| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LocaleServiceProvider.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/spi/LocaleNameProvider.html)   [**NEXT CLASS**](http://docs.google.com/java/util/spi/TimeZoneNameProvider.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/spi/LocaleServiceProvider.html)    [**NO FRAMES**](http://docs.google.com/LocaleServiceProvider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.util.spi**

Class LocaleServiceProvider

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

**Direct Known Subclasses:** [BreakIteratorProvider](http://docs.google.com/java/text/spi/BreakIteratorProvider.html), [CollatorProvider](http://docs.google.com/java/text/spi/CollatorProvider.html), [CurrencyNameProvider](http://docs.google.com/java/util/spi/CurrencyNameProvider.html), [DateFormatProvider](http://docs.google.com/java/text/spi/DateFormatProvider.html), [DateFormatSymbolsProvider](http://docs.google.com/java/text/spi/DateFormatSymbolsProvider.html), [DecimalFormatSymbolsProvider](http://docs.google.com/java/text/spi/DecimalFormatSymbolsProvider.html), [LocaleNameProvider](http://docs.google.com/java/util/spi/LocaleNameProvider.html), [NumberFormatProvider](http://docs.google.com/java/text/spi/NumberFormatProvider.html), [TimeZoneNameProvider](http://docs.google.com/java/util/spi/TimeZoneNameProvider.html)

public abstract class **LocaleServiceProvider**extends [Object](http://docs.google.com/java/lang/Object.html)

This is the super class of all the locale sensitive service provider interfaces (SPIs).

Locale sensitive service provider interfaces are interfaces that correspond to locale sensitive classes in the java.text and java.util packages. The interfaces enable the construction of locale sensitive objects and the retrieval of localized names for these packages. Locale sensitive factory methods and methods for name retrieval in the java.text and java.util packages use implementations of the provider interfaces to offer support for locales beyond the set of locales supported by the Java runtime environment itself.

#### Packaging of Locale Sensitive Service Provider Implementations

Implementations of these locale sensitive services are packaged using the [Java Extension Mechanism](http://docs.google.com/technotes/guides/extensions/index.html) as installed extensions. A provider identifies itself with a provider-configuration file in the resource directory META-INF/services, using the fully qualified provider interface class name as the file name. The file should contain a list of fully-qualified concrete provider class names, one per line. A line is terminated by any one of a line feed ('\n'), a carriage return ('\r'), or a carriage return followed immediately by a line feed. Space and tab characters surrounding each name, as well as blank lines, are ignored. The comment character is '#' ('#'); on each line all characters following the first comment character are ignored. The file must be encoded in UTF-8.

If a particular concrete provider class is named in more than one configuration file, or is named in the same configuration file more than once, then the duplicates will be ignored. The configuration file naming a particular provider need not be in the same jar file or other distribution unit as the provider itself. The provider must be accessible from the same class loader that was initially queried to locate the configuration file; this is not necessarily the class loader that loaded the file.

For example, an implementation of the [DateFormatProvider](http://docs.google.com/java/text/spi/DateFormatProvider.html) class should take the form of a jar file which contains the file:

META-INF/services/java.text.spi.DateFormatProvider

And the file java.text.spi.DateFormatProvider should have a line such as:

com.foo.DateFormatProviderImpl

which is the fully qualified class name of the class implementing DateFormatProvider.

#### Invocation of Locale Sensitive Services

Locale sensitive factory methods and methods for name retrieval in the java.text and java.util packages invoke service provider methods when needed to support the requested locale. The methods first check whether the Java runtime environment itself supports the requested locale, and use its support if available. Otherwise, they call the getAvailableLocales() methods of installed providers for the appropriate interface to find one that supports the requested locale. If such a provider is found, its other methods are called to obtain the requested object or name. If neither the Java runtime environment itself nor an installed provider supports the requested locale, a fallback locale is constructed by replacing the first of the variant, country, or language strings of the locale that's not an empty string with an empty string, and the lookup process is restarted. In the case that the variant contains one or more '\_'s, the fallback locale is constructed by replacing the variant with a new variant which eliminates the last '\_' and the part following it. Even if a fallback occurs, methods that return requested objects or name are invoked with the original locale before the fallback.The Java runtime environment must support the root locale for all locale sensitive services in order to guarantee that this process terminates.

Providers of names (but not providers of other objects) are allowed to return null for some name requests even for locales that they claim to support by including them in their return value for getAvailableLocales. Similarly, the Java runtime environment itself may not have all names for all locales that it supports. This is because the sets of objects for which names are requested can be large and vary over time, so that it's not always feasible to cover them completely. If the Java runtime environment or a provider returns null instead of a name, the lookup will proceed as described above as if the locale was not supported.

**Since:** 1.6

| **Constructor Summary** | |
| --- | --- |
| protected | [**LocaleServiceProvider**](http://docs.google.com/java/util/spi/LocaleServiceProvider.html#LocaleServiceProvider())()            Sole constructor. |

| **Method Summary** | |
| --- | --- |
| abstract  [Locale](http://docs.google.com/java/util/Locale.html)[] | [**getAvailableLocales**](http://docs.google.com/java/util/spi/LocaleServiceProvider.html#getAvailableLocales())()            Returns an array of all locales for which this locale service provider can provide localized objects or names. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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)) |

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

### LocaleServiceProvider

protected **LocaleServiceProvider**()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

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

### getAvailableLocales

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

Returns an array of all locales for which this locale service provider can provide localized objects or names.

**Returns:**An array of all locales for which this locale service provider can provide localized objects or names.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LocaleServiceProvider.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/spi/LocaleNameProvider.html)   [**NEXT CLASS**](http://docs.google.com/java/util/spi/TimeZoneNameProvider.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/spi/LocaleServiceProvider.html)    [**NO FRAMES**](http://docs.google.com/LocaleServiceProvider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).