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

## **javax.naming.spi**

Interface DirStateFactory

**All Superinterfaces:** [StateFactory](http://docs.google.com/javax/naming/spi/StateFactory.html)

public interface **DirStateFactory**extends [StateFactory](http://docs.google.com/javax/naming/spi/StateFactory.html)

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

The JNDI framework allows for object implementations to be loaded in dynamically via object factories.

A DirStateFactory extends StateFactory by allowing an Attributes instance to be supplied to and be returned by the getStateToBind() method. DirStateFactory implementations are intended to be used by DirContext service providers. When a caller binds an object using DirContext.bind(), he might also specify a set of attributes to be bound with the object. The object and attributes to be bound are passed to the getStateToBind() method of a factory. If the factory processes the object and attributes, it returns a corresponding pair of object and attributes to be bound. If the factory does not process the object, it must return null.

For example, a caller might bind a printer object with some printer-related attributes.

ctx.rebind("inky", printer, printerAttrs);

An LDAP service provider for ctx uses a DirStateFactory (indirectly via DirectoryManager.getStateToBind()) and gives it printer and printerAttrs. A factory for an LDAP directory might turn printer into a set of attributes and merge that with printerAttrs. The service provider then uses the resulting attributes to create an LDAP entry and updates the directory.

Since DirStateFactory extends StateFactory, it has two getStateToBind() methods, where one differs from the other by the attributes argument. DirectoryManager.getStateToBind() will only use the form that accepts the attributes argument, while NamingManager.getStateToBind() will only use the form that does not accept the attributes argument.

Either form of the getStateToBind() method of a DirStateFactory may be invoked multiple times, possibly using different parameters. The implementation is thread-safe.

**Since:** 1.3 **See Also:**[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)), [DirObjectFactory](http://docs.google.com/javax/naming/spi/DirObjectFactory.html)

| **Nested Class Summary** | |
| --- | --- |
| static class | [**DirStateFactory.Result**](http://docs.google.com/javax/naming/spi/DirStateFactory.Result.html)            An object/attributes pair for returning the result of DirStateFactory.getStateToBind(). |

| **Method Summary** | |
| --- | --- |
| [DirStateFactory.Result](http://docs.google.com/javax/naming/spi/DirStateFactory.Result.html) | [**getStateToBind**](http://docs.google.com/javax/naming/spi/DirStateFactory.html#getStateToBind(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable,%20javax.naming.directory.Attributes))([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, [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) inAttrs)            Retrieves the state of an object for binding given the object and attributes to be transformed. |

| **Methods inherited from interface javax.naming.spi.**[**StateFactory**](http://docs.google.com/javax/naming/spi/StateFactory.html) |
| --- |
| [getStateToBind](http://docs.google.com/javax/naming/spi/StateFactory.html#getStateToBind(java.lang.Object,%20javax.naming.Name,%20javax.naming.Context,%20java.util.Hashtable)) |

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

### getStateToBind

[DirStateFactory.Result](http://docs.google.com/javax/naming/spi/DirStateFactory.Result.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,  
 [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) inAttrs)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the state of an object for binding given the object and attributes to be transformed.

DirectoryManager.getStateToBind() successively loads in state factories. If a factory implements DirStateFactory, DirectoryManager invokes this method; otherwise, it invokes StateFactory.getStateToBind(). It does this until a factory produces a non-null answer.

When an exception is thrown by a factory, the exception is passed on to the caller of 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, inAttrs, 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. The object returned by this method is owned by the caller. The implementation will not subsequently modify it. It will contain either a new Attributes object that is likewise owned by the caller, or a reference to the original inAttrs parameter.

**Parameters:**obj - A possibly 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.inAttrs - The possibly null attributes to be bound with the object. The factory must not modify inAttrs. **Returns:**A Result containing the object's state for binding and the corresponding attributes to be bound; null if the object don't use this factory. **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:**[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/DirStateFactory.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/DirObjectFactory.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/spi/DirStateFactory.Result.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/spi/DirStateFactory.html)    [**NO FRAMES**](http://docs.google.com/DirStateFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

[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).