

[Foundation](#) / [...](#) / [attributed](#) **Deprecated**

Instance Property

attributed **Deprecated**

An attributed format style created from the date format style.

iOS 15.0–18.0 **Deprecated** | iPadOS 15.0–18.0 **Deprecated** | Mac Catalyst 15.0–18.0 **Deprecated** | macOS 12.0–15.0 **Deprecated** | tvOS 15.0–18.0 **Deprecated** | visionOS 1.0+ | watchOS 8.0–11.0 **Deprecated**

```
var attributed: Date.AttributedString { get }
```

Deprecated

Use attributedStyle instead

Discussion

Use a `Date.FormatStyle` instance to customize the lexical representation of a date as a string. Use the format style's `attributed` property to customize the visual representation of the date as a string. Attributed strings can represent the subcomponent characters, words, and phrases of a string with a custom combination of font size, weight, and color.

For example, the function below uses a date format style to create a custom lexical representation of a date, then retrieves an attributed string representation of the same date and applies a visual emphasis to the year component of the date.

```
// Applies visual emphasis to the year component of a formatted attributed date string
private func makeAttributedString() -> AttributedString {
    let date = Date()
    let formatStyle = Date.FormatStyle(date: .abbreviated, time: .standard)
    var attributedString = formatStyle.attributedString(format(date))
}
```

```

for run in attributedString.runs {
    if let dateFieldAttribute = run.attributes.foundation.dateField,
        dateFieldAttribute == .year {
        // When you find a year, change its attributes.
        attributedString[run.range].inlinePresentationIntent = [.emphasized, .stronglyEmphasized]
    }
}
return attributedString

```

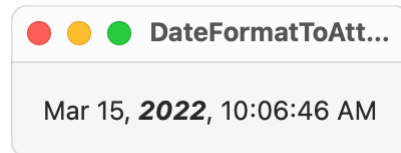
The expression `formatStyle.attributed.format(date)` above creates an attributed string representation of the date. This assigns instances of the `AttributeScopes.FoundationAttributes.DateFieldAttribute` to indicate ranges of the string that represent different date fields. The example then loops over the `runs` of the attributed string to find any run with the `AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.year` attribute. When it finds one, it adds the `inlinePresentationIntent` attributes `emphasized` and `stronglyEmphasized`.

The runs of the resulting attributed string have the following attributes:

Run Text	Attributes
Mar	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.month</code>
15	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.day</code>
2022	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.year</code> <code>emphasized</code> <code>stronglyEmphasized</code>
10	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.hour</code>
06	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.minute</code>
46	<code>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field.second</code>

Run Text	Attributes
AM	<u>AttributeScopes.FoundationAttributes.DateFieldAttribute.Field</u> <u>.amPM</u>

If you create a SwiftUI `Text` view with this attributed string, SwiftUI renders the combination of emphasized and stronglyEmphasized attributes as bold, italicized text, as seen in the following screenshot.



See Also

Applying Visual Attributes to Dates

~~struct~~ `AttributedStyle`

A structure that creates a locale-appropriate attributed string representation of a date instance.

Deprecated