– configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error: (page 7)
    Configures the receiver’s persistent store coordinator with the appropriate stores for a given URL.

– persistentStoreTypeForFileType: (page 10)
    Returns the type of persistent store associated with the specified file type.

Undo Support

– hasUndoManager (page 8)
    Returns YES.

– setHasUndoManager: (page 12)
    Overridden to be a no-op.

– setUndoManager: (page 12)
    Overridden to be a no-op.

– isDocumentEdited (page 8)
    Returns a Boolean value that indicates whether the receiver’s managed object context, or editors registered with the context, have uncommitted changes.

Document Content Management

- readFromURL:ofType:error: (page 10)

  Sets the contents of the receiver by reading from a file of a given type located by a given URL.

- revertToContentsOfURL:ofType:error: (page 11)

  Overridden to clean up the managed object context and controllers during a revert.

- writeToURL:ofType:forSaveOperation:originalContentsURL:error: (page 13)

  Saves changes in the document's managed object context and saves the document's persistent store to a given URL.

Deprecated

- configurePersistentStoreCoordinatorForURL:ofType:error: (page 15) Deprecated in Mac OS X v10.5

  Configures the receiver's persistent store coordinator for a given URL and document type. (Deprecated. Use configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error: (page 7) instead.)

Instance Methods

configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:

Configures the receiver's persistent store coordinator with the appropriate stores for a given URL.

```
- (BOOL)configurePersistentStoreCoordinatorForURL:(NSURL *)url
    ofType:(NSString *)fileType
    modelConfiguration:(NSString *)configuration
    storeOptions:(NSDictionary *)storeOptions
    error:(NSError **)error
```

Parameters

url

An URL that specifies the location of the document's store.

fileType

The document type.

configuration

The name of the managed object model configuration to use. (The managed object model is associated with the persistent store coordinator.) Pass nil if you do not want to specify a configuration.

storeOptions

Options for the store. See "Store Options" in NSPersistentStoreCoordinator for possible values.

error

If the method does not complete successfully, upon return contains an NSError object that describes the problem.

Return Value
YES if the method completes successfully, otherwise NO.

Discussion
This method is invoked automatically when an existing document is opened. You override this method to customize creation of a persistent store for a given document or store type. You can retrieve the persistent store coordinator with the following code:

```
[[self managedObjectContext] persistentStoreCoordinator];
```

You can override this method to create the store to save to or load from (invoked from within the other NSDocument methods to read/write files), which gives developers the ability to load/save from/to different persistent store types (default type is XML).

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

Declared In
NSPersistentDocument.h

hasUndoManager

Returns YES.

```
- (BOOL)hasUndoManager
```

Return Value
YES.

Special Considerations

You should not override this method.

See Also
- managedObjectContext (page 8)

isDocumentEdited

Returns a Boolean value that indicates whether the receiver’s managed object context, or editors registered with the context, have uncommitted changes.

```
- (BOOL)isDocumentEdited
```

Return Value
YES if the receiver’s managed object context, or editors registered with the context, have uncommitted changes, otherwise NO.

See Also
- managedObjectContext (page 8)

managedObjectContext

Returns the managed object context for the receiver.

```
- (NSManagedObjectContext *)managedObjectContext
```

Return Value
The managed object context for the receiver.

Discussion
If a managed object context for the receiver does not exist, one is created automatically. You override this method to customize the creation of the persistence stack.

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

See Also
`- managedObjectModel` (page 9)

Related Sample Code
Departments and Employees

QTMetadataEditor

Declared In
`NSPersistentDocument.h`

managedObjectModel

Returns the receiver's managed object model.

```
- (id)managedObjectModel
```

Return Value
The receiver's managed object model, used to configure the receiver's persistent store coordinator.

Discussion
By default the Core Data framework creates a merged model from all models in the application bundle (`[NSBundle mainBundle]`). You can override this method to return a specific model to use to create persistent stores. A typical implementation might include code similar to the following fragment:

```
NSBundle *bundle = [NSBundle bundleForClass:[self class]];
NSString *path = [bundle pathForResource:@"MyModel" ofType:@"mom"];
NSURL *url = [NSURL fileURLWithPath:path];
NSManagedObjectModel *model = [[NSManagedObjectModel alloc]
initWithContentsOfURL:url];
```

Normally you would cache the model as an instance variable. If all your document instances use the same model, however, you can increase the efficiency of this method by caching a single instance, as illustrated in the following example.

```
- (id)managedObjectModel {
    static id sharedModel = nil;
    if (sharedModel == nil) {
        sharedModel = [[super managedObjectModel] retain];
    }
    return sharedModel;
}
```

Special Considerations

In applications built on Mac OS X v10.4, by default the Core Data framework creates a merged model from all the models found in the application bundle *and the frameworks against which the application is linked*.

Availability

Available in Mac OS X v10.4 and later.

See Also

`- managedObjectContext` (page 8)

Declared In

`NSPersistentDocument.h`

persistentStoreTypeForFileType:

Returns the type of persistent store associated with the specified file type.

```
- (NSString *)persistentStoreTypeForFileType:(NSString *)fileType
```

Parameters

fileType

A document file type.

Return Value

The type of persistent store associated with *fileType*. For possible values, see `NSPersistentStoreCoordinator`.

Discussion

You set the persistent store type in the application's property list (see Storing Document Types Information in a Property List).

Availability

Available in Mac OS X v10.4 and later.

See Also

`- configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7)

Declared In

`NSPersistentDocument.h`

readFromURL:ofType:error:

Sets the contents of the receiver by reading from a file of a given type located by a given URL.

```
- (BOOL)readFromURL:(NSURL *)absoluteURL
    ofType:(NSString *)typeName
    error:(NSError **)outError
```

Parameters

absoluteURL

An URL that specifies the location from which to read the document.

typeName

The document type at *absoluteURL*.

outError

If *absoluteURL* is not valid, or the store at *absoluteURL* cannot be read, upon return contains an `NSError` object that describes the problem

Return Value

`YES` if *absoluteURL* is valid and the file is read correctly, otherwise `NO`.

Discussion

This method sets the URL for the persistent object store associated with the receiver's managed object context to *absoluteURL*.

Availability

Available in Mac OS X v10.4 and later.

See Also

`– revertToContentsOfURL:ofType:error:` (page 11)

`– writeToURL:ofType:forSaveOperation:originalContentsURL:error:` (page 13)

`– configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7)

Declared In

`NSPersistentDocument.h`

revertToContentsOfURL:ofType:error:

Overridden to clean up the managed object context and controllers during a revert.

```
- (BOOL)revertToContentsOfURL:(NSURL *)inAbsoluteURL
    ofType:(NSString *)inTypeName
    error:(NSError **)outError
```

Parameters

inAbsoluteURL

An URL object that specifies the location of the file to which to revert.

inTypeName

The type of the document at *inAbsoluteURL*.

outError

If the method fails to complete correctly, upon return contains an `NSError` object that describes the problem.

Return Value

`YES` if the method completes correctly, otherwise `NO`.

Availability

Available in Mac OS X v10.4 and later.

See Also

`– readFromURL:ofType:error:` (page 10)

`– writeToURL:ofType:forSaveOperation:originalContentsURL:error:` (page 13)

Related Sample Code
QTMetadataEditor

Declared In
`NSPersistentDocument.h`

setHasUndoManager:

Overridden to be a no-op.

```
- (void)setHasUndoManager:(BOOL)flag
```

Parameters

flag

This value is ignored.

Special Considerations

You should not override this method. The persistent document uses the managed object context's undo manager.

See Also
`- managedObjectContext` (page 8)

setManagedObjectContext:

Sets the receiver's managed object context.

```
- (void)setManagedObjectContext:(NSManagedObjectContext *)managedObjectContext
```

Parameters

managedObjectContext

The managed object context for the receiver.

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

See Also
`- managedObjectContext` (page 8)
`- managedObjectModel` (page 9)

Declared In
`NSPersistentDocument.h`

setUndoManager:

Overridden to be a no-op.

```
- (void)setUndoManager:(NSUndoManager *)undoManager
```

Parameters

undoManager

This value is ignored.

Special Considerations

You should not override this method. The persistent document uses the managed object context's undo manager.

See Also

`– managedObjectContext` (page 8)

writeToURL:ofType:forSaveOperation:originalContentsURL:error:

Saves changes in the document's managed object context and saves the document's persistent store to a given URL.

```
- (BOOL)writeToURL:(NSURL *)absoluteURL
    ofType:(NSString *)typeName
    forSaveOperation:(NSSaveOperationType)saveOperation
    originalContentsURL:(NSURL *)absoluteOriginalContentsURL
    error:(NSError **)error
```

Parameters

absoluteURL

An URL that specifies the new location for the document store. It must not be a relative URL.

typeName

The document type.

saveOperation

The save operation type. See the "Constants" section in `NSDocument` for possible values.

absoluteOriginalContentsURL

An URL that specifies the location of the original document store.

error

If the save fails to complete correctly, upon return contains an `NSError` object that describes the problem.

Return Value

`YES` if the save completes correctly, otherwise `NO`.

Availability

Available in Mac OS X v10.4 and later.

See Also

`– readFromURL:ofType:error:` (page 10)

`– revertToContentsOfURL:ofType:error:` (page 11)

`– configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7)

Declared In

`NSPersistentDocument.h`

Deprecated NSPersistentDocument Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

configurePersistentStoreCoordinatorForURL:ofType:error:

Configures the receiver's persistent store coordinator for a given URL and document type. (Deprecated in Mac OS X v10.5. Use `configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7) instead.)

```
- (BOOL)configurePersistentStoreCoordinatorForURL:(NSURL *)url
    ofType:(NSString *)fileType
    error:(NSError **)error
```

Parameters

url

An URL that specifies the location of the document's store.

fileType

The document type.

error

If the method does not complete successfully, upon return contains an `NSError` object that describes the problem.

Return Value

`YES` if the method completes successfully, otherwise `NO`.

Discussion

This method is invoked automatically when an existing document is opened. You override this method to customize creation of a persistent store for a given document or store type. You can retrieve the persistent store coordinator with the following code:

```
[[self managedObjectContext] persistentStoreCoordinator];
```

Availability

Available in Mac OS X v10.4.

Deprecated in Mac OS X v10.5.

See Also

`- persistentStoreTypeForFileType:` (page 10)

`- configurePersistentStoreCoordinatorForURL:ofType:modelConfiguration:storeOptions:error:` (page 7)

Declared In

`NSPersistentDocument.h`

Document Revision History

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

Date	Notes
2008-02-08	Corrected a minor link error.
2007-10-31	Minor formatting changes.
2007-04-06	Updated to include new API introduced in Mac OS X v10.5.
2006-05-23	Added efficiency note to definition of managedObjectModel.
	First publication of this content as a separate document.

Index