

**Oracle® Tuxedo**  
Interoperability  
12c Release 2 (12.1.3)

April 2014

Copyright © 1996, 2014, 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 or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on 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. 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.

# Contents

## 1. Interoperability and Coexistence

|   |      |
|---|------|
| Interoperability Defined . . . . .                | 1-1  |
| Intradomain Interoperability . . . . .            | 1-2  |
| Interdomain Interoperability . . . . .            | 1-4  |
| Client-Server Interoperability . . . . .          | 1-5  |
| Client/Server Affinity Interoperability . . . . . | 1-9  |
| Feature-Specific Interoperability . . . . .       | 1-9  |
| Interoperability with Third-Party ORBs . . . . .  | 1-10 |
| Product Upgrades . . . . .                        | 1-11 |
| Upward Application Compatibility . . . . .        | 1-12 |

## 2. Interoperability with Oracle WebLogic Server

|   |     |
|---|-----|
| Interoperability Software Components . . . . .    | 2-1 |
| Jolt Server Listener . . . . .                    | 2-2 |
| Jolt Server Handler . . . . .                     | 2-2 |
| Oracle Jolt for WebLogic Server . . . . .         | 2-3 |
| IIOP Listener . . . . .                           | 2-3 |
| IIOP Handler . . . . .                            | 2-3 |
| TDomain Gateway . . . . .                         | 2-3 |
| WebLogic Tuxedo Connector . . . . .               | 2-3 |
| Interoperability Programming Interfaces . . . . . | 2-4 |
| ATMI Interface . . . . .                          | 2-4 |

|  |      |
|--|------|
| JATMI Interface . . . . .  | 2-5  |
| Jolt Interface . . . . .   | 2-5  |
| RMI Interface . . . . .  | 2-5  |
| RMI-over-IIOP Interface. . . . .                                   | 2-5  |
| JSL/JSH-Jolt Unidirectional Connectivity . . . . .                 | 2-6  |
| Jolt Connection Pooling . . . . .                                  | 2-7  |
| Jolt Wire-Level Security . . . . .                                 | 2-7  |
| Jolt Transaction and Security Context Propagation . . . . .        | 2-8  |
| Jolt Documentation . . . . .                                       | 2-8  |
| TDomain-WTC Bidirectional Connectivity. . . . .                    | 2-8  |
| Oracle Tuxedo to Oracle WebLogic Server Connectivity. . . . .      | 2-8  |
| Oracle WebLogic Server to Oracle Tuxedo Connectivity. . . . .      | 2-10 |
| TDomain-WTC Wire-Level Security . . . . .                          | 2-11 |
| TDomain-WTC Transaction and Security Context Propagation . . . . . | 2-11 |
| TDomain and WTC Documentation . . . . .                            | 2-11 |
| RMI-over-IIOP Client Direct Connectivity to an EJB. . . . .        | 2-11 |
| Summary of Interoperability Capabilities . . . . .                 | 2-12 |
| Interoperability Sample Applications. . . . .                      | 2-13 |

# Interoperability and Coexistence

The following sections describe how Oracle Tuxedo 12c Release 2 (12.1.3) interoperates with older releases of the Oracle Tuxedo software, Oracle WebLogic Enterprise, and third-party products:

- [Interoperability Defined](#)
- [Intradomain Interoperability](#)
- [Interdomain Interoperability](#)
- [Client-Server Interoperability](#)
- [Interoperability with Third-Party ORBs](#)
- [Product Upgrades](#)
- [Upward Application Compatibility](#)

## Interoperability Defined

Interoperability permits the current Oracle Tuxedo release to communicate over a network connection with Oracle Tuxedo 12c Release 2 (12.1.3) or earlier software. Oracle Tuxedo supports *intradomain interoperability* and *interdomain interoperability*. Note the following:

- **Intradomain interoperability**  
Involves one machine in a multiple-machine Oracle Tuxedo domain (application) running Oracle Tuxedo 12c Release 2 (12.1.3) software, and another machine in the same domain

running Oracle Tuxedo 10gR3. Machines in a multiple-machine domain configuration communicate via Oracle Tuxedo Bridge processes.

- In a multiple-machine Oracle Tuxedo domain running Oracle Tuxedo 10gR3 or earlier system software, the *master* machine (and *master backup* machine if so configured) must run the highest release of the Oracle Tuxedo system software in the domain. Accordingly, the Oracle Tuxedo domain just described qualifies as an “Oracle Tuxedo domain running Oracle Tuxedo release 12c Release 2 (12.1.3) software.”

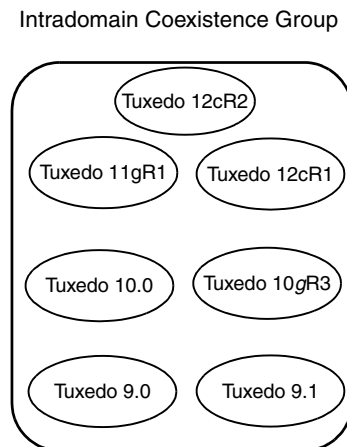
- Interdomain interoperability

Involves one Oracle Tuxedo domain running Oracle Tuxedo release 12c Release 2 (12.1.3) software, and another domain running Oracle Tuxedo release 10gR3 software. Domains involved in a multiple-domain (Domains) configuration communicate via Oracle Tuxedo domain gateway processes.

## Intradomain Interoperability

Message exchange and protocol compatibility exist in the Intradomain Coexistence Group as shown in [Figure 1-1](#):

**Figure 1-1 Intradomain Groups**



Oracle Tuxedo 12c Release 2 (12.1.3) can coexist in the same domain with Oracle Tuxedo 12cR1, 11gR1, 10gR3, 10.0, 9.1, and 9.0. In this environment, the propagation of transaction context

(transactional state information) and security context (user identity) between application clients and servers is fully supported. Administration is fully supported in this environment.

From Oracle Tuxedo 11g Release 1 (11.1.1.1.0), service name length is increased from 15 to 127. Resource name and remote service name length have also increased from 15 to 127 in domain configuration (for TDomain only). You can also configure a local service name or a remote service with a name less than or equal to 127 characters for TDOMAIN. Note the following:

- Long service names are not permitted when Oracle Tuxedo 10gR3 or earlier coexists in, and joins the same Oracle Tuxedo 11gR1 domain.
- In the UBBCONFIG file, the value of the AUTHSVC keyword in the \*RESOURCE section and the SVCNM keyword in the \*SERVICES section are not allowed to use long service names when Oracle Tuxedo 11gR1 or earlier software coexists in the same Oracle Tuxedo 12cR2 domain. If long service names are used, the earlier Oracle Tuxedo release site will not boot.
- Oracle Tuxedo 12c Release 2 (12.1.3) application servers with one or more services using long service names will not boot when Oracle Tuxedo 11gR1 or earlier coexists in same the Oracle Tuxedo domain.
- Any dynamic addition of services with long names fail when Oracle Tuxedo 11gR1 or earlier coexists in the same Oracle Tuxedo domain.
- Oracle Tuxedo 11gR1 or earlier software is not allowed join a domain if one or more services with long names has already booted in the current Oracle Tuxedo domain.
- You cannot configure a long local resource and remote service name for TDOMAIN gateway servers that are deployed in running Oracle Tuxedo 11gR1 or earlier in the same Oracle Tuxedo domain.
- If users specify XPath in ROUTING section, the earlier Oracle Tuxedo release sites will not be booted.
- In MP mode, when the following features are enabled, the slave node fails to be booted up if its Tuxedo version is lower than Tuxedo 12cR2 or if the slave node is running on the platforms that those features do not support.
  - XA Affinity
  - Common XID
  - Single Group Multiple Branches (SGMB)
  - FAN Integration

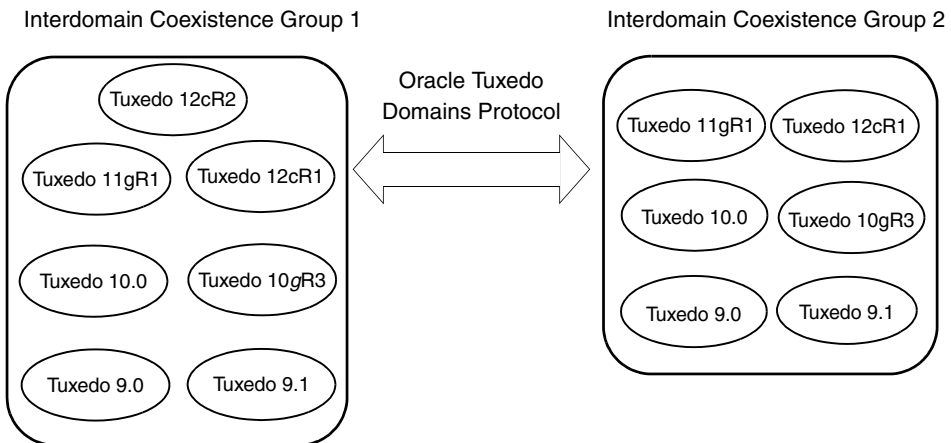
- Direct Cross Domain Communication Leveraging RDMA

**Note:** When the option `EECS` in `OPTIONS` of `UBBCONFIG *RESOURCES` section is specified, the first four features (XA Affinity, Common XID, Single Group Multiple Branches, and FAN Integration) are enabled by default.

## Interdomain Interoperability

Message exchange and protocol compatibility exist in each of the following *interdomain* scenarios as shown in [Figure 1-2](#):

**Figure 1-2 Interdomain Scenario 1**



In this scenario, an Oracle Tuxedo domain (TDomain) gateway process running on a machine in one domain communicates over a network connection with a TDomain gateway process running on a machine in another domain. [Table 1-1](#) lists the supported TDomain gateway processes.



**Table 1-1 Communicating TDomain Gateway Processes**

| A TDomain process in any of these releases . . . | Can communicate with a TDomain process in any of these releases . . . |
|--|---|
| Oracle Tuxedo 12c Release 2 (12.1.3)             | Oracle Tuxedo 12c Release 2 (12.1.3)                                  |
| Oracle Tuxedo 12c Release 1 (12.1.1)             | Oracle Tuxedo 12c Release 1 (12.1.1)                                  |
| Oracle Tuxedo 11gR1 Release 1                    | Oracle Tuxedo 11g Release 1 (11.1.1.3.0)                              |
| Oracle Tuxedo 10g Release 3 (10.3)               | Oracle Tuxedo 10g Release 3 (10.3)                                    |
| Oracle Tuxedo 10.0                               | Oracle Tuxedo 10.0  |
| Oracle Tuxedo 9.1                                | Oracle Tuxedo 9.1   |
| Oracle Tuxedo 9.0                                | Oracle Tuxedo 9.0   |

Connection  
Matrix

Interdomain capabilities available through a pair of communicating TDomain processes are limited to the capabilities available to the TDomain process running in the earlier Oracle release.

In all of these scenarios, administration, transaction context propagation, and security context propagation between domains is fully supported.

Oracle Tuxedo 12c Release 2 (12.1.3) supports interdomain interoperability with Tuxedo 12cR1, 11gR1, 10gR3, 10.0, 9.1, 9.0 CORBA domains. This capability includes the ability to advertise CORBA C++ factories across domain boundaries.

## Client-Server Interoperability

To support customer migration, the following client-server interoperability shown in [Table 1-2](#) is supported for Oracle Tuxedo 12c Release 2 (12.1.3).

**Table 1-2 Client-Server Interoperability**

| <b>This component . . .</b>   | <b>Can interoperate with . . .</b>  |
|---|---|
| Oracle Tuxedo 12c Release 2 (12.1.3) ATMI server                                  | <ul style="list-style-type: none"><li>• ATMI clients running in Oracle Tuxedo 6.5 (workstation client), 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1</li><li>• Jolt clients running in Jolt 9.0, 9.1, 10.0, 10gR3, and 11gR1, and 12cR1 (via Jolt server 12c Release 2 (12.1.3))</li></ul> |
| Oracle Tuxedo 11g Release 1 (11.1.1.1.0, 11.1.1.2.0, and 11.1.1.3.0) ATMI server  | <ul style="list-style-type: none"><li>• ATMI clients running in Oracle Tuxedo 9.0, 9.1, 10.0 and 10gR3</li><li>• Jolt clients running in Jolt 9.0, 9.1, 10.0 and 10gR3 (via Jolt server 11g Release 1)</li></ul>  |
| Oracle Tuxedo 10gR3 ATMI server   | <ul style="list-style-type: none"><li>• ATMI clients running in Oracle Tuxedo 9.0, 9.1 and 10.0</li><li>• Jolt clients running in Jolt 9.0, 9.1, and 10.0 (via Jolt server 10g release 3 (10.3))</li></ul>  |
| Oracle Tuxedo 10.0 ATMI server  | <ul style="list-style-type: none"><li>• ATMI clients running in Oracle Tuxedo 9.0, and 9.1</li><li>• Jolt clients running in Jolt 9.0 and 9.1 (via Jolt server 10.0)</li></ul>  |
| Oracle Tuxedo 9.1 ATMI server   | <ul style="list-style-type: none"><li>• ATMI clients running in Tuxedo 9.0</li><li>• Jolt clients running in Jolt 9.0 (via Jolt server 9.1)</li></ul>   |
| Oracle Tuxedo 12c Release 2 (12.1.3) CORBA server                                 | <ul style="list-style-type: none"><li>• CORBA clients running in Oracle Tuxedo 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1</li></ul>  |
| Oracle Tuxedo 11g Release 1 (11.1.1.1.0, 11.1.1.2.0, and 11.1.1.3.0) CORBA server | <ul style="list-style-type: none"><li>• CORBA clients running in Oracle Tuxedo 9.0, 9.1, 10.0 and 10gR3</li></ul>   |
| Oracle Tuxedo 10g Release 3 (10.3) CORBA server                                   | <ul style="list-style-type: none"><li>• CORBA clients running in Oracle Tuxedo 9.0, 9.1, and 10.0</li></ul>   |
| Oracle Tuxedo 10.0 CORBA server   | <ul style="list-style-type: none"><li>• CORBA clients running in Oracle Tuxedo 9.0 and 9.1</li></ul>  |
| Oracle Tuxedo 9.1 CORBA server  | <ul style="list-style-type: none"><li>• CORBA clients running in Oracle Tuxedo 9.0</li></ul>  |

**Table 1-2 Client-Server Interoperability**

| <b>This component . . .</b>   | <b>Can interoperate with . . .</b>   |
|---|--|
| Oracle Tuxedo 12c Release 2 (12.1.3) ATMI client                                  | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1</li> </ul>  |
| Oracle Tuxedo 11g Release 1 (11.1.1.1.0, 11.1.1.2.0, and 11.1.1.3.0) ATMI client  | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1, 10.0 and 10gR3</li> </ul>                 |
| Oracle Tuxedo 10gR3 ATMI client   | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1, and 10.0</li> </ul>                       |
| Oracle Tuxedo 10.0 ATMI client  | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, and 9.1</li> </ul>                             |
| Oracle Tuxedo 9.1 ATMI client   | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0</li> </ul>                                      |
| Oracle Tuxedo 12c Release 2 (12.1.3) .NET client                                  | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1</li> </ul>  |
| Oracle Tuxedo 11g Release 1(11.1.1.1.0 and 11.1.1.2.0) .NET client                | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1, 10.0 and 10gR3</li> </ul>                 |
| Oracle Tuxedo 10g Release 3(10.3) .NET client                                     | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0, 9.1 and 10.0</li> </ul>                        |
| Oracle Tuxedo 10.0 .NET client  | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0 and 9.1</li> </ul>                              |
| Oracle Tuxedo 9.1 .NET client   | <ul style="list-style-type: none"> <li>• ATMI servers running in Oracle Tuxedo 9.0</li> </ul>                                      |
| Oracle Tuxedo 12c Release 2 (12.1.3) CORBA client                                 | <ul style="list-style-type: none"> <li>• CORBA servers running in Oracle Tuxedo 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1</li> </ul> |
| Oracle Tuxedo 11g Release 1 (11.1.1.1.0, 11.1.1.2.0, and 11.1.1.3.0) CORBA client | <ul style="list-style-type: none"> <li>• CORBA servers running in Oracle Tuxedo 9.0, 9.1 and 10.0, and 10gR3</li> </ul>            |
| Oracle Tuxedo 10gR3 CORBA client  | <ul style="list-style-type: none"> <li>• CORBA servers running in Oracle Tuxedo 9.0, 9.1 and 10.0</li> </ul>                       |
| Oracle Tuxedo 10.0 CORBA client   | <ul style="list-style-type: none"> <li>• CORBA servers running in Oracle Tuxedo 9.0, and 9.1</li> </ul>                            |

**Table 1-2 Client-Server Interoperability**

| <b>This component . . .</b>   | <b>Can interoperate with . . .</b>   |
|---|--|
| Oracle Tuxedo 9.1 CORBA client  | <ul style="list-style-type: none"><li>• CORBA servers running in Oracle Tuxedo 9.0</li></ul>   |
| Oracle Jolt 12c Release 2 (12.1.3) client                                 | <ul style="list-style-type: none"><li>• ATMI servers running in Oracle Tuxedo 9.0 (via Jolt server 9.0)</li><li>• ATMI servers running in Oracle Tuxedo 9.1 (via Jolt server 9.1)</li><li>• ATMI servers running in Oracle Tuxedo 10.0 (via Jolt server 10.0)</li><li>• ATMI servers running in Oracle Tuxedo 10gR3 (via Jolt server 10gR3)</li><li>• ATMI servers running in Oracle Tuxedo 11gR1 (via Jolt server 11gR1)</li><li>• ATMI servers running in Oracle Tuxedo 12cR1 (via Jolt server 1cR1)</li></ul> |
| Oracle Jolt 11g Release 1 (11.1.1.1.0, 11.1.1.2.0, and 11.1.1.3.0) client | <ul style="list-style-type: none"><li>• ATMI servers running in Oracle Tuxedo 9.0 (via Jolt server 9.0)</li><li>• ATMI servers running in Oracle Tuxedo 9.1 (via Jolt server 9.1)</li><li>• ATMI servers running in Oracle Tuxedo 10.0 (via Jolt server 10.0)</li><li>• ATMI servers running in Oracle Tuxedo 10gR3 (via Jolt server 10gR3)</li></ul>  |
| Oracle Jolt 10.0 client   | <ul style="list-style-type: none"><li>• ATMI servers running in Oracle Tuxedo 9.0 (via Jolt server 9.0)</li><li>• ATMI servers running in Oracle Tuxedo 9.1 (via Jolt server 9.1)</li></ul>  |
| Oracle Jolt 9.1 client  | <ul style="list-style-type: none"><li>• ATMI servers running in Oracle Tuxedo 9.0 (via Jolt server 9.0)</li></ul>  |

The capabilities available to a client for a particular client-server pair depend on the release of both the application client and the server application. For example, if you have an Oracle Tuxedo 12c Release 2 (12.1.3) ATMI client interoperating with an Oracle Tuxedo 9.1 server application, only Oracle Tuxedo 9.1 functionality is available to the client.

## Client/Server Affinity Interoperability

Oracle Tuxedo Client/Server Affinity interoperability is supported as follows:

- The Client/Server Affinity feature *does not* work with Oracle Tuxedo 10gR3 or earlier native clients; however, it *does* work with older /WS or Jolt client versions connecting to Oracle Tuxedo 12cR2.

- MP mode

If Client/Server Affinity is configured in the UBBCONFIG file and there are slave nodes that use Oracle Tuxedo 10gR3 or earlier, the master node and *only* slave nodes installed with Oracle Tuxedo 12cR2 or later are affected.

- Domain mode

To use Client/Server Affinity in a multiple-domain session, Oracle Tuxedo 12cR2 or later must be installed on all domains involved in the session.

## Feature-Specific Interoperability

The Oracle Tuxedo 12c Release 2 (12.1.3) feature-specific interoperability is supported as follows:

- XML Based Data Dependent Routing (DDR)

To use the XML based DDR, all the machines in one domain must use Oracle Tuxedo 12c Release 2 (12.1.3). If any one of them uses a lower version, tmboot prints an error message "Invalid release".

- Exalogic Features

- Oracle Tuxedo 12c Release 2 (12.1.3) can interoperate with Oracle Tuxedo 11gR1 (11.1.1.3.0) or above if any Exalogic feature that is introduced prior to Oracle Tuxedo 12c Release 2 (12.1.3) is enabled in the UBBCONFIG file.
- Oracle Tuxedo 12c Release 2 (12.1.3) cannot interoperate with Oracle Tuxedo 11gR1 (11.1.1.2.0) or earlier if any Exalogic feature is enabled in the UBBCONFIG file.
- Oracle Tuxedo 12c Release 2 (12.1.3) cannot interoperate with previous releases if any Exalogic feature that is introduced in Oracle Tuxedo 12c Release 2 (12.1.3) is enabled in the UBBCONFIG file.
- Oracle Tuxedo 12c Release 2 (12.1.3) cannot interoperate with Oracle Tuxedo 11gR1 (11.1.1.3.0) if Read-Only Optimization for XA feature is enabled.

For more information, see [Oracle Tuxedo/Oracle Exalogic Users Guide](#).

- Millisecond Granularity for Timeouts

If an MP deployment environment is configured with millisecond `SCANUNIT`, all the nodes in this domain should use Oracle Tuxedo 12c Release 2 (12.1.3).

- ECID Propagation

End-to-end ECID propagation can only be guaranteed for machines/domains running Oracle Tuxedo 12c Release 2 (12.1.3).

- Cross Domain Event Broker

This feature is supported only when both GWT and EvtBroker are running Oracle Tuxedo 12c Release 2 (12.1.3).

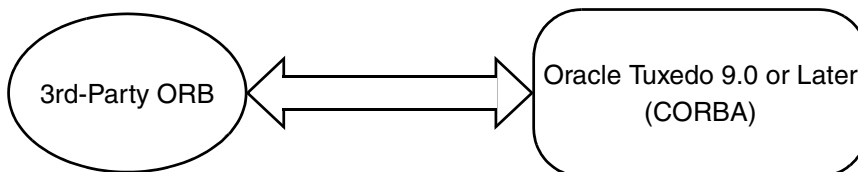
## Interoperability with Third-Party ORBs

Bootstrapping an Oracle Tuxedo CORBA domain establishes communication between a CORBA application client and the domain. Two bootstrapping mechanisms are available: (1) the Oracle mechanism using the Bootstrap object and (2) the CORBA Interoperable Naming Service (INS) bootstrapping mechanism specified by the OMG.

Support for INS was added in Oracle Tuxedo release 8.0. With the addition of INS, third-party ORBs that use INS are able to interoperate with the Oracle Tuxedo CORBA server ORB.

[Figure 1-3](#) shows the interoperability with third-party ORBs.

**Figure 1-3 Interoperability with Third-Party ORBs**



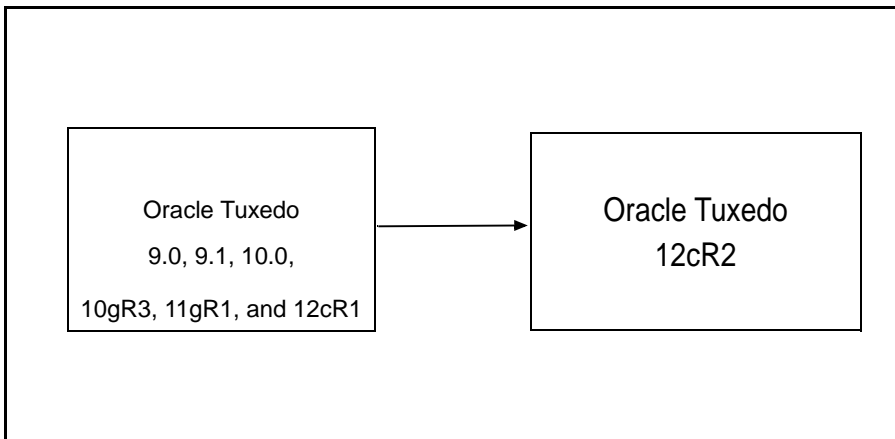
**Note:** The Oracle Tuxedo CORBA client environmental objects continue to be supported in the current Oracle Tuxedo release, just as they were supported in Oracle Tuxedo 9.0, 9.1, 10.0, 10gR3, 11gR1, and 12cR1.

A CORBA application client uses the Oracle Tuxedo Bootstrap object or the INS bootstrapping mechanism to obtain references to the objects in an Oracle Tuxedo CORBA domain. Oracle client ORBs use the Oracle mechanism, and third-party client ORBs use the CORBA INS mechanism. For more information about bootstrapping an Oracle Tuxedo domain, see [Oracle Tuxedo CORBA Programming Reference](#).

## Product Upgrades

Figure 1-4 shows the existing Oracle Tuxedo and Oracle WebLogic Enterprise products that can be upgraded to Oracle Tuxedo 12c Release 2 (12.1.3).

**Figure 1-4 Upgrade Paths**



When you shut down the domain (application) targeted for the upgrade, you must shut down the domain and perform a *simple upgrade*. If you cannot shut down the domain targeted for the upgrade, perform a *hot upgrade*, that is, add the Oracle Tuxedo 12c Release 2 (12.1.3) system software to the existing Oracle Tuxedo or Oracle WebLogic Enterprise domain without shutting down the domain.

For instructions on performing a simple upgrade or a hot upgrade, see "[Upgrading the Oracle Tuxedo System to 12c Release 2 \(12.1.3\)](#)" in *Installing the Oracle Tuxedo System*.

# Upward Application Compatibility

Applications developed with Oracle Tuxedo 9.0, 9.1, 10.0 , 10gR3, 11gR1, and 12cR1 applications must be recompiled to run on Oracle Tuxedo 12c Release 2 (12.1.3).

**Notes:** For XML-related applications, you must conform to Xerces C++ 2.5 interface requirements.

On Windows platform, the binary must be relinked if FML-related functions are used.



# Interoperability with Oracle WebLogic Server

The following sections present interoperability capabilities between Oracle Tuxedo and Oracle WebLogic Server:

- [Interoperability Software Components](#)
- [Interoperability Programming Interfaces](#)
- [JSL/JSH-Jolt Unidirectional Connectivity](#)
- [TDomain-WTC Bidirectional Connectivity](#)
- [RMI-over-IIOP Client Direct Connectivity to an EJB](#)
- [Summary of Interoperability Capabilities](#)
- [Interoperability Sample Applications](#)

## Interoperability Software Components

Interoperability between Oracle Tuxedo and Oracle WebLogic Server is implemented as the following three sets of communicating software processes.

| Set | Tuxedo Component                             | Interoperability Direction | WebLogic Server Component                 | Interoperability   |
|-----|--|----------------------------|---|--|
| 1   | Jolt Server Listener/<br>Jolt Server Handler | ←                          | Oracle Jolt for Oracle<br>WebLogic Server | Enables WebLogic Server application servers to call Tuxedo ATMI services.  |
| 2   | TDomain gateway                              | ↔                          | WebLogic Tuxedo<br>Connector (WTC)        | <p>Enables WebLogic Server application servers to call Tuxedo ATMI services.</p> <p>Enables WebLogic Server application servers to call Tuxedo CORBA C++ objects.</p> <p>Enables Tuxedo ATMI clients or servers to call WebLogic Server application servers.</p> <p>Enables Tuxedo CORBA C++ clients or servers to call WebLogic Server application servers.</p> |

## Jolt Server Listener

A Jolt Server Listener (JSL) is a listening process, running on the Tuxedo server, that accepts connection requests from Jolt clients and assigns connections to a Jolt Server Handler also running on the Tuxedo server. It also manages the pool of Jolt Server Handler processes, starting them in response to load demands.

## Jolt Server Handler

A Jolt Server Handler (JSH) is a gateway process, running on the Tuxedo server, that handles communications between Jolt clients and the Tuxedo ATMI server application. A JSH process resides within the administrative domain of the application and is registered in the local Tuxedo bulletin board as a client.

Each JSH process can manage multiple Jolt clients. A JSH multiplexes all requests and replies with a particular Jolt client over a single connection.

## Oracle Jolt for WebLogic Server

Oracle Jolt is a Java-based client API that manages requests to Tuxedo services via a Jolt Service Listener (JSL) running on the Tuxedo server. The Jolt API is embedded within the WebLogic API and is accessible from a servlet or any other Oracle WebLogic application.

## IIOP Listener

An IIOP Listener (ISL) is a listening process, running on the Tuxedo server, that accepts connection requests from CORBA clients and assigns connections to an IIOP Handler also running on the Tuxedo server. It also manages the pool of IIOP Handler processes, starting them in response to load demands.

## IIOP Handler

An IIOP Handler (ISH) is a gateway process, running on the Tuxedo server, that handles IIOP communications between CORBA clients and the Tuxedo server application. An ISH process resides within the administrative domain of the application and is registered in the local Oracle Tuxedo bulletin board as a client.

Each ISH process can manage multiple CORBA clients. An ISH multiplexes all requests and replies with a particular CORBA client over a single connection.

## TDomain Gateway

The TDomain gateway, implemented by the `GWTDOMAIN` server process, provides interoperability between two or more Oracle Tuxedo domains through a specially designed transaction processing protocol that flows over network protocol TCP/IP. Working with the WebLogic Tuxedo Connector gateway, the Oracle Tuxedo TDomain gateway can also provide interoperability between Tuxedo domains and WebLogic Server applications.

## WebLogic Tuxedo Connector

The WebLogic Tuxedo Connector (WTC) enables bi-directional interoperability between the WebLogic Server and Tuxedo ATMI and CORBA environments. The WTC gateway supports the TDomain gateway protocol.

Oracle Tuxedo 12c Release 2 (12.1.3) supports the following WebLogic/WTC versions:

- WLS 12cR2 (12.2.1)

- WLS 12cR1 (12.1.3)
- WLS 12cR1 (12.1.2)
- WLS 12cR1 (12.1.1)
- WLS 11gR1 (10.3.x) (x>0)
- WLS 10gR3 (10.3.0)
- WLS 10.0
- WLS 9.2
- WLS 9.1
- WLS 9.0

For a complete list of supported WLS versions through all Oracle Tuxedo release, see [Oracle Tuxedo Certified Platform Tables](#).

## Interoperability Programming Interfaces

Interoperability between Oracle Tuxedo and Oracle WebLogic Server is achieved using the following application programming interfaces:

- Application-to-Transaction Monitor Interface (ATMI)
- Java Application-to-Transaction Monitor Interface (JATMI)
- Jolt API
- Remote Method Invocation (RMI)
- Remote Method Invocation (RMI) over Internet Inter-ORB Protocol (IIOP) (RMI-over-IIOP)
- CORBA Java

### ATMI Interface

ATMI provides an interface for communications, transactions, and data-buffer management that works in all ATMI environments supported by the Oracle Tuxedo system. ATMI is described in [Introducing Oracle Tuxedo ATMI](#).

## JATMI Interface

JATMI is the Oracle WebLogic Server Java implementation of the Oracle Tuxedo ATMI. It allows WebLogic Server application servers to access Tuxedo ATMI services. JATMI is described in *WebLogic Tuxedo Connector Programmer's Guide* at:

[http://download.oracle.com/docs/cd/E12840\\_01/wls/docs103/wtc\\_atmi/index.html](http://download.oracle.com/docs/cd/E12840_01/wls/docs103/wtc_atmi/index.html).

## Jolt Interface

Oracle Jolt for Oracle WebLogic Server is a Java-based client API that manages requests to Oracle Tuxedo services running on the Tuxedo server. The Jolt API is embedded within the WebLogic API and is accessible from a servlet or any other Oracle WebLogic application. Jolt API is described in *Using Oracle Jolt with Oracle WebLogic Server*.

## RMI Interface

Remote Method Invocation is a Java-based API set and protocol that allows an object running in one Java virtual machine to invoke methods on an object running in a different Java virtual machine. RMI specifies how distributed Java applications should operate over multiple Java virtual machines. RMI's native protocol is called Java Remote Method Protocol (JRMP).

For more information about RMI, see *Programming WebLogic RMI* at:

[http://download.oracle.com/docs/cd/E12840\\_01/wls/docs103/rmi/index.html](http://download.oracle.com/docs/cd/E12840_01/wls/docs103/rmi/index.html).

## RMI-over-IIOP Interface

RMI-over-IIOP provides interoperability with CORBA objects implemented in any language if all the remote interfaces are originally defined as RMI interfaces. RMI-over-IIOP is also known as RMI-on-IIOP, RMI/IIOP, or RMI-IIOP. The term RMI-over-IIOP is used in the discussions that follow.

With RMI and CORBA, programmers must decide between RMI, with its easy programming features, and CORBA, with its broad interoperability. IBM and Sun's JavaSoft, with the cooperation of the Object Management Group (OMG), jointly developed RMI-over-IIOP to solve this dilemma. JavaSoft includes RMI-over-IIOP in its Java Development Kit (JDK).

With RMI-over-IIOP, Java programmers can create applications in RMI that include CORBA connections. And with CORBA 2.3 support for Objects-by-Value, CORBA programmers can create applications in CORBA that include EJB connections.

**Note:** For information on Objects-by-Value and supported value types in Oracle Tuxedo CORBA, see “Mapping of OMG IDL Statements to C++” in *Oracle Tuxedo CORBA Programming Reference*.

With RMI-over-IIOP and CORBA support for Objects-by-Value, the following client-server interfaces are possible:

- RMI client → RMI-over-IIOP server
- CORBA client → RMI-over-IIOP server
- RMI-over-IIOP client → RMI server
- RMI-over-IIOP client → CORBA server
- RMI-over-IIOP client → RMI-over-IIOP server

**Note:** For the “RMI-over-IIOP client → CORBA server” interface, an RMI-over-IIOP client cannot necessarily access all existing CORBA objects because the semantics of CORBA objects defined in IDL are a superset of those of RMI-over-IIOP objects. Thus, an existing CORBA object’s IDL cannot always be mapped into an RMI-over-IIOP Java interface.

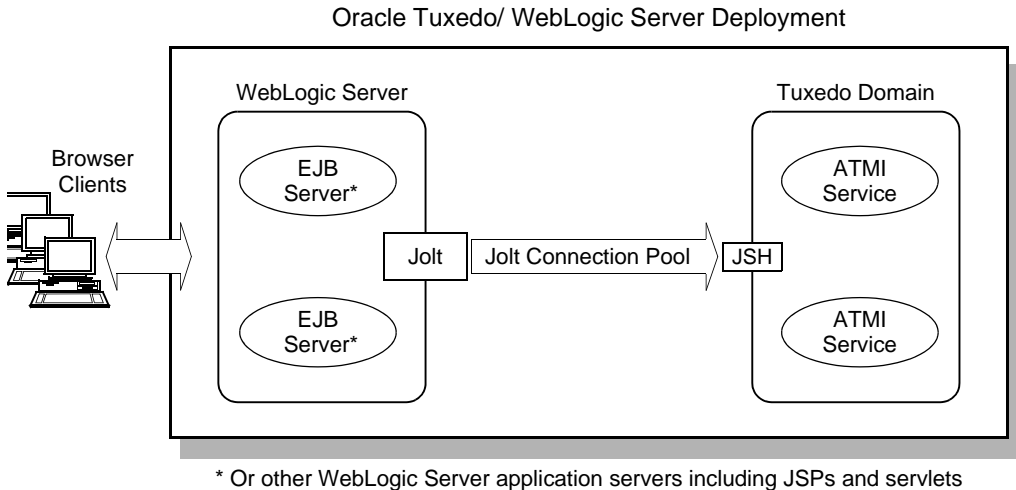
A server binary (a class file) created using RMI-over-IIOP APIs can be exported as JRMP (RMI native protocol), IIOP, or both. Exporting an RMI-over-IIOP object to both JRMP and IIOP simultaneously is called *dual export*.

For more information about RMI-over-IIOP, see *Programming WebLogic RMI over IIOP* at: [http://download.oracle.com/docs/cd/E12840\\_01/wls/docs103/rmi/iiop\\_basic.html](http://download.oracle.com/docs/cd/E12840_01/wls/docs103/rmi/iiop_basic.html).

## JSL/JSH-Jolt Unidirectional Connectivity

Oracle Jolt for WebLogic Server provides unidirectional connectivity from Oracle WebLogic Server applications to Oracle Tuxedo 9.1 or later ATMI services. With Oracle Jolt for WebLogic Server, an application administrator can enable Tuxedo services for the Web, using the WebLogic Server as the front-end HTTP and application server. [Figure 2-1](#) shows how this connectivity is implemented.

**Figure 2-1 WebLogic Server to Oracle Tuxedo Connectivity Using Jolt**



Oracle Jolt is a Java-based client API that manages requests to Oracle Tuxedo services using a Jolt Server Listener running on the Tuxedo server. The Jolt API is accessible to an EJB, a JSP, a servlet, a Java HTML (JHTML), or other Oracle WebLogic application server.

## Jolt Connection Pooling

WebLogic Server uses a variation of the Jolt session pool called a *servlet session pool*, commonly referred to as simply a *Jolt connection pool*. The Jolt connection pool provides extra functionality that is convenient for use inside an HTTP servlet.

Jolt connection pooling allows WebLogic Server application servers to invoke Oracle Tuxedo services in an Oracle Tuxedo application. The pooling feature supports connection pool reset in the event of connection pool failure, which eliminates the need to restart WebLogic Server if the connection pool requires a restart.

## Jolt Wire-Level Security

The following wire-level security is supported on the network connection between the Java Server Handler and WebLogic Server: 40-bit, 56-bit, or 128-bit LLE. LLE, for Link-Level Encryption, is an Oracle Tuxedo-based protocol for establishing data privacy over network links.

## Jolt Transaction and Security Context Propagation

Jolt supports transaction demarcation, propagation of security, and connection reset. Jolt provides a mechanism for propagating the security context established in WebLogic Server to the Oracle Tuxedo application.

User credentials authenticated by WebLogic Server are mapped to the appropriate security interfaces/protocols. An incoming request does not require re-authentication before invoking Tuxedo ATMI services.

## Jolt Documentation

For complete information on using Oracle Jolt with WebLogic Server, see [Using Oracle Jolt with Oracle WebLogic Server](#). This document explains the operation of Oracle Jolt for WebLogic Server, and describes how to use, configure, and integrate Oracle Jolt, Oracle Tuxedo ATMI, and Oracle WebLogic Server.

## TDomain-WTC Bidirectional Connectivity

The TDomain and WTC gateways provide bidirectional connectivity *between* ATMI services/ CORBA objects deployed in an Oracle Tuxedo application *and* EJB objects deployed in a WebLogic Server application. Together, the gateways allow the following interoperability for an Oracle Tuxedo/ WebLogic Server deployment:

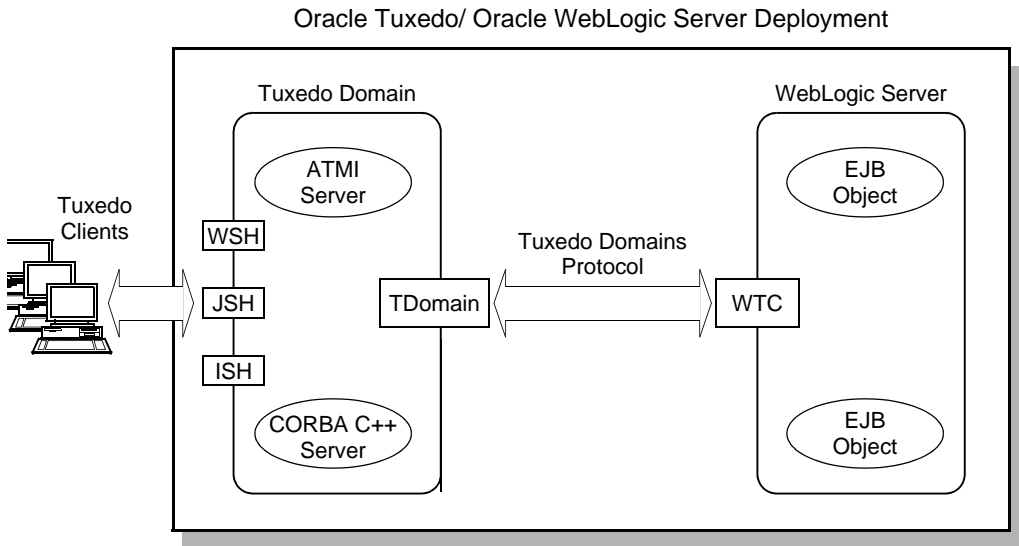
- Allow Tuxedo ATMI clients, and Tuxedo ATMI servers acting as clients, to access WebLogic Server EJB servers via the ATMI interface.
- Allow Tuxedo CORBA clients, and Tuxedo CORBA servers acting as clients, to access WebLogic Server EJB servers via RMI-over-IIOP.
- Allow WebLogic Server application servers (EJBs, JSPs, Java servlets) acting as clients to access:
  - Tuxedo ATMI servers via JATMI
  - Tuxedo CORBA servers via CORBA Java or RMI-over-IIOP

## Oracle Tuxedo to Oracle WebLogic Server Connectivity

Oracle Tuxedo application clients and servers can invoke EJB objects in a WebLogic Server application, which in turn can invoke other EJB objects, JSPs, or Java servlets. [Figure 2-2](#) shows how this connectivity is implemented.



Figure 2-2 Tuxedo to WebLogic Server Connectivity



**Note:** Oracle Tuxedo clients include ATMI clients, Jolt clients, and CORBA C++ clients. For a high-level view of Tuxedo clients, see “Client and Server Components” in [Oracle Tuxedo Product Overview](#).

The TDomain gateway not only enables Tuxedo domains to share services with other Oracle Tuxedo domains, but it enables Tuxedo domains to share services with WebLogic Server 6.1or later installations through the WTC gateway. The WTC gateway supports the TDomain gateway protocol.

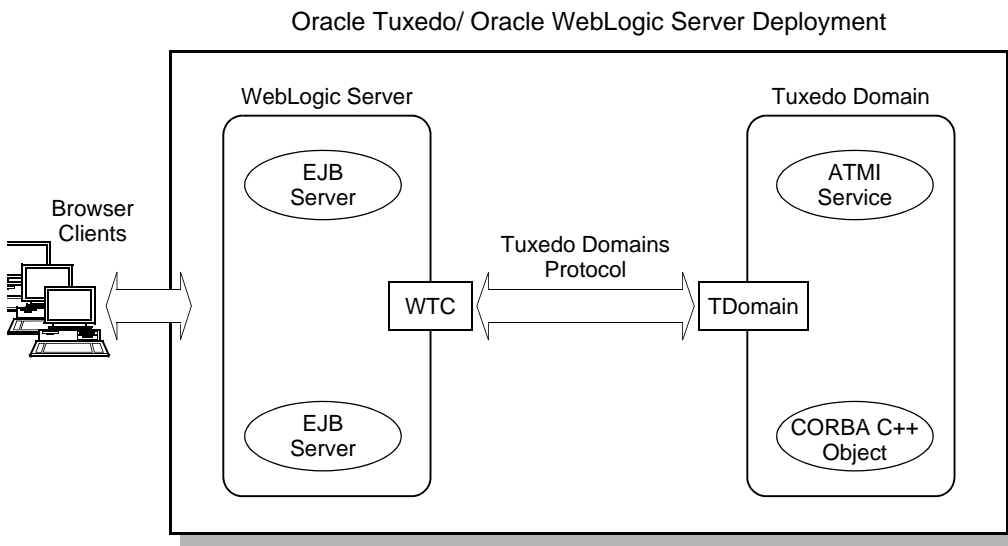
The gateways allow Tuxedo ATMI clients, and Tuxedo ATMI servers acting as clients, to access WebLogic Server EJB objects. The TDomain gateway delivers the ATMI client request to the WTC gateway, and the WTC gateway converts the request to an RMI call to access the appropriate EJB object.

Similarly, the gateways allow Oracle Tuxedo CORBA clients, and Tuxedo CORBA servers acting as clients, to access WebLogic Server EJB objects. The TDomain gateway delivers the CORBA RMI-over-IIOP client request to the WTC gateway, and the WTC gateway forwards the request to the appropriate EJB object.

## Oracle WebLogic Server to Oracle Tuxedo Connectivity

EJB application servers in a WebLogic Server application can invoke services and CORBA objects in an Oracle Tuxedo application using the WTC and TDomain gateways. [Figure 2-3](#) shows how this connectivity is implemented.

**Figure 2-3 WebLogic Server to Tuxedo Connectivity Using WTC**



The WTC and TDomain gateways allow WebLogic Server EJBs, JSPs, or Java servlets acting as clients to access Oracle Tuxedo services. The WTC gateway converts the EJB/JSP/servlet JATMI request to an ATMI request, and the TDomain gateway delivers the ATMI request to an Oracle Tuxedo ATMI server offering the requested service.

Similarly, the gateways allow WebLogic Server EJBs, JSPs, or Java servlets acting as clients to access Oracle Tuxedo CORBA objects. The WTC gateway inserts the EJB/JSP/servlet CORBA Java or RMI-over-IIOP request inside of an Oracle Tuxedo GIOP (TGIOP) request message, and the TDomain gateway delivers the TGIOP request to an Oracle Tuxedo CORBA server offering the requested object.

## TDomain-WTC Wire-Level Security

The following wire-level security is supported on the network connection between the TDomain and WTC gateways: 40-bit, 56-bit, or 128-bit LLE. LLE, for Link-Level Encryption, is a Tuxedo-based protocol for establishing data privacy over network links.

## TDomain-WTC Transaction and Security Context Propagation

Bidirectional propagation of transaction context and security context between application clients and servers in an Oracle Tuxedo/ WebLogic Server deployment is fully supported through the TDomain and WTC gateways.

## TDomain and WTC Documentation

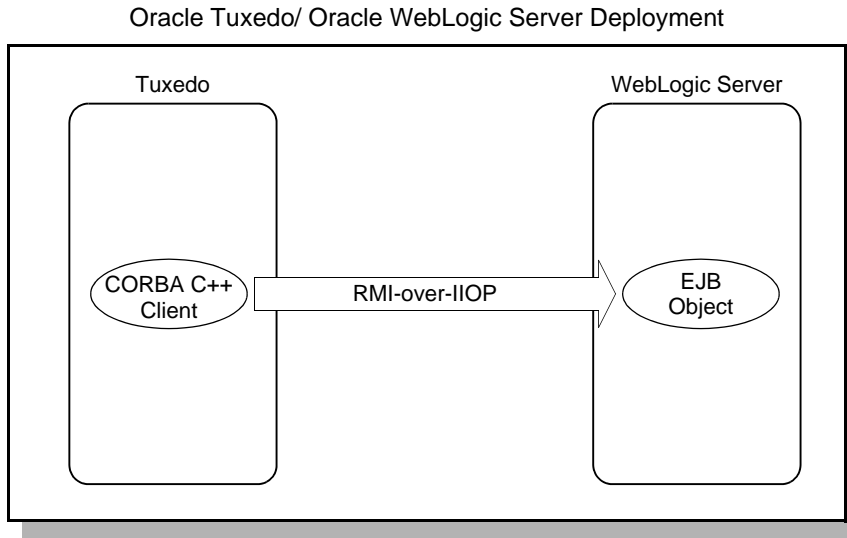
For details about the Oracle Tuxedo Domains gateway, see [Using the Oracle Tuxedo Domains Component](#). For details about the WTC gateway, see [WebLogic Tuxedo Connector at:](#)

[http://download.oracle.com/docs/cd/E12840\\_01/wls/docs103/wtc.html](http://download.oracle.com/docs/cd/E12840_01/wls/docs103/wtc.html).

## RMI-over-IIOP Client Direct Connectivity to an EJB

In addition to using the TDomain and WTC gateways to achieve connectivity from Oracle Tuxedo CORBA to Oracle WebLogic Server, Tuxedo CORBA C++ clients or servers can call WebLogic Server application servers *directly* using RMI-over-IIOP and CORBA Interface Definition Language (IDL) interfaces. [Figure 2-4](#) demonstrates this type of connectivity.

**Figure 2-4 Direct EJB Connectivity Using RMI-over-IIOP and IDL Interfaces**



For a sample application describing how a CORBA C++ client application developed in Oracle Tuxedo can directly interact with an EJB in WebLogic Server, see [Connectivity Between an Oracle Tuxedo CORBA Client and an EJB in WebLogic Server](#).

## Summary of Interoperability Capabilities

Table 2-1 summarizes the interoperability capabilities for an Oracle Tuxedo/ WebLogic Server deployment.

**Table 2-1 Oracle Tuxedo/ WebLogic Server Interoperability Capabilities (Sheet 1 of 2)**

| This component . . .   | Can call a . . .           | Through . . .          |
|------------------------|----------------------------|------------------------|
| Tuxedo ATMI client *   | WebLogic Server EJB object | WSH ** → TDomain → WTC |
| Tuxedo Jolt client *** | WebLogic Server EJB object | JSH → TDomain → WTC    |

\* A native Tuxedo ATMI or CORBA C++ client does not use Tuxedo handler gateway processes (WSH, ISH).  
 \*\* WSH stands for Workstation Handler.  
 \*\*\* The Tuxedo Jolt client connection to a WebLogic Server EJB object has not been tested.

**Table 2-1 Oracle Tuxedo/ WebLogic Server Interoperability Capabilities (Sheet 2 of 2)**

| This component . . .                    | Can call a . . .           | Through . . .  |
|---|----------------------------|--|
| Tuxedo CORBA C++ client *               | WebLogic Server EJB object | ISH → TDomain → WTC <i>or</i><br>RMI-over-IIOP client direct<br>connectivity to an EJB |
| Tuxedo ATMI server                      | WebLogic Server EJB object | TDomain → WTC  |
| Tuxedo CORBA C++ server                 | WebLogic Server EJB object | TDomain → WTC <i>or</i> RMI-over-IIOP<br>client direct connectivity to an EJB          |
| WebLogic Server EJB, JSP, or<br>servlet | Tuxedo ATMI service        | WTC → TDomain <i>or</i> Jolt for<br>WebLogic Server → JSH                              |
| WebLogic Server EJB, JSP, or<br>servlet | Tuxedo CORBA C++ object    | WTC → TDomain  |

\* A native Tuxedo ATMI or CORBA C++ client does not use Tuxedo handler gateway processes (WSH, ISH).  
\*\* WSH stands for Workstation Handler.  
\*\*\* The Tuxedo Jolt client connection to a WebLogic Server EJB object has not been tested.

## Interoperability Sample Applications

A large variety of interoperability sample applications are available at [Oracle Tuxedo Sample Code](#). The sample applications provide client and server programmers with information about the basic concepts of (1) combining Oracle Tuxedo ATMI services and WebLogic Server EJB objects in an application and (2) combining Oracle Tuxedo CORBA objects and WebLogic Server EJB objects in an application.

The ATMI examples show how to configure and set up WebLogic Server to work with Oracle Tuxedo ATMI servers and clients, using the underlying WTC technology.

The RMI-over-IIOP code examples show how to configure and set up WebLogic Server to work with Oracle Tuxedo CORBA servers and clients, using the underlying WTC technology.

For additional information on how to develop interoperability applications employing ATMI, JATMI, CORBA Java, or RMI-over-IIOP API, see [Administering WebLogic Tuxedo Connector for Oracle WebLogic Server](#) and [Developing Oracle WebLogic Tuxedo Connector Applications for Oracle WebLogic Server](#).

