| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ObjectFactory.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/javax/naming/spi/NamingManager.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/ObjectFactory.html)    [**NO FRAMES**](http://docs.google.com/ObjectFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.naming.spi**

Interface ObjectFactory

**All Known Subinterfaces:** [DirObjectFactory](http://docs.google.com/javax/naming/spi/DirObjectFactory.html)

public interface **ObjectFactory**

This interface represents a factory for creating an object.

The JNDI framework allows for object implementations to be loaded in dynamically via *object factories*. For example, when looking up a printer bound in the name space, if the print service binds printer names to References, the printer Reference could be used to create a printer object, so that the caller of lookup can directly operate on the printer object after the lookup.

An ObjectFactory is responsible for creating objects of a specific type. In the above example, you may have a PrinterObjectFactory for creating Printer objects.

An object factory must implement the ObjectFactory interface. In addition, the factory class must be public and must have a public constructor that accepts no parameters.

The getObjectInstance() method of an object factory may be invoked multiple times, possibly using different parameters. The implementation is thread-safe.

The mention of URL in the documentation for this class refers to a URL string as defined by RFC 1738 and its related RFCs. It is any string that conforms to the syntax described therein, and may not always have corresponding support in the java.net.URL class or Web browsers.

**Since:** 1.3 **See Also:**[NamingManager.getObjectInstance(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable)), [NamingManager.getURLContext(java.lang.String, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.html#getURLContext(java.lang.String,%20java.util.Hashtable)), [ObjectFactoryBuilder](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html), [StateFactory](http://docs.google.com/javax/naming/spi/StateFactory.html)

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObjectInstance**](http://docs.google.com/javax/naming/spi/ObjectFactory.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable))([Object](http://docs.google.com/java/lang/Object.html) obj, [Name](http://docs.google.com/javax/naming/Name.html) name, [Context](http://docs.google.com/javax/naming/Context.html) nameCtx, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> environment)            Creates an object using the location or reference information specified. |

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

### getObjectInstance

[Object](http://docs.google.com/java/lang/Object.html) **getObjectInstance**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 [Name](http://docs.google.com/javax/naming/Name.html) name,  
 [Context](http://docs.google.com/javax/naming/Context.html) nameCtx,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> environment)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Creates an object using the location or reference information specified.

Special requirements of this object are supplied using environment. An example of such an environment property is user identity information.

NamingManager.getObjectInstance() successively loads in object factories and invokes this method on them until one produces a non-null answer. When an exception is thrown by an object factory, the exception is passed on to the caller of NamingManager.getObjectInstance() (and no search is made for other factories that may produce a non-null answer). An object factory should only throw an exception if it is sure that it is the only intended factory and that no other object factories should be tried. If this factory cannot create an object using the arguments supplied, it should return null.

A *URL context factory* is a special ObjectFactory that creates contexts for resolving URLs or objects whose locations are specified by URLs. The getObjectInstance() method of a URL context factory will obey the following rules.

1. If obj is null, create a context for resolving URLs of the scheme associated with this factory. The resulting context is not tied to a specific URL: it is able to handle arbitrary URLs with this factory's scheme id. For example, invoking getObjectInstance() with obj set to null on an LDAP URL context factory would return a context that can resolve LDAP URLs such as "ldap://ldap.wiz.com/o=wiz,c=us" and "ldap://ldap.umich.edu/o=umich,c=us".
2. If obj is a URL string, create an object (typically a context) identified by the URL. For example, suppose this is an LDAP URL context factory. If obj is "ldap://ldap.wiz.com/o=wiz,c=us", getObjectInstance() would return the context named by the distinguished name "o=wiz, c=us" at the LDAP server ldap.wiz.com. This context can then be used to resolve LDAP names (such as "cn=George") relative to that context.
3. If obj is an array of URL strings, the assumption is that the URLs are equivalent in terms of the context to which they refer. Verification of whether the URLs are, or need to be, equivalent is up to the context factory. The order of the URLs in the array is not significant. The object returned by getObjectInstance() is like that of the single URL case. It is the object named by the URLs.
4. If obj is of any other type, the behavior of getObjectInstance() is determined by the context factory implementation.

The name and environment parameters are owned by the caller. The implementation will not modify these objects or keep references to them, although it may keep references to clones or copies.

**Name and Context Parameters.**      The name and nameCtx parameters may optionally be used to specify the name of the object being created. name is the name of the object, relative to context nameCtx. If there are several possible contexts from which the object could be named -- as will often be the case -- it is up to the caller to select one. A good rule of thumb is to select the "deepest" context available. If nameCtx is null, name is relative to the default initial context. If no name is being specified, the name parameter should be null. If a factory uses nameCtx it should synchronize its use against concurrent access, since context implementations are not guaranteed to be thread-safe.

**Parameters:**obj - The possibly null object containing location or reference information that can be used in creating an object.name - The name of this object relative to nameCtx, or null if no name is specified.nameCtx - The context relative to which the name parameter is specified, or null if name is relative to the default initial context.environment - The possibly null environment that is used in creating the object. **Returns:**The object created; null if an object cannot be created. **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - if this object factory encountered an exception while attempting to create an object, and no other object factories are to be tried.**See Also:**[NamingManager.getObjectInstance(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable)), [NamingManager.getURLContext(java.lang.String, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.html#getURLContext(java.lang.String,%20java.util.Hashtable))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ObjectFactory.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/javax/naming/spi/NamingManager.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/ObjectFactory.html)    [**NO FRAMES**](http://docs.google.com/ObjectFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).