# JBoss Enterprise Application Platform 5

## リリースノート 5.1.2

JBoss Enterprise Application Platform 5.1.2 の使用向け

エディッション 5.1.2

Last Updated: 2017-10-26

# JBoss Enterprise Application Platform 5 リリースノート 5.1.2

JBoss Enterprise Application Platform 5.1.2 の使用向け
エディッション 5.1.2

Jared Morgan

Eva Kopalova

Russell Dickenson

## 法律上の通知



## 概要

これらのリリースノートには、JBoss Enterprise Application Platform 5.1.2 に関する重要な情報が含まれています。これらの情報は現在、製品マニュアルに含まれていない可能性があるため、JBoss Enterprise Application Platform 5.1.2 をインストールする前にリリースノート全体に目を通すようにしてください。

# 目次

# 第1章 はじめに

本リリースノートには、JBoss Enterprise Application Platform 5.1.2 関連の重要な情報が含まれており、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の説明がなされています。

## 1.1. JBOSS ENTERPRISE APPLICATION PLATFORM について

JBoss Enterprise Application Platform は次段階に向け進化したオープンソースのエンタープライズソフトウェアです。純粋な Java Platform で高性能な Web 2.0 アプリケーションを開発するための強力なツールです。

JBoss Seam、Hibernate、CXF Web Services、JBoss Cache、JBoss Messaging などの最高のオープンソースフレームワークを統合することで、本プラットフォームはオープンソースコミュニティーによるイノベーションを有効活用しています。また、JBoss Enterprise Application Platform のテストやサポート全般は Red Hat が行っており、主要なエンタープライズハードウェアやソフトウェア製品の多くで機能認定を受けています。

## 1.2. 本リリースについて

JBoss Enterprise Application Platform 5.1.2 はマイナーリリースとなっています。マイナーリリースでは、以前のパッチおよび累積パッチ (CP: Cumulative Patch) のコンテンツを集約し、場合によって新機能を追加します。後続のパッチや累積パッチについては、それ以前のマイナーアップデートがインストールされていることが前提となっています。

Red Hat は、製品のメジャーバージョンのライフサイクルにわたりビジネスレベル相応の努力を行い、マイナーリリース、非同期パッチすべてにおいて API レベルの互換性を維持します。例えば、JBoss Enterprise Application Platform 5.1.2 は、JBoss Enterprise Application Platform 5 の最初のリリースである JBoss Enterprise Application Platform 5.0.0 と API レベルの互換性を維持します。このルールへの例外として考えられるのは、極めて重要なセキュリティー問題に対応するための修正などです。

技術プレビューだった PicketLink (JBoss Enterprise Application Platform 5.1.1 で v1.0.3 を提供) が v2.0.1 GA となりました。

JBoss Enterprise Application Platform 5.1.2 がリリースされたため、JBoss Enterprise Application Platform 5 を使用しているお客様は 5.1.2 へアップデートするようにしてください。

『JBoss Enterprise ミドルウェア製品アップデートおよびサポートの方針』に関する詳細情報は、http://www.redhat.com/security/updates/jboss_notes/ を参照してください。

## 1.3. HORNETQ

HornetQ は**フレッシュ**インストール (アップグレードではない) のデフォルトの JBoss Messaging の代わりに使用できる JBoss Application Platform のオプションの JMS プロバイダです。Boss Enterprise Application Platform のリリース後、非同期的に Red Hat Network (RHN) とカスタマーポータルより入手可能となり、同じチャンネルで発表されます。本リリースより HornetQ は技術プレビューから GA v2.2.10-Build2 となりました。

**警告**

既に JBoss Messaging が JBoss Enterprise Application Platform 5.1.2 に設定されていたり、HornetQ TP が JBoss Enterprise Application Platform 5.1.1 に設定されている場合、HornetQ パッケージ (RPM) をインストールしないでください。このような環境で HornetQ をインストールしようとすると、パッケージの競合が発生する原因となります。HornetQ のインストールはフレッシュインストールの場合のみ可能です。

## 1.4. 除外、削除、廃止予定のアイテム

**定義**

**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となっているアイテム

**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

**削除されるアイテム**

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

JBoss Enterprise Application Platform 5.1.2 はマイナーリリースです。メジャーリリースと全マイナーリリースで互換性が維持されます。つまり、5.x リリースはすべて初期リリースの 5.0.0 とバイナリ互換を維持することになります。

### 1.4.1. 廃止予定のアイテム

Enterprise Application Platform バージョン 5.1.1 以降、バージョン 4.3 からアップグレードを実行することができなくなりました。特に RPM を使用した 4.3 から 5.1.1 へのアップデートはサポートされておらず、プラットフォームの障害が発生する原因となります。

### 1.4.2. 削除されるアイテム

Java EE 5 のプラットフォーム仕様では、仕様の実装で Stax API が含まれる必要がありました。その後、Stax API は Java Standard Edition 6 の一部となりました。JBoss Enterprise Application Platform 5 は最低限の SE の要件にて Java SE 6 でビルドされ配布されるようになりました。Stax API との互換性を維持するため、Stax API の jar はディストリビューションにパッケージ化されなくなりました。

# 第2章 インストール時の注意点

## 2.1. 対応の構成

最新の互換構成および認定済み構成に関する表
がhttp://www.jboss.com/products/platforms/application/supportedconfigurations/で入手できます。
テスト済みの対応構成に関する情報は、この一覧を参照してください。

## 2.2. JBOSS ENTERPRISE APPLICATION PLATFORMのインストール

JBoss Enterprise Application Platform設定のインストールや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。

## 2.3. デフォルトの起動プロファイル

デフォルトの起動プロファイルは **default**で、Java EE 5 の基本的なサーバープロファイルでデフォルトのサービスが一式含まれています。このプロファイルには、Java EE 5 アプリケーションのデプロイ時に必要で、最も頻繁に利用されるサービスが含まれています。ただし、JAXRサービス、IIOPサービスあるいは、いずれのクラスタリングサービスも含まれていません。

**default** プロファイルは実稼働向けあるいは、ロード、負荷、可用性または性能テストを実行するために設計されておりません。

## 2.4. ソースファイル

**ソースのZIPファイル**

ftp://ftp.redhat.com/pub/redhat/jbeap/5.1.2/en/source/jboss-eap-src-5.1.2.zip

## 2.5. 製品サポート

本製品で問題を発見された場合は、カスタマーサポートポータル (https://access.redhat.com) から JBoss サポート案件として起票してください。

# 第3章 新機能

Enterprise Application Platform 5.1.2 の新機能についてコンポーネント別に説明します。

## コンソール

### JBPAPP-6889

<security-role> タグが含まれるため、**deploy\management\console-mgr.sar\web-console.war\WEB-INF\jboss-web.xml** の DOCTYPE が DTD Web Application 5.0 にアップグレードされました。

## 文書

### JBPAPP-6214

JBoss セキュリティガイドには JBoss Enterprise Application Platform に同梱される SPNEGOLoginModule に関する情報の記載がありませんでした。JBoss セキュリティガイドに SPNEGOに関する記載が追加され、JBoss Negotiation ユーザーガイドが EAP ドキュメンテーションに追加されました。

## HornetQ

### JBPAPP-5760

メッセージ配布のデフォルトの遅延が **disabled** から **60000** (1 分) に変更になりました。クラスターノードのキューにあり、この周期内にクライアントによって消費されないメッセージは一致するコンシューマーを持つ他のノードへ自動的に配布されます。

## インストーラー

### JBPAPP-6680

スプラッシュスクリーンを izpack より使用できるようになりました。 **<xfragment>** タグ内で **resources.xml** より次のタグを提供する必要があります:**<res id="splash" src="@{install.config.dir}/images/splash.png" />**。**src** はユーザーによる設定が可能ですが、**id** は splash のままにしなければなりません。スプラッシュサイズは提供されるイメージのサイズに直接比例します。

### JBPAPP-6514

インストーラーが改良され、GUI インストールの最後に選択したオプションをファイルに保存し、インストーラーへの入力として使用して全く同じインストールを再作成することができるようになりました。クラスターノードや新しい EAP サーバーを大量にデプロイする場合など、同じ EAP インストールを複数のサーバーにデプロイする必要がある場合に便利なオプションです。次の手順に従ってこの機能を使用します。

1. インストールを完了します。
2. 最後のパネルで「Generate an automatic installation script」(自動インストールスクリプトを生成する)をクリックし、ファイルを保存します。
3. **java -jar <path to jar> <path to generated file>** のようにインストールスクリプトを使用します (例: **java -jar /home/tester/enterprise-installer.jar /home/tester/generatedfile**)。

### Seam2

#### JBPAPP-6498

キャッシュ制御の HTTP ヘッダーを Seam リソースサーブレットによって提供されるリソースへ追加するよう Seam を設定することができませんでした。components.xml の要求された URI に応じてヘッダを自動的に追加するよう Seam を設定できるようになりました。

### セキュリティ

技術プレビューだった PicketLink が完全サポートの対象となりました。

### システム

#### JBPAPP-6716

MainDeployerMBean によって使用されたデプロイメントメカニズムが JBoss Enterprise Application Platform 5 向けに書き換えられたため MainDeployerMBean 操作の一部が出力を返さなかったり、エラーを返したりしました。デプロイメント機能を復元するよう MainDeployerMBean メソッドが更新され、MBean サーバーや $JBOSS_HOME/bin/twiddle.sh より呼び出しが可能になりました。

# 第4章 修正済の問題

Enterprise Application Platform 5.1.2 で修正された問題をコンポーネント別に一覧にまとめています。

### Apache サーバーとコネクター

#### JBPAPP-5585

**apr** と **apr-util** パッケージは Enterprise Application Platform の一部として配布されないようになりました。基本のオペレーティングシステムや Red Hat Enterprise Linux 5、Red Hat Enterprise Linux 6 で配布される同じライブラリを使用してください。新しい Enterprise Application Platform のインストールを実行している場合、これらのパッケージがインストールされていることを確認してください。

### クラスタリング

#### JBPAPP-6175

マスターノードから各クラスターメンバーへ暗号化キーを配布する間、設定された非対称プロバイダと対称プロバイダが使用されませんでした。暗号を取得する時にこれらのオプションが使用されるよう設定されました。

#### JBPAPP-6408

場合によって HA-JNDI が完全に初期化される前に要求の処理を開始できたため、NullPointerException により要求に失敗しました。この問題は解決され、処理開始前に初期化が確認されるようになりました。

#### JBPAPP-6797

依存関係にある MBean が起動していないと HASingletonController を MBean とするデプロイメントに失敗しました。この問題を修正するため、デプロイメントの前に依存関係の MBean を最初に確認するようになったため、HASingletonController のデプロイメントが確実になりました。

#### JBPAPP-6899

UnifiedInvokerHAProxy 関数への並行呼び出しがあった場合、最後のサーバーがシャットダウンした時に NullPointerException エラーが発生することがありました。この関数を取り巻く論理がターゲットを確認しなかったため、次のエラーが発生することが問題でした。

```
FATAL [org.jboss.invocation.unified.interfaces.UnifiedInvokerHAProxy]
(TP-Processor242) Could not initialize UnifiedInvokerProxy.
```

プロキシ関数を取り巻く論理が変更され、最初にターゲットの可用性を確認するようになったため、NPE エラーの発生を防ぐようになりました。

#### JBPAPP-6989

レガシーのエビクションポリシー設定の形式を使用する際、wakeUpIntervalSeconds の指定がない場合デフォルト値が 5 秒ではなく 5000 秒に設定されました。そのため、wakeUp の周期が予想よりもはるかに長くなりました。この問題は修正され、デフォルトは 5 秒になりました。

#### JBPAPP-6538

CacheJmxWrapper を使用してキャッシュを作成する時に ClusterConfig セクションが無視され、デフォルトの JGroups 設定が使用されました。この問題はパーサーが <ClusterConfig> セクションの内容を無視することが原因でした。この問題は修正され、クラスター設定ファイル全体が正しく

読み取られるようになりました。

### JBPAPP-7015

EJB3 負荷バランシングポリシーに対してスティッキートランザクションがデフォルトで有効になっていました。そのため、別の負荷バランシングポリシーを作成、設定してこの機能を有効にする必要はありません。

### JBPAPP-6869

EJB に @Clustered が既に存在したため、クラスター化された EJB3 にデフォルトの loadBalancePolicy を指定することができませんでした。この問題は修正されたため、$JBOSS_HOME/server/all/deployers/ejb3.deployer/META-INF/ejb3-deployers-jboss-beans.xml を編集し、RoundRobin のようなプロパティを追加したり、org.jboss.ha.client.loadbalance.RoundRobin のような完全修飾クラス名を指定することができるようになりました。指定がない場合、defaultLoadBalancePolicy のデフォルトは **`org.jboss.ha.client.loadbalance.RoundRobin`** になります。defaultStickyLoadBalancePolicy のデフォルトは指定がない場合 **`org.jboss.ha.client.loadbalance.FirstAvailable`** になります。

### JBPAPP-6972

クラスター化されたプロキシファクトリ実装が上書きされたスタック名を考慮せず、代わりにデフォルトのクラスター化されたクライアントインターセプタースタックを使用する場合、EJB3 のクラスター化されたプロキシのバグを修正します。これにより、@RemoteBinding(interceptorStack=...) アノテーションと jboss.xml ファイルの <interceptor-stack> タグはクラスター化された EJB に対して無視されます。

### JBPAPP-6448

EAP の管理コンソールまたは JBoss Operations Network より Web アプリケーション (WAR) をクラスターファームにデプロイすると、次のような NullPointerException が結合するノードで発生しました。

```
16:16:53,586 ERROR [ScopedProfileServiceController] Error installing to
Create: name=ProfileKey@27905a42[domain=default, server=default,
name=farm] state=Configured mode=On Demand requiredState=Installed
java.lang.reflect.InvocationTargetException
...
Caused by: java.lang.NullPointerException
at
java.util.concurrent.ConcurrentHashMap.get(ConcurrentHashMap.java:796)
at
org.jboss.ha.timestamp.TimestampDiscrepancyService.getTimestampDiscrepan
cy(TimestampDiscrepancyService.java:328)
at
org.jboss.profileservice.cluster.repository.DefaultSynchronizationPolicy
.getTimestampDiscrepancy(DefaultSynchronizationPolicy.java:158)
at org.jboss.profileservice.cluster.repository.DefaultSynchronizationPo
licy.acceptUpdate(DefaultSynchronizationPolicy.java:121)
```

デプロイヤクラスターノードとデプロイされるクラスターノードの間で時間の同期をチェックする際の不完全な内部データがこのエラーの原因でした。この問題は修正され、クラスターノードへの WAR をデプロイできるようになりました。

### JBPAPP-6932

分裂後に 2 つのクラスターノードを結合する時、1 つのクラスターノードのみ HASingletons を継続

する必要がありますが、両方のクラスターノードが HASingletons を継続したため、クラスター操作を信用できませんでした。これはノードの結合時に適用された論理にエラーがあったことが原因でした。この問題は解決され、クラスターの結合が予想通り行われるようになりました。

### JBPAPP-7011

パッシベーションを設定されたキャッシュローダーを持つ JBoss Cache の cache.getNode を呼び出すとデータの損失が発生することがありました。パッシベートされたノードを cache.getNode がロードするとデータがロードされずにエビクションイベントを引き起こしました。エビクションがノードを再度パッシベートするとデータは上書きされました。この問題は解決され、このような状況でもデータの損失が発生しなくなりました。

## コンソール

### JBPAPP-6683

JBoss Web コンソールのページに直接アクセスすると、例外エラーが発生しました。スタートページからのみ Web コンソールにアクセスできるようにしたため、この問題は修正されました。

### JBPAPP-6682

JBoss Web コンソールより新たにしきい値モニターを作成すると例外が発生しました。DeploymentFileRepository MBean があるかを確認し、ない場合はエラーメッセージを表示するよう修正されました。

## 文書

### JBPAPP-7236

『管理設定ガイド』 の『標準のサーバープロファイル』の項で、明示的に記載または設定されたプロファイルがない場合「All」プロファイルが使用されるという内容の誤った記載がありました。このような場合、実際は「Default」という名前のプロファイルが使用されます。『インストールガイド』には「『Red Hat Enterprise Linux 上でアプリケーションサーバーをサービスとして実行する』」という新しい項も追加されました。

### JBPAPP-7361

『管理設定ガイド』の項「Oracle に対する回避法」が最新情報にて更新されました。Oracle をご使用の場合はこの項の内容がご使用のインストールに関連しているか確認してください。

### JBPAPP-4828

EAP を 5.1.1 から 5.1.2 およびそれ以降のリリースにアップグレードする手順が『インストールガイド』に追加されました。

### JBPAPP-4958

『Hibernate Core ユーザーガイド』では誤った POM ファイルに参照がされていました。本リリースの文書ではこの参照は修正されています。

### JBPAPP-5566

org.jboss.mail.SessionObjectFactory クラスでのバグにより、JNDI ツリーに複数のサービスが設定され存在する場合でもメールサービスが 1 つしか解決されませんでした。 1 つのサービスが設定されている場合、このクラスによって複数のメールサービスが解決されるようになりました。

### JBPAPP-6138

HornetQ ユーザーガイドの「メッセージ駆動型 Bean」の項に新しい項が追加されました。新しい項「メッセージ駆動型 Bean の高可用性」ではクラスター化された環境で使用されるメッセージ駆動型 Bean に「クラスター」アクティベーションプロパティが必要であることを説明しています。

### JBPAPP-6213

データベースへの接続が切断された場合、ノードはすべてのメッセージの読み取りに失敗します。これは、ノードはデータベースへの接続が切断するとメッセージを応答確認できないからです。この問題を回避するにはノードパラメータ ***MaxRetry*** を **-1** よりも大きい値に設定します。属性の値は **[database]-persistence-service.xml** の **PersistenceManager**、 **PostOffice**、**JMSUserManager** に設定することができます。MaxRetry パラメータの詳細とその他の関連パラメータとともに設定する方法が Messaging ユーザーガイドに追加されました。

### JBPAPP-6741

『管理設定ガイド』 に「EJB3 サービスを持つエンタープライズアプリケーション」と「レガシー EJB サポート」の 2 つの章が追加されました。

### JBPAPP-6887

『セキュリティガイド』の項、『セキュリティマネージャーの使用』に記載されていた JBoss 署名キーのインポート手順に誤りがありました。keytool ユーティリティへのパスが **$JBOSS_HOME/bin/keytool** となっていましたが、正しくは **JAVA_HOME/bin/keytool** でした。そのため、セキュリティガイドの本項は修正されました。

### JBPAPP-6958

HornetQ ユーザーガイドに GFS2/SAN のフェンシングについての記載が全くありませんでした。HornetQ にはフェンシングが必要ですが、フェンシングを有効にする方法はたくさんあるため、ユーザーガイドに限定的な説明を含めるのは困難になります。この問題に対応するため、「高可用性とフェイルオーバー」の章に新たに「フェンシング」の項が追加されました。この項では、他の Red Hat ドキュメントでフェンシングの設定に関する情報を探す方法が記載されています。

### JBPAPP-7000

HornetQ ユーザーガイド にて、Java 5 仮想マシン内で実行しているクライアントに関する情報に誤りがあり、クライアントクラスパスの章にはない jar を参照していました。この章に適切な情報が含まれるようになりました。

### JBPAPP-7200

管理設定ガイドのクラスタリングの項において、JBoss Messaging に必要となる ServerPeerID の範囲 (0 から 1023）が指定されていませんでした。そのため、この範囲外の値を指定した場合に java.lang.IllegalArgumentException が発生する問題の原因となりました。そのため、「初期準備」の項に必要なノードの範囲を記した勧告が追加され、ユーザーがノードの ServerPeer ID を設定する時に可能な範囲が分かるようになりました。

### JBPAPP-7324

Windows のユーザーがグラフィカルインストーラーを使用してプラットフォームをインストールすると問題が発生することが判明しました。インストーラーを実行する前に JAVA_HOME 環境変数を正しく設定しないと、警告メッセージが表示されます。インストールガイドが改訂され、JAVA_HOME の設定に関する必要条件とその手順が記載されている付録へのリンクが追加されました。

## EJB

### JBPAPP-4656

タイムアウトコールバック中にトランザクションロールバックが発生した時、EJB 3.1 FR 18.4.3 の仕様通りタイムアウトコールバックが読み出されなければなりません。コンテナがタイマーをトランザクションへ enlist せず、ロールバック時にタイマーが読み出されませんでした。トランザクション内での実行をサポートするため、タイマーコールバックの実装が変更になり、タイムアウトコールバックがロールバックされた後、サービスによってコールバックが読み出されるようになりました。

### JBPAPP-5516

メッセージの受信に対して完全に準備されていない MDB にメッセージが送信されると、MDB コンテナが DispatcherConnectException をスローしました。これはチェック例外であったため、トランザクションセマンティックの判断には無視され、トランザクションがコミットし、MDB がメッセージを受信しないままメッセージが確認応答されました。この問題を修正するため、インフローメソッドの署名に対してすべての例外が適切にチェックされたか確認する新しい論理が導入されました。その結果、無効な例外に対して UndeclaredThrowableException (/RuntimeException) が発生します。メッセージインフロー中に RuntimeException が発生するとトランザクションはロールバックされます。そのトランザクション内のメッセージは後の区間で確認応答および読み出しされません。メッセージの正常受信時までに MDB は準備できている状態で、通常の動作が再開されることが前提となります。

### JBPAPP-6953

EJB jar デプロイメントが依存関係を指定し、その関係が再起動されると、次のような IllegalStateException メッセージによってデプロイメントまたは EJB の開始に失敗します。

```
Caused by: java.lang.IllegalStateException:
jboss.j2ee:ear=helloWorld.ear,jar=helloWorld-
ejb.jar,name=HelloBean,service=EJB3 is already installed.
```

これは EJB のデプロイメントがインストールまたは起動中に重複されたことが原因です。EAP 5.1.2 へアップグレードする前にこの問題を解決するには、依存する EJB を再デプロイするかサーバーを再起動します。この問題は修正されたため、アップグレード実行後に回避法を実行する必要はありません。

### JBPAPP-6789

クラスター化された EJB が @RemoteBinding(jndiBinding=...) または <remote-binding> を誤って使用すると、JNDI にプロキシが作成され、そのプロキシによって正しいプロキシが置き換えられました。クラスターメンバーシップが変更されると、正しくないプロキシのクラスターメンバーリストが更新されたため、EJB クライアントは新しいメンバーについて通知されませんでした。追加のプロキシは作成および削除できなくなりました。クラスターメンバーリストが適切に更新されるようになり、EJB コールが新しいノードへロードバランスし、フェイルオーバーできるようになりました。

## 組み込み Jopr

### JBPAPP-6765

admin-console (プロファイルサービス API) を使用してアプリケーションがデプロイされた場合、起動に失敗した時にアプリケーションのデプロイメントが削除されました。デプロイメントや起動の失敗が発生した時にデプロイメントが削除されたため、admin-console を使用してアプリケーションを再デプロイすることができませんでした。修正が加えられたため、デプロイメントが起動に失敗しても管理コンソールがアプリケーションやリソースのデプロイメントを許可するようにな

りました。この機能は、依存関係の問題や HA シングルトンの設定によりアプリケーションが起動できないと予想する場合に使用できますが、デプロイメントは EAP サーバー上にある必要があります。

### HornetQ

#### JBPAPP-6303

HornetQ のキューがリモートの HornetQ ブローカー上でホストされ、HornetQ ブローカーのローカルインスタンス上で定義されていない場合、HornetQ アクティベーションは JNDI キューパスの複数のフォワードスラッシュを無視しました。キューが <queue name="jms/queue/myQueue"> として指定されると、HornetQ は <queue name="myQueue"> として解釈しました。キュー名からフォワードスラッシュが削除されたため、リモートキューでリッスンする MDB はキューに接続できませんでした。この問題は本リリースで修正されました。

#### JBPAPP-7205

サーバーが SA Nへ接続できなくなる場合など、クラスター化された HornetQ インスタンスにてクラスターノードのジャーナルへの接続が切断されると、他のノードが失敗したノードを引き継ぎませんでした。この問題は修正され、クライアントを中断せずに失敗した HornetQ ノードからのフェイルオーバーが発生するようになりました。

#### JBPAPP-7556

メッセージ再発信のデフォルトの遅延が変更になり、60000 ミリ秒 (1 分) になりました。以前のデフォルト値は 0 であったため、再発信の遅延がありませんでした。 **`hornetq-configuration.xml`** に変更が加えられました。

#### JBPAPP-5895

クラスター化された HornetQ 設定において、アクティブノードに障害が発生するとクライアントが JNDI を使用してバックアップインスタンスに接続できない問題が発見されました。この問題の根本的な原因が特定され、その原因が修正されたため、クライアントがアクティブノードからバックアップノードへフェイルオーバーできるようになりました。

#### JBPAPP-7455

クラスター設定にて、設定に指定されている **`min-large-message-size`** がブリッジサービスによって無視され、デフォルト値の 100 KiB が使用される問題が判明しました。この問題は解決され、設定に **`min-large-message-size`** の値が指定されている場合、その値が認識されるようになりました。

#### JBPAPP-5080

サーバーがシャットダウンすると、HornetQXAResourceWrapper は接続障害に警告を記録します。これは、HornetQXAResourceWrapper のライフサイクルはトランザクションマネージャによって制御されるためです。リカバリ API が HornetQ に追加され、この問題は修正されました。

#### JBPAPP-6368

HornetQ には Microsoft Windows サーバー上で HornetQ を実行するユーザー向けの **`switch.bat`** スクリプトが含まれていませんでした。**`build.xml`** ファイルが含まれる HornetQ ディレクトリで ant を実行して HornetQ をインストールすることもできますが、Linux 環境と一貫していません。Linux 環境における switch.sh の動作を再現する Microsoft Windows ユーザー向けの switch.bat スクリプトが同梱されるようになりました。

#### JBPAPP-6522

270MB ほどもあるサイズが大きな *bytemessage* の配信中に目的のサーバーが中断されると HornetQ のコアブリッジサービスがハングすることが判明しました。目的のサーバーの **data/large-messages** ディレクトリより失敗したメッセージの内容が消去されませんでした。

大容量の *bytemessage* の通信中にサーバーが複数回中断されると、**data/large-messages** ディレクトリで複数のファイルが永続化されます。ブリッジサービスがデプロイされたサーバーを再起動することがこの問題を修正する唯一の方法となります。

この修正方法により、コアブリッジに変更が実装され、目的のサーバーに無事再接続するまで再起動するようになります。目的のサーバーが中断してもブリッジはハングしなくなるため、この問題が修正されます。

### JBPAPP-7115

クラスター化された HornetQ 設定では容量が 1MB 以上のメッセージはキューから取り出せなくなり、NullPointerException が発生する原因となります。サイズが大きなメッセージもクラスター化の設定で対応できるようになったため、この問題は解決しました。

### JBPAPP-7161

サイズが大きなメッセージの送信中にサーバーがクラッシュすると、手作業で削除されるまでメッセージが large message ディレクトリに残ってしまいました。ファイルのジャーナル作成に一時的な記録が追加され、ファイルが破損したら削除されるようになりました。

## IIOP

### JBPAPP-6469

ファイル **ejb3-iiop-deployers-jboss-beans.xml** が **$JBOSS_HOME/server/standard/deployers/ejb3.deployer/META-INF** になかったため、標準のプロファイルを使用すると IIOP のデプロイメントがデプロイされませんでした。このファイルが存在するようになったため、IIOP デプロイメントが標準のプロファイルで動作可能になりました。

### JBPAPP-6956

新しい CORBA のネーミングコンテキスト実装が追加され、ローカルキャッシュを使用して名前を解決し不必要なリモート呼び出しを防止するようになりました。

### JBPAPP-6462

分離された EAR 内でカスタムの CORBA サーバントを使用した場合、JacORB が正しくないスレッドコンテキストクラスローダー (TCCL) を使用するとサーバントが EAR 内の EJB を呼び出そうとしたため、ClassNotFoundExceptions が発生する原因となりました。サーバントクラスの TCCL が確実に使用されるようにしたため、この問題の原因は修正されました。

## インストーラー

### JBPAPP-6954

インストーラーには **Previous** ボタンがありましたが、このボタンを押すと既にインストールされているファイルがそのままインストールされた状態になりました。この問題が発生しないようにするため、**Previous** ボタンは無効になりました。

### JBPAPP-7540

Red Hat Enterprise Linux にて、チェックボックス「すべてのユーザーにショートカットを作成す

る」を選択してもインストーラーによって作成されたメニュー項目をすべてのユーザーが使用できませんでした。これはインストーラーの逆論理が原因で、「すべてのユーザーにショートカットを作成する」を選択**しない**場合のみ作成されたメニュー項目をすべてのユーザーが使用できました。この問題は修正されました。

### JBPAPP-7565

Microsoft Windows 上のインストーラーで作成されたメニュー項目が Red Hat Enterprise Linux 上で作成されたメニュー項目と一貫していませんでした。メニュー項目の一部に使用した run コマンドが Windows 上では無効であったことが原因でした。インストーラーが修正されたため、両方のオペレーティングシステムですべてのメニュー項目が有効になりました。

### JBPAPP-7568

Microsoft Windows のインストーラーによって作成されたショートカットに誤ったパスが指定されていたため、ショートカットがプラットフォームを起動できませんでした。パスには jboss-as が含まれていましたが、jboss-as-web であるべきでした。インストーラーが修正されたため、ショートカットに正しいパスが含まれるようになりました。

### JBPAPP-7566

Windows 上でアジアの文字が含まれるショートカットがアンインストーラーによって削除されませんでした。この問題は誤った文字列のエンコーディングが原因であったことが分かりました。この問題は修正され、インストーラによって作成されたすべてのショートカットはアンインストーラーによって完全に削除されるようになりました。

### JBPAPP-6376

Enterprise Application Platform で提供されるDVDストアのデモ Seam アプリケーションがデプロイできませんでした。このアプリケーションは更新され、完全にデプロイできるようになりました。

### JBPAPP-6603

インストーラーを OpenJDK と実行すると、ウインドウの下にあるボタンが切り取られ、アクセスに問題がありました。位置を若干変更してこの問題は解決しました。

### JBPAPP-6405

Windows で日本語か中国語を地域設定した場合、Enterprise Application Platform のインストーラーがスタートメニューやデスクトップのショートカットを作成できなかったため、ユーザーは Windows Explorer またはコマンドプロンプトより手動でアプリケーションを開始しなければなりませんでした。この問題は修正され、このような状況でもインストーラーが正しくショートカットを作成するようになりました。

## JCA

### JBPAPP-6939

JCA DTD (`jboss-ds_5_1.dtd`) は background-validation-minutes という名前の要素を参照しましたが、これは誤りで background-validation-millis を参照するべきでした。この問題は修正され、DTD に正しい要素名が含まれるようになりました。

### JBPAPP-7102

RMI よりリモートで公開されるデータソースはスレッドセーフではない方法で接続とステートメントのマップを使用しました。そのため、クライアントが「unable to find statement」(ステートメントを探せません) というエラーメッセージを受信するか、マップのデータ構造の破損が原因でサー

バーが無限ループに陥りました。リモートデータソースのマップへアクセスする方法を変更してこの問題は解決され、スレッドセーフになりました。

### JBPAPP-6855

アクティベーション仕様に推奨プロパティがない場合、次のような DeploymentException エラーが発生しました。

```
08:20:50,262 ERROR [AbstractKernelController] Error installing to Start:
name=jboss.j2ee:binding=message-driven-
bean,jndiName=local/gss_mdb_20@32140521,plugin=invoker,service=EJB
state=Create mode=Manual requiredState=Installed
org.jboss.deployment.DeploymentException: Unable to create activation
spec ra=jboss.jca:service=RARDeployment,name='wmq.jmsra.rar'
...
```

プロパティの検証が変更され、未知の設定プロパティやサポートされていない設定プロパティが検出された場合に次の警告メッセージが出力されるようになりました。

```
17:52:45,801 WARN  [ActivationSpecFactory] Unable to set 'foo' property
on org.jboss.resource.adapter.jms.inflow.JmsActivationSpec
```

### JBPAPP-6684

複数のスレッドが JCA レイヤーへアクセスし、トランザクションをコミットまたはロールバックするとデッドロックが発生しました。これらの問題を解決するため JCA コードが更新されたため、複数スレッドによるアクセスが正しく動作するようになりました。

### JBPAPP-6943

**allocation-retry** や **allocation-retry-wait-millis**、 **no-tx-separate-pools** の設定が設定ファイル (*-ds.xml) に指定されていても動作せず、JMX コンソールより変更する場合は動作しました。この問題の原因は修正されたため、どちらの方法でも設定可能になりました。

## JMX

### JBPAPP-6900

**bin/server.policy.cert** に定義された Java セキュリティポリシーが厳密すぎたため、Java セキュリティマネージャーが有効になっている場合、JMX コンソールがロードしない原因となっていました。この問題を解決するため、**bin/server.policy.cert** ファイルにエントリが追加され、セキュリティマネージャーの制約が緩和されました。

## メッセージング

### JBPAPP-5280

MessageSucker はクラスター内の別メンバーとの間でメッセージを移行します。**onMessage** ルーチンはメッセージをローカルキューに配信しようとします。配信に失敗すると、メッセージが紛失したような状態になりました。メッセージはデータベースにあったのですが再配信は困難でした。これは JBoss Messaging コンポーネントの不具合が原因でしたが、この不具合は修正されました。この修正は複雑で、https://issues.jboss.org/browse/JBMESSAGING-1822 にある JBMESSAGING-1822 に説明が記載されています。簡単に説明すると、信頼性の新しいレイヤーがメッセージ配信論理に追加されたことになります。

### JBPAPP-6620

HSQLDB データベース以外を使用するよう設定すると、デフォルトのプロファイルでメッセージングサーバーが起動しませんでした。これは、***データベースタイプ*-persistence-service.xml** ファイルにあるクラスター化されたプロファイルのみに適用される行がコメントアウトされていなかったことが原因でした。

この問題を修正するため、***データベースタイプ*-persistence-service.xml** ファイルに規範的なコメントが追加され、非クラスター化のメッセージングサーバーインスタンスに対して不必要な設定指示文をコメントアウトするよう指示されています。

## ネーミング

### JBPAPP-6431

すべての新しい InitialContext がプロバイダーリストを順番にループする org.jnp.interfaces.NamingContext.checkRef でパフォーマンス上の問題が判明しました。次のノードをチェックする前にタイムアウトが発生するまで待機するため、プロバイダーリストの最初にある利用できないノードによって遅延が発生することがありました。修正によって利用可能な JNDI プロパティ (jnp.unorderedProviderList) とシステムプロパティ (jboss.global.jnp.disableDiscovery) が追加されました。これらのプロパティは複数のプロバイダが使用された時に最初の JNDI サーバールックアップの順番を変更します。プロパティがなくても (デフォルト )新しい InitialContext ごとに各プロバイダーが順番に試行されます。新しいプロパティのセットを用いると、プロバイダーリストを再度順番にチェックする前にプロバイダーへの既存の接続がすべて使用されます。

## その他

### JBPAPP-5148

以前、適切なパスであるかデプロイメントファイル名がチェックされませんでした。不適切なファイル名が原因で予期しないファイルの削除や変更が発生することがありました。デプロイメントファイルのパスが適切であるかチェックされるようになったため、不正パスが使用された場合は例外がスローされるようになりました。

### JBPAPP-6046

endorsed クラスパスなどトップレベルのクラスローダーで SLF4J (Simple Logging Facade for Java) が使用されると NullPointerExceptions が発生する原因となりました。この問題の原因は修正され、SLF4J はトップレベルのクラスローダーで使用できるようになりました。

### JBPAPP-6486

要素の前に **xsi:nil=true** があると、JAXB 要素が適切に処理されませんでした。この問題は解決されたため、JAXB 要素が正しく処理されるようになりました。

### JBPAPP-6524

commons-logging.properties ファイルが複数あると use_tccl=false|true の設定が信用できないことがありました。この問題を修正するため、新しいシステムプロパティ **org.apache.commons.logging.use_tccl** が作成されましたが、デフォルトはこれまで通り **true** になっています。

### JBPAPP-6824

リソースをロードする時 (getResource)、parent-first=true のデプロイメント設定が親に委譲しませんでした。この機能のセットを有効にするには、以下を設定します。

```
-Dorg.jboss.classloader.honor.resource.delegation=true
```

この設定により親が先となる順番でリソースがロードされます。run.sh コマンドか JAVA_OPTS 環境変数の $JBOSS_HOME/bin/run.conf で設定可能です。

### JBPAPP-6849

「HornetQ への移行」の項に、高可用性を考慮しなければならない状況で JBoss Messaging アプリケーションを移行する特定の情報が含まれていませんでした。HornetQ フェイルオーバー例外を適切に処理するために必要な追加コードについての情報が記載されている新しい項「クライアント側の障害処理」がガイドに追加されました。

### JBPAPP-6884

XML スキーマ宣言の jboss-xa-ds.xml が削除されました。EAP 5.1.x のすべてのデータソースデプロイメントは代わりに jboss-ds_5_1.dtd を使用しなければなりません。

### JBPAPP-7128

getModulePath メソッドがトップレベルの ear でないデプロイメントに空の文字列を返したため、SimpleJavaEEModuleInformer の getModulePath メソッドが JavaEEModuleInformer の仕様に従いませんでした。このようなデプロイメントに対してこのメソッドがモジュール名を返すようになったため、JavaEEModuleInformer 仕様に適合するようになりました。

### JBPAPP-7241

まれな状況でメッセージが消費される前に PagedReference がガベッジコレクションされたため、pagecredits の順番が乱れ、影響があったメッセージが止まることがありました。この問題を引き越す要因の 1 つが最低ページサイズです。このような状況を検知し、適切に対応することでこの問題は解決しました。

### JBPAPP-7093

メモリーリークを解決するため、以前のリリースで XMLReaderManager のキャッシングの挙動が削除されました。これにより、各呼び出しによる XMLReader の再作成が強制され、同期化メソッドが存在するため他のスレッドとの競合が発生しました。SAXParser クラスの使用によりキャッシングが回復したため、スレッド競合の問題は解決されました。最初の問題だったメモリーリークは SAXParser 再設定メソッドを使用して回避されます。

## スクリプトおよびコマンド

### JBPAPP-5901

RHEL 6 にて JBoss サービススクリプトのバックスラッシュの一部が誤ってエスケープされていたため、無効なコマンドが実行されたり、JBoss の起動や停止ができない原因となっていました。このスクリプトは修正され、JBoss はRHEL 6 上で確実に起動や停止ができるようになりました。

### JBPAPP-6944

Red Hat Enterprise Linux 上で「サービス停止」の要求があると、JBoss Enterprise Application Server のサービスがタイムアウト失敗エラーによりシャットダウンできませんでした。サービス停止の要求があった時にサービスが確実にシャットダウンするよう、init スクリプトが変更されました。

### JBPAPP-6951

ログ **$JBOSS_HOME/server/$JBOSSCONF/log/server.log** にサーバーの起動停止シーケンスのインスタンスが複数あると、シャットダウンスクリプトが誤って検出されサーバーが適切に

シャットダウンする前にシャットダウンしました。ログファイルの代わりにサーバーのプロセス ID (PID) を監視するようスクリプトを変更し、この問題は修正されました。

### JBPAPP-7013

すべての *.xml ファイルと *.property ファイルが Windows 互換の形式 (行末) に変更されました。UNIX 互換の形式に変更するには、スクリプト **`jboss-eap-5.1/jboss-as/bin/convert-dos2unix.sh`** を **`jboss-eap-5.1/jboss-as/bin/`** ディレクトリで実行します。

## Seam

### JBPAPP-5039

Microsoft Internet Explorer 8 と Seam のタスク例との互換がなかったため、タスクに追加されたテキストがタスク一覧に空白の行として表示されました。Seam のタスク例が改善され、 Microsoft Internet Explorer 8 上で編集されたタスクが適切に表示されるようになりました。

### JBPAPP-6611

EAP 5.1.1 からの seam-gen を使用して生成された EAR プロジェクトの .classpath ファイルに誤った依存関係 (**`core.jar`** および **`janino.jar`**) が含まれていました。これらの依存関係は不必要になったため Eclipse のクラスパスファイルから削除されました。 また、既存の EAR アプリケーションにて EAP5.1.2 より Seam と Drools の jar を更新するとこれらの依存関係が既存の EAR プロジェクトより削除されます。

### JBPAPP-6650

Seam Mail タグ **`<ui:repeat...>`** を使用してメールメッセージが作成されると、添付リストがキャッシュされたため受信側が添付を複数回受信しました。メッセージのボディーへの追加後に添付リストを再設定するようにし、この問題は修正されました。

### JBPAPP-6751

ServerConversationContext.flush() メソッドが PersistentContext インスタンスと BusinessContext インスタンスを作成することがあり、これらのインスタンスは HTTP セッションが無効になるまでメモリに存在しました。必要な時のみこれらのインスタンスが作成されるようになったため、この問題は解決しました。

### JBPAPP-6946

ヘッダー行の CRLF+CRLF がバッファー境界で区切られると (ヘッダーのバッファーが CRLF で終わり、次のバイトが CRLF となる場合) マルチパートリクエストが正しく解析されませんでした。そのため、MultipartRequest エラーが発生し、CPU メモリーが過剰に使用される原因となりました。このようなマルチパートリクエストが適切に解析されるようになったため、エラーが発生しなくなりました。

### JBPAPP-7084

org.apache.el.util.concurrentcache クラスが WeakHashMap へ同時アクセスする要求を同期化しませんでした。そのため、場合によっては無限ループが発生し、CPU メモリーを過剰に消費しました。根本的な問題は解決され、org.jboss.el.util.concurrentcache クラスでの WeakHashMap へのアクセスは同期化されるようになりました。

### JBPAPP-7139

OpenID4Java の属性交換 (AX) 拡張は属性が署名されていることを確認していないことが判明しました。AX を使用して、アプリケーションはアイデンティティプロバイダのみを信用しアサートするという情報を受信すると、リモート攻撃者はこの欠陥を利用して介入者攻撃を実行し、特別に作

成された要求を用いて情報の整合性を危険にさらすことが可能でした。デフォルトでは JBoss Seam openid 例のアプリケーションのみが OpenID4Java を使用します(CVE-2011-4314)。

- openid4jav-nodeps.jar
- httpclient.jar
- httpcore.jar
- nekohtml.jar
- jcip-annotations.jar
- guice.jar
- commons-codec.jar

### セキュリティ

#### JBPAPP-6364

httpha-invoker よりデフォルトでデプロイされたインボーカーサーブレットは HTTP GET と POST メソッド上でのみアクセス制御を実行したため、リモートの攻撃者が他の HTTP メソッドを使用して非認証の要求を作成することが可能であることが判明しました。セキュリティインターセプターによって提供される第 2 レイヤーの認証により、管理者がセキュリティインターセプターの設定を誤ったり、無効にしない限りデフォルトインストールではこの問題は悪用することができません (CVE-2011-4085)。

#### JBPAPP-6893

以前の EAP リリースでは cglib によって生成されたコードが署名された cglib JAR パッケージとオーバーラップするため、cglib.jar を使用するとセキュリティ例外が発生しました。この問題を解決するため、cgilib.jar が非署名となり、この変更に対応するためセキュリティポリシーが変更されました。

#### JBPAPP-6996

高負荷の状況下では SimpleRoleGroup クラスで ConcurrentModificationException が発生したため、ユーザーが認証されませんでした。原因は特定され、修正されたため、この問題は発生しなくなりました。

### SNMP

#### JBPAPP-6504

SNMP アダプターに long 値が渡されると、**Counter64** タイプに変換できず、**Gauge32** のみに変換可能でした。そのため、戻り値が 64 ビットではなく、誤った 32 ビットタイプでした。JVM 引数 ***-Djboss.snmp.use64bit*** が使用できるようになりました。この引数を使用すると、SNMP アダプターによって 64 ビットデータタイプが返されるよう指定することができます。***-Djboss.snmp.use64bit=true*** のように使用します。デフォルト値は **false** です。設定すると、すべての Long タイプは 64 ビットの値として返されます。新しいバージョンのコードが使用された時に既存の設定が変更しないよう、後方互換性を維持するためデフォルトが **false**になっています。

### システム

### JBPAPP-6545

既存のセキュリティコンテキストがない場合、セキュリティインターセプターがセキュリティコンテキストを作成しませんでした。新しいクリーンなスレッド (セキュリティコンテキストのないスレッド) は EJB のローカルインターフェースを呼び出しすることができませんでしたが、EJB のリモートインターフェースは問題なく呼び出すことができました。PreSecurityInterceptor が変更され、セキュリティコンテキストが存在しない時は新しいセキュリティコンテキストを作成するようになりました。また、セキュリティコンテキストを持たない新しいスレッドも EJB のローカルインターフェースを呼び出せるようになりました。

## Web

### JBPAPP-7267

Jasper が JAXP の ParserUtils を使用していると、フレームワークのクラスとデプロイメントのクラスが意図的に分離されたため、これらのクラス間で呼び出しが阻止されました。この問題を回避するため呼び出しが変更され、分離された環境でも呼び出しができるようになりました。

### JBPAPP-6141

Tomcat の問題により、サマータイム適用による時間の調節が正しくありませんでした。この問題は修正され、正しく計算されるようになりました。

### JBPAPP-7089

@PostConstruct アノテーションや他のアノテーションがあった場合、同じ名前の別のメソッドによってオーバーロードされたメソッドでそのメソッドへの呼び出しが失敗し、例外エラーが発生しました。これは、そのメソッドに引数が渡されず、そのメソッドの有無のみが確認されたためです。この問題を解決するため追加のチェックが導入され、メソッドが呼び出される前にパラメータの存在を確認するようになりました。

### JBPAPP-6724

1 つのデプロイメントに複数の認証メソッドが使用されると、WebAuthentication メソッドによってシングルサインオン (SSO) セッションの開始が阻止されました。この問題は解決され、WebAuthentication が SSO セッションに対して正しく動作するようになりました。

### JBPAPP-6501

getter メソッドが空の文字列を返し、数値変換が試行されるとエラーが発生しました。空の文字列をテストし、処理を改良したためこの問題は修正されました。

### JBPAPP-7104

@EJB または @Resource フィールドを JSF 管理の Bean のスーパークラスに挿入できませんでした。@EJB がアノテーションより挿入される場合、これらのアノテーションが処理されなかったため、フィールドが null のままになったり未設定になりました。この問題の原因は解決され、予想通り挿入が動作するようになりました。

## Web サービス

### JBPAPP-7124

jettison.jar、 stax-ex.jar、streambuffer.jar のファイルが存在する場合、CXF インストーラーによって client/ ディレクトリより削除されましたが、WS 以外のクライアントはこれらのファイルを使用することができたため、CFX のアップグレードやインストールの後にアプリケーションの一部が動

作しないことがありました。CFX インストーラーはこれらのファイルを削除しなくなったため、CXF のインストールやアップグレードの後でもアプリケーションが予想通り動作するようになりました。

### JBPAPP-6822

@HandlerChain アノテーションが付けられ、WS セキュリティ (WSSE) が有効になっている終点の実装が原因で重複の SOAP ボディーが追加されました。この問題の原因は解決されたため、重複が発生しなくなりました。

### JBPAPP-6593

JBoss EAP 5.1.1 で httpbinding の例を実行するとエラーが発生しました。これは、JBossWS Native が JBoss Remoting にストリームを閉じるよう強制したためです。この強制により不具合が再発しましたが、JBoss EAP 5.1.2 では修正されました。

### JBPAPP-6729

使用された文字セットが暗黙のデフォルトである ISO-8859-1 でない場合、JBoss Web サービスが charset パラメータをルートの MIME パートに追加できなかったため、JBoss から送信され MTOM/XOP でエンコードされた web サービスの応答を一部のクライアントが解析できませんでした。必要な場合、charset パラメータが web サービスの応答に追加されるようになりました。

### JBPAPP-6734

wsdl ファイルにてポリシー参照が <wsp:policy> でラッピングされていないとシステムが RuntimeException をスローしました。これは、ポリシー参照を解決できなったため発生しました。ポリシー参照は <wsp:Policy> 要素でラッピングするか、スタンドアロン要素 (<wsp:Policy> 要素でラッピングされない) として定義できるようになりました。

### JBPAPP-6371

CXF スタックからの JBossWS シェルスクリプトとバッチファイル (wsconsume、 wsprovide、 wsrunclient) は `-Djava.net.preferIPv4Stack` 設定オプションの使用と一貫していませんでした。Native スタックと CXF スタックの両方が IPv4 優先で設定されるよう変更になりました。

### JBPAPP-6389

スキーマ検証機能はインポートされたスキーマを 1 つしかサポートしませんでした。インポートされたスキーマを複数定義できる WSDL で @SchemaValidation を定義できるようになりました。

### JBPAPP-5866

SOAPFactoryImpl は名前空間を宣言している要素の直接の子要素でない子要素すべてに対して追加の名前空間宣言を強制しました。この明示的宣言は冗長であるため、追加されなくなりました。

### JBPAPP-6285

EnpointReference でディスパッチクライアントを作成しようとすると JBoss Web サービスが org.w3c.dom.DOMException をスローしました。これは、ある並列環境にてサーバーが別のドキュメントからディスパッチクライアントへノードを追加したため発生しました。JBoss Web サービスは適切なノードをディスパッチクライアントへ追加し、EndpointReference で正しくディスパッチクライアントを作成するようになりました。

### JBPAPP-6186

JBoss Web サービスで使用された UsernameToken プロファイルのパスワードダイジェストは base64 でエンコードされたノンスを使用し、Oasis の仕様に従いませんでした。その結果、処理は UsernameToken プロファイルをサポートする他の Web サービスと互換性がありませんでした。ノ

ンスのバイナリ値を使用するようパスワードダイジェストが修正され、JBoss Web サービスが UsernameToken プロファイルの他の実装と動作するようになりました。

### JBPAPP-6245

JBossWS ネイティブは組み込み DTD の再帰的なエンティティ解決を適切に防ぎませんでした。リモート攻撃者はデプロイされた Web サービスへ慎重に作成した POST 要求を使用し、CPU のリソースを消耗し DoS 攻撃を実行することが可能でした (CVE-2011-1483)。

### JBPAPP-6249

JBossWS-CXF JAXWS クライアントは Bus インスタンスを必要とする CXF ProviderImpl に依存しています。Bus が現スレッドに関連付けされていないと、Bus の読み出し処理がサーバー向けに作成された Bus インスタンスを返しました。カスタマイズされたプロバイダ実装がクライアント向けに新しい Bus インスタンスを作成するようになり、Bus 要求はこの Bus インスタンスを返すようになりました。

### JBPAPP-6529

CXF の終点が HTTP GET 要求を受け取ると、org.apache.cxf.staxutils.StaxUtils.readDocElements で NullPointerException をスローしました。終点への GET 要求により例外が発生するはずですが、NullPointerException は適切ではありません。終点が GET 要求をフィルターするようになり、例外の原因となるコードを安全にスキップするようになりました。

### JBPAPP-6479

wsdl パートに partIndex 0 を持つヘッダーが含まれると、ディスパッチ/プロバイダサービスの誤った引数の数により JAX-WS ランタイムが IllegalArgumentException をスローしました。今回のアップデートにより引数がチェックされるようになり、引数が予想通り引数リストに追加されるようになりました。

### JBPAPP-6267

64K 以上の MTOM (メッセージ転送最適化メカニズム) が添付されたメッセージをクライアントが JBoss Web サービスに送信すると、応答メッセージの処理が完了する前にサーバーが ServiceEndpointInvoker のスワップファイルを削除し、FileNotFoundException を返しました。応答処理の完了後にスワップファイルが削除されるようになったため、このエラーは発生しないようになりました。

# 第5章 既知の問題

Enterprise Application Platform 5.1.2 での既知の問題をコンポーネント別に一覧にまとめています。

### クラスタリング

#### JBPAPP-6428

同じ EJB が 2 つのクラスタにデプロイされた場合、クラスタ 2 のサーバーの JNDI に InitialContext が示されていても、InvokerInterceptor が クラスタ 2 上の EJB ではなくローカルの EJB を呼び出します。この問題は解決していませんが、この問題を回避するには次の設定内容を JAVA_OPTS 環境変数または $JBOSS_HOME/bin/run.conf に設定します。

```
-Dorg.jboss.invocation.use.partition.name=true
```

### コンソール

#### JBPAPP-5285

Admin Consoleを使ってデータベースプロパティを変更すると、変更が永続化されます。しかし、この変更を元に戻しアプリケーションサーバーを再起動すると以前の値が永続されてしまいます。この問題はまだ解決されていませんが、**server/*PROFILE*/data/attachments** にある関連の添付ファイルを編集しデータソース設定が新規設定を強制的に使用させることで回避可能です。

### 文書

#### JBPAPP-6294

バイナリインストーラーを使用して JBoss Messaging から HornetQ へ Enterprise Application Platform をアップグレードすることができません。個別に HornetQ バイナリをダウンロードし、HornetQ User Guide の手順に従って JBoss Messaging から HornetQ へ移行することが可能です。

### EJB

#### JBPAPP-5980

設定名を「Standard Message Inflow Driven Bean」で上書きし、メッセージインフローを使用するよう設定されていた EJB 2.0 MDB をデプロイしようとした際、サーバーがメッセージタイプを知らないまま JCA インフローを設定しようとしたため、「java.lang.ClassNotFoundException: Null class name」エラーが発生しました。これは EJB 2.0 FR2 15.4.2 および EJB 2.1 FR 15.4.2 では不正な設定となります。この問題を回避するには、**standard-jboss.xml** でインボーカープロキシバインディング名「message-driven-bean」を使用してサーバーが各 MDB を JMS MDB として扱うよう強制します。また、**jboss.xml** ファイルに MDB を個別に設定して対処することも可能です。このように設定された MDB は EJB 2.1 JMS MDB として扱われ、メッセージングタイプ MessageListener を取得します。そのため、メッセージングタイプを上書きする他の動作と競合する可能性があります。

### Hibernate

#### JBPAPP-3034

バッチ挿入ステートメントが要求された時、組み込みクラスが考慮されません。この問題を回避する方法は 2 つあります。 1 つ目は組み込みクラスが使用される時に **ORDER_INSERTS** を **FALSE** に設定する方法です。2 つ目は子オブジェクトで **session.save()** を明示的に呼び出し、SQL の挿

入命令を強制する方法です。

### JBPAPP-5965

フラッシュ中にレイジーコレクションやプロキシを初期化してはいけません。この問題を回避するには、リスナーが別のセッションを使用してコレクションを初期化するようにします。回避方法の例は次の通りです。

```
SessionImplementor si = (SessionImplementor)(event.getSession());
Session anotherSession =
si.getFactory().openSession(si.getJDBCContext().connection());
Object obj = anotherSession.get(
event.getEntity().getClass(),event.getId());
if(o instanceof Parent){
 Parent parent = (Parent)obj;
 Iterator it = parent.getChildren().iterator();
 while(it.hasNext()){
  Child child = (Child)it.next();
 }
}
anotherSession.close()
```

### JBPAPP-6395

LEFT OUTER JOIN で Criteria API を利用し子に基準を追加する場合、子のコレクションにはこの基準と一致する子しか含まれなくなります。この動作は、フィルターを利用しない場合でも適用されます (例: criteria.createCriteria("children", JoinFragment.LEFT_OUTER_JOIN))。この問題の回避する方法はありません。

### JBPAPP-6579

SessionFactoryObjectFactory から発生した警告メッセージ「InitialContext did not implement EventContext」がデバッグ情報の確認時に不安を与える原因となっていました。Hibernate はサーバー設定からセッションファクトリ名を見つけられれば起動後に SessionFactory を JNDI にバインドすることができます。しかし、この警告メッセージは重度な誤解を招くおそれがあります。ユーザーはこのメッセージが表示されないようにすることはできず、通常メッセージ自体は環境固有の問題が原因となっています。この警告メッセージを見ないようにするには、ロギングレベルをデバッグに調整します。

## HornetQ

### JBPAPP-6659

さまざまな理由により、HornetQ にて提供される次の例が動作しません。

- jca-remote
- jms-bridge
- mdb-remote-failover-static
- mdb-remote-failover
- xarecovery
- embedded-simple

- embedded
- failover-manual-stop
- jaas
- spring-integration
- stomp-websockets
- stop-server-failover
- transaction-failover

### JBPAPP-6931

HornetQ のインストーラーは **netty.jar** の独自のバージョンを **jboss-as/common/lib** ディレクトリにコピーしますが、**jboss-as/client** ディレクトリの EAP インストーラーによって同じファイルが既にインストールされているため異なるバージョンの同じファイルが存在することになります。この問題を回避するには、**jboss-as/common/lib** ディレクトリより **netty.jar** を削除します。

### JBPAPP-7612

専用または同じ場所に配置されたトポロジの HornetQ クラスタノードでは、ノードが同時に起動すると Null パラメーター例外が発生する可能性があることが判明しました。ノード間で渡されるトポロジメッセージのタイミングが原因となっているようです。この問題を回避するにはノードを順次起動するようにします。

### JBPAPP-7628

サーバーが要求を処理している最中にプラットフォームがシャットダウンすると RejectExecutionException が発生することがあることが判明しました。以下のようなメッセージがログに記録されます。この問題を回避するには、サーバーをシャットダウンする前にアクティビティがないか最初に確認します。

```
[JBoss] java.util.concurrent.RejectedExecutionException
[JBoss] at
java.util.concurrent.ThreadPoolExecutor$AbortPolicy.rejectedExecution(Th
readPoolExecutor.java:1768)
[JBoss] at
java.util.concurrent.ThreadPoolExecutor.reject(ThreadPoolExecutor.java:7
67)
[JBoss] at
java.util.concurrent.ScheduledThreadPoolExecutor.delayedExecute(Schedule
dThreadPoolExecutor.java:215)
[JBoss] at
java.util.concurrent.ScheduledThreadPoolExecutor.scheduleWithFixedDelay(
ScheduledThreadPoolExecutor.java:443)
[JBoss] at org.hornetq.core.server.impl.QueueImpl.<init>
(QueueImpl.java:306)
[JBoss] at
org.hornetq.core.server.impl.QueueFactoryImpl.createQueue(QueueFactoryIm
pl.java:97)
[JBoss] at
org.hornetq.core.server.impl.HornetQServerImpl.loadJournals(HornetQServe
rImpl.java:1553)
```

```
[JBoss] at
org.hornetq.core.server.impl.HornetQServerImpl.initialisePart2(HornetQSe
rverImpl.java:1429)
[JBoss] at
org.hornetq.core.server.impl.HornetQServerImpl.access$1200(HornetQServer
Impl.java:137)
[JBoss] at
org.hornetq.core.server.impl.HornetQServerImpl$SharedStoreBackupActivati
on.run(HornetQServerImpl.java:1935)
[JBoss] at java.lang.Thread.run(Thread.java:662)
```

### JBPAPP-7792

プラットフォームのシャットダウン中に再接続が再試行されると、OpenJDK を持つ Red Hat Enterprise Linux のプラットフォームで NullPointerException が発生することが判明しました。次のようなメッセージがログに報告されます。この問題を回避するには、サーバーをシャットダウンする前にアクティビティがないことを確認するようにします。

```
[Thread-8 (HornetQ-server-HornetQServerImpl::server 2-10536304)] 16-Dec
7:46:11,567 WARNING [ServerLocatorImpl] NULL
java.lang.NullPointerException
at
org.hornetq.core.server.cluster.impl.ClusterConnectionImpl.onConnection(
ClusterConnectionImpl.java:565)
at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.getConnection(Clie
ntSessionFactoryImpl.java:1313)
at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.getConnectionWithR
etry(ClientSessionFactoryImpl.java:990)
at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.connect(ClientSess
ionFactoryImpl.java:223)
at
org.hornetq.core.client.impl.ServerLocatorImpl$StaticConnector$Connector
.tryConnect(ServerLocatorImpl.java:1778)
at
org.hornetq.core.client.impl.ServerLocatorImpl$StaticConnector.connect(S
erverLocatorImpl.java:1615)
at
org.hornetq.core.client.impl.ServerLocatorImpl.connect(ServerLocatorImpl
.java:592)
at
org.hornetq.core.client.impl.ServerLocatorImpl$3.run(ServerLocatorImpl.j
ava:559)
```

### JBPAPP-7824

ブリッジ間でメッセージの送信中に HornetQ が障害を起こすと、目的のキューの管理カウンターが少数の要素によって影響を受けます。この問題は HornetQ の実際の操作自体には影響せず、統計の報告のみ影響を受けます。

### JBPAPP-7841

稼働中のサーバーがシャットダウンする前に HornetQ のバックアップサーバーがシャットダウンすると IllegalStateException が発生し、次のようなメッセージがログに記録されます。

-

```
11:18:56,234 INFO  [ServerImpl] Runtime shutdown hook called, forceHalt:
true
11:18:56,237 WARN  [StartStopLifecycleAction] Error during stop for
org.hornetq:module=JMS,type=Queue,name="ExpiryQueue"
java.lang.IllegalStateException: Cannot access JMS Server, core server
is not yet active
 at
org.hornetq.jms.server.impl.JMSServerManagerImpl.checkInitialised(JMSSer
verManagerImpl.java:1399)
 at
org.hornetq.jms.server.impl.JMSServerManagerImpl.removeQueueFromJNDI(JMS
ServerManagerImpl.java:618)
 at
org.jboss.as.integration.hornetq.deployers.pojo.HornetQQueueDeployment.s
top(HornetQQueueDeployment.java:60)
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

この問題の回避法はありません。

### JBPAPP-7879

アクティビティの発生中にプラットフォームがシャットダウンすると **ServerLocatorImpl** で **NullPointerException** が発生することが判明しました。次のようなメッセージがログに記録されます。この問題を回避するにはサーバーがシャットダウンする前にアクティビティがないことを確認します。

```
[junit] * [Thread-6 (HornetQ-server-
HornetQServerImpl::serverUUID=147484f9-3aa9-11e1-aff0-1fd4a2111687-
22518320)] 9-Jan 5:2:46,510 WARNING [ServerLocatorImpl] NULL
[junit] java.lang.NullPointerException
[junit] at
org.hornetq.core.client.impl.ServerLocatorImpl$StaticConnector.connect(S
erverLocatorImpl.java:1633)
[junit] at
org.hornetq.core.client.impl.ServerLocatorImpl.connect(ServerLocatorImpl
.java:592)
[junit] at
org.hornetq.core.client.impl.ServerLocatorImpl$3.run(ServerLocatorImpl.j
ava:559)
[junit] at
org.hornetq.utils.OrderedExecutorFactory$OrderedExecutor$1.run(OrderedEx
ecutorFactory.java:100)
[junit] at
java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecuto
r.java:886)
[junit] at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.ja
va:908)
```

### JBPAPP-7880

アクティビティの発生中にプラットフォームがシャットダウンすると **ClientSessionFactoryImpl** で **NullPointerException** が発生することが判明しました。次のようなメッセージがログに記録されます。この問題を回避するにはサーバーがシャットダウンする前にアクティビティがないことを確認します。

```
[junit] java.lang.NullPointerException
    [junit]  at
org.hornetq.core.client.impl.ClientSessionFactoryImpl$PingRunnable.send(
ClientSessionFactoryImpl.java:1656)
    [junit]  at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.getConnection(Clie
ntSessionFactoryImpl.java:1289)
    [junit]  at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.getConnectionWithR
etry(ClientSessionFactoryImpl.java:992)
    [junit]  at
org.hornetq.core.client.impl.ClientSessionFactoryImpl.connect(ClientSess
ionFactoryImpl.java:223)
    [junit]  at
org.hornetq.core.client.impl.ServerLocatorImpl$StaticConnector$Connector
.tryConnect(ServerLocatorImpl.java:1792)
    [junit]  at
org.hornetq.core.client.impl.ServerLocatorImpl$StaticConnector.connect(S
erverLocatorImpl.java:1629)
    [junit]  at
org.hornetq.core.client.impl.ServerLocatorImpl.connect(ServerLocatorImpl
.java:592)
    [junit]  at
org.hornetq.core.client.impl.ServerLocatorImpl$3.run(ServerLocatorImpl.j
ava:559)
    [junit]  at
org.hornetq.utils.OrderedExecutorFactory$OrderedExecutor$1.run(OrderedEx
ecutorFactory.java:100)
    [junit]  at
java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecuto
r.java:886)
    [junit]  at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.ja
va:908)
    [junit]  at java.lang.Thread.run(Thread.java:662)
```

### メッセージング

#### JBPAPP-4668

Oracle 11g R1、R2、RAC で最新の JDBC ドライバーであるバージョン 11.2.0.1.0 を使うと、JBoss Messaging Test Suite の 2 つのテストが失敗します。

- QueueManagementTest.testDestroyDestinationProgrammatically
- QueueManagementTest.testDestroyDestinationProgrammaticallyWithParams

これらのテストは、fullSize キュー設定パラメータに大きな値を使い、**`java.sql.PreparedStatement`**で**`setFetchSize`** メソッドへ渡します。JDBCドライバの問題は、**`executeQuery()`** が呼び出されると、通常のメモリ以上の量を消費していることになります。その結果、**`java.lang.OutOfMemoryError`**が発生しテストが失敗してしまいます。

### RESTEasy

### JBPAPP-4995

2010 年 8 月に Twitter が基本認証メソッドを廃止したため、TwitterClient の例は RESTEasy 1.2.x で廃止されました。そのため、アプリケーションはすべて OAuth を使う必要があります。RESTEasy 2.x には OAuth を用いて編集された TwitterClient の例が含まれています。テスト目的で使用するには、最新バージョンの RESTEasy から例をダウンロードしてください。

## Seam

### JBPAPP-6347

バージョン 4.1.1 以前の JBDS を使用して作成された Seam プロジェクトは次のメッセージを出力してテストフェーズに失敗しました。

```
Could not instantiate Seam component:
org.jboss.seam.security.ruleBasedPermissionResolver
...
Caused by: java.lang.RuntimeException: The Eclipse JDT Core jar is not
in the classpath
```

Drools が 5.1 にアップグレードされ、以前 **lib/core.jar** にあった JBDS が必要とする関数が **lib/ecj.jar** に移動したためこのエラーが発生します。この問題を回避するには、次のオプションの 1 つを実行します。オプション 1 が推奨オプションとなります。

1. JBDS を 4.1.1 にアップグレードします。
2. プロジェクトに lib/ecj.jar をコピーします。

## TCK

### JBPAPP-3929

**java.sql.Date.valueOf** が日付の形式 yyyy-mm-dd を解析しようとすると、TCKテストがjava.lang.IllegalArgumentExceptionをスローしていました。これは、最新のSun JVM、Sun JDK 1.6.0_24の再発バグが原因となっています (詳細はhttp://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6898593 を参照)。Sun JDK 1.6.0_17にダウングレードすることで、この問題を回避できます。

## トランザクションマネージャー

### JBPAPP-7274

XTS トランザクションをデプロイしようとすると次と似たエラーが発生する問題が JBoss Web Services で確認されました。回避するには **-Dorg.jboss.ws.enable_doctype_decl=true** を JBoss Enterprise Application Platform の実行パラメーターに追加します。実行パラメータを追加する方法については『スタートガイド』を参照してください。

```
14:00:57,054 ERROR [AbstractKernelController] Error installing to Real:
name=vfsfile:/tmp/EAP5.1.2/jboss-eap-5.1/jboss-
as/server/all/deploy/jbossxts.sar/ state=PreReal mode=Manual
requiredState=Real
        org.jboss.deployers.spi.DeploymentException: Error during
deploy: vfszip:/tmp/EAP5.1.2/jboss-eap-5.1/jboss-
as/server/all/deploy/jbossxts.sar/ws-c11.war/
                at
```

```
org.jboss.deployers.spi.DeploymentException.rethrowAsDeploymentException
(DeploymentException.java:49)
        .
        .
        Caused by: org.jboss.ws.WSException: Cannot publish wsdl to:
/tmp/EAP5.1.2/jboss-eap-5.1/jboss-
as/server/all/data/wsdl/jbossxts.sar/ws-c11.war/wscoor-activation-
binding.wsdl
                at
org.jboss.wsf.stack.jbws.WSDLFilePublisher.publishWsdlFiles(WSDLFilePubl
isher.java:145)
        .
        .
        Caused by: java.io.IOException: org.xml.sax.SAXParseException:
DOCTYPE is disallowed when the feature
"http://apache.org/xml/features/disallow-doctype-decl" set to true.
                at
org.jboss.wsf.common.DOMUtils.parse(DOMUtils.java:247)
                at
org.jboss.wsf.stack.jbws.WSDLFilePublisher.publishSchemaImports(WSDLFile
Publisher.java:270)
        .
        .
        DEPLOYMENTS IN ERROR:
          Deployment "vfsfile:/tmp/EAP5.1.2/jboss-eap-5.1/jboss-
as/server/all/deploy/jbossxts.sar/" is in error due to the following
reason(s): java.io.IOException: org.xml.sax.SAXParseException: DOCTYPE
is disallowed when the feature "http://apache.org/xml/features/disallow-
doctype-decl" set to true.
```

### Web

#### JBPAPP-4912

Web アプリケーションと依存関係を持つデータソースが再起動されると IllegalStateException がスローされます。

# 付録A 改訂履歴

**改訂 5.1.2-2.400** **2013-10-31** **Rüdiger Landmann**
Rebuild with publican 4.0.0

**改訂 5.1.2-2** **2012-07-18** **Anthony Towns**
Rebuild for Publican 3.0

**改訂 5.1.2-108** **Wed 9 February 2012** **Russell Dickenson**
Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=788543 に従い HornetQ のバージョンを修正。

**改訂 5.1.2-107** **Thu 19 January 2012** **Russell Dickenson**
JBoss Enterprise Application Platform 5.1.2 GA への変更を反映しました。本ガイドの内容変更に関する情報は『リリースノート 5.1.2』 を参照してください。