

[SwiftData](#) / PersistentModel

## Protocol

# PersistentModel


An interface that enables SwiftData to manage a Swift class as a stored model.

iOS 17.0+ | iPadOS 17.0+ | Mac Catalyst 17.0+ | macOS 14.0+ | tvOS 17.0+ | visionOS 1.0+ | watchOS 10.0+ |

Swift 5.9+

```
protocol PersistentModel : AnyObject, Observable, Hashable, Identifiable, Sendable
Metatype
```

## Mentioned in

 Preserving your app's model data across launches

## Topics

### Creating a persistent model

```
init(backingData: any BackingData<Self>)
```

Required

### Identifying the model instance

```
var persistentModelID: PersistentIdentifier
```

```
struct PersistentIdentifier
```

A type that describes the aggregate identity of a SwiftData model.

```
var modelContext: ModelContext?
```

## Accessing a value by key path

```
func getValue<Value, OtherModel>(forKey: KeyPath<Self, Value>) -> Value
```

```
func getValue<Value>(forKey: KeyPath<Self, Value>) -> Value
```

```
func getValue<Value>(forKey: KeyPath<Self, Value>) -> Value
```

```
func getValue<Value, OtherModel>(forKey: KeyPath<Self, Value>) -> Value
```

```
func getValue<Value>(forKey: KeyPath<Self, Value?>) -> Value?
```

```
func getTransformableValue<Value>(forKey: KeyPath<Self, Value>) ->  
Value
```

## Modifying a value by key path

```
func setValue<Value, OtherModel>(forKey: KeyPath<Self, Value>, to:  
Value)
```

```
func setValue<Value, OtherModel>(forKey: KeyPath<Self, Value>, to:  
Value)
```

```
func setValue<Value>(forKey: KeyPath<Self, Value?>, to: Value?)
```

```
func setValue<Value>(forKey: KeyPath<Self, Value>, to: Value)
```

```
func setValue<Value>(forKey: KeyPath<Self, Value>, to: Value)
```

```
func setTransformableValue<Value>(forKey: KeyPath<Self, Value>, to:  
Value)
```

## Accessing supplementary information

```
static var schemaMetadata: [Schema.PropertyMetadata]
```

**Required**

```
var persistentBackingData: any BackingData<Self>
```

**Required**

```
var hasChanges: Bool
```

```
var isDeleted: Bool
```

# Internal

Internal symbols

Restricted-use symbols that the framework requires for macro expansion and other internal tasks.

## Associated Types

```
associatedtype Root : PersistentModel = Self
```

Required

## Type Methods

```
static func createBackingData<P>() -> some BackingData<P>
```

## Default Implementations

Equatable Implementations

Hashable Implementations

---

# Relationships

## Inherits From

- Equatable
- Hashable
- Identifiable
- Observable
- SendableMetatype

---

# See Also

Creating a model container

```
init(for: Schema, migrationPlan: (any SchemaMigrationPlan.Type)?,
configurations: [ModelConfiguration]) throws
```

Creates a model container using the specified schema, migration plan, and configurations.

```
convenience init(for: any PersistentModel.Type..., migrationPlan: (any
SchemaMigrationPlan.Type)?, configurations: ModelConfiguration...)
throws
```

Creates a model container using the specified model types, migration plan, and zero or more configurations.

```
convenience init(for: Schema, migrationPlan: (any SchemaMigrationPlan
.Type)?, configurations: ModelConfiguration...) throws
```

Creates a model container using the specified schema, migration plan, and zero or more configurations.

```
struct ModelConfiguration
```

A type that describes the configuration of an app's schema or specific group of models.

```
class Schema
```

An object that maps model classes to data in the model store, and helps with the migration of that data between releases.

```
protocol SchemaMigrationPlan
```

An interface for describing the evolution of a schema and how to migrate between specific versions.