

[Foundation](#) / NSCoding

Protocol

NSCoding

A protocol that enables an object to be encoded and decoded for archiving and distribution.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.0+ | macOS 10.0+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
protocol NSCoding
```

Overview

The NSCoding protocol declares the two methods that a class must implement so that instances of that class can be encoded and decoded. This capability provides the basis for archiving (where objects and other structures are stored on disk) and distribution (where objects are copied to different address spaces).

In keeping with object-oriented design principles, an object being encoded or decoded is responsible for encoding and decoding its instance variables. A coder instructs the object to do so by invoking [encode\(with:\)](#) or

[doc://com.apple.documentation/documentation/oslog/oslogentry/init\(coder:\)](doc://com.apple.documentation/documentation/oslog/oslogentry/init(coder:)). [encode\(with:\)](#) instructs the object to encode its instance variables to the coder provided; an object can receive this method any number of times.

[doc://com.apple.documentation/documentation/oslog/oslogentry/init\(coder:\)](doc://com.apple.documentation/documentation/oslog/oslogentry/init(coder:)) instructs the object to initialize itself from data in the coder provided; as such, it replaces any other initialization method and is sent only once per object. Any object class that should be codeable must adopt the NSCoding protocol and implement its methods.

It is important to consider the possible types of archiving that a coder supports. In macOS 10.2 and later, keyed archiving is preferred. You may, however, need to support classic archiving. For details, see [Archives and Serializations Programming Guide](#).

Topics

Encoding with a coder

```
func encode(with: NSCoder)
```

Encodes the receiver using a given archiver.

Required

Initializers

```
init?(coder: NSCoder)
```

Required

Relationships

Inherited By

NSSecureCoding

Conforming Types

ByteCountFormatter

CachedURLResponse

DateComponentsFormatter

NSDateFormatter

DateIntervalFormatter

Dimension

EnergyFormatter

FileHandle

FileWrapper

Formatter

HTTPURLResponse

ISO8601DateFormatter

LengthFormatter

ListFormatter

MassFormatter

MeasurementFormatter
MessagePort
NSAffineTransform
NSAppleEventDescriptor
NSArray
NSAttributedString
NSCalendar
NSCharacterSet
NSCloneCommand
NSCloseCommand
NSComparisonPredicate
NSCompoundPredicate
NSCountCommand
NSCountedSet
NSCreateCommand
NSData
NSDataDetector
NSDate
NSDateComponents
NSDateInterval
NSDecimalNumber
NSDecimalNumberHandler
NSDeleteCommand
NSDictionary
NSError
NSException
NSExistsCommand
NSExpression
NSExtensionItem
NSFileSecurity
NSGetCommand
NSHashTable
NSIndexPath
NSIndexSet
NSIndexSpecifier
NSLocale
NSLogicalTest
NSMachPort
NSMapTable
NSMeasurement
NSMiddleSpecifier
NSMoveCommand
NSMutableArray

NSMutableAttributedString
NSMutableCharacterSet
NSMutableData
NSMutableDictionary
NSMutableIndexSet
NSMutableOrderedSet
NSMutableSet
NSMutableString
NSMutableURLRequest
NSNameSpecifier
NSNotification
NSNull
NSNumber
NSOrderedSet
NSOrthography
NSPersonNameComponents
NSPointerArray
NSPredicate
NSPropertySpecifier
NSPurgeableData
NSQuitCommand
NSRandomSpecifier
NSRangeSpecifier
NSRegularExpression
NSRelativeSpecifier
NSScriptCommand
NSScriptCommandDescription
NSScriptObjectSpecifier
NSScriptWhoseTest
NSSet
NSSetCommand
NSSortDescriptor
NSSpecifierTest
NSString
NSTextCheckingResult
NSTimeZone
NSURL
NSURLQueryItem
NSURLRequest
UUID
NSUniqueIDSpecifier
NSValue
NSWhoseSpecifier

NSXPCListenEndpoint
NumberFormatter
PersonNameComponentsFormatter
Port
RelativeDateTimeFormatter
SocketPort
URLAuthenticationChallenge
URLCredential
URLProtectionSpace
URLResponse
Unit
UnitAcceleration
UnitAngle
UnitArea
UnitConcentrationMass
UnitConverterLinear
UnitDispersion
UnitDuration
UnitElectricCharge
UnitElectricCurrent
UnitElectricPotentialDifference
UnitElectricResistance
UnitEnergy
UnitEnergy.EnergyKit
UnitFrequency
UnitFuelEfficiency
UnitIlluminance
UnitInformationStorage
UnitLength
UnitMass
UnitPower
UnitPressure
UnitSpeed
UnitTemperature
UnitVolume

See Also

Adopting Codability

Encoding and Decoding Custom Types

Make your data types encodable and decodable for compatibility with external representations such as JSON.

```
typealias Codable = Decodable & Encodable
```

A type that can convert itself into and out of an external representation.

```
protocol NSSecureCoding
```

A protocol that enables encoding and decoding in a manner that is robust against object substitution attacks.