Class

NSPersistentDocument

A document object that can integrate with Core Data.

macOS

```
@MainActor
class NSPersistentDocument
```

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 `configurePersistentStoreCoordinator(for:ofType:modelConfiguration:storeOptions:)` 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, it 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.

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 `managedObjectModel` property and `configurePersistentStoreCoordinator(for:ofType:modelConfiguration:storeOptions:)` method. You might wish to do this, for example, to specify a particular managed object model.

Important

`NSPersistentDocument` does not support some document behaviors:

- File wrappers.
- `NSDocument.SaveOperationType.saveToOperation` operation type.

Core Data does not support saving changes to a new document while maintaining the unsaved state in the current document.

- Asynchronous saving.

`NSPersistentDocument` does not support the asynchronous saving API of `NSDocument` because that API requires accessing the document's state on multiple threads and that violates the requirements of the `NSManagedObjectContext` class. Do not override `canAsynchronouslyWrite(to:ofType:for:)`.

Undo Support

The persistent document uses the managed object context's undo manager.

Important

Do not override the following properties, their getters, or their setters:

- `hasUndoManager`
- `undoManager`

The `isDocumentEdited` method returns `true` if the persistent document's managed object context, or editors registered with the context, have uncommitted changes, otherwise it returns `false`.

Topics

Managing the Persistence Objects

`var managedObjectContext: NSManagedObjectContext?`

The managed object context for the document.

`var managedObjectModel: NSManagedObjectModel?`

The managed object model of the document.

`func configurePersistentStoreCoordinator(for: URL, ofType: String, modelConfiguration: String?, storeOptions: [String : Any]?) throws`

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

`func persistentStoreType(forFileType: String) -> String`

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

Document Content Management

`func read(from: URL, ofType: String) throws`

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

`func revert(toContentsOf: URL, ofType: String) throws`

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

`func write(to: URL, ofType: String, for: NSDocument.SaveOperationType, originalContentsURL: URL?) throws`

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

Relationships

Inherits From

NSDocument

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSEditorRegistration
NSFilePresenter
NSMenuItemValidation
NSObjectProtocol
NSUserActivityRestoring
NSUserInterfaceValidations

See Also

Documents

{} Developing a Document-Based App

Write an app that creates, manages, edits, and saves text documents.

class NSDocument

An abstract class that defines the interface for macOS documents.

class NSDocumentController

An object that manages an app's documents.