Class

MeasurementFormatter

A formatter that provides localized representations of units and measurements.

iOS 10.0+ | iPadOS 10.0+ | Mac Catalyst 13.1+ | macOS 10.12+ | tvOS 10.0+ | visionOS 1.0+ | watchOS 3.0+

```
class MeasurementFormatter
```

Overview

You use the `string(from:)` method to create a localized representation of an `NSMeasurement` object, and you use the `string(from:)` method to create a localized representation of an `Unit` object. The formatter takes into account the specified `locale`, `unitStyle`, and `unitOptions` when producing string representations of units and measurements.

> **Tip**
>
> In Swift, you can use `Measurement.FormatStyle` rather than `MeasurementFormatter`. The `FormatStyle` API offers a declarative idiom for customizing the formatting of various types. Also, Foundation caches identical `FormatStyle` instances, so you don't need to pass them around your app, or risk wasting memory with duplicate formatters.

Topics

Specifying the Format

```
var unitOptions: MeasurementFormatter.UnitOptions
```

The options for how the unit is formatted.

`var unitStyle: Formatter.UnitStyle`

The unit style.

`var locale: Locale!`

The locale of the formatter.

`var numberFormatter: NumberFormatter!`

The number formatter used to format the quantity of a measurement.

Converting Measurements

`func string(from: Measurement<Unit>) -> String`

Creates and returns a localized string representation of the provided measurement.

`func string(from: Unit) -> String`

Creates and returns a localized string representation of the provided unit of measure.

Constants

`struct UnitOptions`

Measurement formatter options.

Instance Methods

`func string<UnitType>(from: Measurement<UnitType>) -> String`

Relationships

Inherits From

`Formatter`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`

CustomStringConvertible
Equatable
Hashable
NSCoding
NSCopying
NSObjectProtocol
NSSecureCoding