

## Overview

Java Remote Method Invocation (Java RMI) lets you create distributed applications in Java.

RMI allows an object to invoke methods of remote Java objects running on another Java Virtual Machine (JVM), possibly on different hosts. RMI uses object serialization to marshal and unmarshal parameters, and doesn't truncate types, supporting true object-oriented polymorphism.

## RMI Security Recommendations

Follow these recommendations to improve the security of your RMI applications.

- See [Serialization Filtering](#) and follow the best practices there to protect your applications.
- Follow [Secure Coding Guidelines for Java SE](#).
- Optionally, you can run a security manager when using RMI, either on a client or server.

### **WARNING:**

The Security Manager and APIs related to it have been deprecated and are subject to removal in a future release. There is no replacement for the Security Manager. See [JEP 411](#) for discussion and alternatives.

- Establish a reasonable security policy. For example, grant [SocketPermission](#) and allow listen, accept, connect, and resolve actions only among hosts communicating with RMI. Don't have the security policy grant [AllPermission](#). See Permissions in the Java Development Kit and Default Policy Implementation and Policy File Syntax.
- Restrict the communication to be local if RMI is being used only for communication among JVMs on the local host. To accomplish this task, specify the appropriate

socket permissions in the security policy file. Alternatively, you can use RMI APIs directly to restrict connections only to the local host. See the [RMISocketFactory](#) class.

- Ensure that the value of the `java.rmi.server.useCodebaseOnly` property is `True`. By default, the `java.rmi.server.useCodebaseOnly` property is set to `True`. If you set this property to `False`, then remote code loading is enabled, which increases the level of security risk to the system.
- Run RMI over Secure Sockets Layer (SSL)/Transport Layer Security (TLS) and request authentication for both server and client. This is possible using custom socket factories. An application can export a remote object to use custom socket factories that create sockets of a desired type (for example, SSL sockets). Using this technique, an application can use SSL socket communication instead of the default socket communication. See the following:
  - [SslRMIClientSocketFactory](#) class
  - [SslRMIServerSocketFactory](#) class
  - Java Secure Socket Extension (JSSE) Reference

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

---

Java Platform, Standard Edition Java Remote Method Invocation API Guide, Release 23  
G10091-01

Copyright © 1993, 2024, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit,

perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.