Jini version[[edit](https://en.wikipedia.org/w/index.php?title=Java_remote_method_invocation&action=edit&section=2)]

[Jini](https://en.wikipedia.org/wiki/Jini) offers a more advanced version of RMI in Java. It functions similarly but provides more advanced security, object discovery capabilities, and other mechanisms for distributed object applications.[[1]](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_note-From_P2P_to_Web_Services_and_Grids_2005-1)

Example[[edit](https://en.wikipedia.org/w/index.php?title=Java_remote_method_invocation&action=edit&section=3)]

The following classes implement a simple client-server program using RMI that displays a message.

**RmiServer class** — listens to RMI requests and implements the interface which is used by the client to invoke remote methods.

**import** **java.rmi.Naming**;

**import** **java.rmi.RemoteException**;

**import** **java.rmi.server.UnicastRemoteObject**;

**import** **java.rmi.registry.\***;

**public** **class** **RmiServer** **extends** UnicastRemoteObject **implements** RmiServerIntf {

**public** **static** **final** String MESSAGE = "Hello World";

**public** RmiServer() **throws** RemoteException {

**super**(0); *// required to avoid the 'rmic' step, see below*

}

**public** String getMessage() {

**return** MESSAGE;

}

**public** **static** void main(String args[]) **throws** Exception {

System.out.println("RMI server started");

**try** { *//special exception handler for registry creation*

LocateRegistry.createRegistry(1099);

System.out.println("java RMI registry created.");

} **catch** (RemoteException e) {

*//do nothing, error means registry already exists*

System.out.println("java RMI registry already exists.");

}

*//Instantiate RmiServer*

RmiServer server = **new** RmiServer();

*// Bind this object instance to the name "RmiServer"*

Naming.rebind("//localhost/RmiServer", server);

System.out.println("PeerServer bound in registry");

}

}

**RmiServerIntf interface** — defines the interface that is used by the client and implemented by the server.

**import** **java.rmi.Remote**;

**import** **java.rmi.RemoteException**;

**public** **interface** **RmiServerIntf** **extends** Remote {

String getMessage() **throws** RemoteException;

}

**RmiClient class** — this is the client which gets the reference (a proxy) to the remote object living on the server and invokes its method to get a message. If the server object implemented java.io.Serializable instead of java.rmi.Remote, it would be serialized and passed to the client as a value.[[2]](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_note-2)

**import** **java.rmi.Naming**;

**public** **class** **RmiClient** {

**public** **static** void main(String args[]) **throws** Exception {

RmiServerIntf server = (RmiServerIntf)Naming.lookup("//localhost/RmiServer");

System.out.println(server.getMessage());

}

}

Before running this example, we need to make a 'stub' file for the interface we used. For this task we have the RMI compiler - 'rmic'

* Note: we make a stub file from the '\*.class' file with the implementation of the remote interface, not from the '\*.java' file.

rmic RmiServer

Note that since version 5.0 of J2SE support for dynamically generated stub files has been added, and rmic is only provided for backwards compatibility with earlier runtimes,[[3]](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_note-3) or for programs that don't provide an explicit port number (or zero) when exporting remote objects, which is required for generated stubs to be possible, as described in the Javadoc for UnicastRemoteObject. See the comment in the constructor above.

References[[edit](https://en.wikipedia.org/w/index.php?title=Java_remote_method_invocation&action=edit&section=4)]

* 1. [**^**](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_ref-From_P2P_to_Web_Services_and_Grids_2005_1-0) *Taylor, Ian J (2005).*[*From P2P to Web Services and Grids : Peers in a Client/Server World*](https://archive.org/details/fromp2ptowebserv0000tayl)*. Computer Communications and Networks. London: Springer-Verlag.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1007/b138333*](https://doi.org/10.1007%2Fb138333)*.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*1852338695*](https://en.wikipedia.org/wiki/Special:BookSources/1852338695)*.*[*OCLC*](https://en.wikipedia.org/wiki/OCLC_(identifier))[*827073874*](https://www.worldcat.org/oclc/827073874)*.*[[*page needed*](https://en.wikipedia.org/wiki/Wikipedia:Citing_sources)]
  2. [**^**](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_ref-2) *Wilson, M. Jeff (November 10, 2000).*[*"Get smart with proxies and RMI"*](https://www.infoworld.com/article/2076234/get-smart-with-proxies-and-rmi.html)*.*[*JavaWorld*](https://en.wikipedia.org/wiki/JavaWorld)*. Retrieved 2020-07-18.*
  3. [**^**](https://en.wikipedia.org/wiki/Java_remote_method_invocation#cite_ref-3) [*"Java RMI Release Notes"*](http://docs.oracle.com/javase/1.5.0/docs/guide/rmi/relnotes.html)*. Oracle. Retrieved 9 May 2012.*

External links[[edit](https://en.wikipedia.org/w/index.php?title=Java_remote_method_invocation&action=edit&section=5)]

* [*"Remote Method Invocation Home"*](http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html)*. Oracle Technology Network for Java Developers. Redwood Shores, CA, USA:*[*Oracle Corporation*](https://en.wikipedia.org/wiki/Oracle_Corporation)*. Retrieved 2014-07-14.*
* [The Java RMI tutorial](http://download.oracle.com/javase/tutorial/rmi/index.html) - a good starting point to learn RMI. Also check the [Hello World in RMI](http://java.sun.com/j2se/1.5.0/docs/guide/rmi/hello/hello-world.html)
* [the Java RMI online training](http://java.sun.com/developer/onlineTraining/rmi/RMI.html) - Very good for training JavaRMI and as reference
* [The RMI page in the JDK docs](http://docs.oracle.com/javase/7/docs/technotes/guides/rmi/index.html)
* [java.rmi](https://docs.oracle.com/javase/10/docs/api/java/rmi/package-summary.html) (Sun's Java API Reference for the RMI package)
* *Ann Wollrath; Roger Riggs;*[*Jim Waldo*](https://en.wikipedia.org/wiki/Jim_Waldo)*.*[*"A Distributed Object Model for the Java System"*](http://pdos.csail.mit.edu/6.824/papers/waldo-rmi.pdf)*(PDF). Retrieved 2009-02-11.*
* [Programming WebLogic RMI](http://docs.oracle.com/cd/E12840_01/wls/docs103/rmi/rmi_intro.html) - an introduction to RMI in Oracle Weblogic.
* [General Remote Method Invocation](http://notes.corewebprogramming.com/student/RMI.pdf)