JADE™

Migrating to JADE

Consolidated Release Information

VERSION 6.3.04

Contents

Changes in JADE Release 6.3.04, continued

Migrating to JADE Release 6.3.04

This document covers the following topics.

Refer to **RelInfo6303.pdf** in your JADE **documentation** directory for details about JADE release 6.3.03 changes that may affect your JADE 6.2 existing schemas and changes in JADE release 6.3.03.

Tip For information about using Acrobat Reader to view JADE documents, see "JADE Product Information Library in Portable Document Format", in Chapter 2 of your JADE *User's Guide* in your JADE **documentation** directory.

The JADE *Product Information Library* document (**JADE.pdf**) provides a summary of JADE product information library documents and navigation to them.

If you want to develop your own installation process:

- For Windows, the JADE install and upgrade steps are documented in the **ReadmeInstallSteps.txt** file in the **\documentation** directory.
- For Linux, the steps are as executed in the **pre_i** and **post_i** scripts in the **bin** directory in the **/opt/jade** subdirectory for the release.

To customize the deployment upgrade on Windows, see Appendix A, "Customizing the Deployment Upgrade Process", in your JADE *Runtime Application Guide*.

JADE Release Support

With the release of JADE 6.3, support for JADE 6.1.15 (the final JADE 6.1 release) continues until October 2010. Prior versions of JADE 6.1 will not be supported.

For details about the JADE release policy, go to:

http://www.jadeworld.com/downloads/jade/JADE_ReleasePolicy.pdf

For details about the JADE release schedule, go to:

http://www.jadeworld.com/jade/updates.htm

Deimplementations and Deprecations

This section contains the deimplementations and deprecations in JADE 6.3.

Exposure Deimplementation

Notice is given of the intent to deimplement JADE's ActiveX exposure feature, starting from JADE release 7.0. This means that from JADE 7.0, the ActiveX exposure feature will become unavailable.

The ActiveX exposure enables you to expose selected features of your JADE system to application development tools such as Microsoft's Visual Basic and C++ languages through ActiveX technologies. JADE implements ActiveX generation through the wizard feature of Microsoft's Visual Studio 6. However, this product is no longer supported by Microsoft. In addition, new users of JADE may be unable to source a copy of Visual Studio 6.

In recent years, ActiveX technologies have been replaced by .NET. From JADE 6.3, you can now generate exposures using these .NET technologies. This provides a more modern, flexible, and easier to develop mechanism than that provided by ActiveX.

It is recommended that, where required, you re-write ActiveX exposures using the new JADE .NET exposure. For details, see Chapter 19 of the JADE *Development Environment User's Guide*.

JADE Portable Graphical User Interface (GUI) Client

JADE portable GUI client is deprecated for both Linux and Windows platforms.

In the second half of 2009, we plan to release a Silverlight-based thin client that will enable JADE user interfaces to be deployed in Web browsers via Silverlight and Moonlight. Silverlight is Microsoft's cross-browser, cross-platform plug-in for delivering media and rich user interfaces to the Web.

The Silverlight plug-in is freely available for all major browsers including Internet Explorer (IE), Firefox, and Safari on Windows and the Macintosh operating system; and the Mono Project has released Moonlight, which is an open source implementation of the Silverlight plug-in for Firefox on Linux.

PKWare Compression

As notified in JADE 6.2 release information, PKWare compression is not supported in this release.

Before upgrading to JADE 6.3:

1. Change existing code in your JADE 6.2 applications to use the **compressToBinary** method of the **Binary**, **String**, and **StringUtf8** primitive types and the **uncompressToBinary**, **uncompressToString**, and **uncompressToStringUtf8** methods of the **Binary** primitive type.
2. If you have persistent data that has been compressed using the PKWare compression libraries, update that data by uncompressing it using the appropriate **uncompress**, **uncompressString**, or **uncompressStringUtf8** decompression method, and then recompress it using the **compressToBinary** method using a compression option from the **Binary** primitive **Compression_Zlib**, **Compression_ZLibFast**, or **Compression_ZLibSmall** constant.

Note JADE strongly recommends that you make the changes necessary to transition to the use of zlib compression.

However, if you elect to continue to use PKWare compression, you should be aware that before you can upgrade to a 64-bit version of JADE, coding changes to use new methods and data recompression are necessary. For details about using PKWare compression in 32-bit editions of JADE 6.3, see "Compression and Decompression Methods Deimplemented", in the JADE 6.3.03 Release Information document (that is, **RelInfo6303.pdf**).

Real[10] Parameters in External Function Calls

As notified in JADE 6.2 release information, external function calls with **Real[10]** parameters are no longer supported.

The upgrade validation process checks external functions for **Real[10]** parameters, logs any detected usages, and the upgrade fails. You must change the usages to Real[8] and re-run the validation.

RPS SQL Script Execution

In this release, **sqlcmd** has replaced the ODBC interface as the default mechanism for SQL script execution on an RPS node.

SQL scripts are used to create or alter table definitions and to load data. These scripts can be executed from the RPS Manager utility or automatically by the **Datapump** application.

The SQL Server **sqlcmd** utility is the preferred mechanism for SQL script execution. In order to use **sqlcmd**, it must be installed on the machine hosting the RPS node. The SQL Server instance name is specified from the RPS manager node configuration dialog.

The advantages of using **sqlcmd** are as follows.

- Error results, which are lost when using the ODBC interface, are correctly reported.
- The error information from SQL Server is saved in a log file.

Use of the ODBC interface for script execution is supported in JADE release 6.3. However, it will be deimplemented in JADE 7.0.

Accessing Details about Faults Fixed in Releases

To access the complete documentation about the PARs fixed in this release, you can directly access **Parsys**, our Fault Managements and Customer Contact system. This enables you not only to view the progress of your own contacts but also to view all of the PARs fixed in a specific release.

If you have any queries about **Parsys**, please direct them to JADE Support in the first instance.

You can download the install shield for **Parsys** from the following URL.

http://www.jadeworld.com/Jade6/jade6_parsys.htm

When you first run the **Parsys** application, it downloads an update via the automatic thin client download feature. When this has completed and you have the log-on form ready and waiting, please contact JADE Support, who will then send you an e-mail message with your user code and password details. **Parsys** requires you to change your password when you first log on.

Note Because the encryption of passwords is a one-way algorithm, we cannot advise you of your password should you forget it, but we can reset it to a known value again.

How to Locate PARs Fixed in a Specific Release

This section describes the actions that enable you to locate Product Anomaly Reports (PARs) fixed in a specific release.

➢ **To perform an advanced search**

1. Select the **Advanced Search** command from the Search menu with the following settings on the **Basic Search Criteria** sheet.
   a. The **Latest** option button is selected in the Mode group box.
   b. **All** is selected in the **Priority** list box.
   c. The **PAR** check box is checked in the Phase group box.
   d. The **Fault** check box is checked in the Type group box.
   e. The **Closed** and **Patched** check boxes are checked in the Status group box.

   Note If you want to restrict the search to the hot fixes that were produced, check the **A hot fix was created** check box on the **Advanced Search Criteria II (Optional)** sheet.

2. On the **Advanced Search Criteria III (Optional)** sheet:
   - In the **Closed** list box of the Releases group box, select the release whose fixed PARs you want to locate (for example, the **6.3.0** list item).
3. Click the **Search** button.

Upgrading to JADE 6.3.04 from a JADE 6.2 or Earlier JADE 6.3 Release

This section covers the following topics.

Upgrading to JADE 6.3 from a Windows JADE 6.2 or Earlier JADE 6.3 Release

You can upgrade from a 6.2 release to the 64-bit edition of JADE only if patch versioning has never been used in the JADE system. If patch versioning has ever been used, you must first upgrade to the 32-bit edition before you can upgrade to the 64-bit edition.

If you are upgrading to the 32-bit edition of JADE, the Microsoft Windows C++ 2005 Redistributable Package (x86) called **vcredist_x86.exe** is required. If you are upgrading to the 64-bit edition, the Microsoft C++ 2008 SP1 Redistributable Package (x64) called **vc_x64Runtime.exe** is required. (These executables are supplied on the JADE distribution media.)

Note Installing this Microsoft redistributable package requires administration privileges. If possible, deploy this package to all workstations *before* upgrading to JADE 6.3, using the appropriate techniques that allow for privileged installations. If the required package is not already installed, it will be installed during the JADE installation.

The JADE Setup program enables you to upgrade your binary and database files to JADE 6.3 from a JADE 6.2 or earlier 6.3 release on Windows, by performing the following actions.

1. Ensure that your JADE environment is JADE release 6.2 or 6.3.
2. If you are upgrading from JADE 6.2, uninstall any **jadrap** database services (that is, JADE Remote Node Access services) that you have set up. (This is required because of changes in the underlying registry entries that are expected or created in JADE 6.3.)
3. If you are upgrading to JADE release 6.3 under Vista, ensure that you have the appropriate privileges or capabilities to install applications.

   The configuration of Vista's User Account Control (UAC) and your current user account privileges may affect the behavior of the upgrade to JADE 6.3.

   For details about Windows Vista UACs, standard user accounts, and administrator accounts, see:

   http://technet2.microsoft.com/WindowsVista/en/library/00d04415-2b2f-422c-b70e-b18ff918c2811033.mspx?mfr=true

4. Take a full backup copy of your existing JADE directories, first ensuring that your database is not in recovery mode.

   Caution As roll-forward recovery of the installation and upgrade process is not supported, it is important that you backup your database *before* starting the JADE Setup process to install JADE 6.3 and upgrade your existing data.

5. To start the JADE Setup program, invoke the **setup.exe** program from the **Jade63** release medium or execute the executable program downloaded from the JADE Web site.
6. The Microsoft Windows C++ Redistributable Package is installed, if not already installed.
7. In the Welcome folder, click the **Next>** button to continue the upgrade process.
8. Read the entire software license agreement in the Software License Agreement folder and then click the **Yes** button to continue the installation.
9. In the Installation Type folder, select the **Feature Upgrade** option button, to specify that you want to upgrade an existing JADE release. By default, the **Fresh Copy** option is selected.
10. In the Setup Type folder, select the type of upgrade that you require. By default, the **Development** option is selected. If you do not want development files upgraded, select the **Application Runtime**, **Presentation Client**, **Jade Client**, or **SDS/RPS Database Server** option button, as required.

    Note The **Custom** type applies only to a **Fresh Copy** installation type, and is not relevant when upgrading binary and database files. The **SDS/RPS Database Server** option applies only to the **Feature Upgrade** installation type.

11. In the Select Installation Folders folder, specify the locations of the JADE files that are to be upgraded.

    The upgrade process defaults to the most-recently used JADE files, and displays these values in the **Install Directory**, **Executable Directory**, and **Database Directory** text boxes. The installation directory is most likely to be the root directory in which you installed JADE, unless you subsequently renamed the root directory or moved the files to another location.

    If the locations are not as required, click the adjacent browse buttons (indicated by the ... ellipsis symbols) to display the common File Selection dialog that enables you to select the appropriate directories and files. By default, the **jade.ini** file located in your specified database directory is used.

    If required, use the **JADE INI File** text box to specify a different valid fully qualified directory and name of the JADE initialization file; for example:

    ```
    d:\mysys\jadetest\system\jadetest.ini
    ```

    If Program Start folders are to be updated, specify the name of the folder in the **JADE Program Folder** text box. If you are unsure of the folder to be updated, click the adjacent **browse** button to display the common Folder selection dialog that enables you to select the folder.

    The **Database Directory** text box enables you to explicitly specify the location in which the database (system) files are installed.

    When installing on a non-Vista operating system or on Vista when the destination folder is not **\Program Files**, the database destination defaults to **system** under the install folder (for example, if you specify **c:\Jade63** in the **Install Directory** text box, the database directory defaults to **c:\Jade63\system**).

If the installation directory is a subdirectory of the programmatically determined location of **\Program Files** on Vista, the **\Program Files** portion of the install directory is replaced with programmatically discovered location for the common application data directory (for example, if you specify **c:\Program Files\Jade63** in the **Install Directory** text box, the default database location is **c:\ProgramData\Jade63\system**).

The process checks whether the specified database directory is a valid system and that it is the correct ANSI or Unicode type.

12. The **JADE JIT Debugger** folder is displayed. The JADE Just-In-Time (JIT) debugger is required to reliably acquire a dump and crash log when an exception occurs in a system running with the Microsoft Visual C++ run time.

    Note It is recommended that you install this on all machines hosting JADE nodes.

13. The Start Copying Files folder summarizing your upgrade options is displayed. If the selections displayed in the Start Copying Files folder are correct, click the **Next>** button. Alternatively, click the **<Back** button to modify your selections.

14. The Question dialog is displayed, advising you to ensure that you have taken a full backup of that database before you proceed with the upgrade process.

    When you are sure that you are upgrading the correct system (and that it has been backed up), click the **Yes** button to start the upgrade process.

15. A warning message box is then displayed, advising you that Dynamic Link Libraries (DLLs) may need to be recompiled. Click the **OK** button, to continue.

16. A warning message may be displayed if the upgrade validation process has not completed. If so, check the **jadeupgrade.log** file for details about what needs to be modified in your user schemas to pass the validation and enable application execution.

17. When the upgrade is complete, the JADE Setup program informs you that the JADE Setup was successfully completed and that you can now view the **ReadMe.txt** file.

    To view the **ReadMe.txt** file, ensure that the check box is checked (the default). The **ReadMe.txt** file is then displayed in a text editor (for example, Notepad). The **ReadMe.txt** file is a read-only text file installed in your JADE root directory that you can print or delete, if required. This file contains a reference to other JADE-related documents.

18. Click the **Finish** button to end the JADE upgrade process.

19. Install any **jadrap** database services (that is, JADE Remote Node Access services) you had set up in JADE 6.2. (For details, see "Running the Server Node as a Service", in the JADE *Remote Node Access Utility User's Guide*.)

Caution As with any JADE release, you may need to recompile any external method Dynamic Link Libraries (DLLs) or external programs using the JADE Object Manager Application Programming Interfaces (APIs) with the new JADE **\Include** and **\Library** files before you attempt to run your upgraded JADE systems. (For details about the JADE Object Manager APIs, see Chapter 3 of the JADE *Object Manager Guide*.)

Some obsolete files are deleted from the JADE directories when upgrading from JADE 6.2. If you require these files for your JADE system, you must save them before you upgrade or restore them from the original JADE 6.2 release medium.

Running Two Windows Releases of JADE on the Same Workstation

You can have two releases of JADE installed on the same workstation, if the files are in different directories.

If ODBC is installed, only the last installation of the JADE ODBC driver is available from the ODBC Data Source Administrator.

Upgrading to JADE 6.3 from a Linux JADE 6.2 or Earlier JADE 6.3 Release

You can upgrade from JADE 6.2 to the 64-bit edition of JADE only if patch versioning has never been used in the JADE system. If patch versioning has ever been used, you must first upgrade to the 32-bit edition before you can upgrade to the 64-bit edition.

To upgrade from an existing JADE release to JADE release 6.3 on UNIX servers under SUSE Linux Enterprise Server 10.0 or Red Hat 5.0 or higher, perform the following actions.

1. Ensure that your JADE environment is JADE release 6.2 or earlier JADE 6.3 release.
2. Take a full backup copy of your existing JADE directories, first ensuring that your database is not in recovery mode.
3. Install the required JADE Red Hat Package Manager (RPM) file directly by using the standard Linux RPM install tools. This puts the files in the /**opt/jade** directory.

   For details, see "Installing JADE", in Chapter 3 or Chapter 4 of the JADE *Installation and Configuration Guide*.
4. To upgrade your JADE installation, use the **jadeinstall –U** parameter, as shown in the following example.

   ```
   /opt/jade/sbin/jadeinstall -i /home/jade -U -v 6.3.03.000 --all
   ```

   The parameter values are as follows.

   - **-i <*dir*>** is the previously installed JADE directory
   - **-v <*version*>** is the new JADE version to be installed
   - **--all** indicates that all components previously installed into the directory specified in the **-i** parameter will be upgraded

The upgrade process copies over the new binaries and required system map files, resets timestamps, and performs any other steps necessary to complete the upgrade.

Caution As with any JADE release, you may need to recompile any external method libraries or external programs using the JADE Object Manager APIs with the new JADE **/include** and **/lib** files before you attempt to run your upgraded JADE systems. (For details about the JADE Object Manager APIs, see Chapter 3 of the JADE *Object Manager Guide*.)

For more details, see "Installing JADE on a UNIX Server under Linux" and "Parameters for the jadeinstall Command", in Chapter 3 or Chapter 4 of the JADE *Installation and Configuration Guide*.

JADE Thin Client Upgrade

A JADE 6.3 presentation client upgrade:

- Rejects a presentation client upgrade from 5.2.08 or earlier. You must handle a presentation client download from JADE 6.1 by first upgrading JADE on the presentation client to release 6.0, 6.1, or 6.2.
- Cannot handle a reversion to JADE 6.1 or earlier.
- Rejects a reversion to JADE 6.0 or 6.1 if the JADE 6.0 or 6.1 application server attempts to download files to the **DownloadDirectory2** directory. The only reversion that is guaranteed is from JADE release 6.3 to JADE release 6.2.
- If you are upgrading presentation clients to JADE release 6.3 under Vista, ensure that you have the appropriate privileges or capabilities to install applications. The configuration of Vista's User Account Control (UAC) and your current user account privileges may affect the behavior of the upgrade to JADE 6.3.

  For details about Windows Vista UACs, standard user accounts, and administrator accounts, see:

  http://technet2.microsoft.com/WindowsVista/en/library/00d04415-2b2f-422c-b70e-b18ff918c2811033.mspx?mfr=true

When running JADE in thin client mode under Microsoft Windows Vista or Windows Server 2008, if the presentation client is installed:

- Under the **\Program Files** directory (or the **\Program Files (x86)** directory on a 64-bit machine with 32-bit JADE binaries), when an automatic presentation client upgrade occurs, you must have administrator rights or know the user name and password of a user with administrator rights. (This is a normal Microsoft requirement that updating of files under the **Program Files** directory requires administrative rights.)

  If the Vista machine has had UAC disabled, the thin client upgrade will fail because of lack of permissions for standard users. For administration users, the necessary privileges are automatically granted so the upgrade will succeed.

  If UAC is not disabled, administrative users are prompted with an **Allow** or a **Cancel** choice but standard users must know and supply the logon and password of a user with administrative privileges to enable the upgrade to succeed.
- Outside of the **\Program Files** directory (or the **\Program Files (x86)** directory), privilege elevation to perform an automatic thin client upgrade is not requested.

For more details, see Appendix B, "Upgrading Software on Presentation Clients", in the JADE *Thin Client Guide*.

Upgrading a 32-Bit Presentation Client Connecting to a 64-Bit Application Server

When a 32-bit presentation client connects to a 64-bit application server, the application server upgrades the version of the presentation client but it does not change the 32-bit to 64-bit type of the presentation client, because:

- The presentation client does not check to see if the operating system on which it is running is 64-bit-capable (and it would have to inform the application server about this).

- Any support libraries needed by the presentation client (for example, ActiveX control and automation libraries) would also have to be downloaded or already installed in the presentation client.

The 32-bit version of presentation client binaries must be installed on the 64-bit application server, in the ***<jade-program-data-directory>*/i686-msoft-win32-ansi/download/** directory structure.

If you require a 64-bit presentation client, you must manually install it. Once installed, it will automatically upgrade with 64-bit binaries.

Upgrading a Synchronized Database Environment (SDE)

To upgrade a Synchronized Database Service (SDS) installation from JADE release 6.2.12 or higher to JADE 6.3, perform the following actions.

1. Upgrade the primary system, as specified in earlier sections of this document.

   Note Upgrading a primary database causes a special **version check** trigger record to be written to the database journal to mark the boundary where conversion from JADE 6.2 to 6.3 occurred.

2. Connect any JADE 6.2 native or RPS secondary database servers to the JADE 6.3 primary so that any remaining JADE 6.2 database journals are transferred and applied.

   When the upgrade **version check** record is replayed, the following messages are recorded in the **jommsg.log** and tracking is halted.

   ```
   SDS: Secondary upgrade version mismatch: tracking will now halt

   SDS: Upgrade to the same software release level as the primary and
   restart server
   ```

3. When database tracking halts at the upgrade trigger point, shut down the server and upgrade the secondary system by performing one of the following actions.

   - On Windows, use the automated InstallShield script provided in this release and select the **Feature Upgrade** option on the Installation Type dialog.

     On the Setup Type dialog that is then displayed, select the **SDS/RPS Database Server** option.

   - Copy the:

     i. Binary files (including **_sys*.bin** and **_jad*.bin** files)

     ii. Monitor and dmpload map files

     iii. Reset the timestamps

Upgrading an RPS Node from JADE 6.2 to 6.3

On upgrade from 6.2 to 6.3, the database type on the RPS node is set to **SQL Server 2000**, since SQL Server 2000 data types only were used in JADE 6.2.

If the RPS mapping data base type was set to **SQL Server 2005** in the 6.2 system and you want the RPS mapping to use the SQL Server 2005 types, after the upgrade to 6.3 has completed, execute the following steps.

1. Start the RPS Manager on the RPS node.
2. Stop the **Datapump** application.

3. In the Configure RPS Node dialog, change the data base type to **SQL Server 2005**.

   This causes SQL scripts to be created and run to modify the columns to use the **SQL Server 2005** database types.

4. Start the **Datapump** application.

Upgrade Validation

During the upgrade process, a validation script is run to check the integrity of the upgraded system. Any user schema entities that conflict with system schema entities are logged as errors in the **jommsg*n*.log** file.

All errors must be corrected and validation re-run before user applications can be executed on the updated system. If the system is in the un-validated state, a message box is displayed when you log on to the JADE development environment, asking if validation should be re-run.

Hot Fix Releases

Hot fixes for JADE are released as binary files. To apply the hot fix:

1. Shut down the system.
2. Copy the hot fix system files into the appropriate directory.
3. Start up the system.

Caution You must apply all of the files contained in the hot fix at the same time. You do not need to reset the timestamps.

It is important to ensure that versions of JADE system files do not diverge from dependent binaries. Doing this ensures that dependent code files (JADE system files and libraries) are backed up and restored together. The default location of the JADE system files is the installation directory (that is, the **bin** directory for Windows and **$JADEHOME/runtime** for Linux).

When it is necessary to restore a database from backup and perform recovery, you must avoid reverting to earlier JADE system file and binary versions. When restoring the binaries directory, ensure that it is from the latest backup.

Changes in JADE Release 6.3.04

This section contains changes in JADE release 6.3.04.

For details about changes in JADE 6.3.03 (the first general release of JADE 6.3), see **RelInfo6303.pdf**, in your JADE **documentation** directory.

Highlights in Release 6.3.04

The following table summarizes the highlights in this release.

Feature	Description
Faster Classes In Use Browser	Faster display of in-use classes for large development systems
Excluding offline objects when iterating a collection	Enhance performance by easily excluding offline objects from collection iterations
Profiler cache statistics	See the size of discarded methods in a profile period
Thin client download process	Avoids unnecessary downloads via stricter file checks

Application Class *doWindowEvents* Method

In earlier releases, the value of the **waitTime** parameter of the **Application** class **doWindowEvents** method was not honored if the message queue contained many callback-type requests (for example, notifications, timers, TCP/IP completion routines, and so on). When the **app.doWindowEvents** method was called, JADE attempted to empty the Windows message queue, regardless of the wait time that was specified. This ensured that all outstanding paint requests were always processed on exit. (Paint requests are generated by Windows only when no other messages are queued.) As a result, a call to **app.doWindowEvents** may have taken considerably longer than the requested time when the queue of callback function requests was large or it was never emptied.

From this release, calling the **Application** class **doWindowEvents** method now results in the processing of queued messages associated with the Graphical User Interface (GUI) and the user interface until:

- The queue is empty *and* the specified wait time has expired (which was the case in earlier releases).
- The specified wait time has expired and another callback-type messages is encountered (the first is always processed).
  - In JADE thin client mode, all other message queue entries are processed and then the callback request is re-posted on exit from the method.
  - In standard client mode, the callback request is processed and the method is exited. In this situation, paint requests and possibly user requests remain unprocessed.

Note The reason for this difference in handling is that in JADE thin client mode, there is only ever one message posted on the presentation client for all outstanding callback-type requests. Conversely, a standard client can have many (that is, hundreds or even thousands).

Classes In Use Browser Performance

The performance of the Classes In Use Browser has been improved by making the load time of classes constant, regardless of the number of processes using the class or classes.

In addition, you can now use the **Show Process Usages** command from the Classes menu when a class is selected in the Class Browser, to view details of the processes using the selected class. For details, see "Browsing Classes that Are in Use", in Chapter 3 of the JADE *Development Environment User's Guide*.

Defining a Web Service Exposure

In JADE 6.2, Web services exposures were defined in the Web Services Description Language (WSDL) generation.

In JADE 6.3, Web service exposures must be defined by using the Web Service Exposure Wizard. (For details, see "Using the Web Service Exposure Wizard", in Chapter 16 of the JADE *Developer's Reference*.)

Disabled Text Color of a Text Box under Portable GUI

The [Jade] section of the JADE initialization file can now contain the **DisabledTextColor** parameter, which sets the disabled text color of a text box under Portable GUI.

If this parameter is present and the value is not zero (**black**), all disabled text boxes use the specified Integer value to set the color of disabled text in a text box. The default value is zero (**0**); that is, black.

This parameter is read the first time a disabled text box is encountered.

Error Messages

This section describes new error messages in this release.

3115 - GetObject request buffer too small

Exception 3115 is raised when there is a mismatch between the class definitions loaded into memory and the actual definitions in the database. It can also indicate an internal error due to character conversion issues between nodes has occurred.

If a schema has been changed at that time of the exception or shortly beforehand, the problem can be rectified by stopping and restarting the node on which it occurred. This will refresh the class definitions loaded into memory. In these circumstances, this exception is similar to the 3039 exception *(Schema to Database object size mismatch)*.

If there were no schema changes around that time, or stopping and restarting does not resolve the problem, please report this to JADE Support.

6428 - Cannot access versioned feature in current version of method

Exception 6428 is raised and a method is marked as being in error if an unversioned method is compiled in the context of the current schema and it references a versioned feature that would result in different compile code.

Excluding Offline Objects when Iterating a Collection

To specify that iterators exclude offline objects, call one of the following methods.

- Call the **Process** class **iteratorsExcludeOfflineObjects** method with the **enable** parameter set to **true**, to cause objects stored in offline partitions to be excluded from the iteration.

  This affects explicit collection iterators and **foreach** iterations over object collections executed by the process.

- Call the **Iterator** class **excludeOfflineObjects** method with the **enable** parameter set to **true**, to cause the receiver to exclude objects stored in offline partitions from the iteration.

  The exclusion state takes effect on the iteration (that is, the call to the **next** or **back** method) that follows.

JADE Initialization File

For historical reasons, when reading JADE initialization file parameter values, the code in previous releases checked the section name for a **Jade** or **Jom** prefix and ignored the prefix if the section without the prefix was present. For example, for initialization file parameters in the [JadeLog] section, if a [Log] section were present in the JADE initialization file, the [Log] section would have been used instead of the [JadeLog] section.

This check is no longer done.

Object::*hasMembers* Method

The **Object** class **hasMembers** method, which formerly could be used only with exclusive collections signified to be *delete when emptied*, is no longer condition-safe and cannot be used in constraints. However, you can now use it with any collection.

For exclusive collections that have not been populated or instantiated, this method enables you to determine if the collection is empty without having to access or lock the collection.

Partition Awareness

The Logical Certifier now checks online instances of a class only. However, it can still encounter offline instances when checking inverses.

Patch Number Batch Extract

When performing a batch extract by patch number (for single or multiple schemas), the command file is always created.

For all types of batch extract *other* than an extract by patch number, the command file is not created by default. If you want the command file to be created, you must specify this on the command line; for example:

```
jadclient path=c:\jade\system ini=c:\jade\system\jade.ini
server=singleUser schema=JadeSchema app=JadeBatchExtract endJade Single
c:\temp\myTest.scm c:\temp\myTest.ddb Test "<jcf>"
```


The **<jcf>** value indicates that you want the command file created. As you can specify more than one option between the < and > characters, you must separate each one with a comma; for example, if you want to encrypt the source and create the command file, your command would look similar to the following.

```
jadclient path=c:\jade\system ini=c:\jade\system\jade.ini
server=singleUser schema=JadeSchema app=JadeBatchExtract endJade Single
c:\temp\myTest.scm c:\temp\myTest.ddb Test "<encrypt,jcf>"
```

Profiler Cache Statistics

Profile cache statistics now include the number of methods that were discarded from the cache to make room for new methods during the profiler run. If there were discarded methods, the total size of the methods discarded is also listed.

Method cache flow statistics are written to the table in the "cache statistics" section of the profiler report when the cache size exceeds the maximum size specified and all methods in the cache were in use and could be discarded.

Relational Population Service (RPS)

The following change has been made to RPS in this release.

Process::*isUserDataPump* Method

The **Process** class now provides the **isUserDataPump** method, which returns **true** if the process is executing as a **Datapump** application on an RPS node; otherwise **false**.

Relational View

The following changes have been made to Relational Views in this release.

Controlling the Size of a Result Set Returned by the ODBC Driver

You can now control the maximum size of a result set to be returned for a call to the **SELECT** statement using the JADE ODBC driver, by specifying the required value in the new **MaximumResultSetSize** parameter in the [JadeOdbc] section of the JADE initialization file.

The default value of zero (**0**) specifies that there is no maximum limit. The value type of this parameter is **Integer64** *prefix multiplier* (for example, 16K).

A message is output to the **jommsg.log** if the result set output is truncated because of the value of this parameter.

Property Details Access

The **RelationalView** class now provides the **getColumnFeature** method, which has the following signature.

```
getColumnFeature(tableName:  String;
                 columnName: String): Feature;
```

This method returns the feature (method or property) associated with the table column specified by the **tableName** and **columnName** parameters.

Calls to this method can raise the following exceptions.

- JErr_Table_Not_Found : Relational Table not found.
- JErr_Column_Not_Found : Column not found in Relational Table.

Renaming or Editing User-Defined Table Names

The **RelationalView** class now provides the **change ColumnName** method, which has the following signature.

```
changeColumnName(tableName:     String;
                 oldColumnName: String;
                 newColumnName: String);
```

This method changes the name of a column in the table specified in the **tableName** parameter from the value specified in the **oldColumnName** parameter to the value specified in the **newColumnName** parameter.

Note This method is valid for ODBC relational views only (that is, it is not valid for RPS mappings).

Calls to this method can raise the following exceptions.

- JErr_Attribute_Name_Conflict : Attribute name null or already used as column in the selected table.
- JErr_Invalid_For_RpsMapping : May only be called for ODBC Relational Views.
- JErr_Table_Not_Found : Relational Table not found.
- JErr_Column_Not_Found : Column not found in Relational Table.
- JErr_ColumnName_Cannot_Change : Column name cannot be changed (for example, oid or index).

  The **Jerr_ColumnName_Cannot_Change** global constant is a new global constant in the **JadeOdbc** category.

Release Notes Splash Screen Display

In JADE 6.3.03, the release note splash screen, which displays the major features of the current release and hyperlinks to further information, was displayed every time a system with no user-defined schemas was started, regardless of the value of the **ShowSplashScreen** parameter in the [Jade] section of the JADE initialization file.

From this release, the release note splash screen is not displayed when a system with no user-defined schemas is started if the value of the **ShowSplashScreen** parameter in the [Jade] section of the JADE initialization file is set to **false**.

Reorganization Updates

The **Allow updates** check box on the Classes Needing Reorg dialog is disabled and unselected when the **FastBuildBTreeCollections** parameter in the [JadeReorg] section of the JADE initialization file is set to **true**.

Thin Client Download Process

The automatic JADE thin client download process has been changed so that it now determines if files are identical without relying on their modified timestamps being different.

The automatic download rules are now as follows.

- If the file is not present, the file is downloaded.
- If the file length is different, the file is downloaded.
- If the file is an executable (that is, a file of type .**dll** or .**exe**), the timestamps of when the two files were linked are compared, rather than the timestamps of when the files were modified. (This timestamp is held in the file contents.)
- If the file is not an executable and the modified timestamps of the two files do not match, an MD5 comparison of the file contents is used to determine if the two files are different.

A file download therefore occurs only when the two files are actually different, instead of the timestamps not matching.

In addition, the length of the string value of the **DownLoadVersion** and **PreDownLoadVersion** parameters in the [JadeThinClient] and [JadeAppServer] sections of the JADE initialization file has been increased from 30 characters to 60 characters.

Web Services

The Web service consumer import routine now loads and saves the **nillable= "true"** attribute.

This change may require you to reload a Web Service Description Language (WSDL) file, as the WSDL import was not previously setting this attribute.

Word Wrapping in Tables

When the values of the **autoSize** and **wordWrap** properties are set to **true**, if the column width has not been set by logic or by the user, the height of the cell is affected only if the value of:

- The **widthPercent** property for the column is greater than zero (**0**)
- **Table::autoSize** is **AutoSize_Row(1)** and the row height has not been specifically set

If neither of these applies, non-word wrap display is assumed.

If the column width has been set by logic or by the user, the height is automatically sized if the row height has not been specifically set and **Table::autoSiz**e is one of **AutoSize_Both**, **AutoSize_BothColumnMinimum**, or **AutoSize_Row**.

WSDL Nested complexType Definition Load

Classes built from nested **complexType** definitions in the WSDL schema that do not have an actual name to match on when reloading the WSDL are now saved during the import process with a name derived from the item name and parent name on which the JADE class name is based.

Note If your WSDL schema has nested **complexType** definitions, to enable JADE to save your definitions with derived names and retain your JADE mapping class names, you must import the WSDL into JADE. Subsequent reloading of the WSDL then retains the derived class names.

ZLIB Compression on Compact JADE

For ZLIB compression in single user mode for a device hosting Compact JADE, depending on the model of your Windows Mobile device, the **zlibce.dll** version 1.2.3 or higher may need to be preloaded.

For details, see the **PreLoadLibraryList** parameter in the [JadeEnvironment] section of the JADE *Initialization File Reference*.

