
NSLocale Class Reference

[Cocoa](#) > [Internationalization](#)



2008-02-08



Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSLocale Class Reference 5

Overview	5
Tasks	6
Getting and Initializing Locales	6
Getting Information About a Locale	6
Getting System Locale Information	6
Converting Between Identifiers	6
Getting Preferred Languages	7
Class Methods	7
autoupdatingCurrentLocale	7
availableLocaleIdentifiers	7
canonicalLocaleIdentifierFromString:	8
commonISOCurrencyCodes	8
componentsFromLocaleIdentifier:	9
currentLocale	9
ISOCountryCodes	10
ISOCurrencyCodes	10
ISOLanguageCodes	11
localeIdentifierFromComponents:	11
preferredLanguages	12
systemLocale	12
Instance Methods	13
displayNameForKey:value:	13
initWithLocaleIdentifier:	14
localeIdentifier	14
objectForKey:	15
Constants	15
NSLocale Component Keys	15
NSLocale Calendar Keys	18
Notifications	19
NSCurrentLocaleDidChangeNotification	19

Document Revision History 21

Index 23

NSLocale Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	NSLocale.h
Companion guides	Locales Programming Guide Data Formatting Programming Guide for Cocoa
Related sample code	Mountains

Overview

Locales encapsulate information about linguistic, cultural, and technological conventions and standards. Examples of information encapsulated by a locale include the symbol used for the decimal separator in numbers and the way dates are formatted.

Locales are typically used to provide, format, and interpret information about and according to the user's customs and preferences. They are frequently used in conjunction with formatters (see *Data Formatting Programming Guide for Cocoa*). Although you can use many locales, you usually use the one associated with the current user.

`NSLocale` is “toll-free bridged” with its Core Foundation counterpart, `CFLocale`. 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 `NSLocale *` parameter, you can pass a `CFLocaleRef`, and in a function where you see a `CFLocaleRef` parameter, you can pass an `NSLocale` instance (you cast one type to the other to suppress compiler warnings). See *Interchangeable Data Types* for more information on toll-free bridging.

Tasks

Getting and Initializing Locales

- [initWithLocaleIdentifier:](#) (page 14)
Initializes the receiver using a given locale identifier.
- + [systemLocale](#) (page 12)
Returns the “root”, canonical locale, that contains fixed “backstop” settings that provide values for otherwise undefined keys.
- + [currentLocale](#) (page 9)
Returns the logical locale for the current user.
- + [autoupdatingCurrentLocale](#) (page 7)
Returns the current logical locale for the current user.

Getting Information About a Locale

- [displayNameForKey:value:](#) (page 13)
Returns the display name for the given value.
- [localeIdentifier](#) (page 14)
Returns the identifier for the receiver.
- [objectForKey:](#) (page 15)
Returns the object corresponding to the specified key.

Getting System Locale Information

- + [availableLocaleIdentifiers](#) (page 7)
Returns an array of `NSString` objects, each of which identifies a locale available on the system.
- + [ISOCountryCodes](#) (page 10)
Returns an array of `NSString` objects that represents all known legal country codes.
- + [ISOCurrencyCodes](#) (page 10)
Returns an array of `NSString` objects that represents all known legal ISO currency codes.
- + [ISOLanguageCodes](#) (page 11)
Returns an array of `NSString` objects that represents all known legal ISO language codes.
- + [commonISOCurrencyCodes](#) (page 8)
Returns an array of common ISO currency codes

Converting Between Identifiers

- + [canonicalLocaleIdentifierFromString:](#) (page 8)
Returns the canonical identifier for a given locale identification string.
- + [componentsFromLocaleIdentifier:](#) (page 9)
Returns a dictionary that is the result of parsing a locale ID.

+ [localeIdentifierFromComponents:](#) (page 11)

Returns a locale identifier from the components specified in a given dictionary.

Getting Preferred Languages

+ [preferredLanguages](#) (page 12)

Returns the user's language preference order as an array of strings.

Class Methods

autoupdatingCurrentLocale

Returns the current logical locale for the current user.

+ (id)autoupdatingCurrentLocale

Return Value

The current logical locale for the current user. The locale is formed from the settings for the current user's chosen system locale overlaid with any custom settings the user has specified in System Preferences.

The object always reflects the current state of the current user's locale settings.

Discussion

Settings you get from this locale do change as the user's settings change (contrast with [currentLocale](#) (page 9)).

Note that if you cache values based on the locale or related information, those caches will of course not be automatically updated by the updating of the locale object. You can recompute caches upon receipt of the notification (`NSCurrentLocaleDidChangeNotification`) that gets sent out for locale changes (see *Notification Programming Topics for Cocoa* to learn how to register for and receive notifications).

Availability

Available in Mac OS X v10.5 and later.

See Also

+ [systemLocale](#) (page 12)

+ [currentLocale](#) (page 9)

Related Sample Code

Mountains

Declared In

NSLocale.h

availableLocaleIdentifiers

Returns an array of `NSString` objects, each of which identifies a locale available on the system.

+ (NSArray *)availableLocaleIdentifiers

Return Value

An array of `NSString` objects, each of which identifies a locale available on the system.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [ISOLanguageCodes](#) (page 11)
- + [ISOCountryCodes](#) (page 10)
- + [ISOCurrencyCodes](#) (page 10)
- + [commonISOCurrencyCodes](#) (page 8)

Declared In

`NSLocale.h`

canonicalLocaleIdentifierFromString:

Returns the canonical identifier for a given locale identification string.

```
+ (NSString *)canonicalLocaleIdentifierFromString:(NSString *)string
```

Parameters

string

A locale identification string.

Return Value

The canonical identifier for an the locale identified by *string*.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [componentsFromLocaleIdentifier:](#) (page 9)
- + [localeIdentifierFromComponents:](#) (page 11)

Related Sample Code

Mountains

Declared In

`NSLocale.h`

commonISOCurrencyCodes

Returns an array of common ISO currency codes

```
+ (NSArray *)commonISOCurrencyCodes
```

Return Value

An array of `NSString` objects that represents common ISO currency codes.

Discussion

Common codes may include, for example, AED, AUD, BZD, DKK, EUR, GBP, JPY, KES, MXN, OMR, STD, USD, XCD, and ZWD.

Availability

Available in Mac OS X v10.5 and later.

See Also

- + [availableLocaleIdentifiers](#) (page 7)
- + [ISOCountryCodes](#) (page 10)
- + [ISOCurrencyCodes](#) (page 10)

Declared In

NSLocale.h

componentsFromLocaleIdentifier:

Returns a dictionary that is the result of parsing a locale ID.

```
+ (NSDictionary *)componentsFromLocaleIdentifier:(NSString *)string
```

Parameters

string

A locale ID, consisting of language, script, country, variant, and keyword/value pairs, for example, "en_US@calendar=japanese".

Return Value

A dictionary that is the result of parsing *string* as a locale ID. The keys are the constant NSString constants corresponding to the locale ID components, and the values correspond to constants where available. For the complete set of dictionary keys, see “[Constants](#)” (page 15).

Discussion

For example: the locale ID "en_US@calendar=japanese" yields a dictionary with three entries: NSLocaleLanguageCode=en, NSLocaleCountryCode=US, and NSLocaleCalendar=NSJapaneseCalendar.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [localeIdentifierFromComponents:](#) (page 11)
- + [canonicalLocaleIdentifierFromString:](#) (page 8)

Declared In

NSLocale.h

currentLocale

Returns the logical locale for the current user.

```
+ (id)currentLocale
```

Return Value

The logical locale for the current user. The locale is formed from the settings for the current user’s chosen system locale overlaid with any custom settings the user has specified in System Preferences.

This method may return a retained cached object.

Discussion

Settings you get from this locale do not change as System Preferences are changed so that your operations are consistent. Typically you perform some operations on the returned object and then allow it to be disposed of. Moreover, since the returned object may be cached, you do not need to hold on to it indefinitely. Contrast with [autoupdatingCurrentLocale](#) (page 7).

Availability

Available in Mac OS X v10.4 and later.

See Also

+ [systemLocale](#) (page 12)

+ [autoupdatingCurrentLocale](#) (page 7)

Declared In

NSLocale.h

ISOCountryCodes

Returns an array of `NSString` objects that represents all known legal country codes.

```
+ (NSArray *)ISOCountryCodes
```

Return Value

An array of `NSString` objects that represents all known legal country codes.

Discussion

Note that many of country codes do not have any supporting locale data in Mac OS X.

Availability

Available in Mac OS X v10.4 and later.

See Also

+ [availableLocaleIdentifiers](#) (page 7)

+ [ISOLanguageCodes](#) (page 11)

+ [ISOCurrencyCodes](#) (page 10)

+ [commonISOCurrencyCodes](#) (page 8)

Declared In

NSLocale.h

ISOCurrencyCodes

Returns an array of `NSString` objects that represents all known legal ISO currency codes.

```
+ (NSArray *)ISOCurrencyCodes
```

Return Value

An array of `NSString` objects that represents all known legal ISO currency codes.

Discussion

Note that some of the currency codes may not have any supporting locale data in Mac OS X.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [availableLocaleIdentifiers](#) (page 7)
- + [ISOCountryCodes](#) (page 10)
- + [ISOLanguageCodes](#) (page 11)
- + [commonISOCurrencyCodes](#) (page 8)

Declared In

NSLocale.h

ISOLanguageCodes

Returns an array of `NSString` objects that represents all known legal ISO language codes.

```
+ (NSArray *)ISOLanguageCodes
```

Return Value

An array of `NSString` objects that represents all known legal ISO language codes.

Discussion

Note that many of the language codes will not have any supporting locale data in Mac OS X.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [availableLocaleIdentifiers](#) (page 7)
- + [ISOCountryCodes](#) (page 10)
- + [ISOCurrencyCodes](#) (page 10)
- + [commonISOCurrencyCodes](#) (page 8)

Declared In

NSLocale.h

localeIdentifierFromComponents:

Returns a locale identifier from the components specified in a given dictionary.

```
+ (NSString *)localeIdentifierFromComponents:(NSDictionary *)dict
```

Parameters

dict

A dictionary containing components that specify a locale. For valid dictionary keys, see [“Constants”](#) (page 15).

Return Value

A locale identifier created from the components specified in *dict*.

Discussion

This reverses the actions of [componentsFromLocaleIdentifier:](#) (page 9), so for example the dictionary `{NSLocaleLanguageCode="en", NSLocaleCountryCode="US", NSLocaleCalendar=NSJapaneseCalendar}` becomes `"en_US@calendar=japanese"`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [componentsFromLocaleIdentifier:](#) (page 9)
- + [canonicalLocaleIdentifierFromString:](#) (page 8)
- + [ISOLanguageCodes](#) (page 11)

Declared In

NSLocale.h

preferredLanguages

Returns the user's language preference order as an array of strings.

```
+ (NSArray *)preferredLanguages
```

Return Value

The user's language preference order as an array of `NSString` objects, each of which is a canonicalized IETF BCP 47 language identifier.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

Mountains

Declared In

NSLocale.h

systemLocale

Returns the “root”, canonical locale, that contains fixed “backstop” settings that provide values for otherwise undefined keys.

```
+ (id)systemLocale
```

Return Value

The “root”, canonical locale, that contains fixed “backstop” settings that provide values for otherwise undefined keys.

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [autoupdatingCurrentLocale](#) (page 7)
- + [autoupdatingCurrentLocale](#) (page 7)

Declared In
NSLocale.h

Instance Methods

displayNameForKey:value:

Returns the display name for the given value.

```
- (NSString *)displayNameForKey:(id)key value:(id)value
```

Parameters

key

Specifies which of the locale property keys *value* is (see “Constants” (page 15)),

value

A value for *key*.

Return Value

The display name for *value*.

Discussion

Not all locale property keys have values with display name values.

You can use the `NSLocaleIdentifier` key to get the name of a locale in the language of another locale, as illustrated in the following examples. The first uses the `fr_FR` locale.

```
NSLocale *frLocale = [[[NSLocale alloc] initWithLocaleIdentifier:@"fr_FR"]
autorelease];
NSString *displayNameString = [frLocale displayNameForKey:NSLocaleIdentifier
value:@"fr_FR"];
NSLog(@"displayNameString fr_FR: %@", displayNameString);
displayNameString = [frLocale displayNameForKey:NSLocaleIdentifier
value:@"en_US"];
NSLog(@"displayNameString en_US: %@", displayNameString);
```

returns

```
displayNameString fr_FR: français (France)
displayNameString en_US: anglais (États-Unis)
```

The following example uses the `en_GB` locale.

```
NSLocale *gbLocale = [[[NSLocale alloc] initWithLocaleIdentifier:@"en_GB"]
autorelease];
displayNameString = [gbLocale displayNameForKey:NSLocaleIdentifier
value:@"fr_FR"];
NSLog(@"displayNameString fr_FR: %@", displayNameString);
displayNameString = [gbLocale displayNameForKey:NSLocaleIdentifier
value:@"en_US"];
NSLog(@"displayNameString en_US: %@", displayNameString);
```

returns

```
displayNameString fr_FR: French (France)
```

`displayNameString en_US: English (United States)`

Availability

Available in Mac OS X v10.4 and later.

See Also

- [localeIdentifier](#) (page 14)

Declared In

NSLocale.h

initWithLocaleIdentifier:

Initializes the receiver using a given locale identifier.

```
- (id)initWithLocaleIdentifier:(NSString *)string
```

Parameters

string

The identifier for the new locale.

Return Value

The initialized locale.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

Mountains

Declared In

NSLocale.h

localeIdentifier

Returns the identifier for the receiver.

```
- (NSString *)localeIdentifier
```

Return Value

The identifier for the receiver. This may not be the same string that the locale was created with, since `NSLocale` may canonicalize it.

Discussion

Equivalent to sending `objectForKey:withKey NSLocaleIdentifier`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displayNameForKey:value:](#) (page 13)

Related Sample Code

Mountains

Declared In
NSLocale.h

objectForKey:

Returns the object corresponding to the specified key.

- (id)objectForKey:(id)key

Parameters

key

The key for which to return the corresponding value. For valid values of *key*, see “Constants” (page 15).

Return Value

The object corresponding to *key*.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displayNameForKey:value:](#) (page 13)

Declared In
NSLocale.h

Constants

NSLocale Component Keys

The following constants specify keys used to retrieve components of a locale with [objectForKey:](#) (page 15).

```
extern NSString * const NSLocaleIdentifier;
extern NSString * const NSLocaleLanguageCode;
extern NSString * const NSLocaleCountryCode;
extern NSString * const NSLocaleScriptCode;
extern NSString * const NSLocaleVariantCode;
extern NSString * const NSLocaleExemplarCharacterSet;
extern NSString * const NSLocaleCalendar;
extern NSString * const NSLocaleCollationIdentifier;
extern NSString * const NSLocaleUsesMetricSystem;
extern NSString * const NSLocaleMeasurementSystem;
extern NSString * const NSLocaleDecimalSeparator;
extern NSString * const NSLocaleGroupingSeparator;
extern NSString * const NSLocaleCurrencySymbol;
extern NSString * const NSLocaleCurrencyCode;
```

Constants

NSLocaleIdentifier

The key for the locale identifier.

The corresponding value is an NSString object. An example value might be "es_ES_PREEURO".

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleLanguageCode

The key for the locale language code.

The corresponding value is an NSString object. An example value might be "es".

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleCountryCode

The key for the locale country code.

The corresponding value is an NSString object. An example value might be "ES".

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleScriptCode

The key for the locale script code.

The corresponding value is an NSString object.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleVariantCode

The key for the locale variant code.

The corresponding value is an NSString object. An example value might be "PREEURO".

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleExemplarCharacterSet

The key for the exemplar character set for the locale.

The corresponding value is an NSCharacterSet object.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSLocaleCalendar

The key for the calendar associated with the locale.

The corresponding value is an `NSCalendar` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleCollationIdentifier

The key for the collation associated with the locale.

The corresponding value is an `NSString` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleUsesMetricSystem

The key for the flag that indicates whether the locale uses the metric system.

The corresponding value is a Boolean `NSNumber` object. If the value is `NO`, you can typically assume American measurement units (for example, the statute mile).

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleMeasurementSystem

The key for the measurement system associated with the locale.

The corresponding value is an `NSString` object containing a description of the measurement system used by the locale, for example "Metric" or "U.S."

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleDecimalSeparator

The key for the decimal separator associated with the locale.

The corresponding value is an `NSString` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleGroupingSeparator

The key for the numeric grouping separator associated with the locale.

The corresponding value is an `NSString` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleCurrencySymbol

The key for the currency symbol associated with the locale.

The corresponding value is an `NSString` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

NSLocaleCurrencyCode

The key for the currency code associated with the locale.

The corresponding value is an `NSString` object.

Available in Mac OS X v10.4 and later.

Declared in `NSLocale.h`.

Declared In

NSLocale.h

NSLocale Calendar Keys

These constants identify NSCalendar instances.

```
extern NSString * const NSGregorianCalendar;
extern NSString * const NSBuddhistCalendar;
extern NSString * const NSChineseCalendar;
extern NSString * const NSHebrewCalendar;
extern NSString * const NSIslamicCalendar;
extern NSString * const NSIslamicCivilCalendar;
extern NSString * const NSJapaneseCalendar;
```

Constants

NSGregorianCalendar

Identifier for the Gregorian calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSBuddhistCalendar

Identifier for the Buddhist calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSChineseCalendar

Identifier for the Chinese calendar (unsupported).

Note that the Chinese calendar is not supported in Mac OS X v10.4-10.5. Although you can create a calendar using this constant, the object will not function correctly.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSHebrewCalendar

Identifier for the Hebrew calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSIslamicCalendar

Identifier for the Islamic calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSIslamicCivilCalendar

Identifier for the Islamic civil calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

NSJapaneseCalendar

Identifier for the Japanese calendar.

Available in Mac OS X v10.4 and later.

Declared in NSLocale.h.

Discussion

You use these identifiers to initialize a new `NSCalendar` object, using `initWithCalendarIdentifier:`. You get one of these identifiers as the return value from `calendarIdentifier`.

Declared In

`NSLocale.h`

Notifications

NSCurrentLocaleDidChangeNotification

Notification that indicates that the user's locale changed.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSLocale.h`

Document Revision History

This table describes the changes to *NSLocale Class Reference*.

Date	Notes
2008-02-08	Enhanced the description of the <code>NSLocaleMeasurementSystem</code> constant.
2007-10-31	Clarified the return value of the <code>preferredLanguages</code> method.
2007-05-21	Updated to include API introduced in Mac OS X v10.5.
2006-10-03	Added a note to <code>NSChineseCalendar</code> stating that the Chinese calendar is not supported.
2006-05-23	Clarified use of <code>displayNameForKey:value:</code> .
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

autoupdatingCurrentLocale **class method** [7](#)
availableLocaleIdentifiers **class method** [7](#)

C

canonicalLocaleIdentifierFromString: **class method** [8](#)
commonISOCurrencyCodes **class method** [8](#)
componentsFromLocaleIdentifier: **class method** [9](#)
currentLocale **class method** [9](#)

D

displayNameForKey:value: **instance method** [13](#)

I

initWithLocaleIdentifier: **instance method** [14](#)
ISOCountryCodes **class method** [10](#)
ISOCurrencyCodes **class method** [10](#)
ISOLanguageCodes **class method** [11](#)

L

localeIdentifier **instance method** [14](#)
localeIdentifierFromComponents: **class method** [11](#)

N

NSBuddhistCalendar **constant** [18](#)

NSChineseCalendar **constant** [18](#)
NSCurrentLocaleDidChangeNotification **notification** [19](#)

NSGregorianCalendar **constant** [18](#)
NSHebrewCalendar **constant** [18](#)
NSIslamicCalendar **constant** [18](#)
NSIslamicCivilCalendar **constant** [18](#)
NSJapaneseCalendar **constant** [18](#)
NSLocale Calendar Keys [18](#)
NSLocale Component Keys [15](#)
NSLocaleCalendar **constant** [17](#)
NSLocaleCollationIdentifier **constant** [17](#)
NSLocaleCountryCode **constant** [16](#)
NSLocaleCurrencyCode **constant** [17](#)
NSLocaleCurrencySymbol **constant** [17](#)
NSLocaleDecimalSeparator **constant** [17](#)
NSLocaleExemplarCharacterSet **constant** [16](#)
NSLocaleGroupingSeparator **constant** [17](#)
NSLocaleIdentifier **constant** [16](#)
NSLocaleLanguageCode **constant** [16](#)
NSLocaleMeasurementSystem **constant** [17](#)
NSLocaleScriptCode **constant** [16](#)
NSLocaleUsesMetricSystem **constant** [17](#)
NSLocaleVariantCode **constant** [16](#)

O

objectForKey: **instance method** [15](#)

P

preferredLanguages **class method** [12](#)

S

systemLocale **class method** [12](#)