



JBoss Enterprise Application Platform 5

リリースノート 5.1.0

JBoss Enterprise Application Platform 5.1.0 向け
エディション 5.1.0

JBoss Enterprise Application Platform 5 リリースノート 5.1.0

JBoss Enterprise Application Platform 5.1.0 向け
エディション 5.1.0

Laura Bailey
lbailey@redhat.com

Rebecca Newton
rnewton@redhat.com

Joshua Wulf
jwulf@redhat.com

法律上の通知

Copyright © 2011 Red Hat.

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.

概要

本リリースノートには現在の製品マニュアルには記載されていない JBoss Enterprise Application Platform 5.1.0 関連の重要な情報が含まれている可能性があります。本リリースノートをすべて読んでから JBoss Enterprise Application Platform 5.1.0 をインストールするようにしてください。

目次

1. はじめに	3
1.1. 概要	3
2. JBOSS ENTERPRISE APPLICATION PLATFORM 5.1.0で追加された機能	3
2.1. Red Hat Enterprise Linux 6 の対応	3
2.2. Native Package	4
2.3. JBoss Web Services	4
2.4. Hibernate	4
2.5. PicketLink Federation	4
2.6. 以前のリリースで導入されていた機能	4
3. コンポーネントのバージョン	6
4. インストールの注意点	8
4.1. 対応の構成	8
4.2. JBoss Enterprise Application Platformのインストール	8
4.3. デフォルトの起動プロファイル	9
4.4. ソースファイル	9
5. 製品サポート	9
6. 文書	9
7. TECHNOLOGY PREVIEW 機能	9
7.1. PicketLink	10
8. 除外、削除、廃止予定のアイテム	10
8.1. 除外アイテム	10
8.2. 廃止予定のアイテム	10
8.3. 削除アイテム	11
9. 本リリースで修正された問題	11
9.1. 修正済みの一般的な問題	11
9.2. Hibernate 関連で修正済の問題	12
9.3. Remoting 関連で修正済みの問題	15
9.4. Seam 関連で修正済の問題	15
9.5. セキュリティ関連で修正された問題	19
9.6. Application Server 関連で修正された問題	19
9.7. 文書関連で修正された問題	19
9.8. Messaging 関連で修正された問題	20
9.9. Web Services 関連で修正済みの問題	20
9.10. RESTEasy関連で修正された問題	21
9.11. クラスターリング関連で修正済みの問題	21
9.12. Cache 関連で修正済みの問題	21
9.13. Connector 関連で修正済の問題	21
10. 本リリースにおける既知の問題	22
10.1. 対応していない JAR	22
10.2. 一般的な既知の問題	22
10.3. Installer における既知の問題	25
10.4. Hibernate における既知の問題	25
10.5. JBoss Messaging における既知の問題	32
10.6. JBoss Web Servicesにおける既知の問題	32
10.7. Remoting における既知の問題	33

10.8. RESTEasy における既知の問題	33
10.9. Seam における既知の問題	34
10.10. EJB における既知の問題	34
10.11. Clustering における既知の問題	35
10.12. Connector における既知の問題	35
A. 改訂履歴	37

1. はじめに

本リリースノートには、JBoss Enterprise Application Platform 5.1.0の重要な情報が含まれています。新機能、本リリースで修正された問題、その他の既知の問題について記載されています。

1.1. 概要

JBoss Enterprise Application Platformについて

JBoss Enterprise Application Platform は、オープンソースのエンタープライズソフトウェアの中でも次の段階に進化しています。Pure Java プラットフォーム上にて、機能豊富でパフォーマンスの高い Web 2.0 アプリケーションを開発する際に力を発揮するツールです。

JBoss Seam、Hibernate、CXF Web Services、JBoss Cache、JBoss Messagingなど、ベストオブブリードのオープンソースフレームワークを統合することで、当プラットフォームはオープンソースコミュニティでのイノベーションを有効活用しています。また、JBoss Enterprise Application Platform は、Red Hat による完全なテストやサポートが行われており、代表的なエンタープライズハードウェア、ソフトウェア製品の多くで機能すると認定されています。

本リリースについて

JBoss Enterprise Application Platform 5.1.0は **マイナーリリース**です。マイナーリリースは、以前のパッチや累計パッチ (CP: Cumulative patch) のコンテンツを集約し、別途、新機能を追加する場合もあります。今後のパッチや累計パッチは、以前のマイナーアップデートをインストールしていることを前提としています。Red Hat は、製品のメジャー版のライフサイクルにおいて、マイナーリリースや非同期パッチすべてで API レベルの互換を保つよう、ビジネスレベルで取り組んでいきます。こうすることで、例えば、JBoss Enterprise Application Platform 5.1.0 がAPIレベルの互換を JBoss Enterprise Application Platform 5.0.0やJBoss Enterprise Application Platform 5の初期リリースと取るようにしています。このルールの例外として考えられるのは、非常に重要なセキュリティーの問題に対する修正などとなっています。

JBoss Enterprise Application Platform 5.1.0がリリースされたため、JBoss Enterprise Application Platform 5.0.x をお使いの方はJBoss Enterprise Application Platform 5.1.0にアップデートしてください。

詳細情報については[JBoss Enterprise Middlewareの製品アップデートとサポート方針](#)を参照してください。

2. JBOSS ENTERPRISE APPLICATION PLATFORM 5.1.0で追加された機能

JBoss Enterprise Application Platform 5.0.x には多数の更新済みコンポーネントや新規コンポーネントが含まれます。JBoss Enterprise Application Platform 5.1.0 はパフォーマンス強化、JBoss Web Services CXF Stackの追加、JDBC4への対応が追加され、さらにはTechnology PreviewとしてPicketLink Federationも導入しています。本項でこれらのコンポーネントについて詳しく見ていきます。

2.1. Red Hat Enterprise Linux 6 の対応

JBoss Enterprise Application Platform 5.1 のZIP ディストリビューションは、Red Hat Enterprise Linux 6 で使用できるよう完全なテストが実施されています。該当のNative Package は、<http://access.redhat.com>から入手可能です。

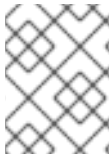
2.2. Native Package

Native Package は `jboss-ep-native-5.1.0*.zip` のパターンに準拠するよう名前が変更されました。この新規統合版のNative Packageは、JBoss Enterprise Application Platform あるいは JBoss Enterprise Web Platformと合わせて利用可能です。

Native packageも拡張され Solaris 技術に対しNSAPIが提供されています。NSAPI はSun Java System Web Server とJBoss Enterprise Application Platform をつなぐ際に使うコネクタです。

2.3. JBoss Web Services

JBoss Enterprise Application Platform 5.0 では、プラグ可能なアーキテクチャが導入され、JBoss Web Services Native に合わせて、対応しているWeb Services プロバイダとして Web Services Stack を選ぶことができます。JBoss Enterprise Platform 5.1.0 では、Web Service Stack の代わりとなる、JBoss Web Services CXF が発表されています。



注記

当プラットフォームで使うWeb Services Stack はインストール時に決定する必要があります。さらなる詳細については、『Installation Guide』を参照してください。

JBoss Web Services Native

JBoss Web Services Native は、JBoss のWeb サービス規格実装で Java EE 5 に準拠しています。JBoss Enterprise Application Platform 5.1 以前では唯一の Web サービススタックで、JBoss Enterprise Application Platform 5.1.0 ではデフォルトの Web サービスとなっています。

JBoss Web Services CXF

JBoss Web Services CXF には、Apache CXF で提供されている機能 (WS-Security、WS-Policy、WS-Addressing、WS-ReliableMessaging、basic WS-Trust、MTOM)の多く、さらにエンドポイントメトリクス、記録管理、エンドポイントアドレスのリライトなど、一般的なJBoss Web Services Stack 機能が存在します。JBoss Enterprise Application Platform 5.1.0 は、JBoss Web Services CXF Stack をオプションの Web サービススタックとして導入しています。

2.4. Hibernate

Hibernate は、Java database Connectivity 4 (JDBC 4)へ完全対応するようになりました。結果、Clob および Blob サポートが拡大し、例外処理の高度化が可能になります。

2.5. PicketLink Federation

PicketLink Federation はSAML v2.0やWS-Trustに対応するセキュリティ、アイデンティティフレームワークとなっています。JBoss Enterprise Application Platform 5.1.0 では、Technology Preview として PicketLink を導入しています。

2.6. 以前のリリースで導入されていた機能

2.6.1. JBoss Enterprise Application Platform 5.0.1で導入された機能

2.6.1.1. Native Package

Native Package が拡張されJBoss Native や mod_jkだけでなく、Windows テクノロジー向けのprovide mod_cluster や ISAPI を提供しています。追加機能を以下にて説明しています。

- **mod_cluster** は httpd-ベースのロードバランサです。 **mod_jk**のように、通信チャネルを使ってリクエストをhttpd からアプリケーションサーバーノードに転送しますが、サーバーノードからロードバランサへの通信チャネルを使いリアルタイムフィードバックを行い、より効率的にロード最適化とリクエストルーティングができるようにします。詳細については、 <http://docs.redhat.com/>にある mod_cluster 文書を参照してください。
- **ISAPI** は、Microsoft IIS Web サーバからJBoss Enterprise Application Platformをつなぐ際に利用するコネクタです。

2.6.1.1.1. Native Solaris SPARC パッケージ

Native Solaris SPARC パッケージは、JBoss Enterprise Application Platformの本リリース以降にて提供されています。Native package のインストールに関する説明は、JBoss Enterprise Application Platform 『Installation Guide』を参照してください。

2.6.2. JBoss Enterprise Application Platform 5.0で導入された機能

2.6.2.1. JBoss AS

JBoss AS 5 製品群は第二世代の Microcontainer ベースの最新型エンタープライズ Java runtimeや世界でもっとも人気のあるアプリケーションサーバーの最新リリースです。最新の Java EE 仕様 (Java EE 5) に対応しているだけでなく、最も優れたエンタープライズクラスサービスの多くを統合し高度なメッセージング、永続性、トランザクション、キャッシング、高可用性を実現しています。

2.6.2.2. JBoss Microcontainer

JBoss Microcontainer はモジュラ JBoss JMX Microkernelのリファクタリングです。また、JBoss Microcontainer は軽量のカーネルで、ローディング、ライフサイクル、POJO 間の依存関係を管理します。エンタープライズサービス、JBoss Microcontainer と Servlet/JSP コンテナ、EJB コンテナ、デプロイヤ、管理ユーティリティを統合するために使用され、標準の Java EE 5 プロファイルを提供します。

2.6.2.3. JBoss Messaging

JBoss Enterprise Application Platform 5.1では、JBoss Messaging はデフォルトの JMS プロバイダです。JBoss Messaging は、レガシーのJBoss JMS プロバイダであるJBossMQ の完全なリライトで、JBoss MQ と比較するとシングルノードおよびクラスタ環境の両方でパフォーマンスが向上しています。

2.6.2.4. JBoss Cache

JBoss Cache は EJB と HTTP セッションステートの複製に使用され、JPAに対する拡散エンティティキャッシングに対応しています。その一方で、新しく効率的なロッキングスキーム (MVCC – MultiVersion Concurrency Control : 多版型同時実行制御) で、パフォーマンスやスケーラビリティのエンベロープを継続的にプッシュしています。

2.6.2.5. JBoss Web Services Native

JBoss Web Services Native は、JBoss で実装されているJava EE 5準拠のWeb サービス規格です。JBoss Enterprise Application Platform 5.1 以前のバージョンで唯一提供されてた Web Services スタッ

クで、JBoss Enterprise Application Platform 5.1ではデフォルトの Web Services スタックとなっています。

2.6.2.6. Native Package

Native package は、JBoss Enterprise Application Platformに対するオプションのコンポーネントで、JBoss Native と mod_jkテクノロジーが組み込まれています。これらのテクノロジーについては以下にて説明しています：

- JBoss Native は、Apache Portable Runtime (APR)、OpenSSL、Tomcat Native (TC-native)で構成されています。
 - **Apache Portable Runtime (APR)** は、スケーラビリティ、パフォーマンスが向上されており、ネイティブのサーバー技術との統合性もよくなっています。Apache Portable Runtime は移植性の高いライブラリでApache HTTP Server 2.xの中核に置かれており、多くの用途があります。例えば、高度なIO機能（例：sendfile、epoll、OpenSSL）、オペレーティングシステムレベルの機能（例：任意の数生成、システムの状態）、ネイティブのプロセス処理（共有メモリ、NTパイプ、Unix ソケット）などです。
 - **OpenSSL** は、Secure Sockets Layer (SSL) および Transport Layer Security (TLS) プロトコルを実装し、基本的な暗号ライブラリが含まれています。
 - **Tomcat Native (TC-Native)** はJava Native Interface (JNI) で、Tomcat の中核機能の大半をJava ではなくネイティブコードで提供しています。こうすることで、サーバーのスピードが全体的に改善されるのです。
- **mod_jk** は、Tomcat JSP コンテナとApache などの別の Webサービスをつなぐのに利用するコネクタです。



注記

Native Package のRed Hat Enterprise Linux 5ディストリビューションにはOpenSSL あるいは Apache Portable Runtimeが含まれていないのは、オペレーティングシステム自体にすでにこれらの技術が含まれているためです。

3. コンポーネントのバージョン

本項では、Enterprise Application Platform 5.1を構成するコンポーネントのバージョンについて説明します。

表1 リリース5.1.0 と 5.0.1との間でコンポーネントのバージョンを比較

コンポーネント	EAP 5.1	EAP 5.0.1
JBoss Application Server	5.1.0.GA	5.1.0.GA
JBoss Microcontainer	2.0.10.GA	2.0.9.GA
JBoss Native	2.0.9.GA	2.0.9.GA
Hibernate Core	3.3.2.GA_CP03	3.3.2.GA_CP01
Hibernate Entity Manager	3.4.0.GA_CP03	3.4.0.GA_CP01

コンポーネント	EAP 5.1	EAP 5.0.1
Hibernate Annotations	3.4.0.GA_CP03	3.4.0.GA_CP01
Hibernate Search	3.1.1.GA_CP03	3.1.1.GA_CP01
Hibernate Validator	3.1.0.GA	3.1.0.GA
クラスタ化リモートEJB3 プロキシ	1.3.6.GA	1.1.22.GA
JBoss Cache	3.2.5.GA	3.2.1.GA
JBoss HA Server API	1.1.3.Final	1.1.1.GA
JBoss JAXR	2.0.1	2.0.1
JGroups	2.6.16.GA	2.6.13.GA
JBoss EJB3	1.3.6.GA	1.1.22.GA
JBoss JTA	4.6.1.GA_CP07	4.6.1.GA_CP03
JBoss JTS	4.6.1.GA_CP07	4.6.1.GA_CP03
JBoss Negotiation	2.0.3.SP3	2.0.3.SP1
JBoss Managed	2.1.1.GA	2.1.1.GA
JBoss Messaging	1.4.7.GA	1.4.6.GA
JBoss Metadata	1.0.6.GA	1.0.2
JBoss Web	2.1.10.GA	2.1.7.GA
JBoss Web Services - Native	3.1.2.SP7	3.1.2.SP3
JBoss Web Services - CXF	3.1.2.SP7	-
JBoss AOP	2.1.6.CP01	2.1.6.GA
JBoss Remoting	2.5.3.SP1	2.5.2
JBoss Serialization	1.0.4.FINAL	1.0.3.GA
JBoss XB	2.0.1.GA	2.0.1.GA

コンポーネント	EAP 5.1	EAP 5.0.1
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.SP4	2.0.4.SP1
JBoss Profiler-jvmti	1.0.0.CR5	1.0.0.CR5
Admin Console	1.3.4.SP1	1.3.2.GA
mod_jk	1.2.30	1.2.28
mod_cluster	1.0.4.GA	1.0.3.GA
Seam	2.2.2.EAP5	2.2.1.EAP5
RESTEasy	1.2.GA_CP01	1.1.GA_CP01
RichFaces (Seam内)	3.3.1.SP2	3.3.1.GA



注記

JBoss Enterprise Application Platform 5.1には、Technology Preview として PicketLink Federation が含まれています。

Technology Preview 機能は、Red Hat サブスクリプションレベルアグリーメント (SLA) で完全に対応されておらず、機能的にも完全でない場合があるため、実稼働環境での使用を目的として設定されていません。しかし、これらの機能により、今後の製品イノベーションを簡単に入手でき、開発段階でお客様が機能のテストやフィードバックを行えるようになっています。Red Hatは今後もTechnology Preview 機能を通常利用できるよう検討しており、これらの機能を利用中の問題で報告されたものについては解決に向けビジネスレベルで相応に取り組んでいく予定です。

4. インストールの注意点

4.1. 対応の構成

<http://www.jboss.com/products/platforms/application/supportedconfigurations/>にて互換性があり、認定済みの構成の最新版が表にまとめられています。テスト済み、対応構成に関する情報についてはこのリストを参照してください。

4.2. JBoss Enterprise Application Platformのインストール

JBoss Enterprise Application Platform設定のインストール、確認に関する説明は、『インストールガイド』を参照してください。

4.3. デフォルトの起動プロファイル

デフォルトで起動するプロファイルは、**default**で、デフォルトのサービスセットを含んだbase Java EE 5 サーバープロファイルです。Java EE 5アプリケーションをデプロイするのに必要なサービスでもっとも頻繁に利用されるものが含まれています。JAXR サービス、IIOP サービス、またはクラスタリングサービスは含まれていません。

default プロファイルは、ロード、ストレス、可用性、パフォーマンステストの実行や、実稼働環境を目的に設計されていません。また、このベータリリースのパフォーマンスが最終版のリリースとなるわけではありません。

4.4. ソースファイル

ソースZIPファイル

- <ftp://ftp.redhat.com/pub/redhat/jbeap/5.1.0/en/source/jboss-eap-src-5.1.0.zip>

5. 製品サポート

バグ、バグに発展する可能性のあるもの、開発の問題、および／あるいは質問がある場合は、Customer Support Portal (<http://access.redhat.com>) から JBoss サポート案件として起票してください。

6. 文書

本リリースでは、JBoss Transactions の文書が再構築され、『Transactions 開発クイックスタートガイド』、『Transactions 管理者ガイド』、『Transactions 開発ガイド』に統合されています。

本リリースにおいても『JBoss Security Guide』が追加されており、本プラットフォームにおけるセキュリティ関連の情報がまとめられています。

本リリースで発表された JBoss WS CXF に関するユーザガイドは当プラットフォームの文書に追加されています。

API全文書、コード例、オンライン上にあるリリースノートへのリンクは、別のZIPファイル **jboss-eap-docs-<version>.zip**で配布されています。

その他のガイド、文書はオンライン上の<http://docs.redhat.com>から入手可能です。最初のページの画像のように、該当の製品名(JBoss Enterprise Application Platform)を選び、サイドバーの stream (5) を選択します。

Red Hat 文書に対するフィードバックにご協力をお願いします。<http://jira.jboss.org>から直接ドキュメンテーションチームに連絡できます。

7. TECHNOLOGY PREVIEW 機能

本項では、JBoss Enterprise Application Platformと合わせてリリースされたTechnology Preview 機能、その設定、既知の問題について説明しています。



警告

Technology Preview 機能は、Red Hat サブスクリプションレベルアグリーメント (SLA) で完全に対応されておらず、機能的にも完全でない場合があるため、実稼働環境での使用を目的として設定されていません。しかし、これらの機能により、今後の製品イノベーションを簡単に入手でき、開発段階でお客様が機能のテストやフィードバックを行えるようになっています。Red Hatは今後もTechnology Preview 機能を通常利用できるよう検討しており、これらの機能を利用中の問題で報告されたものについては解決に向けビジネスレベルで相応に取り組んでいく予定です。

7.1. PicketLink

PicketLink Federation は、SAML 2.0、WS-Trust 1.3、XACML 2.0 (JBossXACML経由)、OpenID 1.1 / 2.0 に対応する、アイデンティティフェデレーションおよびシングルサインオンを当プラットフォーム提供しています。PicketLink のインストールに関する情報は、『Installation Guide』を参照してください。

8. 除外、削除、廃止予定のアイテム

除外、削除、廃止予定のアイテム

除外されるアイテム

製品リリースの機能として組み込まれていないが同製品のオープンソースコンポーネントの一部である場合は、*除外アイテム*です。

廃止予定のアイテム

今後のリリース (通常、次のメジャーリリース) から削除予定のアイテムは、*廃止予定のアイテム*。

削除アイテム

今まで製品リリースに含まれていたが含まれなくなったアイテムは、*削除アイテム*となります。通常、削除前に廃止予定アイテムになります。

8.1. 除外アイテム

本リリースで除外されたアイテムはありません。

8.2. 廃止予定のアイテム

以下が本リリースにて廃止予定アイテムとなっています。

- JBoss Cache [JBAS-7837](#)
 - Pojo Cache FIELD replication granularity [JBAS-7560](#)
- JMX Console [JBAS-7509](#)

- JFreeChart [JBAS-7716](#)
- JMX Remoting [JBAS-7510](#)
- org.jboss.jdo.castor.CastorJDOImpl MBean [JBAS-7511](#)
- JDBC3 (JDK5) classes in the JDBC adaptor [JBAS-7527](#)
- Java Remote Method Protocol Invoker and HA Invoker [JBAS-7563](#)
- PooledInvoker and PooledInvokerHA [JBAS-7564](#)
- HASessionStateTopologyComputer および HASessionStateTopologyComputerImpl [JBAS-7992](#)
- SubPartitionsInfo and SubPartitionInfo interfaces [JBAS-7992](#)
- cglib

8.3. 削除アイテム

本リリースで削除されたアイテムはありません。

9. 本リリースで修正された問題

本リリースで修正された問題は以下の通りです：

9.1. 修正済みの一般的な問題

JBPAPP-6987

Microcontainer bean は、デプロイメント時にMBeans として登録されていました。これらのMBeans は任意で、デバッグの情報を一部提供します。この動作はデフォルトではオフになりました。デフォルトでは、Microcontainer Beans はデプロイメント時に MBeans として登録されなくなりました。

JBPAPP-4730

64ビットハードウェアにてJBoss Nativeを使い、**-javaagent**のJavaオプションを実行するという特定な場合のみ、**run.sh** スクリプトの実行時にシステムが停止してしまっていました。このスクリプトは、システムアーキテクチャーのチェックをするために呼出しを行っている際に、ユーザ提供のパラメータをJava に渡していました。この呼出しが行われている時に、**run.sh** スクリプトがユーザ提供のパラメータを無視するように修正され、この条件でこのスクリプトがハングしなくなりました。

JBPAPP-4683

管理ツールが (埋め込みコンソールのように) ProfileService管理層を使って、ServiceBindingManagerサービスをアップデートすると、受け取った情報が正しくマッピングがされませんでした。バインディングの"fixedHostName"プロパティに対して受け取った値は"fixedPort"として適用されていました (逆の場合も同じ)。プロパティは両方 "false"であるため、多くの場合無害でした。しかし、いずれかのプロパティが true の場合、更新済みの値が適用されると、この間違っただマッピングが原因で、サーバー再起動時のバインディングが不正となっていました。この問題は修正され、マッピングは期待どおりに適用されるようになりました。

JBPAPP-4471

user-type-mappedタイプを MIN や MAX などの ejbql 集約関数で使う場合、CCE の例外がスローされていました。これは修正され、user-type-mappingタイプにマッピングされたCMPフィールドを EJB-QL クエリのSELECT 句にある集約関数に表示することができるようになっています。

JBPAPP-3942

`run.sh` スクリプトはファイルを作成することで、プロセスを追跡しプロセス ID、`jboss.pidfile`をより簡単に取得できるようになっています。詳細はJIRAを参照してください。

JBPAPP-3674

<distributable> なWeb アプリケーションに対して、ステータスサーブレットエラーが存在していました。Cache セッションマネージャは、Web マネージャと違う属性を持ちますが、大文字小文字の区別がないため、同じJMX オブジェクトとして認識されてました。StandardManager は'activeSessions' を、クラスタ版は'ActiveSessions' を使っていました。サーブレットがステータス情報をレンダリングするのに利用するコードで、セッションマネージャの JMX 属性で使われている大文字、小文字を検知、処理することができます。

JBPAPP-3171

JBPAPP-544にて、/statusのマッピングを削除し、セキュアな/web-console/statusのみを残すことで、セキュリティの問題が修正されました。しかし、JBPAPP-1146で再度セキュアでない/statusが含まれるようになりました。これは、**ROOT.war** webappのセキュリティを設定することで修正されました。/statusを失うことなしに、元にあったセキュリティ化の問題が解決されています

JBPAPP-2571

Microsoft JDBC drivers 2.0 を持つMicrosoft SQL サーバを実行すると、JBoss Messaging のビルドが不安定になります。この問題は**Adaptive Buffering** がドライバのデフォルト動作となっていることから発生しています。これは、`get<Type>Stream` メソッドを使うことで大きな値のみの読み込みを可能にするためです。この問題に対する現在の回避策は、JDBC 接続の URL の**responseBuffering** パラメータを**adaptive**から**full**に変更することです (JDBC ドライバのバージョン 1.2 と同じです)。

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

JBPAPP-1774

Red Hat Enterprise Linux 5 上で JBoss Enterprise Application Platform rpm を**yum** を使ってインストールする場合、Sun あるいはIBM Java Development Kit との JBoss Enterprise Application Platform での Java 要件を満たすため、Red Hat Enterprise Linux Extras チャンネルをサブスクライブする必要があります。

9.2. Hibernate 関連で修正済の問題

JBPAPP-4579

`NativeSQLQuerySpecification` の**equals** と **hashCode** メソッドが正しく実装されていないため、`session.createSQLQuery` を使うとメモリリークが発生していました。また、**LRUMap** の同時利用により、マッピングが最大許容範囲よりも大きくなっていました。これらの問題は修正され、メモリリークも発生しなくなっています。

JBPAPP-4326

JBPAPP-4471 へ変更を加えたため、ユーザがJDK Serializable 以外のクラス

と **SerializableType** をマッピングしこれらの値を二次キャッシュに問題がキャッシュしようとする
ると問題が発生していました。 **SerializableType** が **java.io.Serializable** のクラスローダ
をデシリアライズ用のクラスローダとしてレポートしていましたが、このクラスローダはクラス
ローダや非 JDK クラスの判断が付きません。

org.hibernate.util.SerializationHelper.deserialize(InputStream, ClassLoader) からのクラスローダは、デシリアライズが必要なクラスローダを解決しようと試み
るようになりました。これに失敗すると、現在のスレッドからのコンテキストクラスローダと
Hibernate クラスをロードするクラスローダを使います。

JBPAPP-4224

Hibernate は、キャッシュからクエリをリトリブすると、キャッシュ化されたクエリのタイプスタ
ンプと、現在のセッションのタイムスタンプを比較するため、Query Cache を有効にする前にセッ
ションマネージャあるいはエンティティマネージャを閉じる必要がありました。これは修正され、
Hibernate は **nextTimestamp()** を使い、キャッシュされたタイムスタンプではなく、現在のタイ
ムスタンプをフェッチしています。

JBPAPP-4183

@CollectionId は、bag の識別子カラムを記述します (例 : idbag)。 **nullable** は明示的に **false**
に設定する必要がありました。これは修正され **nullable** はデフォルトで **false** に設定されていま
す。

JBPAPP-4123

制約名が変更された場合、PostgreSQL は **SchemaExport** のドロップに失敗していました。
Hibernate は、テーブルのドロップが試行されるとカスケードを発行することで、この問題を解決し
ています。

JBPAPP-4105

Hibernate プロパティファイル、 **hibernate.cfg.xml**、
persistence.xml に、 **hibernate.dialect** プロパティが見付からない場合、Hibernate
は **StandardDialectResolver** を使い Hibernate 方言を推測しています。データベースのメタ
データから抽出されたデータベース名が **Sybase SQL Server** や **Adaptive Server
Enterprise** の場合、 **StandardDialectResolver** は **SybaseDialect** を返していました。しか
し、 **SybaseDialect** は廃止予定であるため、これは若干問題でした。 **hibernate.dialect** が設
定されていない場合、 **StandardDialectResolver** は正しい方言を使い推測するようになりまし
た。

JBPAPP-4095

HQL クエリをコレクション時に join fetch でスクロールしてもカーソルが最後の結果よりも後ろに
ある場合、 **FetchingScrollableResultsImpl.last()** は最後の結果まで移動しませんでした。
代わりに、カーソルは最後の結果の後ろで留まっていた。結
果、 **org.hibernate.exception.GenericJDBCException: could not perform
sequential read of results** が発生する可能性がありました。この問題の修正により、期待通
り、カーソルが最後の結果に移動するようになっています。

JBPAPP-4088

カラム名がバッククォート (`) で定義された場合、バッククォートの付いたカラム名が参照されてい
ると **@JoinTable** と **@JoinColumn** のマッピングが **AnnotationException** で失敗してしま
した。Hibernate は、テーブルのカラム名ではなく、 **referencedColumnName** とカラムの引用名を
比較するようになっています。

JBPAPP-4022

OneToOne マッピングのエンティティを比較すると `getColumnSpan` は `0` を返していました。同じエンティティを比較している場合でも、`TypeMismatchException`の結果を返していました。カラムスパンが識別子あるいは一意のキータイプにより識別されるように、`getColumnSpan` のアルゴリズムが変更されました。

JBPAPP-3946

`ignoreCase` が `false` に設定されている場合、`LikeExpression` は `ignoreCase` フラグを正しく処理しませんでした。正しいSQLの `property like ?` が構築されていますが、`getTypedValues` 内のフラグが使われておらず小文字の値が常に生成されていました。この問題は修正され、`ignoreCase` が `false` に設定されている場合、小文字、大文字のいずれかが利用されるようになっていきます。

JBPAPP-3737

ステートレスセッションでは、クエリは2フェーズプロセスでオブジェクトをロードします。最初のフェーズで、一時的な永続性コンテキストが空のオブジェクトで生成され、第二フェーズで、オブジェクトのメンバーデータがデータベースから読み込まれます。オブジェクトが関係あるいはコレクションを含んでいる場合、クエリは、セッションの `get()` メソッドに対し再帰呼出しを実行します。これにより、一時的な永続性コンテキストは削除されます。

親オブジェクトが第二フェーズの一部で読み込むべき別の紐付けを含んでいる場合、永続コンテキストに紐付けが見付からないため、Hibernate はアサーションをスローしていました。

新しいメソッド `org.hibernate.engine.PersistenceContext.isLoadFinished()` を導入することで、これは修正されました。このメソッドが `StatelessSession` に一時的な永続コンテキストを消去するタイミングを指示します。

JBPAPP-3565

`org.hibernate.cfg.OneToOneSecondPass` コードのバグにより、一対一のマッピングの処理時に `org.hibernate.PropertyValueException` が発生することがありました。JOIN に `otherSideProperty` が含まれていない場合、`otherSideJoin` が `null` ではなく古いデータを保持してしまうため、この例外が発生していました。`org.hibernate.cfg.OneToOneSecondPass` コードが修正されています。

JBPAPP-3487

Hibernate は `table-pre-class` の継承ストラテジに間違ったエイリアスを生成することがありました(例についてはJIRAを参照してください)。`AbstractCollectionPersister` は、`Column.getAlias(Dialect, Table)` により生成されたキーカラムのエイリアスを使うようになり、`table-pre-class` 継承ストラテジのマッピングが訂正されています。

JBPAPP-2440

キャッシュプロバイダが見付からない場合、`NoClassDefFoundError` が以下のメッセージ付きでスローされていました。

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

接続プロバイダが見付からない場合、以下のメッセージ付きで `HibernateException` がスローされていました。

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

これらのエラーが発生した場合、より分かりやすいメッセージが出されるようになりました。

JBPAPP-2276

JDK 6 に対する **HashMaps** と **HashSets** のイテレーション順番が原因で、union 句あるいは union サブクラスのカラムの順番が JDK 5、6 のいずれが使われているかにより違いました。カラムの順番の変化が union 句全体で一貫しているため、結果クエリは有効でした。しかし、この変化によりパフォーマンスが影響を受けることがありました。HashSet の実装を **LinkedHashSet** に変更することで JDK が違ってても全く同じパフォーマンスが実現できるようになっています。

JBPAPP-2275

以下のクラスが JDK 6 インターフェースを完全実装しないため、Hibernate は以前は JDK 6 にまとめられていました。

- **org.hibernate.jdbc.ResultSetWrapper**
- **java.sql.Blob** を実装するための **org.hibernate.lob.BlobImpl**
- **org.hibernate.lob.ClobImpl**
- **org.hibernate.lob.SerializableBlob**
- **org.hibernate.lob.SerializableClob**

アプリケーションがこのようなクラスから抜けているメソッドを必要とする場合、**NoSuchMethodError** が発生していました。これらのメソッドは実装され、このエラーも発生しなくなっています。

9.3. Remoting 関連で修正済みの問題

JBPAPP-3709

org.jboss.remoting.marshall.MarshallerLoaderHandler がクラスに対してリクエストを受け取った場合、**org.jboss.remoting.loading.ClassBytes** のインスタンスを返していました。**MarshallerLoaderHandler** が希望のクラスを見つけられない場合、返された **ClassBytes** オブジェクトはクラスの byte array 値が null となります。しかし、**org.jboss.remoting.loading.ClassByteClassLoader** は、byte array が null の可能性があること前提にチェックしないため、**NullPointerException** が返されていました。この問題を起こしている **JBREM-1184** は、修正されています。

9.4. Seam 関連で修正済みの問題

JBPAPP-4876

jBPM **ProcessInstance** を Seam のコンポーネントにインジェクションする際、返されるプロセスインスタンスは自動保存に登録されていました。jBPM **ManagedContext** は終了すると **autoSaveProcessesList** に全インスタンスを保存するため、パフォーマンスの問題が発生していました。インジェクション時に **ProcessInstance** を返すコードが変更され、プロセスインスタンスも自動保存に登録されなくなっています。

JBPAPP-4685

JBoss Seam が特定のパラメータ化された JBoss Expression Language (EL) 表現を処理する方法にて、入力のサニタイジングエラーが発見されました。リモートからの攻撃者はこのエラーを使い、

特別に作り上げられた正規言語の添付パラメータを含むURL経由で任意のコードを実行することができました。JBoss Seam フレームワークをベースとする特定のアプリケーションに渡されていました。注意 : Java Security Manager を正しく設定、有効化することで、このエラーを利用するのを阻止できるでしょう(CVE-2010-1871)。

責任を持ってこの問題を報告してくださったGoogle Security Team のMeder Kydraliev 氏に、Red Hat は感謝の意を表します。

JBPAPP-4577

Seam の`getInstanceFromFactory` メソッドを変更し並行処理の問題を修正すると、アプリケーションによっては非常に特定の環境の組み合わせの下でデッドロックの状況に陥ることがありました。元の`getInstanceFromFactory`に対する修正が改正されてから、この問題は発生していません。

JBPAPP-4568

Seam は`/lib` ディレクトリに`activation.jar`のコピーとともに配信されていました。このファイルはJDK 6 demo提供されており、Seam のほうは削除されました。

JBPAPP-4509

`seam-gen` で作成されたSeam アプリケーションのRich:comboBoxコンポーネントには、ドロップダウンの矢印画像がありませんでした。Web ブラウザのコンポーネント表示を制御するのに使われるCSS (カスケードスタイルシート) を変更することで、これは修正されています。矢印の画像は正しく表示されるようになっています。

JBPAPP-4317

Seam は、同じクラス名を持つコンポーネントがクラスパスで2つあると、**"java.lang.IllegalStateException: Two components with the same name and precedence -"**のエラーの後にコンポーネントとクラス名が表示され、機能しなくなっていました。Seam はアップデートされ、重複したクラスを無視するようになっています。同じクラス名を持つ重複コンポーネントがあった場合、最初に出てきたコンポーネントがロードされ、その後出てくる同クラス名を持つコンポーネントは無視されます。

JBPAPP-4273

`MockServletContext` は、JBoss Enterprise Application Platform 5 が仮想ファイルシステム (VFS) を利用しているとの認識を持っていません。そのため、URIへファイルとしてアクセスしようとしていました。`MockServletContext` は、Web アプリケーションのルートを認識し、URL クラスのパスをフェッチすることで、プロトコルが`java.io.File`コンストラクタパラメータの一部を形成しないようにしています。

JBPAPP-4259

Seam-gen テンプレートは、クラスパスに存在しないオブジェクトに`resource://` を使っていました。結果、RichFaces テンプレート (XCSS) をCSS ファイルに変換しようと試行するとエラーが発生していました。Seam-gen で`resource://`を使用しないようにすることで、この問題は修正されています。

JBPAPP-4015

複数のWARファイルを含むEARに対してホットデプロイメントが利用されると、最初にアクセスする Web アプリケーションは正しく機能しますが、それ以降のWeb アプリケーションに対するWeb リクエストが原因で`NullPointerException`が発生していました。これは、ホットデプロイメン

トフィルタ実行時のアプリケーションコンテキストが設定されていなかったためです。Web アプリケーションコンテキスト関連の各オペレーションで、現在の `javax.servlet.ServletContext` を確認するようになり、この問題は解決されています。

JBPAPP-4013

Internet Explorer 8 で Seambay 例の Web サービスページを利用すると、JavaScript のオブジェクトが定義されていないため、ユーザ名、パスワードフィールドに入力した値はリクエストエリアに伝搬されませんでした。これは修正されています。

JBPAPP-4012

Tasks の例は Internet Explorer 8 では正しく機能しませんでした。Resolved Tasks ページで "Undo this Task" を選択すると、このタスクは Tasks ページを返しますが Resolved Tasks ページから消えませんでした。Ajax リクエストキャッシングをオフにすることで、この問題は修正されています。

JBPAPP-4010

CAPTCHA の画像など、リソースサブレットからのリソースはブラウザでキャッシュ化され、`org.jboss.seam.captcha.CaptchaImage` のレスポンスヘッダが正しくないため、再レンダリングが正しく行われませんでした。このレスポンスヘッダが修正され、この問題は発生しなくなっています。

JBPAPP-4009

Seam Mail で、添付はインラインに含まれておらず、メールクライアントによっては添付が表示されませんでした。この問題のパッチは [JBSEAM-4630](#) からバックポートされ、添付は表示されるようになっているはずです。

JBPAPP-4008

最初のパラメータ 'obj' が `BigDecimal` インスタンス、またタイプが `BigDecimal.class` で、`ELSupport.coerceToType` が呼び出されると、`BigDecimal` は倍に変換され、`BigDecimal` に戻されるため、違った値が返される結果となっていました。

アップストリームで対処され、JBoss-EL の Seam 依存性を 1.0_02.CR5 に変更することで、JBoss Enterprise Application Platform 5.1.0 にて修正されました。

特定のタイプに代入可能な場合、`ELSupport.coerceToType` は入力値を返すようになっています。

JBPAPP-4007

`org.jboss.seam.transaction.EjbSynchronizations` は、Java 仕様に準拠していませんでした。http://java.sun.com/blueprints/guidelines/designing_enterprise_applications/transaction_management 記載されているように、`SessionSynchronization` を実装するクラスは、`TransactionsAttributeTransactionAttributeType.SUPPORTS` を持つことができません。`TransactionsAttributeType.SUPPORTS` は、`TransactionsAttributeType.REQUIRED` で置き換えられました。

JBPAPP-4006

`org.jboss.seam.core.Interpolator` は例外を報告せず、限られた情報を警告で出力していました。例外が報告されるようになり、警告の情報が詳細になり説明が分かりやすくなりました。

JBPAPP-4005

マスターノードへの接続に失敗すると `SubscriptionRegistry` オブジェクト

の **topicConnection** フィールドが無効になるため、Seam JMSサポートはスレーブノードに対してフェールオーバーした後、機能しませんでした。それ以降の **SubscriptionRegistry.subscribe()** に対する呼出しは **SpyJMSEException** で失敗し、JMSサブスクリプションは、マスターノードが復活しても機能しなくなっていました。この問題は修正されています。

JBPAPP-4004

"Exception thrown while executing asynchronous call" 処理時に **org.jboss.seam.async.AsynchronousExceptionHandler** によってログされたメッセージに誤植がありました。"Exception" は "ExceptionHandler" であるべきで、これについては修正されています。

JBPAPP-4003

JBoss Enterprise Application Platform 4.x と JBoss Enterprise Application Platform 5.x の間の要件が **ResourceLoader** にあるページレベルの **messages.properties** は **META-INF/classes** に格納される必要があり、さもなければ機能しないというように変更されました。本文書は、これについて明示的ではありません。この問題については修正されています。

JBPAPP-4002

Initialization.redeploy には、新規ロックを作成し即座にロックするというコードが含まれていました。これには保護が全くなく、デバッグモードで問題が発生することがありました。これは、このロック自体を完全に消去することで修正されています。

JBPAPP-3855

SeamSpace **testCreateBlog** は、IBM JDK 1.6上で **NullPointerException** をスローして失敗していました。getIdentifierメソッドはJDK 1.6 で正しく動作するようになっています。

JBPAPP-3769

production 以外のプロファイルでは、**MockResponseStateManager** は **ResponseStateManager#isPostBack** をオーバーライドしませんでした。つまり、Facesリクエスト、Faces 以外のリクエスト両方が **postback** として評価されていたということです。**isPostBack(FacesContext context)** メソッドが **org.jboss.seam.mock.MockResponseStateManager** に追加され、古いメソッドをオーバーライドし、**NonFacesRequest** を許可することで **postback** を認識しています。

JBPAPP-3713

EJB トランザクション内で実行される JAX-RS Web サービスは、Seam HTTP 認証を利用できません。**AuthenticationFilter** で保護された Web サービスへの HTTP リクエストは、2つの **ContextualHttpRequests** に分割し、2つ目で **EjbSynchronization bean** をルックアップできないため、**Component.getInstance("org.jboss.seam.transaction.synchronizations", ScopeType.EVENT)** が **NoSuchEjbException** の例外で失敗していました。

この問題は修正されています。

JBPAPP-3525

org.jboss.seam.ui.jar/MANIFEST.MF の **s.tld** は、JSP Taglibrary 記述子の XML スキーマに従って誤って定義されている属性エントリを含んでいました。これは、Richfaces 3.3.1.SP1 にアップグレードすることで修正されています。

JBPAPP-2384

ユーザがログインしようとする、Seam Chatroom の例は、**EJBTransactionRolledbackException** をスローしていました。この例外は、アプリケーションの機能に影響を与えるものではありませんが、JBoss Cache へのアップデートで修正されません。

9.5. セキュリティ関連で修正された問題

JBPAPP-4144

/status サブレットは認証なしに、**http://localhost:8080/status?full=true**からアクセスできていました (CVE-2008-3273)。

この問題は修正され、このサブレットでは認証が必要となりました。

9.6. Application Server 関連で修正された問題

JBPAPP-4244

hibernate.cfg.xml の解析、デプロイの仕組みは不安定でした。これにより、プロパティは**org.jboss.hibernate.jmx.Hibernate class**にマッピングされる必要があるため、正規のHibernate プロパティ名を利用できませんでした。解析されたkey/value ペアの元のセットは、新たな**setProperties(Set<BaseNamedElement>)**メソッドで、**o.j.h.hmx.Hibernate** オブジェクトに挿入されるようになっています。その後、**start()** メソッドは、そのプロパティセットから設定を構築します。

9.7. 文書関連で修正された問題

JBPAPP-5036

Seam のリファレンスガイドの誤植が数点、修正されています。

JBPAPP-4849

『インストールガイド』の**Windows上でサービスとして実行**に関する項で、誤ったサービス名(長短問わず)や**service.bat** スクリプトの場所をリストアップしていました。これは修正されています。

JBPAPP-4391

Drools 5 のRuleBasedPermissionResolver要件は最新ではありませんでした。結果、文書に誤った情報が記載されていました。古くなった情報は最新の正確なコンテンツで置き換えられています。

JBPAPP-4390

Seam リファレンスガイドは、重複情報が含まれていましたが、これは削除されています。

JBPAPP-4389

Seam リファレンスガイドの'Charting (図表)'の項で、<p:chart> タグの情報が抜けていました。これは「Seam コンポーネントですでにJava で作成された表を表示」という説明つきで修正されています。

JBPAPP-4387

Seam の参照文書にあるソースコードブロックには構文がハイライトされておらず、ソースコードを読むのが困難でした。このコードは、正しいハイライトを使い、更新されています。

JBPAPP-4386

Seam Reference Guide はコードブロック例で付記の番号が抜けていました。コードと関連する凡例をつなぎあわせるのが困難でした。付記を導入することで、これは修正されています。

JBPAPP-3818

『Seam Reference Guide』の"Using Seam with JBoss Tools" は、JBoss Enterprise Application Platform で利用するように改訂されておらず、古い情報とみなされるべきでしたが、

文書は修正、更新されています。

9.8. Messaging 関連で修正された問題

JBPAPP-4135

`ServerConsumerEndpoint.localClose()` は、シャットダウン時に `NullPointerException` をスローしていました。JBoss Messagingの更新版では修正されています。

JBPAPP-3157

unicode マルチバイト文字がメッセージのセクタで利用されると、JBoss Messaging は `TokenMgrError` をスローします。

この問題は修正され、マルチバイト文字は利用可能になりました。

JBPAPP-3003

JBoss Transactionsのバージョンが4.4.0 から4.6.1にアップグレードされた場合、JBoss Messaging の `XAResourceRecoveryTest` は失敗します。このテストはコミット処理中に障害を起こさせ、Recovery Manager がそのトランザクションを回復させるよう促進するものですが、JBoss Transactions 4.6.1ではこのトランザクションは回復されません。

この問題は修正され、新たなテストを作成し、トランザクションの動作の変化に対応するようになっています。

9.9. Web Services 関連で修正済みの問題

JBPAPP-4501

Enterprise Application Platform 5.1.0がMTOMを有効化せずに大容量のファイルを転送処理する方法に問題があるとテストで判明しました。大容量のファイルを転送するにはMTOM を有効にするように注記すべきで、この種のサービスは大量のメモリを消費します。しかし、MTOMを使用しないと決めた場合、以下を実装しApplication Server がクラッシュしないよう阻止することができます。

- このサービスに対して制限つきのリスナを設定します。
- より多くのメモリを配分し、正しいガベージコレクション (GC) ポリシー をJava Virtual Machine に対して設定します。

JBPAPP-4243

Web Services コンテキストの例は機能せず、`ClassCastException` をスローしました。この例は修正され、期待どおりに機能します。

JBPAPP-3785

ヘッダパラメータに追加の名前空間を持つメソッドが、この追加の名前空間を持たないメソッドの後に呼び出された場合、このメソッドがシリアル化される際に間違った名前空間がSOAP オペレーションで利用されていました。この問題は、Java オブジェクトとXMLの間での移行方法が修正されることで解決されており、最後のSOAP メッセージが正しい名前空間で生成されるようになっていきます。

JBPAPP-3019

doc/examples/jboss-web-services-examplesのコンテキストにより、いくつか例外が発生していました。このコンテキストエラーは、JBoss Web Services の例は正しく機能しないという意味です。このコード例は [JBPAPP-2995](#)に従い変更され、JBoss WS の例は正しく実行されます。

JBPAPP-2162

Sun JAXB は、致命的でないエラーを持つメッセージを拒否するべきにも拘らず、警告なしに受け入れます。[JBPAPP-2114](#)に対するパッチはこれを修正し、正しく構成されていないメッセージは拒否されるようになっています。

9.10. RESTEasy関連で修正された問題

JBPAPP-4061

インタフェースが実装ではなくアノテーションがつけられているため、**POJOResourceFactory** で**IndexOutOfBoundsException**が発生していました。アノテーション付きのインターフェースにも対応が追加されました。

9.11. クラスタリング関連で修正済みの問題

JBPAPP-4887

クラスタ化されたHibernate の二次キャッシュを使うようにJPAあるいはHibernate を使うアプリケーションが設定された場合、アプリケーションサーバーのCacheManager Serviceに対する明示的な依存性を導入します。この依存性は、アプリケーションサーバーのサービス依存性の仕組みに記録されておらず、当プラットフォームが停止した場合、CacheManager はアプリケーションがシャットダウンする前にシャットダウンすることができます。これにより、アプリケーションがシャットダウンされるとWARNメッセージがログに残されてしまいます。この問題によりエラーログの警告のみが発生していましたが、HA Server Cache アップグレードにより修正されています。

9.12. Cache 関連で修正済みの問題

JBPAPP-4254

ha-server-cache-jbc は2.0.3.Finalにアップグレードされました。

9.13. Connector 関連で修正済みの問題

JBPAPP-4307

SystemMemoryUsageLoadMetricを使用時、mod_cluster は、他のマシンより多くのシステムメモリを利用するマシンに対するディスパッチリクエストを優先化します。このメトリックは、期待どおりの「使用済メモリ」ではなく「空きメモリ」として「ロード」を返していました。これは修正され、このエラーも発生しなくなりました。

MODCLUSTER-113

`org.jboss.modcluster.demo.servlet.ThreadCountLoadServlet`が`mod_cluster`が削除されていますが、`web.xml` ファイルでは`load-demo.war`に属しているように指定されています。結果、デプロイメントエラーが発生していました。この問題は修正され、存在しないサーブレットは削除されています。

10. 本リリースにおける既知の問題

以下でリリース時で既知の問題を一覧にしています。

10.1. 対応していない JAR

JBoss Enterprise Application Platformの本バージョンでは、以下のライブラリは含まれていますが対応していません。

- `client/hornetq-core-client.jar`
- `client/hornetq-jms-client.jar`

`hornetq RPM` も本リリースでは対応していません。

10.2. 一般的な既知の問題

JBPAPP-5110

グラフィカルインストーラでは、ユーザがカスタムのJAASセキュリティドメインを指定しコンソールや呼出しのセキュリティを確保することができますが、JAASセキュリティドメイン"`jmx-console`"が存在しないにも拘らずTomcatコンソールをJAASセキュリティドメイン"`jmx-console`"で常にセキュリティ確保しています。カスタムのJAASセキュリティドメインを使うには、インストール時の"`JMX Security`"の画面で名前を指定してください。インストーラが終了すると、`jboss-as/server/$PROFILE/deploy/ROOT.war/WEB-INF/jboss-web.xml` ファイルを編集します。`<security-domain>` 要素では、"`jmx-console`"の名前をインストール時に使ったセキュリティドメイン名で置き換えます。こうすることで、インストール時に選択した呼出し、Adminコンソールやその他のコンソールのセキュリティを保つカスタムのJAASセキュリティドメインと同じものでTomcatコンソールのセキュリティ確保を行います。

JBPAPP-5014

Webプロファイルは、JBoss Messagingのリカバリ設定がWebプロファイルの一部でないにも拘らず間違っ含まれています。結果、`server.log`に`ClassNotFoundException`が発生します。

この問題を回避するには、`jbossts-properties.xml`から以下の行を削除してください。

```
<property
  name="com.arjuna.ats.jta.recovery.XAResourceRecovery.JBMESSAGING1"

  value="org.jboss.jms.server.recovery.MessagingXAResourceRecovery;java:/DefaultJMSProvider"/>
```

JBPAPP-4542

セキュリティパッチが原因で、本リリースでは`jboss-sx.jar`、`jboss-javaee.jar`、`jboss-security-spi.jar`、`jboss-sx-server.jar`の場所が変わっています。本リリースでは、`jboss-as/lib/`に移動しています。`jboss-sx.jar`の場所に依存するスクリプトは、この変更を反映するためアップデートする必要があります。

JBPAPP-4665

Resteasy-guice アプリケーションは、`java.lang.SecurityException`が原因でデプロイに失敗します。以下に似たエラーメッセージが表示されます。class
`"org.jboss.resteasy.examples.guice.hello.DefaultGreeter$$FastClassByGuice$$70fd68d0"s signer information does not match signer information of other classes in the same package"`

これは、JBoss Enterprise Application Platformの`cglib.jar`が署名され、`cglib-instrumented`プロキシがアプリケーションのターゲットクラスにある署名者情報ではなく `cglib.jar` の署名者情報を使います。この問題のパッチがJBoss Enterprise Application Platform 5.1.0 とともにリリースされており、[Red Hat Support](#)からダウンロードできます。

JBPAPP-4456

その他のagent-enabled ツールで、フィールド、メソッド、コンストラクタ、あるいはJavaassistでロードされていないものを含むクラスに遭遇する可能性があります。このような場合、Advisor がそのメソッドを操作しようとする、`javassist.NotFoundException` がスローされます。この例外は無視し、フィールドは存在しないとみなしてください。

JBPAPP-4130

`org.jboss.console.plugins.monitor.CreateThresholdMonitorServlet`でモニタを作成している場合、`org.jboss.ejb3.stateless.StatelessDelegateWrapper`は`org.jboss.system.ServiceMBean`に代入できません。このモニタは`<depends>`ではなく属性として作成されるため、再起動のデプロイで失敗します。この問題の回避策とし、作成された`*-service.xml`を手動で変更することでモニタが再起動時に正しくデプロイされます。

JBPAPP-4041

`ServiceMetaDataParser.parseValueFactoryParameter()`は、`<parameter>`要素の最初の子のみを考慮します。そのため、`<null/>`の子要素がキャリッジリターン(「Enter」あるいはホワイトスペース)で囲まれている場合、ノードは文字値として処理され、パラメータは後に正しく代用されません。以下によりエラーが発生しています。

```
<parameter>
  <null/>
</parameter>
```

回避策ですが、`<parameter>`要素にホワイトスペースがないようにしてください。例は以下の通りです。

```
<parameter><null/></parameter>
```

JBPAPP-3929

`java.sql.Date.valueOf`がyyyy-mm-ddの日付形式を解析しようとする、TCK テストは`java.lang.IllegalArgumentException`をスローします。最新のSun JVM、Sun JDK 1.6.0_18-b07における再発バグが原因です(詳細情報は、http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6898593を参照)。この問題の回避策は、Sun JDK 1.6.0_17-b04にダウングレードすることです。

JBPAPP-3766

`org.jboss.mail.MailService`が設定され、JNDI にバインドされる度に、`SessionObjectFactories`が上書きされます。つまり、複数の`MailService`構成が存在する場合、JNDI にバインドされている`MailService`はすべて、セッション内で最近設定された`MailService`のプロパティを持つこととなります。

JBPAPP-3308

分離デプロイメントで、アプリケーションサーバーが**Timer**に紐付いた**info** オブジェクトをデシリアライズしようとするすると**ClassNotFoundException** がスローされます。これは、間違ったクラスローダ(**threadContextClassLoader**) を使って当オブジェクトのデシリアライズが行われているため発生します。

JBPAPP-3280

管理コンソールにアクセスすると、"**Cannot get the process table information without native support**"との**WARN** メッセージがログに生成されます。これらのメッセージを無視しても問題ありません。管理コンソールの今後のリリースで、これらのメッセージは**INFO**に変更される予定です。

JBPAPP-3133

JVM 実装にバグがあると、不完全なデプロイメントがホットデプロイメント時に新しいバージョンのデプロイメントで上書きされる際にJVMがクラッシュしてしまうことがあります。例えば、デプロイメントに配備記述子のフォーマットエラーが含まれている可能性があります。デプロイ時、このエラーが発生し、デプロイメントが失敗します。修正済のデプロイメントをエラーのあるデプロイメントの上からコピーし置き換えると、JVMは、新しいバージョンのホットデプロイメント時にクラッシュする可能性があります。この問題への回避策は、新しく修正されたデプロイメントを**deploy** ディレクトリにコピーする前に、古く不完全なデプロイ済みアーカイブを削除することです。

JBPAPP-3036

GNU bash シェルで実行時 **jboss_init_hpux** スクリプトは、環境変数をピックアップしません。これは、JBPAPP-2036: <https://jira.jboss.org/jira/browse/JBPAPP-2306>に関連しています。

JBPAPP-3035

shutdown.sh スクリプトの **-e** と **-H** の引数を使って直接JVMを終了することができません。

JBPAPP-2998

サーバーをデスクトップアイコンから起動すると、マシンのデフォルトのJava セットが使用されます。これにより、JDK 1.6 以外のJava バージョンがデフォルトに設定されていると例外が発生してしまう可能性があります。JBoss Enterprise Application Platform をデスクトップアイコンから起動を試みる前に、JDK 1.6 がデフォルトのJava バージョンに設定されているよう確認してください。

JBPAPP-2894

hibernate.bytecode.provider システムプロパティを**jpa-deployers-jboss-beans.xml** に設定すると信頼性に欠けます。この問題の回避策として、**-Dhibernate.bytecode.provider=cglib**を**jboss-as/bin/run.conf**の**\$JAVA_OPTS**に追加してください。

JBPAPP-2871

JBQA-2610で記載のように、MySQL 5.0.41の最適化設定を使用していると、CPU 使用率が高くなり、パフォーマンスやトランザクションスループットが下がります。MySQL 5.0.86 へアップグレードし、**JBQA-2610**に記載のように最適化設定を適用し、CPU使用率を下げパフォーマンスを向上させるよう推奨されます。

JBPAPP-2598

JBAS-7049の問題に対する回避策が適用されると、新たな問題が発生します。OpenJDK 6を使ってセキュリティマネージャを作動しているサーバーがアクセス拒否のエラーで起動しなくなります。この問題に関しては現在、既知の回避策はありません。

JBPAPP-2590

IBM JDK 6 を使用している場合、`${JAVA_HOME}/jre/lib/security/java.security` にて定義されている `policy.provider` で問題が存在します。デフォルトでは `org.apache.harmony.security.fortress.DefaultPolicy` が使われていますが、`policy.provider=sun.security.provider.PolicyFile` でなければなりません。この問題への回避策は、手作業で上記の調整を行ってください。

JBPAPP-2576

現在、MySQL JDBC ドライバはXA Recovery を正しく実装しません。

JBPAPP-2027

OpenJDK 6 には `NullPointerException` チェックがないため、OpenJDK 6 を使用しているとサーバマネージャが正しく機能しません。回避策として、`imports/server-config.xml` ファイルの `java.security.debug` 命令文をコメントアウトしてください。

JBPAPP-2027

IBM のディストリビューションである JDK 6 は、`SSLv2Hello` プロトコルに対応しておらず、このプロトコルを使うと `ERROR [AbstractKernelController]` を生成します。このプロトコルの利用は、現在のところ推奨されていません。

10.3. Installer における既知の問題

JBPAPP-5116

カスタムのJAASセキュリティドメインがグラフィカルインストールのプロセス中に作成された場合、カスタムのJAASセキュリティドメインは、独自に指定したユーザや役割のプロパティファイルではなく、`jmx-console-users.properties` と `jmx-console-roles.properties` をそのまま使い続けます。

JBPAPP-4262

Solaris 10 (32-bit、64-bit アーキテクチャ両方)でのインストール時に全ファイルが展開された後、コンソールはパスワードの必要性がないにも拘らず、パスワード要求を表示します。ユーザが `Enter` を押すまでインストールは進みません。

10.4. Hibernate における既知の問題

JBPAPP-4926

保存キューで正しい順番にテーブルが処理されないため、自動増分のプライマリキーを含むカスケードテーブルは正しく保存されません。

テーブル (A) にプライマリキーが割り当てられ、テーブルAに紐付けられた二番目のテーブル (B) がカスケードを有効にし、自動増分のプライマリキーを使います。テーブルBを保存しようとする、Hibernate はテーブルA を先に保存するように検知し、保存キューに追加します。その後、Hibernate はテーブルBを即座に保存するように試行します。テーブルBの保存を成功させるには、テーブルAからプライマリキーの参照を取得する必要がありますが、テーブルAは保存キューにあると `null` となるため、この保存オプションは失敗してしまいます。

この問題を回避するには、テーブルカスケードを無効にし、自動増分プライマリーキーを持つテーブルの保存が試行される前に割り当てられたプライマリーキーを持つテーブルを保存するようにしてください。

JBPAPP-4895

`insert()`の直前に`refresh()`メソッドが呼び出され、二次キャッシュが有効になると、エンティティはこの二次キャッシュに挿入されます。しかし、`refresh()`のコミットに失敗すると、キャッシュされたデータは自動的に削除されません。これは、`refresh()`がエンティティステートをトラッキングしないため発生します。`refresh()`操作は通常、生成されたプロパティに対して使用されるため、この問題に対する1つ目の回避策は、`@Generated`のアノテーションを使うことです。もう1つの回避策として、キャッシュエンティティを手動で削除することです。

JBPAPP-4738

アップデートされたJDBC 3.0 driver for MS SQL Server 2008は、以下のデータタイプに対応しません。

- time
- date
- datetime2
- datetimeoffset

SQL Server の種類に関する詳細情報は、[日付と時刻のデータ型および関数 \(Transact-SQL\)](#)を参照してください。

このような種類の変更が原因で、`org.hibernate.test.hql.ASTParserLoadingTest` テストと `org.hibernate.test.stateless.StatelessSessionTest` テストが失敗してしまいます。この問題の回避策は、以下のように新しい方言を定義することです。

```
public class SQLServer2008Dialect extends SQLServerDialect
{
    public SQLServer2008Dialect()
    {
        registerColumnType( Types.DATE, "date" );
        registerColumnType( Types.TIME, "time" );
        registerColumnType( Types.TIMESTAMP, "datetime2" );
        registerFunction( "current_timestamp",
            new NoArgSQLFunction("current_timestamp",
                Hibernate.TIMESTAMP, false) );
    }
}
```

JBPAPP-4731

`jconn3.jar`を使って挿入操作を実行した場合、null ビット値が0に変換され例外がスローされません。Sybase はビットデータ型の値にnullは許可されておらず、ビット値がnullであれば例外がスローされるべきです。

Hibernate でこの問題を回避し、目的通りnullの値を維持するには、以下のように`boolean`の代わりに`org.hibernate.test.where.NumericTrueFalseType`のタイプをマッピングしてください。

-

```
type="org.hibernate.test.where.NumericTrueFalseType"
```

JBPAPP-4334

jConnectにバグがあると、**FetchingScrollableResultsImpl.isResultSetEmpty()** メソッドが以下を返しResultSet が空であるか決定することになります。

```
currentPosition == 0 && ! getResultSet().isBeforeFirst() && !
getResultSet().isAfterLast();
```

ResultSetが空の場合、**FetchingScrollableResultsImpl.isResultSetEmpty()**は false を返すべきですが、Subase JDBC は true を返します。この問題に対する回避策は現在ありません。

JBPAPP-4175

Hibernate が**ResultTransformer**を使ってキャッシュ可能なクエリを実行すると、**ResultTransformer**を適用後、結果をキャッシュ化しようとします。しかし、データを変更し、Hibernate が読み取れるようにすることも可能です。この場合、結果をキャッシュ化しようとすると、**ClassCastException**が発生します。

JBPAPP-3913

Oracle 11g R2 (RAC)から、デフォルトオプションで作成されたデータベースに従来のテーブルを作成するとこのテーブルに最初の行が挿入されるまで、最初のセグメントが作成されないというように、CREATEテーブルの命令文における動作が変化します。(詳細は、http://download.oracle.com/docs/cd/E11882_01/server.112/e10595/tables002.htm#ADMIN13319を参照してください。)

この変化により、IDジェネレータのクラスが**sequence-identity**の場合、デフォルトで作成された順番は1ではなく、2か4で始まります。この問題の回避策は以下の通りです。

- 別のIDジェネレータを使用するか、あるいは
- 初期化パラメータ**DEFERRED_SEGMENT_CREATION**を**FALSE**に設定することで、Oracle で遅延セグメント作成を無効にします。CREATEテーブルの命令文には新規句が2つ (SEGMENT CREATION DEFERRED および SEGMENT CREATION IMMEDIATE)あり、**DEFERRED_SEGMENT_CREATION**パラメータの値をオーバーライドします。

JBPAPP-3911

Oracle 11g R2 (RACおよびスタンドアローンの両方)では、**LockMode.UPGRADE** (つまり、"for update") を持つ複雑なクエリは、"No more data to read from socket"というエラーを引き起こします。Hibernate のバグではなく、データベース設定に関係しています。詳細は、[Bug Hunting](#)を参照してください。この問題の回避策は、このようなクエリで**LockMode.UPGRADE**の使用を避けることです。

JBPAPP-3483

現在、Hibernate は各クエリの実行時間をログに残しません。JBoss Enterprise Application Platform の今後のバージョンではこの機能が期待されています。

JBPAPP-3284

これは、JBoss Enterprise Application Platformのリリースで廃止予定の cglib の問題です。詳細は、http://sourceforge.net/tracker/index.php?func=detail&aid=2796998&group_id=56933&atid=482368 を参照してください。cglib は、**visitField** メソッドとの問題が原因で、**TransformingClassGenerator**のクラスを変換時

にフィールドのアノテーションを削除します。フィールドではなくゲッターにアノテーションを付けることでこの問題を回避することができます。ただし、JBoss Enterprise Application Platform 5に移植されている負荷の大きいアプリケーションではできない場合もあります。

JBPAPP-3223

Hibernate は現在、HQLの tuple 構文あるいは、tuple 構文に対応していないデータベースのCriteriaをサポートしていません。例えば、データベースが tuple 構文に対応していない場合：

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

Hibernate は以下に変換するべきです。

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

tuple 構文に対応していないデータベース上で構文クエリの使用を避ける以外、この問題に対する回避策は今のところありません。

JBPAPP-3075

データベースの予約キーワードが、Hibernate Validatorアノテーションでプロパティ名として使用されると、(@Min や @Max など)、列名を指定しても **SchemaExport** で例外が発生します。これは、Hibernate がこの指定名を無視するからです。この問題を回避するには、@Column アノテーションでプロパティ名をデータベースの予約キーワード以外にマップします。この問題は Hibernate 4 で修正される予定です。

JBPAPP-3069

デフォルトで ansinull が off に設定されているため、Sybase で `QueryByExampleTest.testJunctionNotExpressionQBE` テストに失敗します。このテストは、(**OR[^](ex) (NOT ex)**)のような選言述語を構築します。これはデータベースのすべてに一致するはずですが、ANSI SQLはNULL 値が含まれる比較をすべてUNKNOWNとして評価するため、一致したものがすべて返されるわけではありません。この問題を回避するには、JDBC URL に次の文字列を追加します。

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

これが可能でない場合は、Hibernate セッション `s` を取得した後に次の Java コード (または同様の Java コード) を実行します。

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

JBPAPP-3034

バッチ挿入ステートメントが要求されると、組み込みクラスは考慮されません。この問題の回避策は2つあります。1つ目の回避策は、組み込みクラスが使用される時に **ORDER_INSERTS** を **FALSE** に設定します。2つ目は、子オブジェクトに `session.save()` を明示的に呼び出し、SQL 挿入要求を強制します。

JBPAPP-3032

TIME や **TIMESTAMP** などのデータベース値を返す際、MySQL はミリ秒やマイクロ秒の単位をサポートしていません。

JBPAPP-3031

Sybase では、トランスレータによって `current_timestamp` テキストがメソッドモードとして認識されません。現在、`current_timestamp` に代わる関数を使用する以外に回避策はありません。

JBPAPP-3030

Sybase では、チェーンされたトランザクションモードの間に `SchemaExport` を使用して保存されたプロシージャを作成することはできません。回避策として、次のコードを新しく保存されたプロシージャ定義のすぐ後に追加します。

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

JBPAPP-3010

`EntityRegionAccessStrategy` および `CollectionRegionAccessStrategy` の `evict(Object)` メソッドは、トランザクション分離を無視してキャッシュからオブジェクトを削除しようとします。JBoss Cache の `removeNode` メソッドはトランザクションの処理を行わないため、現在サポートされていません。

JBPAPP-2945

PostgreSQL 8.3.7 は、`PreparedStatement` に対するクエリタイムアウトの設定をサポートしていません。この制限により、次のようなアノテーションを使用するとクエリに失敗します。

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

JBPAPP-2971

`java.lang.SecurityException` が原因で、Hibernate が `cglib` をバイトプロバイダとして使用するようマップするアプリケーションをデプロイするのに失敗します。以下のようなエラーメッセージが表示されます。

```
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
```

これは、JBoss Enterprise Application Platform の `cglib.jar` が署名され、`cglib` によってインストールされたプロキシは、アプリケーションターゲットクラスの署名者情報ではなく `cglib.jar` 署名者情報を使用することが原因です。

`cglib` は、JBoss Enterprise Application Platform のリリース 5.1 にて廃止予定となっています。

JBPAPP-2867

Sybase は現在 Hibernate の `Blob` や `Clob` をサポートしておらず、Hibernate は Sybase の `text` や `image` データタイプをサポートしていません。この問題を回避するには、Sybase の `text` と `image` タイプにマップするユーザー定義のタイプを作成します。

JBPAPP-2792

カラムがオーバーフローすると、Sybase は新しいエンティティの挿入に失敗します。しかし、例外はスローされず、Hibernate は挿入に失敗したか判断できません。この問題への回避策の1つとして、`jconn3.jar`を使い、Hibernate 設定ファイルに**DYNAMIC_PREPARE=true**を設定します。

```
<property name="connection.url">jdbc:sybase:Tds:aurum:1503/masterDb?
  DYNAMIC_PREPARE=true</property>
```

もう1つの回避策として、`jconn4.jar`を使います。

JBPAPP-2789

宣言されたモデルによって暗黙に固有として定義される列で、重複した `@Column(unique=true)` や `UniqueConstraint(columnnames={...})` アノテーションが使用されると、Oracle や Sybase 上で `SchemaExport` に失敗します。回避策として、重複した `@Column(unique=true)` や `UniqueConstraint(columnnames={...})` アノテーションを削除します。

JBPAPP-2613

DB2 バージョン v9.7 ドライバをプログレッシブストリーミング (デフォルト) で使用すると、**Blob** ロケータや **Clob** ロケータ上の操作に失敗します。この問題には 回避策が2つ考えられます。

- クラスパスかクラスパスの JAR に **DB2JccConfiguration.properties** という名前のプロパティファイルを作成します。次の行をこのファイルに追加し、DB2 におけるプログレッシブストリーミングを無効にします。

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

- DB2 バージョン 9.7 の代わりに DB2 バージョン 9.1 を使用します。

JBPAPP-2408

ID やネーティブ ID ジェネレータを Hibernate で使用する際、DB2 v9.7 ドライバに問題があります。DB2 の `Statement.getGeneratedKeys()` ドライバメソッドが、生成された鍵でなく空の `resultset` を返すため、ネーティブに生成された ID 値をデータベースが返さなかったという内容の例外を Hibernate がスローする原因となっています。この問題は、Data Studio 2.2 でリリースされたバージョンの DB2 9.7 JDBC ドライバで修正され、DB2 のウェブサイトよりダウンロード可能です。Hibernate ではこのバージョンの使用が推奨されます。

JBPAPP-2278

複数のパスより一時エンティティへアクセス可能で、1つ以上のパスが **Save** 操作をカスケードしない場合、**Save** 操作に失敗することがあります。現在の回避策として、以前に失敗した保存操作を実行する前に一時エンティティを保存します。これが可能でない場合の別の回避策として、カスケードマッピングとエンティティーマッピングのいずれかまたは両方を変更しカスケードパスの順番を変更することで、一時エンティティ以外となる必要があるエンティティへカスケードする前に一時エンティティが保存されるようにします。

JBPAPP-1895

型無しパラメータをサポートしない DB2 向けの型無しパラメータを用いてクエリをテストすると、Hibernate は `JoinTest.java` をテストし、`QueryAndSQLTest.java` に失敗します。この問題を回避するには、JIRA に記載されている通り、適切なデータタイプにパラメータがキャストされるようクエリを編集します。

JBPAPP-1613

Sybase で **Boolean** としてマップされたカラムの null 値は、**null**ではなく **0** として永続されます。この問題を回避するには、**type="boolean"**の代わりに **type="org.hibernate.test.where.NumericTrueFalseType"**をマップするようにします。

JBPAPP-1554

Sybase ではサブクエリ選択リストに1つのエントリ (列名や * など) のみを指定できます。生成された SQL には複数のエントリを持つサブクエリ選択リストが含まれるため、collection 要素に複合 ID (composite ID) がある場合は HQL 関数である **elements()** が失敗します。この問題を回避するため、要素に複合キーがある場合は HQL **elements()** を使用しないようにします。代わりに、サブクエリの選択リストに複数のエントリが存在しないよう、HQL を再構成します。

JBPAPP-1545

Sybase でクエリに1つの ANSI 結合と3つ以上の結合が存在し、1つの結合に union サブクラスが関係する場合、**SySQLException** でクエリに失敗することがあります。これは、カラムが結合されたテーブル表現の範囲内がないからです。現在の推奨事項として、union サブクラスが関係する結合フェッチを使用しないことが挙げられます。

JBPAPP-1230

DetachedCriteria がサブクエリとして使用されると、生成された SQL のサブクエリにカラムのエイリアスが含まれます。Sybase ではサブクエリにカラムのエイリアスは許可されていないため、**SySQLException** がスローされます。この問題を回避するには、サブクエリに **DetachedCriteria** を使用する代わりに HQL クエリを使用します。

JBPAPP-1123

結合されたクラスに **@OrderBy** が使用されると (結合テーブルを使用)、「order by」節は実際のテーブル名を使用してカラムを修飾するため、生成された SQL が MySQL、PostgreSQL、Oracle、MSSQL で無効となります。「order by」節はテーブルエイリアスを使用するようにしなければなりません。

JBPAPP-1082

char プロパティを使用しており、このプロパティが初期化されていない場合、Hibernate は **0** に初期化し、**\u0000**文字を含む文字列を永続化します。PostgreSQL では文字列に**\u0000**の文字を含むことができないため、例外をスローします。PostgreSQL を使って**char**カラムにて**\u0000**を永続化するための回避策は、今のところありません。

初期化されていない**char** プロパティに対して、**\u0000**ではなく NULL を永続化するには、プリミティブな**char** 型の代わりに**java.lang.Character** を使います。こうすることで、プロパティ初期化時の例外を防ぐことができます。**\u0000**に設定されている**java.lang.Character** プロパティを永続化しようとする、やはり例外がスローされてしまいます。

JBPAPP-1071

外部キーの制限がプライマリキーのカラムに定義されている場合、**CREATE TABLE** 命令文が生成されると、**SchemaExport** が Null を使用可能として誤って宣言することがあります。MSSQL、DB2、Sybase はプライマリキーのカラムはNull が使用不可でなければならないため、これに失敗します。

この問題の回避策は、どのカラムがNull 使用不可かを明示的に指定してください。例えば、

- **nullable=false** を**@JoinColumn**へ追加します。

- `@ManyToOne`に`optional=false`を追加します。
- `Map`を使った`@CollectionOfElements`の場合、`@Column(name="mapkey", nullable=false)`で`@AttributeOverride`を追加します。
- `@CollectionId` あるいは`@MapKey`の中の場合、`@Column`に`nullable=false`を追加します。

10.5. JBoss Messaging における既知の問題

JBPAPP-5124

JDBC driver Sybase jConnect JDBC driver v7 (ビルド 26502)が搭載されたSybase データベースを使用する場合、このドライバの`PreparedStatement` クラスの制限により、`sybase-persistence-service.xml` 設定ファイルの`MaxParams` 属性に対して481を超える値を設定できません。`MaxParams` 属性に481を超える値を設定している場合、予期せぬエラーが発生する可能性があります。この問題を回避するには、`MaxParams` を481以下の値に設定してください。

JBPAPP-4668 (以前の JBPAPP-3965)

最新のJDBC ドライバ (バージョン11.2.0.1.0) を搭載している場合、JBoss Messaging Test SuiteのテストがOracle 11g R1、R2、RAC 上で失敗します。

- `QueueManagementTest.testDestroyDestinationProgrammatically`
- `QueueManagementTest.testDestroyDestinationProgrammaticallyWithParams`

これらのテストは、`java.sql.PreparedStatement`上の`setFetchSize` メソッドへ渡される`fullSize` キュー設定パラメータに大きな値を使います。JDBC ドライバの問題とは、`executeQuery()` が呼び出されるとメモリが通常よりも多く消費され、`java.lang.OutOfMemoryError`が発生しテストに失敗してしまうという意味です。

JBPAPP-3904

Oracle JDBC driver version 11.1.0.7.0 では、Oracle 11g R1でJBoss Messaging Test Suite を行うと、`SQLException` ("Bigger type length than Maximum") で失敗します。Oracle JDBC driver 11.1.0.7.0の再発バグにより発生しています。Oracle 11g R1、Oracle 11g R2、Oracle RAC 11g R1、Oracle RAC 11g R2でご利用の場合は、Oracle JDBC driver version 11.2.0.1.0 の使用をお勧めします。

JBPAPP-3352

メッセージ駆動 Bean がデフォルト設定 (`useDLQ=true`, `DLQMaxResent=5`) でデプロイされ、メッセージの再配信が促されると、メッセージがデッドレターキューへ送信された後もメッセージが「配信中」状態のキューに残ります。この問題を回避するには、`useDLQ=false` を設定してデッドレターキューの使用を無効にします。

JBPAPP-2033

`jboss.messaging.ServerPeer` JMXインターフェース上の`EnableMessageCounters` を`true`に設定することができません。同じJMX インターフェース上で`enableMessageCounters()` 操作を呼び出すことで、メッセージカウンタを有効化しこの問題を回避します。

10.6. JBoss Web Services における既知の問題

JBPAPP-5021

JBoss WS CXF インストーラの `jbossws-cxf-installer/docs/Install.txt` ファイルは、JBoss WS CXFを誤ってTechnology Previewとしています。`Install.txt` の文章は以下となるべきです。

```
Simply run "ant" from the jbossws-cxf-installer directory
extracted from the jbossws-cxf-installer.zip into your JBOSS_HOME for
your JBoss Enterprise Platform.
```

JBPAPP-4561

JBoss Web Services Native で、MTOM が有効かつ、return-type クラスのDataHandler ゲッタ署名が JavaBean のゲッタ署名と一致しない場合、XOPではなくインラインの文字列として返答が送信されます。JavaBean ゲッタ署名を基に添付のアノテーションをスキャンするため、これが発生します。

JBPAPP-4506

JBoss Web Services Nativeにサービスがデプロイされると、相対URL とインポートするXML Schema Declaration (XSD) は、WSDL サービスコントラクトを起点とする絶対URLとして書き直されます。インポートされたXSD が相対パスを使って二番目のXSD をインポートする場合、このパスは誤ってWSDL サービスコントラクトではなくXSD を起点として書き直されます。この問題の回避策は、XSD ファイルとWSDLファイルをすべて同じディレクトリに移動し、全インポートを変更することです。

JBPAPP-3028

JBoss Web Services 2.0.1.SP から JBoss Web Services Native 3.1.2.SPへのアップグレードは、変更や新規機能が多数含まれています。新機能のいくつかに関して (リソース注入、`@PostConstruct` や `@PreDestroy`への対応など) は追加の処理時間が必要となっており、全体的なパフォーマンスが若干下がってしまいます。

10.7. Remoting における既知の問題

JBPAPP-3392

EJB3 クライアントは、後続の呼出しで既存のソケット接続を再利用しないため、呼出しごとに新規接続が作成されます。

10.8. RESTEasy における既知の問題

JBPAPP-5038

`jettison.jar` ファイルは `jboss-eap-noauth-5.1.0.zip` の `jboss-eap-5.1/resteasy/lib` ディレクトリに含まれていません。この問題を回避するには、Seam ディストリビューションの `jboss-eap-5.1/seam/lib/jettison.jar` に含まれるファイルを使ってください。

JBPAPP-2993

`spring-hibernate-contacts` の例は、`java.lang.IllegalArgumentException: object is not an instance of declaring class` というエラーで失敗します。これは、`ContactServiceImpl` クラスの `getContactById(@PathParam("id") Long id)` メソッドが `@GET` アノテーションを飛ばしているため発生します。

10.9. Seam における既知の問題

JBPAPP-4808

POJO Cache のある Seam アプリケーションで、フィールドレベルセッションの複製が ON になっている場合、**InstantiationException**の問題が発生する可能性があります。POJO Cache のアーキテクチャの特徴が原因となっており、フィールドレベルのセッションを複製するには、クライアントからの特定の機能や動作が必要となります。

回避策がありますので、JSF 設定 (**faces-config.xml**) に以下のコードを追加してください。

```
<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
</context-param>>
```

JBPAPP-4508

<h:dataTable>内の<s:fileUpload>でファイルをアップロードすると、エラーが発生し誤ったコンテンツを持ち同名ファイルが存在するようになります。これは、**getLocalValue()** メソッドにより返されるローカル値が原因で発生します。

JBPAPP-4231

Seam の例をアンデプロイあるいはアンインストールすると、**NullPointerException**が発生します。これによりアプリケーションがシャットダウンすることはありませんが、このエラーは発生するべきではありません。この問題はnestedbooking、dvdstore、itext、excel の例でも見られますが、ui の例では発生しません。

JBPAPP-3546

iText の例は「プログラミングのスキル」のセクションが含まれています。複数選択の一覧にて選択した項目が生成されたPDFに含まれるはずですが、iText のバグのせいで複数の項目が選択されていても、最初の項目のみが生成されたPDFに含まれます。

JBPAPP-2385

SeamのSpring の例は、IBM の仮想マシンでログインフォームがサブMITTされると**IllegalStateException**で失敗します。これはIBM 仮想マシンの不具合が原因です。この問題の対応は、IBM の仮想マシンが修正されるまで保留となります。

JBPAPP-2377

Seamspaceの例は、IBM 仮想マシンで新規ブログの書き込みをサブMITTすると**NullPointerException**で失敗します。これはIBM 仮想マシンの不具合が原因です。この問題の対応は、IBM の仮想マシンが修正されるまで保留となります。

10.10. EJB における既知の問題

JBPAPP-4308

以下のクラスがjboss-ejb3-deployer.jarに追加されています。

```
org/jboss/ejb3/iiop/
org/jboss/ejb3/iiop/BeanCorbaServant.class
org/jboss/ejb3/iiop/EJB3IIOPWebClassLoader.class
org/jboss/ejb3/iiop/IORFactory$1.class
```

```
org/jboss/ejb3/iiop/IORFactory$2.class
org/jboss/ejb3/iiop/IORFactory.class
org/jboss/as/ejb3/deployers/IORDeployer.class
```

しかし、Enterprise Application Platformの 'Web' 設定や Enterprise Web Platform の@Remote* アノテーションはNullPointerExceptionをスローします。これは、Enterprise Web Platform で必須のjboss-iiop.jar がWeb 設定に含まれていないため起こります。

JBPAPP-4692

永続ユニットがEARの外側にあり、bean がEAR 内から永続ユニットを注入しようとする、このユニットはEAR にデプロイできません。永続ユニットが見付からないため、注入ができなくなっています。この動作はEJB3 の仕様では予測され、EJB3 仕様に厳密に準拠するには、この永続ユニットはEAR 内にパッケージ化する必要があります。

JBoss 固有の動作で、永続ユニットがEAR 外でも存在できます。これはJBoss AS サーバープロファイルのdeployers/ejb3.deployer/META-INF/jpa-deployer-jboss-beans.xmlファイルにて設定されています。この該当箇所は以下の通りです。

```
<!--
    Can be DefaultPersistenceUnitDependencyResolver for spec compliant
    resolving,
    InterApplicationPersistenceUnitDependencyResolver for resolving
    beyond EARs,
    or DynamicPersistencePersistenceUnitDependencyResolver which allows
    configuration via JMX.
-->
<bean name="PersistenceUnitDependencyResolver"
class="org.jboss.jpa.resolvers.DynamicPersistenceUnitDependencyResolver"
/>>
```

デフォルトでは、**DynamicPersistenceUnitDependencyResolver** (<https://svn.jboss.org/repos/jbossas/projects/jpa/trunk/deployers/src/main/java/org/jboss/jpa/resolvers/Dy>) が利用されており、JMX Console のMBean で仕様に準拠した動作を制御することができます。仕様に準拠しないJBoss 変数の検索ストラテジーについては、<https://svn.jboss.org/repos/jbossas/projects/jpa/trunk/deployers/src/main/java/org/jboss/jpa/resolver> から参照していただけます。

10.11. Clustering における既知の問題

JBPAPP-3795

0バイトのファイルを含む展開アーカイブがデプロイされると、リモートノードでNullPointerException がスローされ、結果ファームデプロイメントのプロセスが中止します。現在、回避策はありません。

10.12. Connector における既知の問題

JBPAPP-5048

mod_cluster マネージャモジュールのステータスページがワーカーノードのフェールオーバーの後アップデートされません。つまり、このワーカーは有効であるとリストされ、停止した後も利用可能となっているのです。

JBPAPP-4767

AJP の接続が予期せず閉じられた場合、以下のエラーログのメッセージが生成されます。

```
failed [error] [client <ip>] proxy: error processing body
[error] proxy: dialog to 192.168.0.1:8009 (192.168.0.1)
```

これらのエラーは、エラーというよりは警告あるいはデバッグとしてロギングされるべきです。

JBPAPP-3463

アプリケーションがアンデプロイされると、アンデプロイの通知を受け取る前に `mod_cluster` によりアプリケーションサーバーに送られたセッションで 503 - This application is not currently available のエラーが発生してしまう可能性があります。

MODCLUSTER-123

`root` コンテキスト ("/") がデプロイ、有効化されると、別のコンテキストを無効にすることができません。他のコンテキストを `root` コンテキストにフォワードされないよう指定することもできません。

A. 改訂履歴

改訂 5.1.0-5.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
改訂 5.1.0-5 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
改訂 1.0-0 EAP 5.1 GA 向けのリリースノート (ドラフト版)	Fri Sep 03 2010	Laura Bailey