| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Activatable.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   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/ActivateFailedException.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/Activatable.html)    [**NO FRAMES**](http://docs.google.com/Activatable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

## **java.rmi.activation**

Class Activatable

[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.activation.Activatable**

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

public abstract class **Activatable**extends [RemoteServer](http://docs.google.com/java/rmi/server/RemoteServer.html)

The Activatable class provides support for remote objects that require persistent access over time and that can be activated by the system.

For the constructors and static exportObject methods, the stub for a remote object being exported is obtained as described in [UnicastRemoteObject](http://docs.google.com/java/rmi/server/UnicastRemoteObject.html).

An attempt to serialize explicitly an instance of this class will fail.

**Since:** 1.2

| **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 | [**Activatable**](http://docs.google.com/java/rmi/activation/Activatable.html#Activatable(java.rmi.activation.ActivationID,%20int))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, int port)            Constructor used to activate/export the object on a specified port. |
| protected | [**Activatable**](http://docs.google.com/java/rmi/activation/Activatable.html#Activatable(java.rmi.activation.ActivationID,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, int port, [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf, [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)            Constructor used to activate/export the object on a specified port. |
| protected | [**Activatable**](http://docs.google.com/java/rmi/activation/Activatable.html#Activatable(java.lang.String,%20java.rmi.MarshalledObject,%20boolean,%20int))([String](http://docs.google.com/java/lang/String.html) location, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data, boolean restart, int port)            Constructs an activatable remote object by registering an activation descriptor (with the specified location, data, and restart mode) for this object, and exporting the object with the specified port. |
| protected | [**Activatable**](http://docs.google.com/java/rmi/activation/Activatable.html#Activatable(java.lang.String,%20java.rmi.MarshalledObject,%20boolean,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory))([String](http://docs.google.com/java/lang/String.html) location, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data, boolean restart, int port, [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf, [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)            Constructs an activatable remote object by registering an activation descriptor (with the specified location, data, and restart mode) for this object, and exporting the object with the specified port, and specified client and server socket factories. |

| **Method Summary** | |
| --- | --- |
| static [Remote](http://docs.google.com/java/rmi/Remote.html) | [**exportObject**](http://docs.google.com/java/rmi/activation/Activatable.html#exportObject(java.rmi.Remote,%20java.rmi.activation.ActivationID,%20int))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, int port)            Export the activatable remote object to the RMI runtime to make the object available to receive incoming calls. |
| static [Remote](http://docs.google.com/java/rmi/Remote.html) | [**exportObject**](http://docs.google.com/java/rmi/activation/Activatable.html#exportObject(java.rmi.Remote,%20java.rmi.activation.ActivationID,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id, int port, [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf, [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)            Export the activatable remote object to the RMI runtime to make the object available to receive incoming calls. |
| static [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) | [**exportObject**](http://docs.google.com/java/rmi/activation/Activatable.html#exportObject(java.rmi.Remote,%20java.lang.String,%20java.rmi.MarshalledObject,%20boolean,%20int))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, [String](http://docs.google.com/java/lang/String.html) location, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data, boolean restart, int port)            Registers an activation descriptor (with the specified location, data, and restart mode) for the specified object, and exports that object with the specified port. |
| static [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) | [**exportObject**](http://docs.google.com/java/rmi/activation/Activatable.html#exportObject(java.rmi.Remote,%20java.lang.String,%20java.rmi.MarshalledObject,%20boolean,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, [String](http://docs.google.com/java/lang/String.html) location, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data, boolean restart, int port, [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf, [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)            Registers an activation descriptor (with the specified location, data, and restart mode) for the specified object, and exports that object with the specified port, and the specified client and server socket factories. |
| protected  [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) | [**getID**](http://docs.google.com/java/rmi/activation/Activatable.html#getID())()            Returns the object's activation identifier. |
| static boolean | [**inactive**](http://docs.google.com/java/rmi/activation/Activatable.html#inactive(java.rmi.activation.ActivationID))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)            Informs the system that the object with the corresponding activation id is currently inactive. |
| static [Remote](http://docs.google.com/java/rmi/Remote.html) | [**register**](http://docs.google.com/java/rmi/activation/Activatable.html#register(java.rmi.activation.ActivationDesc))([ActivationDesc](http://docs.google.com/java/rmi/activation/ActivationDesc.html) desc)            Register an object descriptor for an activatable remote object so that is can be activated on demand. |
| static boolean | [**unexportObject**](http://docs.google.com/java/rmi/activation/Activatable.html#unexportObject(java.rmi.Remote,%20boolean))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, boolean force)            Remove the remote object, obj, from the RMI runtime. |
| static void | [**unregister**](http://docs.google.com/java/rmi/activation/Activatable.html#unregister(java.rmi.activation.ActivationID))([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)            Revokes previous registration for the activation descriptor associated with id. |

| **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) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [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)) |

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

### Activatable

protected **Activatable**([String](http://docs.google.com/java/lang/String.html) location,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data,  
 boolean restart,  
 int port)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Constructs an activatable remote object by registering an activation descriptor (with the specified location, data, and restart mode) for this object, and exporting the object with the specified port.

**Note:** Using the Activatable constructors that both register and export an activatable remote object is strongly discouraged because the actions of registering and exporting the remote object are *not* guaranteed to be atomic. Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

This method invokes the exportObject method with this object, and the specified location, data, restart mode, and port. Subsequent calls to [getID()](http://docs.google.com/java/rmi/activation/Activatable.html#getID()) will return the activation identifier returned from the call to exportObject.

**Parameters:**location - the location for classes for this objectdata - the object's initialization dataport - the port on which the object is exported (an anonymous port is used if port=0)restart - if true, the object is restarted (reactivated) when either the activator is restarted or the object's activation group is restarted after an unexpected crash; if false, the object is only activated on demand. Specifying restart to be true does not force an initial immediate activation of a newly registered object; initial activation is lazy. **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if object registration fails. [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if either of the following fails: a) registering the object with the activation system or b) exporting the object to the RMI runtime.**Since:** 1.2

### Activatable

protected **Activatable**([String](http://docs.google.com/java/lang/String.html) location,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data,  
 boolean restart,  
 int port,  
 [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf,  
 [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Constructs an activatable remote object by registering an activation descriptor (with the specified location, data, and restart mode) for this object, and exporting the object with the specified port, and specified client and server socket factories.

**Note:** Using the Activatable constructors that both register and export an activatable remote object is strongly discouraged because the actions of registering and exporting the remote object are *not* guaranteed to be atomic. Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

This method invokes the exportObject method with this object, and the specified location, data, restart mode, port, and client and server socket factories. Subsequent calls to [getID()](http://docs.google.com/java/rmi/activation/Activatable.html#getID()) will return the activation identifier returned from the call to exportObject.

**Parameters:**location - the location for classes for this objectdata - the object's initialization datarestart - if true, the object is restarted (reactivated) when either the activator is restarted or the object's activation group is restarted after an unexpected crash; if false, the object is only activated on demand. Specifying restart to be true does not force an initial immediate activation of a newly registered object; initial activation is lazy.port - the port on which the object is exported (an anonymous port is used if port=0)csf - the client-side socket factory for making calls to the remote objectssf - the server-side socket factory for receiving remote calls **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if object registration fails. [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if either of the following fails: a) registering the object with the activation system or b) exporting the object to the RMI runtime.**Since:** 1.2

### Activatable

protected **Activatable**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 int port)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Constructor used to activate/export the object on a specified port. An "activatable" remote object must have a constructor that takes two arguments:

* the object's activation identifier (ActivationID), and
* the object's initialization data (a MarshalledObject).

A concrete subclass of this class must call this constructor when it is *activated* via the two parameter constructor described above. As a side-effect of construction, the remote object is "exported" to the RMI runtime (on the specified port) and is available to accept incoming calls from clients.

**Parameters:**id - activation identifier for the objectport - the port number on which the object is exported **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if exporting the object to the RMI runtime fails**Since:** 1.2

### Activatable

protected **Activatable**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 int port,  
 [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf,  
 [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Constructor used to activate/export the object on a specified port. An "activatable" remote object must have a constructor that takes two arguments:

* the object's activation identifier (ActivationID), and
* the object's initialization data (a MarshalledObject).

A concrete subclass of this class must call this constructor when it is *activated* via the two parameter constructor described above. As a side-effect of construction, the remote object is "exported" to the RMI runtime (on the specified port) and is available to accept incoming calls from clients.

**Parameters:**id - activation identifier for the objectport - the port number on which the object is exportedcsf - the client-side socket factory for making calls to the remote objectssf - the server-side socket factory for receiving remote calls **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if exporting the object to the RMI runtime fails**Since:** 1.2

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

### getID

protected [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) **getID**()

Returns the object's activation identifier. The method is protected so that only subclasses can obtain an object's identifier.

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

### register

public static [Remote](http://docs.google.com/java/rmi/Remote.html) **register**([ActivationDesc](http://docs.google.com/java/rmi/activation/ActivationDesc.html) desc)  
 throws [UnknownGroupException](http://docs.google.com/java/rmi/activation/UnknownGroupException.html),  
 [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Register an object descriptor for an activatable remote object so that is can be activated on demand.

**Parameters:**desc - the object's descriptor **Returns:**the stub for the activatable remote object **Throws:** [UnknownGroupException](http://docs.google.com/java/rmi/activation/UnknownGroupException.html) - if group id in desc is not registered with the activation system [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation system is not running [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if remote call fails**Since:** 1.2

### inactive

public static boolean **inactive**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)  
 throws [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html),  
 [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Informs the system that the object with the corresponding activation id is currently inactive. If the object is currently active, the object is "unexported" from the RMI runtime (only if there are no pending or in-progress calls) so the that it can no longer receive incoming calls. This call informs this VM's ActivationGroup that the object is inactive, that, in turn, informs its ActivationMonitor. If this call completes successfully, a subsequent activate request to the activator will cause the object to reactivate. The operation may still succeed if the object is considered active but has already unexported itself.

**Parameters:**id - the object's activation identifier **Returns:**true if the operation succeeds (the operation will succeed if the object in currently known to be active and is either already unexported or is currently exported and has no pending/executing calls); false is returned if the object has pending/executing calls in which case it cannot be deactivated **Throws:** [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object is not known (it may already be inactive) [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if group is not active [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if call informing monitor fails**Since:** 1.2

### unregister

public static void **unregister**([ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id)  
 throws [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html),  
 [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Revokes previous registration for the activation descriptor associated with id. An object can no longer be activated via that id.

**Parameters:**id - the object's activation identifier **Throws:** [UnknownObjectException](http://docs.google.com/java/rmi/activation/UnknownObjectException.html) - if object (id) is unknown [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation system is not running [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if remote call to activation system fails**Since:** 1.2

### exportObject

public static [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) **exportObject**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 [String](http://docs.google.com/java/lang/String.html) location,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data,  
 boolean restart,  
 int port)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Registers an activation descriptor (with the specified location, data, and restart mode) for the specified object, and exports that object with the specified port.

**Note:** Using this method (as well as the Activatable constructors that both register and export an activatable remote object) is strongly discouraged because the actions of registering and exporting the remote object are *not* guaranteed to be atomic. Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

This method invokes the exportObject method with the specified object, location, data, restart mode, and port, and null for both client and server socket factories, and then returns the resulting activation identifier.

**Parameters:**obj - the object being exportedlocation - the object's code locationdata - the object's bootstrapping datarestart - if true, the object is restarted (reactivated) when either the activator is restarted or the object's activation group is restarted after an unexpected crash; if false, the object is only activated on demand. Specifying restart to be true does not force an initial immediate activation of a newly registered object; initial activation is lazy.port - the port on which the object is exported (an anonymous port is used if port=0) **Returns:**the activation identifier obtained from registering the descriptor, desc, with the activation system the wrong group **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation group is not active [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if object registration or export fails**Since:** 1.2

### exportObject

public static [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) **exportObject**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 [String](http://docs.google.com/java/lang/String.html) location,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)<?> data,  
 boolean restart,  
 int port,  
 [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf,  
 [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)  
 throws [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html),  
 [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Registers an activation descriptor (with the specified location, data, and restart mode) for the specified object, and exports that object with the specified port, and the specified client and server socket factories.

**Note:** Using this method (as well as the Activatable constructors that both register and export an activatable remote object) is strongly discouraged because the actions of registering and exporting the remote object are *not* guaranteed to be atomic. Instead, an application should register an activation descriptor and export a remote object separately, so that exceptions can be handled properly.

This method first registers an activation descriptor for the specified object as follows. It obtains the activation system by invoking the method [ActivationGroup.getSystem](http://docs.google.com/java/rmi/activation/ActivationGroup.html#getSystem()). This method then obtains an [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) for the object by invoking the activation system's [registerObject](http://docs.google.com/java/rmi/activation/ActivationSystem.html#registerObject(java.rmi.activation.ActivationDesc)) method with an [ActivationDesc](http://docs.google.com/java/rmi/activation/ActivationDesc.html) constructed with the specified object's class name, and the specified location, data, and restart mode. If an exception occurs obtaining the activation system or registering the activation descriptor, that exception is thrown to the caller.

Next, this method exports the object by invoking the [exportObject](http://docs.google.com/java/rmi/activation/Activatable.html#exportObject(java.rmi.Remote,%20java.rmi.activation.ActivationID,%20int,%20java.rmi.server.RMIClientSocketFactory,%20java.rmi.server.RMIServerSocketFactory)) method with the specified remote object, the activation identifier obtained from registration, the specified port, and the specified client and server socket factories. If an exception occurs exporting the object, this method attempts to unregister the activation identifier (obtained from registration) by invoking the activation system's [unregisterObject](http://docs.google.com/java/rmi/activation/ActivationSystem.html#unregisterObject(java.rmi.activation.ActivationID)) method with the activation identifier. If an exception occurs unregistering the identifier, that exception is ignored, and the original exception that occurred exporting the object is thrown to the caller.

Finally, this method invokes the [activeObject](http://docs.google.com/java/rmi/activation/ActivationGroup.html#activeObject(java.rmi.activation.ActivationID,%20java.rmi.Remote)) method on the activation group in this VM with the activation identifier and the specified remote object, and returns the activation identifier to the caller.

**Parameters:**obj - the object being exportedlocation - the object's code locationdata - the object's bootstrapping datarestart - if true, the object is restarted (reactivated) when either the activator is restarted or the object's activation group is restarted after an unexpected crash; if false, the object is only activated on demand. Specifying restart to be true does not force an initial immediate activation of a newly registered object; initial activation is lazy.port - the port on which the object is exported (an anonymous port is used if port=0)csf - the client-side socket factory for making calls to the remote objectssf - the server-side socket factory for receiving remote calls **Returns:**the activation identifier obtained from registering the descriptor with the activation system **Throws:** [ActivationException](http://docs.google.com/java/rmi/activation/ActivationException.html) - if activation group is not active [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if object registration or export fails**Since:** 1.2

### exportObject

public static [Remote](http://docs.google.com/java/rmi/Remote.html) **exportObject**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 int port)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Export the activatable remote object to the RMI runtime to make the object available to receive incoming calls. The object is exported on an anonymous port, if port is zero.

During activation, this exportObject method should be invoked explicitly by an "activatable" object, that does not extend the Activatable class. There is no need for objects that do extend the Activatable class to invoke this method directly because the object is exported during construction.

**Parameters:**obj - the remote object implementationid - the object's activation identifierport - the port on which the object is exported (an anonymous port is used if port=0) **Returns:**the stub for the activatable remote object **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if object export fails**Since:** 1.2

### exportObject

public static [Remote](http://docs.google.com/java/rmi/Remote.html) **exportObject**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 [ActivationID](http://docs.google.com/java/rmi/activation/ActivationID.html) id,  
 int port,  
 [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) csf,  
 [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) ssf)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

Export the activatable remote object to the RMI runtime to make the object available to receive incoming calls. The object is exported on an anonymous port, if port is zero.

During activation, this exportObject method should be invoked explicitly by an "activatable" object, that does not extend the Activatable class. There is no need for objects that do extend the Activatable class to invoke this method directly because the object is exported during construction.

**Parameters:**obj - the remote object implementationid - the object's activation identifierport - the port on which the object is exported (an anonymous port is used if port=0)csf - the client-side socket factory for making calls to the remote objectssf - the server-side socket factory for receiving remote calls **Returns:**the stub for the activatable remote object **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if object export fails**Since:** 1.2

### unexportObject

public static boolean **unexportObject**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 boolean force)  
 throws [NoSuchObjectException](http://docs.google.com/java/rmi/NoSuchObjectException.html)

Remove the remote object, obj, from the RMI runtime. If successful, the object can no longer accept incoming RMI calls. If the force parameter is true, the object is forcibly unexported even if there are pending calls to the remote object or the remote object still has calls in progress. If the force parameter is false, the object is only unexported if there are no pending or in progress calls to the object.

**Parameters:**obj - the remote object to be unexportedforce - if true, unexports the object even if there are pending or in-progress calls; if false, only unexports the object if there are no pending or in-progress calls **Returns:**true if operation is successful, false otherwise **Throws:** [NoSuchObjectException](http://docs.google.com/java/rmi/NoSuchObjectException.html) - if the remote object is not currently exported**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/Activatable.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   [**NEXT CLASS**](http://docs.google.com/java/rmi/activation/ActivateFailedException.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/activation/Activatable.html)    [**NO FRAMES**](http://docs.google.com/Activatable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

[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).