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

## **javax.naming.spi**

Class NamingManager

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.naming.spi.NamingManager**

**Direct Known Subclasses:** [DirectoryManager](http://docs.google.com/javax/naming/spi/DirectoryManager.html)

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

This class contains methods for creating context objects and objects referred to by location information in the naming or directory service.

This class cannot be instantiated. It has only static methods.

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.

NamingManager is safe for concurrent access by multiple threads.

Except as otherwise noted, a Name or environment parameter passed to any method is owned by the caller. The implementation will not modify the object or keep a reference to it, although it may keep a reference to a clone or copy.

**Since:** 1.3

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**CPE**](http://docs.google.com/javax/naming/spi/NamingManager.html#CPE)            Constant that holds the name of the environment property into which getContinuationContext() stores the value of its CannotProceedException parameter. |

| **Method Summary** | |
| --- | --- |
| static [Context](http://docs.google.com/javax/naming/Context.html) | [**getContinuationContext**](http://docs.google.com/javax/naming/spi/NamingManager.html#getContinuationContext(javax.naming.CannotProceedException))([CannotProceedException](http://docs.google.com/javax/naming/CannotProceedException.html) cpe)            Creates a context in which to continue a context operation. |
| static [Context](http://docs.google.com/javax/naming/Context.html) | [**getInitialContext**](http://docs.google.com/javax/naming/spi/NamingManager.html#getInitialContext(java.util.Hashtable))([Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> env)            Creates an initial context using the specified environment properties. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**getObjectInstance**](http://docs.google.com/javax/naming/spi/NamingManager.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable))([Object](http://docs.google.com/java/lang/Object.html) refInfo, [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 instance of an object for the specified object and environment. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**getStateToBind**](http://docs.google.com/javax/naming/spi/NamingManager.html#getStateToBind(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)            Retrieves the state of an object for binding. |
| static [Context](http://docs.google.com/javax/naming/Context.html) | [**getURLContext**](http://docs.google.com/javax/naming/spi/NamingManager.html#getURLContext(java.lang.String,%20java.util.Hashtable))([String](http://docs.google.com/java/lang/String.html) scheme, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> environment)            Creates a context for the given URL scheme id. |
| static boolean | [**hasInitialContextFactoryBuilder**](http://docs.google.com/javax/naming/spi/NamingManager.html#hasInitialContextFactoryBuilder())()            Determines whether an initial context factory builder has been set. |
| static void | [**setInitialContextFactoryBuilder**](http://docs.google.com/javax/naming/spi/NamingManager.html#setInitialContextFactoryBuilder(javax.naming.spi.InitialContextFactoryBuilder))([InitialContextFactoryBuilder](http://docs.google.com/javax/naming/spi/InitialContextFactoryBuilder.html) builder)            Sets the InitialContextFactory builder to be builder. |
| static void | [**setObjectFactoryBuilder**](http://docs.google.com/javax/naming/spi/NamingManager.html#setObjectFactoryBuilder(javax.naming.spi.ObjectFactoryBuilder))([ObjectFactoryBuilder](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html) builder)            The ObjectFactoryBuilder determines the policy used when trying to load object factories. |

| **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** |
| --- |

### CPE

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

Constant that holds the name of the environment property into which getContinuationContext() stores the value of its CannotProceedException parameter. This property is inherited by the continuation context, and may be used by that context's service provider to inspect the fields of the exception.

The value of this constant is "java.naming.spi.CannotProceedException".

**Since:** 1.3 **See Also:**[getContinuationContext(javax.naming.CannotProceedException)](http://docs.google.com/javax/naming/spi/NamingManager.html#getContinuationContext(javax.naming.CannotProceedException)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.spi.NamingManager.CPE)

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

### setObjectFactoryBuilder

public static void **setObjectFactoryBuilder**([ObjectFactoryBuilder](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html) builder)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

The ObjectFactoryBuilder determines the policy used when trying to load object factories. See getObjectInstance() and class ObjectFactory for a description of the default policy. setObjectFactoryBuilder() overrides this default policy by installing an ObjectFactoryBuilder. Subsequent object factories will be loaded and created using the installed builder.

The builder can only be installed if the executing thread is allowed (by the security manager's checkSetFactory() method) to do so. Once installed, the builder cannot be replaced.

**Parameters:**builder - The factory builder to install. If null, no builder is installed. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - builder cannot be installed for security reasons. [NamingException](http://docs.google.com/javax/naming/NamingException.html) - builder cannot be installed for a non-security-related reason. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - If a factory has already been installed.**See Also:**[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)), [ObjectFactory](http://docs.google.com/javax/naming/spi/ObjectFactory.html), [ObjectFactoryBuilder](http://docs.google.com/javax/naming/spi/ObjectFactoryBuilder.html), [SecurityManager.checkSetFactory()](http://docs.google.com/java/lang/SecurityManager.html#checkSetFactory())

### getObjectInstance

public static [Object](http://docs.google.com/java/lang/Object.html) **getObjectInstance**([Object](http://docs.google.com/java/lang/Object.html) refInfo,  
 [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 instance of an object for the specified object and environment.

If an object factory builder has been installed, it is used to create a factory for creating the object. Otherwise, the following rules are used to create the object:

1. If refInfo is a Reference or Referenceable containing a factory class name, use the named factory to create the object. Return refInfo if the factory cannot be created. Under JDK 1.1, if the factory class must be loaded from a location specified in the reference, a SecurityManager must have been installed or the factory creation will fail. If an exception is encountered while creating the factory, it is passed up to the caller.
2. If refInfo is a Reference or Referenceable with no factory class name, and the address or addresses are StringRefAddrs with address type "URL", try the URL context factory corresponding to each URL's scheme id to create the object (see getURLContext()). If that fails, continue to the next step.
3. Use the object factories specified in the Context.OBJECT\_FACTORIES property of the environment, and of the provider resource file associated with nameCtx, in that order. The value of this property is a colon-separated list of factory class names that are tried in order, and the first one that succeeds in creating an object is the one used. If none of the factories can be loaded, return refInfo. If an exception is encountered while creating the object, the exception is passed up to the caller.

Service providers that implement the DirContext interface should use DirectoryManager.getObjectInstance(), not this method. Service providers that implement only the Context interface should use this method.

Note that an object factory (an object that implements the ObjectFactory interface) must be public and must have a public constructor that accepts no arguments.

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. This information could be useful to the object factory or to the object implementation. 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.

**Parameters:**refInfo - The possibly null object for which to create an object.name - The name of this object relative to nameCtx. Specifying a name is optional; if it is omitted, name should be null.nameCtx - The context relative to which the name parameter is specified. If null, name is relative to the default initial context.environment - The possibly null environment to be used in the creation of the object factory and the object. **Returns:**An object created using refInfo; or refInfo if an object cannot be created using the algorithm described above. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - if a naming exception was encountered while attempting to get a URL context, or if one of the factories accessed throws a NamingException. [Exception](http://docs.google.com/java/lang/Exception.html) - if one of the factories accessed throws an exception, or if an error was encountered while loading and instantiating the factory and object classes. A factory should only throw an exception if it does not want other factories to be used in an attempt to create an object. See ObjectFactory.getObjectInstance().**See Also:**[getURLContext(java.lang.String, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.html#getURLContext(java.lang.String,%20java.util.Hashtable)), [ObjectFactory](http://docs.google.com/javax/naming/spi/ObjectFactory.html), [ObjectFactory.getObjectInstance(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/ObjectFactory.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable))

### getURLContext

public static [Context](http://docs.google.com/javax/naming/Context.html) **getURLContext**([String](http://docs.google.com/java/lang/String.html) scheme,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> environment)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Creates a context for the given URL scheme id.

The resulting context is for resolving URLs of the scheme scheme. The resulting context is not tied to a specific URL. It is able to handle arbitrary URLs with the specified scheme.

The class name of the factory that creates the resulting context has the naming convention *scheme-id*URLContextFactory (e.g. "ftpURLContextFactory" for the "ftp" scheme-id), in the package specified as follows. The Context.URL\_PKG\_PREFIXES environment property (which may contain values taken from applet parameters, system properties, or application resource files) contains a colon-separated list of package prefixes. Each package prefix in the property is tried in the order specified to load the factory class. The default package prefix is "com.sun.jndi.url" (if none of the specified packages work, this default is tried). The complete package name is constructed using the package prefix, concatenated with the scheme id.

For example, if the scheme id is "ldap", and the Context.URL\_PKG\_PREFIXES property contains "com.widget:com.wiz.jndi", the naming manager would attempt to load the following classes until one is successfully instantiated:

* com.widget.ldap.ldapURLContextFactory
* com.wiz.jndi.ldap.ldapURLContextFactory
* com.sun.jndi.url.ldap.ldapURLContextFactory

If none of the package prefixes work, null is returned.

If a factory is instantiated, it is invoked with the following parameters to produce the resulting context.

factory.getObjectInstance(null, environment);

For example, invoking getObjectInstance() as shown above on a LDAP URL context factory would return a context that can resolve LDAP urls (e.g. "ldap://ldap.wiz.com/o=wiz,c=us", "ldap://ldap.umich.edu/o=umich,c=us", ...).

Note that an object factory (an object that implements the ObjectFactory interface) must be public and must have a public constructor that accepts no arguments.

**Parameters:**scheme - The non-null scheme-id of the URLs supported by the context.environment - The possibly null environment properties to be used in the creation of the object factory and the context. **Returns:**A context for resolving URLs with the scheme id scheme; null if the factory for creating the context is not found. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception occurs while creating the context.**See Also:**[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)), [ObjectFactory.getObjectInstance(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/ObjectFactory.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable))

### getInitialContext

public static [Context](http://docs.google.com/javax/naming/Context.html) **getInitialContext**([Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> env)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Creates an initial context using the specified environment properties.

If an InitialContextFactoryBuilder has been installed, it is used to create the factory for creating the initial context. Otherwise, the class specified in the Context.INITIAL\_CONTEXT\_FACTORY environment property is used. Note that an initial context factory (an object that implements the InitialContextFactory interface) must be public and must have a public constructor that accepts no arguments.

**Parameters:**env - The possibly null environment properties used when creating the context. **Returns:**A non-null initial context. **Throws:** [NoInitialContextException](http://docs.google.com/javax/naming/NoInitialContextException.html) - If the Context.INITIAL\_CONTEXT\_FACTORY property is not found or names a nonexistent class or a class that cannot be instantiated, or if the initial context could not be created for some other reason. [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If some other naming exception was encountered.**See Also:**[InitialContext](http://docs.google.com/javax/naming/InitialContext.html), [InitialDirContext](http://docs.google.com/javax/naming/directory/InitialDirContext.html)

### setInitialContextFactoryBuilder

public static void **setInitialContextFactoryBuilder**([InitialContextFactoryBuilder](http://docs.google.com/javax/naming/spi/InitialContextFactoryBuilder.html) builder)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Sets the InitialContextFactory builder to be builder.

The builder can only be installed if the executing thread is allowed by the security manager to do so. Once installed, the builder cannot be replaced.

**Parameters:**builder - The initial context factory builder to install. If null, no builder is set. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - builder cannot be installed for security reasons. [NamingException](http://docs.google.com/javax/naming/NamingException.html) - builder cannot be installed for a non-security-related reason. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - If a builder was previous installed.**See Also:**[hasInitialContextFactoryBuilder()](http://docs.google.com/javax/naming/spi/NamingManager.html#hasInitialContextFactoryBuilder()), [SecurityManager.checkSetFactory()](http://docs.google.com/java/lang/SecurityManager.html#checkSetFactory())

### hasInitialContextFactoryBuilder

public static boolean **hasInitialContextFactoryBuilder**()

Determines whether an initial context factory builder has been set.

**Returns:**true if an initial context factory builder has been set; false otherwise.**See Also:**[setInitialContextFactoryBuilder(javax.naming.spi.InitialContextFactoryBuilder)](http://docs.google.com/javax/naming/spi/NamingManager.html#setInitialContextFactoryBuilder(javax.naming.spi.InitialContextFactoryBuilder))

### getContinuationContext

public static [Context](http://docs.google.com/javax/naming/Context.html) **getContinuationContext**([CannotProceedException](http://docs.google.com/javax/naming/CannotProceedException.html) cpe)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Creates a context in which to continue a context operation.

In performing an operation on a name that spans multiple namespaces, a context from one naming system may need to pass the operation on to the next naming system. The context implementation does this by first constructing a CannotProceedException containing information pinpointing how far it has proceeded. It then obtains a continuation context from JNDI by calling getContinuationContext. The context implementation should then resume the context operation by invoking the same operation on the continuation context, using the remainder of the name that has not yet been resolved.

Before making use of the cpe parameter, this method updates the environment associated with that object by setting the value of the property [CPE](#1t3h5sf) to cpe. This property will be inherited by the continuation context, and may be used by that context's service provider to inspect the fields of this exception.

**Parameters:**cpe - The non-null exception that triggered this continuation. **Returns:**A non-null Context object for continuing the operation. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception occurred.

### getStateToBind

public static [Object](http://docs.google.com/java/lang/Object.html) **getStateToBind**([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 [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the state of an object for binding.

Service providers that implement the DirContext interface should use DirectoryManager.getStateToBind(), not this method. Service providers that implement only the Context interface should use this method.

This method uses the specified state factories in the Context.STATE\_FACTORIES property from the environment properties, and from the provider resource file associated with nameCtx, in that order. The value of this property is a colon-separated list of factory class names that are tried in order, and the first one that succeeds in returning the object's state is the one used. If no object's state can be retrieved in this way, return the object itself. If an exception is encountered while retrieving the state, the exception is passed up to the caller.

Note that a state factory (an object that implements the StateFactory interface) must be public and must have a public constructor that accepts no arguments.

The name and nameCtx parameters may optionally be used to specify the name of the object being created. See the description of "Name and Context Parameters" in [ObjectFactory.getObjectInstance()](http://docs.google.com/javax/naming/spi/ObjectFactory.html#getObjectInstance(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable)) for details.

This method may return a Referenceable object. The service provider obtaining this object may choose to store it directly, or to extract its reference (using Referenceable.getReference()) and store that instead.

**Parameters:**obj - The non-null object for which to get state to bind.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 to be used in the creation of the state factory and the object's state. **Returns:**The non-null object representing obj's state for binding. It could be the object (obj) itself. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If one of the factories accessed throws an exception, or if an error was encountered while loading and instantiating the factory and object classes. A factory should only throw an exception if it does not want other factories to be used in an attempt to create an object. See StateFactory.getStateToBind().**Since:** 1.3 **See Also:**[StateFactory](http://docs.google.com/javax/naming/spi/StateFactory.html), [StateFactory.getStateToBind(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/StateFactory.html#getStateToBind(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable)), [DirectoryManager.getStateToBind(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable, javax.naming.directory.Attributes)](http://docs.google.com/javax/naming/spi/DirectoryManager.html#getStateToBind(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable,%20javax.naming.directory.Attributes))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NamingManager.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/InitialContextFactoryBuilder.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/spi/ObjectFactory.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/NamingManager.html)    [**NO FRAMES**](http://docs.google.com/NamingManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [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).