| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResourceBundle.Control.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/ResourceBundle.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Scanner.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/ResourceBundle.Control.html)    [**NO FRAMES**](http://docs.google.com/ResourceBundle.Control.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **java.util**

Class ResourceBundle.Control

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

**Enclosing class:**[ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html)

public static class **ResourceBundle.Control**extends [Object](http://docs.google.com/java/lang/Object.html)

ResourceBundle.Control defines a set of callback methods that are invoked by the [ResourceBundle.getBundle](http://docs.google.com/java/util/ResourceBundle.html#getBundle(java.lang.String,%20java.util.Locale,%20java.lang.ClassLoader,%20java.util.ResourceBundle.Control)) factory methods during the bundle loading process. In other words, a ResourceBundle.Control collaborates with the factory methods for loading resource bundles. The default implementation of the callback methods provides the information necessary for the factory methods to perform the [default behavior](http://docs.google.com/ResourceBundle.html#default_behavior).

In addition to the callback methods, the [toBundleName](http://docs.google.com/java/util/ResourceBundle.Control.html#toBundleName(java.lang.String,%20java.util.Locale)) and [toResourceName](http://docs.google.com/java/util/ResourceBundle.Control.html#toResourceName(java.lang.String,%20java.lang.String)) methods are defined primarily for convenience in implementing the callback methods. However, the toBundleName method could be overridden to provide different conventions in the organization and packaging of localized resources. The toResourceName method is final to avoid use of wrong resource and class name separators.

Two factory methods, [getControl(List)](http://docs.google.com/java/util/ResourceBundle.Control.html#getControl(java.util.List)) and [getNoFallbackControl(List)](http://docs.google.com/java/util/ResourceBundle.Control.html#getNoFallbackControl(java.util.List)), provide ResourceBundle.Control instances that implement common variations of the default bundle loading process.

The formats returned by the [getFormats](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String)) method and candidate locales returned by the [getCandidateLocales](http://docs.google.com/java/util/ResourceBundle.Control.html#getCandidateLocales(java.lang.String,%20java.util.Locale)) method must be consistent in all ResourceBundle.getBundle invocations for the same base bundle. Otherwise, the ResourceBundle.getBundle methods may return unintended bundles. For example, if only "java.class" is returned by the getFormats method for the first call to ResourceBundle.getBundle and only "java.properties" for the second call, then the second call will return the class-based one that has been cached during the first call.

A ResourceBundle.Control instance must be thread-safe if it's simultaneously used by multiple threads. ResourceBundle.getBundle does not synchronize to call the ResourceBundle.Control methods. The default implementations of the methods are thread-safe.

Applications can specify ResourceBundle.Control instances returned by the getControl factory methods or created from a subclass of ResourceBundle.Control to customize the bundle loading process. The following are examples of changing the default bundle loading process.

**Example 1**

The following code lets ResourceBundle.getBundle look up only properties-based resources.

import java.util.\*;  
 import static java.util.ResourceBundle.Control.\*;  
 ...  
 ResourceBundle bundle =  
 ResourceBundle.getBundle("MyResources", new Locale("fr", "CH"),  
 ResourceBundle.Control.getControl(FORMAT\_PROPERTIES));

Given the resource bundles in the [example](http://docs.google.com/ResourceBundle.html#default_behavior_example) in the ResourceBundle.getBundle description, this ResourceBundle.getBundle call loads MyResources\_fr\_CH.properties whose parent is MyResources\_fr.properties whose parent is MyResources.properties. (MyResources\_fr\_CH.properties is not hidden, but MyResources\_fr\_CH.class is.)

**Example 2**

The following is an example of loading XML-based bundles using [Properties.loadFromXML](http://docs.google.com/java/util/Properties.html#loadFromXML(java.io.InputStream)).

ResourceBundle rb = ResourceBundle.getBundle("Messages",  
 new ResourceBundle.Control() {  
 public List<String> getFormats(String baseName) {  
 if (baseName == null)  
 throw new NullPointerException();  
 return Arrays.asList("xml");  
 }  
 public ResourceBundle newBundle(String baseName,  
 Locale locale,  
 String format,  
 ClassLoader loader,  
 boolean reload)  
 throws IllegalAccessException,  
 InstantiationException,  
 IOException {  
 if (baseName == null || locale == null  
 || format == null || loader == null)  
 throw new NullPointerException();  
 ResourceBundle bundle = null;  
 if (format.equals("xml")) {  
 String bundleName = toBundleName(baseName, locale);  
 String resourceName = toResourceName(bundleName, format);  
 InputStream stream = null;  
 if (reload) {  
 URL url = loader.getResource(resourceName);  
 if (url != null) {  
 URLConnection connection = url.openConnection();  
 if (connection != null) {  
 // Disable caches to get fresh data for  
 // reloading.  
 connection.setUseCaches(false);  
 stream = connection.getInputStream();  
 }  
 }  
 } else {  
 stream = loader.getResourceAsStream(resourceName);  
 }  
 if (stream != null) {  
 BufferedInputStream bis = new BufferedInputStream(stream);  
 bundle = new XMLResourceBundle(bis);  
 bis.close();  
 }  
 }  
 return bundle;  
 }  
 });  
  
 ...  
  
 private static class XMLResourceBundle extends ResourceBundle {  
 private Properties props;  
 XMLResourceBundle(InputStream stream) throws IOException {  
 props = new Properties();  
 props.loadFromXML(stream);  
 }  
 protected Object handleGetObject(String key) {  
 return props.getProperty(key);  
 }  
 public Enumeration<String> getKeys() {  
 ...  
 }  
 }

**Since:** 1.6

| **Field Summary** | |
| --- | --- |
| static [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> | [**FORMAT\_CLASS**](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_CLASS)            The class-only format List containing "java.class". |
| static [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> | [**FORMAT\_DEFAULT**](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT)            The default format List, which contains the strings "java.class" and "java.properties", in this order. |
| static [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> | [**FORMAT\_PROPERTIES**](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_PROPERTIES)            The properties-only format List containing "java.properties". |
| static long | [**TTL\_DONT\_CACHE**](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_DONT_CACHE)            The time-to-live constant for not caching loaded resource bundle instances. |
| static long | [**TTL\_NO\_EXPIRATION\_CONTROL**](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_NO_EXPIRATION_CONTROL)            The time-to-live constant for disabling the expiration control for loaded resource bundle instances in the cache. |

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

| **Method Summary** | |
| --- | --- |
| [List](http://docs.google.com/java/util/List.html)<[Locale](http://docs.google.com/java/util/Locale.html)> | [**getCandidateLocales**](http://docs.google.com/java/util/ResourceBundle.Control.html#getCandidateLocales(java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a List of Locales as candidate locales for baseName and locale. |
| static [ResourceBundle.Control](http://docs.google.com/java/util/ResourceBundle.Control.html) | [**getControl**](http://docs.google.com/java/util/ResourceBundle.Control.html#getControl(java.util.List))([List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> formats)            Returns a ResourceBundle.Control in which the [getFormats](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String)) method returns the specified formats. |
| [Locale](http://docs.google.com/java/util/Locale.html) | [**getFallbackLocale**](http://docs.google.com/java/util/ResourceBundle.Control.html#getFallbackLocale(java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a Locale to be used as a fallback locale for further resource bundle searches by the ResourceBundle.getBundle factory method. |
| [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getFormats**](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String))([String](http://docs.google.com/java/lang/String.html) baseName)            Returns a List of Strings containing formats to be used to load resource bundles for the given baseName. |
| static [ResourceBundle.Control](http://docs.google.com/java/util/ResourceBundle.Control.html) | [**getNoFallbackControl**](http://docs.google.com/java/util/ResourceBundle.Control.html#getNoFallbackControl(java.util.List))([List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> formats)            Returns a ResourceBundle.Control in which the [getFormats](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String)) method returns the specified formats and the [getFallbackLocale](http://docs.google.com/java/util/ResourceBundle.Control.html#getFallbackLocale(java.lang.String,%20java.util.Locale)) method returns null. |
| long | [**getTimeToLive**](http://docs.google.com/java/util/ResourceBundle.Control.html#getTimeToLive(java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns the time-to-live (TTL) value for resource bundles that are loaded under this ResourceBundle.Control. |
| boolean | [**needsReload**](http://docs.google.com/java/util/ResourceBundle.Control.html#needsReload(java.lang.String,%20java.util.Locale,%20java.lang.String,%20java.lang.ClassLoader,%20java.util.ResourceBundle,%20long))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale, [String](http://docs.google.com/java/lang/String.html) format, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader, [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html) bundle, long loadTime)            Determines if the expired bundle in the cache needs to be reloaded based on the loading time given by loadTime or some other criteria. |
| [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html) | [**newBundle**](http://docs.google.com/java/util/ResourceBundle.Control.html#newBundle(java.lang.String,%20java.util.Locale,%20java.lang.String,%20java.lang.ClassLoader,%20boolean))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale, [String](http://docs.google.com/java/lang/String.html) format, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader, boolean reload)            Instantiates a resource bundle for the given bundle name of the given format and locale, using the given class loader if necessary. |
| [String](http://docs.google.com/java/lang/String.html) | [**toBundleName**](http://docs.google.com/java/util/ResourceBundle.Control.html#toBundleName(java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) baseName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Converts the given baseName and locale to the bundle name. |
| [String](http://docs.google.com/java/lang/String.html) | [**toResourceName**](http://docs.google.com/java/util/ResourceBundle.Control.html#toResourceName(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) bundleName, [String](http://docs.google.com/java/lang/String.html) suffix)            Converts the given bundleName to the form required by the [ClassLoader.getResource](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)) method by replacing all occurrences of '.' in bundleName with '/' and appending a '.' and the given file suffix. |

| **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)) |

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

### FORMAT\_DEFAULT

public static final [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> **FORMAT\_DEFAULT**

The default format List, which contains the strings "java.class" and "java.properties", in this order. This List is [unmodifiable](http://docs.google.com/java/util/Collections.html#unmodifiableList(java.util.List)).

**See Also:**[getFormats(String)](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String))

### FORMAT\_CLASS

public static final [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> **FORMAT\_CLASS**

The class-only format List containing "java.class". This List is [unmodifiable](http://docs.google.com/java/util/Collections.html#unmodifiableList(java.util.List)).

**See Also:**[getFormats(String)](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String))

### FORMAT\_PROPERTIES

public static final [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> **FORMAT\_PROPERTIES**

The properties-only format List containing "java.properties". This List is [unmodifiable](http://docs.google.com/java/util/Collections.html#unmodifiableList(java.util.List)).

**See Also:**[getFormats(String)](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String))

### TTL\_DONT\_CACHE

public static final long **TTL\_DONT\_CACHE**

The time-to-live constant for not caching loaded resource bundle instances.

**See Also:**[getTimeToLive(String, Locale)](http://docs.google.com/java/util/ResourceBundle.Control.html#getTimeToLive(java.lang.String,%20java.util.Locale)), [Constant Field Values](http://docs.google.com/constant-values.html#java.util.ResourceBundle.Control.TTL_DONT_CACHE)

### TTL\_NO\_EXPIRATION\_CONTROL

public static final long **TTL\_NO\_EXPIRATION\_CONTROL**

The time-to-live constant for disabling the expiration control for loaded resource bundle instances in the cache.

**See Also:**[getTimeToLive(String, Locale)](http://docs.google.com/java/util/ResourceBundle.Control.html#getTimeToLive(java.lang.String,%20java.util.Locale)), [Constant Field Values](http://docs.google.com/constant-values.html#java.util.ResourceBundle.Control.TTL_NO_EXPIRATION_CONTROL)

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

### ResourceBundle.Control

protected **ResourceBundle.Control**()

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

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

### getControl

public static final [ResourceBundle.Control](http://docs.google.com/java/util/ResourceBundle.Control.html) **getControl**([List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> formats)

Returns a ResourceBundle.Control in which the [getFormats](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String)) method returns the specified formats. The formats must be equal to one of [FORMAT\_PROPERTIES](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_PROPERTIES), [FORMAT\_CLASS](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_CLASS) or [FORMAT\_DEFAULT](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT). ResourceBundle.Control instances returned by this method are singletons and thread-safe.

Specifying [FORMAT\_DEFAULT](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT) is equivalent to instantiating the ResourceBundle.Control class, except that this method returns a singleton.

**Parameters:**formats - the formats to be returned by the ResourceBundle.Control.getFormats method **Returns:**a ResourceBundle.Control supporting the specified formats **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if formats is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formats is unknown

### getNoFallbackControl

public static final [ResourceBundle.Control](http://docs.google.com/java/util/ResourceBundle.Control.html) **getNoFallbackControl**([List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> formats)

Returns a ResourceBundle.Control in which the [getFormats](http://docs.google.com/java/util/ResourceBundle.Control.html#getFormats(java.lang.String)) method returns the specified formats and the [getFallbackLocale](http://docs.google.com/java/util/ResourceBundle.Control.html#getFallbackLocale(java.lang.String,%20java.util.Locale)) method returns null. The formats must be equal to one of [FORMAT\_PROPERTIES](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_PROPERTIES), [FORMAT\_CLASS](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_CLASS) or [FORMAT\_DEFAULT](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT). ResourceBundle.Control instances returned by this method are singletons and thread-safe.

**Parameters:**formats - the formats to be returned by the ResourceBundle.Control.getFormats method **Returns:**a ResourceBundle.Control supporting the specified formats with no fallback Locale support **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if formats is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formats is unknown

### getFormats

public [List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)> **getFormats**([String](http://docs.google.com/java/lang/String.html) baseName)

Returns a List of Strings containing formats to be used to load resource bundles for the given baseName. The ResourceBundle.getBundle factory method tries to load resource bundles with formats in the order specified by the list. The list returned by this method must have at least one String. The predefined formats are "java.class" for class-based resource bundles and "java.properties" for [properties-based](http://docs.google.com/java/util/PropertyResourceBundle.html) ones. Strings starting with "java." are reserved for future extensions and must not be used by application-defined formats.

It is not a requirement to return an immutable (unmodifiable) List. However, the returned List must not be mutated after it has been returned by getFormats.

The default implementation returns [FORMAT\_DEFAULT](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT) so that the ResourceBundle.getBundle factory method looks up first class-based resource bundles, then properties-based ones.

**Parameters:**baseName - the base name of the resource bundle, a fully qualified class name **Returns:**a List of Strings containing formats for loading resource bundles. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName is null**See Also:**[FORMAT\_DEFAULT](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_DEFAULT), [FORMAT\_CLASS](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_CLASS), [FORMAT\_PROPERTIES](http://docs.google.com/java/util/ResourceBundle.Control.html#FORMAT_PROPERTIES)

### getCandidateLocales

public [List](http://docs.google.com/java/util/List.html)<[Locale](http://docs.google.com/java/util/Locale.html)> **getCandidateLocales**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a List of Locales as candidate locales for baseName and locale. This method is called by the ResourceBundle.getBundle factory method each time the factory method tries finding a resource bundle for a target Locale.

The sequence of the candidate locales also corresponds to the runtime resource lookup path (also known as the *parent chain*), if the corresponding resource bundles for the candidate locales exist and their parents are not defined by loaded resource bundles themselves. The last element of the list must be a [root locale](http://docs.google.com/java/util/Locale.html#ROOT) if it is desired to have the base bundle as the terminal of the parent chain.

If the given locale is equal to Locale.ROOT (the root locale), a List containing only the root Locale must be returned. In this case, the ResourceBundle.getBundle factory method loads only the base bundle as the resulting resource bundle.

It is not a requirement to return an immutable (unmodifiable) List. However, the returned List must not be mutated after it has been returned by getCandidateLocales.

The default implementation returns a List containing Locales in the following sequence:

Locale(language, country, variant)  
 Locale(language, country)  
 Locale(language)  
 Locale.ROOT

where language, country and variant are the language, country and variant values of the given locale, respectively. Locales where the final component values are empty strings are omitted.

The default implementation uses an [ArrayList](http://docs.google.com/java/util/ArrayList.html) that overriding implementations may modify before returning it to the caller. However, a subclass must not modify it after it has been returned by getCandidateLocales.

For example, if the given baseName is "Messages" and the given locale is Locale("ja", "", "XX"), then a List of Locales:

Locale("ja", "", "XX")  
 Locale("ja")  
 Locale.ROOT

is returned. And if the resource bundles for the "ja" and "" Locales are found, then the runtime resource lookup path (parent chain) is:

Messages\_ja -> Messages

**Parameters:**baseName - the base name of the resource bundle, a fully qualified class namelocale - the locale for which a resource bundle is desired **Returns:**a List of candidate Locales for the given locale **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName or locale is null

### getFallbackLocale

public [Locale](http://docs.google.com/java/util/Locale.html) **getFallbackLocale**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a Locale to be used as a fallback locale for further resource bundle searches by the ResourceBundle.getBundle factory method. This method is called from the factory method every time when no resulting resource bundle has been found for baseName and locale, where locale is either the parameter for ResourceBundle.getBundle or the previous fallback locale returned by this method.

The method returns null if no further fallback search is desired.

The default implementation returns the [default Locale](http://docs.google.com/java/util/Locale.html#getDefault()) if the given locale isn't the default one. Otherwise, null is returned.

**Parameters:**baseName - the base name of the resource bundle, a fully qualified class name for which ResourceBundle.getBundle has been unable to find any resource bundles (except for the base bundle)locale - the Locale for which ResourceBundle.getBundle has been unable to find any resource bundles (except for the base bundle) **Returns:**a Locale for the fallback search, or null if no further fallback search is desired. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName or locale is null

### newBundle

public [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html) **newBundle**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale,  
 [String](http://docs.google.com/java/lang/String.html) format,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader,  
 boolean reload)  
 throws [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html),  
 [InstantiationException](http://docs.google.com/java/lang/InstantiationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Instantiates a resource bundle for the given bundle name of the given format and locale, using the given class loader if necessary. This method returns null if there is no resource bundle available for the given parameters. If a resource bundle can't be instantiated due to an unexpected error, the error must be reported by throwing an Error or Exception rather than simply returning null.

If the reload flag is true, it indicates that this method is being called because the previously loaded resource bundle has expired.

The default implementation instantiates a ResourceBundle as follows.

* The bundle name is obtained by calling [toBundleName(baseName, locale)](http://docs.google.com/java/util/ResourceBundle.Control.html#toBundleName(java.lang.String,%20java.util.Locale)).
* If format is "java.class", the [Class](http://docs.google.com/java/lang/Class.html) specified by the bundle name is loaded by calling [ClassLoader.loadClass(String)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)). Then, a ResourceBundle is instantiated by calling [Class.newInstance()](http://docs.google.com/java/lang/Class.html#newInstance()). Note that the reload flag is ignored for loading class-based resource bundles in this default implementation.
* If format is "java.properties", [toResourceName(bundlename, "properties")](http://docs.google.com/java/util/ResourceBundle.Control.html#toResourceName(java.lang.String,%20java.lang.String)) is called to get the resource name. If reload is true, [load.getResource](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)) is called to get a [URL](http://docs.google.com/java/net/URL.html) for creating a [URLConnection](http://docs.google.com/java/net/URLConnection.html). This URLConnection is used to [disable the caches](http://docs.google.com/java/net/URLConnection.html#setUseCaches(boolean)) of the underlying resource loading layers, and to [get an InputStream](http://docs.google.com/java/net/URLConnection.html#getInputStream()). Otherwise, [loader.getResourceAsStream](http://docs.google.com/java/lang/ClassLoader.html#getResourceAsStream(java.lang.String)) is called to get an [InputStream](http://docs.google.com/java/io/InputStream.html). Then, a [PropertyResourceBundle](http://docs.google.com/java/util/PropertyResourceBundle.html) is constructed with the InputStream.
* If format is neither "java.class" nor "java.properties", an IllegalArgumentException is thrown.

**Parameters:**baseName - the base bundle name of the resource bundle, a fully qualified class namelocale - the locale for which the resource bundle should be instantiatedformat - the resource bundle format to be loadedloader - the ClassLoader to use to load the bundlereload - the flag to indicate bundle reloading; true if reloading an expired resource bundle, false otherwise **Returns:**the resource bundle instance, or null if none could be found. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if bundleName, locale, format, or loader is null, or if null is returned by [toBundleName](http://docs.google.com/java/util/ResourceBundle.Control.html#toBundleName(java.lang.String,%20java.util.Locale)) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if format is unknown, or if the resource found for the given parameters contains malformed data. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the loaded class cannot be cast to ResourceBundle [IllegalAccessException](http://docs.google.com/java/lang/IllegalAccessException.html) - if the class or its nullary constructor is not accessible. [InstantiationException](http://docs.google.com/java/lang/InstantiationException.html) - if the instantiation of a class fails for some other reason. [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html) - if the initialization provoked by this method fails. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - If a security manager is present and creation of new instances is denied. See [Class.newInstance()](http://docs.google.com/java/lang/Class.html#newInstance()) for details. [IOException](http://docs.google.com/java/io/IOException.html) - if an error occurred when reading resources using any I/O operations

### getTimeToLive

public long **getTimeToLive**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns the time-to-live (TTL) value for resource bundles that are loaded under this ResourceBundle.Control. Positive time-to-live values specify the number of milliseconds a bundle can remain in the cache without being validated against the source data from which it was constructed. The value 0 indicates that a bundle must be validated each time it is retrieved from the cache. [TTL\_DONT\_CACHE](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_DONT_CACHE) specifies that loaded resource bundles are not put in the cache. [TTL\_NO\_EXPIRATION\_CONTROL](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_NO_EXPIRATION_CONTROL) specifies that loaded resource bundles are put in the cache with no expiration control.

The expiration affects only the bundle loading process by the ResourceBundle.getBundle factory method. That is, if the factory method finds a resource bundle in the cache that has expired, the factory method calls the [needsReload](http://docs.google.com/java/util/ResourceBundle.Control.html#needsReload(java.lang.String,%20java.util.Locale,%20java.lang.String,%20java.lang.ClassLoader,%20java.util.ResourceBundle,%20long)) method to determine whether the resource bundle needs to be reloaded. If needsReload returns true, the cached resource bundle instance is removed from the cache. Otherwise, the instance stays in the cache, updated with the new TTL value returned by this method.

All cached resource bundles are subject to removal from the cache due to memory constraints of the runtime environment. Returning a large positive value doesn't mean to lock loaded resource bundles in the cache.

The default implementation returns [TTL\_NO\_EXPIRATION\_CONTROL](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_NO_EXPIRATION_CONTROL).

**Parameters:**baseName - the base name of the resource bundle for which the expiration value is specified.locale - the locale of the resource bundle for which the expiration value is specified. **Returns:**the time (0 or a positive millisecond offset from the cached time) to get loaded bundles expired in the cache, [TTL\_NO\_EXPIRATION\_CONTROL](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_NO_EXPIRATION_CONTROL) to disable the expiration control, or [TTL\_DONT\_CACHE](http://docs.google.com/java/util/ResourceBundle.Control.html#TTL_DONT_CACHE) to disable caching. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName or locale is null

### needsReload

public boolean **needsReload**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale,  
 [String](http://docs.google.com/java/lang/String.html) format,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader,  
 [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html) bundle,  
 long loadTime)

Determines if the expired bundle in the cache needs to be reloaded based on the loading time given by loadTime or some other criteria. The method returns true if reloading is required; false otherwise. loadTime is a millisecond offset since the  [Calendar Epoch](http://docs.google.com/Calendar.html#Epoch). The calling ResourceBundle.getBundle factory method calls this method on the ResourceBundle.Control instance used for its current invocation, not on the instance used in the invocation that originally loaded the resource bundle.

The default implementation compares loadTime and the last modified time of the source data of the resource bundle. If it's determined that the source data has been modified since loadTime, true is returned. Otherwise, false is returned. This implementation assumes that the given format is the same string as its file suffix if it's not one of the default formats, "java.class" or "java.properties".

**Parameters:**baseName - the base bundle name of the resource bundle, a fully qualified class namelocale - the locale for which the resource bundle should be instantiatedformat - the resource bundle format to be loadedloader - the ClassLoader to use to load the bundlebundle - the resource bundle instance that has been expired in the cacheloadTime - the time when bundle was loaded and put in the cache **Returns:**true if the expired bundle needs to be reloaded; false otherwise. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName, locale, format, loader, or bundle is null

### toBundleName

public [String](http://docs.google.com/java/lang/String.html) **toBundleName**([String](http://docs.google.com/java/lang/String.html) baseName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Converts the given baseName and locale to the bundle name. This method is called from the default implementation of the [newBundle](http://docs.google.com/java/util/ResourceBundle.Control.html#newBundle(java.lang.String,%20java.util.Locale,%20java.lang.String,%20java.lang.ClassLoader,%20boolean)) and [needsReload](http://docs.google.com/java/util/ResourceBundle.Control.html#needsReload(java.lang.String,%20java.util.Locale,%20java.lang.String,%20java.lang.ClassLoader,%20java.util.ResourceBundle,%20long)) methods.

This implementation returns the following value:

baseName + "\_" + language + "\_" + country + "\_" + variant

where language, country and variant are the language, country and variant values of locale, respectively. Final component values that are empty Strings are omitted along with the preceding '\_'. If all of the values are empty strings, then baseName is returned.

For example, if baseName is "baseName" and locale is Locale("ja", "", "XX"), then "baseName\_ja\_ \_XX" is returned. If the given locale is Locale("en"), then "baseName\_en" is returned.

Overriding this method allows applications to use different conventions in the organization and packaging of localized resources.

**Parameters:**baseName - the base name of the resource bundle, a fully qualified class namelocale - the locale for which a resource bundle should be loaded **Returns:**the bundle name for the resource bundle **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if baseName or locale is null

### toResourceName

public final [String](http://docs.google.com/java/lang/String.html) **toResourceName**([String](http://docs.google.com/java/lang/String.html) bundleName,  
 [String](http://docs.google.com/java/lang/String.html) suffix)

Converts the given bundleName to the form required by the [ClassLoader.getResource](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)) method by replacing all occurrences of '.' in bundleName with '/' and appending a '.' and the given file suffix. For example, if bundleName is "foo.bar.MyResources\_ja\_JP" and suffix is "properties", then "foo/bar/MyResources\_ja\_JP.properties" is returned.

**Parameters:**bundleName - the bundle namesuffix - the file type suffix **Returns:**the converted resource name **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if bundleName or suffix is null

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResourceBundle.Control.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/ResourceBundle.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Scanner.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/ResourceBundle.Control.html)    [**NO FRAMES**](http://docs.google.com/ResourceBundle.Control.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

[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).