



JBoss Enterprise Application Platform 4.3

リリースノート CP09

JBoss Enterprise Application Platform 4.3 累積パッチ 9 向け
エディション 4.3.9

JBoss Enterprise Application Platform 4.3 リリースノート CP09

JBoss Enterprise Application Platform 4.3 累積パッチ 9 向け
エディション 4.3.9

Jared Morgan
jmorgan@redhat.com

法律上の通知

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.

概要

これらのリリースノートには、製品マニュアルで現在利用可能でない可能性がある、JBoss Enterprise Application Platform 4.3.0.CP09 に関連する重要情報が含まれます。JBoss Enterprise Application Platform 4.3.0.CP09 をインストールする前に、このドキュメントをすべてお読みください。

目次

1. 概要	2
2. インストールおよび移行に関する注記	2
2.1. インストールに関する注記	2
2.2. JBoss Enterprise Application Platform 4.3.0.CP08 からのアップグレード	2
3. 重要な注記	2
3.1. Oracle 9i がサポートされない	2
3.2. 名前構造の変更	3
3.3. サンプル Seam アプリケーションの実行	3
3.4. デフォルトのセキュリティ設定	3
3.5. 組み込み Hypersonic データベース	3
3.6. ソースファイル	4
4. 製品サポートとライセンスに関する WEB リンク	4
4.1. ドキュメンテーション	4
5. 解決された問題	5
6. 既知の問題	10
A. 改訂履歴	16

1. 概要

これらのリリースノートには、JBoss Enterprise Application Platform 4.3.0.CP09 に関連する重要な情報が含まれます。ここでは、新機能、既知の問題、リソース、および他の現在の問題について述べられています。

2. インストールおよび移行に関する注記

この項には、ハードウェアおよびプラットフォームの要件と前提条件を含む、JBoss Enterprise Application Platform バージョン 4.3.0.CP09 のインストールまたはこのバージョンへのアップグレードに関する情報が含まれます。

2.1. インストールに関する注記

アプリケーションに十分な容量以外に JDK と JBoss Enterprise Application Platform をインストールするのに適切なディスク容量が必要です。JDK 1.5 のインストールが稼働している必要があります。サポートされるオペレーティングシステム/JVM の組み合わせ、サポートされるデータベースプラットフォームに関する最新情報と、含まれるコンポーネントのリビジョンレベルに関する現在の情報については、<http://www.jboss.com/products/platforms/application/testedconfigurations> を参照してください。JBoss Enterprise Application Platform のインストールおよび検証の詳細な手順については、http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/ でオンラインで利用可能なインストールガイドを参照してください。

2.2. JBoss Enterprise Application Platform 4.3.0.CP08 からのアップグレード

CSP/JON の使用

累積パッチのインストール

1. パッチを適用するリソースに移動するには **[Browse Resources]** を使用します。
2. **[Software]** タブをクリックします。リソースに適用可能なすべてのパッチとこのリソースの JON により以前にインストールされたすべてのパッチが表示されます。
3. **[Install]** リンクが、JON によりインストールできるすべてのパッチの横に表示されます。このリンクがないすべてのパッチは手動でインストールする必要があります。**[Manually Uninstall]** リンクの横に表示されるすべてのパッチは、新しいパッチをインストールする前に手動でアンインストールする必要があります。

CSP/JON でのパッチのインストールの詳細については、[JON ドキュメンテーション](#)を参照してください。

3. 重要な注記

3.1. Oracle 9i がサポートされない

JBoss Enterprise Application Platform 4.3.0 CP08 以上では、Oracle 9i との互換性がテストされなくなります。

3.2. 名前構造の変更

累積パッチ 7 以降のリリースには機能パック (Feature Pack) が計画されていません。したがって、以前の名前構造 (**seamfp** など) は意味をもたなくなりました。JBoss インストーラーでは SeamFP ではなく Seam2 を参照するようになりました。**seam** のディレクトリの場所は **seam1**、**seam2** のディレクトリの場所は **seam2** になりました。

3.3. サンプル Seam アプリケーションの実行

本番稼働設定を使用するドキュメンテーションに含まれるサンプル Seam アプリケーションを実行することが推奨されます。別の設定を使用すると、メモリの問題が発生することがあります。サーバーが起動するときに使用されるデフォルトの設定 (他の設定が指定されていない場合) が本番稼働設定です。



警告

メモリの問題を回避するには、アプリケーションをデプロイする前にメモリ設定を調整します。これは、ファイル **JBASS_DIST/jboss-as/server/production/run.conf** の **JAVA_OPTS** 設定を以下の推奨値で更新することによって実行できます。

```
-Xms1303m -Xmx1303m -XX:PermSize=256m -XX:MaxPermSize=256m
```

Seam サンプルのデプロイに関する重要な情報と、Seam を使用したアプリケーションの開発に関する詳細情報については、ドキュメンテーションセットに含まれる『Seam リファレンスガイド (Seam Reference Guide)』 (**JBASS_DIST/doc/seam/Seam_Reference_Guide.pdf**) を参照してください。



注記

Enterprise Application Platform 4.3 では、Seam1 と Seam2 の 2 つの Seam バージョンが使用されます。このリリースに含まれる Seam1 バージョンは **1.2.1.EAP_4.3_CP09** です。Seam2 の場合、このリリースに含まれるバージョンは **2.0.2-FP** です。

3.4. デフォルトのセキュリティ設定

RPM パッケージまたは Zip ディストリビューションを使用して JBoss Enterprise Application Platform をインストールする場合は、デフォルトで認証が有効になり、ユーザーアカウントが設定されません。これにより、JBoss Application Server 内のシステムコンポーネントへの許可されないアクセスが回避されます。JBoss Application Server コンポーネントへのアクセスを許可するために利用可能なサービスは手動で設定する必要があります。

これらのサービスを再びアクセス可能にする方法については、『インストールガイド (Installation Guide)』を参照するか、http://kbase.redhat.com/faq/FAQ_107_9963.shtm を参照してください。

3.5. 組み込み Hypersonic データベース

Hypersonic SQL は、デフォルトで「すぐに使用できる」評価および開発専用のデータベース機能を提供します。本番稼働用のデータベースとしては推奨またはサポートされません。

このコンポーネントには技術サポートが利用できません。Red Hat はこのコンポーネントに対するバグの報告を歓迎しますが、特定の時間内にバグを修正することを保証しません。

3.6. ソースファイル

ソース ZIP

- ftp://ftp.redhat.com/redhat/jbeap/4.3.0/en/source/jboss-eap-src-4.3.0.GA_CP09.zip

4. 製品サポートとライセンスに関する WEB リンク

サポートプロセス

https://access.redhat.com/support/policy/support_process.html

製品サポート範囲

<https://access.redhat.com/support/offerings/tam/soc.html>

製品サポートサービスレベル契約

<https://access.redhat.com/support/offerings/production/sla.html>

開発者サポート範囲

<https://access.redhat.com/support/offerings/developer/soc.html>

開発者サポートサービスレベル契約

<https://access.redhat.com/support/offerings/developer/sla.html>

製品アップデートおよび製品別サポートポリシー

http://www.redhat.com/security/updates/jboss_notes/

JBoss エンドユーザーライセンス契約

http://www.redhat.com/licenses/jboss_eula.html

4.1. ドキュメンテーション

同梱されるドキュメンテーションのリストについては、ドキュメンテーションディレクトリの **index.html** ファイルを参照してください。

zip ディストリビューションでは、Platform とその個々のコンポーネントに関するドキュメンテーションが個別の zip ファイル **jboss-eap-docs-`<version>`.zip** として配布されます。

Linux システムでは、ドキュメンテーションは、手動でインストールする必要がある 2 つの rpm で提供されます。これらの rpm は、**jboss-seam-docs-`<version>`.noarch.rpm** と **rh-eap-docs-`<version>`.noarch.rpm** です。Red Hat Enterprise Linux での rpm パッケージのインストールについては、<https://access.redhat.com/kb/docs/DOC-1191> にある Red Hat ナレッジベースの記事を参照してください。

- 『『インストールガイド (Installation Guide)』』は、さまざまなインストール方法を使用して JBoss Enterprise Application Platform をインストールし、そのインストールの検証をする方法について説明します。
- 『『スタートガイド (Getting Started)』』は、プラットフォームのディレクトリ構造と、Application Server およびさまざまな設定セット/サービスのクイックツアーについて詳述しま

す。単純な Web アプリケーションを使用することにより、JSF-EJB3 コンポーネントの使用と Seam を使用して JSF コンポーネントと EJB3 コンポーネントを統合する方法について説明します。

- 『『サーバー設定ガイド (Server Configuration Guide)』』は、すべての管理および設定機能について詳述します。

エラーと追加情報を含むドキュメンテーションの更新バージョン、サンプルアプリケーションコード、リリースノートの最新バージョンには、http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/ から Web 経由でアクセスできます。

5. 解決された問題

このリリースで解決された問題のリストは以下のとおりです。

Hibernate に関する問題

JBPAPP-2440

キャッシュプロバイダーが見つからない場合は、`NoClassDefFoundError` がメッセージ `net/sf/ehcache/CacheException` とともにスローされます。

接続プロバイダが見つからない場合に、`HibernateException` が次のメッセージとともにスローされました: `Could not instantiate connection provider: " + providerClass`

これらのエラーが発生したときにより多くの情報を持つメッセージがスローされるようになりました。

JBPAPP-3223

Hibernate Query Language (HQL) はタプル構文をサポートしませんでした。また、タプル構文をサポートしないデータベースで、データベース基準がタプル構文から解釈可能な構文に変換されませんでした。これにより、データベースクエリーと HQL 検索で問題が発生しました。修正では、HQL のタプル構文がサポートされ、タプルサポートが利用できないデータベースの場合に Hibernate でクエリーが適切な書式に変換されるようになりました。たとえば、以下のようになります。

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

が以下のように変換されるようになりました。

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

JBPAPP-3946

`ignoreCase` が `false` に設定されていない場合に、`LikeExpression` が、`ignoreCase` フラグを適切に処理しませんでした。適切な SQL `property like ?` が構築されますが、`getTypedValues` 内部のフラグは使用されず、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` に設定されているときに小文字または元の文字種が使用されます。

JBPAPP-4088

カラム名がバックティック (```) で定義されている場合に、バックティックがあるカラム名が参照されると、`@JoinTable` マッピングと `@JoinColumn` マッピングが `AnnotationException` で失敗しました。Hibernate はテーブルカラム名ではなくカラムの引用名で `referencedColumnName` を比較するようになりました。

JBPAPP-4095

コレクションで結合フェッチを使用して HQL クエリーをスクロールするとき、カーソルが最後の結果の後に配置される場合 **FetchingScrollableResultsImpl.last()** は最後の結果に移動しません。代わりに、カーソルは最後の結果の後に配置されたままになります。これにより、**org.hibernate.exception.GenericJDBCException: could not perform sequential read of results** が発生することがあります。修正により、カーソルは期待されたように最後の結果に移動します。

JBPAPP-4123

制約名が変更された場合に PostgreSQL が **SchemaExport** をドロップするのに失敗し、制約エラーが発生していました。Hibernate は、テーブルをドロップしようとするときにカスケードを発行するようになり、問題は修正されました。

JBPAPP-4326

ユーザーが非 JDK シリアル化クラスを **SerializableType** でマップし、これらの値を 2 次キャッシュにキャッシュしようとしたときに、**JBPAPP-4471** の変更により、問題が発生しました。**SerializableType** が、**java.io.Serializable** のクラスローダーを、非シリアル化に使用するクラスローダーとして報告しました。ただし、このクラスローダーはクラスローダークラスまたは非 JDK クラスを参照できませんでした。

org.hibernate.util.SerializationHelper.deserialize(InputStream, ClassLoader) のクラスローダーが、非シリアル化する必要があるクラスローダーを解決するようになりました。これを正常に行えない場合は、現在のスレッドからのコンテキストクラスローダーと、Hibernate クラスをロードするクラスローダーが使用されます。

JBPAPP-4599

Hibernate のバグにより、次の例外がスローされます: **java.lang.IllegalArgumentException: alias not found: tbl**. A table object could not be saved to Oracle。一時的な回避策は拡張されたシーケンスを使用することでしたが、これは最適なソリューションではありませんでした。この問題を修正するために Hibernate コードが更新されました。テーブルオブジェクトは期待どおりに Oracle に保存できるようになりました。

JBPAPP-4716

アノテーション **@Tables.values** がタイピングミスであり、正しくは **@Tables.value** でした。これは Hibernate で修正されましたが、修正では古いアノテーションに依存するコードとの後方互換性が失われました。**@Tables.values** が再び追加され、**@Deprecated** でタグ付けされました。これは使用されるべきではありません。

JBPAPP-4927

カスケード保存操作が、割り当てられた PK を持つ親テーブルにリンクされ、自動的にインクリメントされた Primary Key (PK) テーブルを不適切に処理していました。修正により、カスケード保存操作がテーブル間の親子関係を処理できるようになりました。

JBPAPP-5156

Hibernate Deployer (**org.jboss.hibernate.jmx.Hibernate**) は **.har** デプロイメントを制御します。

JBoss Enterprise Application Platform 4.3.0 CP08 では、Hibernate Deployer の **AUTO_CLOSE_SESSION** パラメータと **FLUSH_BEFORE_COMPLETION** パラメータがデフォルトで **true** に設定されていました。Hibernate の古いバージョンから移行する場合、**finally** ブロックで **session.close()** を使用するときエラーが発生していました。Hibernate アーカイブ (**.har**) を

デプロイするときに、"Session was already closed" エラーと "Session is closed" エラーが発生していました。このリリースでは、**AUTO_CLOSE_SESSION** と **FLUSH_BEFORE_COMPLETION** がデフォルトで "false" に設定されます。これにより、後方互換性の問題が回避され、機能が期待どおりの動作に戻ります。

クラスタリングに関する問題

JBPAPP-4473

Single Sign On (SSO) は、ユーザーが一度だけ認証すると、そのセッション中に複数のシステム/Web アプリケーションのリソースにアクセスできる特別なユーザー認証形式です。

クラスタ化された SSO は、SSO セッションをカウントするときに Tomcat または JBoss Web ホストまたはコンテキストを考慮しませんでした。ユーザーが同じ SSO セッション ID で複数の Web アプリケーションを使用する場合、ホストまたはアプリケーションのコンテキストが Clustered SSO セッション数から除外されるため、ユーザーはログアウトされることがあります。クラスタ化された SSO 対話は JBoss Cache (JBC) で Tomcat と JBoss Web ホストおよびコンテキストを含む接続可能なコンポーネントにリファクタリングされるようになりました。リファクタリングされた SSO では、ホストとコンテキストの問題が解消されます。

一般的な問題

JBPAPP-4917

Xalan は、XML ドキュメントを HTML、テキスト、または他の XML ドキュメントタイプに変換する XSLT (Extensible Stylesheet Language Transformation) プロセッサです。原因:

Xalan の XMLReaderManager (`org.apache.xml.utils.XMLReaderManager`) クラスは ThreadLocal で XMLReader を作成しますが、XMLReader が終了したときに ThreadLocal (null に設定される) から XMLReader を削除しませんでした。

これにより、Xalan を JDK 1.5.0_6 環境で実行する場合に永久的なリークが発生しました。リークの規模は、XMLReader が解析する必要がある XML ファイルのサイズと、サーバーの負荷によって悪化しました。修正および結果:

Xalan 2.7.0 に対して Xalan が ThreadLocal で XMLReader を処理する方法が変更され、XSLT 変換でエラーのある追加ノードが作成されることが回避されます。

Seam1 の問題

JBPAPP-4260

Seam-gen テンプレートが、クラスパスにないオブジェクトの `resource://` を使用していました。これにより、RichFaces テンプレート (XCSS) を CSS ファイルに変換しようとするときにエラーが発生していました。Seam-gen は `resource://` を使用しなくなり、この問題は解決されました。

JBPAPP-4316

クラスパスで重複する Seam コンポーネントが無視され、初期化中に `IllegalStateException` が発生していました。重複するクラスは無視され、同じコンポーネント名と優先順位を持つ異なるクラスが適切に検出されるようになりました。これらの条件が満たされない場合は、期待された例外がスローされます。

JBPAPP-4803

JBoss Seam がパラメータ化された特定の JBoss Expression Language (EL) 表現を処理するときに入力サニタイゼーションの脆弱性が存在しました。リモート攻撃者はこの脆弱性を使用して JBoss Seam フレームワークに基づいて特定のアプリケーションに提供される URL (アペンドされ、特別に作成される表現言語パラメータ) を介して Seam コンポーネントに対してパラメータがないメソッドを実行できました。このリリースには、EL 表現を使用した悪意のある対話を防ぐ拡張されたセキュリティが含まれます。

Red Hat は、この問題を責任を持って報告された Google Security Team の Meder Kydyraliev 氏に謝意を表します。

Seam2 の問題

JBPAPP-3125

JBoss Enterprise Application Platform の Drools バージョンが SOA 4.3 CP02 で使用されたバイナリにアップグレードされ、jBPM バージョンが v3.2.9 にアップグレードされました。

JBPAPP-4060

一時停止の時間とテストメソッド違反の問題により、Quartz サンプルテストが RHEL4 と IBM JVM 1.6 で失敗しました。テストスイートは各テストメソッドに対して個別のアカウントを使用して個別のテストメソッドを切り離し、一時停止の時間は 50ms から 1000ms に増加しました。この結果、テストスイートでテスト同士が十分に切り離されるようになりました。

JBPAPP-4064

RichFaces の不正確なバージョンが Seam 2.0.2 にバンドルされ、一部の RichFaces コンポーネントが失敗するか、不正に動作していました。これは、RichFaces (3.3.1) の正しいバージョンを Seam にバンドルすることにより修正されました。

JBPAPP-4492

PassivatedEntity.passivateEntity() メソッドの問題により、パッシブ化フェーズで会話コンテキストに余分な属性が追加される可能性があります (**concurrentModificationException** がスローされます)。この問題は修正され、 **concurrentModificationException** がスローされなくなりました。

JBPAPP-4686

JBoss Seam がパラメータ化された特定の JBoss Expression Language (EL) 表現を処理するときに入力サニタイゼーションの脆弱性が存在しました。リモート攻撃者はこの脆弱性を使用して JBoss Seam フレームワークに基づいて特定のアプリケーションに提供される URL (アペンドされ、特別に作成される表現言語パラメータ) を介して Seam コンポーネントに対してパラメータがないメソッドを実行できました。このリリースには、EL 表現を使用した悪意のある対話を防ぐ拡張されたセキュリティが含まれます。

Red Hat は、この問題を責任を持って報告された Google Security Team の Meder Kydyraliev 氏に謝意を表します。

JBPAPP-5312

Bash スクリプト **seam/seam.sh** が、一部の Linux システムでのみ実行可能パーミッションを持ちます。これは、ディストリビューションに含まれる異なる zip ユーティリティ実装によって引き起こされます。これは、Fedora 12 と Red Hat Enterprise Linux 4 で修正され、実行可能パーミッションが **seam/seam.sh** に適切に割り当てられるようになりました。この修正は Ubuntu などの他のオペレーティングシステムで使用される zip ユーティリティには反映されていません。

JBPAPP-5379

Internet Explorer 8 (IE8) で Seambay サンプルの Web サービスページを使用したときに、ユーザー名フィールドとパスワードフィールドで入力された値が要求領域に伝播しませんでした。Seambay を使用するカスタマーはログイン画面以降に進むために要求領域で値を手動で更新する必要がありました。Seambay サンプルでは、IE8 に存在する異なるオブジェクトモデルが許可され、Microsoft IE Document Object Model (DOM) 実装がチェックされるようになりました。

ドキュメンテーションの問題

JBPAPP-2950

Windows ユーザー向けの Java Runtime Environment (JRE) の指定手順で、JRE bin ディレクトリへのパスが環境変数が必要であると間違っていると示されていました。『『インストールガイド (Installation Guide)』』の「『Windows 向け JRE の指定 (Specifying JRE for Windows)』」の手順で、ユーザーは JRE パスを JRE インストールルートディレクトリに設定するよう指示され、Windows で環境変数を設定する手順が記載されるようになりました。

JBPAPP-3207

『『メッセージングユーザーガイド (Messaging User Guide)』』に明確な手順が含まれないため、**All** および **Production** サーバードプロファイルを使用して非クラスタ化の Messaging サンプルを実行するときに問題が発生しました。ほとんどの Messaging サンプルは非クラスタ化サーバードプロファイル (**All** と **Production** を除くすべてのプロファイル) で実行されるよう設計されています。サンプルをクラスタ化セットと非クラスタ化セットに分けることにより、該当する章の内容が改善され、ユーザーは設定手順について、各サンプルに同梱される readme ファイルを参照するよう指示されます。

JBPAPP-4138

『『Seam リファレンスガイド (Seam Reference Guide)』』と『『Seam2 リファレンスガイド (Seam2 Reference Guide)』』に、**mail-ra.rar** コンポーネントを Seam プロジェクトから JBoss Application Server に手動で変更するのに間違った参照が含まれていました。これらの指示に従うと、カスタマーの本番稼働環境で Red Hat によりサポートされない機能が実装される可能性があります。両ガイドの項「電子メールの受信 (Receiving Emails)」で、中核的なコンポーネントを変更した結果とカスタマーサポートへの影響に関する記述が修正されました。

JBPAPP-4638

JBoss Enterprise Application Platform 向けのインストール手順の RNH に関する項では、Supplementary / Extra チャネルを有効にするよう指示されませんでした。**java-1.5.0-ibm** パッケージと **java-1.5.0-ibm-devel** パッケージがインストールされていない場合に、インストールが失敗しました。『インストールガイド (Installation Guide)』は Supplementary / Extra チャネルをインストールし、正しいパッケージを選択する手順を含むよう更新されました。更新された手順に従うと、RPM インストールは成功します。

JBPAPP-5311

『JDK6 互換性に関するノート (JDK6 Compatibility Notes)』ガイドには、JAX-WS サポートでユーザーが特定の **.jar** ファイルを JBoss Enterprise Application Platform ディレクトリの **jboss-as/lib/endorsed** ディレクトリに移動する必要があることが記述されていました。必要なファイルはデフォルトで **/endorsed** ディレクトリに保存されるようになりました。JAX-WS は動作するために特別な設定を必要としなくなりました。

EJB に関する問題

JBPAPP-4029

user-type-mapped タイプが EJB-QL 集約関数で使用されたときに、結果セットリーダーから Class Cast Exception (CCE) がスローされました。EJB-QL 関数は正しく実装され、**user-type-mapped** タイプを使用できるようになりました。

6. 既知の問題

リリース時点の既知の問題のリストは以下のとおりです。

一般的な既知の問題

JBPAPP-3050

JBoss Enterprise Application Platform が **-b <hostname>** フラグを使用して起動されるとき、**jboss.web:name=ajp-<hostname>/<ip_address>-8009,type=ThreadPool,*** でフィルタリングしても結果が返されません。この結果、**deploy/snmp-adapter.sar/attributes.xml** の設定は Bean を見つけることができないため、SNMP を使用してデータを他のモニタリングツールにエクスポートすることはできません。

以下の定義は機能しません。

```
<mbean name="jboss.web:name=ajp-<hostname>/<ip_address>-8009,type=ThreadPool" oid-prefix=".1.2.3.4.1">
```

以下の定義は期待どおりに機能します。

```
<mbean name="jboss.web:name=ajp-<ip_address>-8009,type=ThreadPool" oid-prefix=".1.2.3.4.1">
```

JBPAPP-4172

IBM JDK 1.5 を 64 ビットの Red Hat Enterprise Linux 5 でインストールしようとする、Java バイナリへのリンクが切断されます。これにより、アプリケーションサーバーが起動するのが防がれます。この問題の回避策は代わりに Sun JDK 1.6 を使用することです。

JBPAPP-4917

probe.sh が終了すると、以下のように、該当する文字列に一致する用語のリストと、**-match <string>** パラメータに一致した応答の数を反映する概要の行が生成されます。

Total responses=1, 1 matches, 3 non-matches

タイムアウト時にマルチキャストソケットを閉じるよう設定されたスレッドによって、マルチキャストソケットが例外を生成し、概要の行が表示される前に返されるため、この概要の行は表示されません。

EJB に関する既知の問題

JBPAPP-3908

ValidateDTD が EJB デプロイヤーで **true** に設定された場合、メッセージ駆動 Bean (**JMSCContainerInvoker** の Bean など) のプロパティが **jboss_4_2.dtd** に存在しないため、**standardjboss.xml** の検証に失敗します。この問題を回避するには、**standardjboss.xml**

の `jboss_4_2.dtd` への参照を削除します。

JBPAPP-5190

クラスタ化された環境では、`getCallerPrincipal` メソッドは `ejbActivate` から呼び出されたときに `IllegalStateException` を発生させました。これは、`StatefulHASessionPersistenceManager.synchroSession()` が、最初の作成メソッドで `getCallerPrincipal` を許可しないためです。この問題の回避策は `ejbActivate` から `getCallerPrincipal` を移動することです。

メッセージングに関する既知の問題

JBPAPP-2033

`jboss.messaging.ServerPeer` JMX インターフェースの `EnableMessageCounters` は、`true` に設定できません。メッセージカウンターを有効にする回避策は、同じ JMX インターフェースで `enableMessageCounters()` 操作を呼び出すことです。

JBPAPP-3206

ドキュメンテーションバンドルに含まれる JBoss Messaging のサンプルには、不正確な設定情報が含まれ、これらのサンプルは JBoss Enterprise Application Platform で動作しません。修正されたサンプルアーカイブは、JBoss Enterprise Application Platform 4.3.0 CP09 (http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/) に対して使用できません。

使用しているディストリビューションの `/docs/examples/jboss-messaging-examples` ディレクトリを、このアーカイブの抽出されたコンテンツで置き換えてください。これらのサンプルを手動で修正する場合は、詳細情報が JIRA で利用可能です。Web サービスのサンプルは動作しないことに注意してください。

JBPAPP-3352

メッセージ駆動の 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求められた場合、メッセージは、配信が終了したメッセージのキューに配信されたあとも「配信中」ステータスのキューに残ったままになります。

この問題の回避策は配信されたメッセージのキューを無効にすることです: `useDLQ=false`。

JBPAPP-5420

2 つの JBoss Messaging Test Suite のテストが、Oracle 11g R1、R2、および RAC (JDBC ドライバ `v11.2.0.1.0` および `v11.2.0.2.0` を使用) で失敗します。

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

これらのテストは `fullSize` キュー設定パラメータの大きい値を使用し、値は `java.sql.PreparedStatement` の `setFetchSize` に渡されます。JDBC ドライバの問題は、`executeQuery()` が呼び出されたときに通常よりも大きいメモリ量が消費され、テストが失敗することです。

JBPAPP-5423

Oracle JDBC ドライババージョン 11.1.0.7.0 では、Oracle 11g R1 で JBoss Messaging Test Suite が SQLException ("Bigger type length than Maximum") を発生させて失敗します。これは Oracle JDBC ドライバ 11.1.0.7.0 の問題によって引き起こされます。

この問題を回避するには、Oracle 11g R1、Oracle 11g R2、Oracle RAC 11g R1、および Oracle RAC 11g R2 で Oracle JDBC ドライババージョン 11.2.0.2.0 を使用します。または、Topic または Queue の **FullSize** 属性 (デフォルト値は 200000) に小さい値 (65000 など) を設定します。

Hibernate の既知の問題

JBPAPP-1545

Sybase では、クエリーに 3 つ以上の結合とともに ANSI 結合が含まれ、1 つの結合が union サブクラスに関連する場合に、カラムが結合されたテーブル表記の範囲内でないため、クエリーが **SySQLException** で失敗することがあります。現時点で推奨する解決策は、union サブクラスに関連する結合フェッチを使用することを回避することです。

JBPAPP-1546

Sybase を使用する場合は、**SchemaExport** を使用してチェントランザクションモードでストアドプロシージャを作成できません。この場合の推奨される回避策は、新しいストアドプロシージャの定義後に以下のコードを追加することです。

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

JBPAPP-1554

Sybase では、サブクエリー選択リストで 1 つのエントリ (たとえば、カラム名または '*') しか許可されません。コレクション要素に複合 ID が含まれる場合、HQL 関数 **elements()** は失敗します。これは、生成された SQL に複数のエントリを持つサブクエリー選択リストが含まれるためです。回避策は、要素に複合キーが含まれる場合に HQL **elements()** を使用することを回避することです。代わりに、サブクエリーの選択リストに複数のエントリが含まれないよう HQL を再び作成します。

JBPAPP-1613

Sybase で **Boolean** とマップされたカラムの null 値が、**null** の代わりに **0** として永続化されます。この問題の回避策は **type="boolean"** の代わりに **type="org.hibernate.test.where.NumericTrueFalseType"** をマップすることです。

JBPAPP-1709

JBoss Enterprise Application Platform 4.x の現在のバージョンに同梱された **ejb3-persistence.jar** のバージョンは不正確です。Hibernate Entity Manager には現在 **ejb3-persistence.jar 1.0.0.GA** が同梱されますが、**ejb3-persistence.jar 1.0.1.GA** を使用する必要があります。JAR の 1.0.0.GA バージョンと 1.0.1.GA バージョン間には 2 つの変更があります。

1. JPA 仕様では、クラス名にタイピングエラーがある値を持つ定数が定義されています。

```
javax.persistence.Persistence.PERSISTENCE_PROVIDER =
"javax.persistence.spi.PersistenProvider"
```


JBoss Enterprise Application Platform に含まれる JAR にはこのタイピングエラーが含まれないため、この JAR は JPA 仕様に準拠しません。詳細については、<http://opensource.atlassian.com/projects/hibernate/browse/EJB-321> を参照してください。

2. `javax.persistence.Query.getSingleResult()` の Javadoc には、結果がない場合に、`EntityNotFoundException` がスローされることが記述されています。これは `NoResultException` と記述するべきです。

JBPAPP-1722

エンティティがカラムをオーバーフローする場合に Sybase は新しいエンティティを挿入するのに失敗します。ただし、例外をスローしないため、Hibernate は挿入が失敗したことを認識できません。この問題を回避するには、アプリケーションでエンティティプロパティを検証して値が基礎となるカラムをオーバーフローしないようにします。以下のいずれかの回避策を使用してください。

1. `jconn3.jar` を `DYNAMIC_PREPARE=true` オプション (Hibernate 設定ファイル) で使用し、適切な JDBC を指定します。例を以下に示します。

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

<property
  name="connection.driver_class">com.sybase.jdbc3.jdbc.SybDriver</
property>
```

2. `jconn4.jar` を使用し、適切な JDBC ドライバ (Hibernate 設定ファイル内) を指定してください。例えば、以下ようになります。

```
<property
  name="connection.driver_class">com.sybase.jdbc4.jdbc.SybDriver</
property>
```

JBPAPP-1895

DB2 v91 (型指定のないパラメータをサポートしない) に対して型指定のないパラメータでクエリーをテストする場合は、Hibernate のテスト `JoinTest.java` と `QueryAndSQLTest.java` が失敗します。UPPER 関数はパラメータのデータ型に非常に厳密です。パラメーターマーカーはプリコンパイルされたときにデータ型を持たないため、DB2 は使用するデータ型を認識せず、エラーを発生させません。

この問題を回避するには、パラメータが適切なデータ型にキャストされるようにクエリを変更します。

```
UPPER(CAST(? AS CHAR(10)))
UPPER(CAST(? AS VARCHAR(50)))
```

JBPAPP-2315

Hibernate は、アプリケーションが `java.sql.Types.LONGVARCHAR` または `java.sql.Types.CLOB` カラムでデータを Java String として処理できるようにするプロパティタイプを提供します。 `java.sql.Types.LONGVARBINARY` または `java.sql.Types.BLOB` を Java `byte[]` として処理することもできません。これらの機能は JBoss Enterprise Application Platform 6 ストリームで期待されます。

JBPAPP-2839

エントリがシリアル化解除された **PersistenceContext** から取得されたときに、インストルメント化されたエンティティのインターセプターを再挿入するのに失敗したため、**NullPointerException** が発生します。

JBPAPP-2945

PreparedStatement に対するクエリタイムアウトの設定は PostgreSQL 8.3.7 でサポートされていません。この制限のため、クエリーは以下のようなアノテーションを使用する場合に失敗します。

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

JBPAPP-3069

QueryByExampleTest.testJunctionNotExpressionQBE は、(OR^ (ex) (NOT ex)) などの論理和述語をビルドします。ansinull がデフォルトで無効に設定されているため、テストが Sybase で失敗します。これはデータベースのすべての一致するはずですが、ANSI SQL が NULL 値に関連するすべての比較を UNKNOWN と評価したため、すべての一致対象が返されるわけではありません。以下の回避策が存在します。

1. 次の文字列が JDBC URL に追加される: **?SQLINITSTRING=set ansinull on**
2. Hibernate セッションの取得後に次の Java コード (または同様のもの) を実行:
s.connection().createStatement().execute("set ansinull on");

JBPAPP-3075

データベース予約キーワードが Hibernate Validator アノテーションでプロパティ名として使用された場合 (@Min や @Max など) は、カラム名を指定しても **SchemaExport** で例外が発生します。これは、Hibernate が使用された名前を無視するためです。回避策は、プロパティ名を、@Column アノテーションでデータベース予約キーワードでないものにマップすることです。この問題は、Hibernate 4 で修正される予定です。

JBPAPP-3913

Oracle 11g R2 (RAC とスタンドアロンの両方) で、シーケンスは 1 ではなく 2 で開始されることがあります。原因は分析中です。現時点では、この問題に対して 2 つの回避策が存在します。

1. 別の ID ジェネレータークラスを使用します。
2. 初期化パラメータ **DEFERRED_SEGMENT_CREATION** を **FALSE** に設定して、Oracle の遅延セグメント作成を無効にします。**CREATE TABLE** ステートメントで新しい句 **SEGMENT CREATION DEFERRED** と **SEGMENT CREATION IMMEDIATE** を利用できます。これらの句は、**DEFERRED_SEGMENT_CREATION** 初期化パラメータの設定よりも優先されます。

JBPAPP-4334

jConnect にバグがあるため、**FetchingScrollableResultsImpl.isResultSetEmpty()** メソッドが、ResultSet が空白であるかどうかを調べるために以下のものを返します。

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

ResultSet が空白の場合、**FetchingScrollableResultsImpl.isResultSetEmpty()** は false を返しますが、Sybase JDBC は true を返します。現時点では、この問題の回避策はありません。

JBPAPP-4377

Hibernate が **ResultTransformer** を使用してキャッシュ可能なクエリを実行した場

合、**ResultTransformer** の適用後に結果のキャッシュが試行されます。ただし、データは変更され、Hibernate がデータを読み取れない場合があります。この場合、結果をキャッシュしようとすると、**ClassCastException** が発生します。

JBPAPP-4465

Oracle 11g R2 (RAC とスタンドアロンの両方) で、LockMode.UPGRADE (つまり、「更新用」) を含む複雑なクエリーにより、"No more data to read from socket" エラーが引き起こされることがあります。この回避策はこのようなクエリーで LockMode.UPGRADE を使用しないことです。この原因は分析中です。

JBPAPP-5152

refresh() メソッドが **insert()** の直前に呼び出され、2 次キャッシュが有効な場合は、エンティティが 2 次キャッシュに挿入されます。ただし、**refresh()** が正常にコミットしない場合は、キャッシュされたデータは自動的に削除されません。この理由は、**refresh()** がエンティティステータスを追跡しないためです。生成されたプロパティに対しては通常 **refresh()** 操作が使用されるため、この問題の 1 つの回避策は **@Generated** アノテーションを使用することです。別の回避策はキャッシュを手動で削除することです。

Seam1 に関する既知の問題

JBPAPP-4153

ajax4jsf を使用する Java Server Faces アプリケーションは IBM JDK 1.6 と動作しません。この問題の回避策は、以下の環境変数を設定して OSCache などの異なるキャッシュを使用することです。

```
org.ajax4jsf.cache.CacheFactory=org.ajax4jsf.cache.OSCacheCacheFactory
```

または、カスタマーは **components.xml** で次のパラメータをキャッシュを無効にできます:

enable-cache=false

components.xml は Seam 内の以下の場所に保存されます。

- WAR の **WEB-INF** ディレクトリ
- JAR の **META-INF** ディレクトリ
- **@Name** アノテーションを持つクラスを含む任意の JAR ディレクトリ。

Seam2 に関する既知の問題

JBPAPP-4065

複数の Seam 2.0.2 ユニットおよび統合テストが Red Hat Enterprise Linux の IBM JVM で失敗します。影響を受ける例としては、DVD ストア、予約、ネストされた予約、登録、seamdiscs の例などがあります。

JBPAPP-5350

seam-gen により生成され JBDS にインポートされたプロジェクトに対して、JBDS 4 で Seam Expression Language (EL) コードの完了が機能しません。この問題を回避するには、生成されたプロジェクトで Seam Support を有効にし、再びアクティブにします。

A. 改訂履歴

改訂 4.3.9-2.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
改訂 4.3.9-2 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
改訂 4.3.9-100 Enterprise Application Platform 4.3.0CP09 リリース向けの既知の問題と解決済みの問題が記載されました。	Tue Nov 30 2010	Jared Morgan
改訂 4.3.9-24 候補リリースドラフトであり、以下の項目が追加されています。 JPPAPP-5420 JBPAPP-5423	Tue Nov 23 2010	Jared Morgan
改訂 4.3.9-23 候補リリースドラフト。	Thu Nov 11 2010	Jared Morgan