CFCharacterSet Reference

Core Foundation

2006-12-01

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

CFCharacterSet Reference

Derived From:	CFType
Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Strings Programming Guide for Core Foundation
Declared in	CFCharacterSet.h

Overview

A CFCharacterSet object represents a set of Unicode compliant characters. CFString uses CFCharacterSet objects to group characters together for searching operations, so that they can find any of a particular set of characters during a search. The two opaque types, CFCharacterSet and CFMutableCharacterSet, define the interface for static and dynamic character sets, respectively. The objects you create using these opaque types are referred to as character set objects (and when no confusion will result, merely as character sets).

CFCharacterSet's principal function, `CFCharacterSetIsCharacterMember` (page 11), provides the basis for all other functions in its interface. You create a character set using one of the `CFCharacterSetCreate...` functions. You may also use any one of the predefined character sets using the `CFCharacterSetGetPredefined` (page 10) function.

CFCharacterSet is "toll-free bridged" with its Cocoa Foundation counterpart, NSCharacterSet. This means that the Core Foundation type is interchangeable in function or method calls with the bridged Foundation object. Therefore, in a method where you see an `NSCharacterSet *` parameter, you can pass in a `CFCharacterSetRef`, and in a function where you see a `CFCharacterSetRef` parameter, you can pass in an NSCharacterSet instance. This capability also applies to concrete subclasses of NSCharacterSet. See Interchangeable Data Types for more information on toll-free bridging.

Functions by Task

Creating Character Sets

`CFCharacterSetCreateCopy` (page 7)
Creates a new character set with the values from a given character set.

`CFCharacterSetCreateInvertedSet` (page 7)
Creates a new immutable character set that is the invert of the specified character set.

`CFCharacterSetCreateWithCharactersInRange` (page 9)
Creates a new character set with the values from the given range of Unicode characters.

`CFCharacterSetCreateWithCharactersInString` (page 9)
Creates a new character set with the values in the given string.

`CFCharacterSetCreateWithBitmapRepresentation` (page 8)
Creates a new immutable character set with the bitmap representation specified by given data.

Getting Predefined Character Sets

`CFCharacterSetGetPredefined` (page 10)
Returns a predefined character set.

Querying Character Sets

`CFCharacterSetCreateBitmapRepresentation` (page 6)
Creates a new immutable data with the bitmap representation from the given character set.

`CFCharacterSetHasMemberInPlane` (page 10)
Reports whether or not a character set contains at least one member character in the specified plane.

`CFCharacterSetIsCharacterMember` (page 11)
Reports whether or not a given Unicode character is in a character set.

`CFCharacterSetIsLongCharacterMember` (page 11)
Reports whether or not a given UTF-32 character is in a character set.

`CFCharacterSetIsSupersetOfSet` (page 12)
Reports whether or not a character set is a superset of another set.

Getting the Character Set Type Identifier

`CFCharacterSetGetTypeID` (page 10)
Returns the type identifier of the CFCharacterSet opaque type.

Functions

CFCharacterSetCreateBitmapRepresentation

Creates a new immutable data with the bitmap representation from the given character set.

```
CFDataRef CFCharacterSetCreateBitmapRepresentation (
   CFAllocatorRef alloc,
   CFCharacterSetRef theSet
);
```

Parameters

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

theSet

The set from which to create a bitmap representation. Refer to the comments for `CFCharacterSetCreateWithBitmapRepresentation` (page 8) for the detailed discussion of the bitmap representation format.

Return Value
A new CFData object containing a bitmap representation of *theSet*. Ownership follows the Create Rule.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
`CFCharacterSet.h`

CFCharacterSetCreateCopy

Creates a new character set with the values from a given character set.

```
CFCharacterSetRef CFCharacterSetCreateCopy (
   CFAllocatorRef alloc,
   CFCharacterSetRef theSet
);
```

Parameters

alloc

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

theSet

The character set to copy.

Return Value
A new character set that is a copy of *theSet*. Ownership follows the Create Rule.

Discussion
This function tries to compact the backing store where applicable.

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

Declared In
`CFCharacterSet.h`

CFCharacterSetCreateInvertedSet

Creates a new immutable character set that is the invert of the specified character set.

```
CFCharacterSetRef CFCharacterSetCreateInvertedSet (
   CFAllocatorRef alloc,
   CFCharacterSetRef theSet
);
```

Parameters

alloc

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

theSet

The character set from which to create an inverted set.

Return Value

A new character set that is the invert of *theSet*. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetCreateWithBitmapRepresentation

Creates a new immutable character set with the bitmap representation specified by given data.

```
CFCharacterSetRef CFCharacterSetCreateWithBitmapRepresentation (
   CFAllocatorRef alloc,
   CFDataRef theData
);
```

Parameters

alloc

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

theData

A CFData object that specifies the bitmap representation of the Unicode character points the for the new character set. The bitmap representation could contain all the Unicode character range starting from BMP to Plane 16. The first 8KiB (8192 bytes) of the data represent the BMP range. The BMP range 8KiB can be followed by zero to sixteen 8KiB bitmaps, each prepended with the plane index byte. For example, the bitmap representing the BMP and Plane 2 has the size of 16385 bytes (8KiB for BMP, 1 byte index, and a 8KiB bitmap for Plane 2). The plane index byte, in this case, contains the integer value two.

If the data contains a Plane index byte outside of the valid Plane range (1 to 16), the behavior is undefined.

Return Value

A new character set containing the indicated characters from *theData*. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetCreateWithCharactersInRange

Creates a new character set with the values from the given range of Unicode characters.

```
CFCharacterSetRef CFCharacterSetCreateWithCharactersInRange (
   CFAllocatorRef alloc,
   CFRange theRange
);
```

Parameters

alloc

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

theRange

The Unicode range of characters of the new character set. The function accepts the range in 32-bit in the UTF-32 format. The valid character point range is from 0x00000 to 0x10FFFF.

Return Value

A new character set that contains a contiguous range of Unicode characters. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetCreateWithCharactersInString

Creates a new character set with the values in the given string.

```
CFCharacterSetRef CFCharacterSetCreateWithCharactersInString (
   CFAllocatorRef alloc,
   CFStringRef theString
);
```

Parameters

alloc

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

theString

A string containing the characters for the new set.

Return Value

A new character set containing the characters from *theString*. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetGetPredefined

Returns a predefined character set.

```
CFCharacterSetRef CFCharacterSetGetPredefined (
   CFCharacterSetPredefinedSet theSetIdentifier
);
```

Parameters

theSetIdentifier

A predefined character set. See "Predefined CFCharacterSet Selector Values" (page 13) for the list of available character sets.

Return Value

A predefined character set. This instance is owned by Core Foundation.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFCharacterSet.h

CFCharacterSetGetTypeID

Returns the type identifier of the CFCharacterSet opaque type.

```
CFTypeID CFCharacterSetGetTypeID (
   void
);
```

Return Value

The type identifier of the CFCharacterSet opaque type.

Discussion

CFMutableCharacterSet objects have the same type identifier as CFCharacterSet objects.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFCharacterSet.h

CFCharacterSetHasMemberInPlane

Reports whether or not a character set contains at least one member character in the specified plane.

```
Boolean CFCharacterSetHasMemberInPlane (
   CFCharacterSetRef theSet,
   CFIndex thePlane
);
```

Parameters

theSet

The character set to examine.

thePlane

The plane number to be checked for the membership. The valid value range is from 0 to 16. If the value is outside of the valid plane number range, the behavior is undefined.

Return Value

`true` if at least one member character is in the specified plane, otherwise `false`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetIsCharacterMember

Reports whether or not a given Unicode character is in a character set.

```
Boolean CFCharacterSetIsCharacterMember (
   CFCharacterSetRef theSet,
   UniChar theChar
);
```

Parameters

theSet

The character set to examine.

theChar

The Unicode character for which to test against the character set. Note that this function takes 16-bit Unicode character value; hence, it does not support access to the non-BMP planes.

Return Value

`true` if *theSet* contains *theChar*, otherwise `false`.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetIsLongCharacterMember

Reports whether or not a given UTF-32 character is in a character set.

```
Boolean CFCharacterSetIsLongCharacterMember (
   CFCharacterSetRef theSet,
   UTF32Char theChar
);
```

Parameters

theSet

The character set to examine.

theChar

The UTF-32 character for which to test against the character set.

Return Value

`true` if *theSet* contains *theChar*, otherwise `false`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFCharacterSet.h`

CFCharacterSetIsSupersetOfSet

Reports whether or not a character set is a superset of another set.

```
Boolean CFCharacterSetIsSupersetOfSet (
   CFCharacterSetRef theSet,
   CFCharacterSetRef theOtherset
);
```

Parameters

theSet

The character set to be checked for the membership of *theOtherSet*.

theOtherSet

The character set to be checked whether or not it is a subset of *theSet*.

Return Value

`true` if *theSet* is a superset of *theOtherSet*, otherwise `false`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFCharacterSet.h`

Data Types

CFCharacterSetPredefinedSet

Defines a predefined character set.

```
typedef CFIndex CFCharacterSetPredefinedSet;
```

Discussion
See "Predefined CFCharacterSet Selector Values" (page 13) for values.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFCharacterSet.h`

CFCharacterSetRef

A reference to an immutable character set object.

```
typedef const struct __CFCharacterSet *CFCharacterSetRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFCharacterSet.h`

Constants

Predefined CFCharacterSet Selector Values

Identifiers for the available predefined CFCharacterSet objects.

```
enum {
    kCFCharacterSetControl = 1,
    kCFCharacterSetWhitespace,
    kCFCharacterSetWhitespaceAndNewline,
    kCFCharacterSetDecimalDigit,
    kCFCharacterSetLetter,
    kCFCharacterSetLowercaseLetter,
    kCFCharacterSetUppercaseLetter,
    kCFCharacterSetNonBase,
    kCFCharacterSetDecomposable,
    kCFCharacterSetAlphaNumeric,
    kCFCharacterSetPunctuation,
    kCFCharacterSetCapitalizedLetter = 13,
    kCFCharacterSetSymbol = 14,
    kCFCharacterSetNewline = 15,
    kCFCharacterSetIllegal = 12
};
```

Constants

`kCFCharacterSetControl`
: Control character set (Unicode General Category Cc and Cf).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetWhitespace`
: Whitespace character set (Unicode General Category Zs and U0009 CHARACTER TABULATION).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetWhitespaceAndNewline`
: Whitespace and Newline character set (Unicode General Category Z*, `U000A ~ U000D`, and `U0085`).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetDecimalDigit`
: Decimal digit character set.

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetLetter`
: Letter character set (Unicode General Category L* & M*).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetLowercaseLetter`
: Lowercase character set (Unicode General Category Ll).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetUppercaseLetter`
: Uppercase character set (Unicode General Category Lu and Lt).

  Available in Mac OS X v10.0 and later.

  Declared in `CFCharacterSet.h`.

`kCFCharacterSetNonBase`
Non-base character set (Unicode General Category M*).

Available in Mac OS X v10.0 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetDecomposable`
Canonically decomposable character set.

Available in Mac OS X v10.0 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetAlphaNumeric`
Alpha Numeric character set (Unicode General Category L*, M*, & N*).

Available in Mac OS X v10.0 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetPunctuation`
Punctuation character set (Unicode General Category P*).

Available in Mac OS X v10.0 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetCapitalizedLetter`
Titlecase character set (Unicode General Category Lt).

Available in Mac OS X v10.2 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetSymbol`
Symbol character set (Unicode General Category S*).

Available in Mac OS X v10.3 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetNewline`
Newline character set (`U000A ~ U000D`, `U0085`, `U2028`, and `U2029`).

Available in Mac OS X v10.5 and later.

Declared in `CFCharacterSet.h`.

`kCFCharacterSetIllegal`
Illegal character set.

Available in Mac OS X v10.0 and later.

Declared in `CFCharacterSet.h`.

Discussion
Use these constants with the `CFCharacterSetGetPredefined` (page 10) function to get one of the predefined character sets.

Declared In
`CFCharacterSet.h`

Document Revision History

This table describes the changes to *CFCharacterSet Reference*.

Date	Notes
2006-12-01	Updated definitions of character sets to include Unicode categories.
2006-04-04	Updated to include new API in Mac OS X v10.5.
	Removed Unicode version number from introduction.
2005-12-06	Corrected links to companion documents.
2005-08-11	Corrected minor typographical errors.
2003-08-01	Added descriptions of new constant and new create-copy method.
2003-01-01	First version of this document.

Document Revision History

Index