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

## **java.rmi.dgc**

Interface DGC

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

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

The DGC abstraction is used for the server side of the distributed garbage collection algorithm. This interface contains the two methods: dirty and clean. A dirty call is made when a remote reference is unmarshaled in a client (the client is indicated by its VMID). A corresponding clean call is made when no more references to the remote reference exist in the client. A failed dirty call must schedule a strong clean call so that the call's sequence number can be retained in order to detect future calls received out of order by the distributed garbage collector. A reference to a remote object is leased for a period of time by the client holding the reference. The lease period starts when the dirty call is received. It is the client's responsibility to renew the leases, by making additional dirty calls, on the remote references it holds before such leases expire. If the client does not renew the lease before it expires, the distributed garbage collector assumes that the remote object is no longer referenced by that client.

| **Method Summary** | |
| --- | --- |
| void | [**clean**](http://docs.google.com/java/rmi/dgc/DGC.html#clean(java.rmi.server.ObjID%5B%5D,%20long,%20java.rmi.dgc.VMID,%20boolean))([ObjID](http://docs.google.com/java/rmi/server/ObjID.html)[] ids, long sequenceNum, [VMID](http://docs.google.com/java/rmi/dgc/VMID.html) vmid, boolean strong)            The clean call removes the 'vmid' from the reference list of each remote object indicated in 'id's. |
| [Lease](http://docs.google.com/java/rmi/dgc/Lease.html) | [**dirty**](http://docs.google.com/java/rmi/dgc/DGC.html#dirty(java.rmi.server.ObjID%5B%5D,%20long,%20java.rmi.dgc.Lease))([ObjID](http://docs.google.com/java/rmi/server/ObjID.html)[] ids, long sequenceNum, [Lease](http://docs.google.com/java/rmi/dgc/Lease.html) lease)            The dirty call requests leases for the remote object references associated with the object identifiers contained in the array 'ids'. |

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

### dirty

[Lease](http://docs.google.com/java/rmi/dgc/Lease.html) **dirty**([ObjID](http://docs.google.com/java/rmi/server/ObjID.html)[] ids,  
 long sequenceNum,  
 [Lease](http://docs.google.com/java/rmi/dgc/Lease.html) lease)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

The dirty call requests leases for the remote object references associated with the object identifiers contained in the array 'ids'. The 'lease' contains a client's unique VM identifier (VMID) and a requested lease period. For each remote object exported in the local VM, the garbage collector maintains a reference list-a list of clients that hold references to it. If the lease is granted, the garbage collector adds the client's VMID to the reference list for each remote object indicated in 'ids'. The 'sequenceNum' parameter is a sequence number that is used to detect and discard late calls to the garbage collector. The sequence number should always increase for each subsequent call to the garbage collector. Some clients are unable to generate a VMID, since a VMID is a universally unique identifier that contains a host address which some clients are unable to obtain due to security restrictions. In this case, a client can use a VMID of null, and the distributed garbage collector will assign a VMID for the client. The dirty call returns a Lease object that contains the VMID used and the lease period granted for the remote references (a server may decide to grant a smaller lease period than the client requests). A client must use the VMID the garbage collector uses in order to make corresponding clean calls when the client drops remote object references. A client VM need only make one initial dirty call for each remote reference referenced in the VM (even if it has multiple references to the same remote object). The client must also make a dirty call to renew leases on remote references before such leases expire. When the client no longer has any references to a specific remote object, it must schedule a clean call for the object ID associated with the reference.

**Parameters:**ids - IDs of objects to mark as referenced by calling clientsequenceNum - sequence numberlease - requested lease **Returns:**granted lease **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if dirty call fails

### clean

void **clean**([ObjID](http://docs.google.com/java/rmi/server/ObjID.html)[] ids,  
 long sequenceNum,  
 [VMID](http://docs.google.com/java/rmi/dgc/VMID.html) vmid,  
 boolean strong)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

The clean call removes the 'vmid' from the reference list of each remote object indicated in 'id's. The sequence number is used to detect late clean calls. If the argument 'strong' is true, then the clean call is a result of a failed dirty call, thus the sequence number for the client 'vmid' needs to be remembered.

**Parameters:**ids - IDs of objects to mark as unreferenced by calling clientsequenceNum - sequence numbervmid - client VMIDstrong - make 'strong' clean call **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if clean call fails

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DGC.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/dgc/Lease.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/dgc/DGC.html)    [**NO FRAMES**](http://docs.google.com/DGC.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).