

Framework

# RelevanceKit

Provide on-device intelligence with contextual clues that increase your widget's visibility on Apple Watch.

iOS 26.0+ | iPadOS 26.0+ | Mac Catalyst 26.0+ | macOS 26.0+ | visionOS 26.0+ | watchOS 26.0+

## Overview

On Apple Watch, widgets appear in the Smart Stack in an order that best fits a person's context. To order the widgets that appear in the Smart Stack, watchOS attempts to determine a widget's relevance based on several factors, including contextual clues your app provides to the system.

To give your widget additional visibility in the watchOS Smart Stack and ensure it appears when a person needs it, use RelevanceKit to provide contextual clues that signal the widget's relevance to the system. For example, your widget might be most useful at a specific location or time, or every time a person starts a workout.

Note that you use RelevanceKit in combination with [WidgetKit](#) and [App Intents](#) to provide interactive and contextually relevant widgets in the Smart Stack, including iPhone widgets that appear on Apple Watch. When you add an import statement for App Intents to your code, App Intents implicitly adds a dependence to RelevanceKit . You don't need to explicitly add `import RelevanceKit` to your code.

For more information, refer to [Increasing the visibility of widgets in Smart Stacks](#).

### Note

Smart Stacks are available in iOS, iPadOS, and watchOS. However, functionality provided by RelevanceKit is only available in watchOS. Calling its API on other platforms doesn't have any effect.

# Topics

## Providing relevance information

- Increasing the visibility of widgets in Smart Stacks

Provide contextual information and donate intents to the system to make sure your widget appears prominently in Smart Stacks.

`struct RelevantContext`

Contextual clues the system uses to show relevant widgets in the Smart Stack on watchOS.

## Fitness clues

`static func fitness(RelevantContext.FitnessCondition) -> RelevantContext`

Tells the system a widget is relevant because of a person's fitness activity.

`struct FitnessCondition`

Values that represent a person's fitness activity.

## Hardware clues

`static func hardware(headphones: RelevantContext.HeadphonesCondition) -> RelevantContext`

Tells the system a widget is relevant when a person's headphones are connected.

`struct HeadphonesCondition`

A structure that indicates whether a person's headphones are connected.

## Location clues

`static func location(CLRegion) -> RelevantContext`

Tells the system a widget is relevant at a specific location.

`static func location(inferred: RelevantContext.InferredLocation) -> RelevantContext`

Tells the system a widget is relevant at a person's inferred location.

```
struct InferredLocation
```

A structure with values for a person's inferred home, work, school, and commute locations.

## Sleep clues

```
static func sleep(RelevantContext.SleepCondition) -> RelevantContext
```

Tells the system a widget is relevant because of a person's sleep schedule.

```
struct SleepCondition
```

Values that represent a person's typical bedtime or wakeup time.

## Time clues

```
static func date(Date) -> RelevantContext
```

Tells the system a widget is relevant at a specific date.

```
static func date(Date, kind: RelevantContext.DateKind) -> RelevantContext
```

Tells the system a widget is relevant at a specific date and provides an additional contextual hint.

```
static func date(interval: DateInterval, kind: RelevantContext.DateKind) -> RelevantContext
```

Tells the system a widget is relevant for a time interval and provides an additional contextual hint.

```
static func date(range: ClosedRange<Date>, kind: RelevantContext.DateKind) -> RelevantContext
```

Tells the system a widget is relevant for a known date range and provides an additional contextual hint.

```
struct DateKind
```

Values the system uses as additional context for time-based relevance clues.

~~```
static func date(from: Date, to: Date) -> RelevantContext
```~~

Tells the system a widget is relevant between two dates.

Deprecated