

[Foundation](#) / [NSExtensionContext](#)

Class

NSExtensionContext

The host app context from which an app extension is invoked.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.1+ | macOS 10.10+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class NSExtensionContext
```

Overview

When a host app sends a request to an app extension, it provides an extension context. For many app extensions, the most important part of the context is the data the user wants to work with, which is contained in the [inputItems](#) property.

Topics

Handling requests

```
func completeRequest(returningItems: [Any]?, completionHandler: ((Bool) -> Void)?)
```

Tells the host app to complete the app extension request with an array of result items.

```
func cancelRequest(withError: any Error)
```

Tells the host app to cancel the app extension request, with a supplied error.

```
let NSExtensionItemsAndErrorsKey: String
```

The extension items and errors key.

Opening URLs

```
func open(URL, completionHandler: ((Bool) -> Void)?)
```

Asks the system to open a URL on behalf of the currently running app extension.

Storing extension items

```
var inputItems: [Any]
```

The list of input NSExtensionItem objects associated with the context.

Controlling media playback in notification content extensions

```
func mediaPlayingStarted()
```

Tells the system that the Notification Content app extension began playing a media file.

```
func mediaPlayingPaused()
```

Tells the system that the Notification Content app extension stopped playing a media file.

Populating your share extension with metadata

```
var intent: INIntent?
```

Metadata for populating your share extensions interface.

Getting Siri-related information

```
var hostedViewMinimumAllowedSize: CGSize
```

The minimum size for a Siri hosted view.

```
var hostedViewMaximumAllowedSize: CGSize
```

The maximum size for a Siri hosted view.

```
func interfaceParametersDescription() -> String
```

Returns a human-readable string describing the data that SiriKit displays to the user when you handle an intent.

Supporting broadcasting

```
func loadBroadcastingApplicationInfo(completion: (String, String, UIImage?) -> Void)

func completeRequest(withBroadcast: URL, setupInfo: [String : any NSCoding & NSObjectProtocol]?)
```

Handling notification actions

```
var notificationActions: [UNNotificationAction]

func performNotificationDefaultAction()

func dismissNotificationContentExtension()
```

Notifications

static let NSExtensionHostDidBecomeActive: NSNotification.Name
Posted when the extension's host app moves from the inactive to the active state.

static let NSExtensionHostWillResignActive: NSNotification.Name
Posted when the extension's host app moves from the active to the inactive state.

static let NSExtensionHostDidEnterBackground: NSNotification.Name
Posted when the extension's host app begins running in the background.

static let NSExtensionHostWillEnterForeground: NSNotification.Name
Posted when the extension's host app begins running in the foreground.

Deprecated

~~func completeRequest(withBroadcast: URL, broadcastConfiguration: RPBroadcastConfiguration, setupInfo: [String : any NSCoding & NSObjectProtocol]?)~~

Tells the host app to complete the app extension request with the specified broadcast information.

Deprecated

~~var widgetActiveDisplayMode: NCWidgetDisplayMode~~

The active display mode of the widget.

Deprecated

~~var widgetLargestAvailableDisplayMode: NCWidgetDisplayMode~~

The largest display mode the widget supports.

Deprecated

~~func `widgetMaximumSize(for: NCWidgetDisplayMode) -> CGSize`~~

Returns the maximum size for the specified widget display mode.

Deprecated

Structures

```
struct DidBecomeActiveMessage  
structDidEnterBackgroundMessage  
struct WillEnterForegroundMessage  
struct WillResignActiveMessage
```

Relationships

Inherits From

NSObject

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol

See Also

Extension Support

protocol `NSExtensionRequestHandling`

The interface an app extension uses to respond to a request from a host app.