Class

ByteCountFormatter

A formatter that converts a byte count value into a localized description that is formatted with the appropriate byte modifier (KB, MB, GB and so on).

iOS 6.0+ | iPadOS 6.0+ | Mac Catalyst 13.1+ | macOS 10.8+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class ByteCountFormatter
```

Overview

Tip

In Swift, you can use `ByteCountFormatStyle` or `Measurement.FormatStyle.ByteCount` rather than `ByteCountFormatter`. 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

Creating Strings from Byte Count

```
class func string(fromByteCount: Int64, countStyle: ByteCountFormatter.CountStyle) -> String
```

Converts a byte count into the specified string format without creating an NSNumber object.

`func string(fromByteCount: Int64) -> String`

Converts a byte count into a string without creating an `NSNumber` object.

Setting Formatting Styles

`var formattingContext: Formatter.Context`

Specify the formatting context for the formatted string.

`var countStyle: ByteCountFormatter.CountStyle`

Specify the number of bytes to be used for kilobytes.

`var allowsNonnumericFormatting: Bool`

Determines whether to allow more natural display of some values.

`var includesActualByteCount: Bool`

Determines whether to include the number of bytes after the formatted string.

`var isAdaptive: Bool`

Determines the display style of the size representation.

`var allowedUnits: ByteCountFormatter.Units`

Specify the units that can be used in the output.

`var includesCount: Bool`

Determines whether to include the count in the resulting formatted string.

`var includesUnit: Bool`

Determines whether to include the units in the resulting formatted string.

`var zeroPadsFractionDigits: Bool`

Determines whether to zero pad fraction digits so a consistent number of characters is displayed in a representation.

Constants

`struct Units`

Specifies the units appropriate for the formatter to display. Specifying any units explicitly causes just those units to be used in showing the number.

`enum CountStyle`

Specifies display of file or storage byte counts. The display style is platform specific.

Instance Methods

```
func string(for: Any?) -> String?
```

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

Type Methods

```
class func string(from: Measurement<UnitInformationStorage>, countStyle
: ByteCountFormatter.CountStyle) -> String
```

Relationships

Inherits From

Formatter

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSCoding
NSCopying
NSObjectProtocol