

[Foundation](#) / [Decimal](#) / `Decimal.ParseStrategy`

Structure

Decimal.ParseStrategy

A parse strategy for creating decimal values from formatted strings.

iOS 15.0+ | iPadOS 15.0+ | Mac Catalyst 15.0+ | macOS 12.0+ | tvOS 15.0+ | visionOS 1.0+ | watchOS 8.0+

```
struct ParseStrategy<Format> where Format : FormatStyle, Format.Format  
Input == Decimal
```

Overview

Create an explicit `Decimal.ParseStrategy` to parse multiple strings according to the same parse strategy. In the following example, `usCurrencyStrategy` is a `Decimal.ParseStrategy` that uses US dollars and the `en_US` locale's conventions for number formatting. The example then uses this strategy to parse an array of strings, some of which represent valid US currency values.

```
let usCurrencyStrategy: Decimal.ParseStrategy =  
    Decimal.FormatStyle.Currency(code: "USD",  
                                  locale: Locale(identifier: "en_US"))  
    .parseStrategy  
let currencyValues = ["$100.11", "$1,000.22", "$10,000.33", "€100.44"]  
let parsedValues = currencyValues.map { try? usCurrencyStrategy.parse($0) } // [Opti
```

You don't need to instantiate a parse strategy variable to parse a single string. Instead, use the `init(_:format:lenient:)` initializer, which takes a source `String` and a `format` parameter to parse the string according to the provided `Decimal.FormatStyle`. The following example parses a string that represents a currency value in US dollars.

```
let formattedUSDollars = "$1,234.56"
let parsedUSDollars = try? Decimal(formattedUSDollars, format: .currency(code: "USD"
    .locale(Locale(identifier: "en_US")))) // 1234.56
```

Decimal also has an `init(_ :strategy:)` initializer, if it's more convenient to pass a [Decimal.ParseStrategy](#) instance rather than implicitly derive a strategy from a [Decimal.FormatStyle](#).

Topics

Creating a decimal parse strategy

`init(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified decimal format style.

`init(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified decimal currency format style.

`init(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified decimal percentage format style.

Accessing strategy properties

`var formatStyle: Format`

The format style this strategy uses when parsing strings.

`var lenient: Bool`

A Boolean value that indicates whether parsing allows any discrepancies in the expected format.

Relationships

Conforms To

Decodable

Encodable

Equatable

Parcelable
ParseStrategy
Sendable
SendableMetatype

See Also

Data parsing in Swift

`protocol ParseableFormatStyle`

A type that can convert a given input data type into a representation in an output type.

`protocol ParseStrategy`

A type that parses an input representation, such as a formatted string, into a provided data type.

`struct IntegerParseStrategy`

A parse strategy for creating integer values from formatted strings.

`struct FloatingPointParseStrategy`

A parse strategy for creating floating-point values from formatted strings.