| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RemoteRef.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/server/RemoteObjectInvocationHandler.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/server/RemoteServer.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/server/RemoteRef.html)    [**NO FRAMES**](http://docs.google.com/RemoteRef.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

## **java.rmi.server**

Interface RemoteRef

**All Superinterfaces:** [Externalizable](http://docs.google.com/java/io/Externalizable.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **All Known Subinterfaces:** [ServerRef](http://docs.google.com/java/rmi/server/ServerRef.html)

public interface **RemoteRef**extends [Externalizable](http://docs.google.com/java/io/Externalizable.html)

RemoteRef represents the handle for a remote object. A RemoteStub uses a remote reference to carry out a remote method invocation to a remote object.

**Since:** JDK1.1 **See Also:**[RemoteStub](http://docs.google.com/java/rmi/server/RemoteStub.html)

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**packagePrefix**](http://docs.google.com/java/rmi/server/RemoteRef.html#packagePrefix)            Initialize the server package prefix: assumes that the implementation of server ref classes (e.g., UnicastRef, UnicastServerRef) are located in the package defined by the prefix. |
| static long | [**serialVersionUID**](http://docs.google.com/java/rmi/server/RemoteRef.html#serialVersionUID)            indicate compatibility with JDK 1.1.x version of class. |

| **Method Summary** | |
| --- | --- |
| void | [**done**](http://docs.google.com/java/rmi/server/RemoteRef.html#done(java.rmi.server.RemoteCall))([RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) call)  **Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| [String](http://docs.google.com/java/lang/String.html) | [**getRefClass**](http://docs.google.com/java/rmi/server/RemoteRef.html#getRefClass(java.io.ObjectOutput))([ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html) out)            Returns the class name of the ref type to be serialized onto the stream 'out'. |
| void | [**invoke**](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.server.RemoteCall))([RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) call)  **Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.Remote,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D,%20long))([Remote](http://docs.google.com/java/rmi/Remote.html) obj, [Method](http://docs.google.com/java/lang/reflect/Method.html) method, [Object](http://docs.google.com/java/lang/Object.html)[] params, long opnum)            Invoke a method. |
| [RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) | [**newCall**](http://docs.google.com/java/rmi/server/RemoteRef.html#newCall(java.rmi.server.RemoteObject,%20java.rmi.server.Operation%5B%5D,%20int,%20long))([RemoteObject](http://docs.google.com/java/rmi/server/RemoteObject.html) obj, [Operation](http://docs.google.com/java/rmi/server/Operation.html)[] op, int opnum, long hash)  **Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls on the stub's the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| boolean | [**remoteEquals**](http://docs.google.com/java/rmi/server/RemoteRef.html#remoteEquals(java.rmi.server.RemoteRef))([RemoteRef](http://docs.google.com/java/rmi/server/RemoteRef.html) obj)            Compares two remote objects for equality. |
| int | [**remoteHashCode**](http://docs.google.com/java/rmi/server/RemoteRef.html#remoteHashCode())()            Returns a hashcode for a remote object. |
| [String](http://docs.google.com/java/lang/String.html) | [**remoteToString**](http://docs.google.com/java/rmi/server/RemoteRef.html#remoteToString())()            Returns a String that represents the reference of this remote object. |

| **Methods inherited from interface java.io.**[**Externalizable**](http://docs.google.com/java/io/Externalizable.html) |
| --- |
| [readExternal](http://docs.google.com/java/io/Externalizable.html#readExternal(java.io.ObjectInput)), [writeExternal](http://docs.google.com/java/io/Externalizable.html#writeExternal(java.io.ObjectOutput)) |

| **Field Detail** |
| --- |

### serialVersionUID

static final long **serialVersionUID**

indicate compatibility with JDK 1.1.x version of class.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.rmi.server.RemoteRef.serialVersionUID)

### packagePrefix

static final [String](http://docs.google.com/java/lang/String.html) **packagePrefix**

Initialize the server package prefix: assumes that the implementation of server ref classes (e.g., UnicastRef, UnicastServerRef) are located in the package defined by the prefix.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.rmi.server.RemoteRef.packagePrefix)

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

### invoke

[Object](http://docs.google.com/java/lang/Object.html) **invoke**([Remote](http://docs.google.com/java/rmi/Remote.html) obj,  
 [Method](http://docs.google.com/java/lang/reflect/Method.html) method,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 long opnum)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Invoke a method. This form of delegating method invocation to the reference allows the reference to take care of setting up the connection to the remote host, marshaling some representation for the method and parameters, then communicating the method invocation to the remote host. This method either returns the result of a method invocation on the remote object which resides on the remote host or throws a RemoteException if the call failed or an application-level exception if the remote invocation throws an exception.

**Parameters:**obj - the object that contains the RemoteRef (e.g., the RemoteStub for the object.method - the method to be invokedparams - the parameter listopnum - a hash that may be used to represent the method **Returns:**result of remote method invocation **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - if any exception occurs during remote method invocation**Since:** 1.2

### newCall

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
[RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) **newCall**([RemoteObject](http://docs.google.com/java/rmi/server/RemoteObject.html) obj,  
 [Operation](http://docs.google.com/java/rmi/server/Operation.html)[] op,  
 int opnum,  
 long hash)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

**Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls on the stub's the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.*

Creates an appropriate call object for a new remote method invocation on this object. Passing operation array and index, allows the stubs generator to assign the operation indexes and interpret them. The remote reference may need the operation to encode in the call.

**Parameters:**obj - remote stub through which to make callop - array of stub operationsopnum - operation numberhash - stub/skeleton interface hash **Returns:**call object representing remote call **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if failed to initiate new remote call**Since:** JDK1.1 **See Also:**[invoke(Remote,java.lang.reflect.Method,Object[],long)](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.Remote,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D,%20long))

### invoke

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
void **invoke**([RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) call)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

**Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.*

Executes the remote call. Invoke will raise any "user" exceptions which should pass through and not be caught by the stub. If any exception is raised during the remote invocation, invoke should take care of cleaning up the connection before raising the "user" or remote exception.

**Parameters:**call - object representing remote call **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - if any exception occurs during remote method**Since:** JDK1.1 **See Also:**[invoke(Remote,java.lang.reflect.Method,Object[],long)](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.Remote,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D,%20long))

### done

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
void **done**([RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html) call)  
 throws [RemoteException](http://docs.google.com/java/rmi/RemoteException.html)

**Deprecated.** *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.*

Allows the remote reference to clean up (or reuse) the connection. Done should only be called if the invoke returns successfully (non-exceptionally) to the stub.

**Parameters:**call - object representing remote call **Throws:** [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) - if remote error occurs during call cleanup**Since:** JDK1.1 **See Also:**[invoke(Remote,java.lang.reflect.Method,Object[],long)](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.Remote,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D,%20long))

### getRefClass

[String](http://docs.google.com/java/lang/String.html) **getRefClass**([ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html) out)

Returns the class name of the ref type to be serialized onto the stream 'out'.

**Parameters:**out - the output stream to which the reference will be serialized **Returns:**the class name (without package qualification) of the reference type**Since:** JDK1.1

### remoteHashCode

int **remoteHashCode**()

Returns a hashcode for a remote object. Two remote object stubs that refer to the same remote object will have the same hash code (in order to support remote objects as keys in hash tables).

**Returns:**remote object hashcode**Since:** JDK1.1 **See Also:**[Hashtable](http://docs.google.com/java/util/Hashtable.html)

### remoteEquals

boolean **remoteEquals**([RemoteRef](http://docs.google.com/java/rmi/server/RemoteRef.html) obj)

Compares two remote objects for equality. Returns a boolean that indicates whether this remote object is equivalent to the specified Object. This method is used when a remote object is stored in a hashtable.

**Parameters:**obj - the Object to compare with **Returns:**true if these Objects are equal; false otherwise.**Since:** JDK1.1 **See Also:**[Hashtable](http://docs.google.com/java/util/Hashtable.html)

### remoteToString

[String](http://docs.google.com/java/lang/String.html) **remoteToString**()

Returns a String that represents the reference of this remote object.

**Returns:**string representing remote object reference**Since:** JDK1.1

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RemoteRef.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/server/RemoteObjectInvocationHandler.html)   [**NEXT CLASS**](http://docs.google.com/java/rmi/server/RemoteServer.html) | [**FRAMES**](http://docs.google.com/index.html?java/rmi/server/RemoteRef.html)    [**NO FRAMES**](http://docs.google.com/RemoteRef.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

[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).