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

## **java.beans**

Class Beans

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

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

This class provides some general purpose beans control methods.

| **Constructor Summary** | |
| --- | --- |
| [**Beans**](http://docs.google.com/java/beans/Beans.html#Beans())() |

| **Method Summary** | |
| --- | --- |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**getInstanceOf**](http://docs.google.com/java/beans/Beans.html#getInstanceOf(java.lang.Object,%20java.lang.Class))([Object](http://docs.google.com/java/lang/Object.html) bean, [Class](http://docs.google.com/java/lang/Class.html)<?> targetType)            From a given bean, obtain an object representing a specified type view of that source object. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/java/beans/Beans.html#instantiate(java.lang.ClassLoader,%20java.lang.String))([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls, [String](http://docs.google.com/java/lang/String.html) beanName)             Instantiate a JavaBean. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/java/beans/Beans.html#instantiate(java.lang.ClassLoader,%20java.lang.String,%20java.beans.beancontext.BeanContext))([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls, [String](http://docs.google.com/java/lang/String.html) beanName, [BeanContext](http://docs.google.com/java/beans/beancontext/BeanContext.html) beanContext)             Instantiate a JavaBean. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/java/beans/Beans.html#instantiate(java.lang.ClassLoader,%20java.lang.String,%20java.beans.beancontext.BeanContext,%20java.beans.AppletInitializer))([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls, [String](http://docs.google.com/java/lang/String.html) beanName, [BeanContext](http://docs.google.com/java/beans/beancontext/BeanContext.html) beanContext, [AppletInitializer](http://docs.google.com/java/beans/AppletInitializer.html) initializer)            Instantiate a bean. |
| static boolean | [**isDesignTime**](http://docs.google.com/java/beans/Beans.html#isDesignTime())()            Test if we are in design-mode. |
| static boolean | [**isGuiAvailable**](http://docs.google.com/java/beans/Beans.html#isGuiAvailable())()            Determines whether beans can assume a GUI is available. |
| static boolean | [**isInstanceOf**](http://docs.google.com/java/beans/Beans.html#isInstanceOf(java.lang.Object,%20java.lang.Class))([Object](http://docs.google.com/java/lang/Object.html) bean, [Class](http://docs.google.com/java/lang/Class.html)<?> targetType)            Check if a bean can be viewed as a given target type. |
| static void | [**setDesignTime**](http://docs.google.com/java/beans/Beans.html#setDesignTime(boolean))(boolean isDesignTime)            Used to indicate whether of not we are running in an application builder environment. |
| static void | [**setGuiAvailable**](http://docs.google.com/java/beans/Beans.html#setGuiAvailable(boolean))(boolean isGuiAvailable)            Used to indicate whether of not we are running in an environment where GUI interaction is available. |

| **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** |
| --- |

### Beans

public **Beans**()

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

### instantiate

public static [Object](http://docs.google.com/java/lang/Object.html) **instantiate**([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls,  
 [String](http://docs.google.com/java/lang/String.html) beanName)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Instantiate a JavaBean.

**Parameters:**cls - the class-loader from which we should create the bean. If this is null, then the system class-loader is used.beanName - the name of the bean within the class-loader. For example "sun.beanbox.foobah" **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - if the class of a serialized object could not be found. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### instantiate

public static [Object](http://docs.google.com/java/lang/Object.html) **instantiate**([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls,  
 [String](http://docs.google.com/java/lang/String.html) beanName,  
 [BeanContext](http://docs.google.com/java/beans/beancontext/BeanContext.html) beanContext)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Instantiate a JavaBean.

**Parameters:**cls - the class-loader from which we should create the bean. If this is null, then the system class-loader is used.beanName - the name of the bean within the class-loader. For example "sun.beanbox.foobah"beanContext - The BeanContext in which to nest the new bean **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - if the class of a serialized object could not be found. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### instantiate

public static [Object](http://docs.google.com/java/lang/Object.html) **instantiate**([ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) cls,  
 [String](http://docs.google.com/java/lang/String.html) beanName,  
 [BeanContext](http://docs.google.com/java/beans/beancontext/BeanContext.html) beanContext,  
 [AppletInitializer](http://docs.google.com/java/beans/AppletInitializer.html) initializer)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Instantiate a bean.

The bean is created based on a name relative to a class-loader. This name should be a dot-separated name such as "a.b.c".

In Beans 1.0 the given name can indicate either a serialized object or a class. Other mechanisms may be added in the future. In beans 1.0 we first try to treat the beanName as a serialized object name then as a class name.

When using the beanName as a serialized object name we convert the given beanName to a resource pathname and add a trailing ".ser" suffix. We then try to load a serialized object from that resource.

For example, given a beanName of "x.y", Beans.instantiate would first try to read a serialized object from the resource "x/y.ser" and if that failed it would try to load the class "x.y" and create an instance of that class.

If the bean is a subtype of java.applet.Applet, then it is given some special initialization. First, it is supplied with a default AppletStub and AppletContext. Second, if it was instantiated from a classname the applet's "init" method is called. (If the bean was deserialized this step is skipped.)

Note that for beans which are applets, it is the caller's responsiblity to call "start" on the applet. For correct behaviour, this should be done after the applet has been added into a visible AWT container.

Note that applets created via beans.instantiate run in a slightly different environment than applets running inside browsers. In particular, bean applets have no access to "parameters", so they may wish to provide property get/set methods to set parameter values. We advise bean-applet developers to test their bean-applets against both the JDK appletviewer (for a reference browser environment) and the BDK BeanBox (for a reference bean container).

**Parameters:**cls - the class-loader from which we should create the bean. If this is null, then the system class-loader is used.beanName - the name of the bean within the class-loader. For example "sun.beanbox.foobah"beanContext - The BeanContext in which to nest the new beaninitializer - The AppletInitializer for the new bean **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - if the class of a serialized object could not be found. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### getInstanceOf

public static [Object](http://docs.google.com/java/lang/Object.html) **getInstanceOf**([Object](http://docs.google.com/java/lang/Object.html) bean,  
 [Class](http://docs.google.com/java/lang/Class.html)<?> targetType)

From a given bean, obtain an object representing a specified type view of that source object.

The result may be the same object or a different object. If the requested target view isn't available then the given bean is returned.

This method is provided in Beans 1.0 as a hook to allow the addition of more flexible bean behaviour in the future.

**Parameters:**bean - Object from which we want to obtain a view.targetType - The type of view we'd like to get.

### isInstanceOf

public static boolean **isInstanceOf**([Object](http://docs.google.com/java/lang/Object.html) bean,  
 [Class](http://docs.google.com/java/lang/Class.html)<?> targetType)

Check if a bean can be viewed as a given target type. The result will be true if the Beans.getInstanceof method can be used on the given bean to obtain an object that represents the specified targetType type view.

**Parameters:**bean - Bean from which we want to obtain a view.targetType - The type of view we'd like to get. **Returns:**"true" if the given bean supports the given targetType.

### isDesignTime

public static boolean **isDesignTime**()

Test if we are in design-mode.

**Returns:**True if we are running in an application construction environment.**See Also:**[DesignMode](http://docs.google.com/java/beans/DesignMode.html)

### isGuiAvailable

public static boolean **isGuiAvailable**()

Determines whether beans can assume a GUI is available.

**Returns:**True if we are running in an environment where beans can assume that an interactive GUI is available, so they can pop up dialog boxes, etc. This will normally return true in a windowing environment, and will normally return false in a server environment or if an application is running as part of a batch job.**See Also:**[Visibility](http://docs.google.com/java/beans/Visibility.html)

### setDesignTime

public static void **setDesignTime**(boolean isDesignTime)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Used to indicate whether of not we are running in an application builder environment.

Note that this method is security checked and is not available to (for example) untrusted applets. More specifically, if there is a security manager, its checkPropertiesAccess method is called. This could result in a SecurityException.

**Parameters:**isDesignTime - True if we're in an application builder tool. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPropertiesAccess method doesn't allow setting of system properties.**See Also:**[SecurityManager.checkPropertiesAccess()](http://docs.google.com/java/lang/SecurityManager.html#checkPropertiesAccess())

### setGuiAvailable

public static void **setGuiAvailable**(boolean isGuiAvailable)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Used to indicate whether of not we are running in an environment where GUI interaction is available.

Note that this method is security checked and is not available to (for example) untrusted applets. More specifically, if there is a security manager, its checkPropertiesAccess method is called. This could result in a SecurityException.

**Parameters:**isGuiAvailable - True if GUI interaction is available. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPropertiesAccess method doesn't allow setting of system properties.**See Also:**[SecurityManager.checkPropertiesAccess()](http://docs.google.com/java/lang/SecurityManager.html#checkPropertiesAccess())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Beans.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/beans/BeanInfo.html)   [**NEXT CLASS**](http://docs.google.com/java/beans/ConstructorProperties.html) | [**FRAMES**](http://docs.google.com/index.html?java/beans/Beans.html)    [**NO FRAMES**](http://docs.google.com/Beans.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).