| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ActivationGroup.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/ActivationException.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/ActivationGroup_Stub.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/ActivationGroup.html)    [**NO FRAMES**](http://docs.google.com/ActivationGroup.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.rmi.activation**

Class ActivationGroup

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.rmi.server.RemoteObject](http://docs.google.com/java/rmi/server/RemoteObject.html)  
 [java.rmi.server.RemoteServer](http://docs.google.com/java/rmi/server/RemoteServer.html)  
 [java.rmi.server.UnicastRemoteObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html)  
 **java.rmi.activation.ActivationGroup**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [ActivationInstantiator](http://docs.google.com/java/rmi/activation/ActivationInstantiator.html), [Remote](http://docs.google.com/java/rmi/Remote.html)

public abstract class **ActivationGroup**extends [UnicastRemoteObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html)implements [ActivationInstantiator](http://docs.google.com/java/rmi/activation/ActivationInstantiator.html)

An ActivationGroup is responsible for creating new instances of "activatable" objects in its group, informing its ActivationMonitor when either: its object's become active or inactive, or the group as a whole becomes inactive.

An ActivationGroup is *initially* created in one of several ways:

* as a side-effect of creating an ActivationDesc without an explicit ActivationGroupID for the first activatable object in the group, or
* via the ActivationGroup.createGroup method
* as a side-effect of activating the first object in a group whose ActivationGroupDesc was only registered.

Only the activator can *recreate* an ActivationGroup. The activator spawns, as needed, a separate VM (as a child process, for example) for each registered activation group and directs activation requests to the appropriate group. It is implementation specific how VMs are spawned. An activation group is created via the ActivationGroup.createGroup static method. The createGroup method has two requirements on the group to be created: 1) the group must be a concrete subclass of ActivationGroup, and 2) the group must have a constructor that takes two arguments:

* the group's ActivationGroupID, and
* the group's initialization data (in a java.rmi.MarshalledObject)

When created, the default implementation of ActivationGroup will override the system properties with the properties requested when its ActivationGroupDesc was created, and will set a java.rmi.RMISecurityManager as the default system security manager. If your application requires specific properties to be set when objects are activated in the group, the application should create a special Properties object containing these properties, then create an ActivationGroupDesc with the Properties object, and use ActivationGroup.createGroup before creating any ActivationDescs (before the default ActivationGroupDesc is created). If your application requires the use of a security manager other than java.rmi.RMISecurityManager, in the ActivativationGroupDescriptor properties list you can set java.security.manager property to the name of the security manager you would like to install.

**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), [Serialized Form](http://docs.google.com/serialized-form.html#java.rmi.activation.ActivationGroup)

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.rmi.server.**[**RemoteObject**](http://docs.google.com/java/rmi/server/RemoteObject.html) |
| --- |
| [ref](http://docs.google.com/java/rmi/server/RemoteObject.html#ref) |

| **Constructor Summary** | |
| --- | --- |
| protected | [**ActivationGroup**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#ActivationGroup(java.rmi.activation.ActivationGroupID))([ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) groupID)            Constructs an activation group with the given activation group identifier. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**activeObject**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#activeObject(java.rmi.activation.ActivationID,%20java.rmi.MarshalledObject))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<? extends [Remote](http://docs.google.com/java/rmi/Remote.html)> mobj)            This protected method is necessary for subclasses to make the activeObject callback to the group's monitor. |
| abstract  void | [**activeObject**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#activeObject(java.rmi.activation.ActivationID,%20java.rmi.Remote))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, [Remote](http://docs.google.com/java/rmi/Remote.html) obj)            The group's activeObject method is called when an object is exported (either by Activatable object construction or an explicit call to Activatable.exportObject. |
| static [ActivationGroup](http://docs.google.com/java/rmi/activation/ActivationGroup.html) | [**createGroup**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#createGroup(java.rmi.activation.ActivationGroupID,%20java.rmi.activation.ActivationGroupDesc,%20long))([ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) id, [ActivationGroupDesc](http://docs.google.com/java/rmi/activation/ActivationGroupDesc.html) desc, long incarnation)            Create and set the activation group for the current VM. |
| static [ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) | [**currentGroupID**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#currentGroupID())()            Returns the current activation group's identifier. |
| static [ActivationSystem](http://docs.google.com/java/rmi/activation/ActivationSystem.html) | [**getSystem**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#getSystem())()            Returns the activation system for the VM. |
| protected  void | [**inactiveGroup**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#inactiveGroup())()            This protected method is necessary for subclasses to make the inactiveGroup callback to the group's monitor. |
| boolean | [**inactiveObject**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#inactiveObject(java.rmi.activation.ActivationID))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)            The group's inactiveObject method is called indirectly via a call to the Activatable.inactive method. |
| static void | [**setSystem**](http://docs.google.com/java/rmi/activation/ActivationGroup.html#setSystem(java.rmi.activation.ActivationSystem))([ActivationSystem](http://docs.google.com/java/rmi/activation/ActivationSystem.html) system)            Set the activation system for the VM. |

| **Methods inherited from class java.rmi.server.**[**UnicastRemoteObject**](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html) |
| --- |
| [clone](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html#clone()), [exportObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html#exportObject(java.rmi.Remote)), [exportObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html#exportObject(java.rmi.Remote,%20int)), [exportObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html#exportObject(java.rmi.Remote,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory)), [unexportObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html#unexportObject(java.rmi.Remote,%20boolean)) |

| **Methods inherited from class java.rmi.server.**[**RemoteServer**](http://docs.google.com/java/rmi/server/RemoteServer.html) |
| --- |
| [getClientHost](http://docs.google.com/java/rmi/server/RemoteServer.html#getClientHost()), [getLog](http://docs.google.com/java/rmi/server/RemoteServer.html#getLog()), [setLog](http://docs.google.com/java/rmi/server/RemoteServer.html#setLog(java.io.OutputStream)) |

| **Methods inherited from class java.rmi.server.**[**RemoteObject**](http://docs.google.com/java/rmi/server/RemoteObject.html) |
| --- |
| [equals](http://docs.google.com/java/rmi/server/RemoteObject.html#equals(java.lang.Object)), [getRef](http://docs.google.com/java/rmi/server/RemoteObject.html#getRef()), [hashCode](http://docs.google.com/java/rmi/server/RemoteObject.html#hashCode()), [toString](http://docs.google.com/java/rmi/server/RemoteObject.html#toString()), [toStub](http://docs.google.com/java/rmi/server/RemoteObject.html#toStub(java.rmi.Remote)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [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)) |

| **Methods inherited from interface java.rmi.activation.**[**ActivationInstantiator**](http://docs.google.com/java/rmi/activation/ActivationInstantiator.html) |
| --- |
| [newInstance](http://docs.google.com/java/rmi/activation/ActivationInstantiator.html#newInstance(java.rmi.activation.ActivationID,%20java.rmi.activation.ActivationDesc)) |

| **Constructor Detail** |
| --- |

### ActivationGroup

protected **ActivationGroup**([ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) groupID)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Constructs an activation group with the given activation group identifier. The group is exported as a java.rmi.server.UnicastRemoteObject.

**Parameters:**groupID - the group's identifier **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if this group could not be exported**Since:** 1.2

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

### inactiveObject

public boolean **inactiveObject**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)  
 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)

The group's inactiveObject method is called indirectly via a call to the Activatable.inactive method. A remote object implementation must call Activatable's inactive method when that object deactivates (the object deems that it is no longer active). If the object does not call Activatable.inactive when it deactivates, the object will never be garbage collected since the group keeps strong references to the objects it creates.

The group's inactiveObject method unexports the remote object from the RMI runtime so that the object can no longer receive incoming RMI calls. An object will only be unexported if the object has no pending or executing calls. The subclass of ActivationGroup must override this method and unexport the object.

After removing the object from the RMI runtime, the group must inform its ActivationMonitor (via the monitor's inactiveObject method) that the remote object is not currently active so that the remote object will be re-activated by the activator upon a subsequent activation request.

This method simply informs the group's monitor that the object is inactive. It is up to the concrete subclass of ActivationGroup to fulfill the additional requirement of unexporting the object.

**Parameters:**id - the object's activation identifier **Returns:**true if the object was successfully deactivated; otherwise returns false. **Throws:** [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object is unknown (may already be inactive) [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if call informing monitor fails [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if group is inactive**Since:** 1.2

### activeObject

public abstract void **activeObject**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 [Remote](http://docs.google.com/java/rmi/Remote.html) obj)  
 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)

The group's activeObject method is called when an object is exported (either by Activatable object construction or an explicit call to Activatable.exportObject. The group must inform its ActivationMonitor that the object is active (via the monitor's activeObject method) if the group hasn't already done so.

**Parameters:**id - the object's identifierobj - the remote object implementation **Throws:** [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object is not registered [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if call informing monitor fails [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if group is inactive**Since:** 1.2

### createGroup

public static [ActivationGroup](http://docs.google.com/java/rmi/activation/ActivationGroup.html) **createGroup**([ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) id,  
 [ActivationGroupDesc](http://docs.google.com/java/rmi/activation/ActivationGroupDesc.html) desc,  
 long incarnation)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html)

Create and set the activation group for the current VM. The activation group can only be set if it is not currently set. An activation group is set using the createGroup method when the Activator initiates the re-creation of an activation group in order to carry out incoming activate requests. A group must first be registered with the ActivationSystem before it can be created via this method.

The group class specified by the ActivationGroupDesc must be a concrete subclass of ActivationGroup and have a public constructor that takes two arguments: the ActivationGroupID for the group and the MarshalledObject containing the group's initialization data (obtained from the ActivationGroupDesc.

If the group class name specified in the ActivationGroupDesc is null, then this method will behave as if the group descriptor contained the name of the default activation group implementation class.

Note that if your application creates its own custom activation group, a security manager must be set for that group. Otherwise objects cannot be activated in the group. java.rmi.RMISecurityManager is set by default.

If a security manager is already set in the group VM, this method first calls the security manager's checkSetFactory method. This could result in a SecurityException. If your application needs to set a different security manager, you must ensure that the policy file specified by the group's ActivationGroupDesc grants the group the necessary permissions to set a new security manager. (Note: This will be necessary if your group downloads and sets a security manager).

After the group is created, the ActivationSystem is informed that the group is active by calling the activeGroup method which returns the ActivationMonitor for the group. The application need not call activeGroup independently since it is taken care of by this method.

Once a group is created, subsequent calls to the currentGroupID method will return the identifier for this group until the group becomes inactive.

**Parameters:**id - the activation group's identifierdesc - the activation group's descriptorincarnation - the group's incarnation number (zero on group's initial creation) **Returns:**the activation group for the VM **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if group already exists or if error occurs during group creation [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if permission to create group is denied. (Note: The default implementation of the security manager checkSetFactory method requires the RuntimePermission "setFactory")**Since:** 1.2 **See Also:**[SecurityManager.checkSetFactory()](http://docs.google.com/java/lang/SecurityManager.html#checkSetFactory())

### currentGroupID

public static [ActivationGroupID](http://docs.google.com/java/rmi/activation/ActivationGroupID.html) **currentGroupID**()

Returns the current activation group's identifier. Returns null if no group is currently active for this VM.

**Returns:**the activation group's identifier**Since:** 1.2

### setSystem

public static void **setSystem**([ActivationSystem](http://docs.google.com/java/rmi/activation/ActivationSystem.html) system)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html)

Set the activation system for the VM. The activation system can only be set it if no group is currently active. If the activation system is not set via this call, then the getSystem method attempts to obtain a reference to the ActivationSystem by looking up the name "java.rmi.activation.ActivationSystem" in the Activator's registry. By default, the port number used to look up the activation system is defined by ActivationSystem.SYSTEM\_PORT. This port can be overridden by setting the property java.rmi.activation.port.

If there is a security manager, this method first calls the security manager's checkSetFactory method. This could result in a SecurityException.

**Parameters:**system - remote reference to the ActivationSystem **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation system is already set [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if permission to set the activation system is denied. (Note: The default implementation of the security manager checkSetFactory method requires the RuntimePermission "setFactory")**Since:** 1.2 **See Also:**[getSystem()](http://docs.google.com/java/rmi/activation/ActivationGroup.html#getSystem()), [SecurityManager.checkSetFactory()](http://docs.google.com/java/lang/SecurityManager.html#checkSetFactory())

### getSystem

public static [ActivationSystem](http://docs.google.com/java/rmi/activation/ActivationSystem.html) **getSystem**()  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html)

Returns the activation system for the VM. The activation system may be set by the setSystem method. If the activation system is not set via the setSystem method, then the getSystem method attempts to obtain a reference to the ActivationSystem by looking up the name "java.rmi.activation.ActivationSystem" in the Activator's registry. By default, the port number used to look up the activation system is defined by ActivationSystem.SYSTEM\_PORT. This port can be overridden by setting the property java.rmi.activation.port.

**Returns:**the activation system for the VM/group **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation system cannot be obtained or is not bound (means that it is not running)**Since:** 1.2 **See Also:**[setSystem(java.rmi.activation.ActivationSystem)](http://docs.google.com/java/rmi/activation/ActivationGroup.html#setSystem(java.rmi.activation.ActivationSystem))

### activeObject

protected void **activeObject**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<? extends [Remote](http://docs.google.com/java/rmi/Remote.html)> mobj)  
 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)

This protected method is necessary for subclasses to make the activeObject callback to the group's monitor. The call is simply forwarded to the group's ActivationMonitor.

**Parameters:**id - the object's identifiermobj - a marshalled object containing the remote object's stub **Throws:** [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object is not registered [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if call informing monitor fails [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if an activation error occurs**Since:** 1.2

### inactiveGroup

protected void **inactiveGroup**()  
 throws [UnknownGroupException](http://docs.google.com/java/rmi/activation/UnknownGroupException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

This protected method is necessary for subclasses to make the inactiveGroup callback to the group's monitor. The call is simply forwarded to the group's ActivationMonitor. Also, the current group for the VM is set to null.

**Throws:** [UnknownGroupException](http://docs.google.com/java/rmi/activation/UnknownGroupException.html) - if group is not registered [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if call informing monitor 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/ActivationGroup.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/ActivationException.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/ActivationGroup_Stub.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/ActivationGroup.html)    [**NO FRAMES**](http://docs.google.com/ActivationGroup.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).