| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Locale.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/ListResourceBundle.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Map.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Locale.html)    [**NO FRAMES**](http://docs.google.com/Locale.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2p2csry) | [METHOD](#ihv636) |

## **java.util**

Class Locale

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.util.Locale**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

public final class **Locale**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A Locale object represents a specific geographical, political, or cultural region. An operation that requires a Locale to perform its task is called *locale-sensitive* and uses the Locale to tailor information for the user. For example, displaying a number is a locale-sensitive operation--the number should be formatted according to the customs/conventions of the user's native country, region, or culture.

Create a Locale object using the constructors in this class:

Locale(String language)  
 Locale(String language, String country)  
 Locale(String language, String country, String variant)

The language argument is a valid **ISO Language Code.** These codes are the lower-case, two-letter codes as defined by ISO-639. You can find a full list of these codes at a number of sites, such as:

<http://www.loc.gov/standards/iso639-2/englangn.html>

The country argument is a valid **ISO Country Code.** These codes are the upper-case, two-letter codes as defined by ISO-3166. You can find a full list of these codes at a number of sites, such as:

<http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html>

The variant argument is a vendor or browser-specific code. For example, use WIN for Windows, MAC for Macintosh, and POSIX for POSIX. Where there are two variants, separate them with an underscore, and put the most important one first. For example, a Traditional Spanish collation might construct a locale with parameters for language, country and variant as: "es", "ES", "Traditional\_WIN".

Because a Locale object is just an identifier for a region, no validity check is performed when you construct a Locale. If you want to see whether particular resources are available for the Locale you construct, you must query those resources. For example, ask the NumberFormat for the locales it supports using its getAvailableLocales method.

**Note:** When you ask for a resource for a particular locale, you get back the best available match, not necessarily precisely what you asked for. For more information, look at [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html).

The Locale class provides a number of convenient constants that you can use to create Locale objects for commonly used locales. For example, the following creates a Locale object for the United States:

Locale.US

Once you've created a Locale you can query it for information about itself. Use getCountry to get the ISO Country Code and getLanguage to get the ISO Language Code. You can use getDisplayCountry to get the name of the country suitable for displaying to the user. Similarly, you can use getDisplayLanguage to get the name of the language suitable for displaying to the user. Interestingly, the getDisplayXXX methods are themselves locale-sensitive and have two versions: one that uses the default locale and one that uses the locale specified as an argument.

The Java Platform provides a number of classes that perform locale-sensitive operations. For example, the NumberFormat class formats numbers, currency, or percentages in a locale-sensitive manner. Classes such as NumberFormat have a number of convenience methods for creating a default object of that type. For example, the NumberFormat class provides these three convenience methods for creating a default NumberFormat object:

NumberFormat.getInstance()  
 NumberFormat.getCurrencyInstance()  
 NumberFormat.getPercentInstance()

These methods have two variants; one with an explicit locale and one without; the latter using the default locale.

NumberFormat.getInstance(myLocale)  
 NumberFormat.getCurrencyInstance(myLocale)  
 NumberFormat.getPercentInstance(myLocale)

A Locale is the mechanism for identifying the kind of object (NumberFormat) that you would like to get. The locale is **just** a mechanism for identifying objects, **not** a container for the objects themselves.

**Since:** 1.1 **See Also:**[ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html), [Format](http://docs.google.com/java/text/Format.html), [NumberFormat](http://docs.google.com/java/text/NumberFormat.html), [Collator](http://docs.google.com/java/text/Collator.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.Locale)

| **Field Summary** | |
| --- | --- |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**CANADA**](http://docs.google.com/java/util/Locale.html#CANADA)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**CANADA\_FRENCH**](http://docs.google.com/java/util/Locale.html#CANADA_FRENCH)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**CHINA**](http://docs.google.com/java/util/Locale.html#CHINA)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**CHINESE**](http://docs.google.com/java/util/Locale.html#CHINESE)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**ENGLISH**](http://docs.google.com/java/util/Locale.html#ENGLISH)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**FRANCE**](http://docs.google.com/java/util/Locale.html#FRANCE)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**FRENCH**](http://docs.google.com/java/util/Locale.html#FRENCH)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**GERMAN**](http://docs.google.com/java/util/Locale.html#GERMAN)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**GERMANY**](http://docs.google.com/java/util/Locale.html#GERMANY)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**ITALIAN**](http://docs.google.com/java/util/Locale.html#ITALIAN)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**ITALY**](http://docs.google.com/java/util/Locale.html#ITALY)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**JAPAN**](http://docs.google.com/java/util/Locale.html#JAPAN)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**JAPANESE**](http://docs.google.com/java/util/Locale.html#JAPANESE)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**KOREA**](http://docs.google.com/java/util/Locale.html#KOREA)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**KOREAN**](http://docs.google.com/java/util/Locale.html#KOREAN)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**PRC**](http://docs.google.com/java/util/Locale.html#PRC)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**ROOT**](http://docs.google.com/java/util/Locale.html#ROOT)            Useful constant for the root locale. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**SIMPLIFIED\_CHINESE**](http://docs.google.com/java/util/Locale.html#SIMPLIFIED_CHINESE)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**TAIWAN**](http://docs.google.com/java/util/Locale.html#TAIWAN)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**TRADITIONAL\_CHINESE**](http://docs.google.com/java/util/Locale.html#TRADITIONAL_CHINESE)            Useful constant for language. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**UK**](http://docs.google.com/java/util/Locale.html#UK)            Useful constant for country. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**US**](http://docs.google.com/java/util/Locale.html#US)            Useful constant for country. |

| **Constructor Summary** | |
| --- | --- |
| [**Locale**](http://docs.google.com/java/util/Locale.html#Locale(java.lang.String))([String](http://docs.google.com/java/lang/String.html) language)            Construct a locale from a language code. |
| [**Locale**](http://docs.google.com/java/util/Locale.html#Locale(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) language, [String](http://docs.google.com/java/lang/String.html) country)            Construct a locale from language, country. |
| [**Locale**](http://docs.google.com/java/util/Locale.html#Locale(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) language, [String](http://docs.google.com/java/lang/String.html) country, [String](http://docs.google.com/java/lang/String.html) variant)            Construct a locale from language, country, variant. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/Locale.html#clone())()            Overrides Cloneable |
| boolean | [**equals**](http://docs.google.com/java/util/Locale.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Returns true if this Locale is equal to another object. |
| static [Locale](http://docs.google.com/java/util/Locale.html)[] | [**getAvailableLocales**](http://docs.google.com/java/util/Locale.html#getAvailableLocales())()            Returns an array of all installed locales. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCountry**](http://docs.google.com/java/util/Locale.html#getCountry())()            Returns the country/region code for this locale, which will either be the empty string or an uppercase ISO 3166 2-letter code. |
| static [Locale](http://docs.google.com/java/util/Locale.html) | [**getDefault**](http://docs.google.com/java/util/Locale.html#getDefault())()            Gets the current value of the default locale for this instance of the Java Virtual Machine. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayCountry**](http://docs.google.com/java/util/Locale.html#getDisplayCountry())()            Returns a name for the locale's country that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayCountry**](http://docs.google.com/java/util/Locale.html#getDisplayCountry(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) inLocale)            Returns a name for the locale's country that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayLanguage**](http://docs.google.com/java/util/Locale.html#getDisplayLanguage())()            Returns a name for the locale's language that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayLanguage**](http://docs.google.com/java/util/Locale.html#getDisplayLanguage(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) inLocale)            Returns a name for the locale's language that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/Locale.html#getDisplayName())()            Returns a name for the locale that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/Locale.html#getDisplayName(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) inLocale)            Returns a name for the locale that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayVariant**](http://docs.google.com/java/util/Locale.html#getDisplayVariant())()            Returns a name for the locale's variant code that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayVariant**](http://docs.google.com/java/util/Locale.html#getDisplayVariant(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) inLocale)            Returns a name for the locale's variant code that is appropriate for display to the user. |
| [String](http://docs.google.com/java/lang/String.html) | [**getISO3Country**](http://docs.google.com/java/util/Locale.html#getISO3Country())()            Returns a three-letter abbreviation for this locale's country. |
| [String](http://docs.google.com/java/lang/String.html) | [**getISO3Language**](http://docs.google.com/java/util/Locale.html#getISO3Language())()            Returns a three-letter abbreviation for this locale's language. |
| static [String](http://docs.google.com/java/lang/String.html)[] | [**getISOCountries**](http://docs.google.com/java/util/Locale.html#getISOCountries())()            Returns a list of all 2-letter country codes defined in ISO 3166. |
| static [String](http://docs.google.com/java/lang/String.html)[] | [**getISOLanguages**](http://docs.google.com/java/util/Locale.html#getISOLanguages())()            Returns a list of all 2-letter language codes defined in ISO 639. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLanguage**](http://docs.google.com/java/util/Locale.html#getLanguage())()            Returns the language code for this locale, which will either be the empty string or a lowercase ISO 639 code. |
| [String](http://docs.google.com/java/lang/String.html) | [**getVariant**](http://docs.google.com/java/util/Locale.html#getVariant())()            Returns the variant code for this locale. |
| int | [**hashCode**](http://docs.google.com/java/util/Locale.html#hashCode())()            Override hashCode. |
| static void | [**setDefault**](http://docs.google.com/java/util/Locale.html#setDefault(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) newLocale)            Sets the default locale for this instance of the Java Virtual Machine. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/Locale.html#toString())()            Getter for the programmatic name of the entire locale, with the language, country and variant separated by underbars. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### ENGLISH

public static final [Locale](http://docs.google.com/java/util/Locale.html) **ENGLISH**

Useful constant for language.

### FRENCH

public static final [Locale](http://docs.google.com/java/util/Locale.html) **FRENCH**

Useful constant for language.

### GERMAN

public static final [Locale](http://docs.google.com/java/util/Locale.html) **GERMAN**

Useful constant for language.

### ITALIAN

public static final [Locale](http://docs.google.com/java/util/Locale.html) **ITALIAN**

Useful constant for language.

### JAPANESE

public static final [Locale](http://docs.google.com/java/util/Locale.html) **JAPANESE**

Useful constant for language.

### KOREAN

public static final [Locale](http://docs.google.com/java/util/Locale.html) **KOREAN**

Useful constant for language.

### CHINESE

public static final [Locale](http://docs.google.com/java/util/Locale.html) **CHINESE**

Useful constant for language.

### SIMPLIFIED\_CHINESE

public static final [Locale](http://docs.google.com/java/util/Locale.html) **SIMPLIFIED\_CHINESE**

Useful constant for language.

### TRADITIONAL\_CHINESE

public static final [Locale](http://docs.google.com/java/util/Locale.html) **TRADITIONAL\_CHINESE**

Useful constant for language.

### FRANCE

public static final [Locale](http://docs.google.com/java/util/Locale.html) **FRANCE**

Useful constant for country.

### GERMANY

public static final [Locale](http://docs.google.com/java/util/Locale.html) **GERMANY**

Useful constant for country.

### ITALY

public static final [Locale](http://docs.google.com/java/util/Locale.html) **ITALY**

Useful constant for country.

### JAPAN

public static final [Locale](http://docs.google.com/java/util/Locale.html) **JAPAN**

Useful constant for country.

### KOREA

public static final [Locale](http://docs.google.com/java/util/Locale.html) **KOREA**

Useful constant for country.

### CHINA

public static final [Locale](http://docs.google.com/java/util/Locale.html) **CHINA**

Useful constant for country.

### PRC

public static final [Locale](http://docs.google.com/java/util/Locale.html) **PRC**

Useful constant for country.

### TAIWAN

public static final [Locale](http://docs.google.com/java/util/Locale.html) **TAIWAN**

Useful constant for country.

### UK

public static final [Locale](http://docs.google.com/java/util/Locale.html) **UK**

Useful constant for country.

### US

public static final [Locale](http://docs.google.com/java/util/Locale.html) **US**

Useful constant for country.

### CANADA

public static final [Locale](http://docs.google.com/java/util/Locale.html) **CANADA**

Useful constant for country.

### CANADA\_FRENCH

public static final [Locale](http://docs.google.com/java/util/Locale.html) **CANADA\_FRENCH**

Useful constant for country.

### ROOT

public static final [Locale](http://docs.google.com/java/util/Locale.html) **ROOT**

Useful constant for the root locale. The root locale is the locale whose language, country, and variant are empty ("") strings. This is regarded as the base locale of all locales, and is used as the language/country neutral locale for the locale sensitive operations.

**Since:** 1.6

| **Constructor Detail** |
| --- |

### Locale

public **Locale**([String](http://docs.google.com/java/lang/String.html) language,  
 [String](http://docs.google.com/java/lang/String.html) country,  
 [String](http://docs.google.com/java/lang/String.html) variant)

Construct a locale from language, country, variant. NOTE: ISO 639 is not a stable standard; some of the language codes it defines (specifically iw, ji, and in) have changed. This constructor accepts both the old codes (iw, ji, and in) and the new codes (he, yi, and id), but all other API on Locale will return only the OLD codes.

**Parameters:**language - lowercase two-letter ISO-639 code.country - uppercase two-letter ISO-3166 code.variant - vendor and browser specific code. See class description. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - thrown if any argument is null.

### Locale

public **Locale**([String](http://docs.google.com/java/lang/String.html) language,  
 [String](http://docs.google.com/java/lang/String.html) country)

Construct a locale from language, country. NOTE: ISO 639 is not a stable standard; some of the language codes it defines (specifically iw, ji, and in) have changed. This constructor accepts both the old codes (iw, ji, and in) and the new codes (he, yi, and id), but all other API on Locale will return only the OLD codes.

**Parameters:**language - lowercase two-letter ISO-639 code.country - uppercase two-letter ISO-3166 code. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - thrown if either argument is null.

### Locale

public **Locale**([String](http://docs.google.com/java/lang/String.html) language)

Construct a locale from a language code. NOTE: ISO 639 is not a stable standard; some of the language codes it defines (specifically iw, ji, and in) have changed. This constructor accepts both the old codes (iw, ji, and in) and the new codes (he, yi, and id), but all other API on Locale will return only the OLD codes.

**Parameters:**language - lowercase two-letter ISO-639 code. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - thrown if argument is null.**Since:** 1.4

| **Method Detail** |
| --- |

### getDefault

public static [Locale](http://docs.google.com/java/util/Locale.html) **getDefault**()

Gets the current value of the default locale for this instance of the Java Virtual Machine.

The Java Virtual Machine sets the default locale during startup based on the host environment. It is used by many locale-sensitive methods if no locale is explicitly specified. It can be changed using the [setDefault](http://docs.google.com/java/util/Locale.html#setDefault(java.util.Locale)) method.

**Returns:**the default locale for this instance of the Java Virtual Machine

### setDefault

public static void **setDefault**([Locale](http://docs.google.com/java/util/Locale.html) newLocale)

Sets the default locale for this instance of the Java Virtual Machine. This does not affect the host locale.

If there is a security manager, its checkPermission method is called with a PropertyPermission("user.language", "write") permission before the default locale is changed.

The Java Virtual Machine sets the default locale during startup based on the host environment. It is used by many locale-sensitive methods if no locale is explicitly specified.

Since changing the default locale may affect many different areas of functionality, this method should only be used if the caller is prepared to reinitialize locale-sensitive code running within the same Java Virtual Machine.

**Parameters:**newLocale - the new default locale **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow the operation. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if newLocale is null**See Also:**[SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [PropertyPermission](http://docs.google.com/java/util/PropertyPermission.html)

### getAvailableLocales

public static [Locale](http://docs.google.com/java/util/Locale.html)[] **getAvailableLocales**()

Returns an array of all installed locales. The returned array represents the union of locales supported by the Java runtime environment and by installed [LocaleServiceProvider](http://docs.google.com/java/util/spi/LocaleServiceProvider.html) implementations. It must contain at least a Locale instance equal to [Locale.US](http://docs.google.com/java/util/Locale.html#US).

**Returns:**An array of installed locales.

### getISOCountries

public static [String](http://docs.google.com/java/lang/String.html)[] **getISOCountries**()

Returns a list of all 2-letter country codes defined in ISO 3166. Can be used to create Locales.

### getISOLanguages

public static [String](http://docs.google.com/java/lang/String.html)[] **getISOLanguages**()

Returns a list of all 2-letter language codes defined in ISO 639. Can be used to create Locales. [NOTE: ISO 639 is not a stable standard-- some languages' codes have changed. The list this function returns includes both the new and the old codes for the languages whose codes have changed.]

### getLanguage

public [String](http://docs.google.com/java/lang/String.html) **getLanguage**()

Returns the language code for this locale, which will either be the empty string or a lowercase ISO 639 code.

NOTE: ISO 639 is not a stable standard-- some languages' codes have changed. Locale's constructor recognizes both the new and the old codes for the languages whose codes have changed, but this function always returns the old code. If you want to check for a specific language whose code has changed, don't do

if (locale.getLanguage().equals("he"))   
 ...

Instead, do

if (locale.getLanguage().equals(new Locale("he", "", "").getLanguage()))   
 ...

**See Also:**[getDisplayLanguage()](http://docs.google.com/java/util/Locale.html#getDisplayLanguage())

### getCountry

public [String](http://docs.google.com/java/lang/String.html) **getCountry**()

Returns the country/region code for this locale, which will either be the empty string or an uppercase ISO 3166 2-letter code.

**See Also:**[getDisplayCountry()](http://docs.google.com/java/util/Locale.html#getDisplayCountry())

### getVariant

public [String](http://docs.google.com/java/lang/String.html) **getVariant**()

Returns the variant code for this locale.

**See Also:**[getDisplayVariant()](http://docs.google.com/java/util/Locale.html#getDisplayVariant())

### toString

public final [String](http://docs.google.com/java/lang/String.html) **toString**()

Getter for the programmatic name of the entire locale, with the language, country and variant separated by underbars. Language is always lower case, and country is always upper case. If the language is missing, the string will begin with an underbar. If both the language and country fields are missing, this function will return the empty string, even if the variant field is filled in (you can't have a locale with just a variant-- the variant must accompany a valid language or country code). Examples: "en", "de\_DE", "\_GB", "en\_US\_WIN", "de\_\_POSIX", "fr\_\_MAC"

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the object.**See Also:**[getDisplayName()](http://docs.google.com/java/util/Locale.html#getDisplayName())

### getISO3Language

public [String](http://docs.google.com/java/lang/String.html) **getISO3Language**()  
 throws [MissingResourceException](http://docs.google.com/java/util/MissingResourceException.html)

Returns a three-letter abbreviation for this locale's language. If the locale doesn't specify a language, this will be the empty string. Otherwise, this will be a lowercase ISO 639-2/T language code. The ISO 639-2 language codes can be found on-line at  [http://www.loc.gov/standards/iso639-2/englangn.html.](http://www.loc.gov/standards/iso639-2/englangn.html)

**Throws:** [MissingResourceException](http://docs.google.com/java/util/MissingResourceException.html) - Throws MissingResourceException if the three-letter language abbreviation is not available for this locale.

### getISO3Country

public [String](http://docs.google.com/java/lang/String.html) **getISO3Country**()  
 throws [MissingResourceException](http://docs.google.com/java/util/MissingResourceException.html)

Returns a three-letter abbreviation for this locale's country. If the locale doesn't specify a country, this will be the empty string. Otherwise, this will be an uppercase ISO 3166 3-letter country code. The ISO 3166-2 country codes can be found on-line at  [http://www.davros.org/misc/iso3166.txt.](http://www.davros.org/misc/iso3166.txt)

**Throws:** [MissingResourceException](http://docs.google.com/java/util/MissingResourceException.html) - Throws MissingResourceException if the three-letter country abbreviation is not available for this locale.

### getDisplayLanguage

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayLanguage**()

Returns a name for the locale's language that is appropriate for display to the user. If possible, the name returned will be localized for the default locale. For example, if the locale is fr\_FR and the default locale is en\_US, getDisplayLanguage() will return "French"; if the locale is en\_US and the default locale is fr\_FR, getDisplayLanguage() will return "anglais". If the name returned cannot be localized for the default locale, (say, we don't have a Japanese name for Croatian), this function falls back on the English name, and uses the ISO code as a last-resort value. If the locale doesn't specify a language, this function returns the empty string.

### getDisplayLanguage

public [String](http://docs.google.com/java/lang/String.html) **getDisplayLanguage**([Locale](http://docs.google.com/java/util/Locale.html) inLocale)

Returns a name for the locale's language that is appropriate for display to the user. If possible, the name returned will be localized according to inLocale. For example, if the locale is fr\_FR and inLocale is en\_US, getDisplayLanguage() will return "French"; if the locale is en\_US and inLocale is fr\_FR, getDisplayLanguage() will return "anglais". If the name returned cannot be localized according to inLocale, (say, we don't have a Japanese name for Croatian), this function falls back on the English name, and finally on the ISO code as a last-resort value. If the locale doesn't specify a language, this function returns the empty string.

**Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if inLocale is null

### getDisplayCountry

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayCountry**()

Returns a name for the locale's country that is appropriate for display to the user. If possible, the name returned will be localized for the default locale. For example, if the locale is fr\_FR and the default locale is en\_US, getDisplayCountry() will return "France"; if the locale is en\_US and the default locale is fr\_FR, getDisplayCountry() will return "Etats-Unis". If the name returned cannot be localized for the default locale, (say, we don't have a Japanese name for Croatia), this function falls back on the English name, and uses the ISO code as a last-resort value. If the locale doesn't specify a country, this function returns the empty string.

### getDisplayCountry

public [String](http://docs.google.com/java/lang/String.html) **getDisplayCountry**([Locale](http://docs.google.com/java/util/Locale.html) inLocale)

Returns a name for the locale's country that is appropriate for display to the user. If possible, the name returned will be localized according to inLocale. For example, if the locale is fr\_FR and inLocale is en\_US, getDisplayCountry() will return "France"; if the locale is en\_US and inLocale is fr\_FR, getDisplayCountry() will return "Etats-Unis". If the name returned cannot be localized according to inLocale. (say, we don't have a Japanese name for Croatia), this function falls back on the English name, and finally on the ISO code as a last-resort value. If the locale doesn't specify a country, this function returns the empty string.

**Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if inLocale is null

### getDisplayVariant

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayVariant**()

Returns a name for the locale's variant code that is appropriate for display to the user. If possible, the name will be localized for the default locale. If the locale doesn't specify a variant code, this function returns the empty string.

### getDisplayVariant

public [String](http://docs.google.com/java/lang/String.html) **getDisplayVariant**([Locale](http://docs.google.com/java/util/Locale.html) inLocale)

Returns a name for the locale's variant code that is appropriate for display to the user. If possible, the name will be localized for inLocale. If the locale doesn't specify a variant code, this function returns the empty string.

**Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if inLocale is null

### getDisplayName

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayName**()

Returns a name for the locale that is appropriate for display to the user. This will be the values returned by getDisplayLanguage(), getDisplayCountry(), and getDisplayVariant() assembled into a single string. The display name will have one of the following forms:

language (country, variant)

language (country)

language (variant)

country (variant)

language

country

variant

depending on which fields are specified in the locale. If the language, country, and variant fields are all empty, this function returns the empty string.

### getDisplayName

public [String](http://docs.google.com/java/lang/String.html) **getDisplayName**([Locale](http://docs.google.com/java/util/Locale.html) inLocale)

Returns a name for the locale that is appropriate for display to the user. This will be the values returned by getDisplayLanguage(), getDisplayCountry(), and getDisplayVariant() assembled into a single string. The display name will have one of the following forms:

language (country, variant)

language (country)

language (variant)

country (variant)

language

country

variant

depending on which fields are specified in the locale. If the language, country, and variant fields are all empty, this function returns the empty string.

**Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if inLocale is null

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Overrides Cloneable

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### hashCode

public int **hashCode**()

Override hashCode. Since Locales are often used in hashtables, caches the value for speed.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Returns true if this Locale is equal to another object. A Locale is deemed equal to another Locale with identical language, country, and variant, and unequal to all other objects.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the reference object with which to compare. **Returns:**true if this Locale is equal to the specified object.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Locale.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/ListResourceBundle.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Map.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Locale.html)    [**NO FRAMES**](http://docs.google.com/Locale.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2p2csry) | [METHOD](#ihv636) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).