| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StateFactory.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/ResolveResult.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/StateFactory.html)    [**NO FRAMES**](http://docs.google.com/StateFactory.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 StateFactory

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

public interface **StateFactory**

This interface represents a factory for obtaining the state of an object for binding.

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.

For the reverse process, when an object is bound into the namespace, JNDI provides *state factories*. Continuing with the printer example, suppose the printer object is updated and rebound:

ctx.rebind("inky", printer);

The service provider for ctx uses a state factory to obtain the state of printer for binding into its namespace. A state factory for the Printer type object might return a more compact object for storage in the naming system.

A state factory must implement the StateFactory interface. In addition, the factory class must be public and must have a public constructor that accepts no parameters.

The getStateToBind() method of a state factory may be invoked multiple times, possibly using different parameters. The implementation is thread-safe.

StateFactory is intended for use with service providers that implement only the Context interface. DirStateFactory is intended for use with service providers that implement the DirContext interface.

**Since:** 1.3 **See Also:**[NamingManager.getStateToBind(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.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)), [ObjectFactory](http://docs.google.com/javax/naming/spi/ObjectFactory.html), [DirStateFactory](http://docs.google.com/javax/naming/spi/DirStateFactory.html)

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getStateToBind**](http://docs.google.com/javax/naming/spi/StateFactory.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. |

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

### getStateToBind

[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.

NamingManager.getStateToBind() successively loads in state factories and invokes this method on them until one produces a non-null answer. DirectoryManager.getStateToBind() successively loads in state factories. If a factory implements DirStateFactory, then DirectoryManager invokes DirStateFactory.getStateToBind(); otherwise it invokes StateFactory.getStateToBind().

When an exception is thrown by a factory, the exception is passed on to the caller of NamingManager.getStateToBind() and DirectoryManager.getStateToBind(). The search for other factories that may produce a non-null answer is halted. A factory should only throw an exception if it is sure that it is the only intended factory and that no other factories should be tried. If this factory cannot create an object using the arguments supplied, it should return null.

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. If a factory uses nameCtx it should synchronize its use against concurrent access, since context implementations are not guaranteed to be thread-safe.

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.

**Parameters:**obj - A non-null object whose state is to be retrieved.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 object's state. **Returns:**The object's state for binding; null if the factory is not returning any changes. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - if this factory encountered an exception while attempting to get the object's state, and no other factories are to be tried.**See Also:**[NamingManager.getStateToBind(java.lang.Object, javax.naming.Name, javax.naming.Context, java.util.Hashtable)](http://docs.google.com/javax/naming/spi/NamingManager.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/StateFactory.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/ResolveResult.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/StateFactory.html)    [**NO FRAMES**](http://docs.google.com/StateFactory.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).