CFNumberFormatter Reference

Core Foundation

2007-05-23

Contents

CFNumberFormatter Reference

Derived From:	CFType
Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Data Formatting Guide for Core Foundation
Declared in	CFNumberFormatter.h

Overview

CFNumberFormatter objects format the textual representations of *CFNumber* objects, and convert textual representations of numbers into *CFNumber* objects. The representation encompasses integers, floats, and doubles; floats and doubles can be formatted to a specified decimal position. You specify how strings are formatted and parsed by setting a format string and other properties of a `CFNumberFormatter` object. The format of the format string itself is defined by Unicode Technical Standard #35.

Note that `CFNumberFormatter` is not thread-safe. Do not use a single instance from multiple threads.

The `CFNumberFormatter` opaque type is available in Mac OS X v10.3 and later.

Unlike some other Core Foundation opaque types with names similar to a corresponding Cocoa Foundation class (such as `CFString` and NSString), `CFNumberFormatter` objects cannot be cast ("toll-free bridged") to `NSNumberFormatter` objects.

Functions by Task

Creating a Number Formatter

`CFNumberFormatterCreate` (page 7)
: Creates a new CFNumberFormatter object, localized to the given locale, which will format numbers to the given style.

Configuring a Number Formatter

`CFNumberFormatterSetFormat` (page 12)
: Sets the format string of a number formatter.

`CFNumberFormatterSetProperty` (page 13)
: Sets a number formatter property using a key-value pair.

Formatting Values

`CFNumberFormatterCreateNumberFromString` (page 7)
: Returns a number object representing a given string.

`CFNumberFormatterCreateStringWithNumber` (page 8)
: Returns a string representation of the given number using the specified number formatter.

`CFNumberFormatterCreateStringWithValue` (page 9)
: Returns a string representation of the given number or value using the specified number formatter.

`CFNumberFormatterGetDecimalInfoForCurrencyCode` (page 9)
: Returns the number of fraction digits that should be displayed, and the rounding increment, for a given currency.

`CFNumberFormatterGetValueFromString` (page 12)
: Returns a number or value representing a given string.

Examining a Number Formatter

`CFNumberFormatterCopyProperty` (page 6)
: Returns a copy of a number formatter's value for a given key.

`CFNumberFormatterGetFormat` (page 10)
: Returns a format string for the given number formatter object.

`CFNumberFormatterGetLocale` (page 11)
: Returns the locale object used to create the given number formatter object.

`CFNumberFormatterGetStyle` (page 11)
: Returns the number style used to create the given number formatter object.

Getting the CFNumberFormatter Type ID

`CFNumberFormatterGetTypeID` (page 11)
: Returns the type identifier for the `CFNumberFormatter` opaque type.

Functions

CFNumberFormatterCopyProperty

Returns a copy of a number formatter's value for a given key.

```
CFTypeRef CFNumberFormatterCopyProperty (
   CFNumberFormatterRef formatter,
   CFStringRef key
);
```

Parameters

formatter
: The number formatter to examine.

`key`
: A property key. See "Number Formatter Property Keys" (page 16) for valid values.

Return Value
A `CFType` object that is a copy of the property value for *`key`*. Returns `NULL` if there is no value specified for *`key`*. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterCreate

Creates a new CFNumberFormatter object, localized to the given locale, which will format numbers to the given style.

```
CFNumberFormatterRef CFNumberFormatterCreate (
   CFAllocatorRef allocator,
   CFLocaleRef locale,
   CFNumberFormatterStyle style
);
```

Parameters

`alloc`
: The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

`locale`
: A locale to use for localization. If `NULL`, the function uses the default system locale. Use `CFLocaleCopyCurrent` to specify the locale of the current user.

`style`
: A number style. See "Number Formatter Styles" (page 15) for possible values.

Return Value
A new number formatter, localized to the given locale, which will format numbers using the given style. Returns `NULL` if there was a problem creating the formatter. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterCreateNumberFromString

Returns a number object representing a given string.

```
CFNumberRef CFNumberFormatterCreateNumberFromString (
   CFAllocatorRef allocator,
   CFNumberFormatterRef formatter,
   CFStringRef string,
   CFRange *rangep,
   CFOptionFlags options
);
```

Parameters

alloc
: The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

formatter
: The number formatter to use.

string
: The string to parse.

rangep
: A reference to a range that specifies the substring of *string* to be parsed. If `NULL`, the whole string is parsed. On return, contains the range of the actual extent of the parse (may be less than the given range).

options
: Specifies various configuration options to change the behavior of the parse. Currently, `kCFNumberFormatterParseIntegersOnly` (page 21) is the only possible value for this parameter.

Return Value
A new number that represents the given string. Returns `NULL` if there was a problem creating the number. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterCreateStringWithNumber

Returns a string representation of the given number using the specified number formatter.

```
CFStringRef CFNumberFormatterCreateStringWithNumber (
   CFAllocatorRef allocator,
   CFNumberFormatterRef formatter,
   CFNumberRef number
);
```

Parameters

alloc
: The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

formatter
: The number formatter to use.

number
: The number from which to create a string representation.

Return Value
A new string that represents the given number in the specified format. Returns `NULL` if there was a problem creating the string. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterCreateStringWithValue

Returns a string representation of the given number or value using the specified number formatter.

```
CFStringRef CFNumberFormatterCreateStringWithValue (
   CFAllocatorRef allocator,
   CFNumberFormatterRef formatter,
   CFNumberType numberType,
   const void *valuePtr
);
```

Parameters

alloc
: The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

formatter
: The number formatter to use.

numberType
: The type of value that *valuePtr* references. Valid values are listed in `CFNumberType`.

valuePtr
: A pointer to the value to be converted.

Return Value
A new string that represents the given number or value formatted by *formatter*. Returns `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterGetDecimalInfoForCurrencyCode

Returns the number of fraction digits that should be displayed, and the rounding increment, for a given currency.

```
Boolean CFNumberFormatterGetDecimalInfoForCurrencyCode (
   CFStringRef currencyCode,
   int32_t *defaultFractionDigits,
   double *roundingIncrement
);
```

Parameters

currencyCode
: A string containing a ISO 4217 3-letter currency code. For example, AUD for Australian Dollars, EUR for Euros.

defaultFractionDigits
: Upon return, contains the number of fraction digits that should be displayed for the currency specified by *currencyCode*.

roundingIncrement
: Upon return, contains the rounding increment for the currency specified by *currencyCode*, or `0.0` if no rounding is done by the currency.

Return Value

`true` if the information was obtained successfully, otherwise `false` (for example, if the currency code is unknown or the information is not available).

Discussion

The returned values are not localized because these are properties of the currency.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

CFNumberFormatterGetFormat

Returns a format string for the given number formatter object.

```
CFStringRef CFNumberFormatterGetFormat (
   CFNumberFormatterRef formatter
);
```

Parameters

formatter
: The number formatter to examine.

Return Value

The format string for *formatter* as was specified by calling the `CFNumberFormatterSetFormat` (page 12) function, or derived from the number formatter's style. The format of this string is defined by Unicode Technical Standard #35.. Ownership follows the Get Rule.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

CFNumberFormatterGetLocale

Returns the locale object used to create the given number formatter object.

```
CFLocaleRef CFNumberFormatterGetLocale (
   CFNumberFormatterRef formatter
);
```

Parameters

formatter

The number formatter to examine.

Return Value

The locale used to create *formatter*. Ownership follows the Get Rule.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

CFNumberFormatterGetStyle

Returns the number style used to create the given number formatter object.

```
CFNumberFormatterStyle CFNumberFormatterGetStyle (
   CFNumberFormatterRef formatter
);
```

Parameters

formatter

The number formatter to examine.

Return Value

The number style used to create *formatter*.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

CFNumberFormatterGetTypeID

Returns the type identifier for the `CFNumberFormatter` opaque type.

```
CFTypeID CFNumberFormatterGetTypeID (
   void
);
```

Return Value

The type identifier for the `CFNumberFormatter` opaque type.

Availability

Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterGetValueFromString

Returns a number or value representing a given string.

```
Boolean CFNumberFormatterGetValueFromString (
   CFNumberFormatterRef formatter,
   CFStringRef string,
   CFRange *rangep,
   CFNumberType numberType,
   void *valuePtr
);
```

Parameters

formatter
The number formatter to use.

string
The string to parse.

rangep
A reference to a range that specifies the substring of *string* to be parsed. If `NULL`, the whole string is parsed. Upon return, contains the range of the actual extent of the parse (may be less than the given range).

numberType
The type of value that *valuePtr* references. Valid values are listed in `CFNumberType`.

valuePtr
Upon return, contains a number or value representing the string in the specified format. You are responsible for releasing this value.

Return Value
`true` if the string was parsed successfully, otherwise `false`.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterSetFormat

Sets the format string of a number formatter.

```
void CFNumberFormatterSetFormat (
   CFNumberFormatterRef formatter,
   CFStringRef formatString
);
```

Parameters

formatter
The number formatter to modify.

formatString

The format string to be used by *formatter*. The format of this string is defined by Unicode Technical Standard #35.

Discussion

The format string may override other properties previously set using other functions. If this function is not called, the default value of the format string is derived from the number formatter's style.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

CFNumberFormatterSetProperty

Sets a number formatter property using a key-value pair.

```
void CFNumberFormatterSetProperty (
   CFNumberFormatterRef formatter,
   CFStringRef key,
   CFTypeRef value
);
```

Parameters

formatter

The number formatter to modify.

key

The name of the property of *formatter* to set. See "Number Formatter Property Keys" (page 16) for a description of possible values.

value

The value of the specified key. This must be an instance of the correct `CFType` object for the corresponding key.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFNumberFormatter.h`

Data Types

CFNumberFormatterOptionFlags

Type for constants specifying how numbers should be parsed.

```
typedef CFOptionFlags CFNumberFormatterOptionFlags;
```

Discussion

For values, see "Number Format Options" (page 21).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterPadPosition

Type for constants specifying how numbers should be padded.

```
typedef CFIndex CFNumberFormatterPadPosition;
```

Discussion
For values, see "Padding Positions" (page 22).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterRef

A reference to a `CFNumberFormatter` object.

```
typedef struct __CFNumberFormatter *CFNumberFormatterRef;
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

CFNumberFormatterStyle

Type for constants specifying a formatter style.

```
typedef CFIndex CFNumberFormatterStyle;
```

Discussion
For values, see "Number Formatter Styles" (page 15).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`CFNumberFormatter.h`

Constants

Number Formatter Styles

Predefined number format styles.

```
enum {
    kCFNumberFormatterNoStyle = 0,
    kCFNumberFormatterDecimalStyle = 1,
    kCFNumberFormatterCurrencyStyle = 2,
    kCFNumberFormatterPercentStyle = 3,
    kCFNumberFormatterScientificStyle = 4,
    kCFNumberFormatterSpellOutStyle = 5
};
```

Constants

`kCFNumberFormatterNoStyle`

Specifies no style.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterDecimalStyle`

Specifies a decimal style format.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterCurrencyStyle`

Specifies a currency style format.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPercentStyle`

Specifies a percent style format.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterScientificStyle`

Specifies a scientific style format.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterSpellOutStyle`

Specifies a spelled out format.

Available in Mac OS X v10.4 and later.

Declared in `CFNumberFormatter.h`.

Discussion

The format for these number styles is not exact because they depend on the locale, user preference settings, and operating system version. Do not use these constants if you want an exact format (for example, if you are parsing data in a given format). In general, however, you are encouraged to use these styles to accommodate user preferences.

Declared In
`CFNumberFormatter.h`

Number Formatter Property Keys

The keys used in key-value pairs to specify the value of number formatter properties.

```
const CFStringRef kCFNumberFormatterCurrencyCode;
const CFStringRef kCFNumberFormatterDecimalSeparator;
const CFStringRef kCFNumberFormatterCurrencyDecimalSeparator;
const CFStringRef kCFNumberFormatterAlwaysShowDecimalSeparator;
const CFStringRef kCFNumberFormatterGroupingSeparator;
const CFStringRef kCFNumberFormatterUseGroupingSeparator;
const CFStringRef kCFNumberFormatterPercentSymbol;
const CFStringRef kCFNumberFormatterZeroSymbol;
const CFStringRef kCFNumberFormatterNaNSymbol;
const CFStringRef kCFNumberFormatterInfinitySymbol;
const CFStringRef kCFNumberFormatterMinusSign;
const CFStringRef kCFNumberFormatterPlusSign;
const CFStringRef kCFNumberFormatterCurrencySymbol;
const CFStringRef kCFNumberFormatterExponentSymbol;
const CFStringRef kCFNumberFormatterMinIntegerDigits;
const CFStringRef kCFNumberFormatterMaxIntegerDigits;
const CFStringRef kCFNumberFormatterMinFractionDigits;
const CFStringRef kCFNumberFormatterMaxFractionDigits;
const CFStringRef kCFNumberFormatterGroupingSize;
const CFStringRef kCFNumberFormatterSecondaryGroupingSize;
const CFStringRef kCFNumberFormatterRoundingMode;
const CFStringRef kCFNumberFormatterRoundingIncrement;
const CFStringRef kCFNumberFormatterFormatWidth;
const CFStringRef kCFNumberFormatterPaddingPosition;
const CFStringRef kCFNumberFormatterPaddingCharacter;
const CFStringRef kCFNumberFormatterDefaultFormat;
const CFStringRef kCFNumberFormatterMultiplier;
const CFStringRef kCFNumberFormatterPositivePrefix;
const CFStringRef kCFNumberFormatterPositiveSuffix;
const CFStringRef kCFNumberFormatterNegativePrefix;
const CFStringRef kCFNumberFormatterNegativeSuffix;
const CFStringRef kCFNumberFormatterPerMillSymbol;
const CFStringRef kCFNumberFormatterInternationalCurrencySymbol;
const CFStringRef kCFNumberFormatterCurrencyGroupingSeparator;
const CFStringRef kCFNumberFormatterIsLenient;
const CFStringRef kCFNumberFormatterUseSignificantDigits;
const CFStringRef kCFNumberFormatterMinSignificantDigits;
const CFStringRef kCFNumberFormatterMaxSignificantDigits;
```

Constants

`kCFNumberFormatterCurrencyCode`

Specifies the currency code, a `CFString` object.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterDecimalSeparator`
: Specifies the decimal separator, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterCurrencyDecimalSeparator`
: Specifies the currency decimal separator, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterAlwaysShowDecimalSeparator`
: Specifies if the result of converting a value to a string should always contain the decimal separator, even if the number is an integer.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterGroupingSeparator`
: Specifies the grouping separator, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterUseGroupingSeparator`
: Specifies if the grouping separator should be used, a `CFBoolean` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPercentSymbol`
: Specifies the string that is used to represent the percent symbol, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterZeroSymbol`
: Specifies the string that is used to represent zero, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterNaNSymbol`
: Specifies the string that is used to represent NaN (“not a number”) when values are converted to strings, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterInfinitySymbol`
: Specifies the string that is used to represent the symbol for infinity, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMinusSign`
: Specifies the symbol for the minus sign, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPlusSign`
: Specifies the symbol for the plus sign, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterCurrencySymbol`
: Specifies the symbol for the currency, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterExponentSymbol`
: Specifies the exponent symbol (“E” or “e”) in the scientific notation of numbers (for example, as in `1.0e+56`), a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMinIntegerDigits`
: Specifies the minimum number of integer digits before a decimal point, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMaxIntegerDigits`
: Specifies the maximum number of integer digits before a decimal point, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMinFractionDigits`
: Specifies the minimum number of digits after a decimal point, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMaxFractionDigits`
: Specifies the maximum number of digits after a decimal point, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterGroupingSize`
: Specifies how often the “thousands” or grouping separator appears, as in “10,000,000”, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterSecondaryGroupingSize`
: Specifies how often the secondary grouping separator appears, a `CFNumber` object. See Unicode Technical Standard #35 for more information.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundingMode`
: Specifies how the last digit is rounded, as when `3.1415926535...` is rounded to three decimal places, as in `3.142`, a `CFNumber` object. See "`Rounding Modes`" (page 21) for possible values.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundingIncrement`
: Specifies a positive rounding increment, or `0.0` to disable rounding, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterFormatWidth`
: Specifies the width of a formatted number within a string that is either left justified or right justified based on the value of `kCFNumberFormatterPaddingPosition` (page 19), a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPaddingPosition`
: Specifies the position of a formatted number within a string, a `CFNumber` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPaddingCharacter`
: Specifies the padding character to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterDefaultFormat`
: The original format string for the formatter (given the date and time style and locale specified at creation), a `CFString` object.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMultiplier`
: Specifies the multiplier to use when placing a formatted number within a string, a `CFNumber` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPositivePrefix`
: Specifies the plus sign prefix symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPositiveSuffix`
: Specifies the plus sign suffix symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterNegativePrefix`
: Specifies the minus sign prefix symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterNegativeSuffix`
: Specifies the minus sign suffix symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPerMillSymbol`
: Specifies the per mill (1/1000) symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterInternationalCurrencySymbol`
: Specifies the international currency symbol to use when placing a formatted number within a string, a `CFString` object.

  Available in Mac OS X v10.4 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterCurrencyGroupingSeparator`
: Specifies the grouping symbol to use when placing a currency value within a string, a `CFString` object.

  Available in Mac OS X v10.5 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterIsLenient`
: Specifies whether the formatter is lenient, a`CFBoolean` object.

  Available in Mac OS X v10.5 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterUseSignificantDigits`
: Specifies the whether the formatter uses significant digits, a `CFBoolean` object.

  Available in Mac OS X v10.5 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMinSignificantDigits`
: Specifies the minimum number of significant digits to use, a`CFNumber` object.

  Available in Mac OS X v10.5 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterMaxSignificantDigits`
: Specifies the maximum number of significant digits to use, a`CFNumber` object.

  Available in Mac OS X v10.5 and later.

  Declared in `CFNumberFormatter.h`.

Discussion

The values for these keys are all `CFType` objects. The specific types for each key are specified above.

Declared In
`CFNumberFormatter.h`

Number Format Options

These constants are used to specify how numbers should be parsed.

```
enum {
    kCFNumberFormatterParseIntegersOnly = 1
};
```

Constants

`kCFNumberFormatterParseIntegersOnly`
Specifies that only integers should be parsed.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

Declared In
`CFNumberFormatter.h`

Rounding Modes

These constants are used to specify how numbers should be rounded.

```
typedef enum {
    kCFNumberFormatterRoundCeiling = 0,
    kCFNumberFormatterRoundFloor = 1,
    kCFNumberFormatterRoundDown = 2,
    kCFNumberFormatterRoundUp = 3,
    kCFNumberFormatterRoundHalfEven = 4,
    kCFNumberFormatterRoundHalfDown = 5,
    kCFNumberFormatterRoundHalfUp = 6
} CFNumberFormatterRoundingMode;
```

Constants

`kCFNumberFormatterRoundCeiling`
Round up to next larger number with the proper number of fraction digits.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundFloor`
Round down to next larger number with the proper number of fraction digits.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundDown`
Round down to next larger number with the proper number of fraction digits.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundUp`
: Round up to next larger number with the proper number of fraction digits.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundHalfEven`
: Round the last digit, when followed by a `5`, toward an even digit (`.25 -> .2, .35 -> .4`)

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundHalfDown`
: Round down when a `5` follows putative last digit.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterRoundHalfUp`
: Round up when a 5 follows putative last digit.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

Declared In
`CFNumberFormatter.h`

Padding Positions

These constants are used to specify how numbers should be padded.

```
typedef enum {
    kCFNumberFormatterPadBeforePrefix = 0,
    kCFNumberFormatterPadAfterPrefix = 1,
    kCFNumberFormatterPadBeforeSuffix = 2,
    kCFNumberFormatterPadAfterSuffix = 3
};
```

Constants
`kCFNumberFormatterPadBeforePrefix`
: Specifies the number of padding characters before the prefix.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPadAfterPrefix`
: Specifies the number of padding characters after the prefix.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPadBeforeSuffix`
: Specifies the number of padding characters before the suffix.

  Available in Mac OS X v10.3 and later.

  Declared in `CFNumberFormatter.h`.

`kCFNumberFormatterPadAfterSuffix`
Specifies the number of padding characters after the suffix.

Available in Mac OS X v10.3 and later.

Declared in `CFNumberFormatter.h`.

Declared In
`CFNumberFormatter.h`

Document Revision History

This table describes the changes to *CFNumberFormatter Reference*.

Date	Notes
2007-05-23	Updated to include new API in Mac OS X v10.5.
2006-01-10	Corrected links to ICU resources.
2005-12-06	Made minor changes to text to conform to reference consistency guidelines.
2005-08-11	Updated link to ICU library.
2005-04-29	Updated for Mac OS X v10.4.
2004-04-22	Added references to ICU Library for formatting information.
2003-07-01	First version of this document.

Index