| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Activator.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/rmi/activation/ActivationSystem.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/UnknownGroupException.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/Activator.html)    [**NO FRAMES**](http://docs.google.com/Activator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.rmi.activation**

Interface Activator

**All Superinterfaces:** [Remote](http://docs.google.com/java/rmi/Remote.html)

public interface **Activator**extends [Remote](http://docs.google.com/java/rmi/Remote.html)

The Activator facilitates remote object activation. A "faulting" remote reference calls the activator's activate method to obtain a "live" reference to a "activatable" remote object. Upon receiving a request for activation, the activator looks up the activation descriptor for the activation identifier, id, determines the group in which the object should be activated initiates object re-creation via the group's ActivationInstantiator (via a call to the newInstance method). The activator initiates the execution of activation groups as necessary. For example, if an activation group for a specific group identifier is not already executing, the activator initiates the execution of a VM for the group.

The Activator works closely with ActivationSystem, which provides a means for registering groups and objects within those groups, and ActivationMonitor, which recives information about active and inactive objects and inactive groups.

The activator is responsible for monitoring and detecting when activation groups fail so that it can remove stale remote references to groups and active object's within those groups.

**Since:** 1.2 **See Also:**[ActivationInstantiator](http://docs.google.com/java/rmi/activation/ActivationInstantiator.html), [ActivationGroupDesc](http://docs.google.com/java/rmi/activation/ActivationGroupDesc.html), [ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html)

| **Method Summary** | |
| --- | --- |
| [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<? extends [Remote](http://docs.google.com/java/rmi/Remote.html)> | [**activate**](http://docs.google.com/java/rmi/activation/Activator.html#activate(java.rmi.activation.ActivationID,%20boolean))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, boolean force)            Activate the object associated with the activation identifier, id. |

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

### activate

[MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<? extends [Remote](http://docs.google.com/java/rmi/Remote.html)> **activate**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 boolean force)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Activate the object associated with the activation identifier, id. If the activator knows the object to be active already, and force is false , the stub with a "live" reference is returned immediately to the caller; otherwise, if the activator does not know that corresponding the remote object is active, the activator uses the activation descriptor information (previously registered) to determine the group (VM) in which the object should be activated. If an ActivationInstantiator corresponding to the object's group descriptor already exists, the activator invokes the activation group's newInstance method passing it the object's id and descriptor.

If the activation group for the object's group descriptor does not yet exist, the activator starts an ActivationInstantiator executing (by spawning a child process, for example). When the activator receives the activation group's call back (via the ActivationSystem's activeGroup method) specifying the activation group's reference, the activator can then invoke that activation instantiator's newInstance method to forward each pending activation request to the activation group and return the result (a marshalled remote object reference, a stub) to the caller.

Note that the activator receives a "marshalled" object instead of a Remote object so that the activator does not need to load the code for that object, or participate in distributed garbage collection for that object. If the activator kept a strong reference to the remote object, the activator would then prevent the object from being garbage collected under the normal distributed garbage collection mechanism.

**Parameters:**id - the activation identifier for the object being activatedforce - if true, the activator contacts the group to obtain the remote object's reference; if false, returning the cached value is allowed. **Returns:**the remote object (a stub) in a marshalled form **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if object activation fails [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object is unknown (not registered) [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if remote call fails**Since:** 1.2

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Activator.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/rmi/activation/ActivationSystem.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/UnknownGroupException.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/Activator.html)    [**NO FRAMES**](http://docs.google.com/Activator.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).