Initialization File Reference

VERSION 2018.0.01

jade

Contents

Contents iv

Contents v

Contents vi

Contents vii

Contents viii

Contents

Contents xi

Before You Begin

The *JADE Initialization File Reference* is intended as the main source of information when you are maintaining JADE initialization file parameter values.

Who Should Read this Reference

The main audience for the *JADE Initialization File Reference* is expected to be system administrators.

What's Included in this Reference

The *JADE Initialization File Reference* has one chapter.

Chapter 1	Covers maintaining the JADE initialization file

Related Documentation

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

Title	Related to…
JADE Database Administration Guide	Administering JADE databases
JADE Development Environment Administration Guide	Administering JADE development environments
JADE Development Environment User's Guide	Using the JADE development environment
JADE Developer's Reference	Developing or maintaining JADE applications
JADE Monitor User's Guide	Monitoring and examining your JADE environment
JADE Encyclopaedia of Classes	System classes (Volumes 1 and 2), Window classes (Volume 3)
JADE Encyclopaedia of Primitive Types	Primitive types and global constants
JADE External interface Developer's Reference	Developing JADE applications using external interfaces
JADE Installation and Configuration Guide	Installing and configuring JADE
JADE Synchronized Database Service (SDS) Administration Guide	Administering JADE Synchronized Database Services (SDS), including Relational Population Services (RPS)
JADE .NET Developer's Reference	Developing or maintaining JADE applications from a .NET integrated development environment
JADE Object Manager Guide	JADE Object Manager administration
JADE Platform Differences Guide	Platform differences when running JADE applications
JADE Report Writer User's Guide	Using the JADE Report Writer to develop and run reports
JADE Runtime Application Guide	Administering deployed JADE runtime applications

Title	Related to…
JADE Thin Client Guide	Administering JADE thin client environments
JADE Web Application Guide	Implementing, monitoring, and configuring Web applications

Conventions

The *JADE Initialization File Reference* uses consistent typographic conventions throughout.

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

Key combinations and key sequences appear as follows.

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

Chapter 1 JADE Initialization File

This chapter covers the following topics.

- JADE Monitor Sections
  - JADE Monitor Section [JadeMonitor]
  - JADE Monitor Background Section [JadeMonitorBackground]
  - JADE Monitor Security Section [JadeMonitorSecurity]
- JADE Non-GUI Client Section [NonGuiClient]
- JADE Object Manager Environment Sections
  - Delta Options Section [DeltaOptions]
  - JADE Object Manager Client Module Section [JadeClient]
  - JADE Client Application Restrictions Section [JadeClientAppRestrictions]
  - JADE Loader Section [JadeLoader]
  - JADE Object Manager Server Section [JadeServer]
- JADE ODBC Sections
  - JADE ODBC Section [JadeOdbc]
  - JADE ODBC Server Section [JadeOdbcServer]
- JADE Patch Control Extensions Section [JadePatchControlExtensions]
- JADE Printing Section [JadePrinting]
- JADE Profiler Section [JadeProfiler]
- JADE Reorganization Section [JadeReorg]
- JADE Report Writer Section [JadeReportWriter]
- JADE Security Section [JadeSecurity]
- JADE Sentinel Section [JadeSentinel]
- JADE Start-Up Section [Jade]
- JADE Thin Client Sections
  - Application Server Section [JadeAppServer]
  - Presentation Client Section [JadeThinClient]
  - Environment-Specific Section [*environment-type*]
- JADE Unit Test Runner Section [JadeUnitTestRunnerUI]
- Synchronized Database Service Sections
  - Connection Parameters Section [ConnectionParams]
  - Relational Population Service (RPS) Node Section [JadeRps]
  - RPS Column-Mapping Method Exceptions Section [RpsIgnoreMethodExceptions]
  - RPS Manager Application Section [JadeRpsManager]

- SDS Administration Utility Section [JadeSDSAdmin]
- Synchronized Database Service Section [SyncDbService]

- Web Options Section [WebOptions]
- Web Session Section [WebSession]

Overview

Initialization parameters for JADE modules are grouped in categories. JADE applications and utilities provide the option to include JADE client parameters in an application-defined initialization file. This is done by the application when it first initializes the JADE Object Manager, by passing the directory and file name of the application-defined initialization file in the **jomInitialize** Application Programming Interface (API) call.

For more details, see "Initializing a Node" under "JADE Application Programming Interface (API)", in Chapter 3 of your *JADE Object Manager Guide*.

Note In a multiple-environment, each database can have its own unique JADE initialization file and configuration parameters. For details, see "Installing Multiple JADE Initialization Files", in Chapter 1 of your *JADE Installation and Configuration Guide*.

Location of the JADE Initialization File

The default JADE initialization file is located as follows.

1. If the **ini** parameter is specified on the command line and the argument value includes a directory, the JADE initialization file in the specified directory is used.

   If the directory is a relative path name, it is located relative to the JADE HOME directory (for example, if your installation directory is **Jade\bin**, your JADE HOME directory is **Jade**).

2. If the **ini** parameter is specified on the command line but the argument value does not include a directory:

   a. If the **path** parameter is specified on the command line, the JADE initialization file is prefixed with the directory specified in the **path** argument.

   b. If the **path** parameter is not specified on the command line, the JADE initialization file is prefixed with the install directory.

3. If the **ini** parameter is not specified on the command line:

   a. If the **path** parameter is specified on the command line, the JADE initialization file is set to the specified path followed by **/jade.ini**; for example, **w:/jade/system/jade.ini**.

   b. If the **path** parameter is not specified on the command line, the JADE initialization file is set to the install directory followed by **/jade.ini**; for example, **/usr/jade/jade.ini**.

If the JADE initialization file does *not* exist at the location specified by the preceding rules, an exception (125 - The specified or default Jade INI file does not exist) is raised. The file *can* be empty, however, in which case JADE writes appropriate default values to it, if required. (In earlier releases, an unintentional JADE initialization file was created in that location.)

Note For JADE executables, specify the fully qualified name of your JADE initialization file in the command line.

Conventions Used in this Reference

The sections and parameters in the JADE initialization file need not appear in the order in which they are defined in this chapter.

To change a parameter, search for it within the appropriate section.

This chapter provides the syntax, default value, purpose, and any usage hints for each parameter, using the following format.

Parameter-Name

The JADE initialization file parameter descriptions can have the components listed in the following table.

Component	Description
Value Type	The parameter value (that is, an integer, string, or boolean value, a file name or a range of numbers, or some other type).
Default	The default value for the parameter.
Purpose	A brief description of the function of the parameter.
Parameter is read when...	Indicates when a parameter value change is effective (for example, if a node restart is required). **Note** When all parameter values in a JADE initialization file section are read at the same time, this information is stated in the overview of the specific section rather than in each parameter description.
Hint or Note	Hints to assist in tuning or selecting parameter values or notes about parameter values.

Format of the JADE Initialization File

The JADE-specific or application-defined initialization file contains several JADE initialization sections, each of which consists of a group of related parameters. These sections and parameters are listed in the file in the following format.

```
[section-name]
keyname=value
```

In this example, **[*section-name*]** is the name of a section in the file. The enclosing brackets (**[ ]**) are required and the left bracket must be in the extreme left column on the screen.

The **keyname=*value*** statement defines the value of each parameter, or argument. A keyname is the name of a parameter. It can consist of any combination of digits and letters in uppercase or lowercase characters and it must be followed by an equals sign (**=**). The value is a primitive type (for example, an integer), depending on the parameter.

A semicolon character (**;**) indicates the start of a comment. A comment continues to the end of the line. Everything between the semicolon and the end of the line is ignored.

Two-Level Section Names

JADE supports a two-level hierarchy of section names within the JADE initialization file. To achieve this, you can prefix a section name with a unique identifier that is supported when a JADE program detects the optional **name=*unique-identifier*** command line argument.

For example, you can define two qualified [JadeAppServer] JADE initialization file sections called **[SecureAS.JadeAppServer]** and **[InternalAS.JadeAppServer]**, with the **RPCEncryptionEnabled** parameter in the **[InternalAS.JadeAppServer]** section set to **false** and the same parameter in the **[SecureAS.JadeAppServer]** section set to **true**. You could then have one JADE application server executable (**jadapp**) shortcut with **name=SecureAS** in the command line and another shortcut with **name=InternalAS** in the command line.

The *unique-identifier* value is case-sensitive, cannot include a dot character (**.**), and is limited to 100 characters.

If the **name** argument is present in the command line of the JADE program, the qualified section name (for example, **[SecureAS.JadeAppServer]**) of the JADE initialization file is checked first, followed by the single-level section name (for example, **[JadeAppServer]**). If the **name** argument is not specified on the command line of the JADE program, any JADE initialization file sections that contain a two-level section name are ignored. See also "Placing Initialization File Parameters on the Command Line", in Chapter 1 of the *JADE Installation and Configuration Guide*.

To enable JADE methods to access initialization files, the following must apply.

1. The JADE initialization file *jade-initialization-file* value must be case-identical to the argument value obtained from the command line (that is, **ini**=*jade-initialization-file* or its default location).

2. The first character of the section name must not be a dot (**.**) character.

If these conditions are not met, the code proceeds as if the **name** argument is not present on the command line.

Excluding the dot character (**.**) used to separate the *unique-identifier* value and the standard JADE initialization file section name, JADE assumes that the section name does not include a dot character.

If a dot character other than the two-level separator is detected, it is treated as a single entity. For example, the section name [joe.bloggs] could lead to confusion between accessing a [bloggs] section when the shortcut command line contains the **name**=**joe** argument and accessing the section [joe.bloggs] when there is no **name** argument specified in the command line. If **name**=**joe** is defined and the [joe.bloggs] JADE initialization file section is being accessed, the two-level behavior first checks for a [joe.joe.bloggs] section and then for a [joe.bloggs] section if the [joe.joe.bloggs] section is not present.

For more examples of JADE's handling of the unique section name identifier, see the example under "Supporting Application Class Method Calls and Global Constants", later in this chapter.

When a JADE program whose command line contains the **name=*unique-identifier*** argument updates the JADE initialization file, the qualified section is changed. If there is no qualified two-level JADE initialization file section, the appropriate qualified section is created. If a value is not present in either the qualified or the unqualified section, the value is created in the unqualified section.

Notes You can apply a qualified unique identifier to the optional [JadeCommandLine] section of the JADE initialization file (for example, **[SecureAS.JadeCommandLine]**). For details, see "JADE Command Line Section [JadeCommandLine]", later in this chapter.

You cannot define command line arguments for the **jdbutilb** batch JADE Database utility program in a [JadeCommandLine] section because the format of the command line differs from that of other JADE programs. However, the use of the **name** argument is supported on the **jdbutilb** command line and normal rules apply when the batch JADE Database utility program accesses JADE initialization file values.

For details about placing JADE initialization file parameters on the command line, see "Placing Initialization File Parameters on the Command Line" under "Specifying Arguments in the JADE Command Line", in Chapter 2 of the *JADE Installation and Configuration Guide*. See also "Handling of Parameter Values" and "Sharing JADE Initialization Files", later in this chapter.

The following example shows a sample portion of a JADE initialization file, which assumes that **name**=**Example** is specified on the JADE application server executable (**jadapp**) shortcut command line.

```
[JadeAppServer]
KeepAliveTime=10
ConnectionTimeout=0
EnableAppRestrictions=false
AllowSchemaAndApp1=<schema, application>
WindowPos=0,0,500,100
AutomaticDownLoad=true
CacheEntryTimeout=30
PictureCacheFile=<default>
RPCEncryptionEnabled=false
RPCEncryptionHookDLL=Internal

[Example.JadeAppServer]
NodeName=jadapp-6006
NodeNameDescription=JADE AppServer 6006 (\system)
WindowPos=1,49,924,136
```

Original values that are not updated (for example, **AllowSchemaAndApp1**) stay in the original [JadeAppServer] section. Values that are updated (for example, **WindowPos**) are put into the new [Example.JadeAppServer] section.

Handling of Parameter Values

JADE handles initialization file parameter values as follows.

- The value for a JADE initialization file section and parameter pair can be **<default>** (which is case-insensitive). This applies to all primitive types; for example, **Boolean**, **Integer**, or **String**.

  If you specify a parameter value of **<default>**, JADE uses what it recognizes as the correct default value. This enables you to have a parameter in your initialization file and have a reasonable default value used so that if the default value changes between releases, you do not have to change the parameter value to ensure that an out-dated default value is not used.

- If the **Integer** values of some parameters are sizes (for example, a maximum file size or a buffer size), you can replace a large value decimal notation such as **22020096**, which is difficult to read, with the corresponding measure of storage and memory, or **21M** in this example. As simple numbers continue to be supported, you do not have to change existing parameter values unless you want to do so.

  A number can be followed by a case-insensitive **K**, **M**, **G**, **T**, or **P** *prefix multiplier*. This multiplies the prefix number by **1024** (1 kilo), **1048576** (1 mega), **1073741824** (1 giga), **1099511627776** (1 tera), or **1125899906842624** (1 peta), respectively.

  Note These prefix multipliers are binary multiples; that is, they are multiples of 1,024 rather than of 1,000.

  As the number must be an integral value (which has no decimal points), a value of **1.5M** is treated as **1** and not as **1048576**. You can use **1572864** or **1536K** if you want a value of one and a half megabytes. If the specified number is too large, JADE reduces this to the appropriate value; for example, if you specify an **ObjectCacheSizeLimit** parameter value of **10G** bytes, this is reduced to **2^32** (**4294967296**, or **4G**) bytes.

Note These prefix multipliers apply only to JADE initialization file parameters and sections defined by JADE (that is, this facility is not supported in parameters and sections that you define).

Changing JADE Initialization File Parameters

To change the JADE initialization file parameters

1. Use a text editor (for example, Notepad) to edit the JADE initialization file directly.

   Do not use a formatting editor such as Write, as it can corrupt your JADE initialization file. For more details about Notepad, see Help for Notepad.

2. After you have edited the JADE initialization file, you must perform one of the following actions to enable the parameters to take effect.

   - Close and reopen the relevant JADE database.
   - Restart your JADE application (that is, JADE).

Caution Always back up your JADE initialization file before you make changes, so that you can restore the original file if you accidentally damage the file or make changes that cause problems when running JADE.

You should not change any parameters that are not described in this document. However, if you need to change additional parameters, refer to the relevant documentation. Be careful when using a text editor to edit the JADE initialization file. Incorrect changes can lead to unexpected results.

Default Values for Parameters

The JADE Initialization file that is created when JADE is installed contains a small number of parameters. Subsequently, when a JADE program (for example, the database server, the application server, a standard or presentation client, or one of the utilities) is run, additional parameters are written to the JADE initialization file. For example, after installing JADE and running **jade.exe**, the [JadeClient] section is as follows.

```
[JadeClient]
ServerNodeSpecifications=TcpIpAny,LocalHost,6005
Language=<default>
ObjectCacheSizeLimit=<default>
TransientCacheSizeLimit=<default>
AutomaticCacheCoherency=<default>
SamplingNode=<default>
ReadOnlySchema=<default>
ReadOnlySystemSchema=<default>
InitializationHandlerLibrary=<default>
NotifyQueueDepthWarningThreshold=<default>
```

The value of **<default>** specifies that the parameter has the default value for the release of JADE that is installed. In many cases, the default value does not change from one release to the next. One case where values can change is when a feature requires additional memory or disk storage in a new release. In that case, the **<default>** value for the corresponding parameter will increase.

You can look up the actual value for a parameter with a value of **<default>** in the *JADE initialization File Reference*.

You can override a value of **<default>** with a specific value, as described in the previous section, "Changing JADE Initialization File Parameters".

Updating the JADE Initialization File from Previous Versions

When the JADE Object Manager is initialized without a JADE initialization file, a default version of the file is created automatically in your current JADE database directory (for example, **c:\jade\system\jade.ini**). Various parameters are added to the file with their default values.

If a later version of JADE is installed and an existing version of the JADE initialization file is found when the database is opened, some new entries can be added but unused parameter settings are not deleted automatically.

Sharing JADE Initialization Files

Multiple JADE programs on the same host can share a JADE initialization file. In addition, the command line arguments can be treated as logically part of the initialization file. This enables you to specify arguments on the command line that would normally be defined as parameters in the JADE initialization file. Conversely, you can define parameters in the [JadeCommandLine] section of the initialization file that would normally be arguments specified on the command line.

To enable the sharing of JADE initialization files, section names in the initialization file include a prefix, which is obtained from the **name** command line argument. For details, see "Two-Level Section Names", earlier in this chapter. The sharing of JADE initialization files is useful for the management of JADE servers, by reducing the work required in maintaining multiple initialization files, one for each JADE server on the same host. To use the same initialization file for all clients and servers for a JADE system, each program shortcut has an appropriate identifier defined.

Placing command line arguments in the JADE initialization file simplifies the command line shortcut, by putting the details in the initialization file.

In a thin client environment, a new JADE initialization file can be downloaded as part of a version upgrade, allowing the system administrator to reconfigure the thin client so that it could connect to a different application server and removing the need to supply user instructions about the changes that the user needs make to the JADE application shortcut.

Supporting Application Class Method Calls and Global Constants

Two-level section name behavior applies to JADE initialization file sections if the section name does not start with a dot character (.) and if the first argument to the **Application** class:

- **getProfileString** or **setProfileString** method exactly matches the output of the **Application** class **getIniFileName** method
- **getProfileStringAppServer** or **setProfileStringAppServer** method exactly matches the output of the **getIniFileNameAppServer** method

For details, see "Two-Level Section Names", earlier in this chapter.

If you write JADE code similar to the following, it expects to access the same JADE initialization file that the JADE node is using.

```
app.getProfileString(app.getIniFileName, "JadeClient", "ServerName", null);
```

If you share a JADE initialization file among multiple nodes, to ensure that one node will not change values updated by another node, prefix the section names with a dot character (.) to allow accessing of the same section, if required. Conversely, if a user has his or her own JADE initialization file, you can ensure that behavior is not changed, by prefixing sections names with a *unique-identifier* value.

Calls to the **Application** class **getProfileString**, **getProfileStringAppServer**, **setProfileString**, or **setProfileStringAppServer** method obtain values from or set values on the command line if they are present.

The **ProfileRemoveKey** and **ProfileRemoveSection** global constants in the **JadeProfileString** category remove a parameter (key string) or an entire section from a JADE initialization file, as shown in the following example.

```
begin
    app.setProfileStringAppServer("\Jade\system\jade.ini", "mySection",
                                  ProfileRemoveSection, "");
    // JADE system adds a value of name= (for example, "InternalAS.")
    app.setProfileStringAppServer(app.getIniFileName, "JadeAppServer",
                                  "PictureCacheFile", ProfileRemoveKey);
    // Initialization file name not IDENTICAL, so we can supply a prefix
    // If the user has moved the window, reset it to the default values
    app.setProfileStringAppServer("\jade\system\JADE.ini",
                                  "Test.JadeAppServer", "WindowPos",
                                  ProfileRemoveKey);
    // JADE uses the single-level (top level) section name
    app.setProfileStringAppServer("\Jade\system\Jade.ini",
                                  ".JadeAppServer", "EnableAppRestrictions",
                                  "true");
end;
```

Database Backup Section [DatabaseBackup]

The [DatabaseBackup] section of the JADE initialization file contains parameters that store the settings specified in the Backup Database dialog.

Notes Set these parameters only by using the appropriate controls in the Backup Database dialog. For details, see "Backing Up Your JADE Development Environment", in Chapter 3 of the *JADE Database Administration Guide*. These values are stored in the JADE initialization file so that they can be reused the next time the Backup Database dialog is invoked.

All parameters in this section are read when the Backup Database dialog is next displayed.

The section names used to store the settings specified in the Backup Database dialog for re-use in subsequent database backups are listed in the following table; that is, if the Backup Database dialog is invoked and a backup performed from a database that has a secondary database native subrole, that parameters are created and stored in the [SecondaryDatabaseBackup] section.

Database Role	SDSDatabaseRoles Global Constant Value	JADE Initialization File Section
Non-SDS database	SDS_RoleUndefined	[DatabaseBackup]
Primary database	SDS_RolePrimary	[DatabaseBackup]
Native JADE secondary database	SDS_SubroleNative	[SecondaryDatabaseBackup]
RPS database	SDS_SubroleRelational	[RpsDatabaseBackup]

The database backup section (for example, [DatabaseBackup]) can contain the following parameters.

BackupDirectory

Value Type String *(disk-path)*

Default Not specified

Purpose

The **BackupDirectory** parameter contains the last value of the backup directory used in the Backup Database dialog. The following example uses a network drive as the backup directory.

```
BackupDirectory = \\wilbur1a\cdrive\jade\system\backup
```

BackupMode

Value Type String

Default Not specified

Purpose

The **BackupMode** parameter contains the last value of the option button used in the Backup Mode group box in the Backup Database dialog; that is, **Quiesced** (read-only) or **Hot** (updating allowed).

CompressFiles

Value Type Boolean

Default Not specified

Purpose

The **CompressFiles** parameter contains the last value used in the **Compress Files** check box in the Backup Database dialog.

OverwriteExistingFiles

Value Type Boolean

Default Not specified

Purpose

The **OverwriteExistingFiles** parameter contains the last value used in the **Overwrite Existing Files** check box in the Backup Database dialog.

VerifyFiles

Value Type Boolean

Default Not specified

Purpose

The **VerifyFiles** parameter contains the last value used in the **Verify Files** check box in the Backup Database dialog.

Database Initialization Sections

The JADE initialization file [PersistentDb] and [TransientDb] sections contain information used to initialize individual database engines on each JADE node.

The persistent database engine uses the information contained in the [PersistentDb] section of the JADE initialization file and the transient database engine uses the information contained in the [TransientDb] section. For details about the parameters that enable you to configure JADE in a Synchronized Database Service (SDS) environment, see "Synchronized Database Service Sections", later in this chapter.

Disk Cache Memory Parameters

The JADE database engine can make effective use of memory assigned to file caching to avoid or postpone input-output operations in order to improve database performance.

However, if the total memory in use on your computer exceeds the physical memory available, the negative performance impact incurred by operating system paging activity may more than offset the benefits of this caching. For this reason, disk cache memory parameters need to be carefully selected to make effective use of available physical memory. One way to select optimal values for memory parameters is by testing the effect of changes in the target environment. Effects can be determined by timing specific units of work and by using the database statistics **pdb_statistics.log** file.

As certain parameters interact with each other (and indeed with other operating system-configurable options), it may be necessary to make multiple passes, stabilizing each parameter in turn.

Persistent Database Initialization Section [PersistentDb]

The [PersistentDb] section of the JADE initialization file contains parameters that affect the operation of the JADE Object Manager physical database engine.

The [PersistentDb] section can contain the following parameters.

ActivityLogDirectory

Value Type String *(disk-path)*

Default Value of the **LogDirectory** parameter in the [JadeLog] section

Purpose

The **ActivityLogDirectory** parameter specifies the location of activity logs produced by database administration operations such as a database backup, restore, or file compaction operation. (For details, see "Using Database Operational Commands", in Chapter 1 of your *JADE Database Administration Guide*.)

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

If you specify a relative directory name, the directory is assumed to be relative to the JADE HOME directory. For example, if your installation directory is **Jade\bin** (that is, your JADE HOME directory is **Jade**) and you specify **ActivityLogDirectory=ActivityLogs**, the full directory path is **Jade\ActivityLogs**.

BackupBlockSize

Value Type Integer *(prefix multiplier)*

Default 64K

Purpose

The **BackupBlockSize** parameter specifies the buffer and I/O (input-output) size to be used for reading database files during backup processing.

You can set the block size to values in the range **64K** through **512K** bytes.

Parameter is read when …

The backup process begins.

Hint

Increasing the value of this parameter may improve performance if your machine has a substantial amount of available memory, depending on the mix of active applications. This parameter enables you to decide the values that provide the best trade-off between CPU and memory utilization and the time taken to perform the backup operation.

BackupBuffers

Value Type Integer

Default 4

Purpose

The **BackupBuffers** parameter specifies the number of buffers to be used when reading database files during backup processing. Set the number of buffers to values in the range **2** through **10**.

Parameter is read when …

The backup process begins.

Hint

Increasing the value of this parameter may improve performance if your machine has a substantial amount of available memory, depending on the mix of active applications.

This parameter enables you to decide the values that provide the best trade-off between CPU and memory utilization, and the time taken to perform the backup operation.

BackupCompressedFileGrowthIncrement

Value Type Integer *(prefix multiplier)*

Default 1G

Purpose

The **BackupCompressedFileGrowthIncrement** parameter specifies the size of the increment by which a backup file increases during a compressed backup. If the value supplied is not a multiple of the value of the **BackupBlockSize** parameter, it is rounded up to the next highest multiple.

The minimum value is **64M** bytes. If the file to be backed up is smaller than the value of the **BackupCompressedFileGrowthIncrement** parameter, the backup file is preallocated to that size.

The maximum value is **64G** bytes.

Parameter is read when …

The backup process begins.

BackupCompressionLevel

Value Type String

Default Normal

Purpose

The **BackupCompressionLevel** parameter specifies the extent of data compression to be attempted during a compressed database backup. You can set the compression level to **Fast** (minimal compression), **Normal** (the default), or **Maximum** (maximal compression).

Parameter is read when …

The backup process begins.

Hint

This parameter enables you to determine the best trade-off between the time taken to perform the compression of a file and the compression ratio that is achieved. The **Fast** setting provides the best speed, while the **Maximal** setting provides the best compression in most cases.

BackupIndexCrosscheckDisabled

Value Type Boolean

Default false

Purpose

By default, offline and quiesced backups perform index block to target (object, subobject, and so on) verification, ensuring that index entries resolve to the correct records in the file.

If you would rather have an efficient backup process that fits comfortably in the processing window, you can set the **BackupIndexCrosscheckDisabled** parameter to **true** and then perform the JADE Database utility **certify** action after the backup, to certify the database using a temporary relocated (and checksum-verified) copy of the backup.

The **BackupIndexCrosscheckDisabled** parameter defaults to **false**, so that database files are verified as they are backed up, because most users do not certify their databases.

Setting the **BackupIndexCrosscheckDisabled** parameter to **true** disables the index block to target verification, which has a considerable performance cost.

Parameter is read when …

The backup process begins.

BackupOrphanCheckDisabled

Value Type Boolean

Default true

Purpose

Target (object, subobject, and so on) to index verification ensures that entities in the database file can be reached by their index. File certification always performs target to index verification. By default, offline and quiesced backups do *not* perform target to index verification.

Setting the **BackupOrphanCheckDisabled** parameter to **false** enables target to index verification, which has a considerable performance cost.

Parameter is read when …

The backup process begins.

BackupStats

Value Type Boolean

Default false

Purpose

When the **BackupStats** parameter is set to **true**, the database file backup procedures log statistical information to the **backup.log** file.

Parameter is read when …

The backup process begins.

Hint

Backup statistics are logged only when you set this parameter to **true**, and they are always logged to the **backup.log** file.

BackupThreadPriority

Value Type String

Default Normal

Purpose

The **BackupThreadPriority** parameter specifies the priority to be set on the threads performing the database backup. You can set the thread priority to **Lowest**, **Low**, **Normal** (the default), **High**, or **Highest**.

Parameter is read when …

The backup process begins.

Hint

The priority of the backup threads can be lowered so that backup processing has a lesser impact on your system. This may be beneficial when you are performing an online updating backup and you want minimal impact on the transaction response time of your system.

Alternatively, you can increase the priority of the backup threads so that the backup processing consumes more of your system's resources. This may be beneficial when you are performing an offline backup and you want to try to minimize the time required to complete the database backup.

DefFileGrowthIncrement

Value Type Integer *(prefix multiplier)*

Default 128K

Purpose

The **DefFileGrowthIncrement** parameter specifies the size in bytes by which the database files are incremented as they grow (that is, the logical extent size).

You can set the growth increment to values in the range **64K** through **4G** bytes.

For details about the JADE Database utility **Set File Attributes** operation that enables you to override this value for selected files, see "Using the Set File Attributes Command" or "**fileAttributes**, in Chapter 1 of the *JADE Database Administration Guide*.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

As database files grow in small variable increments on demand and the gradual growth incurs both a direct I/O performance penalty and an indirect performance penalty due to disk fragmentation, use this parameter to tailor the amount by which your database files grow.

DefInitialFileSize

Value Type Integer *(prefix multiplier)*

Default 128K

Purpose

The **DefInitialFileSize** parameter specifies the size to which database files are preallocated when they are first created. You can set the initial file size to values in the range **64K** through **16,384P** bytes.

For details about the JADE Database utility **Set File Attributes** operation that enables you to override this value for selected files, see "Using the Set File Attributes Command" or "**fileAttributes**, in Chapter 1 of the *JADE Database Administration Guide*.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

This parameter enables you to preallocate database files at close to the expected file size.

Pre-allocation provides the opportunity to mitigate the usual overheads, including file system fragmentation (since files can be defragmented prior to use).

DisableAutoReindex

Value Type Boolean

Default false

Purpose

The **DisableAutoReindex** parameter is used to disable automatic reindexing of database files by setting the parameter to **true**. In normal operation of the database, the parameter should not be changed from the default setting of **false**.

When you set this parameter to **true**, recovery is terminated and exception 3113 (*Database recovery not possible - fatal error encountered)* is raised when restart recovery encounters an error redoing or undoing an index update operation. A person who performs the database administration role must then restore the database from backup and perform a roll-forward recovery.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

DiskCacheBlocksPerSegment

Value Type Integer

Default 8192

Purpose

The **DiskCacheBlocksPerSegment** parameter specifies the segment size used by the disk cache.

A segment is the unit of memory that is allocated and deallocated. Memory in a segment is contiguous. The size of a segment is expressed in terms of the number of **8192** byte blocks it can contain.

The minimum value is **2048** (for a segment size of **16M** bytes), the default value is **8192** (for a segment size of **64M** bytes), and the maximum value is **32768** (for a segment size of **256M** bytes).

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

DiskCacheFreeMemoryTarget

Value Type Integer

Default 512M

Purpose

The **DiskCacheFreeMemoryTarget** parameter specifies the amount of physical memory the disk cache will try to keep free while honoring pool minimum and maximum segment settings. The minimum value is **32M** bytes, the default value is **512M** bytes, and the maximum value is one quarter of physical memory.

Because the default value of **512M** could be larger than a quarter of physical memory on a virtual machine configured with 2G bytes, to support virtual machines (for prototyping, testing, and so on), the default value of this parameter for virtual machines is the minimum of 512M or a quarter of physical memory. For production databases, the minimum physical memory that is supported is 2G bytes.

Free physical memory is calculated at regular intervals, the length of which is specified by the value of the **DiskCacheMemorySampleInterval** parameter.

Parameter is read when …

Every 30 seconds.

DiskCacheMaxSegments

Value Type Integer

Default Determined by physical memory

Purpose

The **DiskCacheMaxSegments** parameter specifies the maximum number of segments that the disk cache is to use.

The value of the **DiskCacheMaxSegments** parameter cannot be less than the value specified for the **DiskCacheMinSegments** parameter.

The default value is calculated from the size of physical memory. The available memory is half of the size of physical memory divided by **8K** bytes (the disk block size), to yield the number of blocks it can contain. The default value for the **DiskCacheMaxSegments** parameter then becomes that number of blocks divided by the number of blocks per segment.

The maximum value is also calculated from the size of physical memory. When there is **2G** bytes or less of physical memory, the available memory is two-thirds of the size of physical memory; otherwise, it is the size of physical memory less **256M** byes. This available memory figure is then divided by **8K** bytes (the disk block size), to yield the number of blocks it can contain. The maximum value for the **DiskCacheMaxSegments** parameter then becomes that number of blocks divided by the number of blocks per segment.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

DiskCacheMemorySampleInterval

Value Type Integer *(milliseconds)*

Default 5000

Purpose

The **DiskCacheMemorySampleInterval** parameter specifies the interval, in milliseconds, between successive calculations of free physical memory.

The minimum value of zero (**0**) indicates that calculations are disabled. The maximum value is **30000** (that is, 30 seconds).

Parameter is read when …

Every 30 seconds.

DiskCacheMinSegments

Value Type Integer

Default 4

Purpose

The **DiskCacheMinSegments** parameter specifies the minimum number of segments the disk cache is to use, and it is the initial pool size.

The minimum value is **1**, the default value is **4**, and the maximum value is limited by physical memory, as follows. When there is **2G** bytes or less of physical memory, the available memory is determined to be two-thirds of physical memory; otherwise it is four-fifths of physical memory. This available memory figure is then divided by **8K** bytes (the disk block size), to yield the number of blocks it can contain.

The maximum value allowed for the **DiskCacheMinSegments** parameter then becomes that number of blocks divided by the blocks per segment.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

DiskCacheWriteOnlyDbSegments

Value Type Integer

Default 16 (but determined by available physical memory)

Purpose

The **DiskCacheWriteOnlyDbSegments** parameter specifies the minimum and maximum number of segments that the disk cache is to use when the database mode is write-only; for example, during upgrades.

For the minimum and maximum values, refer to the **DiskCacheMinSegments** and **DiskCacheMaxSegments** parameters, respectively.

Note The value of this parameter can be constrained by the actual maximum number of segments calculated during initialization.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

EnableArchivalRecovery

Value Type Boolean

Default true

Purpose

The **EnableArchivalRecovery** parameter is used to enable or disable the archival recovery option.

The default value of **true** is the recommended value for productions systems. For non-critical systems, where transaction loss is acceptable, you can opt out of the ability to do archival recovery by setting the value of this parameter to **false**.

When you set this parameter to **false** and you perform an online backup, the journal set that the backup spanned is copied to the location of the backup. This journal set is automatically restored, if required, when the database is restored.

This parameter is ignored on an SDS primary system, for which archival recovery is always enabled. The parameter does not apply to an SDS secondary system.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

EnableDeltaLogging

Value Type Boolean

Default true

Purpose

The **EnableDeltaLogging** parameter is used to enable or disable delta logging for the database in the JADE system in which the journals were produced (that is, *not* in the JADE system in which they are analyzed). Delta logging employs a single audit record, which is used to redo or to undo the update operation that it describes. When delta logging is enabled (the default), object updates are audited using a compact delta description of the changes referred.

When you disable delta logging by setting the **EnableDeltaLogging** parameter to **false**, a complete image of the object before the update and a complete image of the object after the update are stored in separate audit records.

Note When the **EnableDeltaLogging** parameter is set to **false**, the **UseJournalDescriptions** parameter is ignored.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

FileNumberHeadroom

Value Type Integer

Default 2048

Purpose

The **FileNumberHeadroom** parameter specifies by how much the highest allocated file number can be increased in the current database server session (in the JADE development environment or by using the **jadload** or **jadloadb** program) without restarting the database server.

The highest database file number cannot be greater than **65,484**.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

If you anticipate that you will add database files that would increase the highest-used file number by more than the default value of 2,048 in the current work session, increase this value so that you do not have to restart the database server. (Exception 3003 is raised when the value defined for this parameter is exceeded in a work session.)

IndexLoadFactor

Value Type Integer

Default 66

Purpose

The **IndexLoadFactor** parameter specifies the percentage of entries that are retained when an index block becomes full and is split. A value of **95** means that 95 percent of entries are retained in the current block and 5 percent of entries are moved to a new block.

Statistically, a 66 percent load factor provides optimal loading when entries are added in random key order and a higher load factor (for example, 95 percent) provides better loading when entries are added in sequential key order.

The minimum parameter value is **50** percent, and the maximum parameter value is **95** percent.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

JournalArchiveDirectory

Value Type String *(disk path)*

Default *journal-root-directory*\archive

Purpose

The **JournalArchiveDirectory** parameter specifies the fully qualified directory for the location of archived transaction journal files for recovery purposes.

If you do not specify a value for this parameter, the **JournalRootDirectory\archive** directory is used.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

JournalCloseAction

Value Type String

Default Move

Purpose

The **JournalCloseAction** parameter specifies what to do for transaction abort processing or restart recovery processing when a journal file that is no longer required is closed and released.

When archival recovery is enabled, the close actions that you can specify are listed in the following table.

Journal Close Action	Processing
None	No action taken.
Remove	Removes the journal file.
Copy	Uses the operating system to copy the file to the archive directory.
Move	Uses the operating system to move the file to the archive directory.
CopyAndCompress	Programmatically creates a compressed copy of the file in the archive directory (using ZLIB compression mechanisms).
MoveAndCompress	Programmatically creates a compressed copy of the file in the archive directory (using ZLIB compression mechanisms) and then removes it.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

JournalMaxSize

Value Type Integer *(prefix multiplier)*

Default 64M

Purpose

The **JournalMaxSize** parameter specifies the size of the transaction journal file used for restart or roll-forward recovery. A journal switch occurs when the journal exceeds this size. You can set the maximum journal size to a value in the range **1M** through **2G** bytes.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

Increase this value if you want to reduce the number of journal switches.

JournalMinSize

Value Type Integer *(prefix multiplier)*

Default 0

Purpose

The **JournalMinSize** parameter specifies the minimum journal size that must be attained before the interval-based switch is honored.

Note The **JournalMinSize** parameter is ignored if the value of the **JournalSwitchInterval** parameter is the default value of zero (**0**).

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

If your site finds time-based switching useful but you do not want empty journals generated, set the **JournalMinSize** parameter to a value other than zero (**0**). For information about journal switching, see "Journal Switches and Journal Numbers", in Chapter 3 of the *JADE Database Administration Guide*.

JournalRootDirectory

Value Type String *(disk path)*

Default *database-directory*\journals

Purpose

The **JournalRootDirectory** parameter specifies the root directory for your recovery journal files. The directory name can be a relative or an absolute path. The value of **<default>** corresponds to a relative directory name of **journals**.

Caution The database does not expect the output of other activities to be sharing this directory. Conflicting usages may result in the accidental removal of files required by the database.

Hint

If you specify a relative directory name, the directory is assumed to be relative to the JADE database directory. For example, if your database directory is **Jade\system** and you specify **JournalRootDirectory=journals**, the full directory path is **Jade\system\journals**.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

JournalSwitchInterval

Value Type Integer *(minutes)*

Default 0

Purpose

The **JournalSwitchInterval** parameter specifies the number of minutes after which the journal is switched, regardless of content.

When the journal switches because of interval expiration, the journal file is truncated at the appropriate upper extent boundary.

When this parameter is set to the default value of zero (**0**) minutes, journal switching is disabled and the **JournalMinSize** parameter is ignored.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

You can use this parameter in conjunction with the **JournalMinSize** and **JournalMaxSize** parameters to support your own disaster recovery automation.

JournalSyncStats

Value Type Boolean

Default false

Purpose

The **JournalSyncStats** parameter specifies whether synchronously related I/O wait times are accumulated and logged when the journal file is closed.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

MaxWaitForQuietPoint

Value Type Integer

Default 5

Purpose

The **MaxWaitForQuietPoint** parameter specifies the maximum time in seconds that the database waits for a quiet point in the following situations.

- Quiescing the database for a quiesced backup or changing the access mode to archive.
- Changing the access mode of a file to read-only by using the **DbFile** class **changeAccessMode** method.
- When locking files that are about to be reorganized. If a reorganization is audited, transactions are temporarily blocked and all files that are going to be reorganized are locked, allowing updating transactions to all files except those that are locked.
- Performing an SDS take-over operation.

If the performing of one of these operations does not correspond to a natural quiet point, the **MaxWaitForQuietPoint** parameter specifies the maximum time in seconds that the database waits for existing transactions to complete before performing the operation after temporarily suspending new transactions from starting.

If a quiet point is not achieved within the time specified in this parameter, a **DB_QUIETPOINT_TIMEOUT** exception is raised.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hints

The minimum value is zero (**0**) and the maximum value is **600** seconds (10 minutes).

Under a heavy transaction load, the waiting operation may not be possible when this parameter is set to the default value of zero (**0**), as a quiet point never occurs.

If you are waiting too long for an operation to be performed, use this parameter to force a quiet point so that the operation can take place.

OpenAllFilesOnDbOpen

Value Type Boolean

Default false

Purpose

The **OpenAllFilesOnDbOpen** parameter, when set to **true**, causes all database files defined in the **_control.dat** database control file (including all online partitions associated with a partitioned database file) to be opened and the timestamps checked when the database is opened.

Note During the upgrade to a new JADE release, the database upgrade process overrides the **OpenAllFilesOnDbOpen** value if it is set to **true**.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hints

Use this parameter if you want your file timestamps checked when the database is opened rather than when the files are referenced.

Setting this parameter to **true** may result in an increase in the time required to open the database, especially in systems that have many map files. If any file is missing, the database open fails.

File timestamp mismatches are recorded in the **jommsg*n*.log** file.

PrintStatistics

Value Type Boolean

Default false

Purpose

The **PrintStatistics** parameter, when set to **true**, causes database usage statistics to be output to a file when the database is closed. Persistent database statistics are output to the **pdb_statistics.log** file. By default, the printing of statistics is disabled.

New statistics are always appended to an existing statistics file.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hints

Set this parameter to **true** if you require database statistics for the purposes of tuning or performance analysis.

When using statistics output to the **pdb_statistics.log** file, note that a non-zero **Aborted transactions** value indicates that transactions are overflowing the client cache and aborting, which impacts on performance. If this figure is too high, investigate why this is happening in your application. This value should be a low number or zero (**0**).

ResourceErrorRetryDelay

Value Type Integer

Default 1000

Purpose

The **ResourceErrorRetryDelay** parameter specifies the time in milliseconds between retries of a database I/O operation that failed because insufficient operating system resources were available.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

ResourceErrorRetryLimit

Value Type Integer

Default 2

Purpose

The **ResourceErrorRetryLimit** parameter specifies the number of times a database I/O operation that failed because insufficient operating system resources were available is retried after the initial failure.

Specify a value of zero (**0**) to indicate that no retries occur.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

RetryResourceErrors

Value Type Boolean

Default true

Purpose

The **RetryResourceErrors** parameter enables or disables the retrying of database I/O operation that fail through insufficient operating system resources being available.

By default, retrying of failed database I/O operations is enabled. If the parameter is set to **false**, an exception (Insufficient system resources) is raised immediately.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

While an operation is retried, the database engine continues to operate (possibly with degraded performance). During the retry interval, the operating system by itself or aided by a system administrator may be able to free up sufficient resources.

StateOverflowDirectory

Value Type String *(disk-path)*

Default *database-directory*\PDB_OSM

Purpose

The **StateOverflowDirectory** parameter specifies the directory used by the Object State Manager module in the database engine to locate the temporary files used to overflow information about committed object states. The directory name can be a relative or an absolute path. If you specify a relative directory, it is relative to the default database directory. The default value for this parameter is **PDB_OSM**-relative to the database directory.

As it is not possible to share a common directory between multiple database instances because file names are not unique, you must ensure that you provide each JADE database with a unique location by using this parameter. The default location is guaranteed to be unique.

On a primary or non-SDS-capable database, the information preserved in the files in this directory is used to provide access to the prior committed states of uncommitted modified objects that have overflowed the persistent database cache.

On an SDS secondary database that has read-only processes accessing the database, the information in these files is used to provide access to the prior committed states of objects modified by replay transactions that are being isolated.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

SuppressDialogs

Value Type Boolean

Default false

Purpose

The **SuppressDialogs** parameter, when set to **true**, causes the suppression of all message dialogs resulting from persistent database exceptions (for example, out of disk, missing files, and allow file replace messages). The default value of **false** displays database message dialogs.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hints

Set this parameter to **true** if you do not want message box dialogs raised on an unattended server; for example, when you are running an operation as an automated background task. When message dialogs are suppressed, exceptions are raised and output to your message log file.

When the server is running as a service, this parameter is ignored and dialogs are always suppressed. Messages are output to your message log file.

SystemFileDirectory

Value Type String *(disk-path)*

Default Installation directory

Purpose

The **SystemFileDirectory** parameter enables you to specify a different location for your system map files (that is, the **_system.bin**, **_sysgui.bin**, **_sysxrf.bin**, **_sysint.bin**, **_sysdef.bin**, **_sysdev.bin**, **_systools.bin**, **_ jadeapp.bin**, and **_ jadedef.bin** files). JADE looks for these files in the directory specified by this parameter. The location of other database files is not affected.

The default value of **SystemFileDirectory=<default>** specifies that JADE looks for the system map files in the installation directory (commonly known as the **bin** directory).

If the system map files are located in a different directory, use this parameter to specify the name of the directory in which the files are located; for example:

```
SystemFileDirectory=\sysdb
```

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Hint

Use this parameter when you have multiple JADE databases and you want to conserve disk space. Because the system map files are common to every database, you can put them into a common directory and share them across all databases, by using this parameter.

The database path cannot be greater than 260 characters.

UseJournalDescriptions

Value Type Boolean

Default Not specified

Purpose

The **UseJournalDescriptions** parameter activates the automatic matching of a description of the JADE environment (that is, the classes and properties) with the audit journal records used by the **JadeAuditAccess** framework.

The description file used to identify classes and properties is automatically created only when the **UseJournalDescriptions** parameter exists and it is set to **true**.

For details about the naming convention and location of the generated journal description files, see the **JadeAuditAccess** class **generateDescription** method.

Note When the **EnableDeltaLogging** parameter is set to **false**, the **UseJournalDescriptions** parameter is ignored.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

VerifyJournal

Value Type Boolean

Default false

Purpose

The **VerifyJournal** parameter enables you to specify that your active (or online) journal file is verified when the journal becomes full and is transferred offline. This operation verifies the physical integrity of your transaction journal file after it has been transferred, performing a series of tests that verify the contents of records, including various cross-checks between adjacent records.

If an error is detected during the verification process and the **SuppressDialogs** parameter is set to **false** (the default), a message is displayed advising you of this. Any error that is detected is output to your ***journal-file-name*.scan** file (for example, **db0000000009.scan**), regardless of the setting of the **SuppressDialogs** parameter.

A progress report while the operation is running and notification of completion at the end of the operation are output to your **jommsg*n*.log** file.

Parameter is read when …

The database server node is initialized; for example, when you restart the database server.

Transient Database Initialization Section [TransientDb]

The transient database engine of the JADE Object Manager provides a subset of functionality of the persistent database engine. It is used to provide transient (non-persistent) temporary object storage for the JADE Object Manager client.

In addition to the **HighestFileNumber** transient database-specific parameter that is documented in full in the following subsection, the [TransientDb] section of the JADE initialization file can contain the following parameters that are a subset of the initialization file parameters in the [PersistentDb] section.

- ActivityLogDirectory
- DefFileGrowthIncrement
- DefInitialFileSize
- PrintStatistics
- ResourceErrorRetryDelay
- ResourceErrorRetryLimit
- RetryResourceErrors

For details about these parameters, see "Persistent Database Initialization Section [PersistentDb]", earlier in this chapter.

Transient database statistics are output to the **tdb_statistics.log** file, located in the path specified by the **ActivityLogDirectory** parameter.

HighestFileNumber

Value Type Integer

Default 2048

Purpose

The **HighestFileNumber** parameter specifies the maximum number of transient overflow files that can be opened on the node; that is, it imposes an upper limit of the number of processes that can run on the node. This parameter corresponds closely to the **FileNumberHeadroom** parameter in the [PersistentDb] section.

The highest database file number cannot be greater than **65,483**.

Parameter is read when …

The node is initialized.

Hint

If you anticipate that you will execute more than 2,047 processes on a node, increase this value so that you do not have to restart the node. (Exception 3003 is raised when the number of active processes exceeds this value.)

External Database Section [ExternalDb]

The [ExternalDb] section of the JADE initialization file contains parameters that affect the operation of your external databases. The

Note All parameters in this section are read when the external database is next opened.

The [ExternalDb] section can contain the following parameters.

AllowProcedureCalls

The **AllowProcedureCalls** parameter is reserved for future use.

ConnectionTimeout

Value Type Integer *(seconds)*

Default 15

Purpose

The **ConnectionTimeout** parameter specifies the maximum number of seconds to wait for the response to an external database connection.

An exception is raised if a response is not returned within the specified timeout period.

Specify a value of zero (**0**) to indicate that there is no connection timeout.

DisableCursorScrolling

Value Type Boolean

Default false

Purpose

The **DisableCursorScrolling** parameter specifies whether the JADE external database query engine avoids using the scrolling features of the **SQLExtendedFetch** API. Set this parameter to **true** if you want to work in a degraded fashion with an ODBC driver that lacks certain required functionality (for example, versions of the EasySoft ODBC driver for C-ISAM). When scrolling is disabled, there is some necessary loss of functionality. In particular, the following operations are not supported.

- **ExternalIterator**::**back** method
- **ExternalCollection**::**last** method
- Reverse iteration using the **foreach** instruction **reversed** modifier

Other scrolling functionality is emulated by the query engine.

Hint

You should normally leave this parameter set to the default value (**false**). However, you may need to explicitly disable scrolling by setting the value to **true** for ODBC drivers that have the following characteristics.

- Supports the **SQLExtendedFetch** API
- Claims to support dynamic cursors but does not support the required fetch orientation parameters to **SQLExtendedFetch**
- Conforms to ODBC level 2 (but not higher)

LoginTimeout

Value Type Integer *(seconds)*

Default 15

Purpose

The **LoginTimeout** parameter specifies the maximum number of seconds to wait for a log-in request to complete when opening a connection to an external database.

An ODBC exception is raised if the login times out.

Specify a value of zero (**0**) to indicate that there is no log-in timeout.

ODBCTraceOn

Value Type Boolean

Default false

Purpose

The **ODBCTraceOn** parameter specifies whether ODBC API calls are traced using ODBC Data Source Administration tracing.

When you specify a value of **true**, the ODBC calls are traced to the **jadesql.log** file located in your current working directory.

QueryTimeout

Value Type Integer *(seconds)*

Default 60

Purpose

The **QueryTimeout** parameter specifies the maximum number of seconds to wait for an SQL query against an external database to execute before returning to the application.

An exception is raised if query execution exceeds the timeout period.

Specify a value of zero (**0**) to indicate that there is no query timeout.

ReadOnlyMode

Value Type Boolean

Default false

Purpose

When the **ReadOnlyMode** parameter is set to **true**, the JADE SQL query engine sets the **SQL_ACCESS_MODE** connection option to **SQL_READ_ONLY**.

This has no effect on cursor concurrency, and some ODBC drivers may still permit update access. See also the **UseReadOnlyCursors** parameter, later in this section.

SQLConnectOption

Value Type String

Default Complete

Purpose

The **SQLConnectOption** parameter controls the **DriverCompletion** option sent to the ODBC **SQLDriverConnect** call. You can set this parameter to one of the values listed in the following table.

Value	DriverCompletion option is set to…
Complete	SQL_DRIVER_COMPLETE
Prompt	SQL_DRIVER_PROMPT
NoPrompt	SQL_DRIVER_NOPROMPT
Complete_Required	SQL_DRIVER_COMPLETE_REQUIRED

For more details about the **DriverCompletion** options for the ODBC **SQLDriverConnect** call, see the Microsoft ODBC documentation.

SuppressCatalogNameUsage

Value Type Boolean

Default false

Purpose

The **SuppressCatalogNameUsage** parameter, when set to **true**, overrides the usage of catalog name qualifiers in SQL queries otherwise supported by the ODBC driver and data source.

Hint

You may need to enable this parameter when using a single JADE external schema specification to connect the different versions of an external database that have different database names but the same logical schema definition.

UseCursorLibrary

Value Type Boolean

Default false

Purpose

The **UseCursorLibrary** parameter, when set to **true**, informs the ODBC driver manager to use the ODBC cursor library to support block scrollable cursors when ODBC drivers do not directly support this.

UseReadOnlyCursors

Value Type Boolean

Default false

Purpose

The **UseReadOnlyCursors** parameter, when set to **true**, provides improved concurrency for read-only access to external databases by causing the SQL query engine to open cursors in read-only mode.

As the JADE query engine opens ODBC cursors with the concurrency option set to **SQL_CONCUR_LOCK** by default, when you set the **UseReadOnlyCursors** parameter to true, cursors are opened in read-only mode, which improves concurrency for read-only applications.

Fault Handling Section [FaultHandling]

The [FaultHandling] section of the JADE initialization file enables you to specify the handling of application faults. A user *crash dump* is a snapshot of the contents of memory assigned to a user process.

If a fatal error results in the failure of a JADE node (for example, the **jade.exe**, **jadapp**, or **jadrap** program), a user dump of the failing process greatly enhances the JADE Support ability to diagnose the problem.

Note Parameter values in this section may apply only to specific operating systems.

The [FaultHandling] section can contain the following parameters.

CustomExitValue<*JADE-error-message-number*>

Value Type Integer

Default Not specified

Purpose

The **CustomExitValue<*JADE-error-message-number*>** parameter specifies whether JADE programs or utilities return a specific exit value rather than exit values that are JADE exit values (error numbers). The **CustomExitValue** parameter is valid only when the **StandardExitValues** parameter is set to **true**.

Set the **CustomExitValue** parameter to the required user-defined value only in the range 32 through 63 if you want to remap a standard exit value.

You can remap one or more JADE errors to one or more standard exit values or you can remap multiple JADE errors to a single standard exit value. For more details, see Appendix A, "Exit Values", in of the *JADE Installation and Configuration Guide*.

Parameter is read…

Just before the JADE program exits.

Hint

This parameter is useful if you want to ignore a number of JADE errors.

Disabled

Value Type Boolean

Default false

Purpose

The **Disabled** parameter specifies how a fault in executing code is handled. If the **Disabled** parameter is set to **true**, the fault is passed to the system default handler (for example, to Windows error reporting). If the **Disabled** parameter is set to **false**, the process is terminated after taking a process dump, where possible.

A JADE exception can be raised instead of terminating the process. This option is available for a function in a third-party DLL where any fault that occurred would never imperil the integrity of the executing process (for example, by corrupting memory or data).

To raise a JADE exception for a function instead of terminating the process, set the **Disabled** parameter to **false** and specify an *action* parameter in the following format.

```
on_library-name_function-name_Fault = action
```

The valid values for the ***action*** value are **JomExceptionWithDump** or **JomException**. For example, to raise a JADE exception when a fault occurs in the **encryptdata** function in the **cryptolib** library, specify the following.

```
on_cryptolib_encryptdata_Fault = JomException
```

Parameter is read when …

An error occurs.

EnableDebugButton

Value Type Boolean

Default true

Purpose

The **EnableDebugButton** parameter controls whether the **Debug** button is enabled on the Unhandled Exception dialog. When this parameter is set to **false**, the **Debug** button on the dialog is disabled.

When the application is run by a presentation client, the value of this parameter is obtained from the JADE initialization file on the application server.

Note The **Debug** button is disabled automatically for a 1201 exception *(Kernel stack overflow)*, because clicking the **Debug** button would generate another kernel stack overflow and for a 3123 exception (*Access to objects in this secondary database is currently disabled)*.

Parameter is read when …

An error occurs.

EnableSentinel

Value Type Boolean

Default true

Purpose

The **EnableSentinel** parameter specifies whether the **jadesentinel.exe** program is started for a JADE executable to monitor the executing parent and when required to do so, carry out a process dump. The **jadesentinel.exe** program runs as a debugger attached to its parent process, looks for specific exceptions, performs out-of-process process dumps, and terminates the program, as appropriate.

Note You should set the value of this parameter to **false** only when requested to do so by JADE Support.

Parameter is read when …

When the **jadesentinel.exe** executable starts up.

LogTcpNormalDisconnects

Value Type Integer

Default 1

Purpose

The **LogTcpNormalDisconnects** parameter specifies whether the logging of TCP/IP normal disconnections is suppressed.

Set the value to zero (**0**) to suppress logging of normal TCP/IP disconnections. The default value of one (**1**) specifies that normal TCP/IP disconnections are logged.

Parameter is read when …

The node is next initialized.

LogTcpNormalDisconnects

Value Type Integer

Default 1

Purpose

The **LogTcpNormalDisconnects** parameter specifies whether the logging of TCP/IP normal disconnections is suppressed.

Set the value to zero (**0**) to suppress logging of normal TCP/IP disconnections. The default value of one (**1**) specifies that normal TCP/IP disconnections are logged.

Parameter is read when …

The node is next initialized.

ProcessDumpDirectory

Value Type String *(disk-path)*

Default ProcessDumps

Purpose

The **ProcessDumpDirectory** parameter specifies the location of the directory in which process dumps are created.

Relative directories are assumed to be relative to the JADE directory. For example, if your installation directory is **Jade\bin** (that is, your JADE HOME directory is **Jade**) and you specify **ProcessDumpDirectory=ProdProcessDumps**, the full directory path is **Jade\ProdProcessDumps**.

Parameter is read when …

The JADE system is started.

StandardExitValues

Value Type Boolean

Default false

Purpose

The **StandardExitValues** parameter specifies whether JADE programs or utilities return generic exit values or exit values that are JADE error numbers.

Set this parameter to **true** if you want JADE programs or utilities to return generic exit values.

For more details, see Appendix A, "Exit Values", in of the *JADE Installation and Configuration Guide*.

Parameter is read when …

Just before the JADE program exits.

SuppressFaultLog

Value Type Boolean

Default false

Purpose

The **SuppressFaultLog** parameter specifies whether the software exception log is created.

Parameter is read when …

The fault occurs.

Caution Under normal circumstances, you should not enable this suppression. Enable it only when there is a problem with fault log creation.

SuppressProcessDump

Value Type Boolean

Default false

Purpose

The **SuppressProcessDump** parameter specifies whether a process dump is automatically created when a JADE node encounters a fatal software exception.

Parameter is read when …

The fault occurs.

JADE Ad Hoc Index Section [JadeAdHocIndex]

The [JadeAdHocIndex] section of the JADE initialization file enables you to specify options for the worker applications that build, drop, and delete an ad hoc index, and for the controller application that starts worker applications when there is an ad hoc index maintenance operation to be performed.

The [JadeAdHocIndex] section can contain the following parameters.

BuildCommitPeriod

Value Type Integer

Default 5

Purpose

The **BuildCommitPeriod** parameter specifies the number of seconds that elapse before a database commit is performed during the building of an ad hoc index. The default value is five (**5**) seconds, the minimum value is one (**1**) second, and the maximum value is **120** seconds.

Parameter is read when …

The ad hoc index worker application is started.

MaxBuildWorkers

Value Type Integer

Default 2

Purpose

The **MaxBuildWorkers** parameter specifies the maximum number of ad hoc index build worker applications that can be active simultaneously.

Parameter is read when …

The ad hoc index controller application is started.

JADE Command Line Section [JadeCommandLine]

The [JadeCommandLine] section of the JADE initialization file enables you to enter command line parameters in the JADE initialization file, with the exception of the **name** command line argument (which although you *can* specify it, is ignored).

For details about command line arguments that you can specify in this section, see "Specifying Arguments in the JADE Command Line", in Chapter 2 of the *JADE Installation and Configuration Guide*. Alternatively, see the appropriate JADE utility user's guide for details about running other JADE applications. For example, see "Running the Application Server and Presentation Clients", in Chapter 2 of the *JADE Thin Client Guide* for details about running an application server or a presentation client.

For details about multiple programs (with the exception of **jdbutilb**) on the same host sharing a JADE initialization file, see "Sharing JADE Initialization Files", earlier in this chapter.

Some command line arguments (for example, **path**, **schema**, and **appServer**) are required for some JADE programs but for other JADE programs they are optional.

Values specified in JADE initialization file parameters can be overridden by command line argument values.

Note Unless your JADE initialization file is located in the **bin** directory, you should specify the **ini** argument 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.

ThinClient

Value Type Boolean

Default Not specified

Purpose

The **ThinClient** parameter specifies whether the **jade.exe** program checks only arguments that are required to run in JADE thin client mode.

If the [JadeCommandLine] section or the **jade.exe** command line has the respective **ThinClient** parameter or **thinClient** argument set to **true**, **jade.exe** checks that the arguments required to run in thin client mode are present, ignoring any that are relevant only to the standard (fat) client mode of operation.

Conversely, if the [JadeCommandLine] section or the **jade.exe** command line has the **thinClient** argument set to **false**, **jade.exe** checks that the arguments required to run in standard client mode are present, ignoring any that are relevant only to the JADE thin client mode of operation.

Parameter is read when …

The **jade.exe** program next starts up.

Hint

Placing command line arguments in the [JadeCommandLine] section may cause an issue in determining the mode in which the **jade.exe** program should run; that is, in standard (fat) client mode or in JADE thin client mode.

JADE Compiler Section [JadeCompiler]

The [JadeCompiler] section of the JADE initialization file enables you to control the destination of compiler warning messages resulting from compiling schema files and the time after which the compiler lock times out in a multiuser JADE development environment.

The [JadeCompiler] section can contain the following parameters.

CompilerLockTimeout

Value Type Integer *(milliseconds)*

Default 10000

Purpose

The **CompilerLockTimeout** parameter specifies the time after which the compiler lock times out in the JADE development environment, the JADE Debugger, and the JADE Painter.

The default value of **10,000** milliseconds specifies that the compiler lock times out after 10 seconds. The minimum value is **200** milliseconds and the maximum value is **3,600,000** milliseconds (that is, one hour).

When the lock timeout occurs, a message box is displayed, with the text *Compiler Lock Timeout* and the user name of the process.

Note This parameter applies only to the JADE development environment, the JADE Debugger, and the JADE Painter and it has no impact in a single user JADE development environment.

Parameter is read when …

The node in which the method is being compiled is next initialized.

NonIDECompilerLockTimeout

Value Type Integer *(milliseconds)*

Default 0

The **NonIDECompilerLockTimeout** parameter specifies the time after which the compiler lock of all JADE processes other than the JADE development environment, the JADE Debugger, and the JADE Painter time out.

The default value of zero (**0**) milliseconds specifies that processes other than the JADE development environment, the JADE Debugger, and the JADE Painter wait forever for the compiler lock.

When you set this parameter to a value greater than zero and lock timeout occurs, a lock exception is raised.

Note To avoid unexpected lock timeout exceptions occurring when multiple non-GUI processes in one node simultaneously execute the **Process** class **createTransientMethod** method, do not set this parameter too low (for example, the **CompilerLockTimeout** parameter default value of **10** seconds is known to be too low when multiple non-GUI processes in one node simultaneously create transient methods).

Parameter is read when …

The node in which the method is being compiled is next initialized.

SuppressWarningDialogs

Value Type Boolean

Default false

Purpose

The **SuppressWarningDialogs** parameter specifies whether all warning message dialogs resulting from compiling schema files (for example, class number conflict messages) are suppressed and warning messages are output only to the message log.

This parameter applies only when loading schemas using the JADE development environment (that is, by using the Load Options dialog).

Set this parameter to **true** if you do not want compiler warning messages displayed on your desktop when loading schemas.

Parameter is read when …

The schema file load is initiated.

JADE Environment Section [JadeEnvironment]

The [JadeEnvironment] section of the JADE initialization file enables you to specify parameters relating to the environment in which a JADE system is installed. For example, you can control the location of directories so that files are created in appropriate locations to meet requirements for sharing, security, and the operating system.

The [JadeEnvironment] section can contain the following parameters.

ACLTemplateDirectory

Value Type String *(disk-path)*

Default JADE HOME directory

Purpose

The **ACLTemplateDirectory** parameter specifies the directory path of the JADE Database Administration utility template file, which has the name **jdbadmin.template**. If a relative path is specified for the **ACLTemplateDirectory** parameter, it is relative to the JADE Home directory.

The **jdbadmin.template** file is created with Access Control Lists (ACLs). A client running the JADE Database Administration utility is authenticated by session key files, which are generated in the database directory using the **jdbadmin.template** file.

Parameter is read when …

The JADE Database Administration utility connects to the database server.

EnhancedLocaleSupport

Value Type Boolean

Default false

Purpose

The **EnhancedLocaleSupport** parameter enables handling of standard client and presentation client locale-sensitive formatting of date, time, and numeric values (that is, numbers and currency) to be standardized and the regional overrides set on the presentation client to be forwarded to the application server so that both nodes use a consistent set of locale settings for the application when running in JADE thin client mode.

Notes Enhanced locale support is enabled (or disabled) throughout a complete JADE environment, with the single setting in the JADE initialization file on the database node.

All client nodes and presentation clients share the same single setting.

The required languages must be installed on both the application server and the presentation client to ensure consistent behavior (for example, sorting and comparison) between the presentation client and the application server.

When this parameter is not defined or it is set to the default value of **false**:

- Calling some methods can cause inconsistent results to be returned to the application server when running in JADE thin client mode and there are regional overrides, as all overrides on the application server are suppressed.
- The **Application** class **setJadeLocale** method does not change the current locale used for date, time,

numeric, and currency parsing and formatting.

- The two-digit edit mask year of **'yy'** is calculated using the current century.

When the **EnhancedLocaleSupport** parameter is set to **true**:

- The required language or languages must be installed on both the application server *and* the presentation client, to ensure consistent behavior (for example, sorting and comparison) between the presentation client and the application server.
- Presentation client regional overrides are applied.
- Locale-sensitive values (for example, the **Date** primitive type **monthName** method) return the values for the current thread locale, which is changed when the **Application** class **setJadeLocale** method is called.
- The Windows Control Panel setting is used to convert a two-digit year into a four-digit year for a two-digit edit mask year of **'yy'**. By default, years:
  - 00 through 29 become 2000 through 2029
  - 30 through 99 become 1930 through 1999
- The values returned from the **Application** class **currentLocaleInfo** property reflect the regional overrides that have been applied to the session locale, if currently in use.
- The **Schema** class **getLocaleFullInfo** method and its associated methods recognize the **LCID_SessionWithOverrides** global constant value, which enables you to retrieve information from the session locale without having to save the initial locale for the call.
- Additional conversion routines have been added to fully support (binary-to-string-to-binary) value round trips; for example, the **Decimal** primitive type **parseCurrencyWithFmtAndLcid** and **userCurrencyFormatAndLcid** methods, and the **Time** primitive type **parseWithFmtAndLcid** and **userFormatAndLcid** methods.
- **TextBox** class **dataType** property values for currency, short date, long date, and time ensure that entered values are limited to match the current locale formatting for each type.

Parameter is read when …

The database node initializes.

EnableWinHTTP

Value Type Boolean

Default true

Purpose

The **EnableWinHTTP** parameter enables the use of the Microsoft Windows HTTP Services (WinHTTP) library to perform low-level network communications for server-type environments.

Note If WinHTTP is not installed, WinINET is used if it is enabled (that is, it is set to **true**).

If the **EnableWinHTTP** parameter and the **EnableWinINET** parameter are both set to **true**, WinHTTP is used in preference to the Microsoft Windows Internet application programming interface (WinINET). If both parameters are set to **false**, Windows HTTP Services and the Windows Internet application programming interface are disabled and a Web consumer can use only the JADE Direct scheme (that is, **jadehttp.tcp**).

WinHTTP proxy settings are configured using the **netsh winhttp** tool. For details about configuring Internet proxy settings, see the relevant Microsoft documentation.

Parameter is read when …

The node is started.

EnableWinINET

Value Type Boolean

Default false

Purpose

The **EnableWinINET** parameter enables the use of the Microsoft Windows Internet application programming interface (WinINET) library to perform low-level client network communications.

If the **EnableWinHTTP** and the **EnableWinINET** parameters are both set to **true**, Microsoft Windows HTTP Services (WinHTTP) is used in preference to WinINET. If both parameters are set to **false**, Windows HTTP Services and the Windows Internet application programming interface are disabled and a Web consumer can use only the JADE Direct scheme (that is, **jadehttp.tcp**).

WinINET proxy settings are configured in the same way as those for Internet Explorer; that is, by using the Internet Options dialog accessed from the Control Panel. For details about configuring Internet proxy settings, see the relevant Microsoft documentation.

Parameter is read when …

The node is started.

ExtendedLengthPaths

Value Type Boolean

Default false

Purpose

JADE provides an extended length path implementation that supports path names up to twice the Windows maximum path length (that is, 520 bytes).

Set the **ExtendedLengthPaths** parameter to **true** to enable the use of the JADE extended length path implementation.

Note When the **ExtendedLengthPaths** parameter is set to the default value of **false** and an operation using a path name with the length greater than the Windows maximum path length of 260 bytes is attempted, the error 5029 (*File name not valid for file system*) is raised.

Parameter is read when …

An operation with a path name length greater than the Windows maximum path length of 260 bytes is attempted.

JadeWorkDirectory

Value Type String *(disk-path)*

Default *JADE-HOME*\temp

Purpose

The **JadeWorkDirectory** parameter specifies the path of the directory where work files are created by JADE.

By default, this directory is created at the same level as the JADE installation directory (that is, the directory in which the **jade.exe** executable program is located) and is named **temp**. For example, if the JADE installation directory is **\jade\bin**, the working directory would be **\jade\temp**. The **JadeWorkDirectory** parameter can specify an absolute path or relative path (relative to the JADE HOME directory, which is **\jade** in the example).

The **JadeWorkDirectory** parameter specifies the location of the cache file for a thin client (which contains all forms and pictures sent by logic from the application server), unless another location is specified by the **FormCacheFile** parameter in the [JadeThinClient] section.

The **JadeWorkDirectory** parameter also specifies where the thin client automatic download interlock file (**thinlock.fil**) is created.

Parameter is read when …

A file in the work directory is created or used or the **getJadeWorkDirectory** method is called.

jomsrvr2

Value Type String

Default jomsrvr2

Purpose

The **jomsrvr2** parameter specifies the dynamic link library that contains authentication and RPC encryption hook routines used when connecting to the database server. It is used if the:

- **AuthenticationHookDLL** parameter in the [JadeSecurity] section is set to **Internal**
- **RPCEncryptionHookDLL** parameter in the [JadeSecurity] section is set to **Internal**

Parameter is read when …

Connecting to the database server.

ProgramDataDirectory

Value Type String *(disk-path)*

Default JADE HOME directory

The **ProgramDataDirectory** parameter specifies the path of the program data directory. The program data directory is used to share files among multiple users of the binaries; for example, the **jommsg.log** file or shared dictionary spelling files that are updated.

The **ProgramDataDirectory** parameter is ignored if JADE is not installed under the **\Program Files** directory, in which case the program data directory is always the JADE HOME directory.

The **ProgramDataDirectory** parameter can have one the following values.

- **<default>**, which is the default value.

  It is from the JADE HOME directory by replacing the **\Program Files** portion with the programmatically obtained location for the common application data directory. For example, a presentation client installed into **\Program Files\Jade Software Corporation\parsys** would have a program data directory of **\ProgramData\Jade Software Corporation\parsys**.

- **<homedir>**, which is the JADE HOME directory.
- **<programdata>**, which is the same as **default**.
- A user-specified directory name.

Parameter is read when …

The **Node** class **getProgramDataDirectory** method is called.

UserDataDirectory

Value Type String *(disk-path)*

Default JADE HOME directory

Purpose

The **UserDataDirectory** parameter specifies the location of the user data directory. The user data directory is used for files that are specific to each user of the JADE executables; for example, if a presentation client installation occurs on a 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, separate directories for each user).

The **UserDataDirectory** parameter is ignored if JADE is not installed under the **\Program Files** directory, in which case the user data directory is always the JADE HOME directory. The **UserDataDirectory** parameter can have one the following values.

- **<default>**, which is the default value.

  It is obtained from the JADE HOME directory by replacing the **\Program Files** portion of the HOME directory with the programmatically obtained 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** would have a user data directory of **\Users\wilbur\AppData\Local\Jade Software Corporation\parsys**.

- **<homedir>**, which is the JADE HOME directory.
- **<userdata>**, which is the same as **default**.
- A user-specified directory name.

Parameter is read when …

The **Node** class **getUserDataDirectory** method is called.

JADE Execute Flags [JadeExecuteFlags]

The [JadeExecuteFlags] section of the JADE initialization file contains a parameter that enables you to specify whether instructions within an **executeWhen** instruction are to be loaded and executed.

FlagName

Value Type Boolean

Default false

Purpose

The ***FlagName*** parameter specifies the name of a boolean flag that determines whether the node loads and executes instructions within an **executeWhen** instruction.

The ***FlagName*** value is a user-defined **Boolean** global constant (for example, **DebugTestFlag=true**). You can set the ***FlagName*** value for the node in the JADE initialization file or it can be changed dynamically in your code at run time, by calling the **Node** class **setExecuteFlagValue** method.

If the value of the parameter is **true**, instructions within an **executeWhen** *FlagName*; instruction are loaded and executed; otherwise, the instructions are not loaded.

Parameter is read when …

The node is initialized.

JADE Extract Sort Section [JadeExtractSort]

The [JadeExtractSort] section of the JADE initialization file contains a parameter that enables you to specify the location of temporary sort files (for example, when using the **File** class **extractSort** method).

The [JadeExtractSort] section can contain the following parameter.

SortDirectory

Value Type String *(disk-path)*

Default Operating system temporary file location

Purpose

The **SortDirectory** parameter specifies the location of temporary sort files, which enables you to specify the location for each node. For example, you can move the files used in a sort process from the default temporary directory if the print output of a report would consume all of the local disk resulting in subsequent reports that require sorting to fail because there is insufficient disk space.

If the value is **<default>** or the **SortDirectory** parameter does not exist, the sort process uses the operating system temporary file location.

If the **SortDirectory** parameter exists but specifies an invalid directory, an exception is raised.

Parameter is read when …

Each sort file is extracted; that is, when the next sort file is opened.

Hint

If you specify a relative directory name, the directory is assumed to be relative to the JADE HOME directory. For example, if your installation directory is **Jade\bin** (that is, your JADE HOME directory is **Jade**) and you specify **SortDirectory=SortLogs**, the full directory path is **Jade\SortLogs**.

JADE Font Substitutions Section [JadeFontSubstitutions]

The [JadeFontSubstitutions] section of the JADE initialization file enables you to substitute your own preferred fonts for those defined in an application.

JADE applications written in a Windows environment rely on the Windows environment to provide specific common fonts.

Application users can substitute their own preferred fonts for those defined in an application, simply by adding the appropriate font entries to the [JadeFontSubstitutions] section of the JADE initialization files on their client nodes.

Note All parameters in this section are read when a standard (fat) client node or presentation client is next initialized. Changes to parameter values in this section are not detected until the client program is restarted.

You can substitute a font so that text is drawn in a different font, altering the name, size, weight (bold), and slant (italic). Print output uses the specified font substitution.

Font substitution does *not* occur in:

- The **JadeRichText** control
- Operating system-provided common dialogs (for example, the File Open or File Save dialog)
- The message box

All applications run from a specific presentation client or standard fat client share the same font substitutions. Each presentation client reads its own set of font substitutions from its own JADE initialization file, allowing presentation clients that connect to the same application server to use different fonts.

The font properties seen by user logic are always the original specified values; that is, the match specification. The substituted values are not visible to user logic.

The [JadeFontSubstitutions] section can contain the parameters described in the following subsections.

LogAvailableFonts

Value Type Boolean

Default Not specified

Purpose

Set the **LogAvailableFonts** parameter to **true** if you want all fonts installed on the client node logged in the **jommsg*n*.log** file; for example:

```
JomLog: Available-Font: Microsoft Sans Serif; TrueType; charset=Western
JomLog: Available-Font: Arial; TrueType; charset=Western
JomLog: Available-Font: Arial Unicode MS; TrueType; charset=Western
JomLog: Available-Font: Times New Roman; TrueType; charset=Western
```

LogFontCreations

Value Type Boolean

Default Not specified

Purpose

Set the **LogFontCreations** parameter to **true** if you want all font creations on the client node logged in the **jommsg*n*.log** file to enable you to determine exactly which fonts are used by the application.

The first font specification is the original value. The second specification that follows the equals symbol is the actual font used (that is, the substituted font). If no substitution match is found, **unmatched** is displayed. In all other cases, the sequentially numbered substituted font specification that is used is displayed at the right of the original font definition used, as shown in the following example.

```
JomLog: Create-Font: MS Sans Serif:9.75:regular:regular = unmatched
JomLog: Create-Font: Arial:12:regular:regular = unmatched
JomLog: Create-Font: MS Sans Serif:8.25:bold:regular = unmatched
JomLog: Create-Font: Arial:8.25:bold:regular = unmatched
JomLog: Create-Font: Courier New:9:regular:regular = Courier New:11:regular:regular
[#4]
```

<match-font-specification>

Value Type String

Default Not specified

Purpose

Use the ***<match-font-specification>*** parameter to specify a font substitution definition consisting of two parts separated by an equals (**=**) character. The left side specifies the original font and the right side specifies the replacement font. For details about defining substitute fonts, see "Font Substitute Definitions", in the following subsection.

Tip

As there is no implied font match precedence and definitions are searched in the order specified in the initialization file, you should position the more-specific matches first.

Font Substitution Definitions

A match font substitution definition consists to two parts, which are separated by an equals (**=**) character. The left side is the original font specification and the right side is the replacement font specification.

A font specification consists of the following four elements, each separated by a colon character (**:**).

1. Font name
2. Size (in points)
3. Weight (bold or regular)
4. Slant (italic or regular; that is, upright)

Tip Set the **LogAvailableFonts** parameter to **true** if you want to output all fonts installed on the client node (that is, match font specifications) to the **jommsg*n*.log** file.

Each font substitution definition in the JADE initialization file is written to the **jommsg*n*.log** file on the client node. An invalid font substitution definition causes a message describing the error to be written to the **jommsg*n*.log** file and that font definition is ignored.

For details about each element of a font specification, see the following subsections.

Font Name Definition

When you specify original and replacement font names, embedded spaces are significant and the name cannot include colon characters (**:**), which are used only to separate font substitution elements.

- Original specification font, which is the value taken from the **Control** class **fontName** property.

  The last (or only) character of the original font name can be an asterisk if you want to specify that the name is a prefix mask that matches any name with the same leading characters; for example, **Arial*** matches **Arial**, **Arial Narrow**, **Arial Black**, or **Arial Unicode MS**. A single asterisk matches any font name.

- Replacement specification font name can be a single asterisk (*****) if you want the same substitute font name as the original font name. A trailing asterisk is not allowed.

Size Definition

Specify a **Real** point size value that is a floating-point value measured in points taken from the **Control** class **fontSize** property.

- Original specification font

  The original font size can be a single asterisk that matches any font size.

- Replacement specification font

  The replacement font size can be a single asterisk that indicates that the font size is the same as the original font size. The replacement font size can include a leading sign character (that is, **+** or **-**), indicating that the value is relative to the original font size; for example, if the original font size is **9** and the replacement font size is **+2**, the substitute font size is **11**.

Weight Definition

The font weight (that is, the bold attribute value) of the original font specification is taken from the **Control** class **fontBold** property.

You can define the following weight values for the original and replacement fonts.

- ***** (a single asterisk)

  For the original font specification, an asterisk indicates that the weight is not a consideration. For the replacement font specification, an asterisk indicates that the weight is the same as the bold attribute value of the original font specification.

- **R**, **r**, or **regular** (that is, the value of the **fontBold** property is **false**)

- **B**, **b**, or **bold** (that is, the value of the **fontBold** property is **true**)

Slant Definition

The font slant (that is, the italic attribute value) of the original font specification is taken from the **Control** class **fontItalic** property.

You can define the following weight values for the original and replacement fonts.

- ***** (a single asterisk)

  For the original font specification, an asterisk indicates that the slant is not a consideration. For the replacement font specification, an asterisk indicates that the slant is the same as the italic attribute value of the original font specification.

- **R**, **r**, or **regular** (that is, the value of the **fontItalic** property is **false**)

- **I**, **i**, or **italic** (that is, the value of the **fontItalic** property is **true**)

Font Substitution Examples

The following example changes the Arial bold font to the Sans Serif font, changing only the name.

```
Arial:*:bold:*=Sans Serif:*:*:*
```

The following example changes the 8 point Courier New font to 9 point Sans Serif font with the bold attribute, changing the font name, size, and weight.

```
Courier New:8:*:*=Sans Serif:9:bold:*
```

The following example changes the Courier New font to the Sans Serif font that is one point larger, changing the name and relative size.

```
Courier New:*:*:*=Sans Serif:+1:*:*
```

The following example increases all font sizes by 1 point.

```
*:*:*:*=*:+1:*:*
```

JADE Generic Messaging Section [JadeGenericMessaging]

The [JadeGenericMessaging] section of the JADE initialization file enables you to specify settings for the JADE Generic Messaging module.

Note All parameters in the [JadeGenericMessaging] section are read when a JADE Generic Messaging feature is used for the first time; for example, creating an instance of the **JadeMessagingFactory** class.

The [JadeGenericMessaging] section can contain the parameters described in the following subsections.

RegisterIBMWMQC

Value Type Boolean

Default false

The **RegisterIBMWMQC** parameter specifies whether the IBM WebSphere MQ message queuing transport is available.

Note To use this transport, a copy of the WebSphere MQ client support software must be obtained and licensed from IBM or an IBM reseller.

RegisterJadeMQ

Value Type Boolean

Default true

The **RegisterJadeMQ** parameter specifies whether the JadeMQ message queuing transport is available.

Caution A number of features in JADE use the JADE message queuing transport, so you are advised not to set this parameter to **false**.

JADE Help Section [JadeHelp]

The [JadeHelp] section of the JADE initialization file contains the parameters described in the following subsections, which enable you to specify values that provide flexibility when using help files, especially Adobe Portable Document Format (PDF) files, in a JADE environment.

Note All parameters in this section are read when the online help is first used in the current work session.

The [JadeHelp] section can contain the following parameters.

ConnectSleepRetries

Value Type Integer, Integer *(sleep, retries)*

Default 5,1

Purpose

The **ConnectSleepRetries** parameter specifies the number of seconds between the specified number of times JADE attempts to retry launching the Adobe Reader and opening a PDF file at the appropriate topic when you press F1 to access context-sensitive help from the JADE development environment.

The first value (*sleep*) is the number of seconds to sleep between connection attempts and the second value (*retries*) is the number of attempts to connect. If the attempt to connect to a PDF file in the launched Acrobat Reader fails, JADE retries the specified number of times with the specified number of seconds between each attempted retry.

You can specify a sleep value of **1** through **60** inclusive and a retries value of **1** through (**120**/**sleep**), inclusive. If JADE cannot attach to the hyperlink indicated by the F1 action in that time, control returns to JADE.

Hint

If the **UseJadeWebHelp** parameter in the [JadeHelp] section of the JADE initialization file is set to **false** and you press F1 from the JADE development environment to launch the Adobe Reader and access the appropriate topic in a JADE PDF file using Adobe Reader 6.0 or higher from a slower machine (for example, from an older laptop), JADE can time-out before it has time to load the appropriate document at the required topic after it has launched the Adobe Reader application. Although the Adobe Reader may be launched, you have to repeat the F1 action to enable JADE to open the PDF document itself at the specified hyperlink topic.

Although you can use the **ConnectSleepRetries** parameter sleep and retries values to allow JADE the maximum of two minutes to launch the Adobe Reader and open the appropriate file at the requested topic if you are running JADE on a slower machine, it is better to specify a smaller number of seconds to sleep (for example, **2**) and a larger number of retries (for example, **10**), to reduce the impact of the connection process.

HelpDirectory

Value Type String *(disk-path)*

Default Not specified

Purpose

The **HelpDirectory** parameter specifies the path in which the help files (which can be .**pdf**, .**hlp**, or .**chm** files) are located.

The path can be the absolute path or it can be relative to the JADE HOME directory. (If your installation directory is **Jade\bin**, your JADE HOME directory is **Jade**.)

Note As some Adobe Reader versions are sensitive to forward slash (/) characters in the path to the PDF file, use this parameter to set the path name (using backslash characters) rather than letting JADE detect the install location.

When a JADE help file is requested (for example, when F1 is pressed for context-sensitive online help), JADE looks for help files in the following search order.

1. If the help file specified in the **Application** class **helpFile** property contains an absolute path, this file and location are used.
2. If the **HelpDirectory** parameter is defined in the [JadeHelp] section of the JADE initialization file, use that value.
3. Checks the **documentation** subdirectory of the JADE HOME directory; for example:

```
JADE docs\documentation
```

4. Checks the JADE **install** directory (for example, **\bin**).
5. Lets the system try to locate the file.

<help-file-name>

Value Type String *(unique-internal-help-file-name)*

Default Not specified

Purpose

The ***<help-file-name>*** parameter remaps a JADE-internal help file name to an external help file name; for example, when:

- Handling case-sensitivity

  As some operating systems are case-sensitive, you can use this parameter to remap a JADE-internal help file name to a help file name that has different case-sensitivity.
- Locating a help file or help files in a different location to that specified in the **HelpDirectory** parameter
- Mapping two or more help files to one help file
- Renaming help files in deployed systems

JADE checks the name and case-sensitivity of each help file in the JADE development environment or JADE user application to ensure that it can locate the exact file name and path.

If a file look-up fails, the file on which the look-up failed is logged to the **jommsg.log** file, to enable you to use this parameter to specify a relative name of a JADE-internal help file or the absolute path of a JADE-internal help file and the associated external help file name or absolute path, as shown in the following example.

```
JadeMsgs.pdf = JADEMsg.pdf
ReportWriter.pdf = \Jade\Reports\JRWriter.pdf
```

If you do not specify an absolute path, JADE uses the value of the **HelpDirectory** parameter.

Notes

The value of this parameter is case-sensitive.

You can specify more than one unique help file name; for example, if you want to remap two or more internal JADE help files to other names or paths.

This file type can be an Adobe Portable Document Format (.**pdf**), Windows help (.**hlp**), or compiled help (.**chm**) format. (Note that it is your responsibility to provide the supporting software that enables the display of the appropriate format if you supply online help for applications that you develop.)

HelpSchemes

Value Type String

Default http://,https://

Purpose

The **HelpSchemes** parameter specifies the URL schemes that can be used to access an HTML-based online help system. The schemes are specified in a comma-separated list.

In the following example, the loading of HTML help pages from your local disk is enabled by adding **file://**.

```
HelpSchemes = http://,https://,file://
```

HtmlHelpContentsUrl

Value Type String *(disk-path)*

Default Not specified

Purpose

The **HtmlHelpContentsUrl** parameter specifies the URL that will be used to access an HTML-based online help system when help is invoked from the **Window** class **showHelp** method or by the user pressing the help key (F1) and the **helpKeyword** property value has not been specified.

If you specify a value for this parameter, it must be a complete URL; for example:

```
HtmlHelpContentsUrl = http://www.example.com/prodhelp/contents.htm
```

HtmlHelpIndexUrl

Value Type String *(disk-path)*

Default Not specified

Purpose

The **HtmlHelpIndexUrl** parameter specifies the URL that will be used to access an HTML-based online help system when a help request is made for the help index, help context, or help finder.

If you specify a value for this parameter, it must be a complete URL; for example:

```
HtmlHelpIndexUrl = http://www.example.com/prodhelp/index.htm
```

JadeHelpBaseUrl

Value Type String *(disk-path)*

Default Not specified

Purpose

The **JadeHelpBaseUrl** parameter specifies the URL that will be used to access topics in the HTML5 Web format product information from context-sensitive help functionality in the JADE development environment when the value of the **UseJadeWebHelp** parameter is set to **true** (the default).

When the **UseJadeWebHelp** parameter is set to **true** and a JADE help file is requested (for example, when F1 is pressed for context-sensitive online help from the JADE development environment), JADE reads the **JadeHelpBaseUrl** parameter. If a value is specified for the **JadeHelpBaseUrl** parameter, it uses that URL. If the value is **<default>** or it is empty, the URL is determined by the internal hard-coded URL for the current release.

If you specify a value for this parameter, it must be a complete URL. For example, the [JadeHelp] section of the JADE initialization file could contain the following parameter values.

```
[JadeHelp]
UseJadeWebHelp  = true
JadeHelpBaseUrl = https://www.jadeworld.com/docs/jade-2018/Default.htm
```

Note Where the .**htm** extension is used, .**html** is also valid.

PdfCloseReader

Value Type Boolean

Default Not specified

Purpose

If you have an open PDF file running under Adobe Reader and you do not want JADE to shut it down (for example, when you select the **Run Server as Service** command from the Service Configuration dialog and then click the **OK** button), set the value of the **PdfCloseReader** parameter to **false**.

By default, the Adobe Reader is closed when you run JADE as a service (that is, the value of the **PdfCloseReader** parameter is **true**).

PdfHelpIdPrefix

Value Type String *(file-prefix)*[, Integer *(expanded-identifier-size)*]

Default Not specified

Purpose

As PDF files require string values as help destinations, the **PdfHelpIdPrefix** parameter enables you to format a **helpContextId** property into a string value.

The *file-prefix* value is the initial string (which has a maximum length of 100 characters) and the optional *expanded-identifier-size* parameter is a positive integer value that defaults to **1**.

The size indicates how many digits to expand the **helpContextId** property with leading zeroes. For example, a **helpContextId** property with a value of **527** and a **PdfHelpIdPrefix** parameter value of **myhelp,6** produces **"myhelp000527"**. Conversely, a **helpContextId** property value of **527** and a **PdfHelpIdPrefix** parameter value of **myhelp,2** produces **"myhelp527"**.

No truncation of the **helpContextId** property value occurs.

Hint

The .**hlp** and .**chm** help files support both strings and numbers as help targets.

UseJadeWebHelp

Value Type Boolean

Default true

Purpose

The **UseJadeWebHelp** parameter specifies whether context-sensitive help from the JADE development environment accesses .**htm** topics in the HTML5 Web format of the JADE product information. When this parameter is set to **true** (the default value), JADE reads the **JadeHelpBaseUrl** parameter to determine the help base URL.

Set this parameter to **false** if you want to context-sensitive help to target specific sections in the appropriate PDF files.

WindowPos

Value Type Integer, Integer, Integer, Integer

Default Not specified

Purpose

The **WindowPos** parameter specifies the position of the Adobe Reader window only when the **WindowShow** parameter is set to **position**.

If any or all of the values are absent, JADE default values are used.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical point coordinates of the window in pixels, respectively, relative to the top left corner of the window.

The third and fourth *integer* values (indicating the *x2* and *y2* points) indicate the width and height of the window in pixels, respectively.

Hint

If the window is positioned off the screen, an attempt is made to reposition it so that at least a quarter of the horizontal or vertical sides are visible.

WindowShow

Value Type String

Default Not specified

Purpose

The **WindowShow** parameter specifies the way in which the Adobe Reader window is displayed when a PDF file is invoked from within JADE.

You can specify one of the case-insensitive values listed in the following table.

Value	Description
restore	Allows the initial Adobe Reader window to be displayed depending on the requirements of the system and application (the default value, if the **WindowShow** parameter is not specified)
maximize	Requests that the initial Adobe Reader window is maximized to full-screen size
minimize	Requests that the initial Adobe Reader window is shrunk to an icon
position	Requests that the initial Adobe Reader window is positioned as specified by the value of the **WindowPos** parameter, if defined

JADE Inspector Section [JadeInspector]

The [JadeInspector] section of the JADE initialization file contains the parameters described in the following subsections, which enable you to specify values that customize the windows that are displayed when you inspect instances in a JADE database.

Note All parameters in the [JadeInspector] section are read when a JADE Inspector form is displayed.

The [JadeInspector] section can contain the following parameters.

Font

Value Type String *(font-name)*, Real *(font-size)*, Boolean *(font-bold)*

Default Tahoma, 8.25 points, regular (that is, **false**)

Purpose

The **Font** parameter specifies the attributes of the font used in the JADE Inspector forms (for example, Verdana, 9, false).

Set these values by using the **Font** command from the JADE Inspector Options menu, to ensure that the selections are valid.

If the information is missing or not in the expected format, the default font is used .

UseSameWindow

Value Type Boolean

Default false

Purpose

The **UseSameWindow** parameter is set to **true** or **false** when the **Use Same Window** command in the Options menu of the Schema Inspector form is selected; that is, toggled.

When this parameter is set to **true**, the initial Inspector form is shown modally and each double-click of an object in an Inspector form re-uses the same form to display the selected object, replacing the previously displayed object. In addition, a pane is displayed on the right of the form, containing a hierarchical list box displaying all of the objects that have been previously inspected. The hierarchy indicates the history of how the objects were inspected. The entries show the value of the **name** property if it exists in the object, followed by the class name and the JADE object identifier (oid) of the object. Clicking on an entry in the hierarchical history list at the right of the form re-displays the selected object.

The default **UseSameWindow** parameter value of **false** specifies that the each form is shown modally and only the top form can be accessed. That form has to be closed to enable access to the prior form displayed.

WindowPos

Value Type Integer, Integer, Integer, Integer

Default Not specified

Purpose

The **WindowPos** parameter specifies the position and size for any subsequent Inspector forms that are displayed in the form *left*, *top*, *width*, *height*. The values are in pixels, and the *left* and *top* coordinates are relative to the top left corner of the window.

If the information is missing or not in the expected format, the default position and size are used.

The values can be set by manually resizing a JADE Inspector form and then using the **Save Position/Size** command in the Options menu.

The **Clear Saved Position/Size** command in the Options menu replaces the value of the **WindowPos** parameter in the [JadeInspector] section with **<default>**. When the next Inspector form is opened, the default position and size will be used.

Note If more than one user uses the same JADE initialization file on the client, they share the same values for parameters in the [JadeInspector] section.

JADE Interpreter Sections

The [JadeInterpreter] and [JadeInterpreterOutputViewer] sections of the JADE initialization file contain parameters that:

- Enable the output of the **write** instruction or method tracing to the Jade Interpreter Output Viewer
- Control method and string caching
- Affect the operation of the Jade Interpreter Output Viewer

By default, output to the Jade Interpreter Output Viewer is enabled.

Interpreter Section [JadeInterpreter]

Note All parameters in the [JadeInterpreter] section are read when the node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

The [JadeInterpreter] section can contain the following parameters.

MethodCache

Value Type String (**Multiple** or **Single**)

Default Multiple

Purpose

The **MethodCache** parameter specifies whether each JADE process has a separate method cache (the default of **multiple**) or there is a single method cache for all processes within a node (that is, **single**).

Hints

The size of each method cache is specified by the **MethodCacheLimit** parameter. Multiple caches result in faster load and execution of methods, especially on Symmetric Multiprocessing (SMP) nodes. However, this improved performance is achieved at the expense of an increased usage of physical memory.

For details about obtaining method load time and cache overflow statistics that enable you to analyze the performance of your methods at run time, see "Reporting Profile Statistics", in Chapter 1 of the *JADE Runtime Application Guide*.

MethodCacheLifetime

Value Type Integer *(milliseconds)*

Default 0 for 32-bit nodes; 600000 (that is, 10 minutes) for 64-bit nodes

Purpose

The **MethodCacheLifetime** parameter specifies the number of milliseconds a method must be in cache before it is made available for removal.

When a method is executed, the JADE Interpreter must load the method code into the interpreter method cache for execution. If the method cache is full, the interpreter systematically remove the oldest unused methods. It keeps removing methods until it has enough space for the new method or it reaches a point where the oldest unused method is younger than the specified method lifetime.

The minimum value is zero (**0**).

Hint

Although specifying a method cache lifetime means that more method cache overruns may occur, methods should be reloaded less frequently, limiting excessive CPU consumption.

MethodCacheLimit

Value Type Integer *(prefix multiplier)*

Default 5M for 64-bit nodes; 512K for 32-bit nodes

Purpose

The **MethodCacheLimit** parameter specifies the maximum amount of memory that is allocated for an interpreter method cache.

The optimum value for this parameter varies, depending on the:

- Amount of physical memory available on your workstation.
- Average size of methods and the number that are executing at one time; that is, being called by another method.

The minimum value is **64**K bytes.

Note When a single cache for all processes within a node is used, this parameter takes effect when the node is initialized.

When multiple caches are used, the parameter takes effect when a process is initiated.

Hints

When a method is executed, the JADE interpreter must load the method code into the interpreter method cache for execution. The current copy is used if the method is not in use, it is already loaded in the cache, and it has not been modified. If a method is called frequently, tuning the cache size may result in substantial time-savings.

To determine method load counts and the amount by which the method cache limit is exceeded, refer to the method load times and cache overflow :

- Sections in your profile report. For details about reporting profile information and tracing methods in runtime applications, see "Reporting Profile Statistics" under "Profiling an Application", in Chapter 1 of the *JADE Runtime Application Guide*.
- Rows in the **Method Cache Statistics** table in the **Process Information** view of the JADE Monitor. For details, see "Displaying Process Method Cache Statistics" under "Performing Popup Menu Command Actions", in Chapter 2 of the *JADE Monitor User's Guide*.
- Values in the notification **userInfo** output when calling the **Process** class **sendMethodCacheStatistics** method.

The **MethodCacheLifetime** parameter, which defaults to zero (**0**) for 32-bit nodes and to **600000** milliseconds (that is, 10 minutes) for 64-bit nodes, specifies the number of milliseconds a method must be in cache before it is made available for removal.

If you specify a **MethodCache** parameter value of **Multiple** when you are running in JADE thin client mode, up to the maximum amount of memory that you allocate by using the **MethodCacheLimit** parameter may be allocated for each end user. When you set cache parameters and you have a lot of JADE nodes, you need to consider the resource that is available on the machine as a whole.

ReadEnabled

Value Type Boolean

Default true

Purpose

The **ReadEnabled** parameter specifies whether the User Input dialog is displayed when a method containing a **read** instruction is executed so that the required value of a primitive type variable can be entered.

When this parameter is set to **false**, the **read** instruction returns null and the User Input dialog is not displayed for user entry.

Valid **ReadEnabled** parameter values are **<default>**, **true**, and **false**.

StringPoolLimit

Value Type Integer *(prefix multiplier)*

Default 5M

Purpose

The **StringPoolLimit** parameter specifies the maximum amount of memory that is allocated for an interpreter string pool.

A string pool, allocated for each method cache, contains the strings and binaries used by methods in the cache.

The minimum value is **64**K bytes. The optimum value for this parameter varies, depending on the amount of physical memory available and the size of the strings and binaries that are accessed frequently.

Note When a single cache for all processes within a node is used, this parameter takes effect when the node is initialized.

When multiple caches are used, the parameter takes effect when a process is initiated.

Hint

To determine the amount by which the string pool limit is exceeded and how often the limit is exceeded, refer to the statistics for string pool usage:

- Section in your profile report. For details about reporting profile information and tracing methods in runtime applications, see "Reporting Profile Statistics" under "Profiling an Application", in Chapter 1 of the *JADE Runtime Application Guide*.
- Row in the **Method Cache Statistics** table in the **Process Information** view of the JADE Monitor. For details, see "Displaying Process Method Cache Statistics" under "Performing Popup Menu Command Actions", in Chapter 2 of the *JADE Monitor User's Guide*.
- Value in the notification **userInfo** output when calling the **Process** class **sendMethodCacheStatistics** method.

WriteEnabled

Value Type Boolean

Default true

Purpose

The **WriteEnabled** parameter, when enabled (the default), specifies that the result of a **write** instruction in a method is output to the Jade Interpreter Output Viewer when the method is executed.

Set this parameter to **false** to disable the output of the **write** instruction to the Jade Interpreter Output Viewer. When this parameter is set to **false**, the **write** instruction has no effect.

Interpreter Output Viewer Section [JadeInterpreterOutputViewer]

The parameters in the [JadeInterpreterOutputViewer] section should be set only from the Jade Interpreter Output Viewer.

Note All parameters in the [JadeInterpreterOutputViewer] section are read when the Jade Interpreter Output Viewer is next opened.

The [JadeInterpreterOutputViewer] section can contain the following parameters.

CaptureFileName

Value Type String *(file-name)*

Default jadeout.log

Purpose

The **CaptureFileName** parameter specifies the file in your JADE working directory to which output from the **write** instruction or traced methods is directed if the [JadeInterpreter] section **WriteEnabled** parameter and the [JadeInterpreterOutputViewer] section **FileCapture** parameter are set to **true**.

Hint

Set this parameter by using the common File dialog that is accessed by selecting the **Set capture file** command from the Jade Interpreter Output Viewer File menu.

FileCapture

Value Type Boolean

Default false

Purpose

The **FileCapture** parameter specifies if the output from the **write** instruction or traced methods is also output to a file.

Set this parameter by using the **Capture to file** command from the Jade Interpreter Output Viewer Options menu. (A check mark is displayed to the left of the command in the Options menu when file capture is set and this parameter is set to **true**.)

FontName

Value Type String

Default System

Purpose

The **FontName** parameter specifies the font that is used for all items displayed by the Jade Interpreter Output Viewer. The *string* value must be a valid installed font.

Hint

Set the font by using the **Fonts** command from the Jade Interpreter Output Viewer Options menu, to ensure that the selection is valid.

FontSize

Value Type Real, Real, Boolean, Boolean

Default 16, 700, false, false

Purpose

The **FontSize** parameter specifies the font attributes that are used for all items displayed by the Jade Interpreter Output Viewer.

The *real* values are the point count and the weight for that font, respectively, and the *boolean* values are for bold and italic attributes, respectively.

Hint

Set these values by using the **Fonts** command from the Jade Interpreter Output Viewer Options menu, to ensure that the selections are valid.

WindowAlwaysOnTop

Value Type Boolean

Default false

Purpose

The **WindowAlwaysOnTop** parameter specifies if the Jade Interpreter Output Viewer is always displayed on top of the current window unless it is minimized.

Set this parameter by using the **Always on top** command from the Jade Interpreter Output Viewer Options menu. (A check mark is displayed to the left of the command in the Options menu when the Jade Interpreter Output Viewer is displayed on top of the current window and this parameter is set to **true**.)

WindowPos

Value Type Integer, Integer, Integer, Integer

Default 500, 50, 300, 400

Purpose

The **WindowPos** parameter specifies the position and size of the Jade Interpreter Output Viewer.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical point coordinates of the Jade Interpreter Output Viewer pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) indicate the width and height of the window in pixels, respectively.

Hint

Set these values from within the Jade Interpreter Output Viewer by positioning the window, to assure their accuracy.

WindowTitle

Value Type String

Default <default>

Purpose

The **WindowTitle** parameter specifies the information that you want to display in the title bar of the Jade Interpreter Output Viewer; for example, **WindowTitle = "Test System"**.

If the value of the **WindowTitle** parameter is:

- Not empty and not **<default>**, the title is:

  ```
  Jade Interpreter Output Viewer - WindowTitle-parameter-string-value
  ```

- Empty or **<default>** and the initiator is a standard client, the title is:

  ```
  Jade Interpreter Output Viewer - path=database-path
  ```

- Empty or **<default>** and the initiator is a presentation client, the title is:

  ```
  Jade Interpreter Output Viewer - AppServer=application-server
  AppServerPort=port App=application-name Schema=schema-name
  ```

If the title does not use the **WindowTitle** parameter value, the information is taken from the command line.

WordWrap

Value Type Boolean

Default false

Purpose

The **WordWrap** parameter specifies if long lines of text output to the Jade Interpreter Output Viewer are wrapped so that additional lines are used to display the complete text. By default, long lines of text are truncated on the right and the text box must be scrolled horizontally to view the extended text.

Set this parameter by using the **Word wrap** command from the Jade Interpreter Output Viewer Options menu. (A check mark is displayed to the left of the command in the Options menu when word wrapping is set in the Jade Interpreter Output Viewer, and this parameter is set to **true**.)

JADE Log Section [JadeLog]

The [JadeLog] section of the JADE initialization file contains parameters that control the logging of JADE messages, including error messages.

Note The JADE message log files should not be confused with JADE transaction journal files.

The [JadeLog] section can contain the following parameters.

DisplayApplicationMessages

Value Type Boolean

Default Not specified

Purpose

The **DisplayApplicationMessages** parameter specifies whether application messages (that is, messages other than those output to the **jommsg.log** file) are output to the display server.

As only messages directed to **jommsg.log** are output to the display server by default, specify a value of **true** for this parameter if you want JADE log messages *and* application messages output to the display server.

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

ExtendedLengthPathMessages

Value Type Boolean

Default true

Purpose

The **ExtendedLengthPathMessages** parameter specifies whether JADE logs messages identifying the length, the path, and the operation when the file name length exceeds the Windows maximum path length of 260 bytes.

To disable this logging, set the value of the **ExtendedLengthPathMessages** parameter to **false**.

Parameter is read when …

An operation with a path name length greater than the Windows maximum path length of 260 bytes is attempted.

LogDirectory

Value Type String *(disk-path)*

Default Not specified

Purpose

The **LogDirectory** parameter specifies the directory for JADE message log files. If the parameter is not specified or the **LogFile** parameter in this section does not contain a path, the directory defaults to the **logs** directory at the same level as the **bin** directory.

Note If you have specified the **LogFile** parameter in the [JadeLog] section and the parameter value includes a path, your specified log file path is used in place of the default **logs** directory.

If you specify a relative directory name, the directory is assumed to be relative to the JADE HOME directory. For example, if your installation directory is **Jade\bin** (that is, your JADE HOME directory is **Jade**) and you specify **LogDirectory=MessageLogs**, the full directory path is **Jade\MessageLogs**.

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

Hint

The application log files (for example, **JadeSchema.log** and **CDFile.log**) and the reorganization (**jomreorg.log**) files are also output to this directory.

The application log files contain a history of all unhandled exceptions that are raised. These directories are relevant to both client and server nodes in a multiuser environment. Exceptions raised by the client node are logged to the log file of the client node that caused the exception.

Caution Although you can configure your JADE initialization file so that the log directory is a shared directory (in which case, all client nodes and JADE applications create log files in the same directory), this could cause problems if files have the same names, as client nodes would create the same file and the output from one exception may get mixed with the output from another client node.

LogFile

Value Type String *(file-name)*

Default jommsg*n*.log

Purpose

The **LogFile** parameter specifies the name of the default JADE messages log file. If the value specified in this parameter does not also contain a path, the file is created in the directory identified by the **LogDirectory** parameter.

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

LogServer

Value Type String

Default Not specified

Purpose

The **LogServer** parameter, which you can set only to a value of **Console** and which must be used with the **UseLogServer** parameter set to **true**, specifies that logging output generated by JADE batch utilities (for example, from the **jadloadb** or **jdbutilb** executable) is redirected to the standard output (**stdout**). This redirects all of your log output (for example, output to the **jommsg.log** file) to **stdout** so that batch wrapper scripts can capture output for redisplay, reformat, customized display, and so on.

If you do not specify the **LogServer** parameter value of **Console** or you specify **false** (the default) for the **UseLogServer** parameter, logging output is not directed to **stdout**. (If you want JADE log messages *and* application messages output to the display server, set the **DisplayApplicationMessages** parameter to **true**.)

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

Hint

You can also specify these parameters (with the preceding **JadeLog.**) as arguments on the command line of the batch utility, as shown in the following example.

```
jadloadb path=\system schemaFile=\scm\test.scm JadeLog.UseLogServer=true
JadeLog.LogServer=Console JadeLog.DisplayApplicationMessages=true
```

Batch utilities can use the specification of these parameters as arguments in the command line of a JADE batch executable or in the [JadeLog] section to capture the **jommsg.log** output to a file by using the relevant operating system file redirection operators for standard output; for example:

```
>> log.out 2>&1
```

If you use this redirection mechanism, *all* output to all JADE log objects is logged to both the relevant log file and to standard output; that is, this mechanism captures output to **jommsg.log**, **jomreorg.log**, **application logs**, and so on.

MaxLogFileSize

Value Type Integer *(prefix multiplier)*

Default 1M

Purpose

The **MaxLogFileSize** parameter specifies the maximum file size (in bytes) of the JADE messages log file. When a log file exceeds the maximum size, a new log file is created. The format of the file name is:

```
jommsgn.log
```

The *n* value is a unique log file version number; for example, **jommsg3.log**.

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

Hint

You can force the creation of a new log file before the specified maximum log file size is reached. For details, see the **JadeLog** class **rollOverLog**, **rollOverLogClient**, or **rollOverLogServer** method in Chapter 1 of the *JADE Encyclopaedia of Classes*.

<selector-name>

Value Type Integer *(unique-selector-number)*

Default Not specified

Purpose

The ***<selector-name>*** parameter is a selector name that matches the value of the **selector** property of the **JadeLog** class, to enable or disable the output of trace text or binary messages to a file for logging. (For more details, see Chapter 1 of the *JADE Encyclopaedia of Classes*.)

If a parameter set to a non-zero value has the same name as the value of the **JadeLog** class **selector** property when the application starts up, trace messages are output to the log file.

If a parameter with a corresponding name as the **JadeLog** class **selector** property is located in the JADE initialization file when the application starts up but it is set to zero (**0**) or it does not have a numeric value, tracing is disabled.

Parameter is read when …

Immediately. This value will be seen by any **JadeLog** instances that are created after the value has been defined and the JADE initialization file saved.

Hints

The value of the **selector** property and this initialization file parameter are case-insensitive.

You can specify more than one ***<selector-name>*** parameter to specify that the output of trace messages is enabled in several applications concurrently, if required.

UseLogServer

Value Type Boolean

Default Not specified

Purpose

The **UseLogServer** parameter, which must be used with the value of the **LogServer** parameter set to **Console**, specifies that logging output generated by JADE batch utilities (for example, from the **jadclient**, **jadloadb**, or **jdbutilb** executable) is redirected to the standard output (**stdout**).

This redirects all of your log output (for example, progress advice, notifications, and so on) to **stdout** so that batch wrapper scripts can capture output for redisplay, reformat, customized display, and so on.

If you do not specify the **LogServer** parameter value of **Console** or you specify **false** (the default) for this parameter, logging output is not directed to **stdout**. (If you want JADE log messages *and* application messages output to the display server, set the **DisplayApplicationMessages** parameter to **true**.)

Parameter is read when …

The node is next initialized; for example, when you restart the database server, application server, or the standard (fat) client.

Hint

You can also specify these parameters (with the preceding **JadeLog.**) as arguments on the command line of the batch utility, as shown in the following example.

```
jadloadb path=\system ini=\Jade\test\jade.ini schemaFile=\scm\test.scm
JadeLog.UseLogServer=true JadeLog.LogServer=Console
JadeLog.DisplayApplicationMessages=true
```

Batch utilities can use the specification of these parameters as arguments sin the command line of a JADE batch executable or in the [JadeLog] section to capture the **jommsg.log** output and application logging to a file by using the relevant operating system file redirection operators for standard output; for example:

```
>> log.out 2>&1
```

If you use this redirection mechanism, *all* output to all JADE log objects is logged to both the relevant log file and to standard output; that is, this mechanism captures output to **jommsg.log**, **jomreorg.log**, application logs, and so on.

UtcDateTime

Value Type Boolean

Default false

Purpose

The **UtcDateTime** parameter specifies whether the date and time stamp for the entries in the JADE message log files are in Coordinated Universal Time (UTC) or local time.

If the **UtcDateTime** parameter is not specified or it is set to **<default>**, the default value of **false** indicates that log entries are local time.

When a JADE executable starts up (for example, **jade.exe** or **jdbutilb**), a new current **UTC Time** and **Local Time** line are logged with the **Invocation:** entry. UTC dates have dash (**-**) separators; local dates have slash (*/*) separators.

The following is an example of the **jommsg.log** file output when the **UtcDateTime** parameter is set to **true**.

```
2014-09-06 22:15:11.202 020ac-302c WILBUR1A: (jade) Version 7.1.0 - Unicode,
Development only features: Enabled, OS Version: NT 6.01 build
7601 (Service Pack 1), Wow64: false, Architecture:
x64-msoft-win64-unicode
2014-09-06 22:15:11.203 020ac-302c WILBUR1A: Invocation:
"C:\Users\wilbur1\Developer\JadeDevel\jade71\Debug_Unicode\x64\bin\jade.exe"
ini=C:\Users\wilbur1\Developer\JadeDevel\jade71\Debug_Unicode\x64\etc\jade.ini
server=singleUser
2014-09-06 22:15:11.207 020ac-302c WILBUR1A: I18N: UTC Time 2013-03-06
22:15:11.204, Local Time 2014/09/07 11:15:11.204 +1300
2014-09-06 22:15:11.201 020ac-302c RPCMgr2:
RPCManager::createInstance: 346B640-1-1-NAG creates 36A1300-1-109-RPC
2014-09-06 22:15:11.218 020ac-302c RPCMgr2: TransportIdlePollInterval = 120000 ms
...
```

The following is an example of the **jommsg.log** file output when the **UtcDateTime** parameter is set to **false**.

```
2014/09/07 11:16:20.710 01b3c-2604 WILBUR1A: (jade) Version 7.1.0 - Unicode,
Development only features: Enabled, OS Version: NT 6.01 build
```

```
7601 (Service Pack 1), Wow64: false, Architecture:x64-msoft-win64-unicode
2014/09/07 11:16:20.711 01b3c-2604 WILBUR1A: Invocation:
"C:\Users\wilbur1\Developer\JadeDevel\jade71\Debug_Unicode\x64\bin\jade.exe"
ini=C:\Users\wilbur1\Developer\JadeDevel\jade71\Debug_Unicode\x64\etc\jade.ini
name=Debug_Unicode_Win64 server=singleUser
2014/09/07 11:16:20.711 01b3c-2604 WILBUR1A: I18N: UTC Time 2013-03-06
22:16:20.711, Local Time 2014/09/07 11:16:20.711 +1300
2014/09/07 11:16:20.709 01b3c-2604 RPCMgr2:
RPCManager::createInstance: 34BB640-1-1-NAG creates 3AF1300-1-109-RPC
2014/09/07 11:16:20.724 01b3c-2604 RPCMgr2: TransportIdlePollInterval = 120000 ms
...
```

Parameter is read when …

A JADE executable starts up (for example, **jade.exe** or **jdbutilb**).

JADE Monitor Sections

The JADE Monitor sections of the JADE initialization file contain information that affect the operation of the JADE Monitor application, including the application of user security restrictions and to the operation of the JADE Monitor background application.

The following sections can be placed in an application-defined JADE initialization file.

```
[JadeMonitor]
[JadeMonitorBackground]
[JadeMonitorSecurity]
```

JADE Monitor Section [JadeMonitor]

The [JadeMonitor] section contains parameters that affect the operation of the JADE Monitor. For details, see Chapter 2, "Using the JADE Monitor for System Instrumentation and Diagnosis", in the *JADE Monitor User's Guide*.

The [JadeMonitor] section can contain the following parameters.

BackgroundWindow

Value Type Integer, Integer, Integer, Integer, Integer

Default Not specified

Purpose

The **BackgroundWindow** parameter specifies the position and size of the JADE Monitor window.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates of the client MDI window in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) are the width and height of the window in pixels, respectively.

The fifth *integer* value is the current window status, indicated by the **Form**::**windowState** property values listed in the following table.

Form Class Constant	Value	Description
WindowState_Normal	0	Normal (the default)
WindowState_Minimized	1	Shrunk to an icon (that is, minimized)
WindowState_Maximized	2	Enlarged to maximum size (that is, maximized)

There is an effective default value of **0, 0, 800, 600, 0** that applies if there is no entry for the **BackgroundWindow** parameter.

The values for the **BackgroundWindow** parameter are updated on closing the JADE Monitor.

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

DefaultLogDirectory

Value Type String *(disk-path)*

Default Not specified

Purpose

The **DefaultLogDirectory** parameter specifies the absolute path of the directory to which samples are logged.

If the parameter is not specified, the directory defaults to the **logs** directory under the JADE HOME directory. For more details, see "Log File Location", in Chapter 1 of the *JADE Monitor User's Guide*.

Hint

Set the values for this parameter from within the JADE Monitor, to ensure their accuracy.

Parameter is read when …

The JADE Monitor program creates or opens a file for sample logging.

DefaultPane

Value Type String|String|String|String|String

Default Summary

Purpose

The **DefaultPane** parameter, specified in the JADE initialization file on the client, specifies the JADE Monitor view that is displayed when the JADE Monitor starts up.

You can specify one of the **Users**|**Locks**|**SystemStatistics**|**MonitorSetup**|**Summary** values. If you specify any other value, the **Summary** view is displayed.

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

DirectivesFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **DirectivesFile** parameter specifies the name and location of the directives file used by the JADE Monitor program. The directives file specifies the sets of statistics to be displayed, recorded, and details of periodic sampling.

Hint

Use the **Save Directives File** command from the File menu in the JADE Monitor program to save a directives file as specified by the **DirectivesFile** parameter in the [JadeMonitor] section, if the parameter is set.

If the parameter is not set, the file is saved as **directives.txt** in the JADE installation directory. (For details, see "Saving Directives" or "Saving the Directives File as another Name", in Chapter 2 of the *JADE Monitor User's Guide*.)

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

NavigatorClickCausesRefresh

Value Type Boolean

Default false

Purpose

The **NavigatorClickCausesRefresh** parameter specifies whether clicking in the Navigator pane of the JADE Monitor program causes a refresh action.

Hint

Use the **Navigator Click Causes Refresh** command from the Options menu to toggle whether a refresh action takes place when an item is clicked in the Navigator pane.

The **NavigatorClickCausesRefresh** parameter in the JADE initialization file is updated with the current value when the JADE Monitor program is closed.

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

ShowOverview

Value Type Boolean

Default true

Purpose

The **ShowOverview** parameter specifies whether overview information from the JADE Monitor knowledge base is displayed below the sheet for an item selected in the Navigator pane. (For details, see "Toggling the Display of Knowledge Base Information", in Chapter 2 of the *JADE Monitor User's Guide*.)

Hint

Click the **Overview** option at the top right of a sheet to toggle the display of overview information. The **ShowOverview** parameter in the JADE initialization file is updated with the current value when the JADE Monitor program is closed.

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

UIFontSize

Value Type Integer *(points)*

Default 10

Purpose

The **UIFontSize** parameter specifies the size in points of the font used to display information in the JADE Monitor program. The value can be **10**, **11**, or **12**.

Hint

Select the **Font Bigger** or **Font Smaller** command from the Options menu to increase or decrease the size of the font by one (**1**) within the allowed range. The **UIFontSize** parameter is updated with the current value when the JADE Monitor program is closed.

Parameter is read when …

The JADE Monitor is next initialized; for example, when you log off from the utility and then log back on.

JADE Monitor Background Section [JadeMonitorBackground]

The [JadeMonitorBackground] section contains parameters that affect the operation of the JADE Monitor background application. For details, see "Running the JADE Monitor as a Background Process", in Chapter 1 of the *JADE Monitor User's Guide*.

The [JadeMonitorBackground] section can contain the following parameters.

DefaultLogDirectory

Value Type String *(disk-path)*

Default Not specified

Purpose

The **DefaultLogDirectory** parameter specifies the absolute path of the directory to which sample logging files are written.

If the parameter is not specified, the directory defaults to the **logs** directory under the JADE HOME directory.

Parameter is read when …

The JADE Monitor background application creates or opens a file for sample logging.

DirectivesFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **DirectivesFile** parameter specifies the name and location of the directives file used by the JADE Monitor background application.

The directives file specifies the sets of statistics to be recorded and details of periodic sampling.

Note The JADE Monitor background application does nothing if a directives file containing the statistics to be recorded has not yet been saved to the path and file specified in the **DirectivesFile** parameter in the [JadeMonitorBackground] section (that is, if the directives file is empty, it does not exist, or it is not specified, the monitor application is launched and then waits for user input), and the monitor log file states that the default **directives.txt** file is the one in the **bin** directory.

Hint

To specify a directives file to be used by the JADE Monitor background application, use the **Save Directives File As …** command from the File menu in the JADE Monitor program. Select the directory specified by the **DirectivesFile** parameter in the [JadeMonitorBackground] section.

Parameter is read when …

The JADE Monitor background application starts.

JADE Monitor Security Section [JadeMonitorSecurity]

The [JadeMonitorSecurity] section contains parameters that define the extent of user access to JADE Monitor functions. For more details, see "JADE Monitor Security", in Chapter 2 of the *JADE Object Manager Guide*.

Note All parameters in this section are read when the JADE Monitor application is next initialized.

The [JadeMonitorSecurity] section can contain the following parameters.

MonitorDefaultUser

Value Type String (ReadOnlyUser|NormalUser|SuperUser)

Default ReadOnlyUser

Purpose

The **MonitorDefaultUser** parameter specifies the default level of access to JADE Monitor functions for a user. It applies only when value of the **MonitorSecurity** parameter is set to **Monitor** and the signed-on user is not defined elsewhere in the [JadeMonitorSecurity] section; that is, individual users can have different security access from the default user, as shown in the following example.

```
MonitorDefaultUser = ReadOnlyUser
wilbur             = SuperUser
```

The levels of access that can be assigned to JADE Monitor users are described in the following table.

Parameter Value	Description
ReadOnlyUser	Provides access to all JADE Monitor functions except those with significant system impact. This security level is the default when development security is enabled provided no other security level or menu items are enabled. The following menus are disabled for this security level. ■ **Force Off User** command ■ **Dump Selected Node** and **Dump All Nodes** commands ■ **Force Off Node** command ■ **Interrupt User** command ■ Process request statistics activities in the Process Information activity group ■ Persistent Object Activity group activities (for example, the **Most Accessed Objects** activity) ■ **Method Analysis** and **Selective Analysis** activities ■ **Lock Summary By Class**, **Oid**, **Chronology**, and **Contention** activities in the Lock Analysis activity group ■ Activities in the Node Sampling activity group
NormalUser	Provides access to all JADE Monitor functions and menus except the: ■ **Force Off User** command ■ **Force Off Node** command ■ **Dump Selected Node** and **Dump All Nodes** commands ■ **Interrupt User** command
SuperUser	Provides access to all JADE Monitor functions and menus.

MonitorSecurity

Value Type String (Monitor|DevelopmentSecurity|None)

Default None

Purpose

The **MonitorSecurity** parameter specifies how security over user access to JADE Monitor functions is determined. The options for specifying security are described in the following table.

Parameter Value	Description

Monitor	Access to JADE Monitor functions is solely defined by parameters in the [JadeMonitorSecurity] section (the **MonitorDefaultUser** parameter together with overrides for individual users). If development security is active as specified by the **DevelopmentSecurityLibrary** parameter in the [JadeSecurity] section, it is used only for initial sign on. In the following example, all users apart from "wilbur" have read-only access.
	`[JadeMonitorSecurity]`
	`MonitorSecurity = Monitor`
	`MonitorDefaultUser = ReadOnlyUser`
	`wilbur = SuperUser`

DevelopmentSecurity	Sign on and access to JADE Monitor functions is defined by values returned by calls to a user-defined library as specified by the **DevelopmentSecurityLibrary** parameter in the [JadeSecurity] section of the JADE initialization file. Other keys in the [JadeMonitorSecurity] section are ignored.
None	Allows unrestricted (**SuperUser**) access to all JADE Monitor functions. If development security is active, it is still used for initial sign on. Other keys in the [JadeMonitorSecurity] section are ignored.

JADE Non-GUI Client Section [NonGuiClient]

The JADE Non-GUI section of the JADE initialization file enables you to configure a standard (fat) client that runs only non-GUI applications. For more details, see "Running a Non-GUI Client Application using jadclient", in Chapter 1 of the *JADE Runtime Application Guide*.

The [NonGuiClient] section contains the following parameters, which affect the **jadclient** program that runs the non-GUI standard client.

GetSignonAuthDetails

Value Type String *(authorization-library-name)*[, String *(function-name)*]

Default Not required

Purpose

The **GetSignonAuthDetails** parameter specifies the name and optionally the entry-point of a library file that supplies a user code and password, if they are needed, to validate the application sign on to a non-GUI client application.

The value of **<default>** indicates that the user code and password are not required or are supplied using the **signonUser** and **signonPassword** arguments to the **jadclient** command line.

If the entry-point of the function is not supplied, it is assumed to be **getSignonDetails**.

For more details, see "Getting Non-GUI Client Sign-On Details", in Chapter 1 of the *JADE Runtime Application Guide*.

Parameter is read when …

The **jadclient** program node is started or the **jadclient** program is installed or removed as a service.

NodeName

Value Type String *(service-name)*

Default jadclient

Purpose

The **NodeName** parameter specifies the name of the service. When you install a non-GUI client service and this parameter is defined, the parameter value is assigned to the service.

If this parameter is not defined in your JADE initialization file when a non-GUI client service is installed, JADE creates the default value.

The **NodeName** parameter is the Windows service name as displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

Notes

The installation of a non-GUI client service fails if the **NodeName** parameter value is already in use; that is, each active service must have a unique **NodeName** parameter value. If you want to run more than one non-GUI client service concurrently, you must therefore have a separate initialization file for each service.

The length of the service name cannot be greater than 100 characters.

Parameter is read when …

The **jadclient** program node is started or the **jadclient** program is installed or removed as a service.

NodeNameDescription

Value Type String

Default Jade nonguiClient – *node (path)*

Purpose

The **NodeNameDescription** parameter specifies the description given to the non-GUI client when it is run as a service. The descriptive name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

When you install a non-GUI client service and this parameter is defined, the parameter value is assigned to the service.

If this parameter is not defined in your JADE initialization file when a non-GUI client service is installed, JADE creates the default value, using the **NodeName** parameter value for the *node* and the **path** argument value from the **jadclient** program command line for the *path*.

Notes

The installation of a non-GUI client service fails if the **NodeNameDescription** parameter value is already in use; that is, each active service must have a unique **NodeNameDescription** parameter value.

If you want to run more than one non-GUI client service concurrently, you must therefore have a separate initialization file for each service.

The length of the service description cannot be greater than 256 characters.

Parameter is read when …

The **jadclient** program node is started or the **jadclient** program is installed as a service.

NodeNameFriendly

Value Type String *(service-name)*

Default Not specified

Purpose

The **NodeNameFriendly** parameter specifies the *friendly* name given to the non-GUI client when it is run as a service. The friendly name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

If this parameter is not defined in your JADE initialization file when a non-GUI client service is installed, JADE uses the value specified for the **NodeName** parameter.

Parameter is read when …

The non-GUI client node is next initialized.

ReadWriteStdio

Value Type Boolean

Default true

Purpose

The **ReadWriteStdio** parameter specifies how JADE **read** and **write** instructions are handled when running the **jadclient** non-GUI client application.

If the parameter is set to the default value of **true**, the Jade Interpreter Output Viewer and User Input windows are not displayed, and the output of **write** statements is directed to the Windows Command window. If the value of the parameter is set to **false**, the Jade Interpreter Output Viewer and User Input windows are displayed as for a standard GUI application.

Parameter is read when …

The **jadclient** program node is started.

RunAsService

Value Type Boolean

Default false

Purpose

The **RunAsService** parameter is set to **true** when the non-GUI client is installed as a service. The default **RunAsService** parameter value of **false** specifies that the non-GUI client is *not* installed as a service.

Parameter is read when …

The **jadclient** program node is started.

Caution Do not change this parameter value directly. Set it only by using the following actions.

- To install a non-GUI client, set the optional **service** argument in the **jadclient** command line to **install**.

  When the non-GUI client is installed as a service, the start-up mode for the service is set to **Automatic**. Alternatively, set this argument to **remove** if you want to remove the non-GUI client application as a service or set it to **cmdline** if you want to run the non-GUI client application from the command prompt even if it is installed as a service.

- Use the standard Control Panel Services window (under **Administrative Tools**) to specify that you want the service to be started manually, or to start or stop an installed service.

- Issue the **net start** or **net stop** command from a command prompt.

For details, see "Controlling Non-GUI Client Application Services" under "Running a Non-GUI Client Application using jadclient", in Chapter 1 of the *JADE Runtime Application Guide*.

Server

Value Type String

Default MultiUser

Purpose

The **Server** parameter specifies a server name and is required only if you have not specified a **server** argument value in the command line for the **jadclient** program. If JADE cannot locate a **server** argument in the **jadclient** program command line or in this section of the initialization file, it looks for the **Server** parameter in the [Jade] section of the JADE initialization file. If no **server** argument or **Server** parameter exists, the value defaults to **MultiUser**. The other option for this parameter is **SingleUser**.

Parameter is read when …

The **jadclient** program node is started.

ServerApplication<*application-number*>

Value Type String, String, Integer, String *(schema-name, application-name[, time[, initialize-method]])*

Default Not specified

Purpose

The **ServerApplication<*application-number*>** parameter specifies the non-GUI server application that is executed when the **jadclient** program node initializes unless the optional *time* variable is also specified, in which case the application starts at the specified time.

The optional *initialize-method* variable has a maximum length of **250** characters and cannot contain commas. If the method requires a **String** or **Object** parameter, it must be specified as shown in the following example.

```
ServerApplication1 = NoteTest, NotesApp, , init(null)
```

The **initialize** method should not use any Graphical User Interface (GUI) facilities and should not invoke printing services.

The *application-number* variable of the first server application parameter must start at **1**, with the variable in other parameters, if any, having unique consecutive numbers; that is, **ServerApplication2**, **ServerApplication3**, and so on.

Each non-GUI server application specified in a **ServerApplication<*application-number*>** parameter that is to be started when the **jadclient** program node initializes executes the **initialize** method defined for the application or specified in the *initialize-method* variable. The application continues execution until it terminates by itself or the JADE system stops its execution.

To start the non-GUI server application at a specific time, specify the optional integer ***time*** variable as a four-digit number representing the start time of the application in a 24-hour time format, with valid values in the range **0000** through **2359**.

Parameter is read when …

The **jadclient** program node is started.

Hint

Non-GUI applications defined in this section are displayed in the JADE Monitor window as processes executing on the **jadclient** program node.

JADE Object Manager Environment Sections

The JADE Object Manager client node and server node initialization sections of the JADE initialization file contain information that defines your JADE environment.

Certain modules of the JADE Object Manager can use the information stored in this file to configure themselves to meet:

- Defined performance or resource constraints
- Your own configuration requirements

The following sections can be placed in an application-defined JADE initialization file or in the default JADE initialization client file.

```
[DeltaDb]
[DeltaOptions]
[JadeClient]
[JadeClientAppRestrictions]
[JadeLoader]
[JadeServer]
```

Note All storage and memory size values in these sections are specified in bytes. For details about the prefix multiplier used for JADE-defined storage and memory parameter values, see "Handling of Parameter Values" under "Format of the JADE Initialization File", earlier in this chapter.

Delta Db Section [DeltaDb]

The [DeltaDb] section of the JADE initialization file is used to specify database engine initialization parameters that are used by the delta database instance. The parameters are a subset of the parameters available in the [PersistentDb] section for a standard database instance.

This discrete section is required to avoid potential conflicts when absolute paths are used for directories such as the **JournalRootDirectory**, for example. Additionally, it allows tuning parameter values to differ from those of the root database.

If this section does not exist when a new delta database instance is initialized, the section is created and parameters with their default values are written to the section.

Delta Options Section [DeltaOptions]

The [DeltaOptions] section of the JADE initialization file contains the following parameter that relates to the checking out of methods to deltas.

SingleDelta

Value Type Boolean

Default false

Purpose

Setting the **SingleDelta** parameter to **true** imposes a restriction for the node that a method can be checked out only to a single delta.

Note Setting the parameter to **true** is ignored for methods that have already been checked out to more than one delta; it is effective only for methods that are subsequently checked out.

For more details, see "Checking Out a Method" under "Maintaining Methods Checked In and Out", in Chapter 2 of the *JADE Development Environment Administration Guide*.

Parameter is read when …

The JADE development environment is next initialized on the node.

JADE Object Manager Client Module Section [JadeClient]

The [JadeClient] section of the JADE initialization file contains parameters that affect the operation of the JADE Object Manager client module. The [JadeClient] section can contain the following parameters.

AllowKeyPathsWithoutInverses

Value Type Boolean

Default false

Purpose

The **AllowKeyPathsWithoutInverses** parameter specifies whether an exception is raised if JADE encounters a key path without inverse references, and that no automatic key maintenance for key path properties other than the first property is performed.

When a collection with key paths participates in a relationship (that is, a reference of the collection type is used as an inverse), with the exception of the first component in a key path, each class on which every subsequent component is defined must have at least one property that is an inverse reference to the previous component in the key path.

Note You must specify this parameter in the [JadeClient] section of your JADE initialization file on your client node and in the [JadeServer] section of your JADE initialization file on your server node.

For more details, see "Maintaining Dictionary Key Paths" under "Defining Keys for a Dictionary Class", in Chapter 3 of the *JADE Development Environment User's Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hints

When you set this parameter to **true**, it is the responsibility of your application to ensure that keys are not changed after an object has been added to a key path collection.

If the keys are changed, your application must properly manage the maintenance of the collection members.

AutomaticCacheCoherency

Value Type String

Default ServerDefault

Purpose

The **AutomaticCacheCoherency** parameter specifies whether objects in the object cache of a client node are always current; for example, so that you do not have to explicitly implement a caching strategy in your applications. The default value of **ServerDefault** enables cache coherency only if the **AutomaticCacheCoherencyDefault** parameter in the [JadeServer] section on the server node is set to **true**.

Set this parameter to **true** in the JADE initialization file used by the client node if you want to enable client cache coherency. When this parameter is set to **true**, cache synchronization is enabled, regardless of the setting of the **AutomaticCacheCoherency** parameter in the [JadeServer] section on the server node.

Set this parameter to **false** in the JADE initialization file used by the client node if you want to disable client cache coherency. When you set this parameter to **false**, cache synchronization is *not* enabled, regardless of the setting on the server.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Notes

With automatic cache coherency, an object updated on another node is automatically reloaded in cache, even when it is the receiver of a method currently being executed.

An out-of-date object can still be accessed even when cache coherency is enabled if the object is retrieved from the local cache while it us being updated by another node.

You can guarantee that an object is not being updated by another process only by locking it. (An out-of-date object can still be accessed while an update (or delete) notification is 'in transit'.)

When automatic cache coherency is enabled, calling the **Object** class **resynch** and **resynchObject** methods has no effect.

BackgroundProcessServerTimeout

Value Type Integer *(milliseconds)*

Default 30000

Purpose

The **BackgroundProcessServerTimeout** parameter specifies the maximum time lock operations for a background process on the client are queued before the request response is returned to you.

Note Avoid changing the value of this parameter unless the background process has locking issues. As an alternative to increasing the value of the **BackgroundProcessServerTimeout** parameter, investigate which objects are being locked and whether locks are being held for too long. For example, new nodes cannot sign on when the **system.nodes** dictionary is locked by the application. In cases like this, it is preferable to change the application to minimize the locking of system collections.

Parameter is read when …

The client node is next initialized.

DefaultProcessSaveLockCallStack

Value Type Boolean

Default false

Purpose

The **DefaultProcessSaveLockCallStack** parameter enables automatic recording of the lock call stack for all processes on all client nodes. This parameter also applies to single user nodes.

Parameter is read when …

The standard (fat) client node is next initialized.

ExternalEventOnSystem

Value Type Integer

Default 2147483647

Purpose

The **ExternalEventOnSystem** parameter specifies an optional numeric value in the range **16** through **2147483647** (**2G**) that indicates the user event caused on the **System** object (for example, to shut down a JADE system from a batch process). The specified event value must match the value specified in the **eventType** parameter in the **Object**::**beginNotification** method.

For details, see "causeEventOnSystem" under "Running a Non-GUI Client Application using jadclient", in Chapter 1 of the *JADE Runtime Application Guide*.

Parameter is read when …

The **jadclient** executable (the non-GUI client application) is next initialized.

Note If the **jadclient** program does not detect an event type specified in the command line (that is, the **causeEventOnSystem** argument), it checks in the [JadeClient] section for an **ExternalEventOnSystem** parameter. If neither of these is found, the event defaults to the maximum user events value of **2,147,483,647** (that is, 2G bytes).

IndividualLocalRequests

Value Type Boolean

Default false

Purpose

The **IndividualLocalRequests** parameter, when set to **true**, causes all requests from the client node to its local database to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual requests that are produced by the JADE sampling libraries in record type **14**, see "Individual Local Requests" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

IndividualPersistentCacheActivities

Value Type Boolean

Default false

Purpose

The **IndividualPersistentCacheActivities** parameter, when set to **true**, causes all activities in the persistent object cache of the client node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual activities that are produced by the JADE sampling libraries in record type **2**, see "Cache Buffer Activity" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

IndividualRemoteRequests

Value Type Boolean

Default false

Purpose

The **IndividualRemoteRequests** parameter, when set to **true**, causes all remote requests from the client node to the server node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual requests that are produced by the JADE sampling libraries in record type **10**, see "Individual Remote Requests" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

IndividualTransientCacheActivities

Value Type Boolean

Default false

Purpose

The **IndividualTransientCacheActivities** parameter, when set to **true**, causes all activities in the transient object cache of the node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual activities that are produced by the JADE sampling libraries in record type **2**, see "Cache Buffer Activity" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

InitializationHandlerLibrary

Value Type String *(file-name)*

Default jomsupp

Purpose

Use the **InitializationHandlerLibrary** parameter to specify the name of the initialization handler library file on the client node that executes code when there is a Remote Procedure Call (RPC) version mismatch or a mismatch in the local files of a read-only schema with those of the JADE database on the server node. If you do not specify your own initialization handler in this parameter, the **jomsupp** library file is used, which aborts the action.

The JADE kernel in the client node creates an **Exception** object when there is a file mismatch, and calls the **"initializationHandler"** entry point in the file specified in this parameter. The **Exception** object is passed to the method. (The **Exception** class **errorItem** property contains the name of the file that is mismatched.) You can code your C or C++ method to continue (if the problem has been resolved) or to abort the action.

When an RPC version mismatch occurs, the **"initializationHandler"** entry point is called with a **null Exception** object pointer. The error code is passed as the first parameter. JADE always aborts the action after this call.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

JadeBytesGetContentLimit

Value Type Integer

Default 64M

Purpose

The **JadeBytesGetContentLimit** parameter specifies the maximum size for the content of a **JadeBytes** object that can be accessed by using the **getContent** method defined in the **JadeBytes** class.

The value for the **JadeBytesGetContentLimit** parameter is in the range 256 bytes through the smaller of 1G byte or a third of the physical memory on the machine executing the **getContent** method.

Note For performance reasons, the **getContent** method should not be used with a large **JadeBytes** object. When the following code fragment is executed, the local variable **bin** must store the entire binary content of the **bytes** object, which would require the use of virtual memory.

```
bin := bytes.getContent;
```

Parameter is read when …

The **getContent** method is called.

Language

Value Type String *(language)*

Default eng

Purpose

The **Language** parameter specifies the translated message file to be used. The first three characters of the *language* that you specify are used as the suffix of the file name for the **jadmsgs** file. The default language file name is **jadmsgs.eng**, where **eng** indicates that the error messages are in English. This is the file that is copied to your JADE **bin** directory when you install JADE.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

Ensure that the first three characters of the parameter value are unique for each language.

MaxLocalProcesses

Value Type Integer *(user-processes)*

Default 0

Purpose

The **MaxLocalProcesses** parameter specifies the maximum number of users that can be signed on to a client node (a standard fat client or an application server) at the same time.

When the specified limit is reached, attempts to sign on cause a 1120 exception to be raised (*The maximum number of users are already signed on*).

The default value of zero (**0**) indicates that there is no restriction to the concurrent number of client node users.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Note The restriction applies only to user applications and JADE development environment work sessions. It does not apply to JADE utilities (for example, the JADE Monitor or the JADE Database utility).

MaxWaitAppStart

Value Type Integer *(milliseconds)*

Default 300,000 (5 minutes)

Purpose

The **MaxWaitAppStart** parameter specifies the maximum time that JADE waits for an application to initiate on another thread before raising an exception.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

If you specify this value, exception 1253 is raised if the application thread does not complete sign-on within the specified wait time.

Take the resource requirements of your system into account when you specify a value; for example, if your machine has a high system load, a value of 45,000 milliseconds (45 seconds) may not be sufficient to complete sign-on.

MozillaPrefs

Value Type String *(file-name)*

Default Not specified

Purpose

The **MozillaPrefs** parameter specifies the user preferences file (and optionally the file path) created by the Mozilla-style Web browser that contains Web browser settings (that is, the types of proxy server, hosts, and ports) that are used when opening a TCP/IP connection using a proxy server.

When a TCP/IP connection using a proxy server is attempted and the **JadeTcpIpProxy** class **browserType** property is set to **BrowserType_Netscape**, JADE checks the **MozillaPrefs** parameter for the name of a valid Mozilla-style Web browser user preferences file (which is usually called **prefs.js**). If it is not found, the values specified for **JadeTcpIpProxy** class properties are used.

An absolute path name (for example, **\Documents and Settings\All Users\Application Data\Mozilla\Profiles\default\ac3gdmwe.slt\prefs.js**) is preferable to only a file name in this parameter. If you do not specify the path, the relative path name from the current directory is used to locate the user preferences file. If the specified file cannot be found or it is not a valid Mozilla-style Web browser user preferences file, an exception is raised.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

NotificationHashSize

Value Type Integer

Default 1019

Purpose

The **NotificationHashSize** parameter specifies the hash value the JADE Object Manager uses for notifications on client nodes.

Parameter is read when …

The application server or standard (fat) client node is next initialized.

Hint

Increasing the notification hash value may be of benefit if your systems have a large number of notifications registered.

NotifyQueueDepthWarningThreshold

Value Type Integer *(*notification warning messages*)*

Default 10000

Purpose

The **NotifyQueueDepthWarningThreshold** parameter specifies the level at which the notification queue depth is reported to the **jommsg.log** file (for example, the default value of **10000** logs a message when the size of the notification delivery queue for a JADE process reaches a multiple of 10,000, 20,000, 30,000, and so on). Set this parameter to zero (**0**) if you want to disable the logging of the notification warning messages.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

ObjectCacheSizeLimit

Value Type Integer *(prefix multiplier)*

Default 80M

Purpose

The **ObjectCacheSizeLimit** parameter specifies the maximum amount of memory that is allocated by the JADE client node for caching persistent objects in the client machine.

The optimum value for this parameter varies, depending on the:

- Amount of physical memory available on the client machine
- Average amount of data modified in a typical database transaction
- Size of the *working set* of persistent objects that is accessed frequently

The minimum value is **3**M bytes and the maximum value is two-thirds of the physical memory on the client machine (with a maximum of 4G bytes on 32-bit JADE systems).

When JADE detects a parameter value that is outside the allowed range, the value is changed to be the appropriate minimum or maximum, a message is output to the **jommsg.log** file, and the JADE initialization file is updated. Node initialization then continues normally.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hints

When the JADE client reads an object, it is loaded from the server node into the client node object cache. When that object is read again, it is obtained directly from the cache. However, lock and write operations still need to coordinate the status of local object buffers with the server node so that the client node is guaranteed to lock or modify the latest object buffer.

Having a large number of objects in the client's cache could significantly reduce the network traffic in certain multiuser applications, thus improving the overall performance of the client.

The impact of this parameter is less for single user applications, where loading buffers into the client node cache does not involve communications across workstations. Additionally, if the *working set* of the application is relatively big or if it changes frequently, the effect of the cache operation is less noticeable.

Use the **persistentCacheStats** parameter in the **logObjectCaches** method of the **Node** or **System** class to log statistics of the persistent objects cache usage.

Increasing the value of the **ObjectCacheSizeLimit** parameter may reduce cache misses. (For details about sampling JADE nodes to obtain statistical information, see Chapter 4 of the *JADE Object Manager Guide*.)

PersistentLockHashSize

Value Type Integer

Default 2053

Purpose

The **PersistentLockHashSize** parameter specifies the hash value used by the JADE Object Manager for storing and locating persistent locks on client nodes.

Parameter is read when …

The application server or standard (fat) client node is next initialized.

Hint

Increasing the persistent lock hash value may be of benefit if your systems are likely to have a large number of concurrent persistent locks.

PlatformOptions

Value Type String

Default PlatformOS

Purpose

The **PlatformOptions** parameter specifies how platform-specific options are handled; that is, whether JADE run time uses current platform-specific default values or it allows flexibility for multiple-platform systems. The valid values are **<default>**, **PlatformOS**, and **MixedOS**.

This parameter defines the default end-of-line handling when using the **File** class **readLine** method.

The **<default>** and **PlatformOS** values use the current platform-specific default values. The default end-of-line handling for the **File** class **readLine** method is **CR/LF**.

The **MixedOS** value allows for multiple-platform systems. The default end-of-line handling for the **File** class **readLine** method is any combination of **CR/LF**, **LF**, or **CR**.

Parameter is read when …

The application server or standard (fat) client node is next initialized.

ProxyAutoconfigUrl

Value Type String *(Uniform Resource Locator (URL))*

Default Not specified

Purpose

When the value of the **ProxySettingsLocation** parameter is set to **ini**, the optional **ProxyAutoconfigUrl** parameter specifies the name of URL used for automatic configuration of the client node.

The value of the **ProxyAutoconfigUrl** parameter is used in the same way as the proxy automatic configuration URL in the Windows registry of the client node.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

Note This parameter applies only when the **JadeTcpIpProxy** class **proxyType** property is set to **ProxyType_Auto** and the **ProxySettingsLocation** parameter is set to **ini**.

ProxyHost

Value Type String *(host-name* or *IP-address)*

Default Not specified

Purpose

When the value of the **ProxySettingsLocation** parameter is set to **ini**, the optional **ProxyHost** parameter specifies the host name or IP address number of the proxy server controller through which a connection is made to the host server.

If the value of the **ProxySettingsLocation** parameter is set to **ini** and this parameter is not defined or a parameter value is not specified, the value of the proxy **host** property in the **JadeTcpIpProxy** class instance is used.

If your application is behind a firewall and your network administrator requires connections to the Internet to be done through a proxy server, this parameter and the **ProxyPort** parameter identify the proxy server controller through which connections are made to the host server.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

Note This parameter applies only when the **JadeTcpIpProxy** class **proxyType** property is set to **ProxyType_Auto** and the **ProxySettingsLocation** parameter is set to **ini**.

ProxyPort

Value Type Integer *(port-number)*

Default Not specified

Purpose

When the value of the **ProxySettingsLocation** parameter is set to **ini**, the optional **ProxyPort** parameter specifies the valid port number of the proxy server controller through which a connection is made to the host server.

If the value of the **ProxySettingsLocation** parameter is set to **ini** and this parameter is not defined or a parameter value is not specified, the value of the proxy **port** property in the **JadeTcpIpProxy** class instance is used.

If your application is behind a firewall and your network administrator requires connections to the Internet to be done through a proxy server, this parameter and the **ProxyHost** parameter identify the proxy server controller through which connections are made to the host server.

Note If the value of the **ProxyType** parameter is set to **auto** and the value of the **ProxyPort** property is not specified or is set to zero (**0**), the default port number is used for each attempted protocol. The default port number is **80** for HyperText Transfer Protocol (HTTP), **3128** for HyperText Transfer Protocol Secure (HTTPS), and **1080** for SOCKS V4.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

Note This parameter applies only when the value of the **JadeTcpIpProxy** class **proxyType** property is **ProxyType_Auto** and the **ProxySettingsLocation** parameter is set to **ini**.

ProxySettingsLocation

Value Type String *(file-name)*

Default Not specified

Purpose

The optional **ProxySettingsLocation** parameter specifies where the proxy server configuration details used to open a TCP/IP network connection are defined.

The valid settings of the **ProxySettingsLocation** parameter are listed in the following table.

Parameter Value	Description
auto	Retrieves network proxy settings automatically
direct or none	No proxies are used
environment	Looks for the **http_proxy** environment variable, to obtain proxy host and port numbers
ini	Reads further settings from the **ProxyAutoconfigUrl**, **ProxyHost**, **ProxyPort**, and **ProxyType** parameters
mozilla or netscape	Reads values from the file specified in the **MozillaPrefs** parameter in the [JadeClient] section
registry	Reads settings from the Windows registry

If the **ProxySettingsLocation** parameter is not defined or the value is not specified, the location of the proxy server configuration behaves as if **ProxySettingsLocation=registry** were specified.

Note This parameter applies only when the value of the **JadeTcpIpProxy** class **proxyType** property is **ProxyType_Auto**.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

ProxyType

Value Type String *(proxy-server-communications-protocol)*

Default Not specified

Purpose

When the value of the **ProxySettingsLocation** parameter is set to **ini**, the **ProxyType** parameter specifies the proxy server communications protocol through which client nodes connect to the host server.

If the value of the **ProxySettingsLocation** parameter is set to **ini** and this parameter is not defined or a parameter value is not specified, it defaults to **auto**.

The valid values of the **ProxyType** parameter are listed in the following table.

Parameter Value	Description
auto	Retrieves network proxy settings automatically (the default).
http	Protocol allowing redirection based on domain that is currently supported.
https	Attempts to connect to the destination host via a proxy that supports the HTTP **CONNECT** protocol.
socks4	Connect part only implemented in this release.
direct or none	Protocol allowing a reference from a **TcpIpConnection** object to a **JadeTcpIpProxy** object to be defined, but the network connection will not attempt to connect via a proxy server. The behavior is equivalent to having the **TcpIpConnection**::**networkProxy** property set to **null**.

Parameter is read when …

An application first uses the **JadeTcpIpProxy** class.

Notes

This parameter applies only when the **JadeTcpIpProxy** class **proxyType** property is set to **ProxyType_Auto** and the **ProxySettingsLocation** parameter is set to **ini**.

Only the HyperText Transfer Protocol (HTTP) proxy type and the connect part of the SOCKS V4 protocol is implemented (that is, SOCKS V4 binding is not implemented).

ReadOnlySchema

Value Type Boolean

Default false

Purpose

The **ReadOnlySchema** parameter directs the client node to read the system map files and user schema files from your local disk or from the network. These files are read-only and must be marked as read-only on the server node, using the **ReadOnlySchema** parameter in the [JadeServer] section of the JADE initialization file.

The local copy of the **_control.dat**, **_userdev.dat**, **_usergui.dat**, **_userint.dat**, **_userscm.dat**, and **_userxrf.dat** file is located in the directory specified by the **ReadOnlySchemaPath** parameter. The system map files are located in the directory specified in the **SystemFileDirectory** parameter of the [PersistentDb] section of the JADE initialization file.

Note This list does not include the **_rootdef.dat** file, which is the default user *data* file.

By default, schema files can be modified, and schema objects are read from the server database.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

This parameter should be set to **true** for applications in *production* mode sessions, when no changes to the database definition are to be made. The only exception to this is in very low-memory client configurations, since additional memory is used for reading the schema file.

For details about production mode, see "Running JADE Production Mode Databases", in Chapter 1 of the *JADE Runtime Application Guide*.

ReadOnlySchemaPath

Value Type String *(disk-path)*

Default Database path

Purpose

The **ReadOnlySchemaPath** parameter specifies the path for the control and user schema files when the **ReadOnlySchema** parameter is set to (**true**). Several users can share these files.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

Use a local disk whenever possible, if network traffic must be kept to a minimum.

ReadOnlySystemSchema

Value Type Boolean

Default false

Purpose

The **ReadOnlySystemSchema** parameter directs the client to read the system map files from your local disk or from the network. This provides performance improvements when:

- Starting the JADE development environment
- Accessing system-supplied objects

The location of the local copy of the **_control.dat file** is specified in the **ReadOnlySystemSchemaPath** parameter in this section. The location of the system map files is specified in the **SystemFileDirectory** parameter of the [PersistentDb] section of the JADE initialization file. See also the **ReadOnlySchema** parameter, earlier in this chapter.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

This parameter should be set to **true** for applications in *development* mode sessions. The only exception to this is in very low-memory client configurations, since additional memory is used for reading the schema file.

Note As the **ReadOnlySchema** parameter implies a read-only system schema, when the **ReadOnlySchema** parameter is set to **true**, JADE does not check the **ReadOnlySystemSchemaPath** parameter.

ReadOnlySystemSchemaPath

Value Type String *(disk-path)*

Default *database-path*

Purpose

The **ReadOnlySystemSchemaPath** parameter specifies the local copy of the control file when the **ReadOnlySystemSchema** parameter is set to (**true**). Several users can share the file.

Note JADE checks this parameter only when the **ReadOnlySchema** parameter is set to **false**.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

Use a local disk whenever possible, if network traffic must be kept to a minimum.

SamplingFilterFile

Value Type String

Default Not specified

Purpose

The **SamplingFilterFile** parameter specifies the name of the optional object filter command file in a user-supplied library, which is read each time the **Node** class or **System** class **beginSample** method is invoked. For details, see "Sampling Filtering" and "Filter Commands", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The **Node** class or **System** class **beginSample** method is next invoked.

SamplingLibraryInitialization

Value Type String *(file-name)*

Default sampling.smp

Purpose

The **SamplingLibraryInitialization** parameter specifies the string that the JADE Object Manager passes to the user-supplied library when the JADE system initializes and automatic JADE node sampling is enabled (that is, when the **SamplingNode** parameter is set to **true**).

You can use this parameter to specify any handshake or initialize information that your library requires. For details, see "JADE Sampling Libraries" under "Sampling Library Interface", in Chapter 4 of the *JADE Object Manager Guide*.

If you are using the **filesmpl** sampling library provided by JADE, you can use this parameter to specify the name of the sampling file (and optionally the file path) to which sampling is output if you want output directed to a file with a name other than the default **sampling.smp** in a destination other than the JADE installation directory (that is, in the directory in which the **jade.exe** executable program is located).

If you are using the **filesmpl** or **tcpsmpl** JADE sampling library, you can set this parameter to **"<null>"** or **""** so that sample values will not be output.

For **filesmpl**, the values will not be written to a file. For **tcpsmpl**, the values will not be sent to a TCP/IP connection. Use this parameter in situations where node sampling needs to be enabled for the **Process** class **getRequestStatistics** method but you do not want file or TCP/IP output.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

SamplingLibraryName

Value Type String *(library-name)*

Default filesmpl

Purpose

The **SamplingLibraryName** parameter specifies the name of the user-supplied library that the JADE Object Manager calls according to the sampling directives issued when JADE node sampling is enabled. For details, see "JADE Sampling Libraries" under "Sampling Library Interface", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

SamplingNode

Value Type Boolean

Default false

Purpose

The **SamplingNode** parameter, when set to **true**, automatically initiates the sampling of statistics in the client (remote) node when the JADE system is initiated. For details about the request statistics that are produced by the JADE sampling libraries in record types **8** and **9**, see "Local Requests Statistics Format" and "Remote Requests Statistics Format" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

ServerNodeSpecifications

Value Type String (*transport-type*,*remote-host*,*remote-port*[,*local-interface*[,*local-port*]]) | (JadeLocal,[Global\]*base-name*) | (HPSM,[Global\]*base-name*)

Default TcpIp, LocalHost, 6005

Purpose

The **ServerNodeSpecifications** parameter enables the client node to connect to the specified server node across the network, using TCP/IP or using local intra-machine transport on the same machine (for example, if your server node has multiple CPUs and the database server and JADE application server reside on the same machine).

Note No spaces are allowed in the server node specifications.

The values for TCP/IP parameters are:

1. *transport-type*

   Specifies the transport type to connect to the server node, which is **TcpIP**, **TcpIPv4**, **TcpIPv6**, or **TcpIPAny**. The *transport-type* literal value is case-insensitive.

   The **TcpIP** value is a synonym for **TcpIPAny**, which supports IP version 6 or IP version 4 connections. **TcpIPv4** provides IP version 4 connections only. **TcpIPv6** provides IP version 6 connections only.

   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.

2. *remote-host*

   You can specify the *remote-host* 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. If you specify a host name, all DNS-provided addresses will be attempted.

3. *remote-port*

   Specifies the valid port number or port name to connect to on the server (remote) node.

   Note A port name cannot begin with a numeric value.

   Select a unique port number or port name on the server node, to guarantee that there are no conflicts with any TCP/IP-based application. Some systems allow only a restricted range of ports. A port number should be in the range **49152** through **65535** inclusive, but any value greater than **1023** is valid. Internet Assigned Numbers Authority (IANA) port numbers in the range 0 through 1023 are reserved, those in the range **1024** through **49151** are registered and may have been assigned to specific applications, and those in the range **49152** through **65534** are dynamic, or set aside for private use.

4. [*local-interface*]

   You can specify the *local-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. If you specify a host name, all DNS-provided addresses will be attempted.

5. [*local-port*]

   Optional port number or port name on the client (local) node. Specify the *local-port* number only if it is required for network security.

If you do not want to specify a local interface but you want to specify a local port, you must specify an empty local interface field; for example:

```
ServerNodeSpecifications=TcpIp,wilbur1a,5016,,5099
```

Tip Use the **ServerNodeSpecifications** parameter if you want to select a specific network interface card to connect to your server, or the interface (host) name or IP address if you want to bind a specific network adapter in a server node that has more than one network adapter installed. (By default, JADE defaults to the primary network adapter in the node.)

To enable local intra-machine transport on the client, specify the following parameter.

```
ServerNodeSpecifications=transport-type,[Global\]base-name
```

The values for local intra-node transport are as follows.

1. *transport-type*

   Specifies the transport type to connect to the server node; that is, **JadeLocal** or **HPSM**.

   The **HPSM** transport has a lower overhead than the **JadeLocal** transport. **HPSM** is provided by the **jomsrvr2.dll**, which implements an RPC transport and server thread model.

2. *base-name*

   Specifies the name of the connection, which is a string with a maximum length of 60 characters, and must be unique to the machine.

The HPSM transport enables a database running as a service under the **Local System** account to be used by a desktop client application running under any user account. To indicate that the named pipe and shared memory are created and global objects will be visible across sessions and logons, specify the **Global** prefix tag.

The following examples show the specification of a **JadeLocal** and an **HPSM** connection.

```
ServerNodeSpecifications=JadeLocal,Server-JadeLocal

ServerNodeSpecifications=HPSM,Global\Server-HPSM
```

For more details about the local intra-machine transports, see "Hybrid Pipe Shared Memory (HPSM) Transport" and "JadeLocal Transport" under "Configuring Your Network Protocol", in Chapter 2 of the *JADE Installation and Configuration Guide*.

Parameter is read when …

The application server or the standard (fat) client node next attempts to connect to the server in multiuser mode.

Hints

If you want to override (but not update) this parameter so that you can run multiple copies of JADE using different shortcuts rather than separate JADE initialization files, you can specify the **host**, **port**, **interface**, and **localport** command line arguments in your **jade.exe** or **jadapp** shortcut.

For details, see "Specifying Arguments in the JADE Command Line" in Chapter 2 and "Handling Multiple Copies of the JADE Program" in Chapter 1, respectively, of the *JADE Installation and Configuration Guide*, and the **ServerNodeSpecifications** parameter in the [ConnectionParams] section, later in this chapter.

SspiAuthServicePrincipalName

Value Type String

Default Not specified

Purpose

When the JADE database is encrypted, a standard client or application server uses the Service Principal Name (SPN) during the connection process, to authenticate the database server. The client node terminates with error 3574 if this parameter is not present.

The **SspiAuthServicePrincipalName** parameter, which is used only when database encryption is enabled, specifies the SPN of the database server account to which the standard client or application server is establishing a connection. The simple form for this parameter is the account name used to run the database server. For more details, see "Service Principal Names", in Chapter 4 of the *JADE Database Administration Guide*.

Parameter is read when …

The client node next attempts to connect to the server in multiuser mode.

TerminateProcessOnDisconnect

Value Type Boolean

Default false

Purpose

Set the **TerminateProcessOnDisconnect** parameter to **true** to specify that the process is terminated immediately when the client detects that the network connection from the JADE server node on which the database is located has been lost.

By default, the client node process is not terminated when the connection with the server node is lost but a message is displayed the next time the client attempts to communicate with the server.

Note When you set the **TerminateProcessOnDisconnect** parameter to **true**, unwanted side effects may occur when the connection to the JADE database server came via ODBC from another application such as Microsoft Access or Excel, as this setting results in the termination of the underlying program that opened the connection.

This also applies to any native JADE Object Manager Application Programming Interface (API) connections in multiuser mode.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

TransientCacheSizeLimit

Value Type Integer *(prefix multiplier)*

Default 40M

Purpose

The **TransientCacheSizeLimit** parameter specifies the maximum amount of memory that is allocated by the JADE client node for caching transient objects in the client machine.

The optimum value for this parameter varies, depending on the amount of physical memory available on the client machine and the size of the *working set* of transient objects that are accessed frequently.

The minimum value is **3**M bytes and the maximum value is two-thirds of the physical memory on the client machine (with a maximum of 4G bytes on 32-bit JADE systems).

When JADE detects a parameter value that is outside the allowed range, the value is changed to be the appropriate minimum or maximum, a message is output to the **jommsg.log** file, and the JADE initialization file is updated. Node initialization then continues normally.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hint

Use the **transientCacheStats** or **remoteTransientCacheStats** parameter in the **logObjectCaches** method of the **Node** or **System** class to log statistics of the transient objects cache usage.

Increasing the value of the **TransientCacheSizeLimit** parameter may reduce cache misses. (For details about sampling JADE nodes to obtain statistical information, see Chapter 4 of the *JADE Object Manager Guide*.)

TransientDbPath

Value Type String *(disk-path)*

Default Database path

Purpose

The **TransientDbPath** parameter specifies the path for the transient object database, allowing each client to specify a unique path for the transient overflow. If the transient cache overflows, transient objects are written out to a .**tmp** file in the transient database specified for each node. If no **transientDbPath** is specified, JADE attempts to use the database path. If that is not available, it uses the directory specified in the **TEMP** environmental variable.

If the **PrintStatistics** parameter in the [TransientDb] section is set to **true**, transient database statistics are output to the **tdb_statistics.log** file, located in the transient database path specified by this parameter.

Parameter is read when …

The application server or the standard (fat) client node is next initialized.

Hints

Use a local disk whenever possible, if network traffic must be kept to a minimum.

If the client node does not have access to the database path or the transient database path, the cache overflow is stored in the **temp** directory. (For more details, see "JADE Object Handling", in Chapter 1 of the *JADE Object Manager Guide*.)

TransientLockHashSize

Value Type Integer

Default 2053

Purpose

The **TransientLockHashSize** parameter specifies the hash value used by the JADE Object Manager for storing and locating transient locks on client nodes.

Parameter is read when …

The application server or standard (fat) client node is next initialized.

Hint

Increasing the transient lock hash value may be of benefit if your systems are likely to have a large number of concurrent transient locks.

JADE Client Application Restrictions Section [JadeClientAppRestrictions]

The [JadeClientAppRestrictions] section of the JADE initialization file contains parameters that specify the applications that can be run from a client node when the **EnableAppRestrictions** parameter in the [JadeServer] section is set to **true**.

A client node attempting to run an application that does not match one of the parameters of the [JadeClientAppRestrictions] section receives an 1148 exception "*Application signon from this node is restricted*.

The common section [JadeClientAppRestrictions] provides default restrictions for all networks for which a specific section is not provided. A network-specific section qualifies the section name with a *network-identifier* as follows [JadeClientAppRestrictions.*network-identifier*]. For details about how the *network-identifier* is constructed from a **NetworkSpecification** parameter, see the **EnableAppRestrictions** parameter in the [JadeServer] section.

The section containing application restrictions for a client node using a TCP/IP connection on port 6005 would have the following name.

```
[JadeClientAppRestrictions.TcpIp_6005]
```

The database server uses the network-specific **JadeClientAppRestrictions** section (for example, **[JadeClientAppRestrictions.TcpIp_6005]**), if it exists, in checking whether the client node is allowed to run the application; otherwise the common section [JadeClientAppRestrictions] with an un-qualified name is used.

Application<*n*>

Value Type String *(schema-name, application-name)*

Default Not specified

Purpose

The **Application<*n*>** parameter specifies the schema name and application name of an application that can be run from a client node connected to the database server.

The ***<n>*** variable in the parameter name is a consecutive number starting at one (**1**). You can allow more than one application, by using multiple entries in the section, as follows.

```
Application1 = MySchema, MyApp
Application2 = MyOtherSchema, MyOtherApp
Application3 = AllAppSchema, *
```

In this example, only the **MyApp** application in the **MySchema** schema, the **MyOtherApp** application in the **MyOtherSchema**, and any application in the **AllAppSchema** can be executed.

An asterisk (*) can be used as a wildcard for the schema name only if an asterisk is also used for the application name, which would imply no application restrictions.

```
Application1 = *, *                  ' This is valid
Application1 = *, MyApp              ' This is not valid
```

Restrictions apply to user schemas and system schemas. If you want to run the JADE development environment, an entry must be defined for **JadeSchema** applications. For example:

```
Application1 = JadeSchema, Jade      ' JADE development only
Application1 = JadeSchema, *         ' JADE development, Painter, and so on
```

Parameter is read when …

A client node signs on to the database server.

JADE Loader Section [JadeLoader]

The [JadeLoader] section contains parameters that affect the operation of the loading of full or partial JADE schemas by using the JADE Schema Load utility.

As the batch Schema Load utility (that is, **jadloadb**) and the **JadeSchemaLoader** application do not read parameter values from this section, you must specify all values on the command line if you do not want to use default values when loading schemas in batch mode.

Notes Set these parameters only by using the appropriate controls in the Load Schema dialog. For details, see Loading Schemas using the Schema Load Utility", in the *JADE Schema Load User's Guide*.

If the Schema Load utility was run previously, these parameters default to the values that were used last time.

All parameters in this section are read when the Load Schema dialog is next displayed.

The [JadeLoader] section can contain the following parameters.

DisplayBufferSize

Value Type Integer

Default 64K

Purpose

The **DisplayBufferSize** parameter specifies the number of characters in the output buffer. The output buffer is an internal buffer for all output to the main display window. New messages are appended to this buffer and the window is refreshed from the buffer when a message is output.

You can specify the number of characters in the metric format (for example, **K**, **M**, and so on).

DontSaveSources

Value Type Boolean

Default false

Purpose

The **DontSaveSources** parameter specifies whether the source code from all JADE methods is deleted when the schema is loaded. By default, source code is retained when the schema is loaded.

If you want to release a JADE environment that does not contain source code, you must perform a full schema load for all schema loads into that environment, check the **Don't Save Sources** check box, and specify the **compileUnchangedMethods** argument with a value of **true** on the **jadload** command line.

When the methods that are being loaded already exist and the source of those methods has not changed, the methods are not compiled. The source is not updated so it is therefore not deleted.

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*.

FormsFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **FormsFile** parameter specifies the file name and the full path of your form and data definitions (**.ddb** or .**ddx**) file.

Set this parameter only by using the **Forms File Name** text box in the Load Schema dialog.

IniFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **IniFile** parameter specifies the name (and optionally the file path) of your JADE initialization file. The database path (that is, **system**) is assumed if you do not specify the file path location.

Set this parameter only by using the **Ini File Name** text box in the Load Schema dialog.

Path

Value Type String *(disk-path)*

Default Database path

Purpose

The **Path** parameter specifies the full path in which your JADE database files are located. Set this parameter only by using the **Database Path** text box in the Load Schema dialog.

SchemaFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SchemaFile** parameter specifies the file name and the full path of your schema file. Set this parameter only by using the **Schema File Name** text box in the Load Schema dialog.

Server

Value Type String

Default Not specified

Purpose

The **Server** parameter specifies the mode in which the load operation is actioned. Set this parameter only by using the **Server Type** combo box in the Load Schema dialog or the optional **server** parameter in the batch loader executable (that is, in **jadloadb**).

JADE Object Manager Server Section [JadeServer]

The [JadeServer] section contains parameters that affect the operation of the JADE server node and can contain the following parameters.

AcceptZeroEnvironmentUUID

Value Type Boolean

Default false

Purpose

When the **AcceptZeroEnvironmentUUID** parameter is set to its default value of **false**, the environment identity specified in the **server** command line argument of a client shortcut must match the identity of the database server, as shown in the following example.

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

For more details, see "**Format of the Server URI String**" in Chapter 2 in the *JADE Installation and Configuration Guide*.

if you set the **AcceptZeroEnvironmentUUID** parameter to **true**, you can specify zeroes for the environment identity in the **server** command line argument of a client shortcut, as shown in the following example.

```
server=TcpIp://localhost:6005/00000000-0000-0000-0000-000000000000
```

Parameter is read when …

The database server node is next initialized.

AllowKeyPathsWithoutInverses

Value Type Boolean

Default false

Purpose

The **AllowKeyPathsWithoutInverses** parameter specifies whether an exception is raised if JADE encounters a key path without inverse references, and that no automatic key maintenance for key path properties other than the first property is performed.

When a collection with key paths participates in a relationship (that is, a reference of the collection type is used as an inverse), with the exception of the first component in a key path, each class on which every subsequent component is defined must have at least one property that is an inverse reference to the previous component in the key path.

Note You must specify this parameter in the [JadeClient] section of your JADE initialization file on your client node and in the [JadeServer] section of your JADE initialization file on your server node.

For more details, see "Maintaining Dictionary Key Paths" under "Defining Keys for a Dictionary Class", in Chapter 3 of the *JADE Development Environment User's Guide*.

Parameter is read when ...

The application server or the standard (fat) client node is next initialized.

Hints

When you set this parameter to **true**, it is the responsibility of your application to ensure that keys are not changed after an object has been added to a key path collection.

If the keys are changed, your application must properly manage the maintenance of the collection members.

AutomaticCacheCoherency

Value Type String

Default ServerDefault

Purpose

The **AutomaticCacheCoherency** parameter specifies whether the cache for server methods and server applications on the server node require cache synchronization; for example, so that you do not have to explicitly implement a caching strategy in your applications. The default value of **ServerDefault** enables cache coherency only if the **AutomaticCacheCoherencyDefault** parameter in the [JadeServer] section on the server node is set to **true**.

Set this parameter to **true** in the JADE initialization file used by the server node if you want to enable client cache coherency.

Set this parameter to **false** in the JADE initialization file used by the server node if you want to disable cache coherency on the server node.

With automatic cache coherency, an object updated on another node is automatically reloaded in cache, even when it is the receiver of a method currently being executed.

Parameter is read when …

The database server node is next initialized.

AutomaticCacheCoherencyDefault

Value Type Boolean

Default true

Purpose

The **AutomaticCacheCoherencyDefault** parameter specifies whether cache coherency is enabled by default on client and server nodes; that is, cache coherency is the default value for clients connecting to the server and for server methods and server applications on the server node when the **AutomaticCacheCoherency** parameter in the [JadeClient] or [JadeServer] section is set to **ServerDefault**.

When the JADE development environment is run in multiuser mode, it requires cache coherency to be enabled.

Set this parameter to **false** if you want to disable cache coherency as the default for client nodes connecting to the server and for server nodes (that is, cache coherency on client nodes will be enabled only if the **AutomaticCacheCoherency** parameter in the [JadeClient] section on client nodes and the [JadeServer] section on server nodes is set to **true**).

Note If you start the JADE development environment in multiuser mode with cache coherency disabled, a message box is displayed, warning you that the system is not using cache coherency. If you choose to proceed (by clicking the **Yes** button), a message is logged in the **jommsg*n*.log** file, indicating that you are using the JADE development environment without cache coherency.

Parameter is read when …

The database server node is next initialized.

BackgroundProcessServerTimeout

Value Type Integer *(milliseconds)*

Default 30000

Purpose

The **BackgroundProcessServerTimeout** parameter specifies the maximum time lock operations for a background process on the server are queued before the request response is returned to you.

Note Avoid changing the value of this parameter unless the background process has locking issues. As an alternative to increasing the value of the **BackgroundProcessServerTimeout** parameter, investigate which objects are being locked and whether locks are being held for too long. For example, new nodes cannot sign on when the **system.nodes** dictionary is locked by the application. In cases like this, it is preferable to change the application to minimize the locking of system collections.

Parameter is read when …

The server node is next initialized.

ConcurrentLongThreads

Value Type Integer

Default 0

Purpose

The **ConcurrentLongThreads** parameter specifies the maximum number of *long threads* that the operating system can allow to concurrently process I/O completion packets. Long threads are threads to handle request types that are expected to have a relatively long execution time (for example, server execution methods).

The thread pool for long threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

If the value of this parameter is the recommended value of zero (**0**), the system allows as many concurrently running threads as there are cores in the system. (For more details, see the Microsoft MSDN **CreateIoCompletionPort** function at http://msdn.microsoft.com.)

Parameter is read when …

The database server node is next initialized.

ConcurrentShortThreads

Value Type Integer

Default 0

Purpose

The **ConcurrentShortThreads** parameter specifies the maximum number of *short threads* that the operating system can allow to concurrently process I/O completion packets. Short threads are threads to handle request types that are expected to have a short execution time (for example, a **getObject** call).

The thread pool for short threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

If the value of this parameter is the recommended value of zero (**0**), the system allows as many concurrently running threads as there are cores in the system. (For more details, see the Microsoft MSDN **CreateIoCompletionPort** function at http://msdn.microsoft.com.)

Parameter is read when …

The database server node is next initialized.

DefaultProcessSaveLockCallStack

Value Type Boolean

Default false

Purpose

The **DefaultProcessSaveLockCallStack** parameter enables automatic recording of the lock call stack for all locks by all server applications on the database server.

Parameter is read when …

The database server node is next initialized.

DisableUserSignOn

Value Type Boolean

Default false

Purpose

The **DisableUserSignOn** parameter, when set to **true**, prevents users from signing on to the JADE database; for example, when you want to backup the database on the server node. By default, users can sign on to the database; that is, this parameter is set to **false**.

When this parameter is set to **true**, any user who then tries to sign on to the JADE database is unable to do so and an exception is raised on the remote node at which the sign on was attempted.

If you set this parameter directly in the JADE initialization file, user sign on is not disabled until you next invoke the JADE Remote Node Access utility; that is, **jadrap** must first be closed down before the setting of this parameter takes effect.

Parameter is read when …

An application server or standard (fat) client attempts to sign on to the server node.

Hint

Set this parameter by using the **Disable User SignOn** command from the File menu in the JADE server node window to immediately disable users from signing on to the database. (For details, see Chapter 1, "Using the JADE Remote Node Access Utility", in the *JADE Remote Node Access Utility User's Guide*.)

DoubleDeadlockException

Value Type Boolean

Default false

Purpose

When the **DoubleDeadlockException** parameter is set to **true**, a deadlock exception is raised on both processes that cause a deadlock when locking objects in a multiuser environment. When this parameter is set to the default value of **false**, a deadlock exception is raised only on the second process that attempted to lock an object causing a deadlock condition, and the second process is forced to release its locks.

Parameter is read when …

The database server node is next initialized.

Hint

This parameter is useful for debugging purposes, as it enables you to view both of the processes on which a deadlock exception is raised. You can also use it in production systems in which you want *all* deadlock situations fully diagnosed.

DeltaDatabaseCapable

Value Type Boolean

Default false

Purpose

When the **DeltaDatabaseCapable** parameter is set to **true**, a delta database can be activated and deactivated in a database server node.

The default value of **false** prevents delta database activation.

Parameter is read when …

The database server node is next initialized.

EnableAppRestrictions

Value Type Boolean

Default <default>

Purpose

When the **EnableAppRestrictions** parameter is not specified or set to the default value (**false**), there are no restrictions on applications that can be run on a client node. When the **EnableAppRestrictions** parameter is set to **true**, only applications specified in a [JadeClientAppRestrictions.*network-identifier*] or [JadeClientAppRestrictions] section can be run, where *network-identifier* relates to a **NetworkSpecification** parameter in the [JadeServer] section.

The following steps construct a *network-identifier* from a **NetworkSpecification** parameter.

1. Begin with the *transport-type* (**Tcplp** or **JadeLocal**).
2. Append an underscore character (**_**).
3. Append the *listener-port* for the **Tcplp** transport or the *base-name* for the **JadeLocal** transport.
4. For the **Tcplp** transport, if the *host-name* or *IP-address* is specified, append an underscore character followed by the *host-name* or *IP-address*.

Note Characters other than alphanumeric characters, the hyphen (**-**), or period (**.**) are ignored.

The following table shows examples of *network-identifiers* constructed from **NetworkSpecification** parameters.

NetworkSpecification Parameter	Corresponding network-identifier
Tcplp,Enabled,63001	Tcplp_63001
JadeLocal,Enabled,Jade99-SHM	JadeLocal_Jade99-SHM
Tcplp,Enabled,63003,172.16.1.1	Tcplp_63003_172.16.1.1
Tcplp,Enabled,63003,a.company.com	Tcplp_63003_a.company.com

When a client node attempts to run an application and the parameter is set to **true**, the database server searches the [JadeClientAppRestrictions.*network-identifier*] section for the application name and schema name. If the application name and schema name are not listed, an exception is raised.

If the [JadeClientAppRestrictions.*network-identifier*] section does not exist, the database server searches the common [JadeClientAppRestrictions] section instead. If neither section exists, the application cannot be started from the client node and an exception is raised.

For details about specifying allowed applications, see the [JadeClientAppRestrictions] section.

Parameter is read when …

The database server node is next initialized.

IndividualLocalRequests

Value Type Boolean

Default false

Purpose

The **IndividualLocalRequests** parameter, when set to **true**, causes all requests from the client node to its local database to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled. For details about the individual requests that are produced by the JADE sampling libraries in record type **14**, see "Individual Local Requests" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

IndividualPersistentCacheActivities

Value Type Boolean

Default false

Purpose

The **IndividualPersistentCacheActivities** parameter, when set to **true**, causes all activities in the persistent object cache of the client node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual activities that are produced by the JADE sampling libraries in record type **2**, see "Cache Buffer Activity" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

IndividualRemoteRequests

Value Type Boolean

Default false

Purpose

The **IndividualRemoteRequests** parameter, when set to **true**, causes all remote requests from the client node to the server node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual requests that are produced by the JADE sampling libraries in record type **10**, see "Individual Remote Requests" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

IndividualRemoteTransientCacheActivities

Value Type Boolean

Default false

Purpose

The **IndividualRemoteTransientCacheActivities** parameter, when set to **true**, causes all remote requests from the client node to the server node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual activities that are produced by the JADE sampling libraries in record type **2**, see "Cache Buffer Activity" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

IndividualTransientCacheActivities

Value Type Boolean

Default false

Purpose

The **IndividualTransientCacheActivities** parameter, when set to **true**, causes all activities in the transient object cache of the node to invoke the corresponding entry point in the user library when automatic JADE node sampling is enabled.

For details about the individual activities that are produced by the JADE sampling libraries in record type **2**, see "Cache Buffer Activity" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

LockQueueCheckInterval

Value Type Integer *(milliseconds)*

Default 1000

Purpose

The **LockQueueCheckInterval** parameter specifies the time interval in which the server node checks if lock entries have timed out. Use this parameter in conjunction with the **ServerTimeout** parameter.

Parameter is read when …

The database server node is next initialized.

Hints

Set this parameter to a low number if you want prompt notification of locked resources or to a larger number if your need to know is not so great.

The smaller the interval, the greater the processor overhead on the server nodes.

Queued lock requests are checked for timeout at the interval specified by this parameter. With the default value of **1,000** milliseconds, there can be a delay of up to 1 second between when the lock timeout time is reached and when the timeout is detected, and an exception is raised. If you need the timeout to cause an exception in a shorter time than this, set this parameter to the appropriate value.

MaxLocalProcesses

Value Type Integer *(user-processes)*

Default 0

Purpose

The **MaxLocalProcesses** parameter specifies the maximum number of local user processes (server applications) that can be signed on at the same time. When the specified limit is reached, attempts to sign on cause a 1120 exception to be raised (*The maximum number of users are already signed on*).

The default value of zero (**0**) indicates that there is no restriction to the concurrent number of users.

Parameter is read when …

The database server node is next initialized.

Note The restriction applies only to user applications. It does not apply to JADE utilities (for example, the JADE Database utility).

MaxLongThreads

Value Type Integer

Default 100

Purpose

The **MaxLongThreads** parameter specifies the maximum number of *long threads* that can be added to the thread pool for long threads. Long threads are threads to handle request types that are expected to have a relatively long execution time (for example, server execution methods).

The minimum value is **1** and the maximum value is **4095**.

The thread pool of long threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

Parameter is read when …

The database server node is next initialized.

MaxNotifyThreads

Value Type Integer

Default 5

Purpose

The **MaxNotifyThreads** parameter specifies the maximum number of server threads that are required to deliver notifications to users on client nodes.

The minimum number of threads is **1**.

Parameter is read when …

The database server node is next initialized.

MaxServerThreads

Value Type Integer

Default 100

Purpose

The **MaxServerThreads** parameter specifies the maximum number of server threads for the JADE database on this server.

Parameter is read when …

The database server node is next initialized.

MaxShortThreads

Value Type Integer

Default 100

Purpose

The **MaxShortThreads** parameter specifies the maximum number of *short threads* that can be added to the thread pool for short threads. Short threads are threads to handle request types that are expected to have a short execution time (for example, **getObject** operations).

The minimum value is **2** and the maximum value is **4095**.

The thread pool of short threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

Parameter is read when …

The database server node is next initialized.

MaxWaitAppStart

Value Type Integer *(milliseconds)*

Default 0

Purpose

The **MaxWaitAppStart** parameter specifies the maximum time that JADE waits for an **ApplicationType_Non_GUI**, **ApplicationType_Non_GUI_Rest**, or **ApplicationType_Non_GUI_Web** to initiate on another thread before raising an exception. The default value of zero (**0**) indicates infinity (that is, that the application never times out).

Parameter is read when …

The database server node is next initialized.

Hints

If you specify this value, exception 1253 is raised if the application thread does not complete sign-on within the specified wait time.

Take the resource requirements of your system into account when you specify a value; for example, if your machine has a high system load, a value of 45,000 milliseconds (45 seconds) may not be sufficient to complete sign-on.

MaxWaitAppStop

Value Type Integer *(milliseconds)*

Default 45000

Purpose

The **MaxWaitAppStop** parameter specifies the maximum time maximum time that JADE waits for an **ApplicationType_Non_GUI**, **ApplicationType_Non_GUI_Rest**, or **ApplicationType_Non_GUI_Web** application to terminate, before attempting to interrupt the JADE process. The minimum value is **10,000** milliseconds.

The application interruption attempt is written to the **jommsg*n*.log** file.

Note The parameter applies only on a database server or an application server.

When the application server or database server shuts down, a message is written to the **jommsg*n*.log** file, indicating the value of this parameter; that is:

```
ApplicationThreadManager::stopAllApplicationThreads - ApplicationThread shutdown
time-out xx seconds, where xx is maxWaitAppStop / 1000
```

In this context, an application thread is a non-GUI application running on a database or application server.

All running application threads are requested to terminate. If one or more non-GUI applications are still active when the specified number of milliseconds has expired, a message is written to the **jommsg*n*.log** file, indicating the number of processes that are still active. JADE then disables execution of the **finalize** method of the application or applications, attempts to interrupt the remaining application threads, and writes a message to the log file stating the name of the schema and application of each process that is still active.

If one or more application threads does not terminate, the process interruption and logging are repeated every 60 seconds. The database server or application server must be terminated by a system operator if the application threads do not respond to interruption (for example, the process is trapped in a blocking operating system or third-party library call).

Parameter is read when …

The database or application server is shut down.

MinLongThreads

Value Type Integer

Default 15

Purpose

The **MinLongThreads** parameter specifies the minimum number of *long threads* that are present in the thread pool for long threads. Long threads are threads to handle request types that are expected to have a relatively long execution time (for example, server execution methods).

The minimum value is **1** and the maximum value is **4095**.

The thread pool of long threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

Parameter is read when …

The database server node is next initialized.

MinServerThreads

Value Type Integer

Default 20

Purpose

The **MinServerThreads** parameter specifies the minimum number of server threads for the JADE database on this server. The minimum number of server threads is **1**.

Parameter is read when …

The database server node is next initialized.

MinShortThreads

Value Type Integer

Default 20

Purpose

The **MinShortThreads** parameter specifies the minimum number of *short threads* that are present in the thread pool for short threads. Short threads are threads to handle request types that are expected to have a short execution time (for example, **getObject** operations).

The minimum value is **2** and the maximum value is **4095**.

The thread pool of short threads uses a Windows I/O completion port to distribute requests among its threads, thereby reducing the thread context switches in the server process.

Parameter is read when …

The database server node is next initialized.

NetworkSpecification<*specification-number*>

Value Type String (*transport-type*,enabled|disabled,*listener-port*[,*interface*]) |
(JadeLocal,enabled|disabled,[Global\]*base-name*) |
(HPSM,enabled|disabled,[Global\]*base-name*)

Default Not specified

Purpose

The **NetworkSpecification<*specification-number*>** parameter specifies how the server node listens for client nodes over a specified network or to connect to client nodes using the local intra-machine transport on the same machine. The **<*specification-number*>** variable in the parameter name indicates a consecutive number starting at **1**.

Multiple network specifications must have contiguous sequence numbers; for example:

```
NetworkSpecification1=TcpIp,disabled,4999
NetworkSpecification2=TcpIp,enabled,5100,127.0.0.1
NetworkSpecification3=TcpIpv6,enabled,5200,::1
NetworkSpecification4=JadeLocal,disabled,Jade
NetworkSpecification5=HPSM,enabled,Global\Jade
```

No spaces are allowed in the server network specification.

The values for the TCP/IP transport type are as follows.

1. *transport-type*

   Specifies the transport type to connect to the server node, which is **TcpIP**, **TcpIPv4**, or **TcpIPv6**. The *transport-type* literal value is not case-sensitive. **TcpIP** is synonymous with **TcpIPv4**.

2. enabled|disabled

   Specifies whether the network state is currently enabled or disabled. (By setting this value to **disabled**, you can temporarily disable the network specification without having to renumber the unique identifiers of any other **NetworkSpecification<*specification-number*>** parameters in your [JadeServer] section.)

3. *listener-port*

   Specifies the listener port number from which TCP/IP accepts connections. Select a unique port number. Each JADE database on a specific node must have a unique server port number in the range 1024 through 65534.

4. [*interface*]

   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 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. (JADE defaults to all network adapters in the node.)

If you require a fast transport mechanism between JADE modules on the same machine (for example, if your server node has multiple CPUs and the database server and JADE application server reside on the same machine), you can use a local intra-machine transport to significantly improve performance. You can also use this transport mechanism 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*.

The parameter values for the fast, local intra-machine transport are as follows.

1. *transport-type*

   Specifies the transport type to connect to the server node; that is, **JadeLocal** or **HPSM**.

   The **HPSM** transport has a lower overhead than the **JadeLocal** transport. **HPSM** is provided by the JADE **jomsrvr2.dll**, which implements an RPC transport and server thread model.

2. enabled|disabled

   Specifies whether the network state is currently enabled or disabled. (By setting this value to **disabled**, you can temporarily disable the network specification without having to renumber the unique identifiers of any other **NetworkSpecification<*specification-number*>** parameters in your [JadeServer] section.)

3. *base-name*

   Optionally specifies a string with a maximum length of 60 characters, which must be a valid server node name.

The HPSM transport enables a database running as a service under the **Local System** account to be used by a desktop client application running under any user account. To indicate that the named pipe and shared memory are created as global objects that are visible across sessions and logons, specify the **Global** prefix tag.

For details about using the **NetworkSpecification** parameter for local intra-machine transport, see "Hybrid Pipe Shared Memory (HPSM) Transport" and "JadeLocal Transport" under "Configuring Your Network Protocol", in Chapter 2 of the *JADE Installation and Configuration Guide*. (See also the [ConnectionParams] section **NetworkSpecification** parameter under "Synchronized Database Service Sections", later in this chapter.)

Parameter is read when …

The database server node is next initialized.

NodeName

Value Type String

Default JadeServerNode

Purpose

The **NodeName** parameter enables the client and server to connect across the network. Use this parameter to specify the unique name of your JADE database server from which TCP/IP accepts connections for this JADE database. Each JADE database on the server must have a unique node name.

The **NodeName** parameter is the Windows service name as displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

Notes The length of the node name cannot be greater than 255 characters.

The installation of a database server service fails if the **NodeName** parameter value is already in use or it is not specified; that is, each active service must have a unique **NodeName** parameter value.

Parameter is read when …

The database server node is next initialized.

Hint

Choose a unique name to guarantee no conflicts with other server nodes on the network; for example, **JadeTestSys**.

NodeNameDescription

Value Type String

Default Not specified

Purpose

The **NodeNameDescription** parameter specifies the description given to the database server when it is run as a service. The descriptive name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

Note The installation of a database server service fails if the **NodeNameDescription** parameter value is already in use; that is, each active service must have a unique **NodeNameDescription** parameter value.

Parameter is read when …

The database server node is next initialized.

NodeNameFriendly

Value Type String *(service-name)*

Default Not specified

Purpose

The **NodeNameFriendly** parameter specifies the *friendly* name given to the database server when it is run as a service. (The length of the node name cannot be greater than 255 characters.) The friendly name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

If this parameter is not defined in your JADE initialization file when a database server service is installed, JADE uses the value specified for the **NodeName** parameter.

Parameter is read when …

The database server node is next initialized.

NotificationHashSize

Value Type Integer

Default 1019

Purpose

The **NotificationHashSize** parameter specifies the hash value the JADE Object Manager uses for notifications on server nodes.

Parameter is read when …

The database server is next initialized.

Hint

Increasing the notification hash value may be of benefit if your systems have a large number of notifications registered.

ObjectCacheSizeLimit

Value Type Integer *(prefix multiplier)*

Default 80M

Purpose

The **ObjectCacheSizeLimit** parameter specifies the maximum amount of memory that is allocated by the JADE server node for caching persistent objects to be used by server methods and server applications. The optimum value for this parameter varies, depending on the:

- Amount of physical memory available on the database server
- Average amount of data modified in a typical server method transaction
- Size of the *working set* of persistent objects that is accessed frequently within server methods or JADE applications running on a server

The minimum value is **3**M bytes and the maximum value is two-thirds of the physical memory on the database server.

When JADE detects a parameter value that is outside the allowed range, the value is changed to be the appropriate minimum or maximum, a message is output to the **jommsg.log** file, and the JADE initialization file is updated. Node initialization then continues normally.

Parameter is read when …

The database server node is next initialized.

Hints

The JADE Object Manager maintains a cache on the server node of objects that are accessed during the execution of server methods and JADE applications running on a server.

Access to an object is faster if it is found in the cache instead of it having to be loaded from the physical database.

The size of this parameter determines the number of objects that can remain in the object cache of the server node.

Use the **persistentCacheStats** parameter in the **logObjectCaches** method of the **Node** or **System** class to log statistics of the persistent objects cache usage. Increasing the value of the **ObjectCacheSizeLimit** parameter may reduce cache misses. (For details about sampling JADE nodes to obtain statistical information, see Chapter 4 of the *JADE Object Manager Guide*.)

PersistentLockHashSize

Value Type Integer

Default 2053

Purpose

The **PersistentLockHashSize** parameter specifies the hash value used by the JADE Object Manager for storing and locating persistent locks on server nodes.

Parameter is read when …

The database server is next initialized.

Hint

Increasing the persistent lock hash value may be of benefit if your systems are likely to have a large number of concurrent persistent locks.

ReadOnlySchema

Value Type Boolean

Default false

Purpose

The **ReadOnlySchema** parameter, when set to **true**, prevents users from opening read-only schema files for update. If this parameter is **true** and a client node attempts to open schema files for update access, the operation fails with an *ACCESS_MODE_VIOLATION* error. If this parameter is **false** and a client node attempts to access the server node in read-only schema mode, a 3080 exception is raised.

By default, update access of the schema files is allowed. (For details about schema files, see "ReadOnlySchema" under "JADE Object Manager Client Module Section [JadeClient]", earlier in this chapter.)

Parameter is read when …

The database server node is next initialized.

ReadOnlySystemSchema

Value Type Boolean

Default false

Purpose

The **ReadOnlySystemSchema** parameter in this section no longer applies, as system map files are now read-only files.

See the **ReadOnlySystemSchema** parameter in the [JadeClient] section for details about using local system map files.

RemoteTransientCacheSizeLimit

Value Type Integer *(prefix multiplier)*

Default 40M

Purpose

The **RemoteTransientCacheSizeLimit** parameter specifies the maximum amount of memory that is allocated by the JADE server node for caching transient objects in the database server for server methods. The optimum value for this parameter varies, depending on the:

- Amount of physical memory available on the database server
- Size of the *working set* of transient objects that is accessed frequently

The minimum value is **3**M bytes and the maximum value is two-thirds of the physical memory on the database server (with a maximum of 4G bytes on 32-bit JADE systems). When JADE detects a parameter value that is outside the allowed range, the value is changed to be the appropriate minimum or maximum, a message is output to the **jommsg.log** file, and the JADE initialization file is updated. Node initialization then continues normally.

For details about caching transient objects for JADE applications running on a server, see the **TransientCacheSizeLimit** parameter, later in this section.

Parameter is read when …

The database server node is next initialized.

Hint

Use the **remoteTransientCacheStats** parameter in the **logObjectCaches** method of the **Node** or **System** class to log statistics of the remote transient objects cache usage. Increasing the value of the **RemoteTransientCacheSizeLimit** parameter may reduce cache misses. (For details about sampling JADE nodes to obtain statistical information, see Chapter 4 of the *JADE Object Manager Guide*.)

ResetDeltaModeOnRestart

Value Type Boolean

Default false

Purpose

When the **ResetDeltaModeOnRestart** parameter is set to **true** and the database server node is restarted, the root database is taken out of delta mode.

Caution This effectively results in the contents of the delta database being lost on server restart, as it is always recreated whenever activated.

The default value of **false** indicates that if the database server node is terminated while in delta mode, the database server node attempts to activate the delta database when it restarts.

Parameter is read when …

The database server node is next initialized.

RunAsService

Value Type Boolean

Default false

Purpose

The **RunAsService** parameter specifies whether the server node is to run as a service. By default, the server node is not run as a service. When the server node is installed as a service, the start-up mode for the service is set to automatic.

Use the Control Panel Services window to specify that you want the service to be started manually.

Set this parameter by using the Service Configuration dialog. (For details, see "Running the Server Node as a Service" under "Using the JADE Remote Node Access Utility", in the *JADE Remote Node Access Utility User's Guide*.)

Setting this parameter directly in the JADE initialization file does not install or uninstall the service.

Parameter is read when …

The database server node is next initialized (that is, the JADE Remote Node Access utility is next invoked).

Hint

The service status is displayed on the title bar when the database server is running as a service.

SamplingFilterFile

Value Type String

Default Not specified

Purpose

The **SamplingFilterFile** parameter specifies the name of the optional object filter command file in a user-supplied library, which is read each time the **Node** or **System** class **beginSample** method is invoked. For details, see "Sampling Filtering" and "Filter Commands", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The **Node** or **System** class **beginSample** method is next invoked.

SamplingLibraryInitialization

Value Type String *(file-name)*

Default sampling.smp

Purpose

The **SamplingLibraryInitialization** parameter specifies the string that the JADE Object Manager passes to the user-supplied library when the JADE system initializes and automatic JADE node sampling is enabled (that is, when the **SamplingNode** parameter is set to **true**). You can use this parameter to specify any handshake or initialize information that your library requires.

For details, see "JADE Sampling Libraries" under "Sampling Library Interface", in Chapter 4 of the *JADE Object Manager Guide*.

If you are using the **filesmpl** sampling library provided by JADE, you can use this parameter to specify the name of the sampling file (and optionally the file path) to which sampling is output if you want output directed to a file with a name other than the default **sampling.smp** in a destination other than the JADE installation directory (that is, in the directory in which the **jade.exe** executable program is located).

If you are using the **filesmpl** or **tcpsmpl** JADE sampling library, you can set this parameter to **"<null>"** or **""** so that sample values will not be output. For **filesmpl**, the values will not be written to a file.

For **tcpsmpl**, the values will not be sent to a TCP/IP connection. Use this parameter in situations where node sampling needs to be enabled for the **Process** class **getRequestStatistics** method but you do not want file or TCP/IP output.

Parameter is read when …

The database server node is next initialized.

SamplingLibraryName

Value Type String *(library-name)*

Default filesmpl

Purpose

The **SamplingLibraryName** parameter specifies the name of the user-supplied library that the JADE Object Manager calls according to the sampling directives issued when JADE node sampling is enabled. For details, see "JADE Sampling Libraries" under "Sampling Library Interface", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

SamplingNode

Value Type Boolean

Default false

Purpose

The **SamplingNode** parameter, when set to **true**, automatically initiates the sampling of statistics in the client (remote) node when the JADE system is initiated.

For details about the request statistics that are produced by the JADE sampling libraries in record types **8** and **9**, see "Local Request Statistics Format" and "Remote Requests Statistics Format" under "Statistics File Format", in Chapter 4 of the *JADE Object Manager Guide*.

Parameter is read when …

The database server node is next initialized.

ServerApplication<*application-number*>

Value Type String, String, Integer, String *(schema-name, application-name[, time[, initialize-method]])*

Default Not specified

Purpose

The **ServerApplication<*application-number*>** parameter specifies the non-GUI server application that is executed when the server node initializes or at the time specified by the optional ***time*** variable.

The optional *initialize-method* variable has a maximum length of **250** characters and cannot contain commas. If the method requires a **String** or **Object** parameter, it must be specified as shown in the following example.

```
ServerApplication1 = RootSchema, JadeRpsDataPump, , setApplicationSkin(null)
```

The **initialize** method should not use any Graphical User Interface (GUI) facilities and should not invoke printing services.

The *application-number* variable of the first server application parameter must start at **1**, with the variable in other parameters, if any, having unique consecutive numbers; that is, **ServerApplication2**, **ServerApplication3**, and so on.

Each non-GUI server application specified in a **ServerApplication<*application-number*>** parameter that is to be started when the server node initializes executes the **initialize** method defined for the application or specified in the *initialize-method* variable. When this execution has completed, client nodes can connect. The application continues execution until it terminates by itself or the database server node stops its execution. To start the non-GUI server application at a specific time, specify the optional integer ***time*** variable as a four-digit number, representing the start time of the application in a 24-hour time format, with valid values in the range **0000** through **2359**.

For details about the automatic initiation of applications in a JADE system running on a secondary database in an SDE, see "Automatically Starting Server Applications in Secondary Systems", in Chapter 10 of the *JADE Developer's Reference*.

Parameter is read when …

The database server node is next initialized.

Hints

Server non-GUI applications are displayed in the JADE Monitor window as processes executing on the server node.

If you add or remove a **ServerApplication<*application-number*>** parameter or you change any variable values for timed starts after the server has started, select the **Synch Server Apps** command from the JADE Remote Node Access utility File menu so that JADE reads the JADE initialization file again and caches the changed values.

ServerImageFile

Value Type String *(file-name)*

Default jadelogo.bmp

Purpose

The **ServerImageFile** parameter specifies the name of a valid image file that is displayed in the JADE server node window of the JADE Remote Node Access utility.

Parameter is read when …

The database server node is next initialized (that is, the JADE Remote Node Access utility is next invoked).

Hint

You can specify any valid bitmap (.**bmp**), Graphics Interchange Format (.**gif**), Tag Image File Format (.**tif**), or Joint Photographic Experts Group (.**jpg**) image format file. If your image file is not located in your JADE working directory, you must also specify the full path name.

ServerThreadIdleTimeout

Value Type Integer *(milliseconds)*

Default 300000

Purpose

The **ServerThreadIdleTimeout** parameter specifies the maximum time idle server threads remain active before they timeout.

The default value of **300,000** milliseconds specifies that server threads time out after 5 minutes. The minimum value is **10,000** milliseconds, or 10 seconds. If you do not want idle threads to time out, specify a null value (that is, **0**), to indicate infinity.

Parameter is read when …

The server node is next initialized.

ServerTimeout

Value Type Integer *(milliseconds)*

Default 10000

Purpose

The **ServerTimeout** parameter specifies the maximum time lock operations are queued on the server before the request response is returned to you. (See also the **LockQueueCheckInterval** parameter, earlier in this section.)

The optimum value for this parameter depends on how quickly you need to know that a resource is locked; that is, it is application-dependent. If it is essential that you know immediately that the resource is unavailable, you should set this time to a low value.

Parameter is read when …

The server node is next initialized.

Hint

The cost of setting this parameter to a low value is an increase in network traffic (with the message being sent to and from the server node) and an increased load on the server node (retrying the transaction when the resource is still locked).

TerminationMsgbox

Value Type String

Default None

Purpose

The **TerminationMsgbox** parameter determines the buttons on the message box that is displayed when the JADE Remote Node Access utility is closed down.

If the **TerminationMsgbox** parameter has a value of **TryAgain**, the message box displays the following buttons.

- **Cancel**

  Closes the message box.

- **Try Again**

  Rechecks to see if there are any active nodes. If there are no active nodes, **jadrap** terminates. If there are active nodes, the message box is redisplayed.

- **Continue**

  Shuts down **jadrap**, even if active nodes are attached.

If you do not specify **TryAgain** or the parameter has a **<default>** value, the message box displays **Yes** and **No** buttons only.

Parameter is read when …

The server node is closed down.

TransientCacheSizeLimit

Value Type Integer *(prefix multiplier)*

Default 40M

Purpose

The **TransientCacheSizeLimit** parameter specifies the maximum amount of memory that is allocated by the JADE server node for caching transient objects in the database server for JADE applications running on a server node.

The optimum value for this parameter varies, depending on the:

- Amount of real memory available on the database server
- Size of the *working set* of transient objects that is accessed frequently

The minimum value is **3**M bytes and the maximum value is two-thirds of the physical memory on the database server (with a maximum of 4G bytes on 32-bit JADE systems). When JADE detects a parameter value that is outside the allowed range, the value is changed to be the appropriate minimum or maximum, a message is output to the **jommsg.log** file, and the JADE initialization file is updated. Node initialization then continues normally.

See the **RemoteTransientCacheSizeLimit** parameter earlier in this section for details about caching transient objects for server methods.

Parameter is read when …

The database server node is next initialized.

Hint

Use the **transientCacheStats** parameter in the **logObjectCaches** method of the **Node** or **System** class to log statistics of the transient objects cache usage. Increasing the value of the **TransientCacheSizeLimit** parameter may reduce cache misses. (For details about sampling JADE nodes to obtain statistical information, see Chapter 4 of the *JADE Object Manager Guide*.)

TransientDbPath

Value Type String *(disk-path)*

Default Database path

Purpose

The **TransientDbPath** parameter specifies the path for the transient object database. If no **transientDbPath** is specified, JADE attempts to use the database path. If that is not available, it uses the directory specified in the **TEMP** environmental variable.

Parameter is read when …

The database server node is next initialized.

Hint

Use a local disk whenever possible if network traffic must be kept to a minimum.

TransientLockHashSize

Value Type Integer

Default 2053

Purpose

The **TransientLockHashSize** parameter specifies the hash value used by the JADE Object Manager for storing and locating transient locks on server nodes.

Parameter is read when …

The database server is next initialized.

Hint

Increasing the transient lock hash value may be of benefit if your systems are likely to have a large number of concurrent transient locks.

TransportIdlePollInterval

Value Type Integer *(milliseconds)*

Default 120000

Purpose

The **TransportIdlePollInterval** parameter specifies the ping interval and dead connection interval before the server disconnects the client node. The value is logged.

If the server is very busy for some reason, the timeout value on the client node may be too low. (The client timeout period is calculated as twice the value of the **TransportIdlePollInterval** parameter on the server node.)

The minimum value for this parameter is **5000** milliseconds and the maximum value is **Max_Integer** (#7FFFFFFF).

Parameter is read when …

When each node initializes.

UseSystemTrayIcon

Value Type Boolean

Default true

Purpose

The **UseSystemTrayIcon** parameter specifies whether the JADE server node window is automatically minimized when the JADE Remote Node Access utility starts up and has an icon placed in the system tray at the right of the Taskbar. (For details, see "Minimizing or Restoring the JADE Server Node Window", in the *JADE Remote Node Access Utility User's Guide*.)

Parameter is read when …

The database server node is next initialized.

Hints

Use the **Use System Tray Icon** command from the JADE Remote Node Access utility File menu to toggle the automatic minimizing of the window and the use of the system tray icon.

Set this parameter to **false** if you do not want the window automatically minimized after start up and displayed as an icon in the system tray.

Window

Value Type Integer, Integer, Integer, Integer

Default 160, 160, 380, 250

Purpose

The **Window** parameter specifies the position and size of the JadeServer window (accessed from the **Remote Node Access** icon).

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical point coordinates of the JadeServer window in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) indicate the width and height of the window in pixels, respectively.

Set these values only from within the JADE Remote Node Access utility, by positioning the window in its required position.

Parameter is read when …

The JADE Remote Node Access utility is next initialized.

JADE ODBC Sections

The JADE ODBC sections of the JADE initialization file contain information that applies when accessing a JADE database through the Open Database Connectivity (ODBC) interface provided by JADE and to the configuration of the server application when using the JADE thin client ODBC driver.

The following sections can be placed in an application-defined JADE initialization file.

```
[JadeOdbc]
[JadeOdbcServer]
```

JADE ODBC Section [JadeOdbc]

The [JadeOdbc] section of the JADE initialization file on the server node contains parameters that apply to accessing a JADE database through the JADE Open Database Connectivity (ODBC) driver.

For details about:

- Setting up relational views, see "Defining ODBC Inquiry Relational Views and Ad Hoc Indexes", in Chapter 9 of the *JADE Development Environment User's Guide*.
- The software that must be installed to use the JADE ODBC driver see "ODBC Requirements for External Database Coexistence or JADE ODBC Driver Usage", in Chapter 1 of the *JADE Installation and Configuration Guide*.
- Configuring the JADE ODBC driver see "Configuring a JADE ODBC Driver", in Chapter 2 of the *JADE External Interface Developer's Reference*.

The [JadeOdbc] section can contain the following parameters.

DefaultDecimalMethodLength and DefaultDecimalMethodScale

Value Type Integer, Integer

Default [46, 23]

Purpose

Set the **DefaultDecimalMethodLength** and **DefaultDecimalMethodScale** parameters to appropriate values if the ODBC client is unable to adjust the default length (precision) and scale (decimal places) values of **46** and **23**, respectively, that are used for JADE methods that have a **Decimal** return type.

In most cases, the ODBC client adjusts these to the appropriate values for the calling program. However, when using Linked Servers with SQL Server, this is not done and an error occurs. The **DefaultDecimalMethodLength** and **DefaultDecimalMethodScale** parameters enable you to work around this SQL Server limitation.

An alternative approach is to adjust the **Decimal** specification for each method that returns a **Decimal** value, using the JADE Relational View Wizard. For details, see "Renaming Tables and Columns", in Chapter 9 of the *JADE Development Environment User's Guide*.

When using the JADE ODBC thin client driver, this parameter must be set in the JADE initialization file for the server application.

Parameter is read when …

The first method returns a decimal with length and scale values of **46** and **23**, respectively, is used in the standard client ODBC driver or in the server application.

ExtractDirectory

Value Type String *(directory-name)*

Default **temp** directory

Purpose

The **ExtractDirectory** parameter specifies the location where the extract files for the result set are to be written, if required. For details, see the **ServerResultSetBufferSize** parameter.

Parameter is read when …

The extract file is required to be written.

LogException

Value Type String *(file-name)*

Default No exception is logged

Purpose

The **LogException** parameter specifies the name of the file to which exception information is logged when an exception is caught by the JADE ODBC driver.

If you do not specify the full path of the file, it is located in the directory specified by the value of the **LogDirectory** parameter in the [JadeLog] section.

When using the JADE ODBC thin client driver, this parameter must be specified in the JADE initialization file for the server application.

Parameter is read when …

The first exception occurs in a user-defined method used in an ODBC query.

LogUserMethodExceptions

Value Type Boolean

Default false

Purpose

The **LogUserMethodExceptions** parameter specifies whether an error message is output to the **jommsg.log** file if a user method used in an ODBC query raises an exception. When an exception is raised, the value of the column is returned as NULL.

Parameter is read when …

An exception occurs in a user method used in an ODBC query.

Hint

For methods or mapping methods that are mapped to ODBC columns, it is more efficient to handle exceptions within your user code and return a valid value rather than propagating the error out to the ODBC driver code.

OidFieldSeparator

Value Type Character

Default . (the period, or dot, character)

Purpose

Set the **OidFieldSeparator** parameter in the JADE initialization file on the server to a single punctuation or similar character (for example, **@**) if you want to customize the OID field separator. This can be useful when the default OID String values that are produced are misinterpreted by a third-party tool as a different ODBC type such as a Decimal.

The JADE ODBC driver formats and parses OID strings using the specified field separator.

One character only can be specified. You can specify only a punctuation or similar character; alphanumeric (alphabetic and numeric) characters and whitespace are not allowed. When the value of this parameter is set to an invalid character or a whitespace, a message is logged and the default value is used.

Notes

When using the JADE ODBC thin client driver, this parameter must be set in the JADE initialization file for the server application.

This parameter is global; that is, it takes effect for every ODBC query made in your JADE system.

Parameter is read when …

The ODBC connection is established.

QueryExecutionTraceOn

Value Type Boolean

Default false

Purpose

The **QueryExecutionTraceOn** parameter switches on or off the output of query execution tracing.

Set this parameter to **true** if you want to output query execution trace information to the **jommsg.log** file during the execution of the query.

Parameter is read when …

The ODBC connection is established.

ServerResultSetBufferSize

Value Type Integer *(prefix modifier)*

Default 1M

Purpose

The **ServerResultSetBufferSize** parameter, which accepts the metric format, specifies the size of memory to be allocated for the result set buffer. If the result set exceeds the buffer size, the result set is written out to a temporary file in the directory specified by the **ExtractDirectory** parameter. When running an ODBC server application, a buffer is allocated for each worker.

The parameter is read when the connection is established.

Parameter is read when …

The ODBC connection is established.

SortMemory

Value Type Integer

Default Value of the **ServerResultsSetBufferSize** parameter

Purpose

The **SortMemory** parameter specifies the size of memory to be allocated when sorting the result set. This value is used when result sets that overflow the result set buffer size must be sorted (for the **ORDER BY** or **DISTINCT** clause).

Multiple allocations of the specified **SortMemory** size can occur, depending on the result set size.

Parameter is read when …

A result set which was written to an extract file is sorted.

JADE ODBC Server Section [JadeOdbcServer]

The [JadeOdbcServer] section of the JADE initialization file contains parameters that apply to the configuration of an ODBC server application running on a standard client, application server, or database server node.

For details about setting up an ODBC server application and accessing it through a JADE ODBC thin client, see "Obtaining a Relational View of Your JADE Database", in Chapter 2 of the *JADE External Interface Developer's Reference*.

Note All parameters in this section are read when the JADE ODBC service application starts.

The [JadeOdbcServer] section can contain the following parameters.

ApplicationConfigFile

Value Type String *(file-name)*

Default None

Purpose

The **ApplicationConfigFile** parameter specifies the file name and the full path of a separate XML configuration file that contains runtime settings for JADE ODBC server applications.

If an XML configuration file is specified, ODBC option settings are obtained from this file. If an XML configuration file is not specified, ODBC option settings are obtained from the parameters in the [JadeOdbcServer] section.

JADE provides an application that enables you to create and maintain XML configuration files. For details about specifying options for ODBC server applications in a separate XML file, see Chapter 2 "Creating and Maintaining the Configuration File", in the *JADE External Interface Developer's Reference*.

ListenerHostName

Value Type String *(file-name)*

Default Not specified

Purpose

The **ListenerHostName** parameter specifies the host name or IP address of the local interface card to which the ODBC service listens.

If a value for this parameter is set to **0.0.0.0** or it is not set, the ODBC service listens on all interfaces. The value can be an IP version 4 or an IP version 6 address.

ListenPort

Value Type Integer *(port-number)*

Default Not specified

Purpose

The **ListenPort** parameter specifies the listener port number for which the ODBC server transport accepts connections. This must be a unique port number in the range **1024** through **65534** that does not conflict with any other TCP service running on the machine.

MaxWorkers

Value Type Integer

Default MinWorkers

Purpose

The **MaxWorkers** parameter specifies the maximum number of copies of the ODBC worker application that are available to service requests.

The number of available workers is dynamically adjusted depending on the workload.

MinWorkers

Value Type Integer

Default 1

Purpose

The **MinWorkers** parameter specifies the minimum number of copies of the ODBC worker application that are available to service requests.

The number of available workers is dynamically adjusted depending on the workload.

The process of starting up additional workers depends on at least one of the existing workers being idle. This means that you need to configure an extra worker that can respond to management events when the other workers are busy. In practical terms, this means setting **MinWorkers** = **QueueDepthLimit** + 1.

QueueDepthLimit

Value Type Integer *(outstanding-requests)*

Default 1

Purpose

The **QueueDepthLimit** parameter specifies the number of outstanding requests that can be queued before an additional ODBC worker application is started. When this limit is exceeded for a time specified by the **QueueDepthLimitTimeout** parameter, an additional ODBC worker application is started unless this causes the number of workers to exceed the value specified in the **MaxWorkers** parameter.

The process of starting up additional workers depends on at least one of the existing workers being idle. This means that you need to configure an extra worker that can respond to management events when the other workers are busy. In practical terms, this means setting **MinWorkers** = **QueueDepthLimit** + 1.

QueueDepthLimitTimeout

Value Type Integer *(seconds)*

Default 2

Purpose

The **QueueDepthLimitTimeout** parameter specifies in seconds the length of time to wait when the number of queued entries exceeds the value of the **QueueDepthLimit** parameter before an additional ODBC worker application is started.

The worker application is started only if the current number of worker applications is less than the value specified in the **MaxWorkers** parameter.

ReadTimeout

Value Type Integer *(seconds)*

Default 120

Purpose

The **ReadTimeout** parameter specifies in seconds the length of time to wait before a network read request is terminated.

The value of this parameter is used as the **JadeMultiWorkerTcpConnection** class **timeout** property value. The parameter value is translated from seconds in the **ReadTimeout** parameter to milliseconds in the **timeout** property for the TCP/IP connection between the thin client ODBC driver and the ODBC server application.

WorkerIdleTimeout

Value Type Integer *(seconds)*

Default 120

Purpose

The **WorkerIdleTimeout** parameter specifies in seconds the length of time an ODBC worker application that is idle waits before it is terminated.

The worker application is terminated only if the current number of worker applications is greater than the value specified in the **MinWorkers** parameter.

JADE Patch Control Extensions Section [JadePatchControlExtensions]

The [JadePatchControlExtensions] section of the JADE initialization file on the server node contains parameters that apply to patch control extensions. For details about patch control extensions, see "Maintaining Patch Numbers" under "Patch Versioning", in Chapter 3 of the *JADE Development Environment Administration Guide*. See also "Patch Control Hook", in Chapter 2 of the *JADE Object Manager Guide*.

Patch control extensions are enabled only when one of the following applies.

- A library is specified in the [JadeSecurity] section **DevelopmentSecurityLibrary** parameter and the [JadeSecurity] section **JadePatchControlSecurity** parameter is set to **true**.
- The [JadeSecurity] section **DevelopmentSecurityLibrary** parameter is set to the default value of **<none>** and the [JadePatchControlExtensions] section **EnablePatchControlExtensions** parameter is set to **true**.

Note All parameters in this section are read when the JADE development or Painter application next starts up.

The [JadePatchControlExtensions] section can contain the following parameters.

AutoAssignPatchNumbers

Value Type Boolean

Default Not specified

Purpose

Set the **AutoAssignPatchNumbers** parameter to **true** when adding a new patch and you want JADE to assign a patch number that cannot be altered.

By default, this parameter is not defined (that is, patch numbers are not automatically assigned).

For details about adding a patch number when patch control extensions are enabled, see "Adding a Patch Number" under "Maintaining Patch Numbers", in Chapter 3 of the *JADE Development Environment Administration Guide*.

DefaultPatchNumber

Value Type Integer

Default Not specified

Purpose

Set the **DefaultPatchNumber** parameter to the required patch number if you want to ensure that all developers work on the same patch number. When patch extensions are enabled and a user enters zero (**0**) in the **Patch No** combo box on the sign-on form, the default patch number specified in this parameter is used.

To prevent changes being made in patches other than the one required, when users have logged off from the JADE development environment and you have extracted and closed a patch number, you can then set the patch number to the required default value so that all users who log on and specify zero (**0**) in the **Patch No** combo box on the sign-on form acquire the default patch number. In addition, you should set the **PatchNumberMustExist** parameter to **true** and the **PatchNumberCanBeReopened** parameter to **false**.

Note The default patch number applies only when users enter zero (**0**) in the **Patch No** combo box on the sign-on form.

EnablePatchControlExtensions

Value Type Boolean

Default Not specified

Purpose

Set the **EnablePatchControlExtensions** parameter to **true** if you want to enable patch control extensions for your JADE database.

This parameter is ignored unless the [JadeSecurity] section **DevelopmentSecurityLibrary** parameter is set to the default value of **<none>**.

By default, this parameter is not defined (that is, patch control extensions are disabled).

PatchNumberCanBeReopened

Value Type Boolean

Default Not specified

Purpose

Set the **PatchNumberCanBeReopened** parameter to **true** if you want patch numbers specified in the **Patch No** combo box by developers on the JADE sign-on screen to be opened if they are closed when a user confirms that the patch number is to be reopened.

By default, this parameter is not defined (that is, an error message is displayed if the user tries to sign on using a closed patch number and the patch number can be reopened only by using the Patches Browser).

PatchNumberMustBeUnique

Value Type Boolean

Default Not specified

Purpose

Set the **PatchNumberMustBeUnique** parameter to **true** if you want a patch number assigned to one developer only.

By default, this parameter is not defined (that is, patch numbers can be assigned to more than one user).

An error message is displayed if this parameter is set to **true** and a user tries to use a patch number that is already in use.

PatchNumberMustExist

Value Type Boolean

Default Not specified

Purpose

Set the **PatchNumberMustExist** parameter to **true** if you want the patch numbers specified by developers on the JADE sign-on screen to be an existing patch number.

By default, this parameter is not defined (that is, a new patch number can be specified for the work session to which the developer is signing on).

PatchNumberRequired

Value Type Boolean

Default Not specified

Purpose

Set the **PatchNumberRequired** parameter to **true** if you want to enforce the specification or selection of a patch number in the **Patch No** combo box on the JADE sign-on screen by developers signing on to the JADE development environment and in the Advanced Load Options dialog when loading a schema into the JADE development environment.

By default, this parameter is not defined (that is, the specification or selection of a patch number is optional when developers sign on to JADE).

JADE Printing Section [JadePrinting]

The [JadePrinting] section of the JADE initialization file contains parameters that provide portable printing by controlling the meta file and print data formats.

Graphics commands and objects are standard Windows GUI objects. Graphical Device Interface (GDI) commands are saved in the database as Windows Enhanced Meta Files (EMF) for later reuse, replayed to the screen for print preview, or replayed to a printer for print output. These meta files can be played directly back into a GUI control (for print preview) or to a printer (for output).

The commands used for drawing are captured and converted into a format for saving in the database, in the Scalable Vector Graphics (SVG) format. (Only a small fraction of the SVG specification is used.) An application can also save printing commands in SVG format.

SVG files can be previewed. The SVG commands are converted to GDI commands. For a caveat on the use of SVG files, see "Portable Printing" under "Printer Class", in Chapter 1 of the *JADE Encyclopaedia of Classes (Volume 2)*.

As printing is also done by GDI, SVG commands are converted to GDI and passed to the printer.

The combinations of GDI, PS, and SVG formats that are permitted depend on the operating system and executable in which the JADE system is running. You can choose SVG or EMF to save print commands in the database.

Not all options are available under all operating systems and executables. Valid **jade.exe** executable combinations are listed in the following table.

Print File Format	PS Data Format	GDI Data Format
SVG	Yes	Yes
EMF	No	Yes

The format of the data sent to the printer depends on the format in which the command data has been saved and the operating system and executable that is used.

If the commands are saved as:

- EMF output can be **GDI** format only
- SVG printer output can be **GDI** or **PS**.

Notes All parameters in this section are read when the presentation client is started or when user logic creates a new instance of the **Printer** class.

To use Postscript printing, your printer must support Postscript level 2 or greater.

The [JadePrinting] section can contain the following parameters.

GeneratedImageResolution

Value Type Integer *(dots-per-inch)*

Default 300

Purpose

The **GeneratedImageResolution** parameter specifies the resolution of the images of **JadeDotNetVisualComponent**, **JadeRichText**, **ActiveXControl**, and **OleControl** controls that are generated when the format of the print meta files is SVG. The parameter is ignored when the format is EMF.

These controls cannot generate SVG commands and so an image is generated of the control at the resolution specified by the **GeneratedImageResolution** parameter.

The value of the parameter should be the number of dots per inch, in the range of the screen resolution, which is typically **96** through **2400**.

When choosing a value, balance the image quality against the size of print file; a high value for the **GeneratedImageResolution** parameter produces higher quality output but the file could take longer to print.

The following table shows the quality to expect from typical resolution settings.

Resolution	Quality of Output
150	Fair
300	Good
600	Very good

For more details, see "JADE Printing Section [JadePrinting]", earlier in this section.

PrintDataType

Value Type String

Default Operating system-dependent

Purpose

The **PrintDataType** parameter specifies the format of print data sent to a printer. Data can be sent to the printer in the following print data formats.

- Postscript (PS).
- Windows GDI commands (GDI), which is the default value.

For more details, see "JADE Printing Section [JadePrinting]", earlier in this section.

PrintFileFormat

Value Type String

Default SVG

Purpose

The **PrintFileFormat** parameter specifies the format in which print meta files are generated. These meta files can be saved in the database.

JADE print data can be saved in the database in the following file formats.

- Scalable Vector Graphics (SVG)
- Windows Enhanced Meta Files (EMF)

For more details, see "JADE Printing Section [JadePrinting]", earlier in this section.

JADE Profiler Section [JadeProfiler]

The [JadeProfiler] section of the JADE initialization file contains parameters that enable you to specify initial values for the corresponding properties in the **JadeProfiler** class and a name for the JADE profile statistics output file.

Note All parameters in this section are read when the instance of the **JadeProfiler** class is created.

For details about recording and reporting profile statistics, see the description in the **report** method of the **JadeProfiler** class, in Chapter 1 of the *JADE Encyclopaedia of Classes (Volume 1)*.

The [JadeProfiler] section can contain the following parameters.

CodeCoverageDirectory

Value Type String *(directory-name)*

Default logs/CodeCoverage

Purpose

The **CodeCoverageDirectory** parameter contains the name of the directory for the output file for the reported data during your code coverage session.

CodeCoverageFileName

Value Type String *(file-name)*

Default None

Purpose

The **CodeCoverageFileName** parameter contains the name of the output file for the reported data during your code coverage session. If you do not specify a value for the **CodeCoverageFileName** parameter or assign a value to the **codeCoverageFileName** property, the name of the output file is ***application-name_YYYYMMDD_hhmmss*.ccd**; for example, **interpreter_20090302_133220.ccd**.

If the file name does not contain a path specification, the file is output to the directory specified in the **CodeCoverageDirectory** parameter.

You can override the parameter at run time by assigning a value to the **codeCoverageFileName** property of the **JadeProfiler** instance.

MethodCount

Value Type Integer

Default 100

Purpose

The **MethodCount** parameter contains the number of methods in an application that are profiled during your profiling session.

You can override the parameter at run time by assigning a value to the **methodCount** property of the **JadeProfiler** instance.

ProfileRemoteExecutions

Value Type Boolean

Default false

Purpose

The **ProfileRemoteExecutions** parameter specifies whether method executions on remote nodes are profiled during your profiling session.

You can override the parameter at run time by assigning a value to the **profileRemoteExecutions** property of the **JadeProfiler** instance.

Set the value of this parameter to **true** if you want to profile method executions on both the client and the server nodes when running in multiuser mode.

Note Profiling method executions on both the server and client nodes causes the profiler to incur additional overhead.

ReportActualTime

Value Type Boolean

Default true

Purpose

The **ReportActualTime** parameter specifies whether method profile times are reported during your profiling session.

You can override the parameter at run time by assigning a value to the **reportActualTime** property of the **JadeProfiler** instance.

ReportCacheStatistics

Value Type Boolean

Default true

Purpose

The **ReportCacheStatistics** parameter specifies whether cache statistics are reported during your method profiling session.

You can override the parameter at run time by assigning a value to the **reportCacheStatistics** property of the **JadeProfiler** instance.

ReportInCSVFormat

Value Type Boolean

Default false

Purpose

The **ReportInCSVFormat** parameter specifies whether method profile times are output to a list of the called methods with the execution times as comma-separated values.

Set the value of this parameter to **true** if you want method profile times directed to a CSV file instead of a method call summary report log file.

You can override the parameter at run time by assigning a value to the **reportInCSVFormat** property of the **JadeProfiler** instance.

ReportLoadTime

Value Type Boolean

Default true

Purpose

The **ReportLoadTime** parameter specifies whether method load times are reported during your method profiling session.

You can override the parameter at run time by assigning a value to the **reportLoadTime** property of the **JadeProfiler** instance.

ReportMethodSize

Value Type Boolean

Default true

Purpose

The **ReportMethodSize** parameter specifies whether sizes of methods in the cache are reported during your method profiling session.

You can override the parameter at run time by assigning a value to the **reportMethodSize** property of the **JadeProfiler** instance.

Note This report is available only if the **MethodCache** parameter in the [JadeInterpreter] section is set to the default value of **multiple**.

You can use the **clearMethodCache** method to flush all methods in the cache for the process that is currently being profiled

ReportStatistics

Value Type Boolean

Default true

Purpose

The **ReportStatistics** parameter specifies whether statistics are reported during your method profiling session.

You can override the parameter at run time by assigning a value to the **reportStatistics** property of the **JadeProfiler** instance.

ReportTotalTime

Value Type Boolean

Default true

Purpose

The **ReportTotalTime** parameter specifies whether the total method profiling time is reported.

You can override the parameter at run time by assigning a value to the **reportTotalTime** property of the **JadeProfiler** instance.

ResultsFile

Value Type String *([disk-path]file-name)*

Default **JadeProfiler_<*application-name*>_*yyyyMMdd_hhmmss* .log** or **.csv**

Purpose

The **ResultsFile** parameter specifies an optional disk path and a profile file name to be used for the output of profile statistics recording times spent in methods in an application. A new method call summary report file is generated for each call to the **JadeProfiler** class **report** method.

You can override the parameter at run time by assigning a value to the **fileName** property of the **JadeProfiler** instance.

If a path is not specified in this parameter, the file is located in the log or CSV file directory specified by using the **LogDirectory** parameter in the [JadeLog] section of the JADE initialization file.

JADE Reorganization Section [JadeReorg]

The [JadeReorg] section of the JADE initialization file contains the following information relating to database reorganization.

- Directories for files created during reorganization
- Whether fast building of collections is enabled
- Memory allocated to reorganizations worker threads

For details about JADE reorganizations, see "Database Reorganization", in Chapter 14 of the *JADE Developer's Reference*.

Note All parameters in this section are read when a reorganization is next initiated.

The [JadeReorg] section can contain the following parameters.

FastBuildBTreeCollections

Value Type Boolean

Default false

Purpose

The **FastBuildBTreeCollections** parameter specifies whether the building or rebuilding of **ObjectSet** and **MemberKeyDictionary** collections is moved from the Relationship Maintenance phase to the Object Conversion phase.

Fast building of collections significantly reduces the elapsed time of large database reorganizations involving collection maintenance by using a faster extract/sort/build algorithm and by allowing the collection maintenance to be performed in parallel by multiple reorganization workers. However, additional disk space up to three times the total size of the collections being built or rebuilt is required for extract and sort files.

Note Fast building of collections is available only for non-updating reorganizations. If you enable fast building of collections, updates are disallowed when you initiate the reorganization.

For details, see "Fast Building of Collections", in Chapter 14 of the *JADE Developer's Reference*.

ReorgBackupDirectory

Value Type String *(disk-path)*

Default Database directory

Purpose

The **ReorgBackupDirectory** parameter specifies the backup directory into which are placed the temporary backups (**.bak** files) of the original database files (**.dat** files) that are modified by the reorganization or compaction.

If you specify the **<filepath>** token, the database file location is used; that is, the .**bak** files are located with the database (.**dat**) files.

If you do not specify the parameter, the database directory (**system**) is assumed.

Hint

This parameter is useful for reducing the database volume free disk space requirements of reorganizations or compactions, by locating the directory on a different physical volume to the database.

ReorgSortDirectory

Value Type String *(disk-path)*

Default Database directory

Purpose

The **ReorgSortDirectory** parameter specifies the directory in which the extract and sort files are placed when the fast building of **ObjectSet** and **MemberKeyDictionary** collections has been enabled by setting the **FastBuildBTreeCollections** parameter to **true**.

For details, see "Fast Building of Collections", in Chapter 14 of the *JADE Developer's Reference*.

If the parameter is not specified, the database directory (**system**) is assumed.

ReorgWorkDirectory

Value Type String *(disk-path)*

Default *database-directory*

Purpose

The **ReorgWorkDirectory** parameter specifies the working directory for temporary files (**.reo** files) created during file reorganization or compaction.

If you specify the **<filepath>** token, the database file location is used; that is, the .**reo** files are located with the database (.**dat**) files.

If you do not specify the parameter, the database directory (**system**) is assumed.

Hint

This parameter is useful for reducing the database volume free disk space requirements of reorganizations or compactions, by locating the directory on a different physical volume to the database.

ReorgWorkerThreads

Value Type Integer

Default 1

Purpose

The **ReorgWorkerThreads** parameter specifies the number of concurrent file reorganization operations that are allowed. The minimum number of reorganization file operations is the default value of **1** and the maximum number is **8**.

Hint

This parameter is useful for reducing the elapsed times for reorganization or compaction of database files, by reorganizing a specified number of files concurrently.

WorkerSortMemory

Value Type Integer

Default 50M

Purpose

The **WorkerSortMemory** parameter specifies the memory used by each reorganization worker for sorting dictionary keys when the fast building of **ObjectSet** and **MemberKeyDictionary** collections has been enabled by setting the **FastBuildBTreeCollections** parameter to **true**.

You can increase the amount of sort memory to improve the sort performance when the total key length exceeds 30 bytes.

For details, see "Fast Building of Collections", in Chapter 14 of the *JADE Developer's Reference*.

JADE Report Writer Section [JadeReportWriter]

The JADE Report Writer section of the JADE initialization file provides 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. (For details about the JADE Report Writer, see the JADE *Report Writer User's Guide*.)

The [JadeReportWriter] section can contain the following parameters.

PassBackException

Value Type Boolean

Default false

Purpose

The **PassBackException** parameter passes exceptions trapped by the JADE Report Writer back to the exception handler of the current application.

Parameter is read when …

The exception is raised.

Hint

If the application is running in thin client mode, specify this parameter in the JADE initialization file on the application server. If the application is running in standard (fat) client mode, specify this parameter in the [JadeReportWriter] section on each node that requires this functionality.

QueryDataFailureTraceOn

Value Type Boolean

Default false

Purpose

The **QueryDataFailureTraceOn** parameter switches on or off the Query Data Failure logging diagnostics.

Set this parameter to **true** if you want to log query data failure diagnostics, by logging the object id and kind of error that occurred when a read or lock failure error occurs during the execution of the query.

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

QueryExecTraceOn

Value Type Boolean

Default false

Purpose

The **QueryExecTraceOn** parameter switches on or off the Query Execution logging diagnostics. Set this parameter to **true** if you want to log execution information for each query to the **jommsg.log**.

Note If the **QueryStatisticsTraceOn** parameter is set to **true**, the execution statistics are suppressed when the **QueryExecTraceOn** parameter is set to **true** and they are output only with the query statistics.

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

QueryOptimizationTraceOn

Value Type Boolean

Default false

Purpose

The **QueryOptimizationTraceOn** parameter switches on or off the Query Optimization logging diagnostics.

Set this parameter to **true** if you want optimization trace information to be logged to the **jommsg log** file during the execution of the query.

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

QueryPrepTraceOn

Value Type Boolean

Default false

Purpose

The **QueryPrepTraceOn** parameter switches on or off the Query Preparation logging diagnostics.

Set this parameter to **true** if you want to log the source of the query that is to be prepared and the query plan; that is, how the query is to be evaluated in terms of tasks that can be distinguished.

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

QueryReadFailureOptionOff

Value Type Boolean

Default false

Purpose

The **QueryReadFailureOptionOff** parameter switches on or off the passing back to the calling program any read or lock error that occurs in the query phase when objects are read.

By default, when a read or lock error occurs, the object being read by the query phase defaults to a null value. In particular, this means that data paths that contain null entries will print blank.

If you want errors passed back to the calling program (that is, the JADE Report Writer) instead of being ignored when a JADE Report Writer application reads an object and a read or lock error occurs, specify this parameter, with a value of **true**. (Although this parameter is set to **false** by default, it is not displayed in your JADE initialization file until you specifically define it.)

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

QueryStatisticsTraceOn

Value Type Boolean

Default false

Purpose

The **QueryStatisticsTraceOn** parameter switches on or off the Query Execution Statistics logging diagnostics.

Set this parameter to **true** if you want to log the execution statistics for the query, including the statistics listed in the following table.

Statistic	Description
ElapsedTime	Length of time for the query to execute
NumAuthorized	Number of objects that passed the security filter method
NumDbReads	Number of objects read from the database by the query
NumIfDefineds	Number of times the query processor has converted a run-time null object reference or an object of an invalid type in an expression into a null value
NumLockFailures	Number of times the query processor detected a lock exception
NumOperations	Number of internal operations that were performed during the query (for example, database reads, join operations, and so on), which are a measure of how much work the overall query has done
NumProgressCalls	Number of times the progress dialog was updated
NumReadFailures	Number of times the query processor was unable to read an object, usually because of an *Object Not Found* error
NumResults	Number of result objects produced by the query
NumTruncations	Number of times the query processor was forced to truncate a value to fit it into the provided tuple
NumSecurityCalls	Number of times the security filter methods are actually called

Parameter is read when …

The executable first uses the JADE Report Writer to run a report.

UseAppFont

Value Type Boolean

Default true

Purpose

The **UseAppFont** parameter specifies whether JADE Report Writer applications use the font defined for the application from which they are started or **MS Sans Serif 8.25 point**. Set this parameter to **false** if the application font causes truncation on the report writer forms, which would not occur using the standard **MS Sans Serif 8.25 point** font. By default, this parameter is set to **true**, so that the JADE Report Writer application uses the font defined for the application.

Parameter is read when …

The JADE Report Writer **Configuration** or **Designer** application is started.

JADE Security Section [JadeSecurity]

The [JadeSecurity] section of the JADE initialization file contains parameters that:

- Enable the JADE authentication and encryption hooks
- Provide security access to tasks in your JADE development environment

With the exception of schema source file encryption and JADE development environment security that are disabled by default, all other security elements are enabled by default. For details about JADE security, see "JADE Security", in Chapter 2 of the *JADE Object Manager Guide.*

The security parameters are server-driven; that is, if they are configured on a server-capable node, client nodes attaching to the server must conform to the security requirements. If the security parameters are altered on a client node, security cannot be breached.

Notes This server and client security enforcement assumes that the environment hosting a JADE server node prevents unauthorized tampering with this configuration, either directly by physical access or indirectly by remote access. This level of security can be achieved by using file security.

All parameters in this section are read when the database is next initialized; for example, when you restart the database server.

The [JadeSecurity] section can contain the following parameters.

AuthenticationEnabled

Value Type Boolean

Default true

Purpose

When the **AuthenticationEnabled** parameter is set to **true** (enabled), the connection authentication protocol is enforced.

When enabled, remote connection is prevented if the authentication library or entry-points are missing or invalid at a JADE server.

Set this parameter to **false** to disable the enforcement of the authentication protocol.

AuthenticationHookDLL

Value Type String *(encryption-library-name)*

Default Internal

Purpose

Use the **AuthenticationHookDLL** parameter to identify the JADE or user-supplied authentication library. If connection authentication is enabled, JADE attempts to load the specified library and the required entry points dynamically during initialization.

If the **AuthenticationHookDLL** parameter is set to **Internal**, the library that is used is the one specified by the **jomsrvr2** parameter in the [JadeEnvironment] section.

When authentication is enabled:

- If the dynamic load of the library or any of the required entry-points fails on a JADE client-only node, the server connections are refused and an exception is raised.
- If the dynamic load fails on a remote access-capable node, an exception is logged in the JADE event log and remote access provider support is disabled, preventing any clients from connecting.

DevelopmentSecurityLibrary

Value Type String *(security-library-name)*

Default None

Purpose

Use the **DevelopmentSecurityLibrary** parameter to:

- Specify that security is enabled for the JADE development environment. JADE development environment security is enabled only when you specify a value in this parameter.
- Identify your JADE development environment security library.

When you specify a value in this parameter, JADE attempts to load the specified library and the required entry points when the first user signs on to the JADE development environment.

If you have specified a value for this parameter but JADE cannot find the library that contains your security hook entry points:

- JADE development environment security is enabled.
- Any developer who does not have the specified library or who does not have access to this library is prevented from using the JADE development environment.

For details, see "JADE Development Environment Security", in Chapter 2 of the *JADE Object Manager Guide*.

JadePatchControlSecurity

Value Type Boolean

Default Not specified

Purpose

Set the **JadePatchControlSecurity** parameter to **true** (enabled) when a library is specified in the **DevelopmentSecurityLibrary** parameter and you want to enable patch control extensions. (For details, see "Maintaining Patch Numbers", in Chapter 3 of the *JADE Development Environment Administration Guide*.)

Patch control extensions are enabled only when one of the following applies.

- A library is specified in the **DevelopmentSecurityLibrary** parameter in the [JadeSecurity] section and the **JadePatchControlSecurity** parameter in the [JadeSecurity] section is set to **true**.
- The **DevelopmentSecurityLibrary** parameter is set to the default value of **<none>** and the **EnablePatchControlExtensions** parameter in the [JadePatchControlExtensions] section is set to **true**.

MethodTrackingEnabled

Value Type Boolean

Default false

Purpose

Use the **MethodTrackingEnabled** parameter in the JADE initialization file of the database server to control whether method tracking is allowed. Set this parameter to **true** to allow method tracking. This setting applies system-wide.

When set to **false**, any attempt to use the **Process** class **startMethodTracking** method results in an exception being raised.

The parameter value is checked for a process the first time the process calls the **startMethodTracking** method.

You can change the JADE initialization file at any time to alter the value of this parameter without having to stop and restart the database server node.

RPCEncryptionEnabled

Value Type Boolean

Default true

Purpose

When the **RPCEncryptionEnabled** parameter is set to **true** (enabled), the RPC encryption protocol is enforced. Set this parameter to **false** to disable the enforcement of the encryption protocol.

When enabled, remote connection is prevented if the encryption library or entry points are missing or invalid at a JADE server.

RPCEncryptionHookDLL

Value Type String *(encryption-library-name)*

Default Internal

Purpose

Use the **RPCEncryptionHookDLL** parameter to identify the JADE or user-supplied encryption library. If RPC encryption is enabled, JADE attempts to load the specified library and the required entry points dynamically during initialization.

If the **RPCEncryptionHookDLL** parameter is set to **Internal**, the library that is used is the one specified by the **jomsrvr2** parameter in the [JadeEnvironment] section.

When RPC encryption is enabled:

- If the dynamic load of the library or any of the required entry points fails on a JADE client-only node, the server connections are refused and an exception is raised.
- If the dynamic load fails on a remote access-capable node, an exception is logged in the JADE event log and remote access provider support is disabled, preventing any clients from connecting.

For details, see "Network Message Encryption" under "JADE Security", in Chapter 2 of the *JADE Object Manager Guide*.

SchemaEncryptionHookLibrary

Value Type String *(encryption-library-name)*

Default Default encryption algorithm

Purpose

Use the **SchemaEncryptionHookLibrary** parameter to identify your user-supplied schema source or the JADE default algorithm encryption library. (JADE provides a default encryption algorithm that is used if you do not specify an encryption library and schema source encryption is set.)

JADE attempts to load the specified library and the required encryption and decryption entry points when extracting a schema if source encryption is enabled or when loading a schema file with encrypted method sources.

For details, see "Encrypting Schema Source Files", in Chapter 10 of the *JADE Development Environment User's Guide*.

JADE Sentinel Process Dump Section [JadeSentinel]

The [JadeSentinel] section of the JADE initialization file contains a parameter that specifies how an out-of-process process dump is carried out. For more details about Sentinel process dumps, see "Using the Sentinel Process Dumps Program", in Chapter 2 of the *JADE Installation and Configuration Guide*.

ExcludeDiskCache

Value Type Boolean

Default true

Purpose

When the **ExcludeDiskCache** parameter is set to **true** (enabled), disk cache information is not included in the process dump that is generated for a JADE executable by its associated **jadesentinel.exe** program.

Note You should set the value of this parameter to **false** only when requested to do so by JADE Support.

Parameter is read when …

When the JADE executable starts up.

JADE Start-Up Section [Jade]

Use the [Jade] section of your JADE initialization file to customize the start-up image displayed when the JADE executable program (**jade.exe**) opens the database and initializes the application.

Note The **StatusPos**, **ApplicationPos**, **SchemaPos**, **ServerPos**, **PathPos**, **VersionPos**, and **AviPos** splash screen parameter values in the [Jade] section are based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The following subsections describe the parameters in the [Jade] section of the JADE initialization file.

ApplicationFontColors

Value Type String, Real, Boolean, Integer, Integer

Default Correct default values

Purpose

The **ApplicationFontColors** parameter specifies the font attributes and color of the **Application:** label and your application name passed by the command line on the splash screen.

The parameter values represent *font-name*, *font-size*, *font-bold*, *color1*, and *color2*. The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

The *color1* value applies to the **Application:** label, and the *color2* value applies to the label containing the name of application. For example, if you set this parameter to **Arial, 12, true, 16777215, 255** and then start the JADE development environment, the **Application:** label is white, the **Jade** label is red, and both labels use the Arial 12 point bold font.

Parameter is read when …

The client node is next initialized.

ApplicationPos

Value Type Integer, Integer, Integer, Integer

Default Correct default values

Purpose

The **ApplicationPos** parameter specifies the position of the **Application:** label and your application name passed by the command line on the start-up, or splash, screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates of the **Application:** label in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) are the horizontal and vertical coordinates of the application name text in pixels, respectively, relative to the top left corner of the window.

This parameter positions the displayed application label and text, which are displayed in the following form.

```
Application: target-application-name
```

If the positions are not valid or you specify **No**, the application label and text are not displayed.

Parameter is read when ...

The client node is next initialized.

AviFile

Value Type String *(file-name)*

Default Internal AVI

Purpose

The **AviFile** parameter specifies a video image (AVI file) to be played on start-up. The AVI file can be any file handled by the presentation client interface, including the following.

- AVI files with sound
- GIF (Graphics Interchange Format) file, which is positioned according to the value of the **AviPos** parameter.
- MIDI files
- WAV files, in which no visible image is displayed

If the specified file does not exist or it is not a valid type (for example, if you specify **None**), it is ignored. The value of **<default>** or **jade** specifies that JADE displays an internal resource (rather than an external file).

The value of **<none>** or **none** disables the display of an AVI file on the splash screen.

Note When creating your AVI image, ensure that your bitmap is realistically sized so that it fits the JADE start-up window.

Parameter is read when ...

The client node is next initialized.

Hints

Use a simple AVI file without sound, as it displays faster and loops seamlessly. (It uses the Microsoft Animation class.)

Do not use AVI files wrapped in an OLE object (for example, those created by CorelDraw), as this causes the start-up process to be delayed while OLE libraries are loaded and initialized and while the OLE Server is loaded to play the OLE file.

AviPos

Value Type Integer, Integer

Default Correct default values

Purpose

The **AviPos** parameter specifies the position of your AVI window. The AVI window is automatically sized to the AVI or GIF (Graphics Interchange Format) image.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

This parameter is ignored if you specify an invalid AVI or GIF file.

The *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates in pixels, respectively, relative to the top left corner of the window.

Parameter is read when …

The client node is next initialized.

FormsUseDefaultSchemaLocale

Value Type Boolean

Default false

Purpose

When the **FormsUseDefaultSchemaLocale** parameter is set to **true**, the translation of a form is always selected from the schema default locale, regardless of the current locale of the application.

Parameter is read when …

The application is next initialized.

LoadStyleDefault

Value Type String

Default Not specified

Purpose

The **LoadStyleDefault** parameter specifies the default load style that is selected when the Load Options dialog is displayed in the JADE development environment (that is, you clicked the **Load** toolbar button or selected the **Load** command from the Schema menu).

Note Specifying a valid value in this parameter is the minimum requirement to set the default load style value.

The valid values that you can define for this parameter are as follows.

- **PRIOR**, which displays *Load into the existing current schema version*
- **LATEST**, which displays *Add new schema, or load as latest schema version*
- **STRUCTURAL**, which displays *Load into latest versioning only structural changes*

If you do not specify a valid value for this parameter, the JADE-defined values are used (that is, three entries are displayed in the **Load Style** combo box for selection, with *Add new schema, or load as latest schema version* selected by default).

Parameter is read when …

The Load Options dialog is opened.

LoadStyleLookOnClient

Value Type Boolean

Default true

Purpose

The **LoadStyleLookOnClient** parameter on the server node enforces the load style selected in the Load Options dialog in the JADE development environment (that is, when you click the **Load** toolbar button or select the **Load** command from the Schema menu).

The default value of **true** indicates that the **LoadStyleDefault**, **LoadStyleSecond**, and **LoadStyleThird** parameters specified on the server are ignored and the JADE initialization file on the client node is checked for the load style values. If load style values are not specified in the JADE initialization file on the client node, the standard JADE-defined load style values are used.

Parameter is read when …

The Load Options dialog is opened.

LoadStyleSecond

Value Type String

Default Not specified

Purpose

The optional **LoadStyleSecond** parameter specifies the load style that is displayed second in the **Load Style** combo box in the Load Options dialog in the JADE development environment (that is, you clicked the **Load** toolbar button or selected the **Load** command from the Schema menu) when the **LoadStyleDefault** parameter is set to a valid value.

The valid values that you can define for this parameter are as follows.

- **PRIOR**, which displays *Load into the existing current schema version*
- **LATEST**, which displays *Add new schema, or load as latest schema version*
- **STRUCTURAL**, which displays *Load into latest versioning only structural changes*

Notes If a valid value is not specified for the **LoadStyleDefault** parameter, this parameter is ignored and JADE-defined values are used (that is, *Load into the existing current schema version* is displayed as the second item in the **Load Style** combo box).

If you comment out this parameter or you do not define it with a valid value when the **LoadStyleDefault** parameter is set to a valid value, a second item is not displayed in the combo box.

Parameter is read when …

The Load Options dialog is opened.

LoadStyleThird

Value Type String

Default Not specified

Purpose

The **LoadStyleThird** parameter specifies the load style that is displayed third in the **Load Style** combo box in the Load Options dialog in the JADE development environment (that is, you clicked the **Load** toolbar button or selected the **Load** command from the Schema menu) when the **LoadStyleDefault** parameter is set to a valid value.

The valid values that you can define for this parameter are as follows.

- **PRIOR**, which displays *Load into the existing current schema version*
- **LATEST**, which displays *Add new schema, or load as latest schema version*
- **STRUCTURAL**, which displays *Load into latest versioning only structural changes*

Notes If a valid value is not specified for the **LoadStyleDefault** parameter, this parameter is ignored and JADE-defined values are used (that is, *Load into latest versioning only structural changes* is displayed as the third item in the **Load Style** combo box).

If you comment out this parameter or you do not define it with a valid value when the **LoadStyleDefault** parameter is set to a valid value, a third item is not displayed in the combo box.

Parameter is read when …

The Load Options dialog is opened.

PathFontColors

Value Type String, Real, Boolean, Integer, Integer

Default Correct default values

Purpose

The **PathFontColors** parameter specifies the font attributes and color of the **Path:** label and your fully qualified database path passed by the command line on the splash screen.

The parameter values represent *font-name*, *font-size*, *font-bold*, *color1*, and *color2*. The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

A *font-name* value of **Default** indicates that the default system font is used and that the *font-size* and *font-bold* values are ignored.

The *color1* value applies to the **Path:** label, and the *color2* value applies to the label containing the path to the database. For example, if you set this parameter to **Arial, 12, true, 16777215, 255** and then start the JADE development environment, the **Path:** label is white, the **c:\jade\system** label is red, and both labels use the Arial 12 point bold font.

Parameter is read when …

The client node is next initialized.

PathPos

Value Type Integer, Integer, Integer, Integer

Default Correct default values

Purpose

The **PathPos** parameter specifies the position of the **Path:** label and your fully qualified database path passed by the command line on the start-up screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical screen coordinates of the **Path:** label in pixels, respectively, relative to the top left corner of the window.

The third and fourth *integer* values (indicating the *x2* and *y2* points) are the horizontal and vertical coordinates of the path text in pixels, respectively, relative to the top left corner of the window.

This positions the displayed path label and text, which are displayed in the following form.

```
Path: target-database-path
```

If the positions are not valid or you specify **No**, the path label and text are not displayed.

Parameter is read when …

The client node is next initialized.

SchemaFontColors

Value Type String, Real, Boolean, Integer, Integer

Default Correct default values

Purpose

The **SchemaFontColors** parameter specifies the font attributes and color of the **Schema:** label and your database schema name passed by the command line on the splash screen.

The parameter values represent *font-name*, *font-size*, *font-bold*, *color1*, and *color2*. The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

A *font-name* value of **Default** indicates that the default system font is used and that the *font-size* and *font-bold* values are ignored.

The *color1* value applies to the **Schema:** label, and the *color2* value applies to the label containing the name of the schema. For example, if you set this parameter to **Arial, 12, true, 16777215, 255** and then start the JADE development environment, the **Schema:** label is white, the **JadeSchema** label is red, and both labels use the Arial 12 point bold font.

Parameter is read when …

The client node is next initialized.

SchemaPos

Value Type Integer, Integer, Integer, Integer

Default Correct default values

Purpose

The **SchemaPos** parameter specifies the position of the **Schema:** label and your database schema name passed by the command line on the start-up, or splash, screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates of the **Schema:** label in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) are the horizontal and vertical coordinates of the schema name text in pixels, respectively, relative to the top left corner of the window.

This positions the displayed schema label and text, which are displayed in the following form.

```
Schema: target-schema-name
```

If the positions are not valid or you specify **No**, the schema label and text are not displayed.

Parameter is read when …

The client node is next initialized.

Server

Value Type String

Default SingleUser

Purpose

The **Server** parameter specifies the mode in which the database is to be accessed and is required if you have not specified a **server** argument value in the command line for the **jade.exe** executable program. Valid options for this parameter are **MultiUser**, **SingleUser**, and ***server-URI*** (that is, the server URI string).

For details about using the server Uniform Resource Identifier (URI) string to specify the target database and the client-server transport, see "Format of the Server URI String", in Chapter 2 of the *JADE Installation and Configuration Guide*.

Parameter is read when …

The client node is next initialized.

ServerFontColors

Value Type String, Real, Boolean, Integer, Integer

Default Correct default values

Purpose

The **ServerFontColors** parameter specifies the font attributes and color of the **Server:** label and your server setting on the splash screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

The parameter values represent *font-name*, *font-size*, *font-bold*, *color1*, and *color2*.

A *font-name* value of **Default** indicates that the default system font is used and that the *font-size* and *font-bold* values are ignored.

The *color1* value applies to the **Server:** label, and the *color2* value applies to the label containing the mode in which the database is opened. For example, if you set this parameter to **Arial, 12, true, 16777215, 255** and then start the JADE development environment in single user mode, the **Server:** label is white, the **SingleUser** label is red, and both labels use the Arial 12 point bold font.

Parameter is read when …

The client node is next initialized.

ServerPos

Value Type Integer, Integer, Integer, Integer

Default Correct default values

Purpose

The **ServerPos** parameter specifies the position of the **Server:** label and your server name on the start-up, or splash, screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates of the **Server:** label in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) are the horizontal and vertical coordinates of the server name text in pixels, respectively, relative to the top left corner of the window. This positions the displayed server label and text, which are displayed in the following form.

```
Server: server-name
```

The *server-name* value is obtained from the **Server** parameter (described earlier in this section) or from the command line if it is specified there.

If your specified positions are not valid or you specify **No**, the server label and name are not displayed.

Parameter is read when …

The client node is next initialized.

ShowSplashScreen

Value Type Boolean

Default true

Purpose

The **ShowSplashScreen** parameter specifies that an image is to be displayed on start-up. The default value of **true** specifies that a splash screen is displayed on start-up. If you specify **false**, no splash screen is displayed and the other parameters in the [Jade] section relating to what is displayed when the application starts up do not apply.

The default splash screen option depends on the schema and application being run. There are specific splash screens for the JADE Translator utility, the JADE Monitor, the JADE Schema Inspector utility, and the JADE development environment apart from the default JADE splash screen.

The release note splash screen, which displays the major features of the current release and hyperlinks to further information, is displayed in the JADE development environment:

- The first time an upgraded JADE system is started, regardless of the value of this parameter.
- The first time a new installation of JADE is started.
- Every time a system with no user-defined schemas is started, regardless of the value of this parameter.

Check the **Do not show at startup** check box on the release note splash screen if you want to suppress the screen display the next time the JADE development environment starts up.

Parameter is read when …

The client node is next initialized.

ShowUserInterrupt

Value Type Boolean

Default true

Purpose

The **ShowUserInterrupt** parameter specifies the Jade User Interrupt icon and menu are displayed. Use the **UseSystemTrayIcon** parameter in the [Jade] section to configure whether the interrupt icon is positioned in the system tray.

When this parameter is set to **true** (the default), the Jade User Interrupt icon and menu are displayed.

If you do not want the Jade User Interrupt icon and menu displayed, set this parameter to **false**.

For more details, see "Using the Jade User Interrupt", in Chapter 1 of the *JADE Runtime Application Guide*.

Parameter is read when …

The client node is next initialized.

Note This parameter is ignored when JADE is running in a production mode database.

SplashScreenFile

Value Type String *(file-name)*

Default Internal splash screen

Purpose

The **SplashScreenFile** parameter specifies an image to be displayed on start-up.

If you specify a bitmap file that does not exist, cannot be read, or does not contain a valid graphic image (for example, a bitmap file), the default JADE image is used.

The start-up image is sized to the size of the graphic without any stretching.

The value of **<default>** specifies that JADE uses an internal resource as the splash screen. The default splash screen displayed depends on the schema and application being run.

There are specific splash screens for the JADE Translator utility, the JADE Monitor, the JADE Schema Inspector utility, and the JADE development environment apart from the default JADE splash screen.

Parameter is read when …

The client node is next initialized.

StatusFontColor

Value Type String, Real, Boolean, Integer

Default Correct default values

Purpose

The **StatusFontColor** parameter specifies the font attributes and color of the text that displays the current opening status on the splash screen.

The parameter values represent *font-name*, *font-size*, *font-bold*, and *color*. The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

A *font-name* value of **Default** indicates that the default system font is used and that the *font-size* and *font-bold* values are ignored. For example, if you set this parameter to **Arial, 12, true, 16777215** and you then run your application, the current opening status is displayed in green Arial 12 points.

Parameter is read when …

The client node is next initialized.

StatusPos

Value Type Integer, Integer

Default Correct default values

Purpose

The **StatusPos** parameter specifies the position of the text that displays the current opening status on the start-up, or splash, screen.

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates in pixels, respectively, relative to the top left corner of the window. If the positions are not valid or you specify **No**, the opening status is not displayed.

Parameter is read when …

The client node is next initialized.

UseSystemTrayIcon

Value Type Boolean

Default true

Purpose

The **UseSystemTrayIcon** parameter specifies whether the Jade User Interrupt icon is positioned in the system tray at the right of the Taskbar when the **ShowUserInterrupt** parameter is set to **true** or if the application has no visible forms. (For more details, see "Using the Jade User Interrupt", in Chapter 1 of the *JADE Runtime Application Guide*.)

Set this parameter to **false** if you want the Jade User Interrupt icon is positioned on the Taskbar.

Parameter is read when …

The client node is next initialized.

VersionFontColors

Value Type String, Real, Boolean, Integer, Integer

Default Correct default values

Purpose

The **VersionFontColors** parameter specifies the font attributes and color of the version label and your application version on the splash screen. The value of **<default>** specifies that JADE uses what it recognizes as the correct default values.

The parameter values represent *font-name*, *font-size*, *font-bold*, *color1*, and *color2*. A *font-name* value of **Default** indicates that the default system font is used and that the *font-size* and *font-bold* values are ignored.

The *color1* value applies to the **Release:** label, and the *color2* value applies to the label containing the version of JADE. For example, if you set this parameter to **Arial, 12, true, 16777215, 255** and then start the JADE development environment, the **Release:** label is white, the label containing the version number is red, and both labels use the Arial 12 point bold font.

Parameter is read when …

The client node is next initialized.

VersionPos

Value Type Integer, Integer, Integer, Integer

Default Correct default values

Purpose

The **VersionPos** parameter specifies the position of the version label and your application version on the start-up, or splash, screen. (For JADE, this displays the **Release:** label and the release version.

You can use the **appVersion** property of the **Application** class to specify the application documentation that is displayed in your default About box; for example, the **Version:** label on your customized start-up screen.)

The value of **<default>** specifies that JADE uses what it recognizes as the correct default values. The parameter value is based on 96 dots per inch (that is, 96 dpi). The splash screen positions are scaled if the user has a different dpi setting.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical coordinates of the version label in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) are the horizontal and vertical coordinates of the version text in pixels, respectively, relative to the top left corner of the window.

This positions the displayed version label and text, which are displayed in the following form.

```
Version: version
```

The *version* value is obtained from the **Application** class when the database is opened. If the positions are not valid or you specify **No**, the version label and text are not displayed.

Parameter is read when ...

The client node is next initialized.

JADE Thin Client Sections

The [JadeAppServer], [JadeThinClient], and [*environment-type*] sections of the JADE initialization file contain parameters that control the running of the JADE thin client application server and presentation clients, respectively.

Note The JADE presentation client executable program **jade.exe** uses its own initialization file, located by default in the installation directory of **jade.exe** (that is, the **bin** directory) unless you override this by specifying the **ini** argument in the JADE presentation client command line. This initialization file holds parameters that affect the JADE presentation client local environment.

For details about the parameters you can specify in the [JadeAppServer] and [JadeThinClient] sections for Secure Sockets Layer (SSL) security of JADE thin clients, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Application Server Section [JadeAppServer]

The [JadeAppServer] section can contain the following parameters.

AllowSchemaAndApp<*n*>

Value Type String *(schema-name[, application-name][, time-out])*

Default Not specified

Purpose

When the **EnableAppRestrictions** parameter is set to **true**, only the schema and optional applications specified in the **AllowSchemaAndApp<*n*>** parameters can be executed from presentation clients attached to the application server. The **<*n*>** variable in the parameter name indicates a unique number; for example:

```
AllowSchemaAndApp1 = MySchema, MyApp
AllowSchemaAndApp2 = MyOtherSchema, MyOtherApp
AllowSchemaAndApp3 = AllAppSchema
```

In this example, only the **MyApp** application in the **MySchema** schema, the **MyOtherApp** application in the **MyOtherSchema**, and any application in the **AllAppSchema** can be executed.

If a user attempts to start any other application, the initiation request is rejected with a 14139 error, whose text states: *The execution of that application is not available from the connected Application Server.*

All applications can be executed when the **EnableAppRestrictions** parameter is set to **false** (the default).

When your system has application restrictions (that is, the **EnableAppRestrictions** parameter is set to **true** in the [JadeAppServer] section and you want to use the JADE Monitor or to inspect your schemas, you must specify **JadeMonitorSchema** as a schema that can be executed from presentation clients attached to the application server; for example:

```
AllowSchemaAndApp4 = JadeMonitorSchema
```

For details about the syntax of this parameter when the **EnableAppRestrictions** parameter is set to **true** and you want to specify a connection timeout period for a specific schema and optional application, see the **ConnectionTimeout** parameter, later in this section.

When the **EnableAppRestrictions** parameter is set to **true** and the application being started is not a permitted application defined in the JADE initialization file, the following message is displayed when attempting to start the application from the JADE development environment.

```
Application app-name cannot be started from the connected application server
```

In this message, the *app-name* value is the name of the application that is required to run.

To avoid unexpected presentation client rejections, all application servers that are balancing presentation client connections within the same group must use the same value for the **EnableAppRestrictions** and **AllowSchemaAndApp** parameters.

Parameter is read when …

A presentation client attempts to sign on to an application.

AppServer

Value Type String ([*transport-type*,]*interface*)

Default 0.0.0.0

Purpose

The **AppServer** parameter specifies the TCP/IP communications address (for example, **143.57.055.259**) or the name of application server (for example, **mycomputer1a**). By default, when the application server starts up, it listens for network connections from presentation clients on all network interface cards enabled on the host.

The values for the optional *transport-type* variable 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.

Presentation client connection balancing cannot be enabled if the **ExternalAppServerAndPort** parameter or its value is not specified and the **AppServer** parameter or its value is not specified or the value is the default value of **0.0.0.0**.

If the value of the **AppServer** parameter is a Fully Qualified Domain Name (FQDN), it must uniquely identify the machine on which the application server is running. It must not resolve to multiple addresses that point to other machines (otherwise presentation client connection balancing is defeated).

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

Parameter is read when …

The application server node is next initialized.

Hint

Use this parameter to restrict the connections to one specific network interface card on which the application server listens.

AppServerGroupName

Value Type String

Default Not specified

Purpose

The **AppServerGroupName** parameter specifies the name for a group of application servers that evenly share presentation client connections. When a presentation client makes an initial connection, it could be redirected to another application server within the group to balance the number of connections.

All application servers in the same group must have the same value for this parameter. The value of this parameter is a case-sensitive user-defined string of up to 100 characters.

If you do not specify a value for the **AppServerGroupName** parameter, presentation client connection balancing is not enabled.

Parameter is read when …

The application server node is next initialized.

AppServerPort

Value Type Integer *(application-server-port-number)*

Default Not specified

Purpose

The **AppServerPort** parameter specifies the TCP/IP communications port number of the application server. Presentation clients attempting to connect to the application server must use the same value in the command line for the **AppServerPort** parameter.

Parameter is read when …

The application server node is next initialized.

Hint

This value is used if you do not specify an **AppServerPort** argument value in the application server command line.

You cannot invoke an application server if a value is found in neither the command line nor the [JadeAppServer] section of your JADE initialization file.

AttemptReconnect

Value Type Boolean

Default true

Purpose

The **AttemptReconnect** parameter specifies whether presentation clients attached to the application server attempt to reconnect when a TCP/IP connection failure occurs. By default, this parameter is set to **true** on both the application server and presentation client nodes.

Set this parameter to **false** if you do not want presentation clients to attempt a reconnection following a TCP/IP connection failure.

Parameter is read when …

Each application is initiated.

AutomaticDownLoad

Value Type Boolean

Default true

Purpose

The **AutomaticDownLoad** parameter specifies whether the automatic downloading of JADE software is enabled or disabled. The application server obtains this parameter when it is initiated.

By default, this parameter is set to **true** on both the application server and presentation client nodes, indicating that automatic downloading of JADE software is in effect. Set this parameter to **false** if you do not want the automatic downloading of JADE software to occur.

If any library on a presentation client does not match the **jade.exe** release version when the automatic download feature is enabled (that is, this parameter is set to the default value of **true**), a software upgrade is requested. For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

CacheEntryTimeout

Value Type Integer *(number-of-days)*

Default 30

The **CacheEntryTimeout** parameter specifies the number of days entries are held in the cache file on the application server before being discarded if they have not been used.

This parameter provides control over cache file entries that become orphaned when forms and pictures are removed from the JADE system. By default, a form that has not been created or a picture that has not been displayed within the last 30 days is discarded. The removal process occurs only when the application server is initiated.

Copies of this form or picture entry remain in the cache file of each presentation client until the number of days specified in the **CacheEntryTimeout** parameter in the local [JadeThinClient] section has expired or it is resurrected if the form or picture is again transmitted by logic after it has been removed from the application server cache file.

You can set this parameter to zero (**0**) if you want forms and pictures to remain indefinitely in the cache file on the application server. Alternatively, set this value to the number of days after which a picture or form that has not been used is discarded from the cache file when the application server is initiated.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

ConnectionTimeout

Value Type Integer *(minutes)*

Default 0

Purpose

The **ConnectionTimeout** parameter specifies the maximum number of minutes that the application server waits for activity (messages sent by or received) from any client application attached to the server before the application session times out that client. Notification and timer traffic is ignored when determining if the connection has timed out.

If no activity occurs within the specified time, the application thread for that client terminates and the following message is displayed in a message box on the presentation client.

```
The current Application session timed out
```

The default value of zero (**0**) indicates that there is no timeout.

In addition, if the **EnableAppRestrictions** parameter in the [JadeAppServer] section is set to **true**, you can use the **AllowSchemaAndApp*<n>*** parameter to specify a timeout period for a specific schema and optional application, by using the following parameter syntax.

```
AllowSchemaAndApp<number> = schema-name [, application-name], timeout-value
```

The following example shows setting the timeout value in the **MyApp** application in the **MySchema** schema to 30 minutes.

```
[JadeAppServer]
ConnectionTimeout     = 0
EnableAppRestrictions = true
AllowSchemaAndApp1    = JadeSchema
AllowSchemaAndApp2    = MySchema, MyApp, 30
```

The parameter settings shown in this example allow all **JadeSchema** applications to be run without any timeout and the **MyApp** application in the **MySchema** schema to time out if no activity occurs within 30 minutes.

Note When an application times out (that is, the period specified in the **ConnectionTimeout** parameter elapses without activity), the equivalent of a monitor force-off is performed, closing all forms and signing off without calling any methods.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

DisplayFont

Value Type String, Integer

Default Correct default values

Purpose

The **DisplayFont** parameter specifies the font that is used in the Application Server window to display all presentation clients currently attached to the application server.

The *string* value represents the valid installed font with which presentation clients are listed. The *integer* value represents the point count for that font.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

DownLoadDescription

Value Type String

Default Not specified

Purpose

The **DownLoadDescription** parameter enables you to optionally specify an additional description of the download that is displayed to presentation client users when software is automatically upgraded. This description, providing site-specific information about the upgrade, is displayed in the message box that informs users that a version upgrade must be downloaded.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

DownLoadMaximum

Value Type Integer

Default Not specified

Purpose

The **DownLoadMaximum** parameter controls the number of concurrent downloads of files to presentation clients during an automatic software upgrade that can be in effect for an application server. If the specified limit is reached, the preload detection for any other presentation clients signing on is ignored.

If you do not specify this parameter or you set it to zero (**0**), there is no limit to the number of simultaneous downloads that can occur.

If you set this parameter to a value greater than zero (**0**), the application only allows up to the specified number of simultaneous downloads. If maximum number of concurrent downloads has been reached and another presentation client requires a download, the user of that presentation client will be informed that a software download is required but the application server is currently too busy. The user must try the download process again later.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

DownLoadVersion

Value Type String

Default Not specified

Purpose

The **DownLoadVersion** parameter specifies the current version of additional files and directories for an automatic software upgrade that are located in the **download** subdirectory of the specified **DownLoadDirectory** directory of the [*environment-type*] section of the presentation client environment (that is, for the specific environment identified by the hardware type, vendor, operating system version, and the ANSI or Unicode file type).

The parameter value can be a user-defined string of up to 60 characters. (Any additional characters are ignored in the version comparison.)

The version specified in this parameter is case-insensitive.

Notes As you must change the string specified in this parameter each time new or updated files need to be downloaded, your site should adopt a convention that increments this value in some way so that it generates a totally new value for each release.

The JADE binary file versions are checked automatically, and this parameter has no impact on that automatic detection process.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each presentation client connects to the application server.

EnableAppRestrictions

Value Type Boolean

Default false

Purpose

When the **EnableAppRestrictions** parameter is set to **false** (the default value), all applications can be executed from any presentation client attached to that application server.

Set this parameter to **true** to specify that only the schemas and optional applications specified in the **AllowSchemaAndApp*<n>*** parameters can be executed from presentation clients attached to the application server.

For details about specifying a connection timeout period for a specific schema and optional application when this parameter is set to **true**, see the **ConnectionTimeout** parameter, earlier in this section.

When the **EnableAppRestrictions** parameter is set to **true** and the application being started is not a permitted application defined in the JADE initialization file, the following message is displayed when attempting to start the application from the JADE development environment.

```
Application app-name cannot be started from the connected application server
```

In this message, the *app-name* value is the name of the application that is required to run.

To avoid unexpected presentation client rejections, all application servers that are balancing presentation client connections within the same group must use the same value for the **EnableAppRestrictions** and **AllowSchemaAndApp** parameters.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

ExternalAppServerAndPort

Value Type String *(TcpIp-address* or *host-name*, *port-number)*

Default Not specified

Purpose

You can optionally specify the **ExternalAppServerAndPort** parameter for an application server involved in presentation client connection balancing. If you specify this parameter, the value of the parameter is returned to the presentation client as the *redirect* address, which is used when the address and port on which the application server is listening is hidden from the workstation or device running the presentation client by Network Address Translation (NAT). All presentation clients redirected to this application server must be subject to the same NAT rule; that is, only one override address and port is provided.

You must specify this parameter if the application server is not listening on a specific address. This is usually indicated when the value of the **AppServer** parameter is not specified or is **0.0.0.0**.

You cannot enable presentation client connection balancing if the value of this property is invalid; for example, it has invalid characters or the port number is not specified.

If the value includes a Fully Qualified Domain Name (FQDN), it must uniquely identify the machine on which the application or server is running. It must not resolve to multiple addresses that point to other machines (otherwise presentation client connection balancing is defeated).

Parameter is read when …

The application server node is next initialized.

KeepAliveTime

Value Type Integer *(minutes)*

Default 10

Purpose

The **KeepAliveTime** parameter specifies the number of minutes that each running application checks whether the connection to the presentation client is still valid, by sending a *'keep alive'* message to the client after every period specified by this parameter expires without any activity.

This check is necessary because the presentation client can detach without any notification from the TCP/IP connection (for example, the modem is unplugged) and therefore can tie up an in-use licence. If the connection is still valid, this process acts as a *do nothing* event (but obviously at the cost of a message turnaround). If the connection is no longer valid, the client is terminated and signed off.

By default, the application server performs this check after every ten minutes of inactivity. The minimum value that you can specify is **1** minute.

Note When an application server recognizes that a connection has been lost, it does not know whether the presentation client has also realized that state. As a result, it waits for the number of seconds specified by the **ReconnectWaitTime** parameter, to give the presentation client the chance to reconnect. (A **ReconnectWaitTime** parameter value of zero (**0**) indicates that the **KeepAliveTime** parameter value is used.)

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

NodeName

Value Type String *(service-name)*

Default JadApp-*port-number*

Purpose

The **NodeName** parameter specifies the internal name of the service. When you install an application server as a service and this parameter is defined, the parameter value is assigned to the service. If this parameter is not defined in your JADE initialization file when an application server service is installed, JADE creates the default value, using the **AppServerPort** argument from the **jadapp** or **jadappb** program command line for the *port-number*.

The **NodeName** parameter is the Windows service name as displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Note The installation of an application server service fails if the **NodeName** parameter value is already in use; that is, each active service must have a unique **NodeName** parameter value. If you want to run more than one application server service concurrently, you require a separate JADE initialization file for each service.

NodeNameDescription

Value Type String

Default Jade AppServer *port-number (path)*

Purpose

The **NodeNameDescription** parameter specifies the description given to the application server when it is run as a service. The descriptive name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

When you install an application server as a service and this parameter is defined, the parameter value is assigned to the service.

If this parameter is not defined in your JADE initialization file when an application server service is installed, JADE creates the default value, using the **AppServerPort** argument value from the **jadapp** or **jadappb** program command line for the *port-number* and the **path** argument value from the **jadapp** or **jadappb** program command line for the *path*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Note The installation of an application server service fails if the **NodeNameDescription** parameter value is already in use; that is, each active service must have a unique **NodeNameDescription** parameter value. If you want to run more than one application server service concurrently, you require a separate JADE initialization file for each service.

NodeNameFriendly

Value Type String *(service-name)*

Default Not specified

Purpose

The **NodeNameFriendly** parameter specifies the *friendly* name of the service when you install an application server as a service. The friendly name is displayed in Microsoft Management Console (**services.msc**) and by the Services Control command line utility (**sc.exe**).

If this parameter is not defined in your JADE initialization file when an application server service is installed, JADE uses the value specified for the **NodeName** parameter.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

PictureCacheFile

Value Type String *(file-name)*

Default jade<*tcp-port*>.jpf

Purpose

The **PictureCacheFile** parameter specifies the name of the file in which the application server creates a flat file containing all pictures sent by logic to the presentation client.

This file is written by default to your application server installation (**bin**) directory if you do not specify an absolute path.

The picture cache file enables the presentation client to also cache those pictures, eliminating the need to transmit the pictures on the next execution of that logic. If the file does not exist or the JADE release version change, the file is recreated, causing the picture images to be retransmitted to the presentation clients when required.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

PreDownLoadCount

Value Type Integer

Default Not specified

Purpose

The **PreDownLoadCount** parameter contains the number of users who have pre-downloaded the available JADE upgrade software.

As each presentation client completes a pre-download of the JADE software, the value of this parameter is incremented.

Note Manually set this parameter to zero (**0**) each time the value of the **PreDownLoadVersion** parameter changes.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

PreDownLoadDescription

Value Type String

Default Not specified

Purpose

The **PreDownLoadDescription** parameter enables you to optionally specify an additional description of the pre-download that is displayed to presentation client users. This description is displayed in the message box that informs users that pre-downloaded files are available.

You could use this parameter, for example, to inform users of the date by which the pre-download of files should be done. For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

PreDownLoadVersion

Value Type String

Default Not specified

Purpose

The **PreDownLoadVersion** parameter specifies the current version of pre-download files and directories located in the **predownload** subdirectory of the specified **DownLoadDirectory** directory of the [*environment-type*] section for the presentation client environment (that is, for the specific environment identified by the hardware type, vendor, operating system version, and the ANSI or Unicode file type).

The **PreDownLoadVersion** parameter specifies a user-defined string of up to 60 characters. (Any additional characters are ignored in the version comparison.) The version specified in this parameter is case-insensitive.

Note As you must change the string specified in this parameter to cause a new version of software to be pre-downloaded, your site should adopt a convention that increments this value in some way so that it generates a totally new value for each release.

When the software release for pre-downloaded files is installed on the application server, copy the contents of the pre-download directories to the corresponding download directories on that application server and set the **DownLoadVersion** parameter in the [JadeAppServer] section to the value of the **PreDownLoadVersion** parameter. Each presentation client recognizes that a version change has occurred, determines that it already has the files downloaded, and then installs them.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each presentation client connects to the application server.

PreventFileDownload

Value Type Boolean

Default false

Purpose

The **PreventFileDownload** parameter enables you to prevent the download or predownload of files required to upgrade the JADE software of a presentation client connecting to an application server.

Note The value of this parameter is ignored if the value of the **AutomaticDownLoad** parameter is set to **false** in the [JadeAppServer] or the [JadeThinClient] section of the JADE initialization file, as no download is attempted.

When you set the **PreventFileDownload** parameter to **true**, the presentation client expects to find all of the files required to be installed on the presentation client in the **predownload** directory, the **download** directory, or already installed.

Use this option to force presentation client upgrades to occur locally from the **predownload** directory when the files to be installed have been copied there manually or by a non-JADE process.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each presentation client connects to the application server.

PreventFileDownloadDescription

Value Type String

Default Not specified

Purpose

The **PreventFileDownloadDescription** parameter enables you to specify a message that is displayed in a message box on the presentation client when the **PreventFileDownload** parameter is set to **true** but a required file is not available.

If a message is not specified by setting the **PreventFileDownloadDescription** parameter, the following message is displayed.

```
The JADE Thin Client needs to update one or more files, but the latest version of
these cannot be found and updates from the application server are currently
disabled. Please contact your System Administrator.
```

Note The value of this parameter is ignored if the value of the **AutomaticDownLoad** parameter is set to **false** in the [JadeAppServer] or the [JadeThinClient] section of the JADE initialization file, as no download is attempted.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each presentation client connects to the application server.

ReconnectWaitTime

Value Type Integer *(seconds)*

Default 0

Purpose

The **ReconnectWaitTime** parameter specifies the number of seconds that an application server waits after a connection to a presentation client is lost (detected by the lack of a reply to a *keep alive* message) before terminating the presentation client process. The application server waits for the specified time to give the presentation client a chance to reestablish the connection.

The default value of zero (**0**) indicates that the **ReconnectWaitTime** parameter value is the same as the time specified for the **KeepAliveTime** parameter.

Parameter is read when …

Each time that a connection to a presentation client is lost.

RPCEncryptionEnabled

Value Type Boolean

Default false

Purpose

When the **RPCEncryptionEnabled** parameter is set to **false** (the default value), encryption is not enforced; that is, it is disabled. Set this parameter to **true** to enforce (enable) the defined RPC encryption protocol on all presentation clients attached to the application server.

When enabled, remote connection by presentation clients is prevented if the encryption library or entry-points are missing or invalid on the application server.

The JADE presentation client network connection is not secure when this parameter is set to the default value of **false**. When this parameter is set to **true**, the method to secure the connection is controlled by the corresponding **RPCEncryptionHookDLL** parameter for the application server node.

This parameter must be set to **true** on both the presentation clients and the application server to enforce (enable) the defined RPC encryption protocol.

The settings of the **RPCEncryptionEnabled** and **RPCEncryptionHookDLL** parameters in the [JadeAppServer] section override any setting in the [JadeThinClient] section if the combination of these two parameters are not values listed in the following table, except that the connection fails if the **RPCEncryptionEnabled** parameter is not set to **true** on both the application server and the presentation client when the **RPCEncryptionHookDLL** parameter is set to **SSL_TLS**.

Application Server	Presentation Client
false	**false,** or **true** when **RPCEncryptionHookDLL** = **Internal** or **<*dll-name*>**
true, with **RPCEncryptionHookDLL** = **Internal**	**true**, with **RPCEncryptionHookDLL** = **Internal**
true, with **RPCEncryptionHookDLL** = **SSL_TLS**	**true**, with **RPCEncryptionHookDLL** = **SSL_TLS**
true, with **RPCEncryptionHookDLL** = **<*dll-name*>**	**true**, with **RPCEncryptionHookDLL** = **<*dll-name*>**

For more details about encryption security, see "Enabling JADE Smart Thin Client Security Encryption" and "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

RPCEncryptionHookDLL

Value Type String *(encryption-library-name)*

Default Internal/<*name*>

Purpose

Use the **RPCEncryptionHookDLL** parameter to identify the JADE or user-supplied SSL encryption library. The types of encryption that are available are as follows.

- **RPCEncryptionHookDLL=Internal**, which uses the Windows-supplied 40-bit encryption.
- **RPCEncryptionHookDLL=SSL_TLS**, which enables the SSL security feature. When SSL security is enabled, the parameters whose names start with **SSL** (described in following subsections) are used. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.
- **RPCEncryptionHookDLL=<*name*>**, which calls a user-supplied library with the name specified in the **<*name*>** value. For details, see "Network Message Encryption" under "JADE Security", in Chapter 2 of the *JADE Object Manager Guide*.

Note The RPC encryption library must be thread-safe; that is, it must be able to handle multiple threads calling this library simultaneously.

If RPC encryption is enabled (by setting the **RPCEncryptionEnabled** parameter to **true**), JADE attempts to load the specified library and the required entry points dynamically during initialization.

When RPC encryption is enabled, the connections to the application server are refused and an exception is raised if the dynamic load of the library or any of the required entry-points fails.

For details about the valid combinations of the **RPCEncryptionEnabled** parameter and **RPCEncryptionHookDLL** parameter on both the application server and presentation client nodes, see the JADE application server **RPCEncryptionEnabled** parameter under "Application Server Section [JadeAppServer]", earlier in this chapter.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

RunAsService

Value Type Boolean

Default false

Purpose

The **RunAsService** parameter is set to **true** when the application server is installed as a service.

The default **RunAsService** parameter value of **false** specifies that the application server is *not* installed as a service.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Caution Do not change this parameter value directly. Set it only by selecting the **Run As Service** command from the Options menu of the Application Server window.

For details, see "Running an Application Server as a Service", in Chapter 2 of the *JADE Thin Client Guide*.

Server

Value Type String

Default MultiUser

Purpose

The **Server** parameter specifies a server name, and is required only if you have not specified a **server** argument value in the command line for the **jadapp** or **jadappb** executable program. The other valid options for this parameter is **SingleUser**.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

ServerApplication<*application-number*>

Value Type String, String, Integer, String *(schema-name, application-name[, time[, initialize-method]])*

Default Not specified

Purpose

The **ServerApplication<*application-number*>** parameter specifies the non-GUI server application that is executed when the application server node initializes or at the time specified by the optional *time* variable.

The optional *initialize-method* variable has a maximum length of **250** characters and cannot contain commas. If the method requires a **String** or **Object** parameter, it must be specified as shown in the following example.

```
ServerApplication1 = NoteTest, NotesApp, , init(null)
```

The **initialize** method should not use any Graphical User Interface (GUI) facilities and should not invoke printing services.

Each non-GUI server application specified in a **ServerApplication<*application-number*>** parameter that is to be started when the server node initializes executes the **initialize** method defined for the application or specified in the *initialize-method* variable. When the method has completed execution, presentation clients can connect. The application continues to run until it terminates by itself or the application server stops its execution.

To start the non-GUI server application at a specific time, specify the optional integer ***time*** variable as a four-digit number, representing the start time of the application in a 24-hour time format, with valid values in the range **0000** through **2359**.

For details about the automatic initiation of applications in a JADE system running on a secondary database in an SDE, see "Automatically Starting Server Applications in Secondary Systems", in Chapter 10 of the *JADE Developer's Reference*.

Parameter is read when …

The application server node is next initialized.

Hints

The *application-number* variable of the first server application parameter must start at **1**, with the variable in other parameters, if any, having unique consecutive numbers; that is, **ServerApplication2**, **ServerApplication3**, and so on.

Application server non-GUI applications are displayed in the JADE Monitor window as processes executing on the application server node.

SSLCertificateAuthorityFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLCertificateAuthorityFile** parameter specifies the file name of a Privacy-Enhanced Electronic Mail (PEM)-encoded certificate that is acting as the Certificate Authority. Local operating system file naming conventions apply.

The file is located in the directory specified in the **SSLCertificateAuthorityPaths** parameter. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLCertificateAuthorityPaths

Value Type String *(path-name[; path-name])*

Default Not specified

Purpose

The **SSLCertificateAuthorityPaths** parameter specifies one or more directories that contain the master public certificates for Certificate Authorities (for example, **Verisign**). Local operating system file naming conventions apply.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Hint

Separate each path name with a semicolon character (**;**). The path name can contain a drive letter. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

SSLCertificateFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLCertificateFile** parameter specifies the name of the Privacy-Enhanced Electronic Mail (PEM)-encoded file that contains the certificate for this program. You must specify a full file name, which can include a drive and path that locates the certificate. Local operating system file naming conventions apply. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLCipherNames

Value Type String *(cipher-name[: cipher-name])*

Default Not specified

Purpose

The **SSLCipherNames** parameter specifies one or more ciphers (from those in the following list) that are to be made available to the program.

Specify entries in this parameter to restrict the available ciphers. The connection fails if the application server and thin client do not have at least one cipher in common.

If you do not specify this parameter, the available ciphers can be found in the **OpenSSL** online documentation or **openssl.exe**.

The application server and the presentation client select the strongest common cipher available to both ends of the connection. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Hint

Separate each cipher with a colon character; for example:

```
ECDHE+AESGCM:ECDH+AESGCM:EDH+AESGCM
```

SSLMethodName

Value Type String *(method-name)*

Default <default>

Purpose

The **SSLMethodName** parameter specifies the level of SSL protocol that the application server and presentation clients use to communicate.

The SSL method can be one of the following values, listed in increasing level of functionality and security.

1. TLSv1 (Transport Layer Security)
2. TLSv1.1
3. TLSv1.2

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Caution You should not set this parameter to a value other than the default unless you have very specific requirement; for example, the certificates from the issuing Certificate Authority require a specific SSL communication protocol.

SSLPermitClientRenegotiation

Value Type Boolean

Default true

Purpose

The **SSLPermitClientRenegotiation** parameter specifies whether an application server permits an SSL thin client to renegotiate a connection. By setting the parameter to **true**, you are:

- Complying with Payment Card Industry (PCI) checks regarding overcoming vulnerability to CVE-2009-3555-based attacks
- Protecting against Denial of Service (DoS) attacks

If the parameter is set to **false**, any client-initiated renegotiation causes the network connection to be dropped.

An additional log message has been added to log the build version of OpenSSL and the version of the OpenSSL library DLLs used (that is, **ssleay32.dll** and **libeay32.dll**).

Note To support secure client renegotiations, you require a minimum version of 1.0.2g of the OpenSSL libraries.

Parameter is read when …

The first SSL connection is made.

SSLPrivateKeyFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLPrivateKeyFile** parameter specifies the name of your private key in PEM-encoded format.

If you do not define the **SSLPrivateKeyFile** parameter, the private key is assumed included in the same file as the specified **SSLCertificateFile** parameter for that node. The connection fails if there is no private key.

You must specify a full file name, which can include a drive and path that locates the certificate. Local operating system file naming conventions apply. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLRemoteCertCheck

Value Type Boolean

Default false

Purpose

The **SSLRemoteCertCheck** parameter specifies that the local node, or program, asks the remote program to send its certificate when the value is set to **true**. By default, this parameter is set to **false** in the [JadeAppServer] and the [JadeThinClient] section of the JADE initialization file.

The remote program must have a file defined in the **SSLCertificateFile** parameter for that node.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLRemoteVerify

Value Type Boolean

Default false

Purpose

The **SSLRemoteVerify** parameter specifies that the local program authenticates the remote program when the value is set to **true**.

The remote program must have a file defined in the **SSLCertificateFile** parameter for that node.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLSecurePort

Value Type Integer *(port-number)* | String *(port-name)*

Default 443

Purpose

The **SSLSecurePort** parameter specifies the port number or port name of the TCP/IP connection on which the application server and presentation client communicate.

Note A port name cannot begin with a numeric value.

The SSL protocol is defined to use port **443**.

Unless one of the following conditions applies, leave this parameter at the default value.

- Port **443** is already in use on the required interface by another application.
- The application server or presentation client needs more operating system privileges than it currently has to open connection and listen on the default port.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

SSLVerifyDepth

Value Type Integer

Default 9

Purpose

The **SSLVerifyDepth** parameter specifies how far back in the certificate chain checking for the Certificate Authority signature goes. A certificate is signed by a Certificate Authority (CA) certificate. The CA certificate is signed by a more-trusted CA or it is signed by itself.

If you know the maximum depth of certificate chain, set the parameter to that value, or depth. A large value allows more checking but too small a value may not verify the complete certificate chain. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Note The value of this parameter should not be set to zero (**0**).

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

TraceDirectory

Value Type String *(directory-name)*

Default logs

Purpose

When the **TraceAppServer** parameter in the [JadeThinClient] section is set to **true**, the **TraceDirectory** parameter specifies the directory used for the generated presentation client trace files on the application server.

A separate trace file, output to the JADE **logs** directory by default, is opened for each application that is initiated. The trace file has the following format.

```
ThinClientTrace-<user-name>-<date>-<time>.log
```

The content of the trace file includes the form name for any control and the name of the schema and application being initiated.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Hint

Although a generated trace file on the application server contains the same information as that of a presentation client trace file, the entries in the trace file on the application server provide actual time gap information between each GUI logic statement that is called. You can therefore use them as an indication as to where performance issues might be.

Comparing the file with a presentation client trace file also provides information about network performance issues.

UpgradeRuntimeTo64bit

Value Type Boolean

Default false

Purpose

The **UpgradeRuntimeTo64bit** parameter, when set to the default value of **false**, indicates that the normal thin client upgrade process occurs. Set this parameter to **true** to enable the downloading of 64-bit binaries to clients that are running 32-bit JADE binaries on a 64-bit operating system (WoW64). When this parameter is **true** and a thin client connects to the application server, if the client is running 32-bit binaries on a 64-bit operating system, a download of the JADE 64-bit binaries will occur instead of 32-bit binaries.

Note that for the **UpgradeRuntimeTo64bit** parameter to take effect, the presentation client must first be upgraded with the version of JADE that supports this parameter. The previous thin client versions do not pass their operating system type in the initial handshake that is used to determine whether a download may be required. As a result, to get clients upgraded to 64-bit requires two downloads: one to update to JADE 2016 or 2018, and a second to upgrade to 64-bit.

For more details, see "Automatic Download Issues and Considerations", in Appendix B of the *JADE Thin Client Guide*.

Parameter is read when …

The application server starts up.

UseSystemTrayIcon

Value Type Boolean

Default true

Purpose

The **UseSystemTrayIcon** parameter specifies whether the Application Server window is automatically minimized when the application server starts up and has an icon placed in the system tray at the right of the Taskbar. (For details, see "Placing the Application Server Icon in the System Tray", in Chapter 2 of the *JADE Thin Client Guide*.)

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Hints

Use the **Use system tray** command from the Application Server window Options menu, to toggle the automatic minimizing of the window and the use of the system tray icon.

Set this parameter to **false** if you do not want the window automatically minimized after start up and displayed as an icon in the system tray.

WindowPos

Value Type Integer, Integer, Integer, Integer

Default Not specified

Purpose

The **WindowPos** parameter specifies the position of the Application Server window.

The first and second *integer* values (indicating the *x* and *y* points) are the horizontal and vertical point coordinates of the window in pixels, respectively, relative to the top left corner of the window. The third and fourth *integer* values (indicating the *x2* and *y2* points) indicate the width and height of the window in pixels, respectively.

Parameter is read when …

The application server node is next initialized; for example, when you restart the application server.

Hint

Set the values for this parameter by manipulating the Application Server window, to ensure their accuracy. When you move or resize the display window, the values in this parameter are then updated.

JADE Presentation Client Section [JadeThinClient]

The JADE presentation client executable program **jade.exe** uses its own initialization file, located by default in the installation directory of **jade.exe** (that is, the **bin** directory) unless you override this by specifying the **ini** argument in the JADE presentation client command line. This initialization file contains parameters that affect the JADE presentation client local environment.

The [JadeThinClient] section on a local presentation client can contain the following parameters.

AppServer

Value Type String ([*transport-type*,]*interface*)

Default Not specified

Purpose

The **AppServer** parameter specifies the TCP/IP communications address (for example, **143.57.055.259**) or the name of application server (for example, **wilbur1a**).

The values for the optional *transport-type* variable of the **AppServer** parameter 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 protocols 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 argument values in quote marks if you specify **AppServer="*transport-type*,*interface*"** on the command line.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client.

Hint

This value is used if you do not specify an **AppServer** argument value in the presentation client command line. You cannot invoke a presentation client if a value is found in neither the command line nor the [JadeThinClient] section of your JADE initialization file.

AppServerPort

Value Type Integer *(application-server-port-number)*

Default Not specified

Purpose

The **AppServerPort** parameter specifies the TCP/IP communications port number of the application server. The value of this parameter must be the same as the value specified in the **appServerPort** argument for the **jadapp** application server executable command line.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client.

Hint

This value is used if you do not specify an **appServerPort** argument value in the presentation client command line. You cannot invoke a presentation client if a value is found in neither the command line nor the [JadeThinClient] section of your JADE initialization file.

AskToDownload

Value Type Boolean

Default true

Purpose

The **AskToDownload** parameter controls whether the thin client download process is automatically performed on the presentation client in which this parameter is specified.

By default, users are presented with a message box that states *A new version of the thin client software must be downloaded to complete this request - proceed with the download?* Clicking the **Yes** button downloads the files and clicking the **No** button cancels the file download process.

Set this parameter to **false** if you want the download process initiated without requesting user confirmation.

After installation is complete, if the **PostInstallExe** parameter in the appropriate [*environment-type*] section is specified on the application server, the specified program is initiated.

If a post-installation program is not specified and the **AskToDownload** parameter is set to **false**, the JADE application is automatically re-initiated without requesting user confirmation. Setting the **AskToDownload** parameter to **false** automatically performs the entire download process.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

As the JADE automatic download installation program **jaddinst.exe** automatically uses the name of the presentation client executable defined in the **PostInstallExe** parameter if the value is **jade** or **jade.exe**, you can therefore use the same JADE initialization file parameter to re-initiate the presentation client regardless of whether the standard presentation client initiated the download.

AskToPreDownload

Value Type Boolean

Default true

Purpose

The **AskToPreDownload** parameter controls whether the thin client pre-download process is automatically performed on the presentation client in which this parameter is specified.

By default, users are presented with a message box that states *There are download files available for a future release. Do you wish to perform the download now?* Clicking the **Yes** button downloads the files and clicking the **No** button ignores the download and proceeds with the application execution.

Set this parameter to **false** if you want the pre-download process initiated without requesting user confirmation.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

AttemptReconnect

Value Type Boolean

Default true

Purpose

The **AttemptReconnect** parameter specifies whether the presentation client attempts to reconnect to the application server when a TCP/IP connection failure occurs. By default, this parameter is set to **true** on both the application server and presentation client nodes.

Set this parameter to **false** if you do not want the presentation client to attempt a reconnection following a TCP/IP connection failure.

When you set the **AttemptReconnect** parameter to **false** and a TCP/IP connection failure occurs, the application server generates an exception when an abnormal TCP error is detected and then terminates the presentation client.

If you set the **AttemptReconnect** parameter in the [JadeAppServer] section to **false**, no thin client node attempts to reconnect, regardless of the setting of the parameter in the JADE initialization file on the thin client node. Conversely, if the application server value is set to **true**, the thin client attempts to reconnect only if the **AttemptReconnect** parameter in the [JadeThinClient] section is set to **true** in the JADE initialization file on the thin client node.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

Use the **ReconnectTimeout** parameter to specify the number of seconds that the presentation client attempts to continue to reconnect to the application server.

AutomaticDownLoad

Value Type Boolean

Default true

Purpose

The **AutomaticDownLoad** parameter specifies whether the automatic downloading of JADE software is enabled or disabled. The application server obtains this parameter when it is initiated.

By default, this parameter is set to **true** on both application server and presentation client nodes, indicating that automatic downloading of JADE software is in effect. If this parameter is set to **false** on either node, the automatic downloading of JADE software does not occur.

If any library on a presentation client does not match the **jade.exe** release version when the automatic download feature is enabled (that is, this parameter is set to the default value of **true**), a software upgrade is requested.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

CacheEntryTimeout

Value Type Integer *(number-of-days)*

Default 30

The **CacheEntryTimeout** parameter specifies the number of days entries are held in the cache file on the presentation client before being discarded. This parameter provides control over cache file entries that become orphaned when forms and pictures are removed from the JADE system.

By default, a form that has not been created or a picture that has not been displayed within the last 30 days is discarded.

You can set this parameter to zero (**0**) if you want forms and pictures to remain indefinitely in the cache file on the presentation client.

Alternatively, set this value to the number of days after which a picture or form that has not been used is discarded from the cache file when the presentation client is initiated.

See also the **CacheEntryTimeout** parameter in the [JadeAppServer] section.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

DownLoadDirectory

Value Type String *(file-path)*

Default download1

Purpose

The **DownLoadDirectory** parameter specifies the directory and subdirectories on the presentation client into which all read-only files destined for the binary directory on the presentation client are copied temporarily prior to them being installed during the automatic software upgrade process.

If you do not specify this directory, a directory named **download** is created as a subdirectory of the directory specified by the value of the **ProgramDataDirectory** parameter in the [JadeEnvironment] section and the location of the installation directory on the presentation client.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

DownLoadProgramDataDirectory

Value Type String *(file-path)*

Default downloadprogramdata1

Purpose

The **DownLoadProgramDataDirectory** parameter specifies the directory and subdirectories on the presentation client into which all read-write program data files destined for the system directory on the presentation client are copied temporarily prior to them being installed during the automatic software upgrade process.

If you do not specify this directory, a directory named **downloadprogramdata1** is created as a subdirectory of the directory specified by the value of the **ProgramDataDirectory** parameter in the [JadeEnvironment] section and the location of the installation directory on the presentation client.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

DownLoadVersion

Value Type String

Default Last downloaded version

Purpose

The **DownLoadVersion** parameter specifies the current version of additional files and directories that have been previously downloaded to the directories located in the **download** subdirectory of the specified **DownLoadDirectory** directory of the [*environment-type*] section of the presentation client during the automatic software upgrade process.

This value is automatically maintained when running in JADE thin client mode.

Notes This parameter enables you to use the same binary (**bin**) directory for different database systems if you use unique JADE initialization files in each shortcut or command line on the presentation client. As this assumes that each database system is at the same JADE version level, use of each shortcut or command line causes a download to occur when the JADE version level of the database differs.

The JADE binary file versions are checked automatically, and this parameter has no impact on that automatic detection process.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each presentation client connects to the application server.

FormCacheFile

Value Type String *(file-name)*

Default jade<*modified-appserver-name*><*appserver-port*>.jfm

Purpose

The **FormCacheFile** parameter creates a flat file containing all forms, pictures, and skins sent by logic from the application server. This file eliminates the need to transmit those forms, pictures, and skins on the next execution of the relevant logic.

The *modified-appserver-name* value consists of any characters that are alphabetical, numeric, spaces, underscores, dashes, or dots (dots are replaced by underscores).

The format of the cache file name enables the same JADE binaries to connect to different application servers and retain a presentation client cache file that is specific to an application server and port number. Because it is not known whether the previously used cache file is used or required by the initiation of other applications, it is not deleted.

The cache file is written to the directory specified by the value of the **JadeWorkDirectory** parameter in the [JadeEnvironment] section of the JADE initialization file.

If the file does not exist or the JADE release version changes, it is recreated, causing the form and images to be retransmitted when they are next used. If a cached form is modified, the new form definition is automatically retransmitted when that form is next requested.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

MouseMoveTime

Value Type Integer *(milliseconds)*

Default 200

Purpose

The **MouseMoveTime** parameter enables you to specify the time at which **mouseMove** and **dragOver** events are discarded when moving the mouse within the same window if the time since the execution of the last move event is less than the specified mouse move time, unless the mouse comes to rest. (The mouse comes to rest if no **mouseMove** events are received for the minimum of the specified mouse move time or the default value of 200 milliseconds.)

The first **mouseMove** event received after left-clicking a control in thin client mode immediately generates a **mouseMove** event call to the application server (when that control has logic defined for that event). The **mouseMove** time processing then starts with the next **mouseMove** event that is received.

If the user moves the mouse slowly enough, the same results are achieved as those when running your application in standard (fat) client mode. The value of this parameter is used as the default for each application run on that presentation client.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hints

This style of mouse operation is transparent to most application operations and achieves a significant reduction of events that are sent. (The extent of the reduction varies, according to the type of move performed.)

The JADE Painter running in JADE thin client mode uses a fixed **MouseMoveTime** of 200 milliseconds, regardless of the setting of this parameter.

Use the **Application** class **setMouseMoveTime** method to dynamically set the current mouse move time for presentation clients.

PreDownLoadDirectory

Value Type String (*file-path*)

Default preload1

Purpose

JADE enables you to pre-download a future release of the software required on presentation clients. These pre-downloaded read-only files are held on the presentation client until they take effect on the application server.

Tip Pre-download files to reduce the amount of downloading that is necessary when the release goes live.

The **PreDownLoadDirectory** parameter specifies the directory and subdirectories into which all read-only files are copied ready for installation in the future. This directory and subdirectories contain all of the files that will ultimately be installed in the **jade.exe** binary directory on the presentation client.

If you do not specify this directory, a directory named **preload1** is created as a subdirectory of the directory specified by the value of the **ProgramDataDirectory** parameter in the [JadeEnvironment] section and the location of the installation directory on the presentation client.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

PreDownLoadProgramDataDirectory

Value Type String *(file-path)*

Default preloadprogramdata1

Purpose

JADE enables you to pre-download a future release of the software required on presentation clients. These read-write program data files are held on the presentation client until they take effect on the application server.

Tip Pre-download files to reduce the amount of downloading that is necessary when the release goes live.

The **PreDownLoadProgramDataDirectory** parameter specifies the directory and subdirectories into which all read-write program data files are copied ready for installation in the future.

If you do not specify this directory, a directory named **downloadprogramdata1** is created as a subdirectory of the directory specified by the value of the **ProgramDataDirectory** parameter in the [JadeEnvironment] section of the JADE initialization file and the location of the installation directory on the presentation client.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

PreDownLoadVersion

Value Type String

Default Last predownloaded version

Purpose

The **PreDownLoadVersion** parameter specifies the current version of pre-download files and directories that have been previously downloaded to the directories located in the **predownload** subdirectory of the specified **DownLoadDirectory** directory of the [*environment-type*] section for the presentation client during the automatic software upgrade process.

This value is automatically maintained when running in JADE thin client mode.

When a presentation client connects to the application server, the presentation client and application server versions are compared. If they are different, the pre-download process is initiated.

Note This parameter enables you to use the same binary (**bin**) directory for different database systems if you use unique JADE initialization files in each shortcut or command line on the presentation client.

As this assumes that each database system is at the same JADE version level, use of each shortcut or command line causes a download to occur when the JADE version level of the database differs.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

ReconnectTimeout

Value Type Integer *(seconds)*

Default 40

Purpose

When the **AttemptReconnect** parameter in the [JadeThinClient] section is set to **true**, the **ReconnectTimeout** parameter specifies the number of seconds that the presentation client continues to attempt to reconnect to the application server.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hints

The amount of time actually taken can be more than the specified option because the client keeps attempting to open the TCP/IP connection if the time has not yet expired and an open timeout can take a significant amount of time.

For details about the actions that you can perform when a disconnect event occurs (for example, so that you can programmatically retry the connection or programmatically close it), see "Handling the Loss of Thin Client Connections", in Chapter 2 of the *JADE Thin Client Guide*.

RemovePreDownLoadFiles

Value Type Boolean

Default true

Purpose

On presentation clients, the **RemovePreDownLoadFiles** parameter specifies whether files installed from the pre-downloaded directories are deleted during the software download installation process.

The default value of **true** specifies that the files are deleted during the software download installation process. This takes effect only after the latest **jade.exe** file has been installed on each presentation client.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

Set this parameter to **false** if you want pre-download files retained after their successful installation so that multiple PCs can share the same pre-download directories, removing the need for each PC to download the required software from the application server.

RPCEncryptionEnabled

Value Type Boolean

Default false

Purpose

When the **RPCEncryptionEnabled** parameter is set to **false** (the default value), encryption is not enforced; that is, it is disabled. Set this parameter to **true** to enforce (enable) the defined RPC encryption protocol when the presentation client attaches to the application server and the **RPCEncryptionEnabled** parameter in the [JadeAppServer] section on the application server is set to the appropriate value.

For details about the valid combinations of the **RPCEncryptionEnabled** parameter and **RPCEncryptionHookDLL** parameter on both the application server and presentation client nodes, see the JADE application server **RPCEncryptionEnabled** parameter under "Application Server Section [JadeAppServer]", earlier in this chapter.

When enabled, remote connection by the presentation client is prevented if the encryption library or entry-points are missing or invalid. The JADE thin client network connection is not secure when this parameter is set to the default value of **false**.

When this parameter is set to **true**, the method to secure the connection is controlled by the corresponding **RPCEncryptionHookDLL** parameter for that node.

Parameter is read when …

The presentation client next starts up.

RPCEncryptionHookDLL

Value Type String *(encryption-library-name)*

Default Internal/<*name*>

Purpose

Use the **RPCEncryptionHookDLL** parameter to identify the JADE or user-supplied encryption library on the presentation client. The types of encryption that are available are as follows.

- **RPCEncryptionHookDLL=Internal**, which uses Windows-supplied 40-bit encryption.
- **RPCEncryptionHookDLL=SSL_TLS**, which enables the SSL security feature. When SSL security is enabled, the parameters whose names start with **SSL** (described in following subsections) are used. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.
- **RPCEncryptionHookDLL=<*name*>**, which calls a user-supplied library with the name specified in the **<*name*>** value. For details, see "Network Message Encryption" under "JADE Security", in Chapter 2 of the *JADE Object Manager Guide*.

Note The RPC encryption library must be thread-safe; that is, it must be able to handle multiple threads calling this library simultaneously.

If RPC encryption is enabled (by setting the **RPCEncryptionEnabled** parameter to **true**), JADE attempts to load the specified library and the required entry points dynamically during initialization.

When RPC encryption is enabled, the connection to the application server is refused and an exception is raised if the dynamic load of the library or any of the required entry-points fail.

For details about the valid combinations of the **RPCEncryptionEnabled** parameter and **RPCEncryptionHookDLL** parameter on both the application server and presentation client nodes, see the JADE application server **RPCEncryptionEnabled** parameter under "Application Server Section [JadeAppServer]", earlier in this chapter.

Parameter is read when …

The presentation client next starts up.

SSLCertificateAuthorityFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLCertificateAuthorityFile** parameter specifies the file name of a Privacy-Enhanced Electronic Mail (PEM)-encoded certificate that is acting as the Certificate Authority. Local operating system file naming conventions apply. The file is located in the directory specified in the **SSLCertificateAuthorityPaths** parameter; for example, **\Jade\bin\client.pem**.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLCertificateAuthorityPaths

Value Type String *(path-name[; path-name])*

Default Not specified

Purpose

The **SSLCertificateAuthorityPaths** parameter specifies one or more directories that contain the master public certificates for Certificate Authorities (for example, **Verisign**).

Local operating system file naming conventions apply.

Parameter is read when …

The presentation client next starts up.

Hint

Separate each path name with a semicolon character (**;**). The path name can contain a drive letter. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

SSLCertificateFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLCertificateFile** parameter specifies the name of the Privacy-Enhanced Electronic Mail (PEM)-encoded file that contains the certificate for this program.

You must specify a full file name, which can include a drive and path that locates the certificate. Local operating system file naming conventions apply. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLCipherNames

Value Type String *(cipher-name[: cipher-name])*

Default Not specified

Purpose

The **SSLCipherNames** parameter specifies one or more ciphers (from those in the following list) that are to be made available to the program.

Specify entries in this parameter to restrict the available ciphers. The connection fails if the application server and thin client do not have at least one cipher in common.

If you do not specify this parameter, the available ciphers can be found in the **OpenSSL** online documentation or **openssl.exe**.

The application server and the presentation client select the strongest common cipher available to both ends of the connection. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

Hint

Separate each cipher with a colon character; for example:

```
ECDHE+AESGCM:ECDH+AESGCM:EDH+AESGCM
```

SSLMethodName

Value Type String *(method-name)*

Default <default>

Purpose

The **SSLMethodName** parameter specifies the level of SSL protocol that the application server and presentation client use to communicate.

The SSL method can be one of the following values, listed in increasing level of functionality and security.

1. TLSv1 (Transport Layer Security)
2. TLSv1.1
3. TLSv1.2

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

Caution You should not set this parameter to a value other than the default unless you have very specific requirement; for example, the certificates from the issuing Certificate Authority require a specific SSL communication protocol.

SSLPrivateKeyFile

Value Type String *(file-name)*

Default Not specified

Purpose

The **SSLPrivateKeyFile** parameter specifies the name of your private key in PEM-encoded format.

If you do not define the **SSLPrivateKeyFile** parameter, the private key is assumed to be included in the same file as the specified **SSLCertificateFile** parameter for that node. The connection fails if there is no private key.

You must specify a full file name, which can include a drive and path that locates the certificate. Local operating system file naming conventions apply.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLProxyAuthDetails

Value Type String *(authorization-library-name)*[, String *(function-name)*]

Default Authentication not required

Purpose

When a presentation client connects to an application server and the connection must be made via a proxy server that requires authentication, the **SSLProxyAuthDetails** parameter, which specifies the name and optionally the entry-point of a library file that supplies a user code and password, may be required; for example, **SSLProxyAuthDetails=mySignonLibrary, thinclientSignon**.

This parameter enables a support library to obtain the user name and password required by the proxy server, rather than using the default JADE sign-on dialog. The value of **<default>** indicates that the proxy server user code and password authentication are not required or that they are supplied using the default JADE sign-on dialog.

If the entry-point of the function is not supplied, it is assumed to be **getProxyPassword**.

For more details, see "Getting Presentation Client Sign-On Details", in Chapter 1 of the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

SSLProxyDetect

Value Type Boolean

Default true

Purpose

The **SSLProxyDetect** parameter specifies whether a check is performed to determine if a proxy server has been defined in the Local Area Network (LAN) Settings dialog (accessed by clicking the **LAN Settings** button on the **Connections** sheet of the Windows Internet Properties dialog). If a proxy server is defined, this value is used.

Set this parameter to **false** if you want to disable the registry checks and use only the values of the **SSLProxyHost** and **SSLProxyPort** parameters, if they are defined.

Note The **Use automatic configuration script** and **Do not use proxy server for addresses beginning with:** settings located on the Proxy Setting dialog (accessed from the Local Area Network (LAN) Settings dialog, which is accessed from the Internet Options dialog **Connections** sheet) are ignored.

Parameter is read when ...

The presentation client next starts up.

SSLProxyHost

Value Type *interface*

Default Not specified

Purpose

The **SSLProxyHost** parameter specifies the network name or Internet Protocol (IP) address of the proxy host. If your JADE thin client is behind a firewall, your network administrator may require connections to the Internet to be done through a proxy.

This parameter and the **SSLProxyPort** parameter together allow SSL tunneling through the specified Web proxy host and port.

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 **AppServer** 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?

For more details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when ...

The presentation client next starts up.

SSLProxyPort

Value Type Integer *(port-number)* | String *(port-name)*

Default Not specified

Purpose

The **SSLProxyPort** parameter specifies the port number or port name of the proxy host. If your JADE thin client is behind a firewall, your network administrator may require connections to the Internet to be done through a proxy. This parameter and the **SSLProxyHost** parameter together allow SSL tunneling through the specified Web proxy host and port.

Notes This parameter does not apply to the [JadeAppServer] section of the JADE initialization file.

A port name cannot begin with a numeric value.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The application server next starts up.

SSLRemoteCertCheck

Value Type Boolean

Default false

Purpose

The **SSLRemoteCertCheck** parameter specifies that the local node, or program, asks the remote program to send its certificate when the value is set to **true**.

The remote program must have a file defined in the **SSLCertificateFile** parameter for that node. For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLRemoteVerify

Value Type Boolean

Default false

Purpose

The **SSLRemoteVerify** parameter specifies that the local program authenticates the remote program when the value is set to **true**. The remote program must have a file defined in the **SSLCertificateFile** parameter for that node.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLSecurePort

Value Type Integer *(port-number)* | String *(port-name)*

Default 443

Purpose

The **SSLSecurePort** parameter specifies the port number or port name of the TCP/IP connection on which the application server and presentation client communicate.

Note A port name cannot begin with a numeric value.

The SSL protocol is defined to use port **443**. Unless one of the following conditions applies, leave this parameter at the default value.

- Port **443** is already in use on the required interface by another application.
- The application server or presentation client needs more operating system privileges than it currently has to open connection and listen on the default port.

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

SSLVerifyDepth

Value Type Integer

Default 9

Purpose

The **SSLVerifyDepth** parameter specifies how far back in the certificate chain checking for the Certificate Authority signature goes.

A certificate is signed by a Certificate Authority (CA) certificate. The CA certificate is signed by a more-trusted CA or it is signed by itself.

If you know the maximum depth of certificate chain, set the parameter to that value, or depth. A large value allows more checking but too small a value may not verify the complete certificate chain.

Note The value of this parameter should not be set to zero (**0**).

For details, see "Secure Sockets Layer (SSL) Security", in Chapter 2 of the *JADE Object Manager Guide*.

Parameter is read when …

The presentation client next starts up.

TcpBlockSize

Value Type Integer *(prefix multiplier)*

Default 64K

Purpose

The **TcpBlockSize** parameter on the presentation client node specifies the maximum size of any block of data sent from the presentation client to the application server.

The value specified in the **TcpBlockSize** parameter is the size before any Secure Sockets Layer (SSL) header is added.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

On some Novell networks, the Novell software cannot handle blocks of data sent to TCP/IP connections of more than 512 bytes.

If you are using Novell software for your network and you experience difficulties (for example, disconnections) when running your JADE applications in thin client mode, set the **TcpBlockSize** parameter value to **512** on the presentation client node if you are not using SSL.

If you are using SSL, set the maximum TCP/IP block size to **432** bytes, to allow for up to 80 bytes of Server Side Includes (SSI) header.

Trace

Value Type String

Default None

Purpose

The **Trace** parameter specifies whether all requests made to and from the presentation client are traced.

The default value of **None** for this parameter specifies that tracing is off, or disabled.

JADE provides two levels of tracing, whose parameter values and descriptions are listed in the following table.

Parameter Value	Description
messages	Traces only the commands that caused a message to be sent between the presentation client and the application server
all	Traces all commands passed between the presentation client and the application server

The text **Tracing is ON** is displayed directly under the start-up status line on the presentation client splash screen when the **TraceAppServer** parameter is set to **true** or the **Trace** parameter is set to **all** or **messages**.

For details about trace output, see "Tracing Presentation Client Requests and Messages", in Chapter 3 of the *JADE Thin Client Guide*.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

TraceAppServer

Value Type Boolean

Default false

Purpose

The **TraceAppServer** parameter specifies whether a thin client trace file is generated on the application server for each application initiated by the user.

For presentation client tracing, this parameter is independent of the value of the **Trace** parameter. A warning is displayed on the splash screen of the presentation client if the **TraceAppServer** parameter is set to **true**.

The content of the trace file includes the form name for any control and the name of the schema and application being initiated.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

TraceDirectory

Value Type String *(directory-name)*

Default logs

Purpose

The **TraceDirectory** parameter specifies the directory used for the trace file created for tracing all requests made to and from the presentation client when the value of the **Trace** parameter is set to **messages** or **all**.

A separate trace file, output to the JADE **logs** directory by default, is opened for each application that is initiated. The trace file has the following format.

```
ThinClientTrace-<user-name>-<date>-<time>.log
```

The content of the trace file includes the form name for any control and the name of the schema and application being initiated.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

UseCacheFile

Value Type Boolean

Default true

Purpose

The **UseCacheFile** parameter enables you to specify whether form and picture caching is used in JADE thin client mode. By default, forms and pictures are cached (that is, this parameter is set to **true**). On most connections, performance is significantly improved when form and picture information is cached.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

On a high-speed LAN where caching may not be necessary, set this parameter to **false** if you want to transmit form and picture data in full to the presentation client, instead of creating a cache file.

UseCompression

Value Type Boolean

Default true

The **UseCompression** parameter specifies if the data sent to and from the application server is compressed. Set this parameter to **false** if you do not want transmitted data compressed.

The compression ratio that is achieved is appended to the "Thin Client disconnected and signed off" message written to the **jommsg.log** file of the application server when an application closes.

Parameter is read when …

The presentation client using the same JADE initialization file is next initialized; for example, when you restart the presentation client node.

Hint

The compression process is transparent to both you and the user, and enables you to reduce the size of data that is sent over a slow network connection at the cost of the extra processing CPU required to compress and uncompress the data.

Environment-Specific Section [*environment-type*]

As a JADE application server can handle presentation clients simultaneously, the application server requires different sets of files to be available for downloading to presentation clients. Those files must each reside in a different directory identified by the hardware type, vendor, operating system version, and whether they are ANSI or Unicode files.

The default directory names have the following format.

```
<hardware-type>-<vendor>-<operating-system-version>-ansi|unicode
```

The following is an example of a directory file name.

```
i686-msoft-win32-ansi
```

The JADE initialization file on the application server must contain an [*environment-type*] section that controls the download process for each environment type; for example:

```
[i686-msoft-win32-ansi]
DownLoadDirectory        = <default>
FullJadeInstallDirectory =
PostInstallExe           =
```

For more details, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

The [*environment-type*] section can contain the following parameters.

DownLoadDirectory

Value Type String *(file-path)*

Default Not specified

Purpose

The **DownLoadDirectory** parameter specifies the directory and subdirectories that contain any additional files for the specific environment type that are to be downloaded to the **jade.exe** binary directory on the presentation client when software is automatically upgraded. This must be a valid directory.

If you do not specify this parameter, no additional files are downloaded from this directory and any subdirectories. The downloading of the additional files takes place only after you change the **DownLoadVersion** parameter in the [JadeAppServer] section or a version mismatch occurs.

The directory specified in the **DownLoadDirectory** parameter must have the following two subdirectories.

- **Download**, which contains *all* of the directories and their files to be downloaded to the presentation client
- **Predownload**, which contains *all* of the directories and their files to be pre-downloaded to the presentation client

The **<default>** value for the **DownLoadDirectory** parameter is relative to the installation directory on the application server. If the application server is installed in **\Jade\Windows\Jade**, the default location of the download directory files is a directory named **Download** under a directory with the same name as the environment-type section, as in the following examples.

```
c:\jade\Windows\Jade\i686-msoft-win32-ansi\Download

c:\jade\Windows\Jade\armv4i-msoft-wince50-unicode\Download
```

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

FullJadeInstallDirectory

Value Type String *(file-path)*

Default Not specified

Purpose

The **FullJadeInstallDirectory** parameter specifies the directory where a full set of files for a specific JADE environment is located so that a duplicate set of the thin client files does not need to be copied into a directory specified by the **DownLoadDirectory** parameter.

This parameter is ignored if the presentation client and the application server environments are the same. In that case, the required JADE files are copied from the application server installation directory.

If the **FullJadeInstallDirectory** parameter is not specified, the application server expects to find the required JADE files under the **DownLoadDirectory** directory. If the **FullJadeInstallDirectory** value is the same as the **DownLoadDirectory** value, a warning is logged and the **FullJadeInstallDirectory** parameter is ignored. (If it were not ignored, files would be created in an invalid directory structure on the presentation client.)

In the following example, a value for the **FullJadeInstallDirectory** parameter is specified.

```
[i686-msoft-win32-ansi]
DownLoadDirectory=<default>
FullJadeInstallDirectory=C:\Jade\Windows\Jade
```

The required thin client JADE files are expected to be under the **\Jade\Windows\Jade** directory. They are normally in a **bin** subdirectory. Only client-specific files to be downloaded should be located in the **DownLoadDirectory** directory. If the application server is installed in **\Jade\Windows\Jade**, the default location of the client-specific directories and files to be downloaded is under **\Jade\Windows\Jade\i686-msoft-win32-ansi\Download**.

In the next example, all files for download are located under the **DownLoadDirectory**.

```
[i686-msoft-win32-ansi]
DownLoadDirectory=<default>
FullJadeInstallDirectory=
```

If the application server is installed in **C:\Jade\Windows\Jade**, the directories and files to be downloaded would under **C:\Jade\Windows\Jade\i686-msoft-win32-ansi\Download**. The JADE-specific files would be in the **bin** subdirectory.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

PostInstallExe

Value Type String (*[file-path] file-name*)

Default Not specified

Purpose

After files are installed on a presentation client during the automatic software upgrade process, you may want to perform additional installation procedures.

Use the **PostInstallExe** parameter to specify a user-written program that is run after the installation of all JADE software. If your specified parameter value does not include the full path, the user-defined program must be located in the **jade.exe** binary directory.

As the JADE automatic download installation program **jaddinst.exe** automatically uses the name of the presentation client executable defined in this parameter if the value is **jade** or **jade.exe**, you can therefore use the same JADE initialization file parameter to re-initiate the presentation client regardless of whether the standard presentation client initiated the download.

If you do not specify this parameter (that is, it is empty), the command line initiating the **jade.exe** program that started the process on the presentation client is optionally invoked in response to a message box question to the user and the installation process terminates.

If you specify the **PostInstallExe** parameter, the program that you define is run with the command line that originally started the **jade.exe** process with the current directory set to the directory of **jade.exe** and the JADE installation process terminates.

Note It is the responsibility of your user-defined program to determine what happens after the JADE installation process terminates. The JADE installation process is considered complete, regardless of the user-defined program that you run.

For details about automatically upgrading JADE and user software on presentation clients, see Appendix B, "Upgrading Software on Presentation Clients", in the *JADE Thin Client Guide*.

Parameter is read when …

Each automatic download action is initiated.

JADE Unit Test Runner Section [JadeUnitTestRunnerUI]

The [JadeUnitTestRunnerUI] section of the JADE initialization file contains parameters that control the debug, code coverage, and profiling actions that are set when you run a unit test and the tests that are included.

Notes Set these parameters only by using the appropriate commands in the Unit Test Runner form menus. For details, see Chapter 17, "Using the JADE Testing Framework", of the *JADE Developer's Reference*.

If the Unit Test Runner from was used previously to run tests, these parameters default to the values that were used last time.

All parameters in this section are read when the Unit Test Runner form is next displayed.

The [JadeUnitTestRunnerUI] section can contain the parameters described in the following subsections.

CodeCoverage

Value Type Boolean

Default False

Purpose

The **CodeCoverage** parameter specifies whether code coverage is recorded when running unit tests.

Using code coverage enables you to determine which blocks of code have executed in your unit tests.

DebugOnAssert

Value Type Boolean

Default False

Purpose

The **DebugOnAssert** parameter specifies whether the unit test is paused and the call stack is displayed when a unit test fails on assert.

When the call stack window is closed, the unit tests resume.

DebugOnException

Value Type Boolean

Default False

Purpose

The **DebugOnException** parameter specifies whether the unit test is paused and the call stack is displayed when the unit test encounters an exception.

When the call stack window is closed, the unit tests resume.

DebugOnUnexpectedException

Value Type Boolean

Default False

Purpose

The **DebugOnUnexpectedException** parameter specifies whether the unit test is paused and the call stack is displayed when the unit test encounters an unexpected exception. This enables you to debug exceptions other than any that the test has registered with the **JadeTestCase** class **expectedException** method.

When the call stack window is closed, the unit tests resume.

IncludePassedTests

Value Type Boolean

Default False

Purpose

The **IncludePassedTests** parameter specifies whether the tests that are executed successfully are displayed in the **Results** pane (that is, the results of all tests that have passed, been skipped, or that failed are displayed).

Profile

Value Type Boolean

Default False

Purpose

The **Profile** parameter specifies whether the collection of unit test results is enabled on the Unit Test Runner form when unit tests are run.

Unit test profiling enables you to record actual and total times spent in JADE and external methods in an application.

UIFontSize

Value Type Integer *(points)*

Default 10

Purpose

The **UIFontSize** parameter specifies the size in points of the font used to display information in the Unit Test Runner form. The value can be **10**, **11**, or **12**.

Hint

Select the **Font Bigger** or **Font Smaller** command from the View menu to increase or decrease the size of the font by one (**1**) within the allowed range. The **UIFontSize** parameter is updated with the current value when the Unit Test Runner form is closed.

Parameter is read when …

The Unit Test Runner form is next opened.

Synchronized Database Service Sections

The JADE initialization file [ConnectionParams], [JadeRps], [RpsIgnoreMethodExceptions], [JadeRpsManager], [JadeSDSAdmin], and [SyncDbService] sections of the JADE initialization file contain information used to initialize a Synchronized Database Service environment (SDE).

Connection Parameters Section [ConnectionParams]

The parameters in the [ConnectionParams] section of the JADE initialization file are used by SDE nodes to establish connections with other nodes in a Synchronized Database Environment (SDE).

To specify the connection parameters for a specific server, the **MyName** parameter in the [SyncDbService] section for that server is appended to the [ConnectionParams] section name. For example, the connection parameters for an SDS server with a **MyName** of **Production** are specified in the [ConnectionParams.Production] section. This section is used by the SDS server when it has the primary role and it is also used by each secondary server to obtain the parameters required to connect to the **Production** server when it is the primary.

To allow for takeover, the JADE initialization file parameters for each SDS server (primary or secondary) should contain connection parameters for itself as a primary and connection parameters for any SDE node that could become the primary. An SDS secondary with an RPS subrole need only contain connection parameters for any potential primary database servers. For further details, see "Configuring the Synchronized Database Service", "JADE Initialization File Configuration Examples", and "Creating your First SDS Environment", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*. See also "Two-Level Section Names", earlier in this chapter.

The primary database server operates in a passive or listen mode, accepting (or rejecting) in-bound connections, and makes use of the following parameters in the [ConnectionParams.*primary-name*] section.

- AllowedHost<*ordinal-number*>
- NetworkSpecification<*ordinal-number*>
- RestrictedHostAccess
- SocketBufferSize

The secondary database server initiates an active open to its specified primary server and makes use of the following parameter in the [ConnectionParams.*primary-name*] section.

- ServerNodeSpecifications

AllowedHost<*ordinal-number*>

Value Type String (*host-name* |*IP-address*)

Default Not specified

Purpose

When the **RestrictedHostAccess** parameter is set to **true**, only hosts with names or IP addresses specified in the **AllowedHost<*ordinal-number*>** parameters can connect to the primary database server.

The **<*ordinal-number*>** variable in the parameter name indicates a unique number of the enumerated list, in the following format.

```
AllowedHost1=host-name or IP-address
AllowedHost2=host-name or IP-address
```

```
...
AllowedHost<ordinal-number>=host-name or IP-address
```

Use this parameter to specify a list of all host names or IP addresses that are allowed to connect to the primary database server when the **RestrictedHostAccess** parameter is set to **true**. If any other host attempts to connect to the primary database server, the connection is refused.

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

NetworkSpecification<*specification-number*>

Value Type String (*transport-type*,enabled|disabled,*listener-port*[,*interface*])

Default Not specified

Purpose

The **NetworkSpecification<*specification-number*>** parameter enables a primary database server node with the required connection parameters to listen for incoming connections from secondary database server nodes. The <*specification-number*> part of the parameter name is a sequential number starting from **1**. Multiple network specifications must have contiguous sequence numbers, as shown in the following example.

```
NetworkSpecification1 = TcpIp,disabled,5999
NetworkSpecification2 = TcpIp,enabled,5100,NIC1.hosta.ja.net
NetworkSpecification3 = TcpIpv6,enabled,5200,::1
```

The values in an SDS network specification are interpreted as follows.

1. *transport-type*

   Specifies the transport type to connect to the server node, which is **TcpIP**, **TcpIPv4**, or **TcpIPv6**.

   The *transport-type* literal value is not case-sensitive. **TcpIP** is synonymous with **TcpIPv4**.

2. enabled|disabled

   This value specifies whether the network state is currently enabled or disabled. (By setting this value to **disabled**, you can temporarily disable the network specification without having to renumber the unique identifiers of other **NetworkSpecification**<*specification-number*> parameters in the [ConnectionParams] section.)

3. *listener-port*

   This value specifies the TCP port number or service name used to listen for and accept in-bound connections.

   Select a unique port number in the range **1024** through **65534** that does not conflict with other TCP port number (or TCP service) usages on the machine.

   Note A port name cannot begin with a numeric value.

   Internet Assigned Numbers Authority (IANA) port numbers in the range **0** through **1023** are reserved, those in the range **1024** through **49151** are registered and may have been assigned to specific applications, and those in the range **49152** through **65534** are dynamic, or set aside for private use.

4. [*interface*]

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.

No spaces are allowed in the server network specification.

Parameter is read when …

An SDS primary service is next initialized; that is, you can start and stop the service without stopping the server.

RestrictedHostAccess

Value Type Boolean

Default false

Purpose

When the **RestrictedHostAccess** parameter is set to **true**, the primary database server refuses connections from any host that is not included in the enumerated list specified in the **AllowedHost** parameters. When this parameter is set to **false**, the primary database server accepts connections from any host.

Parameter is read when …

The secondary to primary connection is established; for example, you can disable the connection, change the parameter value on the primary, and then reconnect.

ServerNodeSpecifications

Value Type String (*transport-type*,*remote-host*,*remote-port*[,*local-interface*[,*local-port*]])

Default Not specified

Purpose

The **ServerNodeSpecifications** parameter defines the network connection parameters used by a secondary database server node to connect to this primary database server node across a network.

The values in an SDS server node specification are interpreted as follows.

1. *transport-type*

   This value specifies the transport type to connect to the server node, which is **TcpIP**, **TcpIPv4**, **TcpIPv6**, or **TcpIPAny**. The *transport-type* literal value is case-insensitive.

   The **TcpIP** value is a synonym for **TcpIPAny**, which supports IP version 6 or IP version 4 connections. **TcpIPv4** provides IP version 4 connections only. **TcpIPv6** provides IP version 6 connections only.

   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.

2. *remote-host*

   You can specify the *remote-host* 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. If you specify a host name, all DNS-provided addresses will be attempted.

3. *remote-port*

This value specifies a valid port number or service name to connect to on the primary database server host. This value must match the port number of an enabled listener network on the primary server.

Note A port name cannot begin with a numeric value.

4. [*local-interface*]

   You can specify the *local-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. If you specify a host name, all DNS-provided addresses will be attempted.

5. [*local-port*]

   This optional value specifies the port number or service name.

Note A local port name cannot begin with a numeric value.

If you do not want to specify a local interface but you want to specify a local port, you must specify an empty local interface field, as shown in the following example.

```
ServerNodeSpecifications=TcpIp,dolphin.sitea.ja.net,5016,,5099
```

No spaces are allowed in the server node specifications.

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

SocketBufferSize

Value Type Integer

Default 128K bytes

Purpose

The **SocketBufferSize** parameter enables you to customize TCP/IP buffers to match the link characteristics. The minimum value is **8**K bytes and the maximum value is **16**M bytes. You can also set this parameter to zero (**0**), in which case the operating system default value is used (for example, Windows 7 has a default value of **8**K bytes).

Parameter is read when …

The secondary to primary connection is established and when the primary accepts a connection from a secondary.

Relational Population Service (RPS) Node Section [JadeRps]

The [JadeRps] section of the JADE initialization file contains the following parameters, which store information for the Relational Database Management System (RDBMS) database when the **DatabaseSubrole** parameter in the [SyncDbService] section is set to **RelationalRole**.

Note Most parameters in this section are created when you configure the RPS node using the RPS Manager application on the RPS server node. Set or update these parameter values only by using the appropriate controls on the Configure RPS Node dialog of the RPS Manager application.

The values are read by the RPS Manager application when the Configure RPS Node dialog on the RPS node is invoked.

For details about configuring RPS nodes, see "Configuring your RPS Node", in Chapter 2 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

AutoBulkLoadHistoricalTables

Value Type Boolean

Default false

Purpose

The **AutoBulkLoadHistoricalTables** parameter specifies whether automatic bulk loads on the RPS node should include data in historical tables.

Parameter is read when …

The extract script is created.

Applicable to database role or subrole…

RPS node only.

AutoExtractOnPrimary<*n*>

Value Type String, String, String, String, String, Boolean, String, String
(schema, mapping, auto-script-path, rdb-path, rdb-name, true|false*, execution-location, server-name, extract-workers)*

Default Not specified

Purpose

The **AutoExtractOnPrimary<*n*>** parameter on the primary SDS node is necessary only when using the **Working Set** database replication mode. It specifies how data and RPS mapping changes are extracted when a reorganization takes place on a primary SDS node that is supporting an RPS database operating in **Working Set** database replication mode.

The **<*n*>** variable in the parameter name indicates a unique number; for example:

```
AutoExtractOnPrimary1 = ErewhonInvestmentsModelSchema,
                        DocumentationExample, \rps, \rps, ErewhonTables, false,
                        ClientExecution, PrimaryServer

AutoExtractOnPrimary2 = LibraryExampleSchema, LibraryExample, 2 \rps, \rps,
                        LibraryTables, true, ServerExecution
```

Use this parameter for the automatic extract and load activities of a reorganization, specifying a comma-separated string representing the following values.

Value	Description
schema	Schema for which data is extracted.
mapping	RPS mapping for which data is extracted.
auto-script-path	Location of the created load script and extract files, from the viewpoint of the primary SDS node. This location must be visible to the RPS node and to the RDBMS, and it must be the same location as that specified in the **AutoScriptPath** parameter on the RPS node.
rdb-path	Location of the *auto-script-path* directory value from the viewpoint of the RDBMS. This location is used in the load script that is applied to the RDBMS.
rdb-name	Name of the RDBMS database. This name is used in the load script that is applied to the RDBMS.
true|false	Whether automatic bulk extracts on RPS nodes are to include data in historical tables (defaults to **false**).
execution-location	**ClientExecution** or **ServerExecution**.
server-name	Name of server (if *execution-location* was **ClientExecution**).
extract-workers	Number of extract workers to use.

When the schema for which these values are defined is reorganized, the alter script for the RPS mapping (if any) is generated during the transition phase on the primary SDS node.

If the alter script exists and an extract is required to reload data, the data is extracted to a subdirectory of the location specified in the associated path values of this parameter. The subdirectory name includes the mapping instance, to avoid overwriting files with a subsequent reorganization before the **Datapump** application loads them. It is your responsibility to delete these files when they are no longer required.

Note This is the only parameter in this section that is valid on the primary SDS node. It is not created or updated by the RPS Manager application.

Parameter is read when …

A reorganization occurs on the SDS primary node.

Applicable to database role or subrole…

Primary.

AutoRestartDelay

Value Type Integer

Default 10

Purpose

The **AutoRestartDelay** parameter specifies the number of seconds between attempts to restart the **Datapump** application when the **AutoRestartOnError** parameter is set to **true**.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

AutoRestartOnError

Value Type Boolean

Default false

Purpose

The **AutoRestartOnError** parameter specifies whether the **Datapump** application will be automatically restarted after a connection or timeout error.

The following ODBC error states define connection or timeout errors that trigger the automatic restart (if the value of the **AutoRestartOnError** parameter is **true**).

ODBC Error Number	Description
08001	Unable to establish connection
08004	Server rejected the connection
08007	Connection failure during transaction
08S01	Communication link failure
HYT00	Timeout expired
HYT01	Connection timeout expired.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

AutoRestartRetryLimit

Value Type Integer

Default 10

Purpose

The **AutoRestartRetryLimit** parameter specifies the number of attempts that are made to restart the **Datapump** application when the **AutoRestartOnError** parameter is set to **true**. The number of seconds between restart attempts is specified by the **AutoRestartDelay** parameter.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

AutoRunReorgDropScript

Value Type Boolean

Default false

Purpose

When the **AutoRunReorgDropScript** parameter is set to **true** on the RPS node, the following actions take place when the **Datapump** application is automatically restarted after a reorganization on the RPS node that requires one or more relational tables to be dropped and reloaded.

1. Load scripts generated from the database reorganization are applied to the relational database.
2. Data is extracted for the new or modified table definitions. (This applies only to **Full** or **Mapped Extent** mode; not to the **Working Set** mode.)
3. Extracted data is loaded into the relational database.
4. The **Datapump** application resumes normal incremental update of the relational database.

For an RPS node operating in **Working Set** mode, data is not extracted but would be available for loading if it had been extracted on the SDS primary database. For details about extracting data on the primary node, see the **AutoExtractOnPrimary<*n*>** parameter, earlier in this section.

If this parameter is set to **true**, the **AutoStartDataPump** parameter must also be set to **true**. If this parameter is set to the default value of **false**, it is the responsibility of the RDBMS administrator to apply the required changes and then restart the **Datapump** application.

Parameter is read when …

A reorganization occurs on the RPS node.

Applicable to database role or subrole…

RPS node only.

AutoScriptPath

Value Type String *(relational-database-path)*

Default Not specified

Purpose

The **AutoScriptPath** parameter contains the output location for the reorganization scripts, load scripts, format files, and extracted data files.

For **Working Set** nodes, the location must match the location specified on the primary node.

All files are extracted to a subdirectory of the location specified in the **AutoScriptPath** parameter. The subdirectory name includes the mapping instance. It is your responsibility to delete these files when they are no longer required.

Parameter is read when …

A reorganization occurs on the RPS node.

Applicable to database role or subrole…

RPS node only.

AutoStartDataPump

Value Type Boolean

Default false

Purpose

The **AutoStartDataPump** parameter specifies whether the **Datapump** application is started automatically when the RPS node is initialized, and restarted when a reorganization occurs on the RPS node.

Parameter is read when …

The RPS node is started or a reorganization occurs on the RPS node.

Applicable to database role or subrole…

RPS node only.

BulkLoadCodePage

Value Type String

Default None

Purpose

The **BulkLoadCodePage** parameter specifies the code page to be used when loading the extracted data.

Set the value of this parameter to **RAW**, **ACP**, **OEM**, or the code page number. No validation of the parameter value is done before it is passed to **BULK INSERT** or **bcp**.

If the parameter is specified, it is used by **BULK INSERT** or **bcp**, as follows.

- The **CODEPAGE** option specifying the value of the **BulkLoadCodePage** parameter is added to the BULK INSERT syntax.

```
BULK INSERT [<table>] FROM 'tblfile' WITH (FORMATFILE='fmtfile', TABLOCK,
CODEPAGE='<option>')
```

- The **-C** option specifying the value of the **BulkLoadCodePage** parameter is added to the **bcp** command.

Parameter is read when …

The extract script is created.

Applicable to database role or subrole…

RPS node only.

CreateExceptionHandling

Value Type String

Default Determined by RPS mapping

Purpose

The **CreateExceptionHandling** parameter specifies the action that is taken when an exception occurs during an RDBMS create operation.

Set this parameter to override the value of the **rpsExceptionCreate** property of the RPS mapping.

The values for the **CreateExceptionHandling** parameter are listed in the following table.

Value	Description
<default>	Action taken is determined by the **rpsExceptionCreate** property of the RPS mapping.
Halt	Create operation is aborted. Database tracking is stopped.
Alternative Action	Create operation is attempted as an update. Database tracking is not stopped.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

ConnectionTimeout

Value Type Integer *(seconds)*

Default 15

Purpose

The **ConnectionTimeout** parameter specifies the number of seconds that the RPS **Datapump** application waits to establish a connection to the SQL Server before a timeout occurs.

If the value is zero (**0**), the **Datapump** application waits indefinitely for a connection.

Note The **ConnectionTimeout** parameter is not created or updated by the RPS Manager application.

Parameter is read when one of the following occurs…

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

DataPumpApplication

Value Type String, String (*schema-name,application-name*)

Default *schema-where-RPS-mapping-is defined*,JadeRpsDataPump

Purpose

The **DataPumpApplication** parameter specifies a non-GUI application that is to run as the RPS **Datapump** application; for example, **DataPumpApplication=ErewhonInvestmentsViewSchema,ModDataPumper**.

If the value of the **DataPumpApplication** parameter is **<default>**, the **JadeRpsDataPump**application is run from the schema in which the RPS mapping is defined.

Parameter is read when one of the following occurs…

- An application executes the **rpsDataPumpInitialize** method defined in the **Application** class
- A reorganization occurs
- The **Datapump** application is started from the RPS Manager application

Applicable to database role or subrole…

RPS node only.

DatapumpErrorFileDirectory

Value Type String *(physical-directory-path)*

Default Not specified

Purpose

The **DataPumpApplication** parameter specifies the location of the error file created when the **Datapump** application on an RPS node terminates abnormally.

The *location* value is the location in which the .**err** files will be created. If this parameter does not exist in the initialization file, no .**err** files are created. The *location* value follows the same rules as those of the **LogDirectory** parameter in the [JadeLog] section of the JADE initialization file.

If the **Datapump** application fails with an error (that is, connection errors; table mismatch errors; row creation, update, or delete errors; table inconsistency; extract or load errors; or other exceptions), the file ***location*/rps_*unique-id*.err** is created.

The file contains available information about the error, including any ODBC errors reported, the exception stack, or both ODBC errors and the exception stack.

Parameter is read when …

The node is initialized.

Applicable to database role or subrole…

RPS node only.

DeleteExceptionHandling

Value Type String

Default Determined by RPS mapping

Purpose

The **DeleteExceptionHandling** parameter specifies the action that is taken when an exception occurs during an RDBMS delete operation. Set this parameter to override the **rpsExceptionDelete** property of the RPS mapping.

The values for the **DeleteExceptionHandling** parameter are listed in the following table.

Value	Description
<default>	Action taken is determined by the **rpsExceptionDelete** property of the RPS mapping.
Halt	Delete operation is aborted. Database tracking is stopped.
Alternative Action	Delete operation errors are ignored. Database tracking is not stopped.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

DropHistoricalTableOnAddExisting

Value Type Boolean

Default true

Purpose

The **DropHistoricalTableOnAddExisting** parameter controls the behavior when you add an existing property or method to a historical table in an RPS mapping. The default value of **true** specifies that the historical table is dropped if an existing property or method is added.

If you want to retain a historical table (rather than drop the table) when an existing property or method is added, set the value of this parameter to **false**. When the alter table script is created, **ALTER TABLE <> ADD** is then used to modify the table. Any existing rows will have a column value of **NULL**.

Parameter is read when …

The alter table script is created.

Applicable to database role or subrole…

RPS node only.

DumpOnReplicationException

Value Type Boolean

Default false

Purpose

The **DumpOnReplicationException** parameter specifies whether the automatic process dump caused by the **Datapump** application encountering an exception when replicating a transaction is to be allowed or suppressed.

When the **Datapump** application encounters an exception replicating the effects of a transaction, a process dump is taken and recorded in the **jommsg.log** file, as follows:

```
RPS: **** DataPump Exception: Invoking diagnostic process dump *****
JomLog: >>> Process Dump requested <<<
JomLog: >>> Dumping Process memory and O/S handles ....
JomLog: >>> Process Dump complete, dump file:   <dump file name>
```

Suppress the automatic process dump by setting the **DumpOnReplicationException** parameter to **false**.

Note The **DumpOnReplicationException** parameter is not created or updated by the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

ExtractBufferSize

Value Type Integer

Default 1M

Purpose

The **ExtractBufferSize** parameter specifies the buffer size that is allocated for each concurrent file being written when extracting RPS files.

The minimum value is **8K**.

Parameter is read when …

The node is initialized.

Applicable to database role or subrole…

RPS node only.

ExtractWorkers

Value Type Integer

Default 1

Purpose

The **ExtractWorkers** parameter specifies the number of workers to use if you want to initiate multiple data extract worker processes on automatic extracts.

The default value of **1** indicates that a single data extract worker process is used.

Set this parameter only by using the Configure RPS Node dialog or the Data Extract dialog in the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

LoadExecutionLocation

Value Type String

Default ServerExecution

Purpose

The **LoadExecutionLocation** parameter specifies the location to be used for loading data that has been automatically extracted.

The values for the **LoadExecutionLocation** parameter are listed in the following table.

Value	Description
<default>	ServerExecution.
ServerExecution	The RPS node uses BULK INSERT on the RDBMS server to load data into the RDB. The RPS node and the RDBMS server must have shared access to a disk directory.
ClientExecution	The RPS node uses the **bcp** program to load the data into the RDB. This value is used if the RPS node is executing on a different machine to the RDBMS server and sharing disk is not desired.

Note Data loads are faster with the **LoadExecutionLocation** parameter set to the default value of **ServerExecution**.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

LoggingOption

Value Type String

Default Determined by RPS mapping

Purpose

The **LoggingOption** parameter specifies whether exception information for create, update, and delete statements is recorded in a table in the relational database in addition to being recorded in the **jommsg.log** file. Set this parameter to override the **rpsLoggingOptions** property of the RPS mapping.

The values for the **LoggingOption** parameter are listed in the following table.

Value	Description
<default>	Action taken is determined by the **rpsLoggingOptions** property of the RPS mapping.
Default	Exception information recorded in **jommsg.log** file.
Rpstable	Exception information recorded in **jommsg.log** file and in a table in the RDBMS.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

LoginTimeout

Value Type Integer *(seconds)*

Default 15

Purpose

The **LoginTimeout** parameter specifies the number of seconds that the RPS **Datapump** application waits to log-in to the SQL Server before a timeout occurs. If the value is zero (**0**), the **Datapump** application waits indefinitely for the log-in to complete.

Note The **LoginTimeout** parameter is not created or updated by the RPS Manager application.

Parameter is read when one of the following occurs…

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

LogNullColumnOnException

Value Type String *(file-name)*

Default Not specified

Purpose

The **LogNullColumnOnException** parameter specifies the file in which exception information is logged when an exception occurs in a column-mapping method or the mapping method for a virtual property that is not handled by an exception handler armed from within the method.

If the value of the **NullColumnOnException** parameter is **none**, the **LogNullColumnOnException** parameter is ignored.

If a file name is not specified, which is the default case, no exception information is logged for an unhandled exception in a column-mapping method or the mapping method of a virtual property.

If the full path of the file is not specified, it is located in the directory specified by the value of the **LogDirectory** parameter in the [JadeLog] section.

Note The **LogNullColumnOnException** parameter is not created or updated by the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

NullColumnOnException

Value Type String

Default None

Purpose

The **NullColumnOnException** parameter specifies what happens when an exception occurs in a column-mapping method, the mapping method for a virtual property, or blob, slob, or slobutf8 mapping method and the exception is not handled by an exception handler armed from within the method.

The column is set to a null value and the **Datapump** application continues or the transaction is aborted and the **Datapump** application is stopped.

Note Arm a local exception handler in a column-mapping method, the mapping method for a virtual property, or blob, slob, or slobutf8 mapping method for any exception that is expected to occur. Global exception handlers are not called for exceptions within such a method.

The values that you can specify for the **NullColumnOnException** parameter and the consequences of those settings are listed in the following table.

Value	Column is Set to a Null Value and the Datapump Application Continues
all	Yes

Value	Column is Set to a Null Value and the Datapump Application Continues
selected	Yes (if the column-mapping method, the mapping method for a virtual property, or the blob, slob, or slobutf8 mapping method is listed in the [RpsIgnoreMethodExceptions] section) or No (if the method is not listed in the section)
none	No

If the **Datapump** application is stopped, default exception information is logged. You can:

- Change the JADE initialization file so that the column value is set to null for this method and restart the **Datapump** application
- Fix the problem that caused the exception and recreate the RPS database

When an exception occurs and a column is set to a **null** value, the exception is logged to the file specified by the **LogNullColumnOnException** parameter.

If the **LogNullColumnOnException** parameter has a **null** value, no exception information is logged.

Note The **NullColumnOnException** parameter is not created or updated by the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

QueryTimeout

Value Type Integer *(seconds)*

Default 60

Purpose

The **QueryTimeout** parameter specifies the number of seconds that the RPS **Datapump** application on the RPS node waits for a query involved in replicating data to be executed on the SQL Server database before a timeout occurs.

If the value is set to zero (**0**), the **Datapump** application waits indefinitely for the query to complete. If the value is set to **-1**, no call is made to the SQL Server database to explicitly set a timeout value.

Note The **QueryTimeout** parameter is not created or updated by the RPS Manager application.

Parameter is read when one of the following occurs…

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

RDBLogin

Value Type Boolean

Default false

Purpose

The **RDBLogin** parameter specifies whether a connection dialog is displayed when users attempt to connect to the relational database on the RPS node.

The default value of **false** indicates that an empty string is used for the user name and password. Set the value to **true** if you want a connection dialog displayed.

Parameter is read when …

A connection is made to the RDBMS database.

Applicable to database role or subrole…

RPS node only.

Note This parameter must be set to **false** if the value of the **AutoStartDataPump** parameter is **true**.

RDBName

Value Type String *(relational-database-name)*

Default Determined by RPS mapping

Purpose

The **RDBName** parameter contains the name of the target relational database. If the parameter is set to the value **<default>**, the name of the target relational database is always obtained from the value defined in the RPS mapping.

Parameter is read when …

A reorganization occurs on the RPS node.

Applicable to database role or subrole…

RPS node only.

RDBPath

Value Type String *(relational-database-name)*

Default Not specified

Purpose

The **RDBPath** parameter contains the location of the **AutoScriptPath** from the viewpoint of the relational database. This path is included in the load script that is executed on the relational database.

The script is created on the RPS node if the database replication mode is **Mapped Extent** or **Full**. The parameter is ignored if the database replication mode is **Working Set**.

Parameter is read when …

A reorganization occurs on the RPS node.

Applicable to database role or subrole…

RPS node only.

RDBServerName

Value Type String *(relational-database-name)*

Default Not specified

Purpose

The **RDBServerName** parameter specifies the name of the RDB server. When using SQL Server, this is the SQL Server instance name.

This value is used when the value of the **LoadExecutionLocation** parameter is **ClientExecution** or when the **sqlcmd** utility is to be used to execute SQL scripts.

Parameter is read when …

A connection is made to the RDBMS database.

Applicable to database role or subrole…

RPS node only.

RPSConnectionStr

Value Type String *(connection-string)*

Default Determined by RPS mapping

Purpose

The **RPSConnectionStr** parameter contains the ODBC connection string for connection to the relational database. If the parameter is set to the value **<default>**, the ODBC connection string is always obtained from the value defined in the RPS mapping.

Parameter is read when …

A connection is made to the RDBMS database.

Applicable to database role or subrole…

RPS node only.

SetSQLDateForJadeTime

Value Type String

Default Current date

Purpose

The **SetSQLDateForJadeTime** parameter determines the date portion of the value that is output to the RDBMS for a mapped **Time** property or method that returns a value. This applies only to an RDBMS such as SQL Server that does not have a Time data type.

On SQL Server, a JADE **Time** value is mapped to a **DATETIME** value. The value of the date portion of the **DATETIME** value depends on the setting for the **SetSQLDateForJadeTime** parameter, as shown in the following table.

Value	Result
<default>	The date portion of the **DATETIME** value set to the current date.
<basedate>	The date portion of the **DATETIME** value set to the RDBMS base date. For SQL Server, the base date is 1/01/1900.

Note The **SetSQLDateForJadeTime** parameter is not created or updated by the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only, or a primary SDS node when extracting RPS data.

SetZeroTimeStamp

Value Type String

Default Current date

Purpose

The **SetZeroTimeStamp** parameter determines the date portion of the value that is output to the RDBMS for a mapped property or method that returns a zero **TimeStamp** value (**date=0** and **time=0**).

On SQL Server, a JADE **TimeStamp** value is mapped to a **DATETIME** value. If the JADE **TimeStamp** value is zero, the value of the date portion of the **DATETIME** value depends on the setting for the **SetZeroTimeStamp** parameter as shown in the following table.

Value	Result
<default>	The date portion of the **DATETIME** value set to the current date.
<basedate>	The date portion of the **DATETIME** value set to the RDBMS base date. For SQL Server, the base date is 1/01/1900.
<null>	The column is set to null.

Note The **SetZeroTimeStamp** parameter is not created or updated by the RPS Manager application.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only, or a primary SDS node when extracting RPS data.

UpdateExceptionHandling

Value Type String

Default Determined by RPS mapping

Purpose

The **UpdateExceptionHandling** parameter specifies the action taken when an exception occurs during an RDBMS update operation. Set this parameter to override the **rpsExceptionUpdate** property of the RPS mapping.

The values for the **UpdateExceptionHandling** parameter are listed in the following table.

Value	Description
<default>	Action taken is determined by the **rpsExceptionUpdate** property of the RPS mapping.
Halt	Update operation is aborted. Database tracking is stopped.
Alternative Action	Update operation attempted as an insert. Database tracking is not stopped.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

UseSqlCmd

Value Type Boolean

Default true

Purpose

The **UseSqlCmd** parameter specifies whether RPS uses the **sqlcmd** utility to execute SQL scripts, including alter scripts and server bulk load scripts. The **sqlcmd** utility is a Microsoft utility available for SQL Server. For more details, see your Microsoft documentation.

Parameter is read when …

The script is executed.

Applicable to database role or subrole…

RPS node only.

UseTranInfoTable

Value Type Boolean

Default false

Purpose

The **UseTranInfoTable** parameter specifies whether a table containing transaction information is created and used. For more details about this optional transaction table, see "JADE_TRAN_INFO Table", in Chapter 2 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

If this parameter is defined and it is set to **true**:

- The table is created, if it does not exist
- A row is added to the table for every transaction to record id, start time, and end time
- The row is committed to the relational database as part of the transaction
- Rows are never deleted from the table by the JADE RPS node

It is your responsibility to delete the rows from the table when they are no longer required.

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

RPS Column-Mapping Method Exceptions Section [RpsIgnoreMethodExceptions]

The [RpsIgnoreMethodExceptions] section of the JADE initialization file provides additional information when the value of the **NullColumnOnException** parameter in the [JadeRps] section is **selected**. The [RpsIgnoreMethodExceptions] section is ignored for other values of the **NullColumnOnException** parameter.

The section contains parameters that list the column-mapping methods or virtual properties whose corresponding column is to be assigned a null value if an exception occurs in the method and the exception is not handled by a handler armed within the method. Apart from providing a null value for the column, the **Datapump** application essentially ignores the exception and continues to run. However, if an unhandled exception occurs in a column-mapping method or the mapping method for a virtual property not listed in this section, the transaction is aborted and the **Datapump** application terminates.

Method<*n*>

Value Type String *(class-name::method-name*, *class-name::virtual-property-name*, or *class-name::mapping-method-name*)

Default Not specified

Purpose

When the value of the **NullColumnOnException** parameter in the [JadeRps] section is **selected**, only method, virtual property, or mapping method names specified in the **Method<*n*>** parameters allow unhandled exceptions to occur without the **Datapump** application terminating. The column of the RDBMS corresponding to the method receives a null value, in this case.

The **<*n*>** variable in the parameter name indicates a unique number (1, 2, 3, and so on) in ascending order; for example:

```
Method1 = MyRpsClass::firstMethod
Method2 = MyRpsClass::aVirtualProperty
Method3 = AnotherRpsClass::secondMethod
Method4 = Client::deleteClient
```

Parameter is read when …

The **Datapump** application is started.

Applicable to database role or subrole…

RPS node only.

RPS Manager Application Section [JadeRpsManager]

The [JadeRpsManager] section of the JADE initialization file contains the following parameters, which store information for the RPS Manager application.

- BulkLoadHistoricalTables
- CreateCopyMode
- CreateDatabaseDir
- CreateOverwrite
- ExtractDir
- ExtractOverwrite
- ExtractScriptDir
- LoadExecutionLocation
- Mapping
- RDBName
- RDBPath
- RDBServerName

- Schema
- ScriptPath

Notes All parameters in this section are read when the RPS Manager application is next initialized.

These values are stored in the JADE initialization file so that they can be reused the next time the RPS Manager application is invoked. Set these parameters *only* by using the appropriate menu commands in the JADE RPS Manager window. For details, see "Using the RPS Manager Application", in Chapter 2 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

SDS Administration Utility Section [JadeSDSAdmin]

The [JadeSDSAdmin] section of the JADE initialization file contains the following parameters, which store the options selected in SDS Administration application when the **Save Settings on Exit** option is set. (For details, see "Saving Settings when Exiting from the SDS Admin Application", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*.)

- AutoRefresh
- ColorAlarm
- ColorCatch
- ColorDiscon
- ColorSync
- Font
- RefreshInterval
- SaveSettings
- WindowState

All parameters in this section are read when the SDS Admin database view is next displayed.

Note These values are stored in the JADE initialization file so that they can be reused the next time the SDS Administration application is invoked. Set these parameters *only* by using the appropriate Options menu commands in the SDS Admin application.

For details, see "Using the Options Menu" under "Using the SDS Administration Application", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

Synchronized Database Service Section [SyncDbService]

The [SyncDbService] section of the JADE initialization file contains parameters that control the initial configuration of the Synchronized Database Service (SDS) environment for a JADE database server node.

The [SyncDbService] section can contain the following parameters.

AuditCauseEvents

Value Type Boolean

Default false

Purpose

If the **AuditCauseEvents** parameter is set to **true**, non-immediate user events caused on persistent objects are audited in the transaction journal for replay on secondary databases unless they are outside of a database transaction.

Parameter is read when …

A primary database server is next initialized.

Applicable to database role…

Primary.

Rule

This parameter is actioned only on an SDS primary system. See also "Persistent Events and Notifications" under "Inter-System Event Notifications", in Chapter 10 of the *JADE Developer's Reference*.

ConnectionPollInterval

Value Type Integer *(seconds)*

Default 30

Purpose

The **ConnectionPollInterval** parameter, configured on secondary databases, defines the interval at which the secondary database polls the primary database to determine whether it can be accessed through the communication paths. The minimum value is **5** seconds and the maximum value is **3600** seconds.

A secondary database closes communication channels to the primary database when no messages have been received from the primary database and twice the value of the **ConnectionPollInterval** parameter has elapsed. The primary database closes communication channels to a secondary database when no messages have been received from that secondary database and twice the value of the **ConnectionPollInterval** parameter of that secondary database has elapsed.

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

Applicable to database role…

Secondary.

DatabaseRole

Value Type String

Default Not specified

Purpose

The **DatabaseRole** parameter activates SDS on a database server node and determines the starting database role for the server. The valid values for this parameter are **SecondaryRole** and **PrimaryRole**.

This parameter is automatically updated by a take-over operation when a database role changes. (For details, see "SDS Takeover Operations", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*, or the **JadeDatabaseAdmin** class **sdsInitiateTakeover** method, in Chapter 1 of the *JADE Encyclopaedia of Classes*.)

Parameter is read when …

Each node is next initialized and when the role next changes.

Applicable to database role…

Both primary and secondary.

Rule

When a secondary database has been initialized as a secondary system, you cannot use this parameter to change the role to **PrimaryRole**.

DatabaseSubrole

Value Type String

Default Not specified

Purpose

The **DatabaseSubrole** parameter activates the SDS secondary as an RPS node. The value for this parameter is **RelationalRole** for an RPS node and **NativeRole** for a non-RPS node. For details, see "Running an RPS Node", in Chapter 2 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

Parameter is read when …

Each node is first initialized.

Applicable to database role…

Secondary.

JournalReadBuffers

Value Type Integer

Default 4

Purpose

The **JournalReadBuffers** parameter specifies the number of buffers to use when reading a journal file on disk.

The minimum value for this parameter is **2** and the maximum value is **100**.

Parameter is read when …

The SDS service is next initialized.

Applicable to database role…

Primary (applies to journal transfer).

Secondary (applies to journal replay).

JournalReplayBlocksize

Value Type Integer *(prefix multiplier)*

Default 128K

Purpose

The **JournalReplayBlocksize** parameter specifies the size in bytes of each read buffer used when replaying a journal file.

The minimum value for this parameter is **4**K and the maximum value is **1**G.

Parameter is read when …

The SDS secondary service is next initialized.

Applicable to database role…

Secondary.

JournalXferBlocksize

Value Type Integer *(prefix multiplier)*

Default 128K

Purpose

The **JournalXferBlocksize** parameter specifies the size in bytes of journal file disk and network I/O (input-output) operations. When the **JournalXferCompression** parameter is set to **true**, the journal block size represents the maximum or uncompressed size of a journal data block.

The minimum value for this parameter is **32**K and the maximum value is **1**G.

Parameter is read when …

The SDS primary service is next initialized. You can start and stop the SDS service without restarting the node.

Applicable to database role…

Primary.

JournalXferCompression

Value Type Boolean

Default true

Purpose

The **JournalXferCompression** parameter, when enabled, causes SDS to compress journal data blocks for transmission across the network.

Compressed journal data blocks are uncompressed on receipt by secondary database servers.

Parameter is read when …

The SDS primary service is next initialized. You can start and stop the SDS service without restarting the node.

Applicable to database role…

Primary.

MaxDeferredTransactions

Value Type Integer

Default 30

Purpose

The **MaxDeferredTransactions** parameter enables you to configure the threshold of transactions that can remain deferred as a result of lock conflicts with reader processes. The maximum value is **5000** and the minimum value is **1**.

For more details, see "Deferring SDS Transactions", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

Parameter is read when …

The SDS secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

Applicable to database role…

Secondary.

MaxSecondaries

Value Type Integer

Default 16

Purpose

The **MaxSecondaries** parameter specifies an upper limit maximum number of secondary database servers that can attach to a primary database server. The maximum value is **128** and the minimum value is **4**.

Parameter is read when …

The SDS primary service is next initialized. You can start and stop the SDS service without restarting the node.

Applicable to database role…

Primary.

MyName

Value Type String

Default Not specified

Purpose

The **MyName** parameter specifies a unique name for a JADE system operating within a Synchronized Database Service environment (SDE).

As there is no default, you must specify a value for this parameter. (See also "Connection Parameters Section [ConnectionParams]", earlier in this chapter.) If SDS is activated by specifying a value for the **DatabaseRole** parameter and JADE cannot find the **MyName** parameter, the SDS initialization fails with an initialization error (that is, error 3202 - SDS MyName parameter not found in INI file).

Parameter is read when …

Accessed on the primary database and an SDS primary service is next initialized (that is, you can start and stop the service without stopping the server) and when an automatic refresh is performed.

Applicable to database role…

Both primary and secondary.

PrimaryServerName

Value Type String

Default Not specified

Purpose

The **PrimaryServerName** parameter specifies the name of the JADE system with the primary role at a secondary database server.

As there is no default, you must specify a value for this parameter. In addition, the value that you specify must match the value of the **MyName** parameter on the server that has the primary database role. If SDS is activated by setting the **DatabaseRole** parameter to **SecondaryRole** and a **PrimaryServerName** parameter is not found, the initialization of the secondary database fails with an initialization error (that is, error 3203 - SDS PrimaryServerName parameter not found in INI file).

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

Applicable to database role…

Secondary.

ReadAccessDisabled

Value Type Boolean

Default false

Purpose

If the **ReadAccessDisabled** parameter is set to **true** on a secondary database server when SDS starts up, read access to the database is disabled. The default value of **false** indicates to SDS that read access is required. However, read access may not be granted immediately when an SDS secondary database server is restarted with interrupted transactions pending.

This parameter is automatically updated when access is disabled programmatically or from the SDS Administration application. (For details, see "Enabling or Disabling Read Access of Persistent Objects in the Database", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*, or the appropriate **JadeDatabaseAdmin** class methods, in Chapter 1 of the *JADE Encyclopaedia of Classes*.)

Parameter is read when …

The SDS secondary database is next initialized.

Applicable to database role…

Secondary.

ReconnectInterval

Value Type Integer *(seconds)*

Default 300 (5 minutes)

Purpose

The **ReconnectInterval** parameter specifies the frequency at which a secondary database server attempts to reconnect to its primary server when a primary server is not available.

When the connection to the primary fails, a reconnection is attempted after 10 seconds. If this fails, reconnection attempts are performed using the number of seconds specified in this parameter.

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

Applicable to database role…

Secondary.

ResponseTimeout

Value Type Integer *(seconds)*

Default 60 (1 minute)

Purpose

The **ResponseTimeout** parameter, configured on the primary database, specifies the maximum time the primary waits for a request response from a secondary.

The minimum value is 10 seconds and the maximum value is 120 seconds.

If the timeout expires, the request is terminated with error 3212 (*SDS a response was not received within a reasonable timeframe*). If the secondary disconnects while the primary is waiting for a request response, the request is terminated with error 3204 (*SDS secondary not attached*).

Parameter is read when …

The SDS primary service is next initialized. You can start and stop the SDS service without restarting the node.

Applicable to database role…

Primary.

SyncMode

Value Type String

Default JournalBlockWrite

Purpose

The **SyncMode** parameter specifies the write synchronization mode of journals from an SDS database server node to secondary databases, by determining how the secondary database is kept synchronized with the primary.

When the **SyncMode** parameter is set to the default **JournalBlockWrite** value and it is synchronized with the primary database, audit blocks are transferred and written to the secondary database journal at the same time they are written to the journal on the primary.

In effect, primary journal writes are mirrored to the secondary journal. These mirrored journal writes are asynchronous and the primary database does not wait for an acknowledgment that a write to disk has been completed at the secondary.

When the **SyncMode** parameter is set to **JournalSwitch** value, journal data is transferred to the secondary database when the journal is complete and writing switches to a new journal.

For details, see "Selecting the Journal Transfer Mode for a Secondary Database" under "Synchronizing with a Primary", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*.

Parameter is read when …

The secondary to primary connection is next established; for example, you can disable the connection, change the parameter value, and then reconnect.

Applicable to database role…

Secondary.

TrackingDisabled

Value Type Boolean

Default false

Purpose

If the **TrackingDisabled** parameter is set to **true** on a secondary database server when SDS starts up, the database tracking (journal replay) process is disabled.

When database tracking is disabled, journals can still be received and accumulated while a secondary database server remains connected to its primary database. When SDS is started and this parameter is set to the default value of **false** on a secondary database server, tracking is initialized, allowing the replay through all ready journals.

This parameter is automatically updated when access is disabled programmatically or from the SDS Administration application. (For details, see "Enabling or Disabling Tracking", in Chapter 1 of the *JADE Synchronized Database Service (SDS) Administration Guide*, or the appropriate **JadeDatabaseAdmin** class methods, in Chapter 1 of the *JADE Encyclopaedia of Classes*.)

Parameter is read when …

The SDS secondary database is next initialized.

Applicable to database role…

Secondary.

Web Options Section [WebOptions]

The [WebOptions] section of the JADE initialization file contains parameters that enable you to specify options for your Web pages.

Note All parameters in this section are read when the Web application is next initialized; that is, when it starts up or when it is bounced to restart it.

The [WebOptions] section can contain the following parameters.

ApplicationConfigFile

Value Type String *(file-name)*

Default None

Purpose

The **ApplicationConfigFile** parameter specifies the file name and the full path of a separate XML configuration file that contains runtime settings for JADE web-enabled applications; that is, JADE forms applications, HTML Document applications, and Web service provider applications.

Notes The XML configuration file specified by the **ConsumerConfigFile** parameter and the file specified by the **ApplicationConfigFile** parameter are separate configuration files.

JADE web forms use the **Form** class **secureForm** property to control whether the URLs are generated using the HTTP or HTTPS protocol.

The **protocol** tag of the XML Web Configuration file is ignored for JADE web forms.

If the XML configuration of a Web application is specified, Web options settings are obtained from this file. If an XML configuration file is not specified, Web options settings are obtained from the parameters in the [WebOptions] section of the JADE initialization file.

Note As some parameters can be specified only in the XML configuration file of a Web application, you should configure your Web applications in this way.

JADE provides an application that enables you to create and maintain XML configuration files.

For details about specifying options for Web applications in a separate XML file, see Chapter 3 "Configuring Web Applications", of the *JADE Web Application Guide*.

ApplicationCopies

Value Type Integer *(number-of-application-copies)*

Default 1

Purpose

The **ApplicationCopies** parameter specifies the maximum number of copies that are started when the HTML thin client application starts.

All application copies use the same port number. If you specify **6014** as the port number in the **ConnectionName** parameter and the value of the **ApplicationCopies** parameter is three, the copies use port numbers **6014**, **6015**, and **6016**.

If you want to define multiple applications with different numbers of application copies in the same JADE initialization file, prefix the **ApplicationCopies** parameter with the name of your application followed by an underscore character (_); for example:

```
BargainBinWebService_ApplicationCopies=15
```

Hints

Although the recommended value is the default value (that is, **1** copy), you can specify a maximum of 62 copies, which is a Windows-imposed limit.

For details about specifying HTML thin client options in the JADE development environment, see "Specifying Your HTML Thin Client Access Options", in Chapter 1 of the *JADE Web Application Guide*.

ConnectionName

Value Type String *(connection-name)*

Default Application name

Purpose

The **ConnectionName** parameter specifies the TCP/IP address and port number through which your Web application communicates to an HTML thin client through the Internet. The format for the parameter is <*TCP-address*>:<*port-number*> (for example, **143.67.78.90:6014**), which is the same format used in the **Connection Name** text box on the **Web Options** sheet of the Define Application dialog.

If you have more than one application with the same name, specify a unique name to identify the connection. The connection name can include any character or number in the ranges **A** through **Z**, **a** through **z**, or **0** through **9**. The first character must be uppercase.

If you want to define multiple applications with different connection names in the same JADE initialization file, prefix the **ConnectionName** parameter with the name of your application followed by an underscore character (_); for example, **BargainBinWebService_ConnectionName**.

```
BargainBinWebService_ConnectionName=wilbur1a:5200
```

Hint

For details about specifying HTML thin client options in the JADE development environment or obtaining the IP address, see "Specifying Your HTML Thin Client Access Options", in Chapter 1 of the *JADE Web Application Guide*.

ConsumerConfigFile

Value Type String *(file-name)*

Default None

Purpose

The **ConsumerConfigFile** parameter specifies the file name and the full path of a separate XML file that contains the runtime configuration settings for JADE Web service consumer applications.

Note The XML configuration file identified by the **ConsumerConfigFile** parameter and the file specified by the **ApplicationConfigFile** parameter are separate configuration files.

JADE provides an application that enables you to create and maintain XML configuration files.

For details about specifying Web options for a JADE Web service consumer application in a separate XML file, see Chapter 3 "Configuring Web Applications", of the *JADE Web Application Guide*.

DisableLogging

Value Type Boolean

Default false

Purpose

The **DisableLogging** parameter specifies whether information is logged in the JADE Web Application Monitor window. By default, monitor information is logged in the text window.

Set this parameter to **true** if you want to stop the display of logging information in the text window (for example, when you are running an application as a service and the Web Application Monitor window View menu item cannot be used).

Hint

The View menu in the Web Application Monitor window provides the **Disable Logging** or **Enable Logging** command, which you can use to stop the display of information in the text window or cause the redisplay of logging information when the Web Application Monitor is displayed on the client node workstation on which the Web-enabled JADE application is running.

For more details about the Web Application Monitor window, see "Monitoring Your Web Sessions", in Chapter 1 of the *JADE Web Application Guide*. See also the **LogFileName** parameter, later in this section.

Firewall

Value Type Boolean

Default false

Purpose

The **Firewall** parameter specifies whether there is a firewall between the JADE Web Application directory that contains the files transferred over a TCP/IP connection and your JADE Web Application.

For details about configuring a firewall for a JADE Web Application, see "Firewall for the JADE Internet Environment", in Chapter 2 of the *JADE Installation and Configuration Guide*.

Set this parameter to **true** if you require firewall separation.

Note The firewall must also be enabled at the other end of the connection. For Microsoft Internet Information Server (IIS), the **Firewall** parameter in the [Jadehttp Files] section of the JadeHttp initialization file must be set to **true**. For Apache HTTP Server, the **Firewall** directive in the **Apache Configuration Directives File** must be set to **on**.

ImageType

Value Type String *(file-type)*

Default jpg

Purpose

The **ImageType** parameter specifies the type of image that is displayed on your Web pages. The valid values are the default **.jpg** (Joint Photographic Experts Group) file type, the .**png** (Portable Network Graphics) file type, and the .**gif** (Graphics Interchange Format) file type.

Note Pictures on Web pages can be types .**gif** (Graphics Interchange Format), ..**jpg** (Joint Photographic Experts Group), or ..**png** (Portable Network Graphics) only.

Hint

Set this parameter to **png** if you want to use images with lossless compression on your Web pages. Portable Network Graphics (.**png**) files provide greater clarity and no loss of definition.

Caution Portable Network Graphics (.**png**) files may be larger than Joint Photographic Experts Group (.**jpg**) image files.

LimitPortRange

Value Type Boolean

Default false

Purpose

Set the **LimitPortRange** parameter to **true** if you want to prevent your Web applications from using a port number that exceeds the starting port number plus the number of copies of the application. As this parameter is set to **false** by default, the port range is not limited.

When you set this parameter to **true**, an application cannot be started when the port limit is exceeded, and the following message is output to the **jommsg.log** file.

```
Port number range exceeded - application will be terminated
```

LogFileName

Value Type String *(file-name)*

Default Not specified

Purpose

The **LogFileName** parameter specifies the file name and the full path to which output displayed in the Web Application Monitor is directed for later analysis of transaction times, and so on.

If the specified file name cannot be written to or it is not valid (for example, you did not specify an absolute path or the specified path does not exist), a file called **websession.log** is created in the physical directory specified in the **Physical Directory** text box for your JADE forms on the **Web Options** sheet of the Define Application dialog or in the **PhysicalDirectory** parameter in the [WebOptions] section.

For more details about the Web Application Monitor window, see "Monitoring Your Web Sessions", in Chapter 2 of the *JADE Web Application Guide*.

LogMessageContent

Value Type Boolean

Default true

Purpose

The **LogMessageContent** parameter specifies whether Web message content is logged in the JADE Web Application Monitor window. By default, any Web logging includes the **Query String =** ***content*** and **Http String =** ***content*** output.

Set this parameter to **false** if you do not want to include the **Query String =** ***content*** and **Http String =** ***content*** message content.

This parameter applies only if the **DisableLogging** parameter in the [WebOptions] section of the JADE initialization file or the **disable_logging** element in the Web application configuration file is set to **true**, or tracing is turned on when the **Enable Logging** command is displayed in the View menu of the JADE Web Application Monitor window. (The value of this command toggles between **Enable Logging** and **Disable Logging**.)

When you initiate the Web Application monitor and tracing is on, a message is displayed indicating the status of the **LogMessageContent** parameter, as follows.

```
Message content logging is enabled|disabled
```

PhysicalDirectory

Value Type String *(physical-directory-path)*

Default Not specified

Purpose

The **PhysicalDirectory** parameter enables you to specify the physical directory path for Web-enabled applications. Specify this parameter if you want to override the working directory specified in the **Physical Directory** text box for your JADE forms on the **Web Options** sheet of the Define Application dialog for the application.

Note The physical directory, which is used to generate images for the Web, applies only to Web-enabled (that is, HTML-enabled) applications and not to Web service applications.

To specify multiple physical directories in the same JADE initialization file, prefix the **PhysicalDirectory** parameter with the connection name followed by an underscore (_) character. For example, if the connection name is **CustomerInvoice**, you would specify the following parameter and value.

```
CustomerInvoice_PhysicalDirectory=\customer\system
```

Where there is no specific connection name prefix, the connection name without the prefix is used. If you do not specify this parameter, the physical working directory specified in the Define Application dialog is used.

ReadTimeout

Value Type Integer *(seconds)*

Default Not specified

Purpose

The **ReadTimeout** parameter enables you to specify the number of seconds for which Web-enabled applications wait for message transfers to complete before timing out. This parameter can have one of two forms, as follows.

```
ReadTimeout

application-name_ReadTimeout
```

The default value is **600** seconds (10 minutes). If you do not want the Web-enabled application to time out while waiting message transfers to complete, specify this parameter and set it to zero (**0**).

Use the ***application-name*_ReadTimeout** form of this parameter if you want different applications to have different message transfer timeout values.

URLSpecifications | *application-name*_URLSpecifications

Value Type String *(URL-name | application-name_URL-name)*

Default Not specified

Purpose

The **URLSpecifications** parameter enables you to specify Internet server virtual directories for HTML documents for all of your Web-enabled applications or for a specific Web-enabled application.

This parameter can have one of two forms, as follows.

```
URLSpecifications

application-name_URLSpecifications
```

Use the ***application-name*_URLSpecifications** form of this parameter if you want different applications to have different URL specifications.

The values for the ***application-name*_URLSpecifications** form of this parameter are as follows.

```
protocol, machine-name, virtual-directory-name
```

In this format, the *protocol* value must be **http** or **https**, the *machine-name* value is the name of the target host for hyperlinks and POST actions, and the *virtual-directory* value is the name of the virtual directory for hyperlinks and POST actions.

The parameter is ignored if the *protocol* value is not **http** or **https**.

The following is an example of the ***application-name*_URLSpecifications** form of this parameter.

```
CustomerWeb_URLSpecifications = https, JadeAppsServer, /customer
```

If JADE locates both forms of this parameter in the [WebOptions] section, the specific form (that is, ***application-name*_URLSpecifications**) takes precedence. If you do not specify the **URLSpecifications** or ***application-name*_URLSpecifications** parameter in the JADE initialization file, the Internet server directory information is obtained from the Microsoft Internet Information Server (IIS). For details, see "Specifying Your HTML Thin Client Access Options", in Chapter 1 of the *JADE Web Application Guide*.

For details about programmatically setting your machine name and virtual directory to be used when generating HTML pages for the **JadeHTMLClass** class **buildFormActionOnly** and **buildLink** methods, see the **Application** class **setWebMachineName** and **setWebVirtualDirectory** methods in Chapter 1 of the *JADE Encyclopaedia of Classes*.

UseHTML4ForNetscape

Value Type Boolean

Default false

Purpose

The **UseHTML4ForNetscape** parameter specifies whether HTML generation for Netscape and Mozilla browsers is similar to that for Internet Explorer.

For non-Internet Explorer browsers, the version number returned must be 5 or greater. (For Internet Explorer browsers, there is no change to the current implementation.)

Note For a page containing sheets to display correctly in cross-browser compatibility mode or when the value of **UseHTML4ForNetscape** is **false**, all or none of the sheets must have an icon.

VirtualDirectory

Value Type String *(virtual-directory-name)*

Default Not specified

Purpose

The **VirtualDirectory** parameter enables you to specify the name of the virtual directory for Web-enabled applications.

Specify this parameter if you want to override the virtual directory specified in the **Virtual Directory** text box for your JADE forms in the **Web Options** sheet of the Define Application dialog for the application. (This text box enables you to specify a location for your images and Java applets other than the working directory of the JADE application.) There is no default virtual directory.

For details, see "Specifying Your HTML Thin Client Access Options", in Chapter 1 of the *JADE Web Application Guide*.

Note You must also set up this virtual directory mapping in IIS.

To specify multiple virtual directories in the same JADE initialization file, prefix the **VirtualDirectory** parameter with the connection name followed by an underscore (_) character. For example, if the connection name is **CustomerInvoice**, you would specify the following parameter and value.

```
CustomerInvoice_VirtualDirectory=/customer
```

Where there is no specific connection name prefix, the connection name without the prefix is used. If you do not specify this parameter, the virtual directory specified in the Define Application dialog is used.

WebServicesURL

Value Type String *(URL-name)*

Default Not specified

Purpose

The **WebServicesURL** parameter enables you to change Web service URL settings; that is, to override the optional Web services scheme and the machine name, virtual directory, and support library defined in the application so that the Web service test harness functions correctly.

Alternatively, you can change URL settings in the JADE development environment by using the controls in the URL Setting group box on the **Web Services** sheet of the Define Application dialog. For details, see "Defining a Web Services Application", in Chapter 11 of the *JADE Developer's Reference*.

The URL definition is defined as *[scheme,]machine-name,virtual-directory-name,support-library*, as shown in the following example.

```
WebServicesURL=tcp,wilbur1a:6556,jade,jadehttp.dll
```

Notes If you want to specify a direct Web service between JADE systems, specify **tcp** in the optional *scheme* value of the Web services URL. (By default, the communication protocol is **http**.)

For *direct* Web services, the machine name must contain the machine name or the IP address followed by a colon (**:**) character then by a TCP port number on which this service is offered.

For a Web service consumer application, you can dynamically override the WSDL machine name, virtual directory, and support library by calling the **JadeWebServiceConsumer** class **setEndpointURL** method.

Hint

If you need to define multiple applications with different connection names in the same initialization file, prefix the **WebServicesURL** parameter with the name of your application followed by an underscore (**_**) character; for example, if your application is called **MyWebService**, the parameter is **MyWebService_WebServicesURL** so that your initialization file parameter value looks like the following.

```
MyWebService_WebServicesURL=wilbur1a,jade,jadehttp.dll
```

Web Session Section [WebSession]

The [WebSession] section of the JADE initialization file contains parameters that enable you to specify options for your Web sessions.

The [WebSession] section can contain the following parameters.

LockRetries

Value Type Integer

Default 10

Purpose

The **LockRetries** parameter specifies the number of times that a lock encountered in a Web session is retried.

Parameter is read when …

Each lock is encountered.

Hint

The optimum value for this parameter depends on how quickly you need to know that a resource is locked; that is, it is application-dependent. If it is essential that you know immediately that the resource is unavailable, you should set this parameter to a low value.

PromptOnShutdown

Value Type Boolean

Default true

Purpose

The **PromptOnShutdown** parameter, when set to **false**, causes the suppression of the message dialog that is displayed when a user shuts down a Web session.

The default value of **true** displays a message dialog that prompts the user to confirm the shutting down of the Web session.

Parameter is read when …

The Web application is next initialized; that is, when it starts up or when it is bounced to restart it.