



J A D E TM

JADE 6 PRODUCT ROADMAP

FOR WINDOWS AND LINUX



VERSION: 1.15

DATE: 24 APRIL 2009

Feature content and schedules subject to change without notice

Microsoft
GOLD CERTIFIED
Partner

ISV/Software Solutions

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

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

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

Copyright © 2009 Jade Software Corporation Limited.

All rights reserved.

JADE is a trademark of Jade Software Corporation Limited. All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer. Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries.



CONTENTS

JADE 6.3 - APRIL 2009	4
64-BIT EDITION.....	5
DATABASE FILE PARTITIONING.....	5
UPDATE LOCK.....	6
MERGE ITERATOR.....	6
DATABASE REORGANISATION ENHANCEMENTS.....	7
.NET CLASS LIBRARY.....	7
REPLICATION FRAMEWORK.....	8
XML METADATA INTERCHANGE (XMI) SUPPORT.....	8
ODBC ENHANCEMENTS.....	9
RICH INTERNET APPLICATION (RIA) SUPPORT.....	9
RPS ENHANCEMENTS.....	10
UNIT TEST SUPPORT.....	10
CODE COVERAGE.....	11
MULTIPLE DELTAS AND PATCH VERSIONING.....	11
LANGUAGE EXTENSIONS.....	12
JADE 6.2 - RELEASED OCTOBER 2007	13
WEB SERVICE EXTENSIONS.....	13
JAVA CLASS LIBRARY.....	13
RPS ENHANCEMENTS.....	14
GENERIC MESSAGING SERVICES.....	15
.NET INTEROPERABILITY.....	15
FASTER WEB SERVICE (SOAP) MESSAGE HANDLING.....	16
PERFORMANCE AND SCALABILITY OPTIMISATIONS.....	16
SYSTEM ANALYSIS AND DIAGNOSTIC TOOLS.....	17
COMPACT JADE.....	17
ASYNCHRONOUS METHOD CALL FRAMEWORK.....	20
NEW DATA TYPES.....	20
JADE 6.1 - RELEASED NOVEMBER 2005	22
RELATIONAL POPULATION SERVICE (RPS).....	22
SCHEMA EVOLUTION.....	23
ONLINE SCHEMA UPGRADES.....	23
ENHANCED CACHE MANAGEMENT.....	24
CONCURRENCY CONTROL ENHANCEMENTS.....	25
INTERFACES.....	25
DIRECT WEB SERVICES.....	26
SDS ACTIVE REPLAY.....	26
JADE AUDIT ACCESS ENHANCEMENTS.....	27
SINGLE LISTENER/MULTIPLE WORKER CLASS.....	27

JADE 6.3 – APRIL 2009

Based significantly on customer feedback, JADE 6.3 concentrates on three main areas.

- **Scalability – more throughput and easier data management**
The native 64-Bit Edition, plus new Update Lock and Merge Iterator capabilities, give you more ways to increase the scalability of your applications. And for systems with large amounts of historical data, Database File Partitioning will enable easier data management.
- **Interoperability – more ways to connect to other systems**
The .NET Class Library and ODBC Enhancements give you more ways to connect your JADE solutions to other systems. The Replication Framework will make it easier for developers to build disconnected applications. And XMI support enables JADE schemas to be generated from *Enterprise Architect* models.
- **Development Tools**
Unit Testing and Code Coverage will help you to improve the quality of your applications, while several language extensions will enable you to write more flexible application components and diagnostic routines. We have also made it easier for you to work on multiple versions of a method within a patch version.

JADE 6.3 is supported on the following Operating Systems:

- Microsoft Windows
 - Windows 2008
 - Windows 2003
 - Windows 2000
 - Windows XP
 - Windows Vista
- Red Hat Enterprise Linux 5 (RHEL 5)
- SUSE Linux Enterprise Server 10 (SLES 10)
- Microsoft Windows Mobile (Compact JADE only)
 - Windows Mobile 5.0 for Pocket PC Phone Edition (Windows CE 5.0)
 - Windows Mobile 5.0 for Pocket PC (Windows CE 5.0)
 - Windows Mobile 6.0/6.1 Classic (Windows CE 5.02)
 - Windows Mobile 6.0/6.1 Professional (Windows CE 5.02)

Please note that Windows Mobile 6 Standard **is not** supported.

The Relational Population Service (RPS) supports SQL Server 2000, 2005 and 2008.

The Java-based Portable GUI Engine (including the Portable Thin Client) is deprecated in JADE 6.3. Our future portable Thin Client direction is based on the Silverlight technology. In the second half of 2009, we plan to provide a version of the JADE 6.3 Thin Client that will enable JADE user interfaces to be deployed via both Silverlight and Moonlight (Moonlight is an open source implementation of Silverlight for Linux).

As per our [Release Policy](#) and [Release Schedule](#), JADE 6.1.15 (the final 6.1 release) will be supported until October 2010. JADE 6.2 will be supported until October 2012.

SCALABILITY AND PERFORMANCE

64-BIT EDITION

BENEFITS

- Greater performance and scalability
- Configure significantly larger JADE caches to improve system performance

DESCRIPTION

This feature will provide enhanced performance and scalability by taking full advantage of 64-bit hardware and operating system platforms. In particular, each node will have a far greater memory address space, meaning that administrators will be able to configure significantly larger JADE caches if required, to improve system performance. Additionally, internal JADE data structures will be able to grow beyond the current 32-bit limits, providing greater capacity for large JADE systems.

A single Application Server (either 32-bit or 64-bit) will be able to support both 32-bit and 64-bit Thin Clients. A 64-bit Database Server will be able to support both 32-bit and 64-bit Application Servers and Standard Clients.

DATABASE FILE PARTITIONING

BENEFITS

- Easier management of large amounts of historical data
- Achieve cost savings through the use of tiered disk storage
- Improved performance of online and batch operations

DESCRIPTION

This capability will allow customers to more easily manage large amounts of historical data, particularly for bulk data operations such as backups and deletions. It will allow database files to be split into *partitions*, with different partitions then being able to be stored on different media devices and/or tiered disk storage.

It will be possible to *freeze* one or more partitions, thereby making them read-only. This will optimise locks for any objects resident in those partitions, as well as enabling the frozen partitions to be backed up independently of active (i.e. *thawed*) partitions. It will also be possible to mark partitions as *offline*, so that they do not need to be resident in the main online database.

As well as automatically allocating newly created objects to partitions, developers will be able to move objects between partitions programmatically. This feature will also enable developers to avoid accessing historical information if it is not required, making it easier to improve the performance of online and batch operations.

UPDATE LOCK

BENEFITS

- Increased concurrent access to collections, particularly between multiple-reader and single-writer processes
- Improved system performance

DESCRIPTION

This feature will make it easier for developers to improve system performance by providing them with a new *Update Lock*. Use of this lock will allow more concurrent access to collections, particularly in situations where an application requires greater concurrency between frequent read operations and less-frequent update operations. For applications that have such requirements (a common example is “global” collections, where there is often contention between reads and updates), this feature will provide developers with an easy way to deliver immediate performance gains.

MERGE ITERATOR

BENEFITS

- Easier access to a common view of multiple dictionaries
- Improved system performance
- Easier development

DESCRIPTION

The new *MergeIterator* class will enable developers to iterate over a merged view of two or more compatible dictionary objects. When iterating over multiple dictionaries, the merge iterator will return entries in key order, based on the set of high-order keys that are common to all of the dictionaries being merged.

By using a *MergeIterator*, in many cases, developers will be able to avoid the need for global or “catch all” persistent dictionaries that hold objects for inquiry or reporting purposes only. This will reduce update operations and help to avoid lock contention. The *MergeIterator* will also be far more efficient than developers creating their own merged transient copies of persistent dictionaries, thereby providing more opportunities to improve application performance.

DATABASE REORGANISATION ENHANCEMENTS

BENEFITS

- Faster collection population for specific schema changes
- Options to reduce disk space requirements

DESCRIPTION

With the continued growth in the size of customer JADE databases, JADE 6.3 will provide a “fast build” option for collections during database reorganisations. This option will apply to specific schema changes only, but analysis has shown that these are common changes during application upgrades.

New options will also be provided to enable system administrators to reduce the disk space required to perform database reorganisations.

INTEROPERABILITY

.NET CLASS LIBRARY

BENEFITS

- Greater interoperability between JADE and .NET
- Makes it easier for .NET developers to:
 - Use JADE to store and manipulate complex data
 - Take advantage of JADE’s automatic object caching and distributed processing

DESCRIPTION

Building on much of the R&D for the Java Class Library, the .NET Class Library will make it easier for .NET developers to interoperate with JADE. This feature will allow .NET developers to take advantage of JADE’s server-side capabilities by providing a C# framework that encapsulates the JADE Object Manager and the core JADE programming model. This will enable .NET developers to use JADE services from their .NET code (to store and retrieve JADE objects and to invoke JADE business rules, for example).

This feature will make use of the new Language Integrated Query (LINQ) facilities in .NET to allow access to JADE objects in a natural and consistent manner for .NET developers.

REPLICATION FRAMEWORK

BENEFITS

- Easier development of disconnected applications
 - Includes both Standard Client and Compact JADE applications
- Easier replication of data between multiple JADE server-side systems

DESCRIPTION

Many applications today need to provide support for disconnected usage, be it on mobile devices or just the ability for users to run a local version of application on their laptop when not connected to the server. A typical scenario is where these disconnected applications operate in a master/slave or satellite arrangement with the main server system. These systems need to be able to replicate and then synchronise objects in their local database with the main server database.

This feature will make it easier for developers to build such applications by providing a framework to support the definition, replication and resynchronisation of data from a server-side system to any number of offline clients (including Compact JADE Single User Node clients). Object-level and property-level replication will be supported, as well as mapping facilities to enable replication between different schema definitions. In addition to disconnected replication, the framework will also be useful for server-side applications that need to replicate data to other server-side JADE systems.

XML METADATA INTERCHANGE (XMI) SUPPORT

BENEFITS

- Enables JADE schemas to be generated from XMI-compliant modelling tools
- Concentrates initially on the Sparx Systems *Enterprise Architect* product

DESCRIPTION

The XML Metadata Interchange (XMI) is an Object Management Group (OMG) and international standard for exchanging metadata information via XML. The most common use of XMI is as an interchange format for UML models, although it can be used to represent the models of other languages as well.

Concentrating initially on UML models built using the Enterprise Architect product from Sparx Systems, this feature will enable JADE schemas to be generated from XMI files created by Enterprise Architect. This provides an easy way for UML models to be imported into JADE.

ODBC ENHANCEMENTS

BENEFITS

- An ODBC “Thin Client”, for faster query execution and easier deployment
- Access to application-defined “soft” entities and attributes

DESCRIPTION

The new ODBC “Thin Client” architecture will split the current ODBC interface into thin client and server components. The client component will remain an ODBC driver, but will have no object manager functionality. Its main purpose will be to provide the ODBC driver interface and pass requests to its server-side counterpart executing in a JADE node. The server component will execute the query and return the result to the client.

This will make query execution faster by eliminating the need to fetch objects across the network to end-user machines and by taking full advantage of server-side processing capacity. The installation footprint of the ODBC thin client will also be significantly smaller, making it easier to deploy to user desktops.

Many applications also allow users to define their own “soft” entities and attributes that extend the fixed data model defined by the application developers. In JADE 6.3, developers will be able to expose these user-defined entities and attributes via ODBC, as well as implement their own query provider methods that will be used by the ODBC interface to return a result set of objects.

RICH INTERNET APPLICATION (RIA) SUPPORT

BENEFITS

- Easier interoperability with AJAX-style Rich Internet Applications (RIAs)

DESCRIPTION

This feature will provide a mechanism to generate JavaScript to allow for AJAX-style calls to be made to JADE Web Services. The intent of this feature is to provide a framework and a common approach to integrating JADE Web Services with AJAX clients. The framework will make it easier for developers to invoke JADE Web Services from AJAX clients and to process the data that is returned from the Web Services, by removing much of the underlying “plumbing” code that currently needs to be written.

RPS ENHANCEMENTS

BENEFITS

- Enable a single RPS mapping to span multiple separate components
- Easier administration and reduced disk overhead by allowing a single RPS node to replicate data from multiple independent components

DESCRIPTION

In JADE 6.3, customers building modular systems using JADE's *Packages* capability will be able to create a consolidated RPS mapping across all required Packages by including imported classes in an RPS mapping in a single schema. This will make RPS administration easier by consolidating the mapping in one place and it will also reduce the disk requirements for RPS nodes as a single RPS node will now be able to replicate data from multiple Packages.

DEVELOPMENT TOOLS

UNIT TEST SUPPORT

BENEFITS

- Improve quality and reliability by writing unit tests as your system is built
- Perform extensive unit testing with minimal user intervention

DESCRIPTION

Unit Testing is a simple but effective approach to improve the quality and reliability of a system. The key idea is to develop the tests for a unit of code (which may be a fragment, a method or a module) as the code is being written. Development teams can use this approach to ensure that quality is built into a project from the beginning.

JADE 6.3 will provide an integrated framework that will enable developers to define unit test code at the same time that the application code is being written. The framework will enable the automatic execution of these tests, as well as providing the ability to browse tests and extract results. It will be possible to run tests from both the IDE and in batch mode, enabling you to schedule the regular execution of regression tests. An interface will be provided to let developers build their own unit test driver applications if they wish to do so.

CODE COVERAGE

BENEFITS

- Easily quantify how much of your code base is covered by your test suite
- Improve quality by creating tests that increase overall code coverage

DESCRIPTION

Code Coverage is a measure used in software testing to describe the degree to which the methods in a system have been tested. It is a useful measure to assure the quality of a set of tests, as opposed to directly reflecting the quality of the system under test. Code coverage can help testers and developers to:

- Discover methods and blocks of code that are not exercised by a set of tests
- Create tests that increase code coverage
- Quantify the overall code coverage of a system, which is one measure of quality

This feature will provide code coverage analysis of JADE methods; indicating those lines of code that have been executed and those lines of code that have not. It will be possible to save the results as a CSV file for further analysis.

MULTIPLE DELTAS AND PATCH VERSIONING

BENEFITS

- More flexible development by allowing two versions of a method within a single patch version
- Allow two separate versions of a method to be merged back together
- Enable method versions to be deployed and executed independently

DESCRIPTION

This feature will improve development productivity by making it easier for multiple developers to work on independent changes to common methods at the same time. This will be particularly useful in situations where an emergency fix needs to be made to a method, where changes to that method are already underway for a later release.

By enhancing the existing Delta capability, JADE will enable two versions of a method to be created within a single patch version, with developers being able to work on both versions of the method at the same time. In addition to being able to compare the differences between the two versions, JADE will allow them to be merged back together and will provide tools to assist with any conflict resolution. It will be possible to extract method versions together or separately if required. Where two versions of a method exist it will be possible to specify which version is to be executed at runtime.

JADE 6.3 will also enable the batch extract of schema definitions to separate class files, which will make it easier for development teams wanting to save their JADE source code in version control systems such as Subversion.

LANGUAGE EXTENSIONS

BENEFITS

- Easily isolate conditional code with `executeWhen`
- Easy tracking of entry and exit to methods
- Easy access to all objects created, updated or deleted in a transaction

DESCRIPTION

- **executeWhen**
This new language statement will allow developers to bracket a block of code with a Boolean expression, such that if the result of the expression is false, the block of code will not be executed. In fact, the JADE Interpreter will not even load the code into memory. This statement will make it easier for developers to isolate testing and diagnostic code, for example.
- **Method Tracking**
New methods will be provided that will allow developers to register entry and exit callback methods for any other method. The method for which the callbacks are registered is referred to as the *tracked method*. This will make it easy for developers to dynamically “track” calls to any required method in the system. The entry callback will be invoked prior to JADE calling the tracked method. The exit callback will be invoked prior to the tracked method returning. Developers will be able to access the tracked method’s parameters and return value in the callback routines. This feature will be particularly useful when writing testing and diagnostic code.
- **Transaction Trace**
In many applications there is a need to access all of the objects that have been created, updated or deleted in a transaction. Often developers want to be able to access this information from a centralised place in their code (for example, immediately before a `commitTransaction`). This feature will give developers access to all of the objects that have been modified in the current transaction and make it easy to determine what changes have been made to an object. This will make it easier to write diagnostic and auditing routines in particular, as well as code that delegates operations to background task queues.

JADE 6.2 - RELEASED OCTOBER 2007

INTEROPERABILITY

WEB SERVICE EXTENSIONS

BENEFITS

- Improved interoperability with other technologies and systems
- Support for the SOAP 1.2 standard and further compliance with WS-I (Web Services Interoperability) requirements

DESCRIPTION

Web Service standards are continuously evolving and therefore we focus our development on those areas that are of most interest to our customers. JADE 6.2 provides further support for key standards, and makes it easier for developers to expose JADE classes via Web Services. Features include:

- Support for SOAP 1.2, which is the latest version of the standard.
- Further compliance with WS-I (Web Services Interoperability) requirements. WS-I is an open industry organisation chartered to promote Web Services interoperability across different platforms and programming languages.
 - For UTF-8 encodings and styles other than RPC/Literal (which is not supported by JADE), two-way Web Services that do not require UDDI will be 92% compliant with WS-I Basic Profile 1.1 in JADE 6.2.
- More flexibility when exposing JADE classes via Web Services. Finer-grain control over class exposures makes it easier for developers to expose large numbers of classes and properties via Web Services.

JAVA CLASS LIBRARY

BENEFITS

- Greater interoperability between JADE and Java, for both Linux and Windows
- Makes it easier for Java developers to:
 - Use JADE to store and manipulate complex data
 - Take advantage of JADE's automatic object caching and distributed processing

DESCRIPTION

Java remains one of the most significant and popular programming languages, particularly for corporate and enterprise development. In 2005, Evans Data Corporation reported that in the enterprise space, 60% of organisations make use of Java in IT projects.

Java is key to many business IT infrastructures and to the technology stacks of vendors such as IBM, Sun and BEA. Java is also very strong in the Linux community, with Evans Data Corporation reporting that 80% of Java users have more confidence in Linux as a platform for mission-critical applications.

JADE 6.2 provides a Java Class Library that makes it easier for Java developers to interoperate with JADE. This feature allows Java developers to take advantage of JADE server-side capabilities (on both Linux and Windows) by providing a Java framework that encapsulates the JADE Object Manager and the core JADE programming model. This enables Java developers to use JADE services from their Java code (to store and retrieve JADE objects and to invoke JADE business rules, for example).

RPS ENHANCEMENTS

BENEFITS

- Easier interoperability with relational databases for reporting, BI/MIS and data warehousing purposes
- More flexible mapping of JADE classes to relational tables, including the ability to map derived values to table columns
 - Developers have full control of table rows before they are replicated
- Operational flexibility via an RPS Management API (including Web Services)

DESCRIPTION

Released in JADE 6.1, the Relational Population Service (RPS) allows JADE systems to automatically replicate data to relational databases (RDBMS). Microsoft SQL Server is the first RDBMS supported by RPS. Customers have indicated that they want greater control of how JADE objects are mapped to relational tables and they also want to be able to perform more operations on objects before they are replicated to the RDBMS.

JADE 6.2 gives developers more flexibility and makes it easier for them to map JADE objects to relational tables by:

- Allowing methods and virtual properties to be mapped to table columns. This allows derived values to be replicated and enable developers to write methods that aggregate multiple objects into a single table row. These methods must be implemented on a RPS-mapped class and will be invoked when the RPS node processes create, update and delete operations for instances of that class.
- Allowing developers to interrogate, change and process rows prior to them being sent to the relational database, or specify that a row not be sent at all.
- Enabling developers to specify (in the primary environment) those transactions for which deletions are **not** to be replicated to the relational database. This allows selected operations on the primary system (e.g. housekeeping and archiving tasks) to be performed without affecting relational database replicas.

Several operational enhancements are also provided, including an RPS Management API (via both JADE classes and Web Services), the ability for developers to specify customer-specific mappings and support for client-server relational database loads.

GENERIC MESSAGING SERVICES

BENEFITS

- Provides inter-process messaging services within a node
- Provides integration with IBM WebSphereMQ messaging services

DESCRIPTION

JADE has always provided external interfaces that enable JADE systems to interoperate with message-oriented middleware products such as WebSphereMQ, and several reusable schemas have been developed to support inter-system messaging.

In JADE 6.2, messaging services have been added to the core JADE platform. This feature provides a generic framework with a standardised messaging programming model, making it easier for developers to build messaging capabilities into their applications. The first release of this feature supports:

- Messaging between processes running in the same node.
- Transport interoperability. The framework allows different message transport implementations to easily be integrated. JADE 6.2 supports integration with IBM's WebSphereMQ message-oriented middleware.

.NET INTEROPERABILITY

BENEFITS

- Greater interoperability between JADE and .NET applications
- Access .NET class libraries from your JADE applications
- A new XAML control allows JADE applications to display WPF/XAML graphics

DESCRIPTION

JADE 6.2 allows JADE developers to access the growing number of .NET class libraries. This feature is similar to JADE's support for ActiveX Automation libraries. The use of .NET controls on JADE forms is supported via a similar mechanism to that used for ActiveX controls.

JADE interrogates .NET assemblies to generate JADE classes that will act as proxies for .NET objects. JADE developers can use these proxy classes as they would any other JADE class.

A new XAML control is provided that allows JADE applications running Windows-based clients to display WPF/XAML graphics.

PERFORMANCE AND SCALABILITY

FASTER WEB SERVICE (SOAP) MESSAGE HANDLING

BENEFITS

- Faster SOAP message handling for greater Web Service performance

DESCRIPTION

With the importance of Web Services as an interoperability standard, a key feature of JADE 6.2 is increased Web Service (SOAP) message handling performance. Direct support for SOAP processing will be added to the JADE Object Manager to greatly improve the performance of Web Service request messages.

PERFORMANCE AND SCALABILITY OPTIMISATIONS

BENEFITS

- Improved performance and scalability of JADE applications
- Enable developers to deliver greater performance when building applications that work with large objects (BLOBs and SLOBs)

DESCRIPTION

Performance and scalability enhancements are an ongoing activity in all JADE releases. However, in JADE 6.2, specific focus has been placed on:

- Object Manager caching algorithms
- Improved memory efficiency
- Performance enhancements in the management of large objects (BLOBs and SLOBs)

The enhanced management of large objects:

- Allows developers to store and extract large objects from the database without impacting object cache.
- Provides more efficient management of large objects for better cache utilisation to deliver higher performance. This also makes development easier by reducing the occurrence of “no memory for buffers” (1018) exceptions.
- Allows developers to share large picture binaries (BLOBs) across multiple forms in the same node, including forms hosted by different Thin Clients connected to the same Application Server node.

SYSTEM ANALYSIS AND DIAGNOSTIC TOOLS

BENEFITS

- An enhanced JADE Monitor provides more comprehensive and centralised performance information, making analysis and diagnostics easier
- Web-based access to JADE Monitor information
- Web Services enable external systems to access performance information

DESCRIPTION

This feature delivers enhanced analysis and diagnostic tools, enabling customers to more easily monitor, analyse and tune their JADE systems.

A new JADE Monitor (with both Thin Client and web-based access) enables JADE systems to be analysed more easily. It centralises information from a number of sources, including the current Monitor statistics and reports, plus thin client traffic, high-level node sampling statistics, inter-node communication statistics and low-level multi-threading performance statistics. The new JADE Monitor also allows methods to be profiled dynamically, without the need for code changes or system downtime.

Web Services are provided, enabling developers to use the monitoring capabilities from other systems (JADE or non-JADE) if required.

PLATFORM CHOICE

COMPACT JADE

BENEFITS

- The ability to run a JADE Thin Client on small devices including PDAs
- The ability to run a single-user JADE node (including the Object Manager and Database) on small devices

DESCRIPTION

There is a recognised need for customers to be able to access their JADE systems from non-traditional client devices such as PDAs. This feature delivers increased client-side platform choice by enabling Thin Clients to run on Windows Mobile devices.

This feature also delivers a specialised JADE node and specialised RootSchema class library that runs on Windows Mobile devices. In addition to GUI and Web Service consumer capabilities, this Compact JADE Node provides Object Manager and Database services and can host multiple JADE processes. Customers can build JADE applications for deployment in Windows Mobile devices, with those applications being able to store persistent data and interoperate with other systems via Web Services.

The minimum Compact JADE device requirements are shown below.

OPERATING SYSTEMS

- Pocket PC 2003 SE
- Windows Mobile 5.0 for SmartPhone (recommended for Thin Clients only)
- Windows Mobile 5.0 for Pocket PC Phone Edition
- Windows Mobile 5.0 for Pocket PC

Windows Mobile Version	Underlying Windows CE Version	Compact JADE Features
Pocket PC 2003 SE	Windows CE 4.2	<ul style="list-style-type: none"> ✓ Thin Client ✓ Single-User Node ✓ GUI Controls ✓ Web Service Consumer ✓ Serial Port Access ✓ TCP/IP Connectivity ✓ Database Recovery
Windows Mobile 5.0 for SmartPhone	Windows CE 5.0	<ul style="list-style-type: none"> ✓ Thin Client ✓ GUI Controls ✓ Serial Port Access ✓ TCP/IP Connectivity
Windows Mobile 5.0 for Pocket PC Phone Edition and Windows Mobile 5.0 for Pocket PC	Windows CE 5.0	<ul style="list-style-type: none"> ✓ Thin Client ✓ Single-User Node ✓ GUI Controls ✓ Web Service Consumer ✓ Serial Port Access ✓ TCP/IP Connectivity ✓ Database Recovery

PROCESSORS (CPU)

- For devices hosting a Compact JADE Thin Client:
 - ARM V4 400 MHz (minimum), 600 MHz (recommended), or better
- For devices hosting a Compact JADE Single-User Node:
 - ARM V4 600 MHz (minimum), or better

MEMORY

- 32 MB (minimum) of memory available to applications (this most likely means a minimum 64 MB capable device)

STORAGE

- This is additional storage space to the Memory requirements (above). Typically, this will be a separate SD or Flash card of 256 MB minimum size.
- For Thin Client applications that have been designed to suit small devices and limit the use of highly complex forms and large graphics, we expect 256 MB to be sufficient. Additional storage requirements will depend on the size of the Thin Client form cache and this is dependent largely on the size and complexity of application forms.
- For Single-User Node applications, the additional storage space must allow for the database size, software installation size, transient object storage, and any external file requirements that the application may have.
- For the best performance, selecting a fast memory card is important (e.g. the SanDisk 1 GB Extreme III CF Card), especially for Compact JADE Single-User Node applications.

OTHER REQUIREMENTS

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

LICENSING

- Compact JADE Thin Clients consume a server-side process license, in the same manner as any other Thin Client.
- Compact JADE single-user node licenses are available at no cost and it is permitted to deploy such a license on any number of devices for a given application. Compact JADE single-user node licenses limit the Compact JADE environment to a 2 GB database and 25 concurrent processes.

LANGUAGE EXTENSIONS

ASYNCHRONOUS METHOD CALL FRAMEWORK

BENEFITS

- Easier development of parallel operations

DESCRIPTION

This feature makes use of JADE 6.2's Generic Messaging Services to provide a framework that makes it easier to synchronise the execution of multiple methods running in parallel. While all of the parallel methods must initially be invoked in the same node, once invoked they can call `serverExecution` and `clientExecution` methods if required.

This capability provides an efficient and flexible alternative to code that currently must be written manually. It will be very productive for applications that need to execute several concurrent tasks, especially Web Service consumers.

NEW DATA TYPES

BENEFITS

- Easier timestamp arithmetic, string handling and integration with external code

DESCRIPTION

TimeStampInterval represents the interval between two timestamps and makes it easier to perform date and time arithmetic.

Byte represents an 8-bit integer value and is the unit of Binary attribute values. The separate Byte type makes it easier for developers to manipulate Binary values in Unicode systems and to integrate JADE code with external code.

Integer64 represents a 64-bit integer value and makes it easier for developers to interoperate with external code and 64-bit platforms.

MemoryAddress represents either a 32-bit or 64-bit memory address and makes it easier for developers to interoperate with external code and 64-bit platforms.

StringUtf8 represents one or more UTF-8 characters. UTF-8 (8-bit Unicode Transformation Format) is a variable-length character encoding for Unicode. It is able to represent any universal character in the Unicode standard, with the initial encoding of byte codes and character assignments being consistent with ASCII. UTF-8 is becoming the preferred encoding for e-mail, web pages and other situations where characters are stored or streamed. The `StringUtf8` type makes it easier for developers to build JADE systems that interoperate with Unicode applications.

GENERAL PRODUCTIVITY ENHANCEMENTS

Based on developer feedback, a number of popular enhancements have been delivered in JADE 6.2, including:

- **Load multiple schema files in one operation via the IDE**
The Load Options form has been extended to allow the selection of multiple files. If multiple files are selected, a new sheet allows the load order to be specified, as well as allowing files to be added and removed from the selection.
- **Insert a schema into the schema hierarchy**
JADE 6.2 provides a command-line utility that enables one schema to be inserted into the schema hierarchy above another schema. The utility inserts (or moves) an existing schema (that has no subschemas) to a new position in the schema hierarchy. Ideally, the schema being inserted should be an “empty” schema (that is, one containing only the Application, Global and WebSession subclasses, with no properties, constants or methods on any of these classes). The utility can insert a non-empty schema, but any entities that conflict with the schema branch into which it is being inserted will cause the operation to be cancelled.
- **More easily open a Class Browser for the top-most definition of a class**
Pressing <Shift>+<F11> over a class name opens a new Class Browser window for the class in the schema that contains the top-most definition of the class. Pressing <F11> twice over a class name continues to open a Class Browser window for the class in the current schema.
- **Be able to see a history of the most recently visited methods**
A new “History” menu item and tool bar icons in the IDE allows direct navigation to any of the last (up to) 30 methods visited.
- **Default text templates for classes, properties and constants**
In addition to method text templates, JADE 6.2 allows developers to define text templates for classes, properties and constants. This makes it easier for teams to adhere to development standards.
- **Easier drag selection of child controls in the Painter**
In JADE 6.2, <Ctrl>+<Shift>+<drag> allows developers to select controls, even if the drag operation is started over a control. Additionally, if the Painter’s “Control Move Lock” option is set (meaning that controls cannot be repositioned), then normal drag selection is allowed to start over a control.
- **Find a user more easily in the JADE Monitor**
The ability to search for a user has been provided as part of the new JADE 6.2 Monitor.
- **Programmatically invoke the Find/Replace dialog of the RTF control**
A new JadeRichText::findReplaceDialog method has been provided to allow developers to invoke the Find/Replace dialog of the JadeRichText control.
- **Analyse interpreter method cache usage more easily**
Additional logging has been provided to show the interpreter method cache swap rate and usage.

JADE 6.1 – RELEASED NOVEMBER 2005

- The Relational Population Service (RPS) is licensed separately as per the description below.
- JADE's Synchronised Database Service (SDS) continues to be licensed separately as per its introduction in JADE 6.
- For all other feature content, customers can upgrade to JADE 6.1 at no additional charge provided they are fully up-to-date with Platform Maintenance on their existing licenses.

RELATIONAL POPULATION SERVICE (RPS)

BENEFITS

- Automatically synchronised population of secondary relational databases (Microsoft SQL Server will be the first supported database)
- Makes it easy for JADE systems to interoperate with relational databases for reporting, BI/MIS and data warehousing purposes

DESCRIPTION

RPS allows JADE systems to automatically feed (that is, mirror) data into external relational databases. Configurable options for mirroring frequency are available, similar to those provided by SDS. RPS can populate an entirely independent relational database, or tables within an existing relational database where that database may be updated from multiple sources (that is, via JADE's RPS and also non-JADE systems).

New tools are available for developers to specify the classes and objects that are to be replicated to the relational database(s). This allows the entire JADE database to be replicated, or just the data that is required for external purposes. The tools also provide facilities to allow classes and properties to be mapped to the required relational tables and columns.

Note that a secondary relational database is not intended for disaster recovery (JADE's SDS provides disaster recovery capabilities).

RPS is licensed separately, similar to SDS. That is, there is a one-time RPS license fee of 20% of the value of a system's primary processes, with annual 15% Platform Maintenance and 10% Upgrade Assure being charged on this fee. Once RPS has been purchased, any subsequent primary processes added to a system will attract a 20% RPS fee.

Customers can purchase **both** SDS and RPS for a single 20% SDS/RPS license fee.

SCHEMA EVOLUTION

BENEFITS

- Enhanced source control capabilities
- Faster development
- Developers can work in multiple schema versions at the same time

DESCRIPTION

When a structural change is made to a class (for example, adding, deleting or changing a property), the class and other related structures are versioned. Among other things, JADE uses these versions to reorganise the database. For a number of common tasks, this feature improves the performance of both the versioning and reorganisation processes, resulting in faster development. Developers can also make structural changes to classes even when they are in use by running applications.

Multiple Schema Contexts are supported. A Schema Context is a snapshot (that is, a version) of a schema specification. Schema Contexts allow developers to work in multiple schema versions at the same time.

ONLINE SCHEMA UPGRADES

BENEFITS

- Improved availability in production systems through less downtime when deploying changes
- Faster database reorganisation

DESCRIPTION

When changes are deployed to a JADE system, the system is upgraded from its current definition (the *current schema version*) to the new definition (the *latest schema version*). This occurs in two main phases: an *online* phase and an *offline* phase.

This feature significantly increases the amount of work that can be performed in the online phase, greatly improving system availability; especially when large structural or method upgrades are performed.

The online phase is performed first and includes schema and method compilation, ddb file load (some changes to Control subclasses are deferred to the offline phase) and database file reorganisation.

During this phase systems are available to users, and can access and update objects defined by the current schema version without restriction.

The online database file reorganisation performs structural changes that affect individual objects only (as opposed to changes that affect relationships between objects).

These changes include:

- Adding and removing attributes
- Adding and removing embedded references
- Adding and removing exclusive collections
- Changing the type of an attribute (any combination of type-to-type conversion)
- Changing the length of string or binary attributes (includes converting embedded strings/binaries to BLOBs/SLOBs or vice versa, and truncation)
- Changing the precision or scale factor of a decimal attribute
- Moving BLOB and SLOB attributes up or down the class hierarchy
- Removing all inverses from a relationship (that is, after the change there are no inverses between the properties that were involved in the relationship)

The offline phase occurs second and performs all operations not processed during the online phase. It handles transitioning to the latest schema version, validating and reorganising inter-object relationships, and deleting the prior schema version.

The inter-object relationship reorganisation performs all structural changes that are not performed online, including:

- Adding an inverse
- Removing some, but not all, inverses from a relationship (that is, after the change inverses still exist between the properties involved)
- Condition maintenance
- Inverse definition changes, such as changing the inverse-required option
- Changes to dictionary specifications, such as key definitions or the no-duplicates constraint
- Changes to the physical attributes of a collection, such as block size and expected population
- Validation of a collection's member objects against its member type definition
- Validation of reference property type changes
- Moving exclusive collections up or down the class hierarchy

ENHANCED CACHE MANAGEMENT

BENEFITS

- Improved application performance
- Easier development by enabling shared transient transactions to be aborted

DESCRIPTION

Prior to this feature, when a JADE process is updating an object, any other process running in the same node that attempts to access the same object must wait for the first process to release it.



This feature allows processes to read an object even if another process is updating it. Reader processes no longer need to wait. This improves application performance by allowing more concurrent read operations to be performed within each node.

This feature also enables shared transient transactions to be aborted, with JADE automatically rolling back changes to any shared transient objects.

CONCURRENCY CONTROL ENHANCEMENTS

BENEFITS

- Faster concurrent read access to objects, particularly to collections
- Easier development and improved collection performance

DESCRIPTION

This feature improves the performance of applications by giving developers powerful new ways to increase the level of concurrent read access to objects, particularly for collections:

- *Fast collection snapshots* are particularly useful for quickly making a transient copy of a collection for subsequent access, thereby reducing contention on the original collection.
- *Adaptive locking enhancements* make it faster to access objects by reducing lock activity and communication with the database server. This is especially beneficial for collections. Objects can be classified according to how often they are updated: *volatile* (the default classification and the default behaviour prior to this feature), *stable* and *frozen*. Each classification improves performance when accessing the objects.

These capabilities make it easier for developers to optimise their use of collections in particular, and improve the performance of their applications.

INTERFACES

BENEFITS

- Easier development
- Improved reuse, modularisation and encapsulation
- Enables the development of more flexible systems that are easier to change through more loosely coupled designs

DESCRIPTION

An interface defines a protocol of behaviour that can be implemented by any class anywhere in the class hierarchy. An interface defines a collection of method declarations, but does not implement them. A class implements an interface by providing method implementations for each method in the interface.



Interfaces are very useful to define common behaviour among unrelated classes without forcing them to be coupled together via class inheritance. Objects of these classes can then be manipulated via the common interface. This makes it easier for developers to design more flexible and reusable systems, and to refactor existing systems.

DIRECT WEB SERVICES

BENEFITS

- Improved performance
- Fast JADE-to-JADE communication

DESCRIPTION

Web Services by default communicate through a web server using the HTTP protocol. For JADE-to-JADE communication, this feature allows Web Services to communicate directly over TCP/IP. By eliminating the need for HTTP and a web server, this significantly improves communication performance.

SDS ACTIVE REPLAY

BENEFITS

- Closer synchronisation of primary and secondary databases
- Read-only applications running in an SDS environment automatically see a more current database view

DESCRIPTION

The Block Write synchronisation mode introduced in JADE 6.0.19 writes journal blocks to secondary databases as they are generated and written to the primary database journal. This capability allows sites to minimise transaction loss in a disaster recovery takeover scenario, as all committed transactions that have been mirrored on a secondary database will be recovered.

SDS Active Replay extends the Block Write synchronisation mode to replay journal data in step with committed transactions on the primary database. This provides much closer synchronisation between the primary and secondary databases.

Read-only applications running against a secondary database automatically see a more current database view regardless of journal sizes or switching characteristics.

JADE AUDIT ACCESS ENHANCEMENTS

BENEFITS

- Provides access to SLOB and BLOB properties
- Enhanced metadata capabilities make it easier for developers to convert audit records to objects

DESCRIPTION

JADE Audit Access (JAA) was introduced in release 6.0.19, making it easier for developers to build applications that track database activity and access database audit records. JAA provides developers with an easy mechanism to access audit information recorded in database transaction journals, including:

- Embedded properties in created or updated objects, from a representative transient clone of the object
- Access to the modified properties for updated objects
- Access to additional audited control information and events such as reorganisation discontinuities, and user sign-on and sign-off events

JAA provides an object view of audited data and supports the ability for one JADE system to access the journals of another JADE system. In JADE 6.1, JAA provides even more information by making available SLOB and BLOB properties. It also makes it easier for developers to specify how journal records are to be converted into objects.

SINGLE LISTENER/MULTIPLE WORKER CLASS

BENEFITS

- Provides a robust, high-performance, native implementation of the Single Listener/Multiple Worker pattern
- Easier development of message-based TCP/IP network applications

DESCRIPTION

The new `JadeMultiWorkerTcpTransport` class makes it easier to build message-based TCP/IP network applications by:

- Encapsulating and shielding developers from low-level networking complexities.
- Reducing significantly the amount of networking code that needs to be written. Developers will find it much easier to build server-side applications that handle multiple concurrent network clients.
- Automatically handling the queuing of events. Multiple JADE worker processes (i.e. applications or threads) can share a single `JadeMultiWorkerTcpTransport` listener object. This object automatically queues events and dispatches them to idle workers through a consistent interface-based event-handling mechanism.