



Platform Differences Guide

VERSION 2016.0.02

jade

Jade Software Corporation Limited cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages, or loss of profits. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Jade Software Corporation Limited.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Copyright © 2018 Jade Software Corporation Limited.

All rights reserved.

JADE is a trademark of Jade Software Corporation Limited. All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

For details about other licensing agreements for third-party products, you must read the JADE **ReadMe.txt** file.

Contents

Contents	iii
Before You Begin	iv
Who Should Read this Guide	iv
What's Included in this Guide	iv
Related Documentation	iv
Conventions	v
Chapter 1 Multiuser or Client-Server Issues	6
Server Node Method Restrictions	6
Application Execution Restrictions	6
Debugging and Inspecting Server Methods	7
Executable Programs Not Supported	7
Support of Classes and Methods Implemented in jadpmap Library	7
Class Availability	7
Chapter 2 Migrating from a JADE 6.3 Release	9
Overview	9
JadeLocal Intra-Machine Transport Type	10
Most Recently Used (MRU) List of Opened Database Paths	11
Separation of Read-Only and Read-Write Files	11
Service Management	11
Windows Security Model	11
Chapter 3 Miscellaneous Platform Differences	12
Expat XML Parser	12

Before You Begin

The *JADE Platform Differences Guide* is intended as a source of information when you are administering JADE systems or running deployed JADE applications.

Who Should Read this Guide

The main audience for the *JADE Platform Differences Guide* is expected to be system administrators.

What's Included in this Guide

The *JADE Platform Differences Guide* has three chapters.

Chapter 1	Covers multiuser and client-server issues
Chapter 2	Covers migrating from a JADE 6.3 release
Chapter 3	Covers miscellaneous platform differences

Related Documentation

Other documents that are referred to in this guide, or that may be helpful, are listed in the following table, with an indication of the JADE operation or tasks to which they relate.

Title	Related to...
JADE Database Administration Guide	Administering a JADE database
JADE Development Environment User's Guide	Using the JADE development environment to development JADE applications
JADE Encyclopaedia of Classes	System classes (Volumes 1 and 2), Window classes (Volume 3)
JADE Encyclopaedia of Primitive Types	Primitive types and global constants
JADE Initialization File Reference	Maintaining JADE initialization file parameter values
JADE Installation and Configuration Guide	Installing and configuring JADE
JADE Runtime Application Guide	Administering JADE deployed runtime applications
JADE Thin Client Guide	Administering JADE thin client environments
JADE Web Application Guide	Implementing, monitoring, and configuring Web applications

Conventions

The *JADE Platform Differences Guide* uses consistent typographic conventions throughout.

Convention	Description
Arrow bullet (➤)	Step-by-step procedures. You can complete procedural instructions by using either the mouse or the keyboard.
Bold	Items that must be typed exactly as shown. For example, if instructed to type foreach , type all the bold characters exactly as they are printed. File, class, primitive type, method, and property names, menu commands, and dialog controls are also shown in bold type, as well as literal values stored, tested for, and sent by JADE instructions.
<i>Italic</i>	Parameter values or placeholders for information that must be provided; for example, if instructed to enter <i>class-name</i> , type the actual name of the class instead of the word or words shown in italic type. Italic type also signals a new term. An explanation accompanies the italicized type. Document titles and status and error messages are also shown in italic type.
Blue text	Enables you to click anywhere on the cross-reference text (the cursor symbol changes from an open hand to a hand with the index finger extended) to take you straight to that topic. For example, click on the " Service Management " cross-reference to display that topic.
Bracket symbols ([])	Indicate optional items.
Vertical bar ()	Separates alternative items.
Monospaced font	Syntax, code examples, and error and status message text.
ALL CAPITALS	Directory names, commands, and acronyms.
SMALL CAPITALS	Keyboard keys.

Key combinations and key sequences appear as follows.

Convention	Description
KEY1+KEY2	Press and hold down the first key and then press the second key. For example, "press Shift+F2" means to press and hold down the Shift key and press the F2 key. Then release both keys.
KEY1,KEY2	Press and release the first key, then press and release the second key. For example, "press Alt+F,X" means to hold down the Alt key, press the F key, and then release both keys before pressing and releasing the X key.

This chapter covers the differences between JADE running on client and server nodes that occur because of multiuser versus single user issues.

- [Server Node Method Restrictions](#)
- [Application Execution Restrictions](#)
- [Debugging and Inspecting Server Methods](#)
- [Executable Programs Not Supported](#)
- [Support of Classes and Methods Implemented in jadpmap Library](#)

Server Node Method Restrictions

As the [Application::msgBox](#) method has no meaning in a server node environment, a message box is always executed on the client node or presentation client workstation, even if it is called from a server method.

Print facilities (the [Printer](#) class and [CMDPrint](#) class) are not supported on server nodes. An exception is raised if a printing operation (for example, calling the [Printer::setPrinter](#) method to set the output printer) is invoked from any of the following.

- A [serverExecution](#) method.
- A server application running under the [jadrap.exe](#) JADE Remote Node Access utility (because printing requires the [jade.exe](#) program).

An exception is raised or an incorrect result is returned if any of the following methods is invoked from a server method or from a server application.

Application::globalLockException	Collection::inspect	Collection::inspectModal
Exception::debug	Form::showModal	Form::popupMenu
Object::inspect	Object::inspectModal	ObjectArray::limitExceeded
Process::debug	RootSchemaApp::msgBox	

Application Execution Restrictions

The [Application](#) class [startApplication](#), [startApplicationWithParameter](#), and [startAppMethod](#) methods invoked from a server method or server application start only applications of type [ApplicationType_Non_GUI](#) or [ApplicationType_Non_GUI_Web](#). On a client node, they start all types of application.

Non-GUI applications (that is, those whose [Application](#) class [applicationType](#) property is set to [ApplicationType_Non_GUI](#) or [ApplicationType_Non_GUI_Web](#)) for standard (fat) clients are run as GUI applications that do not display forms; that is, [ApplicationType_GUI_No_Forms](#).

When running JADE in standard mode, non-GUI applications behave as follows.

- Form creation raises an exception unless the application is in exception state.
- The application displays an **Interrupt** button on the taskbar.

- Creation of a form is permitted while in exception state so that the **Debug** button on the exception dialog functions correctly.
- The JADE executable program (**jade.exe**) does not exit while non-GUI applications are running.

Debugging and Inspecting Server Methods

Server methods can be neither debugged nor inspected in a multiuser environment on a server node.

Executable Programs Not Supported

The **jdbutil** (JADE Database utility) executable GUI program is not supported in multiuser mode on a server node.

A batch version of this utility is available (that is, **jdbutilb**). For details, see "[Running the JADE Database Utility in Batch Mode](#)", in Chapter 1 of the *JADE Database Administration Guide*.

Support of Classes and Methods Implemented in jadpmap Library

You can access all GUI properties and methods (which are marked as **clientExecution** methods) from a server method except for anything that brings up a modal type dialog; that is, **app.msgBox**, **Form::showModal**, **Form::popupMenu**, and the common dialog class methods.

The other exceptions to this are the **app.doWindowEvents**, **app.checkPictureFile**, and **app.loadPicture** methods, which are executed relative to the server.

Caution Use of GUI methods and properties is *very* expensive in a server method.

A **clientExecution** method requires that all transient objects passed to the server are passed back with the client execution (and passed back to the server after the client execution is complete). Accessing GUI properties and methods within a server execution should therefore be done only in exceptional circumstances.

Class Availability

An exception is raised or an incorrect result is returned if any of the following methods is invoked classes listed in the following table are not available in a JADE application running on a server.

Class	Windows Single User	Windows Client	Windows Server
External database	Yes	Yes	Yes
GUI-classes	Yes	Yes	No
Locale classes	Yes	Yes	No
Common dialog	Yes	Yes	No
InternetPipe	Yes	Yes	Yes
NamedPipe	Yes	Yes	Yes
OleObject	Yes	Yes	Yes
Printer	Yes	Yes	No

Class	Windows Single User	Windows Client	Windows Server
Sound	Yes	Yes	Yes
WebSession	Yes	Yes	Yes

This chapter covers the JADE application differences between releases 6.3 and 7.

- [Overview](#)
- [JadeLocal Intra-Machine Transport Type](#)
- [Most Recently Used \(MRU\) List of Opened Database Paths](#)
- [Separation of Read-Only and Read-Write Files](#)
- [Service Management](#)
- [Security Model](#)

Overview

When JADE is installed under the **\Program Files** directory, it conforms to the expected Windows behavior.

The ability to return locations other than the JADE HOME directory for program and user data requests is dependent on where the JADE binaries have been installed.

If the JADE programs are not installed in the system-wide location for installed applications (in a folder in the **Program Files** directory), returning alternative addresses is not implemented. It is assumed that the programs installed in custom locations have sufficient security privileges to work in the style of earlier JADE 6.3 releases.

Although the alternative names and locations are driven by Windows requirements, the code is portable across Windows implementations.

The **Node** and **Process** class methods summarized in this section provide access to various locations in the file system hierarchy, because security restrictions can deny appropriate access to previously available directories or the directories are a unique location for each user when they were previously shared. (For details about the differences of these methods called on nodes and processes, see "[Directory Locations](#)", in Chapter 2 of the *JADE Installation and Configuration Guide*.)

Method	Returns a string containing the ...
<code>getJadeInstallDirectory</code>	Directory in which the executable of the current executing program is located.
<code>getJadeHomeDirectory</code>	Parent directory of the installation directory.
<code>getProgramDataDirectory</code>	Program data directory, which is dependent on the value of the ProgramDataDirectory parameter in the [JadeEnvironment] section of the JADE initialization file and the location of the installation directory. Files that should be placed under the returned location are entities that should be shared across multiple users of these binaries; for example, the jommsg.log file or shared dictionary spelling files that are updated.
<code>getUserDataDirectory</code>	User data directory, which is dependent on the value of the UserDataDirectory parameter in the [JadeEnvironment] section of the JADE initialization file and the location of the installation directory.

Method	Returns a string containing the ...
	Files that should not be placed under the returned location are entities that should be shared across multiple users of these binaries; for example, if a presentation client installation occurs on a Windows machine running Citrix or Terminal Services and all users run the same thin client binaries, any data created on the client file system should be stored under this location (that is, dictionaries for each user).
getJadeWorkDirectory	Directory in which the JADE work files are created and which is dependent on the value of the JadeWorkDirectory parameter in the [JadeEnvironment] section of the JADE initialization file.

Returned directory strings are normalized; that is, they have the virgule, or forward slash, (/) character as directory separators and they end in the virgule (/) character. Directories are created if they do not exist.

For details about the JADE HOME directory, see "[Specifying Parameters in the JADE Command Line](#)", in Chapter 2 of the *JADE Installation and Configuration Guide*.

For more details, see "[Directory Locations](#)", in Chapter 2 of the *JADE Installation and Configuration Guide, Volume 1* and *Volume 2* of the *JADE Encyclopaedia of Classes*, and "JADE Environment [JadeEnvironment] Section", in the *JADE Initialization File Reference*.

JadeLocal Intra-Machine Transport Type

The **JadeLocal** intra-machine transport between the JADE database and application servers and standard clients is implemented by the use of shared memory and global events and semaphores.

The **JadeLocal** transport type uses the following configuration values in the respective [JadeServer] and [JadeClient] sections of the JADE initialization file.

```
NetworkSpecification<n>=JadeLocal, Enabled, basename
```

```
ServerNodeSpecifications=JadeLocal, basename
```

The **basename** value can have an optional **Local** or **Global** prefix. If the prefix is absent, it defaults to **Local**, which is consistent with running with the least-privileges mode.

When running as a standard user, the value of the **basename** is created in the **Local** or session namespace, which means that all the JADE programs must be running as the same user logon and also in the same Windows session, to be able to connect to this RPC transport. For example, if the database is installed as a service, all application servers and standard clients wanting to connect to this database via **JadeLocal** transport must also be running as services and under the same user logon.

On a machine that is not part of a Windows domain, your Windows administrator must configure the user logon to add the following Windows privilege **Create global objects (SeCreateGlobalPrivilege** at the programming API level), which can be done directly to the user logon or to a group of which that user is a member. This can be done by accessing the **Local Security Policy** in the Administrative tools directory and adding the **Create global objects** policy under **Local Policies / User Rights Assignment** to the desired group or user.

These changes allow JADE programs that need to connect via the **JadeLocal** intra-machine transport to work across multiple Windows sessions or user logons.

If a standard user attempts to create a **Globalbasename** value in either of these initialization file parameters, it fails because of insufficient privileges.

Most Recently Used (MRU) List of Opened Database Paths

Because Windows does not allow standard users to write to the **HKEY_LOCAL_MACHINE** (HKLM) registry hive in a global area, the path to the system directory is recorded in the **HKEY_CURRENT_USER** (HKCU) registry hive but it is limited to the database paths that are accessed by the current user logon.

Separation of Read-Only and Read-Write Files

To support the separation of read-only and read-write files at run time, the **jade.exe** and **jaddinstd.exe** programs do not write any files into the JADE installation directory. (This affects standard clients and thin clients.)

The **JadeWorkDirectory** parameter in the [**JadeEnvironment**] section of the JADE initialization file defines the location of a directory that JADE uses for work files. This directory, which is required for internal use such as interlock files used during downloading, defaults to **<jade-program-data-directory>\temp**.

In addition, the thin client cache file is written into this directory if the location has not been specified by the **FormCacheFile** parameter in the [**JadeThinClient**] section of the JADE initialization file.

The **download.log** file created by the download install process is output to the log directory specified in the **LogDirectory** parameter in the [**JadeLog**] section of the JADE initialization file.

Note To ensure total read-only and read-write file separation, the JADE initialization file cannot be located in the JADE HOME directory.

Service Management

Because standard Windows users do not have the necessary privileges to modify entries in the **HKEY_LOCAL_MACHINE** (HKLM) area of the registry when services are installed or removed, the menu options that enable the JADE Remote Node Access utility (**jadrapp**) and a JADE application server (**jadapp**) to run as a service are disabled if you do not have the necessary privileges to install or remove an application as a service.

When a JADE application server has been installed as a service, it displays the correct state information but does not allow standard users to start, stop, or remove the service.

Windows Security Model

Because of the changes to the security model, ensure that you have the appropriate privileges or capabilities to install applications.

The configuration of the Windows User Account Control (UAC) and your current user account privileges may affect the behavior of the upgrade to JADE 2016.

For details about standard user accounts and administrator accounts, see [http://technet.microsoft.com/en-us/library/cc709691\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc709691(WS.10).aspx).

This chapter covers miscellaneous differences when running JADE applications.

Expat XML Parser

The underlying 64-bit Expat parser version 2.0.1, which is stricter in enforcing Extensible Markup Language (XML) validation rules than the 32-bit 1.95.6 version, can raise exceptions that did not occur in earlier 32-bit JADE releases.