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 © 2015 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

Before You Begin .................................................................................................................. vii
   Who Should Read this Guide ......................................................................................... vii
   What's Included in this Guide ...................................................................................... vii
   Conventions .................................................................................................................... vii
   Related Documentation ............................................................................................... viii

Chapter 1 Installing JADE under Windows ....................................................................... 10
   Operational Requirements ............................................................................................. 10
   Hardware Requirements ................................................................................................. 11
      Minimum Hardware Requirements for a JADE Database Server ................................ 11
         Processors ............................................................................................................... 11
         Memory .................................................................................................................. 11
         Storage .................................................................................................................. 11
         Other ..................................................................................................................... 12
      Minimum Hardware Requirements for an Application Server .................................. 12
         Processors ............................................................................................................... 12
         Memory .................................................................................................................. 12
         Storage .................................................................................................................. 12
         Other ..................................................................................................................... 12
      Minimum Hardware Requirements for Standard Clients .......................................... 12
         Processors ............................................................................................................... 12
         Memory .................................................................................................................. 13
         Storage .................................................................................................................. 13
         Other ..................................................................................................................... 13
      Minimum Hardware Requirements for Running Presentation Clients ...................... 13
         Processors ............................................................................................................... 13
         Memory .................................................................................................................. 13
         Storage .................................................................................................................. 13
         Other ..................................................................................................................... 13
   Software Requirements ................................................................................................. 13
      Database Server ........................................................................................................ 14
      Application Server .................................................................................................... 14
      Standard Clients ....................................................................................................... 14
      Presentation (Thin) Clients ...................................................................................... 15
      Additional Software Requirements for Deploying JADE Web Applications ............. 15
      ODBC Requirements for External Database Coexistence or JADE ODBC Driver Usage 15
      JADE Generic Messaging Requirements .................................................................. 16
      Java Framework Requirements ................................................................................ 16
      .NET Requirements ................................................................................................ 16
      Silverlight XAML Requirements .............................................................................. 16
      Requirement for Postscript Printing ......................................................................... 16
      Requirements for RPS Nodes .................................................................................... 17
      Virtual Environments ............................................................................................... 17

Installing Your JADE Software ......................................................................................... 17
   Initiating the Installation from the JADE Setup Program .............................................. 18
   JADE Software Installation Process ............................................................................. 18
      Displaying the Welcome Dialog .............................................................................. 18
      Accepting the Terms of the JADE License Agreement ............................................. 18
      Selecting the Type of Installation ............................................................................ 18
      Selecting the Type of Setup ...................................................................................... 18
      Selecting the Components to Install ......................................................................... 19
      Specifying Your User Information .......................................................................... 19
      Selecting the Destination Directory ........................................................................ 20
      Selecting the Program Folder for the JADE Icons ................................................... 20
      Running the Installation Program ............................................................................ 20
      Informing You When the Installation is Complete .................................................. 20
## Contents

**JADE Configurations under Windows**  ........................................ 21  
  Multiuser Configuration  .................................................. 21  
  Running a JADE Server in Multiuser Mode  ................................ 22  
  Running a JADE Client in Multiuser Mode  .................................. 22  
  Running a JADE User Application in Multiuser Mode  ................... 22  
  Running an Application Server in JADE Thin Client Mode  ......... 23  
  Running a Presentation Client in JADE Thin Client Mode  ......... 23  
  Single User Configuration  ............................................. 23  
  Running JADE in Single User Mode  ..................................... 23  
  Running a JADE User Application in Single User Mode  ............ 24  
  Installing Multiple JADE Initialization Files  .......................... 24  
  Reregistering JADE with a New Licence  .................................. 24

### Chapter 2  Compact JADE under Windows CE  ............................. 27  
  **Overview** ............................................................... 27  
  **Operational Requirements** .............................................. 27  
    - Microsoft Windows Mobile Operating Systems  .................. 27  
    - Writing DLLs that Support External Methods and External Functions  28  
    - Specifying a Pre-Load Library List for Compact JADE  .......... 28  
  **Hardware Specifications** .............................................. 28  
    - Processors  ......................................................... 28  
    - Memory  ............................................................ 28  
    - Storage  ........................................................... 28  
  **Other Considerations** .................................................. 29  
  Installing Compact JADE on a Desktop Machine  ...................... 29  
  Initiating the Installation from the JADE Setup Program  .......... 29  
  JADE Software Installation Process  ................................... 29  
    - Displaying the Welcome Dialog  ................................ 29  
    - Accepting the Terms of the JADE License Agreement  ........ 30  
    - Selecting the Destination Directory  ............................. 30  
    - Running the Installation Program  ................................ 30  
  Installing Compact JADE on a Windows Mobile Device  .............. 30  
  Configuring Compact JADE  ............................................. 31  
    - Running Compact JADE  ............................................ 31  
    - Automatically Upgrading Compact JADE  ........................... 33

### Chapter 3  Configuring JADE  .............................................. 34  
  Example Production System Configurations  ............................. 35  
  **Directory Locations** ..................................................... 36  
    - Installation Directory Location  ................................ 36  
    - Home Directory Location  ......................................... 37  
    - Program Data Directory Location  ................................ 37  
    - User Data Directory Location  .................................... 37  
    - Work File Directory Location  .................................... 38  
  Configuring Your Network Protocol  .................................... 39  
    - Selecting Network Addresses  .................................... 40  
    - TCP/IP Transport  .................................................. 40  
    - Hybrid Pipe Shared Memory (HPSM) Transport  ............... 41  
    - JadeLocal Transport  ............................................. 42  
  Format of the Server URI String  ....................................... 43  
    - File Scheme (Single User)  ....................................... 43  
    - TcpIP, TcpIPv4, and TcpIPv6 Schemes  .......................... 43  
    - Hybrid Pipe Shared Memory (HPSM) Scheme  .................... 45  
    - JadeLocal Scheme (Shared Memory)  ............................. 45  
  **Server Thread Model** .................................................. 45  
  Connecting to JADE Applications from Internet Information Server (IIS)  46  
  Connecting to JADE Applications from an Apache HTTP Server  .... 46  
  Using the mod_jadehttp Module  ....................................... 48
Before You Begin

The JADE Installation and Configuration Guide is intended as the main source of information when you are installing and configuring JADE.

Who Should Read this Guide

The main audience for the JADE Installation and Configuration Guide is expected to be system administrators.

What's Included in this Guide

The JADE Installation and Configuration Guide has three chapters and one appendix.

<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Covers installing JADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 2</td>
<td>Covers installing Compact JADE under Windows CE</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Covers configuring JADE</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Covers the JADE program and warning exit values</td>
</tr>
</tbody>
</table>

Conventions

The JADE Installation and Configuration Guide uses consistent typographic conventions throughout.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrow bullet (&gt;)</td>
<td>Step-by-step procedures. You can complete procedural instructions by using either the mouse or the keyboard.</td>
</tr>
<tr>
<td>Bold</td>
<td>Items that must be typed exactly as shown. For example, if instructed to type <strong>foreach</strong>, type all the bold characters exactly as they are printed.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Italic</td>
<td>Parameter values or placeholders for information that must be provided; for example, if instructed to enter <em>class-name</em>, type the actual name of the class instead of the word or words shown in italic type.</td>
</tr>
<tr>
<td></td>
<td>Italic type also signals a new term. An explanation accompanies the italicized type.</td>
</tr>
<tr>
<td></td>
<td>Document titles and status and error messages are also shown in italic type.</td>
</tr>
<tr>
<td>Blue text</td>
<td>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 &quot;Work File Directory Location&quot; cross-reference to display that topic.</td>
</tr>
<tr>
<td>Bracket symbols ([])</td>
<td>Indicate optional items.</td>
</tr>
<tr>
<td>Vertical bar (</td>
<td>)</td>
</tr>
<tr>
<td>Monospaced font</td>
<td>Syntax, code examples, and error and status message text.</td>
</tr>
</tbody>
</table>
**Convention**

- **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.

In this document, the term Microsoft Windows refers to Windows 10, Windows 8, Windows 7, Windows Server 2012, Windows Server 2008, Windows Vista, or Windows Mobile. When there are differences between the versions of Microsoft Windows, the specific version of Microsoft Windows is stated.

With the exception of the jade.exe program, when referring to program executables in this document, the .exe file suffix is omitted; for example, jadclient refers to jadclient.exe. Similarly, the .dll (Dynamic Link Library) suffix is omitted. For example, jomos refers to jomos.dll.

**Related Documentation**

Related 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.

<table>
<thead>
<tr>
<th>Title</th>
<th>Related to…</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JADE Database Administration Guide</strong></td>
<td>Administering JADE databases</td>
</tr>
<tr>
<td><strong>JADE Development Environment Administration Guide</strong></td>
<td>Administering JADE development environments</td>
</tr>
<tr>
<td><strong>JADE Development Environment User’s Guide</strong></td>
<td>Using the JADE development environment</td>
</tr>
<tr>
<td><strong>JADE Developer’s Reference</strong></td>
<td>Developing or maintaining JADE applications</td>
</tr>
<tr>
<td><strong>JADE Encyclopaedia of Classes</strong></td>
<td>System classes (Volumes 1 and 2), Window classes (Volume 3)</td>
</tr>
<tr>
<td><strong>JADE Encyclopaedia of Primitive Types</strong></td>
<td>Primitive types and global constants</td>
</tr>
<tr>
<td><strong>JADE External interface Developer’s Reference</strong></td>
<td>Developing JADE applications using external interfaces</td>
</tr>
<tr>
<td><strong>JADE Initialization File Reference</strong></td>
<td>Maintaining JADE initialization file parameter values</td>
</tr>
<tr>
<td><strong>JADE Java Developer’s Reference</strong></td>
<td>Developing or maintaining JADE applications from a Java integrated development environment</td>
</tr>
<tr>
<td><strong>JADE Synchronized Database Service (SDS) Administration Guide</strong></td>
<td>Administering JADE Synchronized Database Services (SDS), including Relational Population Services (RPS)</td>
</tr>
</tbody>
</table>
### Before You Begin

<table>
<thead>
<tr>
<th>Title</th>
<th>Related to…</th>
</tr>
</thead>
<tbody>
<tr>
<td>JADE Object Manager Guide</td>
<td>JADE Object Manager administration</td>
</tr>
<tr>
<td>JADE Platform Differences Guide</td>
<td>Platform differences when running JADE applications</td>
</tr>
<tr>
<td>JADE Runtime Application Guide</td>
<td>Administering deployed JADE runtime applications</td>
</tr>
<tr>
<td>JADE Thin Client Guide</td>
<td>Administering JADE thin client environments</td>
</tr>
<tr>
<td>JADE Web Application Guide</td>
<td>Implementing, monitoring, and configuring Web applications</td>
</tr>
</tbody>
</table>
Chapter 1  Installing JADE under Windows

This chapter covers the following topics.

- Operational Requirements
  - Hardware Requirements
    - Minimum Hardware Requirements for Running JADE Thin Client Software Only
    - Minimum Hardware Requirements for Machines not Hosting a JADE Database
    - Minimum Hardware Requirements for Machines Hosting a JADE Database
  - Software Requirements
    - Additional Software Requirements for Deploying JADE Web Applications
    - ODBC Requirements for External Database Coexistence or JADE ODBC Driver Usage
    - JADE Generic Messaging Requirements
    - Java Framework Requirements
    - .NET Requirements
    - Silverlight XAML Requirements
    - Requirement for Postscript Printing
    - Requirements for RPS Nodes
    - Virtual Environments
- Installing Your JADE Software
  - Initiating the Installation from the JADE Setup Program
  - JADE Software Installation Process
- JADE Configurations under Windows
  - Multiuser Configuration
  - Single User Configuration
  - Installing Multiple JADE Initialization Files
- Reregistering JADE with a New Licence

Operational Requirements

This section describes the JADE operational requirements and contains the following subsections.

- Hardware Requirements
- Software Requirements
Hardware Requirements

This section describes the hardware required for JADE development and JADE applications running under a Windows operating system.

Further requirements that must be met to satisfactorily provide for data recoverability are described in the Environmental Considerations for Deploying JADE white paper on the JADE Web site at https://www.jadeworld.com/developer-center/resource-library/white-papers.

Notes  These specifications represent minimum configurations. Configurations that are more powerful may be advisable, depending on the scale of your JADE applications and the performance requirements of your system.

JADE handles only the first nine monitors running on one workstation. Additional monitors are ignored.

For examples of some business-critical production system hardware considerations, see "Example Production System Configurations", in Chapter 3.

Minimum Hardware Requirements for a JADE Database Server

The following subsections contain the recommended minimum hardware required to deploy JADE on a machine that hosts a JADE database server node.

Processors

For a 64-bit database server with:

- Up to 10 concurrent users, you require an Intel Single Core CPU (Pentium, AMD, or compatible), 1 GHz or faster
- More than 10 concurrent users, you require an Intel Dual Core CPU (Pentium, AMD, or compatible), 1 GHz or faster

For client nodes, the default configuration also requires 64-bit binaries. However, if you require 32-bit client nodes (for example, to allow for the use of 32-bit third-party DLLs), these can be used with some loss of performance and functionality.

Memory

For database servers, you require 2G bytes Error Checking and Correcting (ECC) RAM.

Add 128M bytes for each JADE node.

Storage

For database servers, you require:

- Disk subsystem that guarantees that acknowledged writes are non-volatile (for details, see the Environmental Considerations for Deploying JADE white paper on the JADE Web site at https://www.jadeworld.com/developer-center/resource-library/white-papers)
- Available disk space of four times the expected database size, allowing for growth
- RAID 1+0 configuration (for details, see the Environmental Considerations for Deploying JADE white paper on the JADE Web site at https://www.jadeworld.com/developer-center/resource-library/white-papers)
Other

For database servers:

- Server class hardware is essential
- TCP/IP network environment is required
- Data backup components (for example, a backup disk drive, tape drive, or an optical read-write drive)

Minimum Hardware Requirements for an Application Server

The following subsections contain the recommended minimum hardware required to deploy JADE on machines hosting an application server.

Processors

For a 64-bit application server with:

- Up to 10 concurrent users, you require an Intel Single Core CPU (Pentium, AMD, or compatible), 1 GHz or faster
- More than 10 concurrent users, you require an Intel Dual Core CPU (Pentium, AMD, or compatible), 1 GHz or faster

Memory

For application servers, you require 2G bytes Error Checking and Correcting (ECC) RAM.

Add 128M bytes for each JADE node.

Storage

When computing disk space requirements for application servers, allow for software installation size, transient object storage, and application external file requirements.

Other

For application servers:

- Server class hardware is recommended
- Transmission Control Protocol/Internet Protocol (TCP/IP) network environment is required
- Data backup components (for example, a backup disk drive, tape drive, or an optical read-write drive) if not provided by another server (for example, by the database server)

Minimum Hardware Requirements for Standard Clients

The following subsections contain the recommended minimum hardware required to deploy JADE on a machine that hosts a JADE standard client.

Processors

For a standard client, you require an Intel CPU (Pentium, AMD, or compatible), 1 GHz or faster.
Memory

For standard clients, you require 512M bytes Error Checking and Correcting (ECC) RAM.
Add 128M bytes for each JADE node.

Storage

When computing disk space requirements for standard clients, allow for software installation size, transient object storage, and application external file requirements.

Other

For standard clients:
- Screen resolution of 800x600 or higher is required
- TCP/IP network environment is required

Minimum Hardware Requirements for Running Presentation Clients

The following subsections contain the recommended minimum hardware required to deploy JADE on a presentation (or thin) client node.

Processors

For a presentation client, you require an Intel CPU (Pentium, AMD, or compatible), 1 GHz or faster.

Memory

For presentation clients, you require 512M bytes RAM (recommended) or 256M bytes (minimum, depending on the operating system).

Storage

When computing disk space requirements for presentation clients, allow for software installation size, presentation client forms cache, and application external file requirements.

Other

For presentation clients:
- TCP/IP network environment is required
- Network Interface Card (NIC) or a dial-up adaptor and modem
- Screen resolution of 800x600 or higher is required

Software Requirements

This section describes the software required for JADE development and JADE applications.
Notes  JADE encodes Unicode characters using the wide-character-encoding routines provided by Microsoft Windows.

JADE 7.0 has been compiled using Microsoft:

- Windows C++ 2010 SP1 compiler for 64-bit and 32-bit binaries. It requires the Microsoft C++ 2010 SP1 (10.0.30319) Redistributable Package (x64) called `vcredist_x64.exe` or the Redistributable Package (x86) called `vcredist_x86.exe` to be installed. (This executable is supplied on the JADE distribution media.)

  This redistributable package should be installed on your Microsoft Internet Information Server (IIS) server if `jadehttp.dll` is used, to ensure that `jadehttp.dll` can execute correctly.

- Visual C++ 2005 SP1 compiler, for thin client 32-bit binaries. It requires the Microsoft C++ 2005 Redistributable Package (x86) called `vcredist_x86.exe` to be installed. (This executable is supplied on the JADE distribution media.)

Installation of the required C++ runtimes will be done as part of the normal JADE installation or upgrade process. Installing this Microsoft redistributable package requires administration privileges. Alternatively, you can deploy this package to all workstations before installing or upgrading to JADE 7.0, using the appropriate techniques that allow for privileged installations.

Although minimum Service Pack (SP) levels are specified, we recommend that you keep up-to-date with Service Packs and security updates for the operating system that you are running.

JADE uses different versions of the Expant Extensible Markup Language (XML) and Web service parser for 32-bit and 64-bit installations. (For details, see [http://expat.sourceforge.net/](http://expat.sourceforge.net/).) The 32-bit version is built using `expat_1.95.6` and the 64-bit version is built using `expat_2.0.1`. The 64-bit version of the parser, which is stricter in enforcing validation rules, can raise exceptions in some applications.

Database Server

For production 64-bit databases, Microsoft Windows Server 2012, Windows 10, Windows 8.1, or Windows Server 2008 with the latest security updates is the required operating system for database server nodes.

Note  Windows NT6.0 or later version (that is, Vista or Server 2008, or better) can be used for development and testing.

Application Server

64-bit Microsoft Windows Server 2012, Windows 10, Windows 8.1, Windows Server 2008, or Windows NT6.0 or later version (Vista or Server 2008, or better) with the latest security updates is the required operating system for application server nodes.

Note  To support secure client renegotiations when the `SSLPermitClientRenegotiation` parameter in the [JadeAppServer] section of the JADE initialization file to `false`, you require a minimum version of 1.0.1h of the OpenSSL libraries.

Standard Clients

JADE standard clients require one of:

- Microsoft Windows Server 2012 with the latest security updates
- Microsoft Windows Server 2008 with the latest security updates (64-bit preferred, 32-bit optional)
Chapter 1 Installing JADE under Windows

- Microsoft Windows 10
- Microsoft Windows 8.1 with the latest security updates (recommended)
- Microsoft Windows 8
- Microsoft Windows 7
- Microsoft Vista Business or better

For client nodes, the default configuration requires 64-bit binaries. However, if 32-bit client nodes are required (for example, to allow for the use of 32-bit third-party DLLs), these can be used with some loss of performance and functionality.

**Presentation (Thin) Clients**

Presentation clients require one of:

- Microsoft Windows Server 2012 with the latest security updates (64-bit preferred, 32-bit optional)
- Microsoft Windows Server 2008 with the latest security updates (64-bit preferred, 32-bit optional)
- Microsoft Windows 10
- Microsoft Windows 8.1 with the latest security updates (recommended)
- Microsoft Windows 8
- Microsoft Windows 7
- Microsoft Windows Vista Home Premium or better

**Additional Software Requirements for Deploying JADE Web Applications**

The additional software required to deploy JADE Web applications is the minimum of one of the following.

- Microsoft Internet Information Server (IIS)
- Apache 2.2.3 for Microsoft Windows

In addition, JADE Web services support SOAP version 1.1 and version 1.2.

**Notes** Unicode Web applications are not supported.

To provide greater security with firewalls, the machine hosting the Web server can be different from the machine that is running the JADE application.

**ODBC Requirements for External Database Coexistence or JADE ODBC Driver Usage**

The Microsoft Data Access Components (MDAC) installed with the Windows operating system are sufficient for running an external relational database or using the JADE ODBC standard or thin client driver.
Note  The JADE ODBC drivers are available in 32-bit and 64-bit versions. If running on a 64-bit machine, the driver used must match the third-party tool being used; for example, it may be necessary to install 32-bit JADE ODBC drivers for use with 32-bit tools.

When using an external database on a 64-bit machine, the bit version of the JADE node executing the external database access must match the bit version of the external database ODBC driver that is being used.

If you are running JADE on a 64-bit machine under Windows in 64-bit mode and you are configuring a 32-bit ODBC driver, run the following program:

```<\windows-directory>\SysWOW64\odbcad32.exe```

This runs the 32-bit version of the Microsoft Data Source Administrator program rather than the 64-bit version.

For more details, see "Configuring a JADE ODBC Driver", in Chapter 2 of the JADE External Interface Developer's Reference.

JADE Generic Messaging Requirements

The IBM WebSphere MQ interface in the JADE Generic Messaging module is built using the WMQ version 7.0.1.3 Client interface library.

The WMQ API can be used in both 64-bit and 32-bit client nodes.

Java Framework Requirements

To deploy a Java framework, you require the:

- Sun or IBM Virtual Machine Java 2 Runtime Environment (JRE), Standard Edition (J2SE) Version 6.

Note  Download the Java installation (the software package) from the Sun Microsystems Java Web site http://java.sun.com/ or the IBM Web site http://www.ibm.com/developerworks/java/.

.NET Requirements

To develop and compile .NET applications for .NET exposure, you require a minimum of:

- Microsoft Visual Studio 2010
- Microsoft Visual Studio 2010 Express, the free but limited edition of Visual Studio 2010

For external component libraries, you require:

- .NET 4.x components (or .NET 3.x, if you are using components designed for .NET 3.x)

Note  If you are using the Visual Studio 2005-built version of JADE 7.0, .NET 3.x must be installed.

Silverlight XAML Requirements

Silverlight does not run in a 64-bit mode or from a 64-bit thin client. As Microsoft does not yet support Silverlight in the 64-bit version, if your application server is running 64-bit JADE, the XAML Browser or XAML Painter requires a 32-bit presentation client JADE version.

Requirement for Postscript Printing

If using Postscript printing, your printer must support Postscript level 2 or greater.
Requirements for RPS Nodes

With the Relational Population Service (RPS), the following 64-bit SQL Server ODBC drivers must be used for the SQL Server versions.

- Microsoft SQL Server 2014 requires SQL Server Native Client 11.0 or later
- Microsoft SQL Server 2012 requires SQL Server Native Client 11.0 or later
- Microsoft SQL Server 2008 requires SQL Server Native Client 10.0 or later
- Microsoft SQL Server 2005 requires SQL Native Client or later
- Microsoft SQL Server 2000 requires SQL Native Client or later (recommended) or SQL Server

The `sqlcmd` SQL Server utility must be installed on the RPS node to use the default (recommended) `sqlcmd` to execute SQL scripts. (For details, see step 6.e under "Configuring your RPS Node", in Chapter 2 of the JADE Synchronized Database Service (SDS) Administration Guide.)

Virtual Environments

JADE can be run in a virtual environment on all supported operating systems listed in the previous sections on Intel-compatible hardware.

Support is provided as per the published release policy for faults found in a virtual environment; however, a problem that is suspected to be related to an issue in the virtualization layer may result in requests for additional customer diagnostic actions. In some rare cases, fixes for issues that are specific to the virtualization layer may need to be negotiated with the customer.

Installing Your JADE Software

The JADE software is supplied on the JADE release medium or from the JADE Web site. Install your JADE software from the `setup` executable program file contained on the release medium or from the JADE Web site.

**Notes** If you want to develop your own installation process for Windows, the JADE install and upgrade steps are documented in the `ReadmeInstallSteps.txt` file in the `documentation` directory.

Installing any Microsoft redistributable package requires administration privileges.

The 32-bit and 64-bit Microsoft C++ 2010 SP1 redistributable package (x64) called `vc_red.msi` or the 32-bit Microsoft C++ 2010 SP1 redistributable package (x86) called `vc_red.msi` is required on all Windows systems that run JADE 7.0.

Click the **Next >** button when you have specified the required information in each JADE set-up folder. The folder for the next phase of the installation process is then displayed, if applicable. Use the **< Back** button to return to earlier set-up folders. To abandon the JADE installation, click the **Cancel** button in the current set-up folder.

For details about the customization mechanism that enables you to customize the installation of JADE presentation client software, see "Customizing the Presentation Client Installer" under "Installing Application Servers and Presentation Clients", in Chapter 3 of the JADE Thin Client Guide. For details about customizing the upgrading of deployed JADE systems, see Appendix A of the JADE Runtime Application Guide, "Customizing the Deployment Upgrade Process".

For details about upgrading JADE release 6.3 databases, see "Upgrading to JADE 7.0 from a JADE 6.3.05 Release or Higher", in the `ReleaseInfo.pdf` file on your JADE release medium. (You can upgrade to JADE release 7.0 only from a JADE 6.3 release.)
Note Product information and example files are not part of the installation and must be downloaded and installed from the JADE Web site or the release medium separately, if required, into the documentation folder and examples folder, respectively, of your JADE installation directory.

Initiating the Installation from the JADE Setup Program

To initiate the installation of your JADE software

1. Ensure that you have the appropriate privileges to install applications.
2. Double-click the executable program downloaded from the JADE Web site or setup.exe on the JADE release medium.

The installation of your JADE software is then started and the initialization progress is displayed in the Setup progress dialog.

If JADE is not already installed, the required Microsoft C++ redistributables must be installed.

JADE Software Installation Process

During the installation process, the JADE set-up program performs the events described in the following subsections.

Displaying the Welcome Dialog

The JADE set-up program first displays the Welcome dialog.

Accepting the Terms of the JADE License Agreement

The Software License Agreement folder is then displayed. Use the scroll bar or the PAGE DOWN key to scroll through and read the entire software license agreement.

When you have read the entire software license agreement and you accept the specified terms, click the Yes button to continue the installation. (You cannot install JADE until you accept these license agreement terms.) Alternatively, click the No button to close the JADE set-up process.

Selecting the Type of Installation

The Installation Type folder is then displayed, to enable you to select the type of installation that you require. The types of installation that you can select are:

- **Fresh Copy**, which is selected by default, and installs a brand new copy of JADE.
- **Feature Upgrade**, which you can select if you want to upgrade an existing JADE database or binary files, or both database and binary files. For details, see the ReleaseInfo.pdf file on your JADE release medium.

Selecting the Type of Set Up

The Setup Type folder is then displayed, to enable you to select the type of set up that you require.
The types of setup that you can select are listed in the following table.

<table>
<thead>
<tr>
<th>Type</th>
<th>Installs…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>Binary, database, JADE Report Writer, and general support files required to run single user and multiuser mode of the development and application runtime environments and JADE application server and presentation client files (64-bit only)</td>
</tr>
<tr>
<td>Application Runtime</td>
<td>Only the binary and general support files required to run the JADE application server</td>
</tr>
<tr>
<td>Presentation Client</td>
<td>Only the binary and general support files required to run the presentation client in JADE thin client mode</td>
</tr>
<tr>
<td>Jade Client</td>
<td>Only the binary files and general support files required to run a JADE client</td>
</tr>
<tr>
<td>Custom</td>
<td>User-selected JADE components (for the Fresh Copy installation type only)</td>
</tr>
<tr>
<td>SDS development</td>
<td>Binary and database files for the JADE environment on SDS (Feature Upgrade installation type and 64-bit only)</td>
</tr>
</tbody>
</table>

By default, the Development option is selected for 64-bit, or Jade Client for 32-bit.

**Selecting the Components to Install**

When you specify a Custom installation, the Select Components folder is then displayed, to enable you to select the file components that you want to install. You can select the installation of the following components.

- Binary Files
- Include Files
- Library Files
- Presentation Client Files (64-bit, 32-bit, Visual Studio 2005 32-bit)
- ReadMe File
- Report Writer Files (JADE Report Writer)
- System Files (JADE development database), 64-bit only
- WSDL Files

No components are selected for installation, by default.

**To select or deselect a component for installation**

- Click the appropriate component.

The disk space required for each component is displayed at the right of the component in the Components list box.

The available disk space and the required disk space for all currently selected components are displayed at the bottom of the folder.

**Specifying Your User Information**

If you are installing a JADE database, the User Information folder enables you to specify the Licence Name (that is, your organization name, which is not to be confused with your company name that is displayed in large letters above the licence name and key) and your Licence Key specified on your licence for your JADE database.
Chapter 1 Installing JADE under Windows

An entry is required in both the Licence Name and Licence Key text boxes.

**Notes** To obtain your licence key, contact your local JADE authorized reseller.

The licence key has four text boxes. Enter the licence key exactly as it is specified on your Certificate of Authorisation (which may be an e-mail message providing you with your licence name and key). Enter the first eight characters in the first text box, the next eight characters in the second text box, the next eight characters in the third text box, and the last eight characters in the fourth text box. (Ignore the "-" printed on the licence.)

Ensure that you type the licence name of your organization (which is case-sensitive) correctly, as this is validated against your licence key.

**Selecting the Destination Directory**

The Select Installation Folders folder enables you to specify the locations of the JADE files that are to be installed. The installation process defaults to the JADE70 directory on your c: drive, and displays these values in the Install Directory, Executable Directory, and Database Directory text boxes.

> **To select a destination directory**

- Click the Browse button.

The common File dialog is then displayed, to enable you to specify the destination directory for your JADE files.

The Database Directory text box enables you to explicitly specify the location in which the database (system) files are installed. When the destination folder is not \Program Files, the database destination defaults to system under the install folder (for example, if you specify c:\Jade70 in the Install Directory text box, the database directory defaults to c:\Jade70\system).

If the installation directory is a subdirectory of the programmatically determined location of \Program Files, the \Program Files portion of the install directory is replaced with programmatically discovered location for the common application data directory (for example, if you specify c:\Program Files\Jade70 in the Install Directory text box, the default database location is c:\ProgramData\Jade70\system).

**Selecting the Program Folder for the JADE Icons**

The Select Program Folder folder is then displayed, to enable you to select the program folder in which the JADE icons are installed.

By default, the JADE icons are installed in the JADE program folder. If the program folder specified in the Program Folder text box does not exist, the installation process creates it.

The Existing Folders list box enables you to select an existing program folder, if required.

**Running the Installation Program**

The Start Copying Files folder is then displayed. This dialog displays the installation options that you have selected.

Review your selections and then click the Next > button, to start copying files.

**Informing You When the Installation is Complete**

When the installation is complete, the JADE set-up program informs you that the JADE set-up program was successfully completed and the ReadMe.txt file can now be viewed.
To view the ReadMe.txt file

- Ensure that the check box is checked (the default).

The ReadMe.txt file, which refers you to the document covering upgrading to JADE release 7.0, the changes and new features in this release, and to other documents relating to JADE release 7.0, is then displayed in a text editor (for example, Notepad).

The ReadMe.txt file is a read-only text file installed in your JADE documentation directory that you can print or delete, if required.

Click the Finish button to end the JADE installation process. Alternatively, if a system file could not be updated, the JADE set-up program prompts you to reboot your workstation so that it can complete the installation. In this case, you can view the ReadMe.txt file by using the shortcut in your specified folder.

JADE Configurations under Windows

The JADE database can be run in one of the following configurations.

- Multiuser (the default)
  
  The multiuser configuration allows access to the database from multiple processes across multiple workstations. One workstation must perform the role of server, or remote, node. For more details, see "Multiuser Configuration", in the following subsection.

- Single user
  
  The single user configuration allows access to the database from one process only (for example, only one copy of jade.exe). For more details, see "Single User Configuration", later in this chapter.

For details about accessing JADE applications from IIS or the Apache HTTP Server, see "Connecting to JADE Applications from Internet Information Server (IIS)" and "Configuring JadeHttp for Remote Connections" or see "Connecting to JADE Applications from an Apache HTTP Server" and "Configuring Apache for Remote Connections", in Chapter 3.

Multiuser Configuration

Before running JADE in multiuser mode (the default), ensure that TCP/IP has been loaded on both your client workstations and the server (remote node) workstation. For details, see "Selecting Network Addresses", in Chapter 3, or "Using the JADE Remote Node Access Utility", in Chapter 1 of the JADE Remote Node Access Utility User's Guide.

In the standard initial installation, the following parameters are set in the jade.ini file in the database directory.

```
[Jade]
server=multiUser  // Default

[JadeServer]
nodeName=LocalHost
NetworkSpecifications=TcpIP,LocalHost,6005

[JadeClient]
ServerNodeSpecifications=TcpIP,LocalHost,6005
```

This sets up the server and clients to run on the same installation node. (TcpIP is case-insensitive in JADE initialization file parameter values.)
For more details about the parameters in the JADE initialization file required to run JADE in a multiuser environment, see Chapter 1 of your JADE Initialization File Reference, particularly the "JADE Object Manager Client Module Section [JadeClient]" and "JADE Object Manager Server Section [JadeServer]" sections.

For details about using the server Uniform Resource Identifier (URI) string so that a client node can specify the target database server and transport details, see "Format of the Server URI String", in Chapter 3.

Running a JADE Server in Multiuser Mode

The default multiuser installation sets up the JADE MultiUser \JADE Database Server shortcut for the JADE server.

The following table lists examples of the properties required to run a JADE server.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jadrap.exe path=s:\jade\system ini=s:\jade\system\jade.ini app=Jade</td>
</tr>
<tr>
<td>Working directory</td>
<td>s:\jade\bin</td>
</tr>
</tbody>
</table>

Running a JADE Client in Multiuser Mode

The default multiuser installation sets up the JADE MultiUser JADE Client shortcut for the JADE multiuser client.

**Tip** When running JADE in multiuser mode as a standard (fat) client, the JADE initialization file is usually specified on the command line. If a JADE initialization file is not specified, JADE attempts to use the jade.ini file in the server node system directory.

If this directory is not visible to the client node, the default values used may not be sufficient for your application to run.

The following table lists examples of the properties required to run the JADE development environment in a JADE client.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jade.exe path=s:\jade\system ini=c:\jade\system\jade.ini app=Jade</td>
</tr>
<tr>
<td>Working directory</td>
<td>s:\jade\bin</td>
</tr>
</tbody>
</table>

Ensure that you specify the same path in the path parameter of both the JADE Remote Node Access utility (jadrap) and JADE (jade.exe).

Running a JADE User Application in Multiuser Mode

The following table lists examples of the properties required to run a user application written in JADE on a client node.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jade.exe path=s:\jade\system schema=AccountsApp app=Accounts ini=c:\jade\system\jade.ini</td>
</tr>
<tr>
<td>Working directory</td>
<td>s:\jade\bin</td>
</tr>
</tbody>
</table>
Ensure that you specify the same path in the path parameter of both the JADE Remote Node Access utility (jadrap) and JADE (jade.exe).

**Running an Application Server in JADE Thin Client Mode**

The default multiuser installation sets up the JADE MultiUser | Jade Application Server shortcut for the JADE application server.

The following table lists examples of the properties required to run an application server in JADE thin client mode.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jadapp.exe appServerPort=1500 server=multiUser path=s:\jade\system ini=s:\jade\system\jade.ini</td>
</tr>
<tr>
<td>Working directory</td>
<td>s:\jade\bin</td>
</tr>
</tbody>
</table>

For more details, see "Invoking an Application Server", in Chapter 2 of the JADE Thin Client Guide.

**Running a Presentation Client in JADE Thin Client Mode**

The default multiuser installation sets up the JADE MultiUser | JADE Presentation Client shortcut for the JADE presentation client.

The following table lists examples of the properties required to run a presentation client in JADE thin client mode.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>D:\JADE\bin\jade.exe app=SortApp schema=SortTest appServer=JadeServer appServerPort=1500</td>
</tr>
<tr>
<td>Working directory</td>
<td>D:\jade\bin</td>
</tr>
</tbody>
</table>

If a JADE initialization file is not specified on the command line when you are running an application in thin client mode, JADE attempts to use the jade.ini file in the installation directory of jade.exe (that is, the bin directory) on the presentation client node.

For more details, see "Invoking a JADE Presentation Client", in Chapter 2 of the JADE Thin Client Guide.

**Single User Configuration**

Before you run JADE in single user mode, ensure that the Server parameter in the [Jade] section of the JADE initialization file is set to singleUser. Alternatively, you can specify server=SingleUser in the command line.

**Running JADE in Single User Mode**

The default installation sets up the JADE shortcut for a single user mode client. The following table lists examples of the properties required to run the JADE development environment in single user mode.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jade.exe path=c:\jade\system ini=c:\jade\system\jade.ini app=Jade server=singleUser</td>
</tr>
<tr>
<td>Working directory</td>
<td>c:\jade\bin</td>
</tr>
</tbody>
</table>
Running a JADE User Application in Single User Mode

The following table lists examples of the properties required to run a user application written in JADE in single user mode.

<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>jad.exe path=c:\jade\system ini=c:\jade\system\jade.ini schema=TPC_A app=BenchmarkApp server=SingleUser</td>
</tr>
<tr>
<td>Working directory</td>
<td>c:\jade\bin</td>
</tr>
</tbody>
</table>

Installing Multiple JADE Initialization Files

If you want multiple users to share binary files on your Local Area Network (LAN), install multiple copies of the JADE initialization file. For example, you can install multiple initialization files so that each user has his or her own initialization file.

To install multiple JADE initialization files

1. Copy the JADE initialization file (jad.ini) into the appropriate user directory.
2. Create a JADE shortcut to meet your requirements, with the ini parameter in the command line specifying the JADE initialization file of the user, for example:

   c:\jade\bin\jad.exe ini=c:\current\myjade.ini path=c:\jade\system schema=Faults server=singleUser app=Jade

   The working-directory-name is usually the directory in which the JADE executable file (jad.exe) is located but you can specify another start-in directory when you install JADE, if required. The parameters can be in any order but must be specified once only.

   For details about multiple JADE programs on the same host sharing a JADE initialization file, see "Sharing JADE Initialization Files", in the JADE Initialization File Reference.

Reregistering JADE with a New Licence

The JadReg program installed with your JADE software enables you to reregister JADE with new licence information.

Note JADE licenses are not transferred automatically between databases in an SDE. It is your responsibility to apply new licenses to any existing databases in an SDE. In addition, to ensure proper operation, you must apply the primary license to every secondary.

For details about automating the registration of a JADE system by running the registration program in batch mode, see "Reregistering a JADE System in Batch Mode", in Chapter 3.

To reregister JADE

1. In Explorer or File Manager, access the directory in which your JADE binary files are located (for example, c:\jade\bin) and then double-click the jadreg program.
Chapter 1 Installing JADE under Windows

The Jade Registration Update dialog, shown in the following diagram, is then displayed.

2. In the **Database Directory** combo box, specify your JADE database directory or select it from the drop-down list. The last system directory that was accessed is displayed by default.

3. If you want to update your licence without stopping the JADE server, enter the name and absolute path of the JADE initialization file in the **Ini File** text box and then check the **Online** check box. This enables the **jadreg** program to obtain multiuser initialization file settings that will allow it to sign on to the JADE Remote Node Access utility (**jadrap**).

Wait until the **jadreg** program has signed on successfully before proceeding to the next step.

**Note** You must specify the absolute path and name of the JADE initialization file before you check the **Online** check box.

You must close down the JADE server before you can update your licence key if you do not specify that you want online registration update.

4. In the **Licence Name** text box, specify your organization’s licence name (displayed on your Certificate of Authorisation, which may be an e-mail message providing you with your licence name and key).

**Note** This must be typed correctly (it is case-sensitive), as it is validated against your licence key.

5. In the **Licence Key** text boxes, specify your licence key for your JADE database.
Enter the licence key exactly as specified on your Certificate of Authorisation. Enter the first eight characters in the first text box, the next eight characters in the second text box, the third eight characters in the third text box, and the last eight characters in the fourth text box. (Ignore the "*" printed on the Certificate of Authorisation.)

6. Server licences are split over the two types of run time operation; that is, standard (fat client) and JADE thin client. You can optionally specify the minimum number of licences reserved for both types of run time operation, if required, by performing the following step.

In the **Minimum Usage** text boxes, specify the minimum number of runtime licences you want reserved for:

- **Standard Jade** (that is, standard fat client runtime licences)
- **Thin Jade** (that is, JADE thin client runtime licences)

Specify minimum values for both types of runtime licence so that when you run your JADE applications, you can ensure that a specific number of licences are available (reserved) for that type of operation at any time.

**Note** You cannot specify a minimum number of licences greater than the number of your registered server (run time) licences.

The default value of zero (0) indicates that there is no minimum number of licences for each run time type. (For details about viewing the minimum number of JADE licences that you specify in this dialog and monitoring the number of licences that are available and currently in use, see "Displaying the Licence Information View", in Chapter 2 of the **JADE Monitor User's Guide**.)

7. Click the **Update** button.

The dialog then displays the following information.

- The type of licence is displayed at the bottom of the JADE development environment sign-on screen; that is, an unrestricted primary (**Production**) licence or a **Compact**, **Enterprise**, or **Free** restricted licence.

  A user application developed using an **Enterprise** or **Free** restricted licence that is invoked from outside of the JADE development environment displays a splash screen that states that the application is not for production use. The only action that the user can perform is to click anywhere on the splash screen or to press any key to remove the splash screen. A **Compact** restricted licence can be used only on Windows Mobile devices or the Compact desktop version of JADE.

- The number of developer, process, and secondary database licences for which you are registered at run time.

- If your licence is not perpetual, the expiry date of the licence.

If you specified an invalid licence name or licence key, a message box is displayed, advising you of the invalid entry.

8. Click the **Close** button.
Chapter 2  Compact JADE under Windows CE

This chapter covers the following topics.

- Overview
- Operational Requirements
- Installing Compact JADE on a Desktop Machine
- Installing Compact JADE on a Windows Mobile Device
- Configuring Compact JADE
- Automatically Upgrading Compact JADE

Overview

Compact JADE provides the ability to run a JADE thin client on Windows Mobile devices, including Windows-based Personal Digital Assistants (PDAs).

A thin client running on Windows Mobile 5.0 or 6.0 (or in the desktop emulator) can connect to an application server using Secure Sockets Layer (SSL) encryption. HTTPS-based proxy servers, including basic and digest-based authentication to the proxy server, are supported. (The TCP/IP address and port number must be specified in the JADE initialization file. There is no automatic detection of the proxy configuration from the registry.)

For a summary of the differences between a standard JADE implementation and a Compact JADE implementation on a PDA device, see "Difference between Compact JADE and Standard JADE" and "GUI Differences when Running Compact JADE", in the JADE Platform Differences Guide.

Operational Requirements

This section contains the operational requirements for a device hosting a Compact JADE node.

Microsoft Windows Mobile Operating Systems

To run Compact JADE, you require one of the following Microsoft operating systems.

- Windows Mobile 5.0 for Pocket PC Phone Edition
- Windows Mobile 5.0 for Pocket PC
- Windows Mobile 6.0 or 6.1 Classic
- Windows Mobile 6.0 or 6.1 Professional
- Windows Mobile 6.5 Professional

Note  Windows Mobile 6 Standard and Windows Phone 7 are not supported.
Writing DLLs that Support External Methods and External Functions

If you want to develop external methods or external functions for Compact JADE, you require the following software.

- Microsoft Visual Studio 2010 (or Microsoft Visual Studio 2005 for external functions on non-CRT 32-bit thin clients).

Specifying a Pre-Load Library List for Compact JADE

The [JadeEnvironment] section of the JADE initialization file, which is read from the registry at start up of a presentation client Compact JADE installation, can contain the PreLoadLibraryList parameter; for example:

[JadeEnvironment] PreLoadLibraryList=cmmobile,library2

Use this parameter to specify a comma-separated list of DLLs to be loaded early while jade.exe is loading. This ensures that the DLL is present in memory when your external function call or external method is invoked.

The default value is a blank string. The maximum length of the parameter value is 1,024 characters. Note that the .dll extension is not required.

Hardware Specifications

The following subsections summarize the Compact JADE hardware requirements.

Processors

You require ARM v4 400 MHz (minimum), 600 MHz (recommended), or better.

Memory

You require 16M bytes (minimum) of memory available to user applications.

Storage

For Compact JADE storage:

- This section refers to additional storage space to the memory requirements in the previous section. Typically, this will be a separate Secure Digital (SD) memory card or Compact Flash (CF) card of 256M bytes minimum size.
- For JADE thin client applications that have been designed to suit the small device and limit the use of highly complex forms and large graphics, we expect 256M bytes to be sufficient.

Additional storage requirements will depend on the size of the presentation client form cache, which is dependent largely on the size and complexity of application forms.

- For the best performance, selecting a fast memory card is important (for example, the SanDisk 1 GB Extreme III CF Card).
Other Considerations

For Compact JADE:

- 320 x 240 minimum display resolution is recommended
- TCP/IP network environment is required (802.11 b/g protocols are recommended)
- Depending on the performance of the cradle that comes with your device (some provide only slow transfer speeds), the use of an external Secure Digital (SD) memory card or Compact Flash (CF) card reader/writer may be desirable for faster file transfers to and from your device

On Windows Mobile devices, JADE is installed into the following.

- \storage-medium\Program Files\Jade\bin

Installing Compact JADE on a Desktop Machine

The Windows Compact JADE files include the bin (Windows executable and DLL files) subdirectory.

A separate subdirectory includes the cab and bin files required for deployment to the Windows Mobile device; that is, armv4i-msoft-wce50 for Windows Mobile 5.0 or armv4i-msoft-wm60 for Windows Mobile 6.0 Professional.

Initiating the Installation from the JADE Setup Program

Before installing Compact JADE, make sure that there are no conflicting applications running. The following instruction is an example of the installation of Compact JADE on Windows from a CD-ROM.

Alternatively, you can execute the executable program downloaded from the JADE Web site (that is, https://www.jadeworld.com/developer-center/download-jade/).

To initiate the installation of your JADE software

- Double-click the JADE\win32\Compact.exe program downloaded from the JADE Web site or setup.exe from the CompactCD CD-ROM.

The Microsoft Windows C++ 2005 SP1 Redistributable Package (x86) is installed.

Note: This package is required on all Windows systems that run Compact JADE 7.0.

The installation of your Compact JADE software is then started and the initialization progress is displayed in the Setup progress dialog.

JADE Software Installation Process

During the installation process, the Compact JADE set-up program performs the events described in the following subsections.

Displaying the Welcome Dialog

The Compact JADE set-up program first displays the Welcome folder.
Accepting the Terms of the JADE License Agreement

The Software License Agreement folder is then displayed. Use the scroll bar or the PAGE DOWN key to scroll through and read the entire software license agreement.

When you have read the entire software license agreement and you accept the specified terms, click the Yes button to continue the installation. (You cannot install Compact JADE until you accept these license agreement terms.) Alternatively, click the No button to close the Compact JADE set-up process.

Selecting the Destination Directory

The Select Destination Directory folder is then displayed, to enable you to select the path and directory in which your Compact JADE files are installed on the desktop machine. By default, the selected files are installed in subdirectories in the Jade70 directory on your c: drive.

To select a destination directory

- Click the Browse button.

The common File dialog is then displayed, to enable you to specify the destination directory for your Compact JADE files. For details about the folders in which Compact JADE presentation client files are installed and the parameters that determine the installation and configuration, see Chapter 1, "Installing Application Servers and Presentation Clients", and Appendix B, "Upgrading Software on Presentation Clients", in the JADE Thin Client Guide.

Running the Installation Program

The Start Copying Files folder is then displayed. This dialog displays the installation options that you have selected. Review your selections and then click the Next > button, to start copying files.

When the installation is complete, click the Finish button to end the Compact JADE installation process.

Installing Compact JADE on a Windows Mobile Device

Compact JADE on Windows Mobile devices does not use JADE initialization files. For details about configuring JADE on a Windows Mobile device, see "Configuring Compact JADE", later in this chapter.

The included jadethinclient.cab file contains the necessary files to run JADE in thin client mode on a Windows Mobile device.

To install Compact JADE for thin client mode

1. Copy the appropriate version of the jadethinclient.cab file to the Windows Mobile device.
2. Click on the cab file to install the thin client.
3. Enter the required start-up information in the Startup sheet of the Configure Compact Jade dialog. (For details, see "Configuring Compact JADE", later in this chapter.)

Note: We recommend that you now perform a soft reset.

For details about the Compact JADE thin client download Control Panel applet (cpl) files that are included in the files downloaded from the application server by the thin client automatic download process and copied into the Windows Mobile device Windows directory, see "Download File Types", in Appendix B of the JADE Thin Client Guide.
Chapter 2  Compact JADE under Windows CE

The icon defined for a file of type .cpl in the Windows directory appears in the System folder of the Settings menu of a Windows Mobile device.

Compact JADE includes the file jadeconfig.cpl, which enables the Compact JADE thin client configuration to be established.

Configuring Compact JADE

Compact JADE can be run in the thin client configuration on Windows Mobile devices.

For details about accessing JADE applications from IIS or the Apache HTTP Server, see "Connecting to JADE Applications from Internet Information Server (IIS)" or "Connecting to JADE Applications from an Apache HTTP Server", in Chapter 3.

The Compact JADE control panel configuration program jadeconfig.cpl enables you to:

- Configure the initiation settings for the JADE application to be run on a Windows Mobile device (for example, a PDA).
- Modify JADE initialization settings for the Windows Mobile device.

With the release of Windows Mobile 6.5.3, Microsoft has moved the OK and X buttons from the top of a maximized form to the soft menu bar at the bottom of the screen. However, as JADE creates a user-defined menu bar and Windows does not add the OK or X button to that type of menu bar, users could not close a maximized form. For all supported versions of Windows Mobile:

- The Compact JADE control panel configuration program (jadeconfig.cpl) adds a permanent soft menu bar. Under Windows Mobile 6.5.3, this bar contains an OK button. For earlier Windows Mobile versions, the OK button is displayed in the caption line of forms.

  This enables users to commit configuration changes under version Windows Mobile 6.5.3.

- The jade.exe program for all Windows Mobile versions creates a standard Windows soft menu bar if there are two or fewer visible top-level menu items and they do not have images defined. Those top-level menus are assigned to the soft menu buttons on the menu bar (a maximum of two are available).

  Under Windows Mobile 6.5.3, the menu bar also contains the Windows button and an X or OK button, if required, for a maximized form. For earlier Windows versions, the X or OK button continues to be displayed in the caption line of the form. If there are more than two visible top-level menu items or a top-level item has a defined image, a standard Windows soft menu bar cannot be created. The menu bar is user-defined, and under Windows Mobile 6.5.3, there is no X or OK button displayed.

  Notes  In this situation, your application has to include a user-defined menu item that performs the close functionality using JADE logic. The alternative solution is to redefine the menu structure so that there are two top-level menu items only.

  With the use of a standard Windows menu bar, a soft menu item button displays only a small amount of text. If the text is too long, the text is truncated with trailing points of ellipsis (...).

If you want your application to retain the previous JADE menu bar style:

1. Ensure that the form is created with three or more visible top-level menu items (or one with an image).

2. In the load event of the form, hide any menu items that do not need to be shown. If the number of menu items is decreased to two or fewer, the menu bar style remains as user-defined.
However, if the number of visible top-level items is increased beyond two, Windows changes the menu bar to a user-defined style and the X or OK button will again not be displayed.

The Compact JADE installation process automatically installs the Compact JADE control panel configuration program `jadeconfig.cpl` in the `Windows` directory. When the file is located in the `Windows` directory of the Windows Mobile device, the JADE icon is displayed with a caption of Compact JADE in the System folder of the Settings menu of the Windows Mobile device.

**Note** The thin client download install program automatically copies any downloaded `.cpl` files into the `Windows` directory.

Click the Compact Jade icon in the System folder, to display the Startup sheet of the Configure Compact Jade dialog.

To define your Compact JADE start-up options

1. In the Schema text box, specify the name of the schema that you want to invoke.
2. In the App text box, specify the name of the application that you want to initiate.
3. In A/S Host text box, specify the application server host address.
4. In the A/S Port text box, specify the application server port number.
5. Click the Ini File Options tab at the lower left of the dialog, to display the Ini File Options sheet. The Ini File Options sheet enables you to view and modify JADE initialization options stored in the Windows registry.

The Named combo box currently contains only the name of the CompactJade registry entry that holds all of the JADE initialization sections, and this value cannot be changed. (It is included for possible future enhancement.)

6. In the Section combo box, select the section name that you require from the list of all defined sections in the JADE initialization CompactJade registry section.

If the selected section has no defined parameters (keys), the Delete Key button is displayed as the Delete Section button, enabling you to delete the selected JADE initialization section.

7. In the Key combo box, select the parameter that you require from the list of all the parameters defined within the JADE initialization section selected in the previous step.

8. In the Value text box, modify the displayed value of the selected parameter to meet your requirements.

9. To add a new JADE initialization section, click the Add Section button. This button is enabled only if the name displayed in the Section combo box in step 6 of this instruction has been modified so that it does not match the name of an existing JADE initialization section.

10. Click the Update Value button to set the value of the selected parameter to the contents of the Value text box or to create a new parameter with the value specified in the Value text box if the value displayed in the Key combo box has also been modified so that it does not match the names of an existing parameter in the JADE initialization section displayed in the Section combo box.

11. Click the Delete Key to delete the parameter currently selected in the Key combo box.

12. Click OK button at the top right of the dialog to save any changes made to these controls and to close the program.

**Note** The changes made using this dialog are immediate, and are saved independent of clicking the OK button.
Running Compact JADE

Execute `jade.exe` from the installed directory.

The command line and JADE initialization file settings are read from the registry. For more details, see "Invoking a JADE Presentation Client" in Chapter 2 of the JADE Thin Client Guide.

Automatically Upgrading Compact JADE

In addition to the download directory file names and corresponding JADE initialization file sections documented under "Upgrading Software on Presentation Clients", in Appendix B of the JADE Thin Client Guide (for example, the existing `[i686-msoft-win32_vs2005-unicode]`), the following directory file and section names are required for Compact JADE.

- `[armv4-msoft-wm60-unicode]`
- `[armv4i-msoft-wince50-unicode]`
Chapter 3

Configuring JADE

This chapter covers the following topics.

- Example Production System Configurations
- Directory Locations
- Configuring Your Network Protocol
  - Selecting Network Addresses
  - TCP/IP Transport
  - Hybrid Pipe Shared Memory (HPSM) Transport
  - JadeLocal Transport
  - Format of the Server URI String
  - Server Thread Model
- Connecting to JADE Applications from Internet Information Server (IIS)
- Connecting to JADE Applications from an Apache HTTP Server
- Configuring Your JADE Software
  - Configuring the Internet Protocol Version
    - RPC, Database, and Node Configuration
  - Configuring JadeHttp for Remote Connections
    - Firewall for the JADE Internet Environment
    - Controlling the Location of Files Uploaded via a Web Application
      - [application-name] Section
      - [Jadehttp Files] Section
      - [Jadehttp Logging] Section
  - Configuring Apache for Remote Connections
  - Tuning Your Systems
- Specifying Parameters in the JADE Command Line
- Specifying Your Administration Options
- Reregistering a JADE System in Batch Mode
- Using the JADE Version Information Utility
Example Production System Configurations

The following table represents some business-critical production system hardware configurations as at July 2007. These are intended as indicative examples only.

More-powerful configurations may be required, depending on the scale of your JADE applications and the performance requirements of your system.

**Note** In the following table, all CPUs are Intel or Intel-compatible.

<table>
<thead>
<tr>
<th>Peak Users</th>
<th>Nodes</th>
<th>Database Size</th>
<th>Hardware Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>5G bytes</td>
<td>Single server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 x single core 1 GHz CPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2G bytes ECC RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 72G bytes 15K drives (database/OS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 72G bytes 15K drives (journals/backup)</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>8G bytes</td>
<td>Single server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 x Dual Core 1 GHz CPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.5G bytes ECC RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 72G bytes 15K drives (database/OS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 72G bytes 15K drives (journals/backup)</td>
</tr>
<tr>
<td>190</td>
<td>20</td>
<td>80G bytes</td>
<td>Two servers, clustered, each server with:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x single core 3.4 GHz CPUs with Hyperthreading</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5G bytes ECC RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 72G bytes 15K drives (OS/workspace)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Storage Area Network (SAN) containing:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 x 72G bytes 15K drives (database)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 144G bytes 15K drives (journals)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 x 300G bytes 15K drives (backup)</td>
</tr>
<tr>
<td>225</td>
<td>13</td>
<td>115G bytes</td>
<td>Single server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 x single core 3 GHz CPUs with Hyperthreading</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.5G bytes ECC RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 x 72G bytes 15K drives (database)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 x 144G bytes 15K drives (journals)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 x 144G bytes 15K drives (backup)</td>
</tr>
<tr>
<td>350</td>
<td>24</td>
<td>240G bytes</td>
<td>Single server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 x single core 3 GHz CPUs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20G bytes ECC RAM</td>
</tr>
</tbody>
</table>
### Directory Locations

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 installed in the \Program Files\ system-wide location for installed applications, alternate directory locations are implemented. Because security restrictions can deny appropriate access to directories or the directories are a unique location for each user, the Node and Process classes provide methods that enable you to access other directory locations in the file system hierarchy. In addition, the [JadeEnvironment] section of the JADE initialization file provides the JadeWorkDirectory, ProgramDataDirectory, and UserDataDirectory parameters, which enable you to specify the location of JADE work files, program data, and user data, respectively.

In the methods summarized in the following subsections, a method on a specific:

- Node instance performs its action on the specified node instance, which does not have to be the current node. If the current node is required, use the node environmental object (system variable).
- Process instance performs its action on the owning node (that is, a Process.node instance) if the process is not associated with a presentation client. If the process has an associated presentation client, the action is performed on the presentation client. The presentation client does not have to be the current presentation client or a presentation client attached to the same application server.

For details, see Volume 1 and Volume 2 of the JADE Encyclopaedia of Classes and "JADE Environment [JadeEnvironment] Section", in the JADE Initialization File Reference.

### Installation Directory Location

The getJadeInstallDirectory method defined in the Node and Process classes returns a string containing the directory in which the executable of the current executing program is located; that is, the directory in which the JADE binaries are installed.

See also "Work File Directory Location", later in this chapter.
Home Directory Location

The `getJadeHomeDirectory` method defined in the `Node` and `Process` classes returns a string containing the parent directory of the installation directory.

Program Data Directory Location

The `getProgramDataDirectory` method defined in the `Node` and `Process` classes returns a string containing the program data directory, which is dependent on the:

- Value of the `ProgramDataDirectory` parameter in the `[JadeEnvironment]` section of the JADE initialization file. The parameter is read each time the `getProgramDataDirectory` method is called. When you call this method and the directory does not already exist, JADE creates it based on the value of the `ProgramDataDirectory` parameter.

- 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.

If the installation directory is a subdirectory of the programmatically determined location of `\Program Files`, translation rules are applied. However, if the installation directory is not a subdirectory of the `\Program Files` location, the return value is the same as the JADE HOME directory.

The `ProgramDataDirectory` parameter in the `[JadeEnvironment]` section of the JADE initialization file can have one of the following values:

- `<default>`, which is the default value
  The `<default>` value has the same meaning as the `<programdata>` value.

- `<homedir>`
  The value of the JADE HOME directory is returned.

- `<programdata>`
  The `\Program Files` portion of the HOME directory is replaced with the programmatically obtained location for the common application data directory. For example, a presentation client installed into `\Program Files\Jade Software Corporation\parsys` returns `\ProgramData\Jade Software Corporation\parsys`.

- A user-specified directory name, which returns the specified directory
  The directory location is created, if necessary, and returned in the required format.

User Data Directory Location

The `getUserDataDirectory` method defined in the `Node` and `Process` classes returns a string containing the user data directory, which is dependent on the:

- Value of the `UserDataDirectory` parameter in the `[JadeEnvironment]` section of the JADE initialization file. The parameter is read each time the `getUserDataDirectory` method is called. When you call this method and the directory does not already exist, JADE creates it based on the value of the `UserDataDirectory` parameter.

- Location of the installation directory.
Files that should be placed under the returned location are entities that should be unique to each user 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, unique dictionaries for each user).

If the installation directory is a subdirectory of the programmatically determined location of \Program Files, translation rules are applied. However, if the installation directory is not a subdirectory of the \Program Files location, the return value is the same as the JADE HOME directory.

The UserDataDirectory parameter in the [JadeEnvironment] section of the JADE initialization file can have one of the following values.

- `<default>`, which is the default value
  
The `<default>` value has the same meaning as the `<userdata>` value.

- `<homedir>`
  
The value of the JADE HOME directory is returned.

- `<userdata>`
  
The \Program Files portion of the HOME directory is replaced with the programmatically determined location for the specific user application private data directory. For example, a presentation client installed into \Program Files\Jade Software Corporation\parsys and executed by user wilbur returns \Users\wilbur\AppData\Local\Jade Software Corporation\parsys.

- A user-specified directory name, which returns the specified directory

The directory location is created, if necessary, and returned in the required format.

**Work File Directory Location**

The `getJadeWorkDirectory` method defined in the Node and Process classes returns a string containing the JADE work directory, which is dependent on the:

  
The parameter is read each time the `getJadeWorkDirectory` method is called and the directory is accessed. When you call this method and the directory does not already exist, JADE creates it based on the value of the JadeWorkDirectory parameter.

- Location of the JADE HOME directory.

By default, the jade.exe executable program file does not directly create or update files in the binary directory.

- The JadeWorkDirectory parameter in the [JadeEnvironment] section of the JADE initialization file determines the directory in which work files are created.

  By default, this directory is created at the same level as the JADE binary directory and is named `temp`. For example, the directory is named `c:\jade\temp` if the JADE installation directory is `c:\jade\bin`.

  The JadeWorkDirectory parameter can specify an absolute path or a relative path (relative to the JADE HOME directory, which is `c:\jade` in the above example).

- The presentation cache file is written into the directory defined by the value of the JadeWorkDirectory parameter unless the FormCacheFile parameter in the [JadeThinClient] section of the JADE initialization file specifies the location of the form cache file.

- The presentation client automatic download process lock files are created in the directory specified by the
Chapter 3  Configuring JADE

**JadeWorkDirectory** parameter.

- The automatic presentation client download log file is written to the location specified by the **LogDirectory** parameter in the [JadeLog] section of the JADE initialization file.

The only situations in which a file creation or update will occur within the JADE binary directory are as follows.

- If the JADE initialization file is positioned in the binary directory (the default action for a presentation client). You can avoid this by using the **ini** parameter in the JADE presentation client command line to specify an alternate location on initiation of the presentation client.

- When the **jadinst** executable program installs files downloaded by the JADE thin client automatic download facility.

### Configuring Your Network Protocol

TCP/IP is required for multiuser configurations. TCP/IP must be installed on each workstation that is to participate as a node in the JADE environment. Your network manufacturer supplies TCP/IP and your network administrator supplies the TCP/IP address.

JADE also provides fast transport mechanisms for JADE modules on the same machine (for example, one that has multiple CPUs). For details, see "Hybrid Pipe Shared Memory (HPSM) Transport" and "JadeLocal Transport", later in this chapter.

Use the:

- **NetworkSpecification** parameter in the [JadeServer] section of the JADE initialization file to specify the transport types used by a JADE server.

- **ServerNodeSpecifications** parameter in the [JadeClient] section of the JADE initialization file to specify the transport type used by a JADE client to connect to the server.

- Server URI string in the **server** parameter of a command line, to specify the target database and the client-server transport. For details, see "Format of the Server URI String", later in this chapter.

As JADE database servers and JADE application servers can often reside on the same machine, you can use a local intra-machine transport to significantly improve performance. You can also use these transport mechanisms to communicate from standard (fat) clients and JADE application servers to JADE servers if these processes run on the same physical machine, to significantly improve overall performance. Intra-machine local transport uses **shared memory**.

To detect whether a connection has been lost, a polling mechanism ensures that some minor communication is taking place regularly. See also "TransportIdlePollInterval" under "JADE Object Manager Server Section [JadeServer]", in the JADE Initialization File Reference.

If your database is running as a service but not with the **Local System** account, 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. On a machine that is not part of a Windows domain, 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. This allows JADE programs that need to connect via the **JadeLocal** or HPSM intra-machine transport to work across multiple Windows sessions or user logons.

With the HPSM transport, a database running with a standard account requires the extra privilege. Client nodes do not require the extra privilege.

With the **JadeLocal** and HPSM transports, if a standard user attempts to create a **Globalbasename** value in either of these initialization file parameters, it fails because of insufficient privileges.
Selecting Network Addresses

Use the reserved range of Class C network addresses if you want to set up a small private network for demonstration purposes. (The reserved private Class C network addresses are in the range 192.168.0 through 192.168.255, and are often used for broadcast messages.) If you therefore select network number 192.168.1.1 for example, you can then assign all IP addresses for your host in the range 192.168.1.1 through 192.168.1.254.

Ensure that the first three parts of this address are the same for all interfaces connected to the same LAN segment, and set the subnet mask to 255.255.255.0.

Tip Choose a reserved private Class C IP address when one or more of the workstations also needs to connect to the Internet by using an Internet Service Provider (ISP). If you use these reserved addresses, they should not conflict with any valid Network Interface Controller (NIC)-assigned Internet addresses. Avoid using random numbers for IP addresses, as this usually fails.

When you have configured your network and IP addresses, check that workstations can see each other using the IP protocol.

A simple test is to use ping from a command prompt, by entering one or both of the following command prompts from any workstation:

```
ping computer-name
ping remote-IP-address
```

If this does not provide a response, you will not be able to use the specified workstation name as the server node name in the ServerNodeSpecifications parameter of the [JadeClient] section of your JADE initialization file.

If the IP is functioning correctly and the ping remote-IP-address command prompt provides a positive response, setting the ServerNode parameter to the server node IP address value in the ServerNodeSpecifications parameter in the [JadeClient] section of your JADE initialization file should work. Setting this ServerNode parameter to the workstation name of the server node may not always work even if the ping server-node-computer-name command prompt gives a valid response, as this alias to standard TCP/IP cannot always be relied on to work.

If you do not have Windows Internet Naming Services (WINS) or a Domain Name Server (DNS) server, you should set up a hosts file with IP address-to-name mappings and then use the names from there. If you have a small single-segment LAN and you do not know what Dynamic Host Configuration Protocol (DHCP)/DNS or WINS are, do not use them for your local networking. Only an experienced network administrator should set these up. (If you have Internet access, you will almost certainly have DNS configured.)

TCP/IP Transport

The TCP/IP transport must be used when the database server and client are on different computers. For connections on the same computer, the fast Hybrid Pipe Shared Memory (HPSM) transport is recommended, although TCP/IP or the JadeLocal transport can be used. See also "RPC, Database, and Node Configuration", later in this chapter.

The following JADE initialization file parameters are required to use the TCP/IP transport.

```
[JadeServer]
NetworkSpecification<n>=transport-type,enabled|disabled,
listener-port[,interface]

[JadeClient]
```
Alternatively, use the `server` parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the `server` and `path` parameters in a command line and the `ServerNodeSpecifications` parameter in the `[JadeClient]` section of the JADE initialization file. For details, see "Format of the Server URI String", later in this chapter.

The values (which are case-insensitive) for the `transport-type` variable in the `NetworkSpecification` parameter are:

- **TcpIP**, which is a synonym for **TcpIPv4**.
- **TcpIPv4**, which provides IP version 4 connections only.
- **TcpIPv6**, which provides IP version 6 connections only.

If you want the server to support both network protocols, you must define a network specification for each protocol.

You can specify the `interface` value as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the specified `transport-type` value. Specify the local interface name or IP address if you want to select a specific network adapter in a database server node that has more than one network adapter installed; for example, to enable an administrator to ensure connections from clients connect on the fastest interface or to allow easier security when used in conjunction with a firewall or router access list. (JADE defaults to all network adapters in the node.)

The values for the `transport-type` variable in the `ServerNodeSpecifications` parameter are:

- **TcpIP**, which is a synonym for **TcpIPAny**.
- **TcpIPAny**, which connects using IP version 4 or 6.
- **TcpIPv4**, which connects using IP version 4.

You can specify the `remote-host` and `local-interface` values as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the specified `transport type` value. If you specify a host name, all DNS-provided addresses will be attempted.

For the **TcpIPAny** transport type, the client will first attempt to connect via IP version 6 and then IP version 4 protocol on the provided IP addresses. Each connection failure will be logged, and the next available combination tried.

**Hybrid Pipe Shared Memory (HPSM) Transport**

Hybrid Pipe Shared Memory (HPSM) provides a fast transport mechanism between JADE nodes on the same computer to utilize the multiple CPUs and shared memory, to significantly improve overall performance. (An older shared-memory transport, called **JadeLocal**, is documented in the following section.)

The Remote Procedure Call (RPC) `jomsrvr2.dll` includes a Hybrid Pipe Shared Memory (HPSM) transport, which has lower overhead than the **JadeLocal** (shared memory) transport and it is a better fit with the server thread model, as it results in fewer thread context switches per request.

**Note** The HPSM transport can be used only between nodes residing in the same machine, because it uses shared memory.
The following JADE initialization file parameters are required to use the HPSM transport.

[JadeServer]
NetworkSpecification1=HPSM,enabled|disabled,[Global]\base-name

[JadeClient]
ServerNodeSpecifications=HPSM,[Global]\base-name

The following is an example of the required JADE initialization file definitions to use the HPSM transport.

[JadeServer]
NetworkSpecification1=HPSM,Enabled,TestCRM-HPSM

[JadeClient]
ServerNodeSpecifications=HPSM,TestCRM-HPSM

The base-name value can have an optional Local or Global prefix.

When the Local prefix tag is specified (or no prefix tag is specified), the value of the base-name variable is created in the session namespace. To establish a connection, the JADE database server and the JADE client must be running under the same user logon and in the same Windows session. This would not be the case, for example, if you were running an interactive desktop application that attempted to connect to the JADE database server.

When the Global prefix tag is specified, the base-name variable is created as a global object that is visible across sessions and logons.

Note Use the Global prefix tag, in the NetworkSpecification and the ServerNodeSpecifications parameter to enable an HPSM connection to the JADE Database service across Windows sessions.

The HPSM transport has lower overhead than the JadeLocal transport, and is a better fit with the server thread model, as it has fewer thread context switches per request. For details about the server thread model, see "Server Thread Model", later in this chapter.

On a client node using HPSM, the Node class networkAddress method returns "procNNNN", where the NNNN value is the decimal number of the process at the other end of the connection.

Alternatively, use the server parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the server and path parameters in a command line and the ServerNodeSpecifications parameter in the [JadeClient] section of the JADE initialization file. For details, see "Format of the Server URI String", later in this chapter.

JadeLocal Transport

The JadeLocal transport between the JADE database and standard clients is implemented by the use of shared memory, global events, and semaphores.

Note The JadeLocal transport should be used when a node has one or two processes only.

The following JADE initialization file parameters are required to use the JadeLocal transport.

[JadeServer]
NetworkSpecification<n>=JadeLocal,enabled|disabled,[Global]\base-name

[JadeClient]
ServerNodeSpecifications=JadeLocal,[Global]\base-name
The \textit{base-name} value can have an optional \texttt{Local} or \texttt{Global} prefix, as described for the HPMS transport, earlier in this chapter.

If the prefix is absent, it defaults to \texttt{Local}, which is consistent with running with the least-privileges mode. When running as a standard user, the value of the \textit{base-name} variable is created in the \texttt{Local} or session namespace, which means that all 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 \texttt{JadeLocal} transport must also be running as services and under the same user logon.

\textbf{Note} Local intra-machine transport is used only when the transport type is defined as \texttt{JadeLocal} or \texttt{HPMS} on both the server and client and the network state on the server is defined as \texttt{Enabled}.

Alternatively, use the \texttt{server} parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the \texttt{server} and \texttt{path} parameters in a command line and the \texttt{ServerNodeSpecifications} parameter in the [\texttt{JadeClient}] section of the JADE initialization file. For details, see "Format of the Server URI String", in the following section.

\section*{Format of the Server URI String}

Use the \texttt{server} parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the \texttt{server} and \texttt{path} parameters in a command line and the \texttt{ServerNodeSpecifications} parameter in the [\texttt{JadeClient}] section of the JADE initialization file.

The syntax of the server URI command line parameter is as follows.

\begin{verbatim}
server=scheme://target-address/environment-UUID[/server-UUID][?parameters]
\end{verbatim}

The \texttt{scheme} and \texttt{target-address} values specify the client-server transport definition. The environment identity UUID (in display form) specifies the database to which the client node is connecting. This is optionally followed by the database server identity UUID. For details about the database identities, see "Database Identities", in Chapter 3 of the \texttt{JADE Database Administration Guide}.

Specify both environment and server identities when you want the client node to connect to a specific SDS secondary database.

The following subsections explain how you can specify the target address for each URI scheme.

\section*{File Scheme (Single User)}

The format of the single user File scheme is as follows.

\begin{verbatim}
File://database-directory
\end{verbatim}

The \texttt{database-directory} value specifies the name of the directory where the database server finds the database control file (_\texttt{control.dat}). Use this format when running a single user client node; for example:

\begin{verbatim}
File://c:\jade\system
\end{verbatim}

This example is equivalent to \texttt{server=SingleUser} and \texttt{path=c:\jade\system} in the command line.

\section*{TcpIP, TcpIPv4, and TcpIPv6 Schemes}

The format of the target address in the TcpIP, TcpIPv4, and TcpIPv6 schemes is as follows.

\begin{verbatim}
host:port-number
\end{verbatim}
Chapter 3 Configuring JADE

You can specify the host using a:

- Simple host name; for example, `george`, `server129`, or `localhost`.
- Fully qualified domain name (FQDN); for example, `wilbur.example.com`.
- Dotted decimal IPv4 address; for example, `127.0.0.1`.
- Quadded hexadecimal IPv6 address enclosed in square brackets; for example, `[fe80::bf6d:6902:7deb]`.

The `TcpIPv4` scheme uses the IPv4 protocol, so quadded hexadecimal IPv6 addresses are not allowed.

The `TcpIPv6` scheme uses the IPv6 protocol, so dotted decimal IPv4 addresses are not allowed.

The `port-number` value is a decimal integer in the range 1 through 65535.

When the host is a simple host name or an FQDN, resolution always returns `TcpIPv6` addresses before `TcpIPv4` addresses. When a specific protocol version is not specified, the first returned address is used.

The following examples show target addresses for the `TcpIP`, `TcpIPv4`, and `TcpIPv6` schemes.

```plaintext
server= TcpIP://localhost:6005/48cf13df-bf6d-df11-87e2-2e5925024153
server= TcpIPv4://host.company.com:6005/48cf13df-bf6d-df11-87e2-2e5925024153?localHostname=aHostName
server= TcpIPv6://[fe81::6fe:7fff:fe87:bd20]:6005/48cf13df-bf6d-df11-87e2-2e5925024153?localPort=54321
server= TcpIP://127.0.0.1:6005/48cf13df-bf6d-df11-87e2-2e5925024153/48cf13df-bf6d-df11-87e2-2e5925024153
```

By default, the environment identity specified in the `server` command line parameter of a client shortcut must match the identity of the database server, as shown in the following example.

```plaintext
server= TcpIP://localhost:6005/48cf13df-bf6d-df11-87e2-2e5925024153
```

If zeroes can be specified for the environment identity in the `server` command line parameter of a client shortcut, as shown in the following example, set the value of the `AcceptZeroEnvironmentUUID` parameter in the `[JadeServer]` section of the JADE initialization file to `true`.

```plaintext
server= TcpIP://localhost:6005/00000000-0000-0000-0000-000000000000
```

**Caution** The `AcceptZeroEnvironmentUUID` parameter should not be set to `true` in production databases.

The following optional parameters enable you to specify the client node source address and port.

- **LocalInterface=host**
  
  The `host` value is a simple host name, a fully qualified domain name, or an appropriate IP address. You can use this to select a specific source address when the client machine has more than one network interface.

- **LocalPort=port-number**

  The `port-number` value is a decimal integer in the range 1 through 65535. You can use this when the default random source port number cannot be used; for example, when the client and server machines are separated by a firewall, which limits the client source port numbers.
Hybrid Pipe Shared Memory (HPSM) Scheme

In the Hybrid Pipe Shared Memory (HPSM) scheme, shared memory is used to establish a connection so the client node must be on the same machine as the server.

The format of the target address is as follows.

```
localhost:base-name
```

The host must be localhost. The base-name contains up to 60 ASCII graphic characters excluding the forward slash (/), percent (%), and comma (,) characters. When the database server is installed as a Windows service, the base-name should be prefixed with Global.

The following example shows a target address for the HPSM scheme.

```
server=HPSM://localhost:SRCJade-HPSM/48cf13df-bf6d-df11-87e2-2e5925024153
```

JadeLocal Scheme (Shared Memory)

In the JadeLocal scheme, shared memory is used to establish a connection so the client node must be on the same machine as the server.

The format of the target address is as follows.

```
localhost:base-name
```

The host must be localhost. The base-name contains up to 60 ASCII graphic characters excluding the forward slash (/), percent (%), and comma (,) characters. When the database server is installed as a Windows service, the base-name should be prefixed with Global, and the base-name value name in the range 1 through 60 characters.

The following example shows a target address for the JadeLocal scheme.

```
server=JadeLocal://localhost:SRCJade-SHM/48cf13df-bf6d-df11-87e2-2e5925024153
```

Server Thread Model

The server thread model uses two thread pools, as follows.

- The first pool (short threads) handles request types that have a short execution time (for example, a getobject call).
- The second pool (long threads) handles the request types that are expected to have a relatively long execution time.

Both thread pools use a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

The thread pools are managed with parameters in the [JadeServer] section of the JADE initialization file, as shown in the following example.

```
[JadeServer]
TransportIdlePollInterval=120000
MinShortThreads=10
MaxShortThreads=100
ConcurrentShortThreads=0
MinLongThreads=5
MaxLongThreads=50
ConcurrentLongThreads=0
```
All transports perform a keep-alive check, whose frequency is controlled by the value of the `TransportIdlePollInterval` parameter.

The number of threads in each pool grows dynamically, starting with the values of the `MinShortThreads` and `MinLongThreads` parameters, up to the values of the `MaxShortThreads` and `MaxLongThreads` parameters, depending on the request arrival rate.

The values of the `ConcurrentShortThreads` and `ConcurrentLongThreads` parameters control the maximum number of threads that the operating system can allow to concurrently process I/O completion packets for each I/O completion port. If the value of these parameters is the recommended value of zero (0), the system allows as many concurrently running threads as there are cores in the system. (For further details, see the Microsoft MSDN `CreateIoCompletionPort` function at [http://msdn.microsoft.com](http://msdn.microsoft.com).)

## Connecting to JADE Applications from Internet Information Server (IIS)

Your connections from the Internet Information Server (IIS) to JADE applications are via a TCP/IP connection. (For more details, see "Configuring JadeHttp for Remote Connections", later in this chapter.)

To implement a TCP/IP connection from IIS to JADE applications, define a section within the `jadehttp.ini` file for each application, with the `ApplicationType` parameter and unique multiple `TcpPort[n]`, `MinInUse[n]`, `MaxInUse[n]`, `TcpConnection[n]`, `CloseDelay[n]`, and `ConnectionGroup[n]` parameters within each `[application-name]` section. Set the `ApplicationType` parameter in each `[application-name]` section to `WebEnabledForms`, `WebServices`, or `HtmlDocuments`. The following is an example of these parameters.

```
[HtmlCompany]
ApplicationType = HtmlDocuments
TcpConnection = Host1a
TcpPort = 22000
MinInUse = 2
MaxInUse = 10
CloseDelay = 600
TcpConnection2 = Host2a
TcpPort2 = 22001
MinInUse2 = 2
MaxInUse2 = 10
CloseDelay2 = 600
```

The `jadehttp.dll` provides support for JADE 6.2 Web applications, with the valid combinations listed in the following table.

<table>
<thead>
<tr>
<th>Node</th>
<th>ApplicationType</th>
<th>TCP/IP</th>
<th>Named Pipe</th>
</tr>
</thead>
<tbody>
<tr>
<td>JADE 6.2</td>
<td>WebServices, HtmlDocuments, WebEnabledForms</td>
<td>Single port number, multiple connections, all Web applications</td>
<td>Not allowed</td>
</tr>
</tbody>
</table>

## Connecting to JADE Applications from an Apache HTTP Server

Your connections from the Apache HTTP Server to JADE applications can be via a TCP/IP connection only. The design and configuration of the `mod_jadehttp` module fully conforms to Apache HTTP Server 2.2 module usage and configuration practices.
You can obtain more information about the Apache Software Foundation from one of the following Web sites.

http://www.apache.org
http://httpd.apache.org

JADE supports the Apache HTTP Server by supplying mod_jadehttp. Only JadeInternetTCPICPConnection class TCP/IP communications are supported for all versions of mod_jadehttp, regardless of the operating system; that is, TCP/IP connections only are used for connections to JADE applications running from the Apache HTTP Server. For details, see "Using the mod_jadehttp Module", in the following subsection.

The mod_jadehttp library module supplied by JADE for the Apache HTTP Server implements support for the following Multi-Processing Modules (MPMs). The selection of the default MPM is selected based on the operating system on which the Apache server is running; that is, mpm_winnt is the default value for Microsoft-based operating systems.

If you build the Apache server from source code, you can build and select whichever MPM you want to use, on the operating system of your choice.

Some relevant MPMs with respect to JADE and the mod_jadehttp library are listed in the following table.

<table>
<thead>
<tr>
<th>Multi-Processing Module (MPM)</th>
<th>Implements a …</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpm_winnt</td>
<td>Multiple-threaded single process Web server</td>
</tr>
<tr>
<td>prefork</td>
<td>Non-threaded, pre-forking Web server</td>
</tr>
<tr>
<td>worker</td>
<td>Hybrid multiple-threaded multiple-process Web server</td>
</tr>
</tbody>
</table>

The mod_jadehttp module ensures that the same destination IP address and port number combination are used for a specific logical connection, by using the hidden fields that both the jadehttp and the mod_jadehttp libraries insert into the HTML data.

The mod_jadehttp library module provides support for JADE 6.2 Web applications, with the valid combination of the ApplicationType directive values listed in the following table.

<table>
<thead>
<tr>
<th>Node</th>
<th>ApplicationType Directive Value</th>
<th>TCP/IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>JADE 6.2</td>
<td>WebServices, HtmlDocuments, WebEnabledForms</td>
<td>Single port number, multiple connections, all Web applications</td>
</tr>
</tbody>
</table>

To install the mod_jadehttp module

- Copy the mod_jadehttp.so file into the modules directory of your Apache HTTP Server.

Notes: The name mod_jadehttp.so applies to all operating systems.

Ensure that you use the correct operating system binary.

Apache and the HTTP generally use UTF-8 as the encoding scheme for Unicode data on the Web. As mod_jadehttp currently does not allow for this and it passes the data directly to JADE, only ANSI data can be read from or written to the JadeInternetTCPICPConnection object.

For details about configuring the Apache HTTP Server, see "Configuring Apache for Remote Connections", later in this chapter.
Using the mod_jadehttp Module

JADE includes mod_jadehttp32.so and mod_jadehttp64.so, for the 32-bit and 64-bit version of Apache 2.2.3, respectively.

To use the mod_jadehttp module

1. Ensure that you are running Apache 2.2.3 or a higher 2.2.n version.
2. Determine whether 32-bit or 64-bit Apache is installed. (If the operating system installation is a 64-bit version, it is most likely that Apache is also the 64-bit version.)
3. Load the correct mod_jadehttp module into Apache. Note that path names in following examples may differ from those in your environment.

   For 32-bit versions of Apache:
   
   ```
   LoadModule jadehttp_module modules/mod_jadehttp32.so
   ```

   For 64-bit versions of Apache:
   
   ```
   LoadModule jadehttp_module modules/mod_jadehttp64.so
   ```

Configuring Your JADE Software

The JADE initialization file, which enables you to configure JADE to your requirements, contains:

- [Jade] section, which contains information that customizes the start-up image that is displayed when JADE opens the database and initializes the application.
- [JadeCommandLine] section, which contains information that would normally be placed on the command line.
- [JadeCompiler] section, which contains information that controls the logging of compiler warnings.
- [DeltaOptions], [JadeClient], [JadeClientAppRestrictions], [JadeLoader], and [JadeServer] sections, which contain information that defines your JADE Object Manager environment. Certain modules of the JADE Object Manager can use the information stored in these sections to configure themselves to meet defined performance or resource constraints or your own configuration requirements.
- [JadeEnvironment] section, which contains information that is relevant to both standard clients and thin client nodes (that is, the location of program data, user data, and JADE work file directories).
- [JadeExecuteFlags] section, which contains a parameter that determines whether instructions within an executeWhen instruction are to be loaded and executed.
- [JadeExtractSort] section, which contains the name and optional relative path of the directory in which temporary sort files are located.
- [JadeInterpreter] and [JadeInterpreterOutputViewer] sections, which contain information about the method cache and whether the write instruction can be actioned.
- [JadeLog] section, which contains information that controls the logging of JADE errors.
- [JadeMonitor], [JadeMonitorBackground], and [JadeMonitorSecurity] sections, which contain information used to initialize JADE Monitor sessions.
- [PersistentDb] and [TransientDb] sections, which contain information used to initialize individual database
engines on each JADE node; for example, to enable roll-forward recovery in the event of media failure. (By default, archival recovery is not enabled.)

The physical database layer or database engine of the JADE Object Manager can use the information stored in these sections to configure itself to meet defined performance or resource constraints, or user preferences.

**Note**  In a multiple-database environment, each database can have its own unique JADE initialization file and configuration parameters.

- [JadeSecurity] section, which contains information that enables the JADE authentication and encryption hooks and access to the JADE development environment. For details about JADE security, see "JADE Security", in Chapter 2 of the JADE Object Manager Guide.
- [JadeOdbc] and [JadeOdbcServer] sections, which contains parameters that apply to accessing a JADE database through the Open Database Connectivity (ODBC) interface provided by JADE.
- [JadePatchControlExtensions] section, which controls patch control extensions.
- [JadePrinting] section, which controls the meta file and print data formats.
- [JadeProfiler] section, which contains the name of the JADE profile statistics output file.
- [JadeHelp] section, which provides flexibility when using Adobe Acrobat Portable Document Format (PDF) files in a JADE environment.
- [JadeFontSubstitutions] section, which enables you to substitute preferred fonts on client nodes for those defined in an application. JADE applications written in a Windows environment rely on the Windows environment to provide specific common fonts.
- [DatabaseBackup] section, which contains parameters that store the settings specified in the Backup Database dialog.
- [DeltaOptions] section, which contains a parameter that specifies whether methods can be checked out to one delta only or to multiple deltas.
- [ExternalDb] section, which contains parameters that affect the operation of your external databases.
- [FaultHandling] section, which contains parameters that affect the operation of fault handling.
- [JadeAppServer], [JadeThinClient], and [environment-type] sections, which contain parameters that enable you to configure JADE thin client mode of operation.
- [NonGuiClient] section, which contains parameters that enable you to configure non-GUI clients.
- [JadeReorg] section, which contains parameters that enable you to configure the database reorganization process.
- [JadeReportWriter] section, which contains parameters that you can use as a diagnostic resource to help you to understand the query phase for a specific report and for exception handling.
- [ConnectionParams], [JadeRps], [JadeRpsManager], [JadeSDSAdmin], [RpsIgnoreMethodExceptions], and [SyncDbService] sections, which contains information used to initialize a Synchronized Database Service environment (SDE), including RPS nodes.
- [WebOptions] section, which enables you to specify options for your Web pages in HTML thin client applications.
- [WebSession] section, which enables you to specify options for your Web sessions.

For details, see "JADE Initialization File", in the JADE Initialization File Reference.
You can also:

- Set options to specify your user configuration in the JADE command line, for use when running a JADE application from outside the development environment. For details, see "JADE Configurations under Windows", in Chapter 1, and "Format of the Server URI String", earlier in this chapter.

- Specify your installation global preferences that apply to all JADE Browser and Painter windows for your JADE development environment work sessions in the installed JADE release. For details, see "Specifying Your Administration Options", later in this chapter.

- Specify options in the jadehttp.ini file for use when accessing JADE applications from the Internet Information Server (IIS). For details, see "Configuring JadeHttp for Remote Connections", later in this chapter.

- Specify options in the mod_jadehttp module, for use when accessing JADE applications from the Apache HTTP Server. For details, see "Configuring Apache for Remote Connections", later in this chapter.

Configuring the Internet Protocol Version

JADE supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6).

These transport protocols are used in the following areas of JADE.

- Remote Procedure Call (RPC), between database server and nodes
- Thin client, between the application server and presentation clients
- Synchronized Database Service (SDS), between SDS nodes
- Web, between the node and the Web server add-in (that is, jadehttp.dll or mod_jadehttp.so)
- User code in your application that uses the TcpIpConnection class
- ODBC, between the thin client and server

The following JADE initialization file parameter values refer to networking.

- [JadeServer] section and NetworkSpecification parameter
- [ConnectionParams] section NetworkSpecification and ServerNodeSpecifications parameters
- [JadeClient] section ProxyHost and ServerNodeSpecifications parameters
- [JadeAppServer] section AppServer parameter
- [JadeThinClient] section AppServer and SSLProxyHost parameters
- [JadeOdbcServer] section ListenerHostName and ListenPort parameters

The system administrator should review the following host name and IP address values of the following properties for potential changes, to enable the IPv6 feature. JADE developers who expect a deployed system to support the IPv6 feature should review their code that accesses these properties, as these properties can now return different values and break assumptions that you may have previously made.

The most-common change is that IP addresses will have a dot quad number format (for example, 192.168.254.) and be a maximum of 16 characters. For details, see "Limits and Defaults", later in this chapter.

- TcpIpConnection::localInterface
- TcpIpConnection::localIpAddress
- TcpIpConnection::remoteIpAddress
RPC, Database, and Node Configuration

This section covers the area of behavior and configuration between the database server (including SDS environments) and client nodes.

Configuring the Database Server

Configure the database server by specifying the following values for the `NetworkSpecification` parameter in the `[JadeServer]` section of the JADE initialization file, as follows.

```
NetworkSpecification<specification-number> = transport-type,enabled|disabled,listener-port[,interface]
```

The values for the `transport-type` variable relevant to this feature can be `TcpIP`, `TcpIPv4`, or `TcpIPv6`. The `transport-type` literal value is case-insensitive.

The `TcpIP` value is a synonym for `TcpIPv4`. `TcpIPv4` provides IP version 4 connections only. `TcpIPv6` provides IP version 6 connections only. If you want the server to support both network protocols, you must define a network specification for each protocol.

You can specify the `interface` value as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the specified `transport-type` value.

Configuring the Client Node

Configure the client node by:

- Specifying the following values for the `ServerNodeSpecifications` parameter in the `[JadeClient]` section of the JADE initialization file.
ServerNodeSpecifications = transport-type, remote-host, remote-port
[,]local-interface[,local-port]]

The values for transport-type variable relevant to this feature can be TcpiP, TcpiPV4, TcpiPV6, or TcpiPAny. The transport-type literal value is case-insensitive. The TcpiP value is a synonym for TcpiPAny. TcpiPV4 provides IP version 4 connections only. TcpiPV6 provides IP version 6 connections only. The TcpiPAny value supports IP version 6 or IP version 4 connections.

You can specify the remote-host and local-interface values as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the specified transport type value. If you specify a host name, all DNS-provided addresses will be attempted. For the TcpiPAny transport type, the client will first attempt to connect via IP version 6 and then IP version 4 protocol on the provided IP addresses. Each connection failure will be logged, and the next available combination tried.

- Using the server parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the server and path parameters in a command line and the ServerNodeSpecifications parameter in the [JadeClient] section of the JADE initialization file. For details, see "Format of the Server URI String", earlier in this chapter.

### Configuring SDS Connections

Configure SDS connections by specifying the following values for the NetworkSpecification and ServerNodeSpecifications parameters in the [ConnectionParams] section of the JADE initialization file.

```
NetworkSpecification<specification-number> = transport-type, enabled|disabled, listener-port[,interface]
ServerNodeSpecifications = transport-type, remote-host, remote-port[,local-interface [,local-port]]
```

Both the server side (specified in the NetworkSpecification parameter) and the client side (specified in the ServerNodeSpecifications parameter) follow the same behavior and rules documented for the database server and client node RPC configurations in the previous sections.

### Application Server and Thin Client Configuration

This section covers the area of behavior and configuration between the application server and the presentation client nodes (both Windows and Mobile variants).

### Configuring the Application Server

Configure the application server by specifying the following values for the AppServer parameter in the [JadeAppServer] section of the JADE initialization file, as follows.

```
AppServer = [transport-type,]interface
```

The values for the transport-type variable relevant to this feature can be TcpiP, TcpiPV4, TcpiPV6, or TcpiPAny. The transport-type literal value is case-insensitive.

If you do not specify a transport-type, the default value is TcpiP, which is a synonym for TcpiPV4. TcpiPV4 provides IP version 4 connections only. TcpiPV6 provides IP version 6 connections only. TcpiPAny supports IP version 6 or IP version 4 connections. Depending on your requirements, you can configure a single application server to use TcpiPAny, or you can set up two application servers so that one supports TcpiPV4 and the other TcpiPV6.

You can specify the interface value as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the selected transport-type value.
Because of the way that JADE parses the command line, enclose the parameter values in quote marks if you specify `ApplicationServer="transport-type,interface"` on the command line.

### Configuring a Presentation Client

Configure a presentation client by specifying the following values for the `AppServer` and `SSLProxyHost` parameters in the `[JadeThinClient]` section of the JADE initialization file, as follows.

```plaintext
AppServer = [transport-type,]interface
SSLProxyHost = interface
```

The values for the `transport-type` variable of the `AppServer` parameter relevant to this feature can be `TcpIP`, `TcpIPv4`, `TcpIPv6`, or `TcpIPAny`. The `transport-type` literal value is case-insensitive.

If you do not specify a `transport-type`, the default value is `TcpIP`, which is a synonym for `TcpIPAny`. `TcpIPv4` provides IP version 4 connections only. `TcpIPv6` provides IP version 6 connections only. `TcpIPAny` supports IP version 6 or IP version 4 connections.

You can specify the `interface` value as a host name or an IP address. If you use an IP address, the address must be in an appropriate format for the selected `transport-type` value. If you specify a host name, all DNS-provided addresses will be attempted. For the `TcpIPAny` transport type, the presentation client will first attempt to connect via IP version 6 and then IP version 4 protocol on the provided IP addresses. Each connection failure will be logged, and the next available combination tried.

Because of the way that JADE parses the command line, enclose the parameter values in quote marks if you specify `ApplicationServer="transport-type,interface"` on the command line.

If SSL has been enabled and the presentation client needs to connect to the application server via a proxy host, the presentation client will attempt to connect to the proxy server using the same protocol (`transport-type`) as that specified in the `ApplicationServer` parameter configuration. How the proxy host handles TCP/IP version 6 is dependent on the proxy host implementation.

The areas to check and to test with the proxy host are as follows.

1. Will it accept an IP version 6 connection?
2. What protocol will it use between the proxy and the application server?

### TcplpConnection Class

This section covers the area of behavior and configuration in using JADE RootSchema-based networking classes.

The `TcplpConnection` class provides the virtual `protocolFamily` property, which is an `Integer` value. The values that can be assigned to this property are the new `TcplpConnection` class constants listed in the following table.

<table>
<thead>
<tr>
<th>Class Constant</th>
<th>Integer Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtocolFamilyTcpIPv4</td>
<td>0</td>
</tr>
<tr>
<td>ProtocolFamilyTcpIPv6</td>
<td>1</td>
</tr>
<tr>
<td>ProtocolFamilyTcpIPAny</td>
<td>-1</td>
</tr>
</tbody>
</table>

If you do not change your code, your existing code runs using TCP/IP version 4 only.

**Note** The `JadeTcplpProxy` class works only with the TCP/IP version 4 protocol.
Other User Networking Areas

This section contains other areas of your network that may be affected by this feature.

The TcpConnection parameter in the jadehttp.ini file specifies a host or IP address to which to connect.

If you specify a host name, all DNS-provided address will be attempted. Both the TcpIPv6 and TcpIPv4 protocols will be attempted on the provided IP addresses. Each connection failure will be logged, and the next available combination tried.

Limits and Defaults

Your user code must allow for longer network addresses and host names.

A fully qualified hostname can be up to 255 characters, with each label being up to 63 characters (for details, see the RFC2181 clarification to the DNS specification). However, the underlying C++ code allows for 1,025, so allow for a host name of 1,024, to make your code safe. A fully defined IPv6 address, including port number, can be up to 65 characters.

The following is taken from a Microsoft header file, and agrees with other Internet descriptions.

```
// The totals are derived from the following data:
// 15: Ipv4 address
// 45: Ipv6 address including embedded Ipv4 address
// 11: Scope Id
// 2: Brackets around Ipv6 address when port is present
// 6: Port (including colon)
// 1: Terminating null byte
#
#define INET_ADDRSTRLEN 22
#define INET6_ADDRSTRLEN 65
```

Configuring JadeHttp for Remote Connections

The jadehttp library module can handle multiple TCP/IP address connections to the same application, which allows the Web server to connect to multiple copies of the same JADE application on multiple hosts. For details, see "[application-name] Section", later in this chapter. To handle the transfer of large files from a user’s browser to JADE for JadeInternetTCPConnection class connections, data is read in small pieces and written into a temporary file as it goes.

If the Firewall parameter in the [Jadehttp Files] section of the JadeHttp initialization file or the Firewall configuration directive in the JADE mod_jadehttp module and the Firewall parameter in the [WebOptions] section of the JADE initialization file are set to true, the contents of the temporary file is sent (in pieces) to JADE when the entire data stream has been received. JADE then reads those pieces and writes the file contents as it goes (therefore avoiding large memory allocations). This is transparent if the Firewall parameter or directive is set to false or if it is set to true and one of the following applies.

- The FileTransferDirectory parameter in the [Jadehttp Files] section of the jadehttp.ini file or the FileTransferDirectory configuration directive in the JADE mod_jadehttp module is set to default or to a valid directory (used to hold temporary file contents as the data is read from the Internet).

- The standard readPipeCallback method of the JadeInternetTCPConnection class is used to read the data.
The **jadehttp.ini** file is located in the virtual directory (that is, the IIS server location). Each virtual directory has its own **jadehttp.ini** file. (For details about the paths, see "[JadeHttp Files] Section", later in this chapter.)

The JadeHttp initialization file or JADE mod_jadehttp module is read-only when the first request for an application is received. If you update a parameter value in this file, you must therefore stop the Internet service and then restart it.

### Firewall for the JADE Internet Environment

As the **JadeHttp** library or the JADE mod_jadehttp module can communicate via a TCP/IP connection to a different workstation running the JADE application, you can ensure that the JADE environment is entirely behind the firewall. In the default implementation of the JADE Internet environment, the JADE application writes image files directly into the virtual directory for subsequent Web browser access.

Files transferred by a Web browser to the **JadeHttp** library are written into a directory for access by the JADE application.

Use of a firewall requires that there are no network directory services running between the Web browser and JADE environments.

The only available links are TCP/IP connections. JADE applications do not write files directly into the virtual directory outside the firewall. Instead, the files must be passed via the TCP/IP connection to the **JadeHttp** library, which creates the file in the virtual directory. Similarly, files arriving at the **JadeHttp** library from a Web browser are passed to the JADE application via the TCP/IP connection.

The JADE application writes these files into the JADE Web application directory.

### Files Created by the JADE Application

Use the **Firewall** parameter in the [JadeHttp Files] section of the JadeHttp initialization file or the **Firewall** configuration directive in the JADE mod_jadehttp module to control whether JADE uses this facility for its automatic HyperText Markup Language (HTML) generation. When this parameter or directive is set to the default value of **false**, files are written directly to the virtual directory by JADE.

When you set this parameter or directive to **true** and the **Firewall** parameter in the [WebOptions] section of the JADE initialization file is also set to **true**, files are transferred to the **JadeHttp** library via the TCP/IP connection before being passed to the virtual directory.

The **JadeWebServiceProvider** and **WebSession** classes provide a **createVirtualDirectoryFile** method, which enables you to pass files created by a JADE application to the **JadeHttp** library.

The **createVirtualDirectoryFile** method has the following signature:

```java
createVirtualDirectoryFile(fileName: String,
                           fileContents: Binary,
                           retain: Boolean): Integer;
```
The `createVirtualDirectoryFile` method parameters are listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileName</td>
<td>Name of the file to be created in the virtual directory</td>
</tr>
<tr>
<td>fileContents</td>
<td>Binary holding the file contents</td>
</tr>
<tr>
<td>retain</td>
<td>Creates read-only files when set to <code>true</code> or standard files when set to <code>false</code></td>
</tr>
</tbody>
</table>

The `JadeHttp` library creates the specified file in the directory (the virtual directory visible to Web browsers) in which the library is running. This method returns zero (0) if the method successfully formats a request to the `JadeHttp` library or it returns the non-zero Windows error code indicating the failure to create the file.

The image files must be passed before the final reply to the Web request is returned.

This process is transparent to you if your application is using the standard JADE generated Internet facility. However, if your application logic does additional file generation of its own, you must call the `JadeWebServiceProvider` or `WebSession` class `createVirtualDirectoryFile` method.

For Web services applications, the virtual directory parameter must be specified and the `fileName` parameter contains only the file name. When an application transfers files to the `jadehttp` library by using the `createVirtualDirectoryFile` method for non-Web services applications, the directory that is used is determined as follows.

- If the file name specified in the `fileName` parameter includes a directory, the file is written into that directory.
- If the file name does not include a directory and the application’s section of the `jadehttp.ini` file or the JADE `mod_jadehttp` module contains a directory value for the `VirtualDirectory` parameter or `PhysicalDirectory` directive, respectively, the files are written into that directory.

If neither the file name nor the `[application-name]` section of the `jadehttp.ini` file or the directive `ApplicationType` in the JADE `mod_jadehttp` module contains a directory, the files are written into the same directory as the `jadehttp` library module. (This enables you to control directory permissions so that the directory containing the `jadehttp` library module does not need to have read and write access.)

Use the `JadeWebServiceProvider` or `WebSession` class `isVDFilePresent` method to return whether the requested file specified in the `fileName` parameter of the method is present on the Web server side of the firewall when using the JADE Web interface via the `jadehttp` library file. This method sends a message to the `jadehttp` library file to perform this action. If the specified file name does not have a directory part, the current virtual directory defined in the `VirtualDirectory` parameter in the `JadeHttp.ini` file or the `PhysicalDirectory` directive in the JADE `mod_jadehttp` module for the application is used. The file specified in the `fileName` parameter of the `isVDFilePresent` method is used if the file name has a directory part.

Call the `JadeWebServiceProvider` or `WebSession` class `deleteVirtualDirectoryFile` method to delete files that are in the virtual directory (that is, the directory specified by the value of the `VirtualDirectory` parameter in the `JadeHttp.ini` file or the `PhysicalDirectory` directive in the JADE `mod_jadehttp` module).

The `deleteVirtualDirectoryFile` method parameters are listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileName</td>
<td>Name of the file to be deleted from the virtual directory</td>
</tr>
<tr>
<td>deleteIfReadOnly</td>
<td>Deletes files marked as read-only when set to <code>true</code></td>
</tr>
</tbody>
</table>

This method returns zero (0) if the file deletion is successful or it returns a non-zero error code if the deletion fails.
Notes If your applications are not using the standard JADE generated Internet facility, you need to set the Firewall parameter in the [Jadehttp Files] section of the jadehttp.ini file or the Firewall configuration directive in the JADE mod_jadehttp module and the [WebOptions] section of the JADE initialization file to true and call the JadeWebServiceProvider or WebSession class createVirtualDirectoryFile method only if you require firewall separation. If you do not require firewall separation, JADE creates image files directly into the virtual directory and bypasses the JadeHttp library.

The file cleanup process that is started when the jadehttp.ini file Firewall parameter or the JADE mod_jadehttp module Firewall configuration directive is set to true deletes only files that are not read-only and which are of type .jpg, .png, or .gif. You should therefore make all other files in this directory that you want to retain read-only, by setting the retain parameter to true.

Files Transferred from the Web Browser

Use the Firewall parameter in the [Jadehttp Files] section of the jadehttp.ini file or the Firewall configuration directive in the JADE mod_jadehttp module to control whether firewall separation is used when transferring files to JADE from Web browsers. When the Firewall parameter or directive is set to false, the default mechanism is used.

When you set the Firewall parameter or directive to true and the Firewall parameter in the [WebOptions] section of the JADE initialization file is also set to true, files are transferred to the JADE application over the TCP/IP connection and each file transferred from a Web browser is sent to the JADE application as a separate message. Your JADE application must distinguish this message from a normal Web browser request and store the transferred file.

The JadeHttp library reads data in small pieces and writes it into a temporary file as it goes. The contents of the temporary file are sent (in pieces) when the entire data stream is received. JADE then reads those pieces and writes the file contents as it goes, which avoids large memory allocations. For details, see "Configuring JadeHttp for Remote Connections", earlier in this section.

For details about the location of files, see "Controlling the Location of Files Uploaded via a Web Application", later in this chapter.

The message sent through the firewall is formatted as follows.

<table>
<thead>
<tr>
<th>File Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>TRF, followed by a null</td>
</tr>
<tr>
<td>File name</td>
<td>Name of the file that is to be created, followed by a null</td>
</tr>
<tr>
<td>File contents</td>
<td>Binary image</td>
</tr>
</tbody>
</table>

The handling of this message is placed in the Connection class readPipeCallback method before the rest of the general processing. (Note that no reply should be sent for this message.) When the JADE application receives the Web browser request, it stores the file in the specified Web application directory.

Note There is no cleanup method invoked if your user logic does not remove the files. Your application should delete these files when processing on them is complete.

When a file is transferred to JADE from a Web browser by using the default connection mechanism, the text for the text box that generated the transfer is changed by the JadeHttp library to:

<original-file-name>;<temporary-file-name-and-path>
If you have set the Firewall parameter in the [Jadehttp Files] section in the jadehttp.ini file or the Firewall configuration directive in the JADE mod_jadehttp module and the Firewall parameter in the [WebOptions] section of the JADE initialization file to true, the text is set to:

<original-file-name>;<temporary-file-name>

Your application logic that accesses the file must append the Web application directory to the temporary file name to form the actual path. (To get the name of the Web application directory, call the Application class webApplicationDirectory method, which returns a string containing the directory name.)

Internal Housekeeping of the Virtual Directory

When JADE sends a file to the JadeHttp library, any file that is not to be retained is created as a standard file of type .jpg, .png, or .gif when the jadehttp.ini file Firewall parameter or the JADE mod_jadehttp module Firewall configuration directive, the Firewall parameter in the [WebOptions] section of the JADE initialization file, and the retain parameter in the JadeWebServiceProvider or WebSession class createVirtualDirectoryFile method are set to true. Files that are to be retained are created as read-only files.

Call the JadeWebServiceProvider or WebSession class deleteVirtualDirectoryFile method to delete virtual directory files (that is, the directory specified by the value of the VirtualDirectory parameter in the jadehttp.ini file or the PhysicalDirectory configuration directive in the JADE mod_jadehttp module).

By default, JADE initiates a thread to clean up files in the virtual directory when JadeHttp is started up if the Firewall parameter in the [Jadehttp Files] section of the JadeHttp initialization file or the Firewall configuration directive in the JADE mod_jadehttp module is set to true. Every hour this thread removes any standard files of type .jpg, .png, or .gif that are more than 12 hours old, by default. (As the JadeHttp library has no life of its own until a message is received, this thread is initiated only after the first Web browser message is received.)

Alternatively, set the value of the PurgeDirectoryRule parameter in the [application-name] section of the jadehttp.ini file or the PurgeDirectoryRule configuration directive in the JADE mod_jadehttp module to AllWritable if you want to purge all files in the virtual directory that are not read-only. The default value of Default for this parameter removes any standard files of type .jpg, .png, or .gif that are more than 12 hours old. If you want to specify the length of time since a read-only file was last modified before it is purged from the defined virtual directory for that application, specify the required value in the PurgeFileAge parameter.

Controlling the Location of Files Uploaded via a Web Application

If your JADE application accepts file input in text boxes on a Web page (by using the webInputType property of a TextBox control with the Web_InputType_File setting to upload a file from a Web session), you can use the FileTransferDirectory parameter in the [Jadehttp Files] section of the jadehttp.ini file to specify the directory to which the file is written.

This parameter controls the directory in which any files transferred using the HTML InputType=file option are placed. (This parameter applies only when the Firewall parameter in the [Jadehttp Files] section of the jadehttp.ini file or the Firewall configuration directive in the JADE mod_jadehttp module is set to false.) By default, any transferred files are placed in the same directory as the jadehttp library file or the JADE mod_jadehttp module.

The format of the text property value of the TextBox control is as follows.

<source-file-name>;<destination-file-path><destination-file-name>

The source-file-name value is the name (excluding the path) of the originating file on the client workstation from which the file was loaded (that is, the workstation that is running the Web browser). A semicolon character (;) separates this and the destination-file-path and destination-file-name values, which are the full path to which the file is written (uploaded) and the name of that file; for example:

UsefulStuff.doc;\jade\bin\txf188.tmp
In this example, accessing the txf188.tmp file in the specified directory opens a document file that contains the information in the UsefulStuff.doc file uploaded via the Web browser.

To provide increased security for applications running in HTML thin client mode, a text file input by using the TextBox class webInputType property with a value of Web_InputType_File must be processed in the event that resulted in the file upload occurring (for example, in the click event method of a Completed button). In addition, set the value of the Trace parameter in the [Jadehttp Logging] section of the jadephp.ini file to false to suppress logging completely. When the value of this parameter is set to true, logged messages acknowledge only that a message has been received or sent, because it is not possible to distinguish what is sensitive data and what is not. Setting this parameter to true does not include any of the text sent or received from the client, as this text could contain personal information, passwords, credit card details, and so on.

**Caution** To prevent malicious use of files uploaded to Web-enabled applications, the files are removed as soon as the event that resulted in their upload has completed. You should therefore process the file immediately or move it into a directory that is not available from the Web if you require that file for future processing.

### [application-name] Section

To ensure that an application specified in a Web browser cannot cause an attachment to a non-JADE environment, you must specify the name of the Web application to which users can connect, by defining a unique [application-name] section for each Web-enabled JADE application.

IIS attempts to attach only to the applications whose names are specified as section names in the jadephp.ini file.

**Note** The jadephp library module can handle connections for up to 50 different hosts (that is, you can define up to 50 unique TcpPort[n], MinInUse[n], MaxInUse[n], TcpConnection[n], CloseDelay[n], and ConnectionGroup[n] parameters within each [application-name] section).

The [application-name] section of the jadephp.ini file contains parameters that define TCP/IP connections that are used instead of the default named pipe connections. For more details and the application requirements for TCP/IP connections, see "Connecting to JADE Applications from Internet Information Server (IIS)", earlier in this chapter.

To specify that TCP/IP connections are used to connect to JADE release 7.0 applications from IIS, you must define a section with a unique name within the jadephp.ini file for each application, with the ApplicationType parameter and unique multiple TcpPort[n], MinInUse[n], MaxInUse[n], TcpConnection[n], CloseDelay[n], and ConnectionGroup[n] parameters within each [application-name] section. Set the ApplicationType parameter in each [application-name] section to WebEnabledForms, WebServices, or HtmlDocuments.

The [application-name] section can contain the following parameters.

**ApplicationType**

*Value Type* String

*Default* Not specified

*Purpose*

The ApplicationType parameter, which is required in all [application-name] sections of the jadephp.ini file, specifies the type of application that will be supported and how the interfaces to JADE function.
The valid parameter values are listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebEnabledForms</td>
<td>JADE 6.2 Web-enabled forms. This application type results in all requests from a user being directed to the same JADE application copy during the user’s session. JADEHttp no longer inserts the tags into the HTML output; the JADE software now performs this function. The JADE side uses the multi-worker TCP facility to perform routing and queuing.</td>
<td></td>
</tr>
<tr>
<td>WebServices</td>
<td>JADE 6.2 Web service operation, where user requests are sent to any available connection. The JADE side uses the multi-worker TCP facility to perform routing and queuing.</td>
<td></td>
</tr>
<tr>
<td>HtmlDocuments</td>
<td>JADE 6.2 HTML documents operation where user requests are sent to any available connection. The JADE side uses the multi-worker TCP facility to perform routing and queuing.</td>
<td></td>
</tr>
</tbody>
</table>

**CloseDelay[n]**

- **Value Type**: Integer (number-of-seconds)
- **Default**: Not specified
- **Purpose**

When the current number of in-use connections for the TCP/IP specified in the TcpPort[n] parameter exceeds the value of the MinInUse[n] parameter, the CloseDelay[n] parameter indicates how many seconds before the extra connections are closed after they become inactive.

This value is used only when the value of the ApplicationType parameter is set to WebEnabledForms, WebServices, or HtmlDocuments.

**ConnectionGroup[n]**

- **Value Type**: String
- **Default**: Not specified
- **Purpose**

The ConnectionGroup[n] parameter specifies the name of the connection group associated with the value of the TcpPort[n] parameter (that is, the TCP/IP port to be used for the connection).

The connection group is the unique name that is used by applications of type WebEnabledForms to identify the TCP/IP port (the value of the TcpPort[n] parameter) to which a user request is to be sent.

**Note**

If you do not specify this parameter with a unique connection group name, this process does not function correctly.
**MaxInUse[n]**

**Value Type**  Integer

**Default**  Not specified

**Purpose**

The `MaxInUse[n]` parameter specifies the maximum number of connections that can be opened for the corresponding value of the `TcpPort[n]` parameter only when the value of the `ApplicationType` parameter is set to `WebEnabledForms`, `WebServices`, or `HtmlDocuments`. The `MaxInUse` parameter with no specified suffix is the limit for a `ConnectionGroup` associated with an entry with the `TcpConnection` parameter; it is *not* a global limit.

The `MaxInUse[n]` parameter defines the limit of outstanding requests for a single connection group; that is, the sum of queued and active requests. The total number of outstanding requests for a Web application is the sum of the `MaxInUse[n]` suffix values for all of the connection groups specified for that application.

**Note**

The value of the `MaxInUse` parameter is forced to be equal to or greater than the value of the `MinInUse` parameter.

**MaxMessageSize**

**Value Type**  Integer

**Default**  1000000 bytes

**Purpose**

The `MaxMessageSize` parameter specifies the maximum size allowed for a single-part Web message (that is, a single stream of data with minimal formatting). If the size of a single-part message exceeds the specified value of this parameter, the request is rejected and the following message is displayed to the user.

```
The size of the input data exceeds the limit permitted for this application and has been rejected.
```

**Note**

Multiple-part formatted Web messages are not limited in size (each message is made up of a series of sections that are specifically formatted).

**MessageTimeout**

**Value Type**  Integer (seconds)

**Default**  300

**Purpose**

The `MessageTimeout` parameter specifies the maximum number of seconds that the `JadeHttp.dll` library waits for a reply from a JADE system before sending a failure message to the requesting Web browser.

**Hints**

This parameter enables you to test the effects of unexpectedly long request processing without having to wait the default five minutes for each test.
When Web-enabled applications are all busy, you can control the time that a Web browser window shows no action before returning the *Service unavailable* response.

**MinInUse[n]**

**Value Type** Integer  
**Default** Not specified

**Purpose**

The `MinInUse[n]` parameter specifies the minimum number of connections that can be opened for the corresponding value of the `TcpPort[n]` parameter only when the value of the `ApplicationType` parameter is set to `WebEnabledForms`, `WebServices`, or `HtmlDocuments`.

The default minimum number of connections is 1.

**MinMessageSize**

**Value Type** Integer  
**Default** 10 bytes

**Purpose**

The `MinMessageSize` parameter specifies the minimum size allowed for a Web message received from JADE using the `WebSession` class `reply` method to send HTML string Web requests back to the client node.

The minimum value is 1 byte, the maximum value 1024 bytes, and the default value 10 bytes.

This value is read the first time the application specified in the `[application-name]` section is accessed after `jadehttp.dll` has been loaded by Internet Information Server (IIS).

**PurgeDirectoryRule**

**Value Type** String  
**Default** Default

**Purpose**

The `PurgeDirectoryRule` parameter specifies when files that are not read-only in the virtual directory are purged. The default value of `Default` for this parameter removes any standard files of type `.jpg`, `.png`, or `.gif` that are more than 12 hours old.

Specify a value of `AllWritable` if you want to purge all files in the virtual directory that are not read-only.

**PurgeFileAge**

**Value Type** `time-unit[D|H|M]`  
**Default** 12H

**Purpose**

The `PurgeFileAge` parameter specifies the length of time since a read-only file was last modified before it is purged from the defined virtual directory for that application; for example, `PurgeFileAge=7M`.
The parameter value is a time-unit numeric, optionally followed by a modifier D (days), H (hours), or M (minutes). The D multiplier multiplies by 86,400; the H multiplier multiplies by 3,600; the M multiplier multiplies by 60; and all other time-unit values are treated as seconds.

Specify zero (0) if you want to turn off the purging of the physical directory.

The minimum time-unit value that you can specify is 30 seconds; the maximum value is 365D (days). Values outside this range are forced to their respective limits.

This parameter applies only if the VirtualDirectory parameter has a specified value, and only files that match the PurgeDirectoryRule parameter criteria are considered (that is, all non-read-only files or by default, standard files of type .jpg, .png, or .gif that are more than the specified age).

### TcpConnection[n]

<table>
<thead>
<tr>
<th>Value Type</th>
<th>Integer (tcp-address)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Not specified</td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
</tr>
</tbody>
</table>

The TcpConnection[n] parameter specifies the unique valid TCP/IP address of the port to connect to on the server (remote) node, to enable Web client nodes to connect to the specified server node across the network through a TCP/IP connection.

### TcpPort[n]

<table>
<thead>
<tr>
<th>Value Type</th>
<th>Integer (tcp-port-number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Not specified</td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
</tr>
</tbody>
</table>

The TcpPort[n] parameter specifies the TCP/IP port number to be used for the connection specified in the corresponding TcpConnection[n] parameter for applications of type WebEnabledForms, WebServices, and HtmlDocuments.

### VirtualDirectory

<table>
<thead>
<tr>
<th>Value Type</th>
<th>String (virtual-directory-name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Not specified</td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
</tr>
</tbody>
</table>

The VirtualDirectory parameter enables you to specify a string containing the name of the virtual directory for the HTML-enabled application.

### Hints

The value specified for this parameter is used if the fileName parameter value of the JadeWebServiceProvider or WebSession class createVirtualDirectoryFile method does not include a directory. If neither this parameter nor the file name in the createVirtualDirectoryFile method contains a directory, the files are written into the same directory as the jadehttp library module.

This parameter enables you to control virtual directory permissions so that the directory containing the jadehttp library module does not need to have read and write access.
For details about retaining files passed to the jadehttp library, see "Firewall for the JADE Internet Environment", earlier in this chapter. For details about returning whether a specified file is present on the IIS side of the firewall when using the JADE HTML thin client interface via the jadehttp library file, see "Image Files Created by the JADE Application", earlier in this chapter.

**Sample [application-name] Section**

The following are examples of an [application-name] section in the jadehttp.ini file for four of the application types specified by the ApplicationType parameter.

```ini
[Company]
ApplicationType = WebEnabledForms
TcpConnection = Host1a
TcpPort = 20000
ConnectionGroup = CompanyForms
MinInUse = 1
MaxInUse = 5
CloseDelay = 600

[WebCompany]
ApplicationType = WebServices
TcpConnection = Host1a
TcpPort = 21000
MinInUse = 5
MaxInUse = 15
CloseDelay = 600
TcpConnection2 = Host2a
TcpPort2 = 21001
MinInUse2 = 5
MaxInUse2 = 15
CloseDelay2 = 600

[HtmlCompany]
ApplicationType = HtmlDocuments
TcpConnection = Host1a
TcpPort = 22000
MinInUse = 2
MaxInUse = 10
CloseDelay = 600
TcpConnection2 = Host2a
TcpPort2 = 22001
MinInUse2 = 2
MaxInUse2 = 10
CloseDelay2 = 600
```

**[Jadehttp Files] Section**

The [Jadehttp Files] section of the jadehttp.ini file contains parameters that control firewall security and the location of files uploaded from a Web-enabled application.

**Caution** To prevent malicious use of files uploaded to Web-enabled applications, the files are removed as soon as the event that resulted in their upload has completed. You should therefore process the file immediately or move it into a directory that is not available from the Web if you require that file for future processing.
The **jadehttp** library derives a path for the **jadehttp.ini**, **jadehttp.log**, and transfer files, ensuring that they are completely secure. The directories are derived from the **jadehttp** library directory for these files, as follows.

1. **JadeHttp** attempts to create the required directories if they do not exist if security permits this. These directories, based on the example `c:\jade\bin\jadehttp.dll` file, are listed in the following table.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>dll-path_dll-name\ini\</td>
<td>For the initialization file and the default error dll-name.htm file</td>
<td><code>c:\jade\bin\jadehttp\ini\jadehttp.ini</code></td>
</tr>
<tr>
<td>dll-path_dll-name\logs\</td>
<td>For all dll-name logs</td>
<td><code>c:\jade\bin\jadehttp\logs\jadehttp.log</code></td>
</tr>
<tr>
<td>dll-path_dll-name\transfer\</td>
<td>For the location of any temporary files created during file transfer</td>
<td><code>c:\jade\bin\jadehttp\transfer\</code></td>
</tr>
</tbody>
</table>

2. If the initialization file option to specify the log path is set, the logs directory is not created or used.
3. If the initialization file option to specify the transfer path is set, the transfer directory is not created or used.
4. As the dynamic link library can be renamed, its name is included. The library path therefore does not need to contain any other files, and users cannot access any of the files listed in the previous table.

For details about implementing a TCP/IP connection from the IIS to JADE applications, see "Connecting to JADE Applications from Internet Information Server (IIS)" and "Configuring JadeHttp for Remote Connections", earlier in this chapter.

The [Jadehttp Files] section can contain the following parameters.

**FileTransferDirectory**

<table>
<thead>
<tr>
<th>Value Type</th>
<th>String (disk-path)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Not specified</td>
</tr>
</tbody>
</table>

**Purpose**

The **FileTransferDirectory** parameter specifies the directory to which files are uploaded during Web sessions from JADE applications that accept file input in text boxes on Web pages (by using the webInputType property of a TextBox control with the Web_InputType/File setting to upload a file from a Web session).

**Note** This parameter applies only when the **Firewall** parameter in the [Jadehttp Files] section of the jadehttp.ini file is set to false or it is set to true and the **FileTransferDirectory** parameter is set to default or to a valid directory. (For details, see "Configuring JadeHttp for Remote Connections", earlier in this section.)

This parameter controls the directory in which any files transferred by using the HTML InputType=file option are placed. (The file transfer directory must be a valid directory that is relative to the virtual directory, or IIS server.)

If you define a value for the **FileTransferDirectory** parameter, the transfer directory in the same directory as the jadehttp library is neither created nor used. The file transfer directory path cannot be greater than 260 characters. For more details, including the format of the text property value of the TextBox control, see "Controlling the Location of Files Uploaded via a Web Application", earlier in this chapter.
Chapter 3 Configuring JADE

Firewall

Value Type: Boolean
Default: False

Purpose

The Firewall parameter controls whether firewall separation is used when transferring files to JADE from Web browsers. When this parameter is set to the default value of false, files are written to the transfer directory directly by the JadeHttp library.

The firewall must be enabled at both ends of the connection (that is, if you are uploading images from another machine for a JADE Web-enabled form over a TCP/IP connection, the Firewall parameter in the [Jadehttp Files] section of the JadeHttp initialization file and the Firewall parameter in the [WebOptions] section of the JADE initialization file must both be set to true).

For details about the handling of data from the user’s browser through to JADE, see "Configuring JadeHttp for Remote Connections", earlier in this section.

Set this parameter to true if you want files transferred to the JADE application over the TCP/IP connection. Each file transferred from a Web browser is sent to the JADE application as a separate message. Your JADE application must distinguish this message from a normal Web browser request and store the transferred file. For more details, including the format of messages sent through the firewall, see "Firewall for the JADE Internet Environment", earlier in this chapter.

Sample [Jadehttp Files] Section

The following is an example of a [Jadehttp Files] section in the jadehttp.ini file.

[Jadehttp Files]
FileTransferDirectory = c:\jade\bin_jadehttp\transfer
Firewall = true

[Jadehttp Logging] Section

The [Jadehttp Logging] section of the jadehttp.ini file controls logging by the jadehttp library file. For details about the messages that are logged when tracing is set, see "Message Logging", in the JADE Web Application Guide.

The [Jadehttp Logging] section can contain the following parameters.

Trace

Value Type: Boolean
Default: False

Purpose

The Trace parameter controls jadehttp library file logging of message meta data. When this parameter is set to the default value of false, logging is suppressed completely.

When the value of this parameter is set to true, logged messages acknowledge only that a message has been received or sent, because it is not possible to distinguish what is sensitive data and what is not. Setting this parameter to true does not include any of the text sent or received from the client, as this text could contain personal information, passwords, credit card details, and so on.

Note: You cannot use this parameter to inspect and debug data passing through the jadehttp library.
Installation and Configuration Guide

Chapter 3 Configuring JADE

**TraceFile**

**Value Type** String (file-path)

**Default** Default

**Purpose**

The TraceFile parameter specifies an optional path and a file name that overrides the default jadehttp.log file name so that log files can be placed in other directories.

If you define a value for the TraceFile parameter, the log file in the same directory as the jadehttp library is neither created nor used. The file transfer directory path cannot be greater than 260 characters.

**Sample [Jadehttp Logging] Section**

The following is an example of a [Jadehttp Logging] section in the jadehttp.ini file.

```
[Jadehttp Logging]
Trace = true
TraceFile = c:\jade\bin_jadehttp\logs\jadehttp.log
```

**Configuring Apache for Remote Connections**

TCP/IP connections only are used for connections to JADE applications running from the Apache HTTP Server. This section covers the following topics.

- Apache Configuration Directives
- Apache Configuration Examples
- Apache Considerations

For details about installing the mod_jadehttp library module supplied by JADE for the Apache HTTP Server and the MPMs that it implements, see "Connecting to JADE Applications from an Apache HTTP Server", earlier in this chapter.

**Apache Configuration Directives**

The configuration directives in this section are available to configure the JADE mod_jadehttp module. It uses the standard Apache configuration syntax, rather than implementing configuration via an initialization file.

Set the value of the JadeHttp_Trace directive to true to ensure the security of your data.

The concept and intent of these directives is the same as the jadehttp.ini file parameters, but the syntax differs. For details about the jadehttp.ini file parameters, see "Configuring JadeHttp for Remote Connections", earlier in this chapter.

Most of the directives specific to mod_jadehttp have the same meaning as their IIS counterparts (described in "Controlling the Location of Files Uploaded via a Web Application" under "Configuring JadeHttp for Remote Connections", earlier in this chapter). The JADE mod_jadehttp module configuration directives are as follows.

- SetHandler
- ApplicationType
- Application
Chapter 3 Configuring JADE

- FaultDocument
- FileTransferDirectory
- Firewall
- JadeHttp_Trace
- JadeServer
- JadeTimeout
- LocalInterface
- MaxMessageSize
- MinMessageSize
- PhysicalDirectory
- PurgeDirectoryFrequency
- PurgeDirectoryRule
- PurgeFileAge
- TcpConnection[n]
- VirtualDirectory

For details, see the following subsections.

To specify that TCP/IP connections are used to connect to JADE applications from Apache, you must define a <Location> directive within jadehttp.conf for each application, with an ApplicationType directive and unique multiple TcpConnection[n] directives between each Application directive.

The following mod_jadehttp directives apply between the core Apache <Location> and </Location> directives. Apache processes the <Location> directives in the order in which they are found in the configuration file, following any <Files> and <Directory> sections and after .htaccess files have been read and processed.

```
<Location /jade>
    SetHandler jadehttp-handler
    ApplicationType WebServices
    Application WebApplication
    TcpConnection PC1 21000 5 15 300
</Location>
```

This allows mod_jadehttp directives to be placed in higher locations and their values to become the defaults for lower locations unless you explicitly override them.

In the following example, the handler and FaultDocument directive specified in <Location /jade> also apply to <Location /jade/subloc>.

```
<Location /jade>
    SetHandler jadehttp-handler
    FaultDocument PostTooLarge "http://server/ohdarn.html"
</Location>
<Location /jade/subloc>
    ApplicationType WebEnabledForms
    Application WebApplication
```
SetHandler

The standard SetHandler Apache directive, whose characteristics are listed in the following table, selects the mod_jadehttp module to handle this location.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Forces all matching files to be processed by a handler</td>
</tr>
<tr>
<td>Syntax</td>
<td>SetHandler handler-name</td>
</tr>
<tr>
<td>Context</td>
<td>server config, virtual host, directory, .htaccess</td>
</tr>
<tr>
<td>Module</td>
<td>Core</td>
</tr>
<tr>
<td>Examples</td>
<td>SetHandler jadehttp-info and SetHandler jadehttp-handler</td>
</tr>
</tbody>
</table>

The jadehttp-handler, which is the primary handler inside the mod_jadehttp module, formats and passes data to JADE and returns the results to the Web browser. The jadehttp-info handler reports internal information about the configuration and current status of the mod_jadehttp module back to the Web browser. This handler is similar to the standard Apache mod_info module that provides a comprehensive overview of the server configuration. Access to locations that use this handler should be restricted to trusted sites.

ApplicationType

The ApplicationType directive specifies the type of application that will be supported and how the interfaces to JADE function.

**Note** You must define this directive before the TcpConnection[n] directive within each Application directive.

The application type characteristics are listed in the following table.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the JADE application type</td>
</tr>
<tr>
<td>Syntax</td>
<td>ApplicationType application-type</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>ApplicationType WebServices</td>
</tr>
</tbody>
</table>

The valid application type values are listed in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebEnabledForms</td>
<td>JADE 6.2 Web-enabled forms. This application type results in all requests from a user being directed to the same JADE application copy during the user’s session. JadeHttp no longer inserts the tags into the HTML output; the JADE software now performs this function.</td>
</tr>
</tbody>
</table>
The JADE side uses the multi-worker TCP facility to perform routing and queuing.

**Note** For the WebEnabledForms application type, you must specify the MinInUse, MaxInUse, CloseDelay, and ConnectionGroup values for the TcpConnection[n] directive.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServices</td>
<td>JADE 6.2 Web service operation, where user requests are sent to any available connection.</td>
</tr>
<tr>
<td>HtmlDocuments</td>
<td>JADE 6.2 HTML documents operation where user requests are sent to any available connection.</td>
</tr>
</tbody>
</table>

**Application**

The Application directive, whose characteristics are listed in the following table, specifies the name of the Web-enabled JADE application.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the JADE application to use</td>
</tr>
<tr>
<td>Syntax</td>
<td>Application application-name</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>Application ErewhonShop</td>
</tr>
</tbody>
</table>

This directive is similar to the functionality of the [application-name] section of the jadehttp library module for IIS, described earlier in this chapter.

**FaultDocument**

The FaultDocument directive, whose characteristics are listed in the following table, specifies what the mod_jadehttp module does when it encounters an error.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the URL or path name for redirection when an error occurs</td>
</tr>
<tr>
<td>Syntax</td>
<td>FaultDocument cause redirect-url</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler, jadehttp-info</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example 1</td>
<td>FaultDocument TcpConnectFailed <a href="http://secondary.example.com:80/page.html">http://secondary.example.com:80/page.html</a></td>
</tr>
<tr>
<td>Example 2</td>
<td>FaultDocument PostTooLarge bigpost.html</td>
</tr>
</tbody>
</table>
You can specify the *cause* value in numeric or text form. Valid values for the *cause* are listed in the following table.

<table>
<thead>
<tr>
<th>Numeric Name</th>
<th>Text Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>TcpConnectFailed</td>
</tr>
<tr>
<td>601</td>
<td>PostTooLarge</td>
</tr>
<tr>
<td>602</td>
<td>ServiceFailed</td>
</tr>
<tr>
<td>603</td>
<td>DataReadFailed</td>
</tr>
</tbody>
</table>

You must specify a valid URL or path name for the *redirect-url* value, which can be on the local Web server or on a remote Web server. If you do not specify a *redirect-url* value and an error occurs, an error page is raised internally in the mod_jadehttp module.

### FileTransferDirectory

The **FileTransferDirectory** directive, whose characteristics are listed in the following table, specifies the name of a local directory to use when files are uploaded from the Web browser to the JADE application.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the directory to use when files are uploaded from the Web browser</td>
</tr>
<tr>
<td>Syntax</td>
<td>FileTransferDirectory local-directory-name</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>FileTransferDirectory /var/tmp/appname</td>
</tr>
</tbody>
</table>

The **FileTransferDirectory** directive is similar to the **FileTransferDirectory** parameter in the [Jadehttp Files] section of the jadehttp.ini file for IIS servers, described earlier in this chapter.

As the mod_jadehttp module must put the transfer files somewhere before transferring them to JADE, this directive specifies the directory to which the file or files are written.

### Firewall

The **Firewall** directive, whose characteristics are listed in the following table, specifies whether there is a firewall between the mod_jadehttp module and the JADE application.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Indicates if there is a firewall between mod_jadehttp and JADE</td>
</tr>
<tr>
<td>Syntax</td>
<td>Firewall on</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>Firewall on</td>
</tr>
</tbody>
</table>
The **Firewall** directive is similar to the **Firewall** parameter in the [JadeHttp Files] section of the jadehttp.ini file for IIS servers, described earlier in this chapter. If a firewall is present, file transfers between JADE and the Web server must be done by using the **mod_jadehttp** module.

The firewall must be enabled at both ends of the connection (that is, if you are uploading images from another machine for a JADE Web-enabled form over a TCP/IP connection, the **Firewall** configuration directive in the JADE **mod_jadehttp** module must be set to **on** and the **Firewall** parameter in the [WebOptions] section of the JADE initialization file must be set to **true**).

When transferred, the files will be placed in the directory specified in the **PhysicalDirectory** directive.

### JadeHttp_Trace

The **JadeHttp_Trace** directive, whose characteristics are listed in the following table, controls where any **mod_jadehttp** tracing is written.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
<td>Specifies the log file to which JADE trace messages are sent</td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td>JadeHttp_Trace bit-mask [file-name [size[K][M]]]</td>
</tr>
<tr>
<td></td>
<td>JadeHttp_Trace bit-mask &quot;</td>
</tr>
<tr>
<td><strong>Context</strong></td>
<td>Server config, virtual host, Location</td>
</tr>
<tr>
<td><strong>Module</strong></td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>JadeHttp_Trace 0x000000ff logs/jade%Y%m%d%H%M%S.log 10M</td>
</tr>
</tbody>
</table>
|                | JadeHttp_Trace 0x000000ff "/usr/sbin/rotatelogs2 -l 
|                | /var/logs/apache2/jadehttp_%Y%m%d%H%M.log 10M"

The **JadeHttp_Trace** directive is similar to the **Trace** and **TraceFile** parameters in the [JadeHttp Logging] section of the jadehttp.ini file, documented earlier in this chapter.

If this parameter is not specified, logging is suppressed completely. Specifying this parameter ensures the security of your data. When the value of this parameter is specified, logged messages acknowledge only that a message has been received or sent, because it is not possible to distinguish what is sensitive data and what is not. When this parameter is specified, messages logged to the jadehttp.log file do not include any of the text sent or received from the client, as this text could contain personal information, passwords, credit card details, and so on.

**Note**  You cannot use this parameter to inspect and debug data passing through the jadehttp library.

If the **file-name** is relative, it is created relative to the **ServerRoot** location. If you do not specify the **file-name**, it defaults to "logs/jadehttp%Y%m%d%H%M%S.log".

If you do not specify the **size**, it defaults to **5M** bytes. To disable **mod_jadehttp** from attempting to rotate log files, specify a **size** of zero (0). The **size** value can have a multiplier appended to it. The K multiplier multiplies by 1,024, the M multiplier multiplies by 1,048,576 bytes, and all other values are treated as bytes.

If the **file-name** is in a directory that an Apache child **httpd** process does not have permission to access, the rotation of logs is disabled in **mod_jadehttp**. The file name can include % modifiers, which specify values from the current local time (based on the Apache rotatelogs2 program, whose documentation is available at [http://httpd.apache.org/docs/2.2/programs/rotatelogs.html](http://httpd.apache.org/docs/2.2/programs/rotatelogs.html), for example), permitted values are AaBbcdHIjMmSUWwXxYyZ%.

If no % modifiers are present, ".%Y%m%d%H%M%S" is appended at the end of the file name (before the file extension, if it exists). This adds year, month, day, hour, minute, and second values to the file name.
A new log file is created if the file name changes (due to the changing time and selected modifier values) and if the file exceeds the specified size.

**Note** Multiple log files could be actively written to concurrently, because of the way that the Apache pre-fork MPM works.

Alternatively, you can use an external program to get log files rotated; for example, the one that is normally provided with the **rotatelogs2** Apache distribution. This works well when log files are placed in a directory where the Apache child process does not have permission to create files.

The use of an external program behaves the same way as the Apache piped logs feature, which is described in the Apache documentation. It involves specifying the rotate program and arguments inside double quote characters (**"**), with the first character of the string being the pipe (**|**) character.

**Note** The name of the **rotatelogs** program can differ, depending on your Apache distribution and operating system.

The bit set values for the **bit-mask** value are listed in the following table.

<table>
<thead>
<tr>
<th>Bit Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>When a redirect occurs</td>
</tr>
<tr>
<td>0x00000002</td>
<td>When a special command is issued</td>
</tr>
<tr>
<td>0x00000004</td>
<td>When file purges occur</td>
</tr>
<tr>
<td>0x00000008</td>
<td>When <strong>mod_jadehttp</strong> sends connection information to JADE</td>
</tr>
</tbody>
</table>

**JadeServer**

The **JadeServer** directive, whose characteristics are listed in the following table, is reserved for future implementation.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the JADE server configuration</td>
</tr>
<tr>
<td>Syntax</td>
<td><strong>JadeServer jade-version byte-order character-size</strong></td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td><strong>mod_jadehttp</strong></td>
</tr>
<tr>
<td>Default</td>
<td><strong>JadeServer 7.0 localEndian oneByte</strong></td>
</tr>
<tr>
<td>Example</td>
<td><strong>JadeServer 7 Intel utf-16</strong></td>
</tr>
</tbody>
</table>

This directive allows a single **mod_jadehttp** module to connect to JADE, regardless of the combination of Web server architecture and JADE server architecture, allowing **mod_jadehttp** to adapt to the installed release of the JADE server.

The **jade-version** value allows for the protocol to between JADE and the **mod_jadehttp** module to be changed. The version number can be in the range 1 through 4, with each number separated with numbers inclusive separated with the dot operator (**.**) notation; for example, **7.0.06.007**.
The byte-order value allows for binary numbers and wide characters to be converted to the native byte ordering of the JADE server, which may be different from the byte ordering of the Web server.

In the default example in the previous table, the localEndian value defaults to littleEndian when the mod_jadehttp module is running on Intel-based hosts and bigEndian when on PowerPC-based hosts. The entered value is not case-sensitive.

The primary names and synonyms for the byte-order value are listed in the following table.

<table>
<thead>
<tr>
<th>Primary Name</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>littleEndian</td>
<td>Intel</td>
</tr>
<tr>
<td>bigEndian</td>
<td>PowerPC</td>
</tr>
</tbody>
</table>

The character-size value indicates the size of a JADE character, which is ANSI or Unicode, depending on your JADE installation. If the character size is Unicode where the JADE server is running, the physical size of a JADE character can be one of 1, 2, or 4 bytes in size. This allows mod_jadehttp to send JADE characters of the correct size and type to the JADE server.

The primary names and synonyms for the entered value, which is not case-sensitive, are listed in the following table.

<table>
<thead>
<tr>
<th>Primary Name</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>oneByte</td>
<td>ANSI</td>
</tr>
<tr>
<td>twoByte</td>
<td>Unicode, Utf16, or Utf-16</td>
</tr>
<tr>
<td>fourByte</td>
<td>Utf32 or Utf-32</td>
</tr>
</tbody>
</table>

JadeTimeout

The JadeTimeout directive, whose characteristics are listed in the following table, specifies the maximum number of seconds that mod_jadehttp waits for a reply from a JADE system before sending a failure message to the requesting Web browser.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the maximum number of seconds to wait for a reply from JADE</td>
</tr>
<tr>
<td>Syntax</td>
<td>JadeTimeout number-of-seconds (the default value is 45 seconds)</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Default</td>
<td>JadeTimeout 300 (that is, 5 minutes)</td>
</tr>
<tr>
<td>Example</td>
<td>JadeTimeout 30</td>
</tr>
</tbody>
</table>

A default value of zero (0) indicates that there is not timeout. The maximum timeout value is 1800 (that is, 30 minutes).

Hints

This directive enables you to test the effects of unexpectedly long request processing without having to wait the default five minutes for each test.
When Web-enabled applications are all busy, you can control the time that a Web browser window shows no action before returning the *Service unavailable* response.

**LocalInterface**

The `LocalInterface` directive, whose characteristics are listed in the following table, specifies the local network interface and optionally the port number to use when connecting from the `mod_jadehttp` module to the JADE Web-enabled application.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the local network interface and optionally the port number</td>
</tr>
<tr>
<td>Syntax</td>
<td><code>LocalInterface local-network-interface-handler-name</code></td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td><code>jadehttp-handler</code></td>
</tr>
<tr>
<td>Module</td>
<td><code>mod_jadehttp</code></td>
</tr>
<tr>
<td>Default</td>
<td><code>LocalInterface 0.0.0.0</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>LocalInterface 10.1.1.1</code></td>
</tr>
</tbody>
</table>

If your Web server host has multiple network interface cards, you can specify which one `mod_jadehttp` uses when initiating connections to the Web-enabled JADE application.

The default value of `0.0.0.0` allows the operating system to choose the network interface card.

You would normally require this directive only when there are specific network security or routing issues to be addressed.

**Caution** Although you can also specify the local outgoing port number, you should take great care when doing so and only when you have a specific requirement. Specifying a `LocalInterface` port number limits you to a maximum of one TCP/IP connection to JADE for each Apache `<Location>` directive. In addition, if the Web server terminates abnormally and the port is not properly closed, it may take several minutes before this port becomes available again for use.

**MaxMessageSize**

The `MaxMessageSize` directive, whose characteristics are listed in the following table, specifies the largest size of a POST URL before an error is raised.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the maximum size of a POST string</td>
</tr>
<tr>
<td>Syntax</td>
<td>`MaxMessageSize size[M</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td><code>jadehttp-handler</code></td>
</tr>
<tr>
<td>Module</td>
<td><code>mod_jadehttp</code></td>
</tr>
<tr>
<td>Default</td>
<td><code>MaxMessageSize 1M</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>MaxMessageSize 50K</code></td>
</tr>
</tbody>
</table>
You can append a multiplier to the size value. The \textbf{M} multiplier multiplies by \textbf{1},\textbf{048},\textbf{756}, the \textbf{K} multiplier multiplies by \textbf{1},\textbf{024}, and all other values are treated as bytes.

The minimum value is zero (0) and the maximum value is \textbf{128M}.

**MinMessageSize**

The \textbf{MinMessageSize} directive, whose characteristics are listed in the following table, specifies the minimum size allowed for a Web message received from JADE using the \textbf{WebSession} class \texttt{reply} method to send HTML string Web requests back to the client node.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the minimum number of bytes allowed for a Web message from JADE</td>
</tr>
<tr>
<td>Syntax</td>
<td>MinMessageSize \textit{message-size-in-bytes}</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Default</td>
<td>MinMessageSize 10</td>
</tr>
<tr>
<td>Example</td>
<td>MinMessageSize 100</td>
</tr>
</tbody>
</table>

The minimum value is \textbf{1} byte and the maximum value is \textbf{1024} bytes.

This value is read once, when the Apache Web server starts.

**PhysicalDirectory**

The \textbf{PhysicalDirectory} directive specifies the name of a local directory to use when files are transferred from JADE to the Web server by using the \textbf{JadeWebServiceProvider} or \textbf{WebSession} class \texttt{createVirtualDirectoryFile} method without specifying a directory in the \texttt{fileName} parameter.

This directory is also used as the location for transferring files when the \textbf{Firewall} directive is set to on. In this situation, it behaves in a similar way to the \textbf{FileTransferDirectory} parameter in the \texttt{[Jadehttp Files]} section of the jadehttp.ini file for IIS servers, described earlier in this chapter.

The characteristics of the \textbf{PhysicalDirectory} directive are listed in the following table.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the directory to use as a physical directory</td>
</tr>
<tr>
<td>Syntax</td>
<td>PhysicalDirectory \texttt{local-directory}</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>PhysicalDirectory \texttt{/var/tmp/appname/}</td>
</tr>
</tbody>
</table>

\textbf{Note}  Apache directives require separators to be defined using a forward slash (/) character.
The following is an example of the use of the `PhysicalDirectory` directive.

```
<Directory /var/spool/www/jade/images/>
  order allow,deny
  allow from all
</Directory>

Alias /jade/images/ /var/spool/www/jade/images/

<location /jade/webform>
  SetHandler jadehttp_handler
  PhysicalDirectory /var/spool/www/jade/images/
</location>
```

In this example, the physical directory on disk `/var/spool/www/jade/images/` needs to be the same in the three places. The actual value can change on a site-specific basis.

### PurgeDirectoryFrequency

The `PurgeDirectoryFrequency` directive, whose characteristics are listed in the following table, specifies how often the physical directory is checked to see if files require purging.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the frequency at which the physical directory is purged</td>
</tr>
<tr>
<td>Syntax</td>
<td>`PurgeDirectoryFrequency time-unit[H</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td><code>jadehttp-handler</code></td>
</tr>
<tr>
<td>Module</td>
<td><code>mod_jadehttp</code></td>
</tr>
<tr>
<td>Default</td>
<td><code>PurgeDirectoryFrequency 1H</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>PurgeDirectoryFrequency 5H</code></td>
</tr>
</tbody>
</table>

You can append a multiplier to the `time-unit` value. The `H` multiplier multiplies by 3,600, the `M` multiplier multiplies by 60, and all other `time-unit` values are treated as seconds.

Specify zero (0) for the `time-unit` value to turn off the purging of the physical directory.

The minimum that you can specify for the `time-unit` value is 5 minutes and the maximum is **24H** (hours).

Values outside this range are forced to their respective limits.

### PurgeDirectoryRule

The `PurgeDirectoryRule` directive, whose characteristics are listed in the following table, specifies when files in the virtual directory that are not read-only are purged.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies when non-read-only files in the virtual directory are purged</td>
</tr>
</tbody>
</table>
The default value of `Default` for this directive removes any standard files of type `.jpg`, `.png`, or `.gif` that are more than 12 hours old.

Specify a value of `AllWritable` if you want to purge all files in the virtual directory that are not read-only.

**PurgeFileAge**

The `PurgeFileAge` directive, whose characteristics are listed in the following table, specifies the length of time since a file was last modified before it is purged from the physical directory.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the minimum modified age before the files are purged</td>
</tr>
<tr>
<td>Syntax</td>
<td>`PurgeFileAge time-unit[D</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Default</td>
<td><code>PurgeFileAge Default</code></td>
</tr>
<tr>
<td>Example</td>
<td><code>PurgeFileAge AllWritable</code></td>
</tr>
</tbody>
</table>

You can append a multiplier to the `time-unit` value. The `D` multiplier multiplies by 86,400, the `H` multiplier multiplies by 3,600, the `M` multiplier multiplies by 60, and all other `time-unit` values are treated as seconds.

Specify zero (0) for the `time-unit` value to turn off the purging of the physical directory.

The minimum that you can specify for the `time-unit` value is 30 seconds and the maximum is 35D (days). Values outside this range are forced to their respective limits.

See also the `JadeWebServiceProvider` or `WebSession` class `deleteVirtualDirectoryFile` method.

**TcpConnection[n]**

The `TcpConnection[n]` directive can have one of two types of syntax, depending on the value of the `ApplicationType` directive. The optional `[n]` value can be a unique number in the range 1 through 9. If you do not specify the optional `n` group suffix, JADE treats the value as 1.

You can specify the `host-name` value as a host name or as an IP address. If you specify a host name, it must resolve to a single IP address. If you append the optional `n` value to the `TcpConnection[n]` directive, it indicates the group to which the specified host name port number range is assigned.
If the value of the ApplicationType directive is WebEnabledForms, WebServices, or HtmlDocuments, the characteristics listed in the following table specify the TCP/IP connection details between the mod_jadehttp module and the JADE Web application.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the TCP/IP connection to the JADE application</td>
</tr>
<tr>
<td>Syntax</td>
<td>TcpConnection[n] host-name port [MinInUse [MaxInUse [CloseDelay [ConnectionGroup]]]]</td>
</tr>
</tbody>
</table>

**Note** For the WebEnabledForms application type, you must specify the MinInUse, MaxInUse, CloseDelay, and ConnectionGroup values.

<table>
<thead>
<tr>
<th>Context</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example 1</td>
<td>TcpConnection server1 6008</td>
</tr>
<tr>
<td>Example 2</td>
<td>TcpConnection2 10.1.1.19 51030 3 15 300</td>
</tr>
<tr>
<td>Example 3</td>
<td>TcpConnection3 10.1.1.19 51031 1 10 30 30 Forms</td>
</tr>
</tbody>
</table>

For details about the optional MinInUse, MaxInUse, CloseDelay, and ConnectionGroup values, see the following subsections.

**MinInUse**

The optional MinInUse value specifies the minimum number of connections that can be opened for the TcpConnection[n] directive when the value of the ApplicationType directive is set to WebServices or HtmlDocuments.

This value must be specified when the ApplicationType directive is set to WebEnabledForms.

The default minimum number of connections is 1.

**MaxInUse**

The optional MaxInUse value specifies the maximum number of connections that can be opened for the TcpConnection[n] directive when the value of the ApplicationType directive is set to WebServices or HtmlDocuments.

This value must be specified when the ApplicationType directive is set to WebEnabledForms.

The default maximum number of connections is the value of MinInUse.

**Note** The MaxInUse value is forced to be equal to or greater than the MinInUse value.

**CloseDelay**

When the current number of in-use connections for the TcpConnection[n] directive exceeds the MinInUse value, you can specify the optional CloseDelay value to indicate how many seconds before the extra connections are closed after they become inactive.

This value must be specified when the ApplicationType directive is set to WebEnabledForms.
The close delay defaults to zero (0) seconds before inactive connections are closed, indicating that the connection is never closed. The maximum value is 300 seconds.

**ConnectionGroup**

The optional ConnectionGroup value specifies the name of the connection group associated with the TcpConnection[n] directive; that is, the unique name that is used by applications of type WebEnabledForms to identify the TCP/IP port to which a user request is to be sent.

Although there is no default connection group, you must specify a unique value when the value of the ApplicationType directive is set to WebEnabledForms.

**VirtualDirectory**

The VirtualDirectory directive, whose characteristics are listed in the following table, is an easier way to specify the same value as the PhysicalDirectory directive.

The VirtualDirectory directive is mainly of use when the same jadehttp.conf file is designed to run on multiple hosts.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Specifies the TCP/IP connection to the JADE application</td>
</tr>
<tr>
<td>Syntax</td>
<td>VirtualDirectory directory-name</td>
</tr>
<tr>
<td>Context</td>
<td>Location</td>
</tr>
<tr>
<td>Handler</td>
<td>jadehttp-handler</td>
</tr>
<tr>
<td>Module</td>
<td>mod_jadehttp</td>
</tr>
<tr>
<td>Example</td>
<td>VirtualDirectory /var/web/myWebApp/files</td>
</tr>
</tbody>
</table>

The following is an example of the use of the VirtualDirectory directive.

```xml
<IfModule mpm_winnt.c>
  <Directory "c:/temp/">
    ...
  </Directory>
  Alias /jade/images/ "c:/temp/"
</IfModule>

<IfModule !mpm_winnt.c>
  <Directory "/tmp/">
    ...
  </Directory>
  Alias /jade/images/ "/tmp/"
</IfModule>

<location /jade/webform>
  SetHandler jadehttp_handler
  VirtualDirectory /jade/images/
</location>
```
In this example, depending on where this jadehttp.conf file is used, the physical directory to which transferred files are written is different. The VirtualDirectory directive uses the URL location of /jade/images/ to determine the actual directory name to use. For details about retaining files passed to mod_jadehttp, see "Firewall for the JADE Internet Environment", earlier in this chapter.

Apache Configuration Examples

In the configuration examples in this section, the following has assumed to have been added to the httpd.conf file, which is the default main configuration file for Apache.

The primary Apache configuration file is normally called http.conf. You can place mod_jadehttp directives in this file or in a separate file.

The recommended method is to create a separate jadehttp.conf file that contains the required mod_jadehttp directives, and include this file in the primary http.conf file as follows.

```
LoadModule jadehttp_module modules/mod_jadehttp.so
Include conf/jadehttp.conf
```

Each of the examples in the following subsections has a different jadehttp.conf file.

Minimal Configuration Example

The following is a minimal example of the jadehttp.conf file that enables you to connect to the JADE example Erewhon system and access the Web shop.

```
<IfModule mod_jadehttp.c>
  <Directory "/pictures">
    Order deny,allow
    Allow from All
  </Directory>

  Alias /jadeImages/ "/pictures"

  <Location /jade-info>
    SetHandler jadehttp-info
    order deny,allow
    allow from localhost
  </Location>

  <Location /Erewhon>
    SetHandler jadehttp-handler
    ApplicationType WebEnabledForms
    Application WebShop
    TcpConnection jadeserver 6007 3 15 300 Forms
    VirtualDirectory /jadeImages/
  </Location>
</IfModule>
```

From your Web browser, use the http://webserver/Erewhon URL to connect to the example Erewhon system.

To find out status information about what is happening inside the mod_jadehttp module, use the http://localhost/jade-info URL on the workstation running the Apache Web server.
Extended Configuration Example

Add the following to the jadehttp.conf file, referred to in the httpd.conf file under "Apache Configuration Examples", earlier in this chapter (or you can include it in the httpd.conf file itself).

```xml
<IfModule mod_jadehttp.c>
  <Location /jade>
    SetHandler jadehttp-handler
    FaultDocument TcpConnectFailed "messages/CanNotConnect.html"
  </Location>

  <Location /jade/info>
    SetHandler jadehttp-info
  </Location>

  <Location /jade/Erewhon>
    ApplicationType WebServices
    Application Erewhon
    TcpConnection cnwjdc1a 6007 1 10
  </Location>

  <Location /jade/erewhon/user>
    ApplicationType WebEnabledForms
    Application UserApp
    TcpConnection cnwjdc1a 6007 3 10 30 Group1
  </Location>

  <Location /jade/erewhon/admin>
    ApplicationType WebServices
    Application AdminApp
    TcpConnection cnwjdc1a 6018
  </Location>
</IfModule>

Extended Configuration Example with Additional Apache Directives

Add the following to the jadehttp.conf file, referred to in the httpd.conf file under "Apache Configuration Examples", earlier in this chapter (or you can include it in the httpd.conf file itself).

```xml
<IfModule mod_jadehttp.c>
  <Location /jade-info>
    JadeHttp_Trace Oxffff Logs/jadehttp.log 5M
    SetHandler jadehttp-handler
    FaultDocument TcpConnectFailed "messages/CanNotConnect.html"
    ErrorDocument 503 "http://localhost:8080/messages/Custom503.html"
  </Location>

  <Location /jade/info>
    SetHandler jadehttp-info
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/8 10.1.1.100
  </Location>
</IfModule>
Chapter 3 Configuring JADE

<?Location /jade/BankOfJade>
  JadeHttp_Trace 0xff
  ApplicationType HtmlDocuments
  Application BankApp
  ConnectionGroup FinanceDepartment
  TcpConnection cnwjdc1a 6007
  Order deny,allow
  Deny from all
  Allow from 192.168.0.0/16
</Location>

<?Location>
  ApplicationType WebEnabledForms
  Application Company
  ConnectionGroup Customers
  TcpConnection Host1a 2000 1 5 300 CompanyForms
  Order allow,deny
  Allow from all
</Location>

<?Location /jade/erewhon/user>
  ApplicationType WebServices
  Application WebCompany
  TcpConnection Host1a 21000 5 15 300
  TcpConnection2 Host2a 21001 5 15 300
  Order deny,allow
  Deny from all
  Allow from 143.96.0.0/16
</Location>

<?Location /jade/erewhon/admin>
  ApplicationType WebServices
  Application WebCompany
  TcpConnection Host1a 21000 5 15 300
  TcpConnection2 Host2a 21001 5 15 300
  Order deny,allow
  Deny from all
  Allow from 127.0.0.0/8
</Location>

</IfModule>
Chapter 3  Configuring JADE

Apache Considerations

When using the Apache HTTP Server to connect to your JADE applications from the Internet, consider the following.

- The JADE Web design considerations documented under "Connecting to JADE Applications from an Apache HTTP Server", earlier in this chapter.

- Only JadeInternetTCPConnection class TCP/IP communications are supported. This applies to all versions of mod_jadehttp, regardless of the operating system.

- The security implications for mod_jadehttp are those of the IIS jadehttp.dll. Access to JADE data from the Web server can be controlled by Apache security directives.

- Apache and the HTTP generally use UTF-8 as the encoding scheme for Unicode data on the Web. As mod_jadehttp does not allow for this and passes the data directly to JADE, only ANSI data can be read or written to the JadeInternetTCPConnection object.

- As the JadeServer configuration directive that indicates details about the JADE application to which mod_jadehttp is connected is not yet implemented, it is assumed that the value of this directive is as follows.

  \[ \text{JadeServer 7.0 littleEndian Ansi} \]

Tuning Your Systems

The JADE object manager uses a default hash value of:

- 2053, when storing and locating persistent and transient locks. You can vary these hash values (for example, for systems that are likely to have a large number of concurrent persistent or transient locks) by using the PersistentLockHashSize or TransientLockHashSize parameter in the [JadeClient] and [JadeServer] sections of the JADE initialization file.

- 1019, for notifications. You can vary this hash value (for example, for systems that have a large number of notifications registered) by using the NotificationHashSize parameter in the [JadeClient] and [JadeServer] sections of the JADE initialization file.

Specifying Parameters in the JADE Command Line

The JADE command line (or program target), used when running a JADE application from outside of the development environment, contains parameters that enable you to specify your JADE configuration. Each element of the command line has an identifying prefix, followed by an equals sign (=) and a specified value.

Command line elements can be delimited by a space, a tab, or a comma. Parameters containing these delimiters must be enclosed in single quotes (') or double quote (""") characters.
**Notes** You can define parameters in the [JadeCommandLine] section of the JADE initialization file that you would normally specify on the command line and you can define two-level initialization file section names so that multiple programs on the same host can share a JADE initialization file. For details, see "Two-Level Section Names" and "Sharing JADE Initialization Files", in the JADE Initialization File Reference. See also "Placing Initialization File Parameters on the Command Line", in the following subsection.

As the command line structure of the batch JADE Database utility program (jdbutilb) differs from that of other JADE applications, the specification of a command line name parameter that matches an initialization file section unique identifier does not apply.

Unless your JADE initialization file is located in the bin directory, specify the ini parameter on the shortcut command line rather than in the [JadeCommandLine] section so that the executable knows immediately the location of the JADE initialization file that contains all information required by the program.

You can use the server parameter in a command line to specify the server Universal Resource Identifier (URI) target database and the client-server transport, instead of the server and path parameters in a command line and the ServerNodeSpecifications parameter in the [JadeClient] section of the JADE initialization file. For details, see "Format of the Server URI String", earlier in this chapter.

For details about running JADE in thin client mode, see "Invoking a JADE Presentation Client" under "Running the Application Server and Presentation Clients", in Chapter 2 of the JADE Thin Client Guide. For information about the parameters that enable you to handle multiple copies of JADE, see "Handling Multiple Copies of the JADE Program", later in this chapter.

The command line parameters are listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies …</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>Application name</td>
<td>app=Jade</td>
</tr>
<tr>
<td>path</td>
<td>Full path of your JADE database directory</td>
<td>path=c:\jade\system</td>
</tr>
<tr>
<td>schema</td>
<td>Schema name (mandatory except when app=Jade)</td>
<td>schema=Faults</td>
</tr>
<tr>
<td>ini</td>
<td>JADE initialization file</td>
<td>ini=c:\jade\system\jade.ini</td>
</tr>
<tr>
<td>[server]</td>
<td>Server type</td>
<td>server=SingleUser</td>
</tr>
<tr>
<td>[thinClient]</td>
<td>That jade.exe checks only arguments required to run in thin client mode</td>
<td>thinClient=false</td>
</tr>
<tr>
<td>[name]</td>
<td>That the program checks for a matching initialization file section name</td>
<td>name=SecureAppServer</td>
</tr>
<tr>
<td>[newcopy]</td>
<td>A new copy of jade.exe is to be started</td>
<td>newcopy=false</td>
</tr>
<tr>
<td>[host]</td>
<td>Host server node name or IP address</td>
<td>host=cnwchcs38</td>
</tr>
<tr>
<td>[port]</td>
<td>Port name or number of the host (server) node</td>
<td>port=6015</td>
</tr>
<tr>
<td>[interface]</td>
<td>TCP/IP name or the IP address of the client (local) node</td>
<td>interface=cnwme1c</td>
</tr>
<tr>
<td>[localport]</td>
<td>Port name or number of the client (local) node</td>
<td>localport=5017</td>
</tr>
<tr>
<td>[appServer]</td>
<td>Remote TCP/IP address or name of the application server</td>
<td>appServer=JadeServer</td>
</tr>
<tr>
<td>[appServerPort]</td>
<td>TCP/IP port number of the application server</td>
<td>appServerPort=1500</td>
</tr>
</tbody>
</table>
### Chapter 3 Configuring JADE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies …</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>[endJade]</td>
<td>The end of JADE-specific parameters (applicable only to the non-GUI client jadclient program, and ignored for all other JADE programs)</td>
<td>endJade</td>
</tr>
<tr>
<td>[arguments]</td>
<td>Optional user-defined parameters for the jadclient program</td>
<td>MyNonJadeApp.exe</td>
</tr>
</tbody>
</table>

When specifying your JADE command line:

- You must include the application (app) parameter.
- You must include the schema parameter (except when app=Jade).
- For JADE executables, you must include fully qualified name of your JADE initialization file in the ini parameter.
- You must include the path (path) parameter. The path parameter specifies the location of the JADE database. The database path cannot be greater than 260 characters.

When running in multiuser mode, the path is always relative to the viewpoint of the server node; for example, c:\jadesystem refers to the c: drive of the server node and not that of the local client workstation.

You should specify the fully qualified path name.

When you specify a relative path name in the path parameter of the command line, the path name is first converted to an absolute path, by using the following rule:

- A relative path name with a single leading slash character is pre-pended by the first two characters of the JADE HOME directory on the server node (that is, drive-letter:), which is assumed to be the parent of the bin directory. For example, if your installation directory is Jadelbin (that is, your JADE HOME directory is Jade) and you specify path=system, the full directory path is Jadelbin\system on the server.

In the following examples, the JADE HOME directory is assumed to be c:\jade.

<table>
<thead>
<tr>
<th>Path Specified in the Command Line</th>
<th>Actual Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>path=jade70/system</td>
<td>c:jade70/system</td>
</tr>
<tr>
<td>path=system</td>
<td>c:jade/system</td>
</tr>
</tbody>
</table>

- The ini parameter specifies the location of the JADE initialization file. Letters specifying the drive are relative to the viewpoint of the client node. The initialization file defaults to Jade.ini, located in your database (system) path.
- The optional server parameter specifies SingleUser or MultiUser.

**Note** If you do not specify a server value, the Server parameter in the [Jade] section of the JADE initialization file is used. If this parameter is not specified in the JADE initialization file, the server defaults to MultiUser.

- When you specify the name of a port rather than the number in the optional port parameter, the port name cannot start with a numeric value.
- Identifying prefixes are case-insensitive.
- Spaces are required only between the end of one parameter and the start of the next.
- Names and paths with embedded spaces should be enclosed in double quote characters (""").
Chapter 3  Configuring JADE

- If you want to specify parameters to your JADE non-GUI client application, specify the endJade parameter as the last of the JADE parameters before the first of your own parameters in the jadclient command line.

The jadclient program treats processing arguments enclosed in double (""" or single (""") quotation marks after the endJade parameter as single-string entries in the huge string array. The handling of strings in this huge string array is application-specific. For example, dir= "program files" is treated as a two-string entry and "dir= program files" is treated as a one-string entry. How these entries are handled is determined by your application.

**Note**  All other programs (for example, jade.exe, jadapp.exe, and jadload.exe) do not parse the command line beyond the endJade parameter, so any subsequent arguments will be ignored.

- Specify parameters (options) once only, in any order.

If a parameter on the command line is duplicated, the last occurrence found on the command line is used.

- When running JADE thin client applications in multiuser mode, ensure that the TCP/IP port number specified in the jadapp (or jadappb) command line appServerPort parameter is a different port number from that specified for the database server (for example, the jadrapp JADE Remote Node Access program).

The database server TCP/IP port number is specified in the ServerNodeSpecifications parameter remote port value in the [JadeClient] section and the NetworkSpecification parameter IP-address value in the [JadeServer] section of the JADE initialization file.

In multiuser mode, you can optionally use:

- The host command line parameter, to specify the host server node name or IP address.

This parameter specifies a host name that is resolvable by the Domain Name Server (DNS) or host file, and has a maximum length of 255 characters.

- The port command line parameter, to specify the host port name or number.

If you use this parameter to specify the port name, you must update the applicable file in a Windows operating system, as follows.

```
/<winnt>/system32/drivers/etc/services
/<windows9x>/services
```

**Note**  A port name cannot begin with a numeric value.

- The interface command line parameter, to specify the client (local) node name or IP address.

This parameter specifies a host name that is resolvable by the Domain Name Server (DNS) or host file, and has a maximum length of 255 characters.

- The localport command line parameter, to specify the client (local) port name or number.

If you use this parameter to specify the local port name, you must update the applicable file in a Windows operating system, as follows.

```
/<winnt>/system32/drivers/etc/services
/<windows9x>/services
```

**Note**  A local port name cannot begin with a numeric value.
Specify the optional interface and localport parameters if you want to bind a specific network adapter in a server node that has more than one network adapter installed; for example, to allow an administrator to ensure connections from clients connect on the fastest interface or to allow easier security when used in conjunction with a firewall or router access list. (By default, JADE defaults to the primary network adapter in the node.)

If you do not specify a host or a port parameter, the ServerNodeSpecifications parameter value in the [JadeClient] section of the JADE initialization file is used. If neither of these values is specified in the JADE initialization file, they default to LocalHost and 6005, respectively.

The host, port, interface, and localport command line parameters, if specified, override (but do not update) the ServerNodeSpecifications parameter [host-name or IP-address], [remote-port], [local-interface-name or IP-address], or [local-port] values, respectively, in the [JadeClient] section of the JADE initialization file.

If you are running multiple copies of JADE, you can have multiple jade.exe shortcuts using the same JADE initialization file (jade.ini) by using the optional name parameter, instead of having separate files for each copy of JADE that you run. For details about multiple JADE programs on the same host sharing a JADE initialization file, see "Sharing JADE Initialization Files", in the JADE Initialization File Reference.

Placing Initialization File Parameters on the Command Line

You can override any JADE initialization file values by placing them on the command line.

**Note** For JADE executables, specify the fully qualified name of your JADE initialization file in the command line.

JADE first looks for command line parameters of the form section-name.parameter-name=value and if not present, checks the JADE initialization file for the section name and parameter name combination.

Two-level section names are supported in the command line.

Placing JADE initialization file entries on the command line enables:

- JADE developers and system administrators to quickly test alternate configurations, rather than having to create multiple JADE initialization files.
- JADE developers to share one JADE initialization file.

For example, to allow specific application servers to run specified applications only, on the command line, specify the jadapp application server executable shortcut command line as follows.

```
R:\Test\jade\bin\jadapp appServerPort=1500 thinClient=true path=r:\test\jade\system ini=r:\jade\system\test\myjade.ini JadeAppServer.EnableAppRestrictions=true JadeAppServer.AllowSchemaAndApp1="MySchema,SecretFixupApp" server=multiUser host=devsrvr38 port=6015
```
Notes As the size of the command line is imposed by the operating system, this may restrict the number of JADE initialization file values that you can place on the command line. We recommend that JADE administrators keep the placement of initialization file parameters to a minimum, because long command lines can become hard to maintain. However, if you require only a few non-updating initialization file values, putting them on the command line saves the creation of a new JADE initialization file section.

You cannot use the Application class setProfileString or setProfileStringAppServer method to update JADE initialization file parameter values specified on the command line. (Attempts to do so return a value of false and the parameter values remain unchanged.)

If you specify JADE initialization file section and parameter names in the command line, initialization file parameter values that are normally updated by methods are not updated, as the command line values have precedence. The parameter values in the JADE initialization file remain unchanged in this situation. For example, if you specify JadeServer.MaxServerThreads=10 in the command line of the JADE Remote Node Access program when the value of the [JadeServer] section MaxServerThreads parameter is 5 and you then change the Maximum Server Threads text box value in the Thread Configuration dialog to 7, the command line value of the application remains 10 and the [JadeServer] section MaxServerThreads parameter remains 5.

When obtaining values for command line arguments (for example, path, schema, appServer, and so on), the order of precedence is as follows.

1. Command line
2. If name=unique-identifier is on the command line, the [unique-identifier].JadeCommandLine section parameter values are obtained from the JADE initialization file
3. The parameters from the [JadeCommandLine] section of the JADE initialization file
4. An internal default value, if applicable

See also "Location of the JADE Initialization File", "Two-Level Section Names" under "Format of the JADE Initialization File", in the JADE Initialization File Reference.

Handling Multiple Copies of the JADE Program

Use the newcopy command line parameter to force a new copy of the jade.exe JADE program to exist; for example, for testing in a multuser environment. When this parameter is set to true (the default) and the JADE icon is clicked to launch JADE, a new copy of JADE is launched if the database is already open.

Specify a value of false to indicate that when the JADE icon is clicked and JADE is running, another application is started but another copy of the JADE program is not started; that is, execution is transferred to the copy of JADE that is currently running.

When initiating a thin client-based jade.exe executable with the newcopy command line parameter set to false, execution is transferred to an existing jade.exe copy if that copy is running in thin client mode and it is connected to the same application server (the TCP address and TCP/IP port number must match), so that only one copy of jade.exe runs multiple applications connected to the same application server.

If no other matching copy of jade.exe is running, execution continues as normal.

Note If you create a shortcut that has the newcopy parameter set to false and you specify a different JADE initialization file from the one with which the node was started, the active JADE initialization file is the one that was specified when the node started up and not the one specified in the newcopy=false shortcut. (You can call the Application class getIniFileName method in the new application to get the name of the initialization file that was used when the node started up.)
Specifying Your Administration Options

When you sign on to JADE, you can select the Administration option button from the Select Options group box to specify your installation preferences. The installation preferences are those that apply to all JADE browser and Painter windows in the JADE development database of your installed JADE release.

When JADE has been installed, these preferences can then be changed individually by a user (by selecting Preferences command from the Options menu) or for the whole JADE development environment (by using this option).

The JADE installation preferences provide a form that enables you to select the appropriate sheet to specify the global options for your JADE development environment work sessions.

The options that you can maintain and the sections in this chapter (for options that apply only to the administrative JADE Installation Preferences dialog) or in Chapter 2 of the JADE Development Environment User’s Guide that describe these options are listed in the following table.

<table>
<thead>
<tr>
<th>Installation Option</th>
<th>For details, see…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerator Keys</td>
<td>&quot;Maintaining Accelerator Keys&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Browser</td>
<td>&quot;Maintaining Browser Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Editor</td>
<td>&quot;Maintaining the Editor Display&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Editor Key Bindings</td>
<td>&quot;Maintaining Editor Key Bindings&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Editor Options</td>
<td>&quot;Maintaining Editor Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Exit</td>
<td>&quot;Maintaining Exit Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Lock</td>
<td>&quot;Maintaining Lock Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Method</td>
<td>&quot;Maintaining Method Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>&quot;Maintaining Miscellaneous Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Painter</td>
<td>&quot;Maintaining Painter Options&quot;, later in this chapter</td>
</tr>
<tr>
<td>Relationship</td>
<td>&quot;Maintaining Relationship Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Schema</td>
<td>&quot;Maintaining Schema Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Status list</td>
<td>&quot;Maintaining Status List Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Text Templates</td>
<td>&quot;Maintaining Text Templates&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
<tr>
<td>Window</td>
<td>&quot;Maintaining Window Options&quot;, in Chapter 2 of the JADE Development Environment User’s Guide</td>
</tr>
</tbody>
</table>
The JADE Installation Preferences dialog also contains the File, Admin, Admin, and Help menus.

Using the File Menu

Use the commands in the File menu from the JADE Installation Preferences dialog to administer the installation preferences for your JADE development database.

The File menu commands are listed in the following table.

<table>
<thead>
<tr>
<th>Command</th>
<th>For details, see…</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save &amp; Logoff</td>
<td>Saving Your Options and Logging Off</td>
<td>Saves your options, logs off from JADE, and displays the JADE sign-on dialog</td>
</tr>
<tr>
<td>Logoff</td>
<td>Logging Off from the Installation Preferences Dialog</td>
<td>Logs off from JADE and then displays the JADE sign-on dialog</td>
</tr>
<tr>
<td>Save &amp; Exit</td>
<td>Saving Options and Exiting from the Installation Preferences Dialog</td>
<td>Saves your options and then exits from the JADE Installation Preferences dialog</td>
</tr>
<tr>
<td>Exit</td>
<td>Exiting from the Installation Preferences Dialog</td>
<td>Exits from the JADE Installation Preferences dialog</td>
</tr>
</tbody>
</table>

Saving Your Options and Logging Off

Use the Save & Logoff command from the File menu to save your installation preferences and then exit from the JADE Installation Preferences dialog, in preparation for starting a new work session.

**To save your installation preferences and then log off**

- Select the Save & Logoff command from the File menu.

When all installation preferences have been saved, the JADE Installation Preferences dialog is closed after a momentary delay and the JADE sign-on dialog is then displayed, to enable you to enter your password and start a new JADE work session.

Logging Off from the Installation Preferences Dialog

Use the Logoff command from the File menu to exit from the JADE Installation Preferences dialog, in preparation for starting a new work session.

**To log off from the JADE Installation Preferences dialog**

- Select the Logoff command from the File menu.

After a momentary delay, the JADE Installation Preferences dialog is closed and the JADE sign-on dialog is displayed, to enable you to enter your password and start a new JADE work session.

**Note** No installation preferences that you have specified will be saved.

Saving Options and Exiting from the Installation Preferences Dialog

Use the Save & Exit command from the File menu to save your installation preferences and then exit from the JADE Installation Preferences dialog.
To save your installation preferences and then exit

- Select the **Save & Exit** command from the File menu.

When all installation preferences have been saved, the JADE Installation Preferences dialog is then closed after a momentary delay.

**Exiting from the Installation Preferences Dialog**

Use the **Exit** command from the *File* menu to exit from the JADE Installation Preferences dialog.

**To exit from the JADE Installation Preferences dialog**

- Select the **Exit** command from the *File* menu.

After a momentary delay, the JADE Installation Preferences dialog is then closed.

**Note** No installation preferences that you have specified will be saved.

**Using the Admin Menu**

Use the commands in the Admin menu from the JADE Installation Preferences dialog to administer the installation preferences for your JADE development database.

The Admin menu commands are listed in the following table.

<table>
<thead>
<tr>
<th>Command</th>
<th>For details, see…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset to Defaults</td>
<td>Resetting JADE Default Preferences</td>
</tr>
<tr>
<td>Remove Sources</td>
<td>Removing Source Code</td>
</tr>
<tr>
<td>Backup Database</td>
<td>Backing Up Your Development Database from a Client Workstation</td>
</tr>
</tbody>
</table>

**Resetting JADE Default Preferences**

Use the **Reset to Defaults** command from the Admin menu to reset all installation preferences to the default values.

Any installation preferences that you have specified are then reset to the JADE default preferences. These default values are the installation preferences that were last specified and saved using the JADE Installation Preferences dialog.

If you have not yet specified and saved any installation preferences, the default preferences are those that are supplied with JADE.

**Removing Source Code**

The **Remove Sources** command enables you to remove source code from your user-defined schemas only. You cannot remove source code from JADE-supplied schemas.

Use this command when you want to release JADE applications that do not contain method source code; for example, when you release a schema containing JADE applications developed for a third-party.
Notes  Ensure that you have backed up the schema, including source code, before you use this command.

For details about including recompiled methods in patch versioning when a patch is to be applied to a schema that does not have source available, see "Enabling or Disabling Patch Versioning" and "Setting Up a Patch Number", in Chapter 3 of the JADE Development Environment Administration Guide. For details about finding the position in a method source if you want to locate the position at which an exception occurred in an application from which source code has been removed, see "Finding a Method Source Position", in Chapter 4 of the JADE Development Environment User's Guide. For details about encrypting extracted source code, see "Encrypting Schema Source Files", in Chapter 10 of the JADE Development Environment User's Guide.

Backing Up Your Development Database from a Client Workstation

The Backup Database command enables you to backup your JADE development environment database without having to sign off all users or bring down the server node.

You can backup your JADE development environment database online when the database is active for both read and write access. When backing up your JADE database, you can:

- Specify the backup directory path
- Compress or verify backed up files and optionally overwrite existing files
- Lock the database for write access

For more details, see "Backing Up Your JADE Development Environment", in Chapter 3 of the JADE Database Administration Guide.

Using the Patch Menu

Use the commands in the Patch menu from the JADE Installation Preferences dialog to administer patch version control for your JADE development database.

Patch version control enables you to set a numeric patch version that records all additions, deletions, and updates made to schema entities (for example, methods, properties, constants, and so on) until a new patch version number is set and the patch version history is removed when the changed entities in the previous patch version are extracted.

The Patch menu commands are listed in the following table.

<table>
<thead>
<tr>
<th>Command</th>
<th>Documented under…</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable / Disable</td>
<td>Enabling or Disabling Patch Versioning</td>
<td>Enables or disables patch versioning for your database</td>
</tr>
<tr>
<td>Set Patch Number</td>
<td>Setting Up a Patch Number</td>
<td>Sets a new patch version number</td>
</tr>
<tr>
<td>Extract Patch</td>
<td>Extracting a Patch Version</td>
<td>Extracts all entities that have changed in the current patch version</td>
</tr>
<tr>
<td>Recreate History</td>
<td>Recreating a History of Patch Version Changes</td>
<td>Recreates the history of changes in the current patch version</td>
</tr>
<tr>
<td>Remove Patch History</td>
<td>Removing a Patch History of Changes</td>
<td>Removes the patch history that matched specified criteria</td>
</tr>
</tbody>
</table>

For details, see "Patch Versioning" under "Administering Your JADE Environment", in Chapter 3 of the JADE Development Environment Administration Guide.
Using the Help Menu

Use the commands in the Help menu from the JADE Installation Preferences dialog to access the standard Common User Access (CUA) help options. (For details about these options, see "JADE Portable Document Format (PDF) Online Help", in Chapter 2 of the JADE Development Environment User's Guide.)

The Help menu commands are listed in the following table.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Opens the directory of JADE online help documents</td>
</tr>
<tr>
<td>About</td>
<td>Displays details about the current version of JADE</td>
</tr>
</tbody>
</table>

Specifying Your JADE Installation Preferences

The sheets of the JADE Installation Preferences dialog enable you to maintain your JADE installation preferences. The preferences that you specify apply to all new users.

The preferences for the current users are not modified. For details about maintaining preferences for current users and details about the sheets that are common to administrative installation and to individual user preferences, see "Setting User Preferences", in Chapter 2 of the JADE Development Environment User's Guide.

Specify your installation preferences when you first install JADE and before any users sign on, or at any time for new users of the JADE release.

The preferences that you set (and then save) become the new default preferences for your installation of JADE.
To maintain your installation preferences

1. From the JADE sign-on dialog, select the **Administration** option button from the Select Options group box.

The JADE Installation Preferences dialog, shown in the following diagram, is then displayed.

The types of options are contained in sheets, with the **Editor** sheet displayed by default. (For details about using the **Editor** sheet, see "Maintaining the Editor Display" and "Maintaining Editor Options", in Chapter 2 of the JADE Development Environment User's Guide.)

2. Select the type of options that you want to change, by clicking on the sheet name; for example, click **Status List** if you want to view or change these options.

3. Make the required option changes.

4. To save your changes, use the **Save & Logoff** command or the **Save & Exit** command from the File menu. For details, see "Saving Your Options and Logging Off" or "Saving Options and Exiting from the Installation Preferences Dialog", respectively, earlier in this chapter.

When you have selected and saved your JADE preferences, the specified preferences become the new JADE default preferences when you exit or log off from the JADE Installation Preferences dialog.
Maintaining Painter Options

In JADE Painter, the default display of toolbars and grids and the position of the Properties dialog can be specified. For example, you can specify that the Control palette is not displayed on the Painter toolbar by default.

As painter values apply to the whole of the JADE development environment, the Painter sheet is available only on the administrative JADE Installation Preferences dialog.

To change the default Painter options

1. Click the Painter sheet of the JADE Installation Preferences dialog. The Painter sheet, shown in the following diagram, is then displayed.

2. If you do not want the Control palette displayed in the Painter toolbar, check the Hide Control Palette check box.

3. If you do not want the Alignment and Size palette displayed in the Painter toolbar, check the Hide Alignment/Size Palette check box.

4. If you do not want the status bar displayed in Painter windows, check the Hide Status Bar check box.

5. If you do not want the Tools palette displayed in the Painter toolbar, check the Hide Tools Palette check box.

6. If you want controls on Painter forms to snap to a grid, check the Snap to Grid check box.
7. If you want a grid displayed on Painter forms, check the Show Grid check box.

8. If you do not want a default grid width of 10 pixels, enter the required width in the Grid width text box.

9. If you do not want a default grid height of 10 pixels, enter the required height in the Grid height text box.

10. If you want the Properties dialog to be always positioned on top of other control in Painter forms by default, check the Properties On Top check box.

**Reregistering a JADE System in Batch Mode**

The jadregb program enables you to automate the registration of a JADE system with your new licence information by running the registration program in batch mode (for example, from a command script), specifying the following information.

```
jadregb path=database-directory
    [online [ini=JADE-initialization-file=absolute-path-and-name]]
    [help | report | name="licence-name" key=licence-key
    [minStandard=integer-value] [minJade=integer-value]]
```

The number of server licences is split over the two types of run time operation; that is, standard (fat client) and JADE thin client.

**Note** JADE licenses are not transferred automatically between databases in an SDE. It is your responsibility to apply new licenses to any existing databases in an SDE. In addition, to ensure proper operation, you must apply the primary license to every secondary.

You can optionally specify the minimum number of licences reserved for both types of run time operation, if required. When you have successfully specified the new licence information for the database:

- Standard information is output to stdout and error information is output to stderr.

  For details about displaying and redirecting the output from JADE batch utilities, see the DisplayApplicationMessages, LogServer, and UseLogServer parameters under "JADE Log Section [JadeLog]", in the JADE Initialization File Reference.

- The _control.dat file is updated.

If the jadregb executable program fails, a non-zero exit code is returned and an error message is displayed; for example, the licence key that you specified was invalid or you did not enclose your licence name in double quote characters ("""). The batch registration program parameters are described in the following subsections.

**path**

The path parameter specifies the full path of the JADE database directory; for example:

```
c:\jade\system
```

**ini**

The optional ini parameter enables you to specify the fully qualified name of your JADE initialization file if it is not located in the database directory or it has a file name other than the default value of jade.ini.
name

The name parameter specifies your registered licence name (displayed on your Certificate of Authorisation, which may be an e-mail message providing you with your licence name and key). For example:

"Snazzy Solutions Incorporated"

You must specify your licence name, enclosed in double quote characters. This name must be typed correctly (it is case-sensitive), as it is validated against your licence key.

key

The key parameter specifies the assigned licence key; for example:

99999999FFFFFFFF99999999FFFFFFFF

Enter the licence key exactly as it is specified on your Certificate of Authorisation, but without spaces.

online

The optional online parameter specifies that the jadregb program attempts to sign on to the JADE Remote Node Access utility (jadrap) to update the licence key.

If you specify the online parameter, you may also need to specify the ini parameter, which consists of the full (absolute) path and name of the JADE initialization file that enables the jadregb program to obtain the multiuser initialization file settings that it requires to sign on to the JADE Remote Node Access utility (jadrap).

If you do not specify the online parameter, the _control.dat file is updated directly and no client application or database server can be running.

minStandard

The optional minStandard parameter specifies the minimum number of runtime standard fat client licences that can be reserved at run time to ensure that a specific number of licences is available for standard fat clients at any time; for example:

minStandard=4

You cannot specify a minimum number of standard fat client licences greater than the number of your registered server (run time) licences.

The default value of zero (0) indicates that there is no minimum number of standard fat client runtime licences.

minJade

The optional minJade parameter specifies the minimum number of runtime JADE thin client licences that can be reserved at run time to ensure that a specific number of licences is available for JADE thin clients at any time; for example:

minJade=6

You cannot specify a minimum number of JADE thin client licences greater than the number of your registered server (run time) licences.

The default value of zero (0) indicates that there is no minimum number of JADE thin client runtime licences.
Chapter 3  Configuring JADE

report

The optional report parameter displays information about all existing licences installed in your system; for example, primary, secondary, and relational licences.

help

The optional help parameter displays the required parameters and their values.

Using the JADE Version Information Utility

The jverinfo JADE Version Information utility enables you to check the hot fixes that are applied to your JADE system and to optionally retrieve the version of user data map files. For more details, see the following subsections.

- Checking which Hot Fixes are Applied
- Obtaining the Version Number of User Database Files

The jverinfo program outputs the following version information to a file so that you can view it on your workstation monitor, print it out, or send it to JADE Support if requested to do so.

- System version information (for example, the operating system version, the CPU model, and the registered owner and organization of the software and hardware).
- File version information, starting with the installed location, full release version, and modified timestamp of each executable file followed by the installed location, full version number, and modified timestamp of each Dynamic Link Library (DLL) file.
- Types of CPU and operating system (for example, columns that contain i686 or AMD64 for the CPU type and WinGUI, WinCE, or SYSV for the operating system type).
- User data map file version number, starting with the installed location, version number, build type, architecture, and modified timestamp of each user data file.
- As well as listing the version details of executables and libraries found in the bin directory, jverinfo lists the version information of thin client download binaries if they are in the default download locations (that is, it will not find the download location using the values defined in the JADE initialization file).

If the directory does not exist or no executable or library files are found, nothing is output.

The following is an example subset of directories.

i686-msoft-win32-ansi\download\bin
i686-msoft-win32-unicode\download\bin
armv4i-msoft-wm60-unicode\download\bin
i686-msoft-x86emu-unicode\download\bin

The JADE Version Information utility program is not dependent on any other JADE libraries. You can therefore use one version of the jverinfo program to report on any release, whether it is a Unicode or an ANSI build.

To run the jverinfo program under a Windows operating system:
1. Copy the jverinfo.exe program to the JADE binary directory from which you want it run.
   Alternatively, you can create a shortcut on your desktop or in the appropriate Start folder.
2. Execute the program to run it.
Chapter 3  Configuring JADE

By default, the program writes its output to a file called versioninfo.txt, which is located in the working directory.

To automate the program, you can optionally specify the binary directory, user data map file directory (which is usually the system directory), system file directory, and the output file name, in the following format.

```
jverinfo [binpath=binary-directory-name] [binmask=binary-file-type] [datpath=user-data-directory] [out=output-file-name | stdout] [syspath=system-file-directory] [noThinClients] [extraDirs=[file-name]] [noExtraDirs]
```

The following is an example of the command line set up to run the program.

```
c:\jade\bin\jverinfo binpath=c:\jade\production_bin datpath=c:\jade\system out=c:\jade\producer.txt noExtraDirs
```

**Note**  The jverinfo command line options are case-insensitive.

You can use the optional:

- **binpath** parameter to specify the binary directory that is to be scanned.
- **binmask** parameter to refine the files that are output using the binpath parameter. The default syntax of this parameter is *.exe*.
- **out** parameter to specify stdout or the name of the output file that is created (defaulting to versioninfo.txt).
- **datpath** parameter to specify the user database map file directory (which is usually the system directory) that is to be scanned. It outputs an unformatted list of all user data map files, including the version number, the ANSI, Unicode 16, or Unicode 32 build type, whether the architecture is LittleEndian or BigEndian, and the time and date that each file was last modified.
- **syspath** parameter to specify the system file directory that is to be scanned and outputs the system file version, the ANSI, or Unicode 16, or Unicode 32 build type, and whether the architecture is LittleEndian or BigEndian, and the time and date that each file was last modified.
- **noThinClients** parameter to disable the version details of thin client download binaries, if applicable.
- **extraDirs=[file-name]** parameter to list custom directories to scan for binaries to display version information. The file specified by the extraDirs parameter must contain ANSI strings; wide-character strings are not supported by any version of jverinfo.

If you do not specify the optional *file-name* value, this parameter defaults to jverinfo.extradirs. The *file-name* value can be an absolute path or a relative path from the JADE Home directory (that is, one level up from the bin directory).

If the directory does not exist or it contains no files, it is ignored and nothing is output.

- **noExtraDirs** command line parameter, which you can use to disable the extraDirs functionality (that is, ignore the default jverinfo.extradirs file, if it exists).

If the first character on an input line is the hash (#) character, the line is treated as a comment and is ignored.

The input line can be an absolute path or a relative path from the JADE Home directory (that is, one level up from the bin directory).
Chapter 3  Configuring JADE

Checking which Hot Fixes are Applied

Should a problem arise that you cannot resolve by taking the appropriate action described in the relevant error message (documented in the JADEMsgs.pdf file) and your JADE licences include support, you may be asked for your current JADE version information in addition to the log files and user mode or process dumps specified in the previous section.

If you are uncertain of which hot fixes are applied to your JADE database, you can perform the following actions to determine exactly which hot fixes have been applied.

- Run the jverinfo JADE Version Information utility to obtain your current system and file versions. This stand-alone utility enables you to output your current JADE version information, the consolidated release you are running, and the hot-fixes with which you are running. Although you may think you have applied fixes, you may sometimes find that your records do not reflect the JADE system information itself.

- In the RootSchema, call the Schema class getAppliedPatches method to return a string containing all patches applied by a schema load to system schemas in your database. For details about the format of the string returned by this method, see the getAppliedPatches method in Volume 2 of the JADE Encyclopaedia of Classes.

Obtaining the Version Number of User Database Files

If you want to obtain the version number of user database map files (for example, on Windows Mobile devices running under Compact JADE), you can run jverinfo JADE Version Information utility with the optional datpath parameter, to obtain the current user data map file version number. This parameter can be applied only to user database map files; that is, user schema files (for example, _userdev.dat, _userscm.dat, and so on) represented by the DbFile class Kind_User_Schema constant and user data files (for example, _rootdef.dat, locktest.dat, and so on) represented by the DbFile class Kind_User_Data constant.

Use the optional setUserFileVersion command in the jdbutilb batch JADE Database utility to set the user data file version number to the specified 32-bit unsigned integer value.

When you specify the datpath parameter, the output file contains an unformatted list of all user data map files, including the version number, the ANSI, Unicode 16, or Unicode 32 build type, whether the architecture is LittleEndian or BigEndian, and the time and date that each file was last modified.

In the following example of user data version information, 93 is the specified user data file version number. If a user data version number has not been specified, the version number is zero (0).

<table>
<thead>
<tr>
<th>User:</th>
<th>(path= c:\jade\system)</th>
</tr>
</thead>
<tbody>
<tr>
<td>locktest.data</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>simpleconsumer.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>simpleprovider.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>wsconsumer.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:22 2008</td>
</tr>
<tr>
<td>_environ.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:20 2008</td>
</tr>
<tr>
<td>_monitor.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:22 2008</td>
</tr>
<tr>
<td>_reports.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>_rootdef.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>_stats.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>_userdev.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:21 2008</td>
</tr>
<tr>
<td>_usergui.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:20 2008</td>
</tr>
<tr>
<td>_userint.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:20 2008</td>
</tr>
<tr>
<td>_userscm.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:20 2008</td>
</tr>
<tr>
<td>_userxrf.dat</td>
<td>93  Ansi,LittleEndian Tue Jun 10 08:16:20 2008</td>
</tr>
</tbody>
</table>
In the RootSchema, call the DbFile class getUserPatchVersion method to return a 32-bit unsigned value as an Integer64 primitive type, which is the unformatted version number of user data map files.

If a user data version number is not set, this method returns zero (0).
Appendix A  Exit Values

This appendix covers the following topics.

- Overview
- Enabling the Use of Generic Exit Values for Windows
- General Exit Values Unique to Each Program

Overview

JADE provides exit values that apply to JADE programs and utilities. These exit values enable JADE administrators to develop tools that can take appropriate actions based on the exit values of the programs.

JADE programs can return exit values in the range of zero (0) through $2^{32}$. JADE programs generally returned zero (0) for success or the JADE error number if a problem occurs.

Standard exit values are limited to the range 0 through 127. Any value above this range is reserved by the operating system. You can derive the signal number that caused the program to exit by subtracting 128 from the exit value; for example, an exit value of 139 indicates that the program exited because it received a fatal signal 11 (SIGSEGV - Invalid memory reference).

Using standard exit values, ranges of JADE error numbers are grouped into a common generic exit value. In addition, a range of exit values is set aside for warnings, enabling a program to return information without it being regarded as a fatal problem.

JADE programs or utilities default to using current exit values, or the generic exit values by setting an initialization file parameter. For details, see "Enabling the Use of Generic Exit Values for Windows", in the following section.

**Note** Using generic exit values does not impact on performance of your JADE programs or utilities. It gives the site administrator the ability to write standard support tools.

Enabling the Use of Generic Exit Values for Windows

The [FaultHandling] section of the JADE initialization file provides the StandardExitValues parameter that enables you to specify that JADE programs or utilities return generic exit values.

This parameter defaults to `false`, as JADE programs and utilities can return exit values in the range of zero (0) through $2^{32}$ by default.

To specify that programs and utilities return generic exit values, set the `StandardExitValues` parameter in the [FaultHandling] section of the JADE initialization file to `true`.

In addition, you can use the:

- `CustomExitValue <JADE-error-message-number>` parameter in the [FaultHandling] section of the JADE initialization file when the `StandardExitValues` parameter is set to `true`, to remap JADE exit values (error numbers) to a user-defined standard exit code in the range 32 through 63.
The **CustomExitValue** parameter is valid only when the **StandardExitValues** parameter is set to **true**. If the **StandardExitValues** parameter is set to **false**, JADE exits with the applicable JADE error number. If the **StandardExitValues** parameter is set to **true**, JADE exists with the applicable error number from the remap table (listed in the following section) and checks for the existence of the **CustomExitValue** parameter or parameters in the [**FaultHandling**] section of the JADE initialization file.

**Node** class **userExitCode** property to specify the value that is returned when a JADE program (for example, **jade.exe**, **jadapp**, **jadrap.exe**, **jaded**, and so on) exits. For more details, see Chapter 1 of the JADE Encyclopaedia of Classes.

**Tip** You can use the **userExitCode** property, for example, to set a non-zero exit code that can then be checked in a batch file by using the **ERRORLEVEL** keyword to check for appropriate values of the **userExitCode** property.

### General Exit Values Unique to Each Program

The following table lists the exit values that are unique to each program (that is, an exit value for the **jadload** program may have a different meaning from that for the **jade** program).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>JADE Number Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>127</td>
<td>Fatal network errors</td>
<td>-101 through -120</td>
</tr>
<tr>
<td>0</td>
<td>Success, no error</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Methods in error were detected by the schema load</td>
<td>8510</td>
</tr>
<tr>
<td>2</td>
<td>One or more commands in command file did not complete successfully</td>
<td>8514</td>
</tr>
<tr>
<td>3</td>
<td>Upgrade validation failed</td>
<td>8723</td>
</tr>
<tr>
<td>4</td>
<td>Reorganization is required</td>
<td>8511</td>
</tr>
<tr>
<td>5</td>
<td>Reorganization operation cancelled by user request</td>
<td>3404</td>
</tr>
<tr>
<td>6</td>
<td>Reorganization is suspended waiting for the transition to be initiated</td>
<td>3413</td>
</tr>
<tr>
<td>7</td>
<td>Versioned control subclasses require reorganization during online deployment</td>
<td>8525</td>
</tr>
<tr>
<td>8</td>
<td>Schema load resulted in one or more incomplete schemas</td>
<td>8527</td>
</tr>
<tr>
<td>9 through 31</td>
<td>Reserved, on an individual JADE program basis</td>
<td>Unspecified</td>
</tr>
<tr>
<td>32 through 63</td>
<td>User-defined exit values</td>
<td>Unspecified</td>
</tr>
<tr>
<td>64 through 78</td>
<td>Reserved by the operating system</td>
<td>Unspecified</td>
</tr>
<tr>
<td>105</td>
<td>JADE Object Manager errors</td>
<td>4 through 1299</td>
</tr>
<tr>
<td>106</td>
<td>Collection exception errors</td>
<td>1300 through 1349</td>
</tr>
<tr>
<td>107</td>
<td>Miscellaneous runtime errors</td>
<td>1400 through 1448</td>
</tr>
<tr>
<td>114</td>
<td>Trace exception errors</td>
<td>1500 through 1599</td>
</tr>
<tr>
<td>127</td>
<td>Remote procedure call (RPC) request errors</td>
<td>1600 through 1699</td>
</tr>
<tr>
<td>108</td>
<td>Security errors</td>
<td>1700 through 1799</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
<td>JADE Number Range</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>113</td>
<td>Database engine errors</td>
<td>3000 through 3199</td>
</tr>
<tr>
<td>116</td>
<td>Synchronized Database Service (SDS) database errors</td>
<td>3200 through 3399</td>
</tr>
<tr>
<td>112</td>
<td>Database reorganization errors</td>
<td>3400 through 3499</td>
</tr>
<tr>
<td>127</td>
<td>Database server remote interface errors</td>
<td>3500 through 3599</td>
</tr>
<tr>
<td>120</td>
<td>JADE language interpreter errors</td>
<td>4000 through 4499</td>
</tr>
<tr>
<td>109</td>
<td>JADE Query Engine internal errors</td>
<td>4500 through 4999</td>
</tr>
<tr>
<td>114</td>
<td>File handling errors</td>
<td>5000 through 5099</td>
</tr>
<tr>
<td>114</td>
<td>Multimedia handling errors</td>
<td>5100 through 5299</td>
</tr>
<tr>
<td>114</td>
<td>Sort errors</td>
<td>5300 through 5399</td>
</tr>
<tr>
<td>121</td>
<td>Licence errors</td>
<td>5500 through 5599</td>
</tr>
<tr>
<td>112</td>
<td>JADE Object Manager data interchange errors</td>
<td>5700 through 5799</td>
</tr>
<tr>
<td>112</td>
<td>Internal management message exchange errors</td>
<td>5900 through 5999</td>
</tr>
<tr>
<td>119</td>
<td>Compiler errors</td>
<td>6000 through 6999</td>
</tr>
<tr>
<td>118</td>
<td>Method and schema file syntax errors</td>
<td>7000 through 7999</td>
</tr>
<tr>
<td>110</td>
<td>External database errors</td>
<td>8000 through 8255</td>
</tr>
<tr>
<td>125</td>
<td>ODBC errors</td>
<td>8256 through 8499</td>
</tr>
<tr>
<td>126</td>
<td>JADE Schema Load utility errors</td>
<td>8500 through 8599</td>
</tr>
<tr>
<td>111</td>
<td>Upgrade errors</td>
<td>8700 through 8799</td>
</tr>
<tr>
<td>124</td>
<td>JADE Database utility errors</td>
<td>9000 through 9999</td>
</tr>
<tr>
<td>117</td>
<td>TCP/IP network errors</td>
<td>10000 through 10499</td>
</tr>
<tr>
<td>122</td>
<td>JADE Monitor errors</td>
<td>11000 through 11999</td>
</tr>
<tr>
<td>127</td>
<td>JADE Remote Node Access utility errors</td>
<td>12000 through 12999</td>
</tr>
<tr>
<td>123</td>
<td>JADE user interface run time errors</td>
<td>14000 through 14499</td>
</tr>
<tr>
<td>123</td>
<td>JADE translatable string errors</td>
<td>14500 through 14999</td>
</tr>
<tr>
<td>114</td>
<td>Print errors</td>
<td>15000 through 15099</td>
</tr>
<tr>
<td>114</td>
<td>Editor errors</td>
<td>15500 through 15599</td>
</tr>
<tr>
<td>115</td>
<td>JADE development environment errors</td>
<td>16000 through 16999</td>
</tr>
<tr>
<td>117</td>
<td>Application connection errors</td>
<td>30000 through 30999</td>
</tr>
<tr>
<td>117</td>
<td>TCP/IP connection errors</td>
<td>31000 through 31499</td>
</tr>
<tr>
<td>117</td>
<td>Network proxy errors</td>
<td>31500 through 31999</td>
</tr>
<tr>
<td>117</td>
<td>Secure Sockets Layer (SSL) errors</td>
<td>32000 through 32499</td>
</tr>
<tr>
<td>117</td>
<td>X509 Certificate errors</td>
<td>32500 through 32999</td>
</tr>
<tr>
<td>112</td>
<td>Shared memory errors</td>
<td>33500 through 33599</td>
</tr>
<tr>
<td>116</td>
<td>Kernel, multiple server, errors, SDS</td>
<td>34000 through 34999</td>
</tr>
</tbody>
</table>
### Exit Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>JADE Number Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>116</td>
<td>SDS and inter-node communication errors</td>
<td>35000 through 35999</td>
</tr>
<tr>
<td>64</td>
<td>Command line usage error</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>Data format error</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>Cannot open input</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Address unknown</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>Host name unknown</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Service unavailable</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Internal software error</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>System error (for example, cannot fork)</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>Critical operating system file missing</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Cannot create (user) output file</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Input/output error</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>Temporary failure; user is prompted to retry</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>Remote error in protocol</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>Permission denied</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>Configuration error</td>
<td></td>
</tr>
</tbody>
</table>