



JBoss Enterprise Application Platform 5

Release Notes 5.0.1

for Use with JBoss Enterprise Application Platform 5

Edition 5.0.1

JBoss Enterprise Application Platform 5 Release Notes 5.0.1

for Use with JBoss Enterprise Application Platform 5
Edition 5.0.1

Laura Bailey
lbailey@redhat.com

Legal Notice

Copyright © 2010 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

These release notes contain important information related to JBoss Enterprise Application Platform 5.0.1 that may not be currently available in the product manuals. You should read these Release Notes in their entirety before installing JBoss Enterprise Application Platform 5.0.1.

Table of Contents

1. INTRODUCTION	2
1.1. Overview	2
2. NEW FEATURES IN JBOSS ENTERPRISE APPLICATION PLATFORM 5.0.1	2
2.1. JBoss AS	2
2.2. JBoss Microcontainer	2
2.3. JBoss Messaging	2
2.4. JBoss Cache	3
2.5. JBoss Web Services	3
2.6. The Native Package	3
2.7. Native Solaris SPARC Packages	4
2.8. Database Certifications	4
3. COMPONENT VERSIONS	4
4. INSTALLATION NOTES	6
4.1. Installing JBoss Enterprise Application Platform	6
4.2. Default Startup Profile	6
4.3. Source Files	6
5. PRODUCT SUPPORT	6
6. DOCUMENTATION	6
7. ISSUES FIXED IN THIS RELEASE	7
8. KNOWN ISSUES WITH THIS RELEASE	13
9. TECHNOLOGY PREVIEWS	26
9.1. JBoss Web Services CXF	26
9.1.1. Known Issues with JBoss Web Services CXF	27
A. REVISION HISTORY	28

1. INTRODUCTION

These release notes contain important information related to JBoss Enterprise Application Platform 5.0.1. New features, issues fixed in this release, and other known issues are addressed here.

1.1. Overview

JBoss Enterprise Application Platform is the next evolutionary step in open source enterprise software. It is a powerful tool for developing rich, high performance Web 2.0 applications on a pure Java Platform.

Further, by integrating best-of-breed open source frameworks such as JBoss Seam, Hibernate, Tomcat, and JBoss Cache the Platform takes advantage of innovations in the open source community. As well, JBoss Enterprise Application Platform is fully tested and supported by Red Hat, and is certified to work on many leading enterprise hardware and software products.

2. NEW FEATURES IN JBOSS ENTERPRISE APPLICATION PLATFORM 5.0.1

JBoss Enterprise Application Platform 5.0.1 is the first bug fix release for the 5.0 release, which contained many updated and new components.

2.1. JBoss AS

The JBoss AS 5.1.x family represents the state of the art, second generation Microcontainer based enterprise Java run-time and the latest release of the world's most popular application server. In addition to supporting the latest Java EE specification (Java EE 5), it integrates many of the best enterprise class services for advanced messaging, persistence, transactions, caching and high-availability.

2.2. JBoss Microcontainer

The JBoss Microcontainer is a refactoring of the modular JBoss JMX Microkernel. It is a lightweight kernel that manages the loading, lifecycle and dependencies between POJOs. Used to integrate enterprise services, the JBoss Microcontainer together with Servlet/JSP container, EJB container, deployers and management utilities, provides a standard Java EE 5 profile.

2.3. JBoss Messaging

Hierarchical MBean Names

The version of JBoss Messaging included in JBoss Enterprise Application Platform 4.3 allowed users to declare message queues with hierarchical names such as **app1/emails**, where the forward slash would create a nested JNDI context of **queue/app1.emails**. This behavior prevents name clashes between applications with similarly-named queues:

```
<mbean code="..." name="...,name=emails">
<attribute name="JNDIName">app1/emails</attribute>

<mbean code="..." name="...,name=emails">
<attribute name="JNDIName">app2/emails</attribute>
```

This behavior is now included in JBoss Enterprise Application Platform 5.0.1. However, the syntax has been altered to allow users to exploit wildcard routing in future versions of JBoss Messaging. The point "." should be used instead of the forward slash, "/", like so:

```
<mbean code="..." name="...,name=app1.emails">
<attribute name="JNDIName">app1.emails</attribute>

<mbean code="..." name="...,name=app2.emails">
<attribute name="JNDIName">app2.emails</attribute>
```

2.4. JBoss Cache

JBoss Cache is used to replicate EJB and HTTP session state and supports distributed entity caching for JPA, while continuing to push the performance and scalability envelope with a new and more efficient locking scheme (MVCC – MultiVersion Concurrency Control).

2.5. JBoss Web Services

JBoss Web Services is a framework that supports the latest JAX-WS specification and a pluggable architecture to provide choice of Web Services Stack.

2.6. The Native Package

The Native package is an optional component for the JBoss Enterprise Application Platform and incorporates JBoss Native, `mod_jk`, `mod_cluster` and ISAPI for Windows technologies. These technologies are described as follows:

- JBoss Native consists of the Apache Portable Runtime (APR), OpenSSL and Tomcat Native (TC-native).
 - **Apache Portable Runtime (APR)** provides superior scalability, performance, and improved integration with native server technologies. The Apache Portable Runtime is a highly portable library that is at the heart of Apache HTTP Server 2.x and contains many uses, including access to advanced IO functionality (for example: `sendfile`, `epoll` and OpenSSL), Operating System level functionality (for example: random number generation and system status), and native process handling (shared memory, NT pipes and Unix sockets).
 - **OpenSSL** implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols and includes a basic cryptographic library.
 - **Tomcat Native (TC-Native)** is a Java Native Interface (JNI) that provides much of Tomcat's core functionality in native code rather than Java. This allows for an overall increase in the speed of a server.
- **mod_jk** is a connector that is used to connect the Tomcat JSP container and different web services such as Apache.
- **mod_cluster** is an `httpd`-based load balancer. Like **mod_jk**, it uses a communication channel to forward requests from `httpd` to an application server node. See the `mod_cluster` documentation available from http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/ for installation instructions.
- **ISAPI** is a connector used to connect the Microsoft IIS web server to JBoss Enterprise Application Platform.

**NOTE**

The Red Hat Enterprise Linux 5 distribution of the Native package does not contain OpenSSL or the Apache Portable Runtime since these technologies are already included in the Operating System.

2.7. Native Solaris SPARC Packages

Native Solaris SPARC packages are provided from this release of JBoss Enterprise Application Platform onward. See the JBoss Enterprise Application Platform *Installation Guide* for native package installation instructions.

2.8. Database Certifications

JBoss Enterprise Application Platform 5.0.1 is now certified for the Oracle 11g R2 and Oracle 11g RAC databases when used with Oracle JDBC Driver version 11.2.0.1.0.

It is also certified for Sybase ASE 15.0.3 when used with JConnect driver version 6.0.5.

3. COMPONENT VERSIONS

This section details the versions of the components which create the Enterprise Application Platform 5.0.1.

Table 1. Component Version Differences

Component	EAP 5.0.1	EAP 5.0.0
JBoss Application Server	5.1.0.GA	5.1.0.GA
JBoss Microcontainer	2.0.9.GA	2.0.9.GA
JBoss Native	2.0.9.GA	2.0.6.GA
Hibernate Core	3.3.2.GA_CP01	3.3.2.GA
Hibernate Entity Manager	3.4.0.GA_CP01	3.4.0.GA
Hibernate Annotations	3.4.0.GA_CP01	3.4.0.GA
Hibernate Search	3.1.1.GA_CP01	3.1.1.GA
Hibernate Validator	3.1.0.GA	3.1.0.GA
Clustered Remote EJB3 Proxies	1.1.18.GA	1.1.18.GA
JBoss Cache	3.2.1.GA	3.2.1.GA
JBoss HA Server API	1.1.1.GA	1.1.1.GA

Component	EAP 5.0.1	EAP 5.0.0
JBoss JAXR	2.0.1	2.0.1
RESTEasy	1.1.GA_CP01	1.1.CP01
JGroups	2.6.13.GA	2.6.13.GA
JBoss EJB3	1.1.18	1.1.18
JBoss JTA	4.6.1.GA_CP03	4.6.1.GA_CP03
JBoss JTS	4.6.1.GA_CP03	4.6.1.GA_CP03
JBoss Negotiation	2.0.3.SP1	2.0.3.SP1
JBoss Managed	2.1.1.GA	2.1.1.GA
JBoss Messaging	1.4.6.GA	1.4.6.GA
JBoss Metadata	1.0.2.GA	1.0.1.SP1
JBoss Web	2.1.7.GA	2.1.3.GA
JBoss Web Services - Native	3.1.2.SP3	3.1.2.SP3
JBoss AOP	2.1.6.GA	2.1.6.GA
JBoss Remoting	2.5.2	2.5.2
JBoss Serialization	1.0.3.GA	1.0.3.GA
JBoss XB	2.0.1.GA	2.0.1.GA
JavaServer Faces	1.2_13	1.2_13
JacORB	2.3.1.jboss.patch01	2.3.1.jboss.patch01
JPA	1.0.0	1.0.0
JBoss Security	2.0.4.SP1	2.0.4.SP1
JBoss Profiler-jvmti	1.0.0.CR5	1.0.0.CR5
Seam	2.2.1.EAP5	2.2.0.EAP5
RichFaces (in Seam)	3.3.1.GA	3.3.1.GA

Component	EAP 5.0.1	EAP 5.0.0
JON JASA Console	1.3.2.GA	1.3.2.GA
mod_jk	1.2.28	1.2.28
mod_cluster	1.0.3.GA	1.0.2.GA

**NOTE**

JBoss Web Services CXF 3.1.2.SP2 is also provided as a Technology Preview.

4. INSTALLATION NOTES

4.1. Installing JBoss Enterprise Application Platform

You must have adequate disk space to install JDK and JBoss Enterprise Application Platform while also allowing enough space for your applications. You must have a working installation of JDK 1.6. For the latest information on supported Operating System / JVM combinations, supported Database platforms and current information on the revision level of included components, refer to the installation guide available online from http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/. The installation guide also details instructions to install and verify the installation of the JBoss Enterprise Application Platform.

4.2. Default Startup Profile

The default startup profile is **default** which is a base Java EE 5 server profile containing a default set of services. It contains the most frequently used services required to deploy a Java EE 5 application. It does not include the JAXR service, the IIOp service, or any of the clustering services.

The **default** profile is not intended for production use or for the running of load, stress, availability or performance tests.

4.3. Source Files

Source ZIP

- <ftp://ftp.redhat.com/pub/redhat/jbeap/5.0.1/en/source/jboss-eap-src-5.0.1.zip>

5. PRODUCT SUPPORT

Bugs, potential bugs, and development issues and/or questions should be filed as JBoss support cases via the JBoss Customer Support Portal.

6. DOCUMENTATION

Refer to the index.html file in the documentation directory for a list of included documentation. In this release all API documents, code examples and a link to the online release notes are included with the distribution; all other guides and documentation are available online at http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/.

In the ZIP, the included distribution for the Platform and its individual components can be found in a separate ZIP file, **jboss-eap-docs-<version>.zip**.

Amongst the online documentation you will find the following important guides:

- **Installation Guide** explains how to install and verify the installation of JBoss Enterprise Application Platform using different installation modes.
- **Getting Started** details the directory structure of the platform and a quick tour of the Application Server and different configuration sets and services.
- **Administration and Configuration Guide** explains all administrative and server configuration functions in detail.

The online documentation will be updated as necessary so be sure to check back, especially when a new version of the JBoss Enterprise Application Platform is released.

7. ISSUES FIXED IN THIS RELEASE

Following is a list of issues fixed in this release:

Security Issues

- **JBPAPP-3952**: A security issue in the JMX Console configuration has been identified that allows an attacker to bypass security authentication.

The JMX Console configuration only specified an authentication requirement for requests that used the GET and POST HTTP "verbs". An attacker could create a HTTP request that did not specify GET or POST and it would be executed by the default GET handler without authentication. This release contains a JMX Console with an updated configuration that no longer specifies the HTTP verbs. This means that the authentication requirement is applied to all requests.

For additional information on this vulnerability refer to: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0738>

All users are advised to upgrade to this release to resolve this issue.

If an immediate upgrade is not possible or the server deployment has been customized then the fix can be applied by editing the deployment descriptor (**WEB-INF/web.xml**) of the JMX Console WAR. Details of how to apply this fix can be found at <http://kbase.redhat.com/faq/docs/DOC-30741>. Contact Red Hat JBoss Support for advice before making these changes.

Red Hat would like to thank Stefano di Paola and Giorgio Fedon of Minded Security for responsibly reporting the CVE-2010-0738 issue.

- **CVE-2009-3555**: A vulnerability in the TLS protocol allowed an attacker to inject arbitrary requests into a TLS stream during renegotiation. The JBoss Web blocking IO (BIO) connector uses the JSSE implementation of TLS provided by the JVM; therefore, the BIO connector is vulnerable because the JSSE version used is vulnerable. Until a fix is available in JSSE, a new connector attribute, **allowUnsafeLegacyRenegotiation** has been added to the BIO connector to work around this issue. It should be set to **false** (the default) to protect against this vulnerability. Users should be aware that the impact of disabling renegotiation will vary with both application and client. In some circumstances disabling renegotiation may result in some clients being unable to access the application.

- [JBPAPP-3079](#): Session expiration did not trigger flushing of the JBoss Authentication Cache. The **PrincipalSessionAttributeFilter** has been created in order to place the principal as an attribute of the HTTP session. This attribute is checked when the session expires and, if found, triggers flushing of the authenticated cache. You must uncolmment this filter in Tomcat's **web.xml** to use this feature.
- [JBPAPP-2873](#): Twiddle logged all command line arguments, including the JMX password, to **twiddle.log**. This log file is publicly readable and is created in the current directory. The password argument is now masked in the log file.

JBoss Application Server

- [JBPAPP-3430](#): Undefined behavior occurred on remote clients that used **NestedTransaction** when nested transactions were disabled in **jbossjta-properties.xml**. No nested transaction checking was performed, despite this being unsupported. This update adds a **NotSupportedException** to be thrown when clients attempt to start a nested transaction.
- [JBPAPP-3328](#): Farming's **AddContentStreamAction** attempted to close this **InputStream** as part of cleanup processing, even though it was not responsible for opening the stream. This caused failures in **ClusteredDeploymentRepoAddContentTestCase**, which is responsible for the stream. **AddContentStreamAction** no longer attempts to close the input stream.
- [JBPAPP-3326](#): **ClusteredDeploymentRepository** failed when an exploded deployment was removed because the logic that iterated over the contents of the deployment removed items incorrectly. This meant that when an exploded deployment was placed in the **farm** directory and later removed, a **ConcurrentModificationException** occurred and **ClusteredDeploymentRepository** failed. Items are now removed correctly via **iterator.remove()**.
- [JBPAPP-3234](#): Setting **HDScanner**'s **scanEnabled** attribute to **true** via XML would not disable scanning if set to **false**, and caused a **NullPointerException** if set to **true**. Both issues have been resolved.
- [JBPAPP-3213](#): Deploying EJB3 methods with zero parameters led to **NullPointerExceptions**. This fix ensures that the deployment will not fail on these grounds.
- [JBPAPP-3180](#): Hibernate integration code for unsupported second-level caches and connection pools were not included in JBoss Enterprise Application Platform 5.0. The following JARs have been included in **common/lib** to provide integration for this module:
 - **hibernate-ehcache.jar**
 - **hibernate-oscache.jar**
 - **hibernate-swarmcache.jar**
 - **hibernate-c3p0.jar**
 - **hibernate-proxool.jar**
- [JBPAPP-3029](#): The **jboss_init_redhat.sh** script is used to start and stop a server instance under a given user name. When using a non-loopback bind address, calling **jboss_init_redhat.sh stop** resulted in a **CommunicationException** because of a missing hostname parameter for the remote server the script attempts to contact.

- [JBPAPP-2866](#): The JGroups protocol stack included an incorrect diagnostic address, `224.0.0.75`. The address has been corrected to `224.0.75.75`.
- [JBPAPP-2818](#): The `main/src/bin/run.sh` did not allow users to override `$JBOSS_HOME/bin/run.conf` with a profile-specific `$JBOSS_HOME/server/$PROFILE/run.conf`. This update allows the use of a custom `run.conf`, if specified.

JBoss Web

- [JBPAPP-3220](#): When cookies were disabled for the current context, a session cookie from the parent context overwrote the session ID encoded in a URL. The fix for this issue specifies that when cookies are disabled for the current context, the parent context's session cookie should not be sought, and prevents the session ID in the URL from being overwritten.
- [JBPAPP-2929](#): With buddy replication, when multiple concurrent requests are made with the same session ID after failover, the requests may abort with an `org.jboss.cache.lock.UpgradeException` while attempting to migrate the cache data to the local node. This no longer occurs, and multiple concurrent web requests made after failover with buddy replication enabled now works correctly.

JBoss Seam

- [JBPAPP-3954](#): When a Seam `ManagedDrivenBean` component calls a stateless session bean component in a Seam-managed persistence context, an `IllegalStateException` ("No event context active") may occur. The component now checks if `ContextEvent` is active.
- [JBPAPP-3541](#): Seam could not be compiled from source because its `root.pom.xml` referenced an incorrect version of `javax.transaction:jta:jar`. The JAR referenced has been corrected to the correct version `javax.transaction:jta 1.1`.
- [JBPAPP-3380](#): `jboss-seam-resteasy.jar` was not included in the Seam distribution in JBoss Enterprise Application Platform 5.0. This JAR, and relevant documentation, have been added.
- [JBPAPP-3334](#): The `base` variable in `org.jboss.seam.bpm.JbpmELResolver` was passed into `resolveVariable` instead of the `property` variable. This meant that the method returned null where it should have returned the task instance. `property` is now passed correctly.
- [JBPAPP-3292](#): `com.sun.faces.config.ConfigureListener` was missing from `web.xml`. This meant that JavaServer Faces was not initialized when Seam bootstrapped its application scope components, so the JavaServer Faces application context was not available. This class has been added to `web.xml` and JavaServer Faces now initializes correctly.
- [JBPAPP-3048](#): The Seam booking example and its derivatives contained outdated page footers. These have been updated for Seam 2.2.
- [JBPAPP-3001](#): Bash script `seam/seam.sh` has executable permission only on some Linux systems. This is caused by a different zip util implementation included in the distribution. This has been fixed on Fedora 12 and Red Hat Enterprise Linux 4 and executable permissions are now assigned to `seam/seam.sh` correctly. The fix is not available for the zip util used on other operating systems such as Ubuntu.

- [JBAPP-2733](#): When the Seam examples were tested with the TestNG plugin in JBDS, a `java.lang.AssertionError` was thrown. To avoid this error it is important to test the examples according to the following instructions:
 1. From the example's home directory (e.g. **booking** for the booking example), run **ant test**.
 2. In Eclipse, click on **File > New > Project . . .**
 3. Select **Java Project from Existing Ant Buildfile** from the New Project Wizard, and click **Next**.
 4. Select the example's **build.xml** file as the base for the new Java project.
 5. Select **Testing Suite** or **Testing Class**.
 6. From the **Run As** menu, choose **TestNG Test**. You can cancel the processing of the test run at any time.
 7. Go to **Run > Run configurations** and edit the created TestNG runner.
 8. If JDK 1.6 is used as runtime, add the following JVM argument on the **Arguments** tab:

```
-Dsun.lang.ClassLoader.allowArraySyntax=true
```
 9. Go to the **Classpath** tab and remove all **User** entries.
 10. Add the JARs and folders specified by <http://seamframework.org/Community/GettingStartedDevelopingTheSeamFramework#H-RunningIntegrationTestsFromTheTestNGEclipsePlugin>.

JBoss Hibernate

- [JBAPP-3384](#): Hibernate collection mapping encountered exceptions if **@MapKey** was used without an explicit **@Type** annotation. Without an explicit **@Type** annotation, Hibernate assumed that the property key type was **Serializable** and attempted to deserialize an object stream from the database column value. With this update, if **@MapKey** is not given an explicit **@Type**, Hibernate uses the original property type instead of the serializable type.
- [JBAPP-3371](#): The **round** function is meant to return values of the same type as the first argument provided (integer, double, or decimal). Previously, it rounded all values regardless of type. All values should now return as the correct type.
- [JBAPP-3191](#): The **hibernate-ehcache.jar** was missing from JBoss Enterprise Application Platform 5.0. This meant that applications that used ehcache as the Hibernate second-level cache provider failed with a **NoClassFoundException**. A signed version of **hibernate-ehcache.jar** is available from CSP: <https://support.redhat.com/jbossnetwork/restricted/softwareDetail.html?softwareId=1037>. This JAR should be placed into the following directories:
 - `$JBOSS_HOME/server/all/lib`
 - `$JBOSS_HOME/server/production/lib`
 - `$JBOSS_HOME/server/web/lib`

- [JBPAPP-3173](#): Using Javassist as the bytecode provider to instrument your domain model caused errors if an entity extended a parent class with an abstract method. Hibernate code used **return** instead of **continue** in a **while** statement, which caused the statement to skip all other attributes that should have been used. This has been corrected.
- [JBPAPP-3115](#): The Hibernate Javadoc referenced the wrong version of the JDK objects. These references have been updated to <http://java.sun.com/j2se/1.5.0/docs/api/>.
- [JBPAPP-3098](#): When a filter with a collection type parameter was used, and the number of parameters in that collection changed during the lifetime of the **SessionFactory**, the SQL would not be updated to reflect the change in the number of parameters. This typically resulted in the following error:

```
java.sql.SQLException: Parameter index out of bounds.
    2 is not between valid values of 1 and 1
```

This occurred only with HQL, not Criteria, and has now been corrected.

- [JBPAPP-3089](#): A long IN list could result in stack overflow during parsing. A query element like **where x in (:x)** or a manually-constructed **where x in (1,2,3,...)** could generate a stack overflow if the number of elements referenced by **x** exceeded a number dependent on available stack space. For Java Virtual Machines, the limit is between 9000 and 10000, assuming a relatively empty stack at the point of query execution.

The stack overflow occurred in **org.hibernate.hql.ast.util.NodeTraverser** because it used a recursive algorithm to walk a parse tree. A long IN list generated a very deep sub-tree, so a sufficiently long list caused the stack overflow when **NodeTraverser**'s internal method **visitDepthFirst** calls itself too many times. This recursive algorithm has been replaced with an iterative tree-walking implementation to fix this issue.

- [JBPAPP-2957](#): The **evictAll()** method in **EntityRegionAccessStrategy** and **CollectionRegionAccessStrategy** should remove objects from the cache immediately, without regard for transaction isolation. The Hibernate/JBoss Cache integration did not handle this correctly, as the JBoss Cache **removeNode** calls it made did not deal with transactional issues. This usually results in a **IllegalStateException** or a JBoss Cache **CacheException** when a transaction that had made a bulk update was committed, or when using the Hibernate **SessionFactory** evict methods.

To fix this issue, any ongoing transaction in **evictAll()** will now be suspended before invoking JBoss Cache's **removeNode**. To cater for transactional issues, state is now stored in the integration layer's Region to track where eviction has occurred but may not yet be reflected in JBoss Cache. JBoss Cache is used as a notification bus to propagate the eviction to other nodes. Eviction occurs locally, and fails immediately where lock conflicts occur. State is also checked in the **get()** and **putFromLoad()** methods.

- [JBPAPP-2922](#): Hibernate warns that the cglib BytecodeProvider impl is considered deprecated and is not recommended for use. cglib is not deprecated, so this warning can be safely ignored.
- [JBPAPP-2900](#): MySQL uses the **TEMPORARY** keyword to bypass implicit transaction commits. Previously, Hibernate used **<CREATE TEMPORARY TABLE>** with **<DROP TABLE>**. Omitting the **TEMPORARY** keyword caused an implicit commit, and immediate failure within an XA Transaction. **<DROP TEMPORARY TABLE>** is now supported and this issue no longer presents.
- [JBPAPP-2892](#): When Enterprise JavaBean 3.0 entities were used with optimistic caching, **org.jboss.ejb3.entity.OptimisticJBCCache.DataVersionAdapter.newerThan**

incorrectly returned `true` for `A.newerThan (A)`. This caused a `DataVersioningException` when JBoss Cache attempted to remove the entry. The method has been corrected so that it returns `false`. Note that the recommended approach is to use Multiversion Concurrency Control (`mvcc-entity`) instead of optimistic caching.

- [JBPAPP-2858](#): Native queries were automatically paginated in `getSingleResult()`, which caused `getSingleResult()` to fail for some databases and queries. This behaviour has been changed so that Hibernate no longer alters `setMaxResult` for native queries in `getSingleResult()`.
- [JBPAPP-2277](#): Hibernate uses `ClassLoader.loadClass()` was used in `SerializationHelper$CustomObjectInputStream`, but is no longer supported by default as of JDK 6 (see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6500212 for further information). Attempting to load an array using this method resulted in a `ClassNotFoundException`. `SerializationHelper$CustomObjectInputStream` now uses `Class.forName(className, false, myClassLoader)` to resolve classes.
- [JBPAPP-2082](#): Associations marked as mappedBy must not define database mappings like `@JoinTable` or `@JoinColumn`. This fix adds an `AnnotationsException`, which is thrown when Hibernate receives this invalid mapping.
- [JBPAPP-1998](#): `EntityNotFoundException` is incorrectly thrown upon an optimistic locking failure when one `EntityManager` tries to delete an entity that has been updated by a different `EntityManager` and `hibernate.jdbc.batch_versioned_data` is set to `false` (the default value). `OptimisticLockException` is now thrown instead.
- [JBPAPP-1547](#): `org.hibernate.dialect.SybaseASE15Dialect.areStringComparisonsCaseInsensitive()` now returns `true`. This was done because, by default, Sybase ASE 15 string comparisons are case-insensitive. Since Sybase can be configured to be either case-sensitive or case-insensitive, if the Sybase database is configured for case-sensitive comparisons, the previous setting (`false`) was incorrect.

Documentation

- [JBPAPP-3380](#): RESTEasy integration information has been added to the *Seam Reference Guide*.
- [JBPAPP-3863](#): The *Administration and Configuration Guide* indicated that the JDBC `blocking-timeout-millis` property's default value is 5000 milliseconds. This incorrect value has been replaced with the true default value, 30000 milliseconds.
- [JBPAPP-2948](#): The `deploy/jmx-remoting.sar` service instantiates a JSR-160 adapter for standardized remote access to the JBoss MBeanServer. This service is used with tools such as the JConsole. At present, this service does not support secure access. In production environments where the server binds to a specific address other than `localhost` this presents a potential security risk, so the adapter has been moved from the `deploy` directory into `docs/examples/jmx`. We do not recommend enabling it for production usage. If during development you wish to re-enable the adapter, copy it back to the `deploy` directory.

The adapter has been moved to `/docs/examples`. If you wish to re-enable it, move it back to the `deploy` directory.

- [JBPAPP-2802](#): The JBoss Cache documentation did not indicate that Non-Blocking State Transfer was unsupported. Unsupported information about Non-Blocking State Transfer has

now been removed from the JBoss Cache documentation associated with JBoss Enterprise Application Platform.

8. KNOWN ISSUES WITH THIS RELEASE

Following is a list of known issues at the time of release.

General Known Issues

- **JBPAPP-3929**: When `java.sql.Date.valueOf` attempts to parse dates of the format yyyy-mm-dd, the TCK test threw a `java.lang.IllegalArgumentException`. This was due to a regression in the latest Sun JVM, Sun JDK 1.6.0_18-b07 (see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6898593 for more information). The workaround for this issue is to downgrade to Sun JDK 1.6.0_17-b04.
- **JBPAPP-3766**: Each time a `org.jboss.mail.MailService` is configured and bound to JNDI, the `SessionObjectFactories` properties are overwritten. This means that when multiple `MailService` configurations exist, all `MailServices` bound to JNDI will have the properties of the most recently configured `MailService` in their Session.
- **JBPAPP-3171**: **JBPAPP-544** fixed a security issue by removing the mapping for `/status`, leaving only the secure `/web-console/status` available. However, the unsecured `/status` was reincluded in **JBPAPP-1146**. The workaround for this issue is to remove the `/status` mapping from `ROOT.war/web.xml` as described in the JIRA.
- **JBPAPP-3133**: A bug in JVM implementations means that the JVM may crash when an incomplete deployment is overwritten by a new version of the deployment during hot deployment. For example, a deployment may contain a formatting error in a deployment descriptor. When deployed, the error is encountered and the deployment fails. If a corrected deployment is copied over the top of the erroneous deployment to replace it, the JVM may crash during hot deployment of the new version. The workaround for this issue is to remove the old, incompletely deployed archive before copying the new corrected deployment into the `deploy` directory.
- **JBPAPP-3036**: The `jboss_init_hpux` script does not pick up environment variables when executed in the GNU bash shell. This is related to **JBPAPP-2036**: <https://jira.jboss.org/jira/browse/JBPAPP-2306>.
- **JBPAPP-3035**: The `-e` and `-H` arguments of the `shutdown.sh` script cannot be used to directly terminate the JVM.
- **JBPAPP-2998**: When the server is started via a desktop icon, the machine's default Java set is used. This can cause exceptions when a Java version other than JDK 1.6 is set as the default. Users should ensure that JDK 1.6 is set as the default Java version before attempting to start JBoss Enterprise Application Platform via the desktop icon.
- **JBPAPP-2571**: Running the Microsoft SQL server with Microsoft JDBC drivers 2.0 causes unstable builds in JBoss Messaging. This issue is created by the **Adaptive Buffering** being the default behavior for the driver as this only allows for large values to be read once by using the `get<Type>Stream` method. The current workaround for this issue is to change the `responseBuffering` parameter of the JDBC connection URL from being `adaptive`, to being `full` (as it was for version 1.2 of the JDBC driver).

```
<url>jdbc:sqlserver://[host];database=
[database];responseBuffering=full;</url>
```

- [JBAS-7049](#): The server manager does not function correctly when Open JDK 6 is used because a `NullPointerException` check is missing in Open JDK 6. The workaround is to comment out the `java.security.debug` statement in the `imports/server-config.xml` file.
- [JBPAPP-2598](#): Once the workaround for the [JBAS-7049](#) issue is applied a new issue presents itself. A server running the security manager using Open JDK 6 still fails to start, now with an access denied error. There is no currently known workaround for this issue.
- [JBPAPP-2590](#): An issue exists in the `policy.provider` defined in `${JAVA_HOME}/jre/lib/security/java.security` when IBM JDK 6 is used. By default `org.apache.harmony.security.fortress.DefaultPolicy` is used and this should be `policy.provider=sun.security.provider.PolicyFile`. The workaround for this issue is to manually make this adjustment.
- [JBPAPP-2576](#): The MySQL JDBC driver does not currently implement XA Recovery correctly.
- [JBPAPP-2871](#): High CPU utilization and reduced performance and transaction throughput has been observed when MySQL 5.0.41 when optimized settings are in use, as described in [JBQA-2610](#). We recommend upgrading to MySQL 5.0.86 and applying optimized settings as described in [JBQA-2610](#) to reduce CPU utilization and increase performance.
- [JBPAPP-2162](#): Sun JAXB silently accepts messages with non-fatal errors where they should be rejected. The fix for [JBPAPP-2114](#) corrects this, so that badly-formed messages are rejected.
- [JBPAPP-2765](#): `LoadMgr3` logged a failure to load a class as an error even when the failure to load was expected and intentional. (Seam, for example, catches exceptions in order to disable unnecessary components if a particular class is not found.) Failure to load a class is now logged as a warning instead of an error.
- [JBPAPP-2894](#): Setting the `hibernate.bytecode.provider` system property in `jpa-deployers-jboss-beans.xml` is unreliable. The workaround for this issue is to add `-Dhibernate.bytecode.provider=cglib` to `$JAVA_OPTS` in `jboss-as/bin/run.conf`.
- [JBPAPP-2713](#): `org.jboss.test.xml.DDValidatorUnitTestCase` constantly fails and crashes the Java Virtual Machine. The workaround for this issue is to disable the JIT compiler by setting `JAVA_COMPILER=NONE` or using the command line switch `-Djava.compiler=NONE`.
- [JBPAPP-1774](#): Red Hat Enterprise Linux 5 users installing the JBoss Enterprise Application Platform rpm using `yum` must be subscribed to the Red Hat Enterprise Linux Extras channel in order to satisfy the Java requirement of JBoss Enterprise Application Platform with either the Sun or the IBM Java Development Kit.

OpenJDK is included in the Red Hat Enterprise Linux 5 base channel, but should not be used to install JBoss Enterprise Application Platform. Users who wish to use OpenJDK should first install JBoss Enterprise Application Platform with the Sun or IBM JDK:

```
yum install jbossas
```

Then use **alternatives** for **java** and **javac**:

```
alternatives --config java
```

```
alternatives --config javac
```

Switch to OpenJDK by selecting it from the list presented by **alternatives**.

- [JBAS-6966](#): The IBM distribution of JDK 6 does not support the **SSLv2Hello** protocol and generates a `ERROR [AbstractKernelController]` when used. It is currently recommended not to use this protocol.

Hibernate Known Issues

- [JBPAPP-4175](#): When Hibernate executes a cacheable query using a **ResultTransformer**, it will attempt to cache the results after applying the **ResultTransformer**. However, the data may be modified so that Hibernate cannot read it. In this case, a **ClassCastException** will occur when attempting to cache the results.
- [JBPAPP-4123](#): PostgreSQL fails to drop **SchemaExport** if constraint names change.
- [JBPAPP-4105](#): Hibernate uses **StandardDialectResolver** to guess the Hibernate dialect if the `hibernate.dialect` property is not found in the Hibernate properties file, `hibernate.cfg.xml`, or `persistence.xml`. **StandardDialectResolver** returns **SybaseDialect** if the database name it extracts from database metadata is **Sybase SQL Server** or **Adaptive Server Enterprise**. However, because **SybaseDialect** has been deprecated, this can cause some problems. To work around this issue, set the `hibernate.dialect` property to the correct dialect.
- [JBPAPP-4095](#): When scrolling a HQL query with join fetch on a collection, **FetchingScrollableResultsImpl.last()** does not move to the last result if the cursor is positioned after the last result. Instead, the cursor stays in position after the last result. This can cause `org.hibernate.exception.GenericJDBCException: could not perform sequential read of results`.
- [JBPAPP-4088](#): When a column name is defined in backticks (```), **@JoinTable** and **@JoinColumn** mappings fail with a **AnnotationException** when the column name with backticks is referenced. The following fails:

```
@JoinTable(name = "SYS_GROUPS_USERS", joinColumns = @JoinColumn(name = "USERID", referencedColumnName = "`uid`"), inverseJoinColumns = @JoinColumn(name = "GROUPEID", referencedColumnName = "GROUPEID"))
```

Whereas the following succeeds:

```
@ManyToMany(fetch = FetchType.LAZY)
@JoinTable(name = "SYS_GROUPS_USERS", joinColumns = @JoinColumn(name = "USERID"), inverseJoinColumns = @JoinColumn(name = "GROUPEID"))
```

The workaround for this issue is not to specify the **referencedColumnName** where it refers to the primary key, since Hibernate will choose the primary key automatically.

- [JBPAPP-4022](#): **getColumnSpan** returns **0** when comparing entities with **OneToOne** mappings, even when comparing the same entity, which results in a **TypeMismatchException**. To work around this issue, change the mapping to **ManyToOne**.
- [JBPAPP-3946](#): **LikeExpression** does not handle the **ignoreCase** flag properly when **ignoreCase** is set to false. It builds the correct SQL **property like ?**, but does not use the flag inside **getTypedValues**. A lowercase value is always produced.

- [JBPAPP-3913](#): In Oracle 11g R2 (both RAC and standalone), a sequence may start at 2 instead of 1. The root cause is still under analysis.
- [JBPAPP-3911](#): In Oracle 11g R2 (both RAC and standalone), a complex query with **LockMode.UPGRADE** (that is, "for update") may cause a "No more data to read from socket" error. The workaround for this issue is to avoid using **LockMode.UPGRADE** on such queries. The root cause analysis is still under investigation.
- [JBPAPP-3737](#): In a stateless session, queries load objects in a two-phase process: in the first phase, a temporary persistence context is populated with empty objects; in the second phase, the objects' member data is read from the database. If an object contains an association or a collection, the query performs a recursive call to the session's **get()** method. This clears the temporary persistence context. If the parent object contains any other associations to be read as part of the second phase, Hibernate throws an assertion because the associations cannot be found in the persistence context.

A workaround is to increase the maximum fetch depth until the error no longer occurs, by setting the **hibernate.max_fetch_depth** to a large value.

- [JBPAPP-3565](#): **org.hibernate.cfg.OneToOneSecondPass** metadata causes an **org.hibernate.PropertyValueException** at runtime. If JOINS exist but do not contain the property **otherSideProperty**, **otherSideJoin** will not be null and will retain old invalid data.
- [JBPAPP-3487](#): Hibernate generates the wrong alias in the table-pre-class inheritance strategy in some situations. Take the following mapping as an example:

```
<class abstract="true" name="Customer">
  <composite-id class="PersonID" name="id">
    <key-property column="NR_RZBK" name="num" />
    <key-property column="TXT_OID" name="name" />
  </composite-id>
  <union-subclass name="CarBuyer">
    <property column="PID" name="pid" update="false" />
    <property column="TXT_OID_TESTB" name="sellerName" />
    <many-to-one cascade="persist, merge, save-update" class="Seller"
      insert="false" name="seller" update="false">
      <column name="NR_RZBK" />
      <column name="TXT_OID_TESTB" />
    </many-to-one>
  </union-subclass>
</class>
```

In this case, the SELECT statement issued is:

```
select
  testc0_.NR_RZBK as NR1_1_,
  testc0_.TXT_OID_TESTB as TXT2_1_,
  testc0_.NR_RZBK as NR1_1_0_,
  testc0_.TXT_OID as TXT2_1_0_,
  testc0_.PID as PID2_0_,
  testc0_.TXT_OID_TESTB as TXT2_2_0_,
  testc0_.NR_RZBK as NR1_2_0_
from
  CarBuyer testc0_
```

```

where
    testc0_.NR_RZBK=?
    and testc0_.TXT_OID_TESTB=?

```

In **org.hibernate.collection.PersistentSet**, Hibernate will attempt to construct the **CarBuyer** object:

```

public Object readFrom(
    ResultSet rs,
    CollectionPersister persister,
    CollectionAliases descriptor,
    Object owner) throws HibernateException, SQLException {
    Object element = persister.readElement( rs, owner,
    descriptor.getSuffixedElementAliases(), getSession() );
    if (element!=null) tempList.add(element);
    return element;
}

```

However, the aliases returned by **descriptor.getSuffixedElementAliases()** are from:

```

org.hibernate.persister.collection.AbstractCollectionPersister
.AbstractCollectionPersister(Collection,
CollectionRegionAccessStrategy, Configuration,
SessionFactoryImplementor)

...
keyColumnNames = new String[keySpan];
keyColumnAliases = new String[keySpan];
int k = 0;
while ( iter.hasNext() ) {
    // NativeSQL: collect key column and auto-aliases
    Column col = ( (Column) iter.next() );
    keyColumnNames[k] = col.getQuotedName(dialect);
    keyColumnAliases[k] = col.getAlias(dialect,table);
    k++;
}

```

In this case, according to the **Column.getAlias(Dialect)** algorithm, the alias of key column **TXT_OID** is **TXT2_1_**, which is the same as the column **TXT_OID_TESTB**'s alias.

There are three possible workarounds for this issue:

- Map your column name to something else, for example, change **TXT_OID_TESTB** to **TXT2_OID_TESTB**.
- Use a different inheritance strategy.
- Change the order of the properties mapped in Hibernate mapping. (This has not yet been verified for Hibernate Annotations.)
- **JBPAPP-3485**: Hibernate does not currently contain support for uncommitted "dirty" reads. This can cause problems in DB2 environments, where select statements automatically block tables while the query is being performed. Uncommitted read support is planned for the next version of JBoss Enterprise Application Platform.

- [JBPAPP-3483](#): Hibernate does not currently log the execution time for each query. This is expected in future versions of JBoss Enterprise Application Platform.
- [JBPAPP-3284](#): This is an issue with cglib. See http://sourceforge.net/tracker/index.php?func=detail&aid=2796998&group_id=56933&atid=482368 for more details. cglib removes field annotations when transforming a class with **TransformingClassGenerator** because of a problem with the **visitField** method. It is possible to work around this issue by annotating getters instead of fields, but this may not be possible for heavy applications ported to JBoss Enterprise Application Platform 5.x.
- [JBPAPP-3223](#): Hibernate does not currently support tuple syntax in HQL or Criteria on databases which do not support tuple syntax. For example, if a database does not support tuple syntax:

```
where (a,b) in ( (1,2), (3,4) )
```

Hibernate should translate to:

```
where ( (a=1 AND b=2) OR ( (a=3 AND b=4) )
```

There is currently no workaround for this issue except to avoid using this syntax query on databases that do not support tuple syntax.

- [JBPAPP-3075](#): If a database reserved keyword is used as a property name with a Hibernate Validator annotation (for example, **@Min** or **@Max**), it will cause exceptions in **SchemaExport**, even if you specify a column name. This is because Hibernate ignores the name specified. The workaround is to map the property name to something that is not a database reserved keyword with the **@Column** annotation. The fix for this issue is expected in Hibernate 4.
- [JBPAPP-3069](#): The **QueryByExampleTest.testJunctionNotExpressionQBE** test fails in Sybase because **ansinull** is set to off by default. This test builds a disjunction predicate such as **(OR^ (ex) (NOT ex))**. This should match everything in the database, but because ANSI SQL evaluates all comparisons involving NULL values as UNKNOWN, not all matches are returned. To work around this problem, append the following string to the JDBC URL:

```
?SQLINITSTRING=set ansinull on
```

If this is not possible, an alternative is to execute the following Java code (or similar) after obtaining a Hibernate session **s**:

```
s.connection().createStatement().execute("set ansinull on");
```

- [JBPAPP-3034](#): When batch insert statements are ordered, embedded classes are not taken into account. There are two possible workarounds for this issue. The first is to set **ORDER_INSERTS** to **FALSE** when embedded classes are used. The second option is to explicitly call **session.save()** on child objects to enforce their SQL insertion orders.
- [JBPAPP-3032](#): MySQL does not currently support millisecond and microsecond measurements when returning database values such as **TIME** and **TIMESTAMP**.
- [JBPAPP-3031](#): On Sybase, the **current_timestamp** text is not being recognized by the translator as a *method mode*. There is currently no way to work around this issue except to avoid relying upon function replacement for **current_timestamp**.

- **JBPAPP-3030**: On Sybase, **SchemaExport** cannot be used to create stored procedures while in while in chained transaction mode. The workaround for this case is to add the following code immediately after the definition of the new stored procedure:

```
<database-object>
<create>
sp_procxmode paramHandling, 'chained'
</create>
<drop/>
</database-object>
```

- **JBPAPP-2791**: Applications that map Hibernate to use **cglib** as a byte provider fail to deploy because of a **java.lang.SecurityException**. An error message similar to the following is displayed:

```
Deployment "persistence.unit:unitName=lobtest.ear/
lobtest-ejb-1.0-SNAPSHOT.jar#lobtest-jpa-jndi" is in error due to
the following reason(s):
java.lang.SecurityException: class
"com.redhat.gss.lobtest.jpa.Item$$EnhancerByCGLIB$$defd1a7f"'s
signer information does not
match signer information of other classes in the same package
```

This occurs because the **cglib.jar** in JBoss Enterprise Application Platform is signed, and the cglib-instrumented proxy uses the **cglib.jar** signer information instead of the signer information of the application target class.

The patch for this issue has been released alongside JBoss Enterprise Application Platform 5.0 and can be downloaded from [Red Hat Support](#).

- **JBPAPP-2945**: Setting the query timeout for a PreparedStatement is not supported by PostgreSQL 8.3.7. This limitation means that queries will fail if they use an annotation like the following:

```
@QueryHint(name = "org.hibernate.timeout", value =
"100")
```

- **JBPAPP-2867**: Sybase does not currently support Hibernate **Blobs** or **Clobs**, and Hibernate does not support Sybase **text** or **image** data types. The workaround for this issue is to create user-defined types that map to the Sybase **text** and **image** types.
- **JBPAPP-2789**: **SchemaExport** fails on Oracle and Sybase when a redundant **@Column(unique=true)** or **UniqueConstraint(columnnames={...})** annotation is used on a column that is implicitly defined as unique by the declared model. The workaround is to remove the redundant **@Column(unique=true)** or **UniqueConstraint(columnnames={...})** annotation.
- **JBPAPP-2613**: When the DB2 version 9.7 driver is used with progressive streaming (the default), operations on **Blob** and **Clob** locators fail. There are two possible workarounds to this issue:
 - Create a property file named **DB2JccConfiguration.properties**, either on your class path or in a JAR on your class path. Add the following line to this file to disable progressive streaming on DB2:

```
db2.jcc.override.progressiveStreaming=2
```

- Use the DB2 version 9.1 driver instead of DB2 version 9.7.
- [JBAPP-2440](#): When a cache provider cannot be found, a `NoClassDefFoundError` is thrown with the message:

```
net/sf/ehcache/CacheException
```

When a connection provider cannot be found, a `HibernateException` is thrown with the following message:

```
Could not instantiate connection provider: " + providerClass
```

If you experience these errors, we suggest checking the cache or connection provider configuration to ensure that the provider is included in the classpath.

- [JBAPP-2408](#): An issue exists with the DB2 v9.7 driver when using an identity or native ID generator with Hibernate. The `Statement.getGeneratedKeys()` driver method in DB2 returns an empty *resultset* instead of the generated keys, which causes Hibernate to throw an exception that states that the database returned no natively generated identity value. This issue has been fixed in the version of DB2 9.7 JDBC driver released with Data Studio 2.2 and is available for download from the DB2 website. This is the recommended version for use with Hibernate.
- [JBAPP-2278](#): The **Save** operation may fail when a transient entity is reachable by multiple paths and at least one of those paths does not cascade for the **Save** operation. The current workaround for this is to save the transient entity before executing the save that had previously failed. If this is not possible, another workaround is to modify either or both cascade and entity mappings to change the order of the cascade paths so that the transient entity is saved before it cascades to the entity that requires it to be non-transient.
- [JBAPP-2276](#): The iteration order of **HashMaps** and **HashSets** for JDK 6 causes the order of columns in union clauses or union subclasses to differ depending on whether JDK 5 or 6 is used. Since the change in column order is consistent across union clauses, the resulting queries are valid; however, this change can potentially affect performance.
- [JBAPP-2275](#): Hibernate cannot be compiled under JDK 6. This occurs because the following classes require methods to be added in order to fully implement JDK 6 interfaces:
 - `org.hibernate.jdbc.ResultSetWrapper`
 - `org.hibernate.lob.BlobImpl` to implement `java.sql.Blob`
 - `org.hibernate.lob.ClobImpl`
 - `org.hibernate.lob.SerializableBlob`
 - `org.hibernate.lob.SerializableClob`

If the application you are running requires a method missing from the above classes, a `NoSuchMethodError` will be generated.

- [JBAPP-2792](#): Sybase fails to insert a new entity if it overflows its column. However, it does not throw an exception, so Hibernate cannot tell that the insert failed. To work around this issue, the

application must validate entity properties to ensure that they do not overflow the underlying column.

- **JBPAPP-2791**: **SchemaUpdates** fail in Sybase ASE 15 tables when a new column is added without a default value. To work around this issue, ensure that a default value is included when adding new columns with **SchemaUpdate**.
- **JBPAPP-1895**: The Hibernate tests **JoinTest.java** and **QueryAndSQLTest.java** fail when testing queries with untyped parameters for DB2, which does not support untyped parameters. You can work around this issue by modifying the queries so that the parameters are cast to an appropriate data type, as described on the JIRA.
- **JBPAPP-1613**: Null values for columns mapped as **Boolean** in Sybase are persisted as **0** instead of **null**. The workaround for this issue is to map **type="org.hibernate.test.where.NumericTrueFalseType"** instead of **type="boolean"**.
- **JBPAPP-1554**: Sybase only allows only one entry (for example, column name or **'***) in a subquery select list. The HQL function, **elements()**, fails when the collection elements have a composite ID, because the generated SQL contains a subquery select list with multiple entries. The workaround is to avoid using HQL **elements()** if the elements have a composite key. Instead, reformulate the HQL so that no subquery has multiple entries in its select list.
- **JBPAPP-1545**: On Sybase, when a query has an ANSI join with three or more joins, and one join involves a union subclass, the query may fail with **SySQLException** because a column is not within the scope of the joined table expression. The current recommendation is to avoid using join fetches that involve union subclasses.
- **JBPAPP-1230**: When a **DetachedCriteria** is used as a subquery, the generated SQL contains a column alias in the subquery. On Sybase, a **SySQLException** is thrown because Sybase does not allow column aliases in subqueries. The workaround for this issue is to use an HQL query instead of a **DetachedCriteria** in a subquery.
- **JBPAPP-1123**: When **@OrderBy** is used on joined classes (using a join table), the generated SQL is invalid on MySQL, PostgreSQL, Oracle, and MSSQL because the "order by" clause qualifies the columns using the actual table name. The "order by" clause should use the table alias instead.
- **JBPAPP-1082**: When a **char** property is used and it is not initialized, then Hibernate initializes it to **0** and persists a string containing the character **\u0000**. PostgreSQL throws an exception because it does not allow the character **\u0000** embedded in a string. There is currently no workaround for persisting **\u0000** in a **char** column using PostgreSQL.

To persist a NULL for an uninitialized **char** property instead of **\u0000**, use **java.lang.Character** instead of the primitive **char** type. This avoids the exception when the property is initialized. Attempting to persist a **java.lang.Character** property that is set to **\u0000** still results in an exception.

- **JBPAPP-1071**: In some cases, when there are foreign key constraints defined on columns in a primary key, **SchemaExport** incorrectly declares them as nullable when it generates **CREATE TABLE** statements. This fails on MSSQL, DB2, and Sybase because those databases require primary key columns to be non-nullable.

The workaround for this issue is to explicitly indicate which columns should be non-nullable, such as:

- Add `nullable=false` to `@JoinColumn`
- Add `optional=false` to `@ManyToOne`
- Add an `@AttributeOverride` with `@Column(name="mapkey", nullable=false)` in case of a `@CollectionOfElements` using a Map
- Add `nullable=false` in `@Column` when inside a `@CollectionId` or inside `@MapKey`
- [JBPAPP-3010](#): The `evict(Object)` method in `EntityRegionAccessStrategy` and `CollectionRegionAccessStrategy` attempts to remove objects from the cache without regard for transaction isolation. This is currently unsupported because JBoss Cache's `removeNode` method does not deal with transactions.
- [JBPAPP-3019](#): The `doc/examples/jboss-web-services-examples` context causes a number of exceptions to occur. This context error means that the JBoss Web Services examples will not work correctly.

JBoss Messaging Known Issues

- [JBPAPP-3965](#): Two JBoss Messaging Test Suite tests fail on Oracle 11g R1, R2 and RAC with the latest JDBC driver, version 11.2.0.1.0:
 - `QueueManagementTest.testDestroyDestinationProgrammatically`
 - `QueueManagementTest.testDestroyDestinationProgrammaticallyWithParams`

These tests use a large value for the `fullSize` queue configuration parameter, which is passed to the `setFetchSize` method on the `java.sql.PreparedStatement`. A problem with the JDBC driver means that more than the usual amount of memory is consumed when `executeQuery()` is called, which results in a `java.lang.OutOfMemoryError`, causing the test to fail.

- [JBPAPP-3904](#): Oracle JDBC driver version 11.1.0.7.0 causes the JBoss Messaging Test Suite to fail with a `SQLException` ("Bigger type length than Maximum") on Oracle 11g R1. This is caused by a regression in Oracle JDBC driver 11.1.0.7.0. We recommend Oracle JDBC driver version 11.2.0.1.0 for use with Oracle 11g R1, Oracle 11g R2, Oracle RAC 11g R1 and Oracle RAC 11g R2.
- [JBPAPP-3157](#): When unicode multibyte characters are used in a message Selector, JBoss Messaging throws a `TokenMgrError`.
- [JBPAPP-1745](#): When JBoss Transactions is upgraded from version 4.4.0 to version 4.6.1, the JBoss Messaging `XAResourceRecoveryTest` fails. The test is meant to simulate a failure during the commit process to prompt the Recovery Manager to recover the transaction, but with JBoss Transactions 4.6.1, the transaction is not recovered. The workaround for this failure is to manually start the Recovery Manager with the following code:

```
recoveryManager =
RecoveryManager.manager(RecoveryManager.INDIRECT_MANAGEMENT);
```

- [JBPAPP-1746](#): When JGroups is upgraded to version 2.6.13, the JBoss Messaging `LargeClusterTest` fails. The test attempts to start seven nodes on one machine before testing, but some nodes fail to start. The test passes if the number of nodes is reduced.
- [JBPAPP-2033](#): `EnableMessageCounters` on the `jboss.messaging.ServerPeer` JMX

interface, cannot be set to **true**. A workaround to enable message counters is to invoke the **enableMessageCounters()** operation on the same JMX interface.

JBoss mod_cluster Known Issues

- [JBPAPP-3724](#): In **mod-cluster-jboss-beans.xml**, the default value for **maxAttempts** is incorrectly set to **-1**. This value is invalid. The correct value, **1**, is used by default.
- [JBPAPP-3463](#): When an application is undeployed, sessions that were forwarded to the application server by mod_cluster before undeploy notification was received may result in an error 503 - This application is not currently available.
- [MODCLUSTER-123](#): If the **root** context ("/") is deployed and enabled, other contexts cannot be disabled. It is also impossible to specify that other contexts not be forwarded to the **root** context.
- [MODCLUSTER-120](#): Should you encounter an **[emerg] create_mem_node <node file path> failed** error, use the **ipcrm -m** command before restarting httpd.
- [MODCLUSTER-113](#): **org.jboss.modcluster.demo.servlet.ThreadCountLoadServlet** has been removed from mod_cluster but is still specified in the **web.xml** file belonging to **load-demo.war**. This results in deployment errors. The workaround for this issue is to remove the **<servlet>** and **<servlet-mapping>** sections for the **threads** servlet.

JBoss Web Services Known Issues

- [JBPAPP-3785](#): When a method with an additional namespace for a header parameter is invoked after a method that does not have this additional namespace, the wrong namespace is used in the SOAP operation when it is serialized.
- [JBPAPP-3028](#): The upgrade from JBoss Web Services 2.0.1.SP to JBoss Web Services Native 3.1.2.SP incorporated many changes and new features. The additional processing time required for several new features (resource injection, support for **@PostConstruct** and **@PreDestroy**, etc.) has resulted in a slight degradation in overall performance.

Remoting Known Issues

- [JBPAPP-3709](#): When **org.jboss.remoting.marshall.MarshallerLoaderHandler** gets a request for a class, it returns an instance of **org.jboss.remoting.loading.ClassBytes**. If **MarshallerLoaderHandler** is unable to find the desired class, then the returned **ClassBytes** object has a null value for the class byte array. However, **org.jboss.remoting.loading.ClassByteClassLoader** does not check for the possibility of a null byte array, so a **NullPointerException** results.
- [JBPAPP-3392](#): EJB3 clients do not reuse existing socket connections on subsequent invocations; a new connection is created per invocation.

RESTEasy Known Issues

- [JBPAPP-2993](#): The **spring-hibernate-contacts** example cannot be compiled because it depends on an archive that is no longer available in the Maven repositories.
- [JBPAPP-2992](#): the readme file for the Spring MVC example located at **doc/examples/reteasy-examples/reteasy-springMVC/README.txt** contains an invalid URL. The current text:

```
List all available contacts:
http://localhost:9095/contacts
```

Should be replaced with the following:

```
List all available contacts:
http://localhost:8080/contacts
```

- [JBPAPP-2991](#): The readme file for the API Clients example located in `doc/examples/resteasy-examples/api-clients` contains an invalid command for the Twitter example startup:

```
Twitter Client:
- Run:
  mvn exec:java -
  Dexec.mainClass="org.jboss.resteasy.examples.twitter.Twitter" -
  Dexec.args="<userId> <password>"
```

Replace the previous code with the following:

```
Twitter Client:
- Run:
  mvn exec:java -
  Dexec.mainClass="org.jboss.resteasy.examples.twitter.TwitterClient"
  -Dexec.args="<userId> <password>"
```

`doc/examples/resteasy-examples/api-clients/` also contains two superfluous Eclipse project files, `.classpath` and `.project`. We recommend deleting these files.

Seam Known Issues

- [JBPAPP-4015](#): When hot deployment is used for EARs that contain multiple WAR files, the first web application to be accessed works correctly, but a web request to subsequent web applications causes a **NullPointerException** because the application context is not set up when the hot deployment filter runs.
- [JBPAPP-4013](#): When using the web services page of the Seambay example in Internet Explorer 8, values entered in the username and password fields do not propagate to the request area. To work around this issue, manually enter values for the request area.
- [JBPAPP-4012](#): The Tasks example does not work correctly in Internet Explorer 8. When "Undo this Task" is selected on the Resolved Tasks page, the task returns to the Tasks page, but does not disappear from the Resolved Tasks page.
- [JBPAPP-4010](#): Resources from the resource servlet, such as CAPTCHA images, do not re-render correctly because they are cached by the browser. An example of a workaround to force re-rendering is available on the JIRA.
- [JBPAPP-4009](#): When attachments are included in Seam Mail, the attachment is not show in Outlook 2007. The workaround for this issue is to edit `org.jboss.seam.mail.ui.UIBody`. Instead of:

```
MimeMultipart bodyRootMultipart = new MimeMultipart("related");
```

Use:

```
MimeMultipart bodyRootMultipart = new MimeMultipart("mixed");
```

And then recompile Seam Mail.

- **JBPAPP-4007**: `org.jboss.seam.transaction.EjbSynchronizations` does not comply with Java specifications. As stated in http://java.sun.com/blueprints/guidelines/designing_enterprise_applications/transaction_management classes that implement `SessionSynchronization` may not have the `TransactionsAttribute TransactionAttributeType.SUPPORTS`. To work around this issue, use `TransactionAttributeType.REQUIRED` instead.
- **JBPAPP-4006**: `org.jboss.seam.core.Interpolator` swallows all exceptions and prints warnings with limited information.
- **JBPAPP-4005**: Seam JMS support does not work after failover to a slave node because the `topicConnection` field in the `SubscriptionRegistry` object becomes invalid when the connection to the master node fails. Every subsequent call to `SubscriptionRegistry.subscribe()` fails with `SpyJMSEException`, and JMS subscriptions will no longer work even if the master node revives. To work around this issue, catch the exception and retry with a new `topicConnection` if a subscribe request fails, as described on the JIRA.
- **JBPAPP-4004**: There is a typo in the message logged by the `org.jboss.seam.async.AsynchronousExceptionHandler` when handling "Exception thrown while executing asynchronous call". "Exception" should be "Exception".
- **JBPAPP-4003**: Requirements have changed between JBoss Enterprise Application Platform 4.x and JBoss Enterprise Application Platform 5.x such that page-level `messages.properties` in the `ResourceLoader` must be stored in `META-INF/classes` or they will not work. The documentation is not explicit about this.
- **JBPAPP-4002**: `Initialization.redeploy` includes code that creates a new lock and then locks it immediately. This does not provide any level of protection, and could potentially cause problems in debug mode.
- **JBPAPP-3855**: The SeamSpace `testCreateBlog` unit test fails with a `NullPointerException` on IBM JRE 1.6.0.
- **JBPAPP-3769**: In profiles other than `production`, `MockResponseStateManager` does not override `ResponseStateManager#isPostBack`. This means that both Faces and non-Faces requests are evaluated as postbacks. There is no workaround for this issue.
- **JBPAPP-3764**: `EJBSynchronization` is responsible for cleaning up the `JbpmContext` after the end of the transaction. Both elements are part of the event context, but `EJBSynchronization` has a different lifecycle. This means that the event context ends before `EJBSynchronization` can clean up the `JbpmContext`. This causes a memory leak. This issue occurs only with container-managed transactions; to work around it, use bean-managed transactions instead.
- **JBPAPP-3546**: The iText example contains a "Programming Skills" section. Items selected in the multi-select list are supposed to be included in the generated PDF. However, when multiple items are selected, only the first item is included in the generated PDF.

- [JBPAPP-3525](#): `s.tld` in `jboss-seam-ui.jar/MANIFEST.MF` contains attribute entries that are defined incorrectly according to the XML Schema for the JSP Taglibrary descriptor. The workaround for this issue is to replace `jboss-seam-ui.jar/META-INF/s.tld` with the `s.tld` file attached to [JBPAPP-3525](#).
- [JBPAPP-2385](#): Seam's Spring example fails with an `IllegalStateException` when the login form is submitted on the IBM virtual machine. This is due to a defect in the IBM virtual machine. The fix for this issue has been deferred until the IBM virtual machine has been fixed.
- [JBPAPP-2377](#): The SeamSpace example fails with a `NullPointerException` when submitting a new blog entry on the IBM virtual machine. This is due to a defect in the IBM virtual machine. The fix for this issue has been deferred until the IBM virtual machine has been fixed.

Documentation Known Issues

- [JBPAPP-3818](#): "Using Seam with JBoss Tools" in the *Seam Reference Guide* has not been revised for use with JBoss Enterprise Application Platform and should be considered obsolete.

9. TECHNOLOGY PREVIEWS

This section describes the Technology Preview features released alongside JBoss Enterprise Application Platform, their installation, and any known issues.



WARNING

Technology Preview features are not fully supported under Red Hat subscription level agreements (SLAs), may not be functionally complete, and are not intended for production use. However, these features provide early access to upcoming product innovations, enabling customers to test functionality and provide feedback during the development process. As Red Hat considers making future iterations of Technology Preview features generally available, we will provide commercially reasonable efforts to resolve any reported issues that customers experience when using these features.

9.1. JBoss Web Services CXF



WARNING

You should make a complete backup of your JBoss Enterprise Application Platform installation before installing JBoss Web Services CXF.

Follow these steps to install JBoss Web Services CXF:

1. Download and unzip the **jbossws-cxf-3.1.2.SP3-3.SP3.6.ep5.e14-tp-installer.zip** in the home directory (your **\$JBOSS_HOME**) of a pristine Enterprise Application Platform installation.
2. Run **ant** in the created directory, **jbossws-cxf-tp-installer**.

**NOTE**

This step will replace JBoss Web Services Native with JBoss Web Services CXF in every configuration that contains JBoss Web Services Native.

Documentation is available from the JBoss.org Wiki: <http://community.jboss.org/wiki/JBossWS-StackCXFUserGuide>.

9.1.1. Known Issues with JBoss Web Services CXF

- **JBPAPP-3267**: Running **ant deploy-jboss510** from the stack checkout directory or a generated binary distribution does not result in a working application server configuration. This occurs primarily as a result of changes to **src/main/scripts/assembly-deploy-artifacts.xml**, which were made to produce the **tp-installer**.

A. REVISION HISTORY

Revision 5.0.1-12.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
Revision 5.0.1-12 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
Revision 1.9-0 JIRA corrections.	Tue Feb 02 2010	Laura Bailey