



Red Hat JBoss Enterprise Application Platform 7.0

移行ガイド

For Use with Red Hat JBoss Enterprise Application Platform 7.0

Red Hat JBoss Enterprise Application Platform 7.0 移行ガイド

For Use with Red Hat JBoss Enterprise Application Platform 7.0

法律上の通知

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

本ガイドでは、以前のバージョンの Red Hat JBoss Enterprise Application Platform からアプリケーションを移行する方法を説明します。

目次

第1章 はじめに	6
1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7	6
1.2. 移行ガイド	6
1.3. 移行およびアップグレード	6
1.4. 本書における EAP_HOME の使用	7
第2章 移行の準備	8
2.1. 準備の概要	8
2.2. JAVA EE 7 機能の確認	8
2.3. JBOSS EAP 7 の新機能	8
2.4. 非推奨および未サポート機能リストの確認	9
2.5. JBOSS EAP 7 スタートガイドの確認	10
2.6. 移行の分析および計画	10
2.7. 重要データのバックアップおよびサーバー状態の確認	12
2.8. RPM インストールの移行	12
2.9. サービスとして実行するよう JBOSS EAP を移行	12
第3章 移行に役立つツール	13
3.1. WINDUP を使用した移行のアプリケーションの分析	13
3.2. JBOSS SERVER MIGRATION TOOL を使用したサーバー設定の移行	13
第4章 サーバー設定の変更	14
4.1. サーバー設定移行オプション	14
JBoss Server Migration Tool	14
管理 CLI の migrate 操作	14
4.2. 管理 CLI の移行操作	14
Start the Server and the Management CLI	15
Migrate the JacORB, Messaging, and Web Subsystems	16
4.3. ログインメッセージの接頭辞の変更	19
4.4. WEB サーバー設定の変更	20
4.4.1. Undertow による Web サブシステムの置換	20
4.4.2. JBoss Web リライト条件の移行	21
4.4.3. Migrate JBoss Web System Properties	23
4.4.4. Migrate JBoss Web jboss-web.xml Overlay	23
4.4.5. グローバルバルブの移行	24
JBoss Web バルブの移行	24
JDBCAccessLogValve の手動移行の手順	25
4.5. JGROUPS サーバー設定の変更	26
4.5.1. JGroups はデフォルトでプライベートネットワークインターフェースを使用	26
4.5.2. JGroups チャネルの変更	26
4.6. INFINISPAN サーバー設定の変更	26
4.6.1. Infinispan のデフォルトキャッシュ設定の変更	26
4.6.2. Infinispan のキャッシュストラテジーの変更	27
4.7. EJB SERVER CONFIGURATION CHANGES	27
DuplicateServiceException	27
4.8. メッセージングサーバー設定の変更	28
4.8.1. メッセージングサブシステムサーバー設定の変更	28
管理モデル	28
Messaging サブシステムの移行および前方互換性	29
Change in Behavior of forward-when-no-consumers Attribute	29
Messaging サブシステムの XML 設定	29
4.8.2. メッセージングデータの移行	29

4.8.2.1. エクスポートおよびインポートを使用したメッセージングデータの移行	30
前リリースからのメッセージングデータのエクスポート	30
XML 形式のメッセージングデータのインポート	31
4.8.2.2. JMS ブリッジを使用したメッセージングデータの移行	31
JBoss EAP 6 サーバーの設定	32
JBoss EAP 7 サーバーの設定	32
データの移行	35
4.8.2.3. メッセージングフォルダー名のマッピング	35
4.8.3. JMS 宛先の移行	35
4.8.4. メッセージングインターセプターの移行	36
4.8.5. Netty サブレット設定の変更	36
4.9. JMX MANAGEMENT CHANGES	36
4.10. ORB サーバー設定の変更	38
4.11. THREADS サブシステム設定の移行	41
4.12. REMOTING サブシステム設定の移行	41
4.13. WEBSOCKET サーバー設定の変更	42
4.14. シングルサインオンサーバーの変更	42
4.15. データソース設定の変更	43
4.15.1. JDBC データソースドライバー名	43
単一クラスを含むドライバー	43
複数クラスを含むドライバー	43
4.16. セキュリティサーバー設定の変更	43
4.17. トランザクションサブシステムの変更	43
Removed Transactions Subsystem Attributes	44
非推奨となったトランザクションサブシステム属性	44
4.18. MOD_CLUSTER 設定の変更	44
第5章 アプリケーション移行の変更	45
5.1. WEB サービスアプリケーションの変更	45
5.1.1. JAX-RPC サポートの変更	45
5.1.2. Apache CXF Spring Web サービスの変更	45
Apache CXF インターセプター	46
Apache CXF の機能	46
Apache CXF HTTP トランスポート	47
5.1.3. WS-Security の変更	48
5.1.4. JBoss モジュール構造の変更	48
5.1.5. Bouncy Castle の要件の変更	48
5.1.6. Apache CXF バス選択ストラテジー	48
5.1.7. WebServiceRef の JAX-WS 2.2 要件	49
5.1.8. IgnoreHttpsHost CN チェックの変更	49
5.1.9. サーバー側設定およびクラスローディング	49
5.1.10. Java Endorsed Standards Override Mechanism の非推奨	49
5.1.11. EAR アーカイブでの記述子の仕様	49
5.2. リモート URL コネクターおよびポートの更新	49
5.3. メッセージングアプリケーションの変更	50
5.3.1. JMS デプロイメント記述子の置き換えおよび更新	50
5.3.2. 外部 JMS クライアントの更新	51
5.3.3. HornetQ API の置換	51
5.4. JAX-RS および RESTEASY アプリケーションの変更	51
5.4.1. 非推奨の RESTEasy クラス	52
インターセプターおよび MessageBody クラス	52
クライアント API	54
StringConverter	55

5.4.2. 削除または保護されている RESTEasy クラス	55
ResteasyProviderFactory Add メソッド	55
RESTEasy 3 から削除された他のクラス	55
5.4.3. 他の RESTEasy 変更点	55
SignedInput および SignedOutput	55
セキュリティーフィルター	56
クライアント側のフィルター	56
非同期 HTTP サポート	56
サーバー側のキャッシュ	56
5.4.4. RESTEasy SPI の変更点	56
SPI 例外	56
InjectorFactory および Registry	56
5.4.5. Jackson プロバイダーの変更	57
5.4.6. Spring RESTEasy 統合の変更	57
5.4.7. RESTEasy Jettison JSON プロバイダーの変更	57
5.5. CDI 1.2 アプリケーションの変更	57
Bean アーカイブ	58
会話解決の明確化	59
オブザーバー解決	59
5.6. 明示的なモジュール依存関係の移行	59
利用可能な依存関係の確認	59
アノテーションのスキャンに必要な依存関係	59
5.7. HIBERNATE および JPA の移行の変更	60
5.7.1. Hibernate ORM 3.0	60
5.7.2. Hibernate ORM 4.0 - 4.3	60
5.7.3. Hibernate ORM 5	60
5.7.4. Migrating to Hibernate ORM 5	60
Removed and Deprecated Classes	60
Other Changes to Classes and Packages	60
Type Handling	61
Transaction Management	61
Other Hibernate ORM 5 Changes	62
5.8. HIBERNATE SEARCH の変更	63
Hibernate Search マッピングの変更	63
埋め込み関係の ID フィールドのインデックス化	63
番号および日付のインデックス形式の変更	63
その他の Hibernate Search の変更	63
Hibernate Search の名前変更および再パッケージされたクラス	65
Lucene - 名前変更および再パッケージされたクラス	65
Hibernate Search で非推奨となった API	65
Hibernate Search で非推奨となったインターフェース	65
Hibernate Search で非推奨となったクラス	66
Hibernate Search で非推奨となった列挙	66
Hibernate Search で非推奨となったアノテーション	66
Hibernate Search で非推奨となったメソッド	66
Hibernate Search で非推奨となったコンストラクター	67
上級インテグレーターに影響する変更	67
5.9. エンティティ BEAN の JPA への移行	68
5.10. JPA 永続プロパティの変更	70
5.11. EJB クライアントコードの移行	70
5.11.1. デフォルトのリモート接続ポートの更新	70
5.11.2. デフォルトコネクタの更新	71
5.11.3. リモートネーミングクライアントコードの移行	71

5.12. デプロイメント計画設定の移行	72
5.13. カスタムアプリケーションバルブの移行	72
デプロイメントで設定されたバルブの移行	72
カスタムオーセンティケーターバルブの移行	73
5.14. セキュリティーアプリケーションの変更	73
5.14.1. オーセンティケーターバルブの移行	73
5.14.2. PicketLink の変更	73
5.14.3. その他のセキュリティーアプリケーションの変更	73
5.15. JBOSS LOGGING の変更	73
5.16. JAVASERVER FACES (JSF) のコード変更	74
JSF 1.2 のサポート停止	74
JSF 2.1 および JSF 2.2 の互換性の問題	74
5.17. モジュールクラスローティングの変更	74
5.18. アプリケーションクラスタリングの変更	75
5.18.1. 新しいクラスタリング機能の概要	75
5.18.2. Web セッションクラスタリングの変更	76
5.18.3. ステートフルセッション EJB クラスタリングの変更	78
5.18.4. クラスタリングサービスの変更	79
5.18.5. クラスタリング HA シングルトンの移行	79
第6章 その他の変更	80
6.1. JBOSS EAP ネイティブおよび APACHE HTTP SERVER の提供に関する変更	80
6.2. AMAZON EC2 でのデプロイメントの変更	81
6.3. CHANGES TO JBOSS EAP SCRIPTS	81
6.4. REMOVAL OF OSGI SUPPORT	81
第7章 MIGRATING FROM OLDER RELEASES OF JBOSS EAP	83
7.1. MIGRATING FROM JBOSS EAP 5 TO JBOSS EAP 7	83
7.2. SUMMARY OF CHANGES MADE TO EACH RELEASE	83
7.3. REVIEW THE CONTENT IN THE MIGRATION GUIDES	84
7.4. JBOSS EAP 5 COMPONENT UPGRADE REFERENCE	84
付録A リファレンス資料	91
A.1. JACORB サブシステム移行操作の警告	91
A.2. MESSAGING サブシステム移行操作の警告	92
非推奨の broadcast-group または discovery-group 属性の置き換え	96
A.3. WEB サブシステム移行操作の警告	97
Web Subsystem Migration Operation Attribute Warnings	101
Web SSL Connector Attributes	101
Web Static Resource Attributes	101
Web SSO Resource Attributes	102
Web Access Log Attributes	102
Web Connector Attributes	102
A.4. MIGRATE JBOSS WEB SYSTEM PROPERTIES REFERENCE	103
A.5. リリース間の互換性および相互運用性	111
IIOP 上の EJB リモータリング	112
JNDI を使用した EJB リモータリング	112
@WebService を使用した EJB リモータリング	112
メッセージングスタンドアロンクライアント	112
メッセージング MDB	113
JMS ブリッジ	113

第1章 はじめに

1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7

Red Hat JBoss Enterprise Application Platform 7 (JBoss EAP) は、オープンな標準に基いて構築され、Java Enterprise Edition 7 の仕様に準拠するミドルウェアプラットフォームです。メッセージング、高可用性クラスタリングなどの技術が WildFly Application Server 10 と統合されます。

JBoss EAP には、必要な場合にだけサービスを有効にできるモジュール構造が含まれ、サービスの起動時間が短縮されます。

管理コンソールと管理コマンドラインインターフェース (CLI) により、XML 設定ファイルの編集が不要になり、タスクをスクリプト化および自動化する機能が追加されました。

JBoss EAP は、JBoss EAP インスタンスに対してスタンドアロンサーバーと管理対象ドメインの 2 つの操作モードを提供します。スタンドアロンサーバー操作モードでは、実行している JBoss EAP を 1 つのサーバーインスタンスとして表します。管理対象ドメイン操作モードでは、1 つの制御ポイントから複数の JBoss EAP インスタンスを管理できます。

また、JBoss EAP には、セキュアでスケーラブルな Java EE アプリケーションの迅速な開発を可能にする API と開発フレームワークが含まれます。

1.2. 移行ガイド

The purpose of this guide is to document the changes that are required to successfully run and deploy Red Hat JBoss Enterprise Application Platform 6 applications on Red Hat JBoss Enterprise Application Platform 7. It provides information about the new features available in this release, the deprecated and unsupported features, and any application and server configuration updates that might be required to prevent changes in application behavior.

It also provides information about tools that can help with the migration, such as [Windup](#), which simplifies migration of Java applications, and the [JBoss Server Migration Tool](#), which updates the server configuration.

Once the application is successfully deployed and running, plans can be made to upgrade individual components to use the new functions and features of JBoss EAP 7.

If you plan to migrate your JBoss EAP 5 applications directly to JBoss EAP 7, see [Migrating from Older Releases of JBoss EAP](#).

1.3. 移行およびアップグレード

メジャーアップグレード

A major upgrade or migration is required when an application is moved from one major release to another, for example, from JBoss EAP 6 to JBoss EAP 7. This is the type of migration addressed in this guide. If an application follows the Java EE specifications, does not access deprecated APIs, and does not contain proprietary code, it might be possible to run the application in JBoss EAP 7 without any application code changes. However, server configuration has changed in JBoss EAP 7 and any server configuration settings require migration.

マイナーアップデート

JBoss EAP periodically provides point releases, which are minor updates that include bug fixes, security fixes, and new features. The JBoss EAP [Patching and Upgrading Guide](#) describes how to upgrade from one point release to another, for example from JBoss EAP 7.0 to JBoss EAP 7.1.

累積パッチ

JBoss EAP also periodically provides cumulative patches that contain bug and security fixes. Cumulative patches increment the release by the last digit, for example from 7.0.0 to 7.0.1. Patch installation is covered in detail in the JBoss EAP [Patching and Upgrading Guide](#).

1.4. 本書における **EAP_HOME** の使用

本書では、変数 **EAP_HOME** を使用して JBoss EAP へのパスを示しています。この変数は JBoss EAP インストールへの実際のパスに置き換えてください。

- ZIP インストール方法で JBoss EAP をインストールした場合、インストールディレクトリーは、ZIP アーカイブを抽出した **jboss-eap-7.0** ディレクトリーとなります。
- RPM インストール方法で JBoss EAP をインストールした場合、インストールディレクトリーは **/opt/rh/eap7/root/usr/share/wildfly/** になります。
- インストーラーを使用して JBoss EAP をインストールした場合、**EAP_HOME** のデフォルトのパスは **`\${user.home}/EAP-7.0.0** になります。
 - Red Hat Enterprise Linux、Solaris、および HP-UX では、**/home/USER_NAME/EAP-7.0.0/** になります。
 - Microsoft Windows の場合、**C:\Users\USER_NAME\EAP-7.0.0** になります。
- JBoss Developer Studio インストーラーを使用して JBoss EAP サーバーをインストールおよび設定した場合、**EAP_HOME** のデフォルトのパスは **`\${user.home}/jbdevstudio/runtimes/jboss-eap** になります。
 - Red Hat Enterprise Linux の場合、**/home/USER_NAME/jbdevstudio/runtimes/jboss-eap/** になります。
 - Microsoft Windows の場合、**C:\Users\USER_NAME\jbdevstudio\runtimes\jboss-eap** または **C:\Documents and Settings\USER_NAME\jbdevstudio\runtimes\jboss-eap** になります。



注記

EAP_HOME は環境変数ではありません。**JBoss_HOME** がスクリプトで使用される環境変数です。

第2章 移行の準備

2.1. 準備の概要

In JBoss EAP 7, an effort was made to provide backward compatibility for JBoss EAP 6 applications. However, if your application uses features that were deprecated or functionality that was removed from JBoss EAP 7, you might need to make changes to your application code.

In addition, a number of things have changed in this release that might impact deployment of JBoss EAP 7 applications. It is recommended that you do some research and planning before you attempt to migrate your application.

- [Java EE 7 の機能](#)を理解するようにしてください。
- [JBoss EAP 7 の新機能](#)を確認してください。
- [非推奨の機能およびサポートされない機能](#)のリストを確認してください。
- Review the material in the JBoss EAP 7 [Getting Started Guide](#).
- [移行に便利なツール](#)を確認してください。

機能の変更、開発の資料、および移行に便利なツールについて理解したら、アプリケーションとサーバー設定を評価し、JBoss EAP 7 で実行するために必要な変更について判断します。

2.2. JAVA EE 7 機能の確認

Java EE 7 には、プライベートおよびパブリッククラウドでの機能が充実したアプリケーションの開発や実行を容易にする多くの改良点が含まれています。JBoss EE 7 は、HTML5、WebSocket、JSON、Batch、および Cocurrency Utilities などの新機能や最新の標準が導入されています。アップデートには JPA 2.1、JAX-RS 2.0、Servlet 3.1、Expression Language 3.0、JMS 2.0、JSF 2.2、EJB 3.2、CDI 1.2、および Bean Validation 1.1 が含まれます。

チュートリアルなどの Java EE 7 に関する詳細は、オラクルの Web サイトにある [Java EE at a Glance](#) を参照してください。

2.3. JBOSS EAP 7 の新機能

JBoss EAP 7 には、以前のリリースから大幅に改善されたアップグレードや改良点が含まれています。

Java EE 7

JBoss EAP 7 は Java EE 7 の認定実装で、Web Profile と Full Profile の両方に準拠しています。また、CDI 1.2 および Web Sockets 1.1 の最新のイテレーションもサポートします。

Undertow

Undertow は軽量で柔軟性のあるパフォーマンスに優れた新しい Web サーバーです。JBoss Web は Undertow に置き換えられました。Undertow は Java で書かれ、スループットとスケーラビリティを最大にするよう設計されています。新しい HTTP/2 標準などの最新の Web 技術をサポートします。

Apache ActiveMQ Artemis

Apache ActiveMQ Artemis は JBoss EAP 7 の新しいビルトインメッセージングプロバイダーです。この Apache サブプロジェクトは HornetQ から寄贈されたコードをベースにし、証明された非ブロッキングアーキテクチャーを基に優れたパフォーマンスを実現します。

IronJacamar 1.2

最新の Iron Jacamar は、安定性が高く、機能が充実したサポートを JCA および DataSources に提供します。

JBossWS 5

The fifth generation of JBossWS is a major leap forward, bringing new features and performance improvements to JBoss EAP 7 web services.

RESTEasy 3

JBoss EAP 7 には最新の RESTEasy が含まれています。JSON Web Encryption、Jackson、YAML、JSON-P、Jettison などの便利な拡張を提供し、標準の Java EE REST API (JAX-RS 2.0) を越えた機能性を実現します。

OpenJDK ORB

JBoss EAP では、JacORB IIOP 実装が OpenJDK ORB のダウンストリームブランチに置き換えられ、JVM ORB と Java EE RI との相互運用性が向上されました。

機能が充実したクラスタリング

JBoss EAP 7 ではクラスタリングのサポートが大幅にリファクタリングされ、アプリケーションのアクセスを可能にするパブリック API が複数含まれています。

ポートの削減

JBoss EAP 7 では HTTP のアップグレードを利用し、ほぼすべてのプロトコルが管理ポート (9990) とアプリケーションポート (8080) の 2 つの HTTP ポート上で多重化されます。

ロギングの強化

管理 API が、サーバー上で利用可能なログファイルをリストおよび表示する機能をサポートするようになりました。また、デフォルトのパターンフォーマッター以外のカスタムフォーマッターを定義する機能もサポートするようになりました。さらに、デプロイメントのロギング設定も大幅に向上されました。

For a complete list of new features, see [New Features and Enhancements](#) in the JBoss EAP 7 **Release Notes**.

2.4. 非推奨および未サポート機能リストの確認

Before you migrate your application, be aware that some features that were available in previous releases of JBoss EAP might be deprecated or no longer supported.

メンテナンスコストの高さ、コミュニティでの関心の低さ、および高性能な代替ソリューションの存在などが理由で、一部の技術がサポート対象外になりました。サポートされない機能の一部を以下に示します。

EJB エンティティ Bean

EJB エンティティ Bean はサポート対象外になりました。アプリケーションが EJB エンティティ Bean を使用する場合は、パフォーマンスや柔軟性が高い API を提供する JPA を使用するようコードを移行してください。

JAX-RPC

JAX-WS はより正確で完全なソリューションを提供するため、JAX-RPC 用に書かれたコードは JAX-WS を使用するよう移行する必要があります。

JSR-88

すべての Java EE プラットフォーム製品でアプリケーションを設定およびデプロイするために複数のプロバイダーからツールを有効化するコントラクトを定義する、Java EE Application Deployment API 仕様 (JSR-88) は広く採用されませんでした。管理コンソール、管理 CLI、デプロイメントス

キャナー、Maven など、アプリケーションのデプロイメントでは JBoss EAP がサポートする他のオプションを使用する必要があります。

汎用 JMS リソースアダプター

汎用 JMS リソースアダプターを設定して JMS プロバイダーに接続する機能は、JBoss EAP 7 ではサポート対象外になりました。

For a comprehensive list of deprecated and unsupported features, see [Unsupported and Deprecated Functionality](#) in the JBoss EAP 7 **Release Notes**.

2.5. JBOSS EAP 7 スタートガイドの確認

Be sure to review the JBoss EAP [Getting Started Guide](#). It contains the following important information:

- JBoss EAP のダウンロードおよびインストール方法
- Red Hat JBoss Developer Studio のダウンロードおよびインストール方法
- 開発環境に応じた Maven の設定方法、プロジェクト依存関係の管理方法、および JBoss EAP の Bill of Material (BOM) アーティファクトを使用するようプロジェクトを設定する方法。
- 製品に同梱されたクイックスタートサンプルアプリケーションのダウンロードおよび実行方法

2.6. 移行の分析および計画

Each application and server configuration is unique and you must thoroughly understand the components and architecture of the existing application and server platform before you attempt the migration. Your migration plan should include a detailed roadmap for testing and roll out to production that takes into account the following information.

移行責任者の特定

ステークホルダー、プロジェクトマネージャー、開発者、管理者、およびその他の移行責任者を特定します。

アプリケーションサーバープラットフォーム設定とハードウェアの確認

既存のアプリケーションサーバーとプラットフォーム設定を検証し、JBoss EAP 7 の今後の変更がどのように影響するかを判断します。以下の項目が含まれる必要があります。

- オペレーティングシステムおよびバージョン
- アプリケーションによって使用されるデータベース
- Web サーバー
- セキュリティーアーキテクチャー
- プロセッサの数およびタイプ
- メモリーの容量
- 物理ディスクストレージの容量
- データベースまたはメッセージングデータの移行
- Other components that might be impacted by the migration

現在の本番環境の確認

移行プロセスのテストおよびステージングでは、できるだけ本番環境に近い状態を再現するように計画を立てる必要があります。

- Take into account any clustering configurations. See [Upgrading a Cluster](#) in the [JBoss EAP Patching and Upgrading Guide](#) for more information about how to migrate clusters.
- 現在、大型の管理対象ドメインを実行している場合は、段階的な移行方法を考慮してください。
- データベースまたはメッセージングデータの移行が必要であるかどうかを判断します。

既存アプリケーションの検証および理解

既存の JBoss EAP 6 アプリケーションを完全に検証します。以下を含むアーキテクチャー、関数、機能、およびコンポーネントについて完全に理解してください。

- JVM バージョン
- 他の Red Hat アプリケーションサーバーミドルウェアコンポーネントとの統合
- プロプライエタリーサードパーティソフトウェアとの統合
- 代替機能が必要となる非推奨機能の使用
- デプロイメント記述子、JNDI、永続性、JDBC 設定およびプーリング、JMS トピックおよびキュー、ロギングを含むアプリケーション設定

JBoss EAP 7 への移行中に変更が必要な互換性のないコード設定を特定

詳細テストプランの作成

- 計画には、回帰テストと受け入れ基準の要件が含まれる必要があります。
- パフォーマンステストが含まれる必要もあります。
- 本番環境へロールアウトする前に移行をテストするため、できるだけ本番環境に近くなるようステージング環境を設定します。
- 必ず、バックアップおよびバックアウト計画を作成してください。

移行プロセスに使用できるリソースの確認

- 開発チームのスキルを評価し、トレーニングまたは追加のコンサルティングを計画します。
- 移行プロセス中は、ステージングやテストで追加のハードウェアやその他のリソースが必要になることにも注意してください。
- 正式なトレーニングが必要であるかどうかを判断します。必要な場合はスケジュールに追加します。

計画の遂行

必要なリソースを確保し、移行計画を実行します。



重要

必ずバックアップコピーを作成してからアプリケーションに変更を加えるようにしてください。

2.7. 重要データのバックアップおよびサーバー状態の確認

アプリケーションを移行する前に、以下の問題が発生する可能性があることを認識しておく必要があります。

- The migration might remove temporary folders. Any deployments stored in the **data/content/** directory must be backed up prior to the migration and restored after it completes. Otherwise, the server will fail to start due to the missing content.
- 移行前に、開かれたトランザクションをすべて処理し、**data/tx-object-store/** トランザクションディレクトリーを削除します。
- **data/timer-service-data** にある永続タイマーデータをチェックし、互換性があるかどうかを判断する必要があります。移行前に、このディレクトリーにある **deployment-*** ファイルをチェックし、使用されているタイマーを確認します。

移行を開始する前に、現在のサーバー設定とアプリケーションもバックアップするようにしてください。

2.8. RPM インストールの移行



重要

単一の Red Hat Enterprise Linux サーバーでサポートされるのは、RPM でインストールされた JBoss EAP のインスタンス 1 つまでです。そのため、JBoss EAP 7 に移行の際は、使用中の JBoss EAP インストールを新規マシンに移行することが推奨されます。

JBoss EAP RPM インストールを JBoss EAP 6 から JBoss EAP 7 に移行する際は、JBoss EAP 7 のインストール先となるマシンに既存の JBoss EAP RPM インストールがないことを確認してください。

To install JBoss EAP 7 using RPMs, see the JBoss EAP [Installation Guide](#).

The migration advice in this guide also applies to migrating RPM installations of JBoss EAP, but you might need to alter some steps (such as how to start JBoss EAP) to suit an RPM installation compared to a ZIP or installer installation.

2.9. サービスとして実行するよう **JBoss EAP** を移行

If you run JBoss EAP 6 as a service, be sure to review [Configuring JBoss EAP to Run as a Service](#) in the JBoss EAP [Installation Guide](#) for updated configuration instructions for JBoss EAP 7.

第3章 移行に役立つツール

3.1. WINDUP を使用した移行のアプリケーションの分析

Red Hat JBoss Migration Toolkit の一部である Windup は、拡張およびカスタマイズ可能なルールベースのツールで、Java アプリケーションの移行を容易にします。Windup は移行予定のアプリケーションによって使用される API、技術、およびアーキテクチャーを分析し、各アプリケーションの詳細な移行レポートを提供します。レポートには以下の情報が含まれます。

- 必要な移行変更の詳細
- 変更が必須または任意であるかどうか
- 変更が複雑または簡単であるかどうか
- 移行変更が必要なコードへのリンク
- 必要な変更を行うためのヒントおよび情報へのリンク
- 見つかった各移行問題の推定作業量レベルおよびアプリケーションを移行するための推定合計作業量

Windup を使用すると、JBoss EAP 6 アプリケーションを JBoss EAP 7 へ移行する前にこれらのアプリケーションのコードやアーキテクチャーを分析できます。JBoss EAP 6 から JBoss EAP 7 への移行に対する Windup のルールセットは XML 記述子についてレポートし、JBoss EAP 7 への移行時に代替の設定に置き換える必要がある特定のアプリケーションコードおよびパラメーターについてもレポートします。

For more information about how to use Windup to analyze your JBoss EAP 6 applications, see the [Windup User Guide](#).

3.2. JBOSS SERVER MIGRATION TOOL を使用したサーバー設定の移行

現在開発中の JBoss Server Migration Tool を使用して、既存の設定を保持しながら JBoss EAP 7 の新機能や新設定が含まれるように設定を更新することが推奨されます。

JBoss Server Migration Tool は既存の JBoss EAP 6 サーバー設定ファイルを読み取り、新しいサブシステムの設定を追加します。さらに、既存のサブシステム設定を新機能で更新し、廃止されたサブシステム設定を削除します。

JBoss Server Migration Tool の最新のプレリリース版バイナリディストリビューションは、<https://github.com/wildfly/wildfly-server-migration/releases> からダウンロードできます。このバージョンは、JBoss EAP 6.4 から JBoss EAP 7.0 へのスタンドアロンサーバーの移行をサポートします。このツールの使用方法については、[JBoss Server Migration Tool User Guide](#) を参照してください。



注記

JBoss Server Migration Tool のプレリリース版はサポートされません。

第4章 サーバー設定の変更

4.1. サーバー設定移行オプション

サーバー設定を JBoss EAP 6 から JBoss EAP 7 に移行するには、JBoss Server Migration Tool 使用するか、管理 CLI の `migrate` 操作を使用して手作業で移行します。

JBoss Server Migration Tool

現在開発中の JBoss Server Migration Tool を使用して、既存の設定を保持しながら JBoss EAP 7 の新機能や新設定が含まれるように設定を更新することが推奨されます。

管理 CLI の `migrate` 操作

管理 CLI の `migrate` 操作を使用して JBoss EAP 6 設定ファイルの `jacorb`、`messaging`、および `web` サブシステムを更新すると、新しいリリースで実行できるようになりますが、完全な JBoss EAP 7 の設定にはならないことに注意してください。例を以下に示します。

- この操作は、元の `remote` プロトコルおよびポート設定を JBoss EAP 7 で使用される新しい `http-remoting` およびポート設定に更新しません。
- 設定には、新しい JBoss EAP サブシステム、クラスター化されたシングルtonデプロイメントなどの機能、正常シャットダウンが含まれません。
- 設定には、バッチ処理などの新しい Java EE 7 の機能が含まれません。
- The `migrate` operation does not migrate the `ejb3` subsystem configuration. For information about possible EJB migration issues, see [EJB Server Configuration Changes](#).

`migrate` 操作を使用したサーバー設定の移行に関する詳細は、[管理 CLI の移行操作](#)を参照してください。

4.2. 管理 CLI の移行操作

管理 CLI を使用して、JBoss EAP 7 上で実行するよう JBoss EAP 6 のサーバー設定ファイルを更新することができます。管理 CLI では、`jacorb`、`messaging`、および `web` サブシステムを以前のリリースから新しい設定へ自動的に更新する `migrate` 操作を実行できます。また、`jacorb`、`messaging`、および `web` サブシステムに `describe-migration` 操作を実行すると、移行を行う前に提案された移行設定の変更を確認することもできます。`cmp`、`jaxr`、および `threads` サブシステムの代替はないため、これらのサブシステムをサーバー設定から削除する必要があります。



重要

`migrate` 操作の制限については、[サーバー設定移行オプション](#)を参照してください。現在開発中の JBoss Server Migration Tool を使用して、既存の設定を保持しながら JBoss EAP 7 の新機能や新設定が含まれるように設定を更新することが推奨されます。

表4.1 サブシステムの移行および管理 CLI 操作

JBoss EAP 6 サブシステム	JBoss EAP 7 サブシステム	管理 CLI 操作
<code>cmp</code>	代替なし	<code>remove</code>
<code>jacorb</code>	<code>iiop-openjdk</code>	<code>migrate</code>

JBoss EAP 6 サブシステム	JBoss EAP 7 サブシステム	管理 CLI 操作
jaxr	代替なし	remove
messaging	messaging-activemq	migrate
threads	代替なし	remove
web	undertow	migrate

Start the Server and the Management CLI

以下の手順に従って、JBoss EAP 7 上で実行されるよう JBoss EAP 6 のサーバー設定を更新します。

1. 移行を始める前に、[重要データのバックアップおよびサーバー状態の確認](#)の内容を見直してください。ここでは、サーバーを良好な状態にし、適切なファイルをバックアップするための重要な情報が記載されています。
2. JBoss EAP 6 の設定で JBoss EAP 7 のサーバーを起動します。
 - a. JBoss EAP 7 サーバー設定ファイルをバックアップします。
 - b. 以前のリリースの設定ファイルを JBoss EAP 7 ディレクトリーにコピーします。

```
$ cp EAP6_HOME/standalone/configuration/standalone-full.xml
EAP7_HOME/standalone/configuration
```

- c. JBoss EAP 7 のインストールディレクトリーへ移動し、**--admin-only** 引数を使用してサーバーを起動します。

```
$ bin/standalone.sh -c standalone-full.xml --admin-only
```



注記

サーバーを起動すると、以下の

org.jboss.as.controller.management-operation エラーがサーバーログに記録されます。これらのエラーは予期されたエラーで、レガシーサブシステムの設定を削除するか JBoss EAP 7 に移行する必要があることを示しています。

- WFLYCTL0402: Subsystems [cmp] provided by legacy extension 'org.jboss.as.cmp' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [jacob] provided by legacy extension 'org.jboss.as.jacob' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [jaxr] provided by legacy extension 'org.jboss.as.jaxr' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [messaging] provided by legacy extension 'org.jboss.as.messaging' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [threads] provided by legacy extension 'org.jboss.as.threads' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [web] provided by legacy extension 'org.jboss.as.web' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.

3. 新しいターミナルを開いて、JBoss EAP 7 のインストールディレクトリーへ移動し、**--controller=remote://localhost:9999** 引数を使用して管理 CLI を開始します。

```
$ bin/jboss-cli.sh --connect --controller=remote://localhost:9999
```

Migrate the JacORB, Messaging, and Web Subsystems

1. 移行を行う前にサブシステムに追加した設定変更を確認するには、**describe-migration** 操作を実行します。

describe-migration 操作では以下の構文を使用します。

```
/subsystem=SUBSYSTEM_NAME:describe-migration
```

以下の例は、JBoss EAP 7 に移行されるときに JBoss EAP 6.4 の **standalone-full.xml** 設定ファイルに加えられる設定の変更を示しています。見やすくするため、エントリーは出力から削除されています。

例: describe-migration 操作

```
[standalone@localhost:9999 /] /subsystem=messaging:describe-
migration
{
  "outcome" => "success",
  "result" => {
    "migration-warnings" => [],
    "migration-operations" => [
      {
        "operation" => "add",
        "address" => [{"extension" =>
"org.wildfly.extension.messaging-activemq"}],
        "module" => "org.wildfly.extension.messaging-
activemq"
      },
      {
        "operation" => "add",
        "address" => [{"subsystem" => "messaging-
activemq"}]
      },
      <!-- *** Entries removed for readability *** -->
      {
        "operation" => "remove",
        "address" => [{"subsystem" => "messaging"}]
      },
      {
        "operation" => "remove",
        "address" => [{"extension" =>
"org.jboss.as.messaging"}]
      }
    ]
  }
}
```

2. **migrate** 操作を実行し、サブシステムの設定を JBoss EAP 7 の代替サブシステムに移行します。この操作では以下の構文を使用します。

```
/subsystem=SUBSYSTEM_NAME:migrate
```

**注記**

messaging サブシステムの **describe-migration** および **migrate** 操作を使用すると、引数を渡してレガシークライアントによるアクセスを設定することができます。コマンド構文の詳細は、[Messaging サブシステムの移行および前方互換性](#)を参照してください。

3. このコマンドの結果を確認します。必ず、操作が正常に完了し、"migration-warning" エントリがないことを確認してください。これは、サブシステムの移行設定が完了したことを意味します。

例: 警告のない成功した migrate 操作

```
[standalone@localhost:9999 /] /subsystem=messaging:migrate
```

```
{
  "outcome" => "success",
  "result" => {"migration-warnings" => []}
}
```

サーバー設定の移行が正常に完了したにも関わらず、すべての要素と属性を移行できなかった場合、ログに "migration-warnings" エントリーが表示されます。"migration-warnings" が示す提案に従い、追加の管理 CLI コマンドを実行してこれらの設定を編集する必要があります。"migration-warnings" を返す **migrate** 操作の例を以下に示します。

例: 警告のある migrate 操作

```
[standalone@localhost:9999 /] /subsystem=messaging:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYMSG0080: Could not migrate attribute group-address from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupB\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupB\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute local-bind-address
from resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute local-bind-port
from resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-address from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"broadcast-group\\" => \\"groupA\\")
]. Use instead the socket-binding attribute to configure this
broadcast-group."
  ]
}
```

```

    ], Use instead the socket-binding attribute to configure this
    broadcast-group."
  ]}
}

```

注記

各サブシステムの **migrate** および **describe-migration** 警告のリストは、本ガイドの最後にある [リファレンス資料](#) に記載されています。

- [Jacorb サブシステム移行操作の警告](#)
- [メッセージングサブシステム移行操作の警告](#)
- [Web サブシステム移行操作の警告](#)

4. サーバー設定ファイルを確認し、拡張、サブシステム、および名前空間が更新され、既存のサブシステム設定が JBoss EAP 7 に移行されたことを確認します。

注記

この手順は、以下のコマンドを使用して **jacorb**、**messaging**、および **web** の各サブシステムに対して繰り返す必要があります。

```

/subsystem=jacorb:migrate
/subsystem=messaging:migrate
/subsystem=web:migrate

```

5. **cmp**、**jaxr**、および **threads** サブシステムおよび拡張をサーバー設定から削除します。管理 CLI プロンプトで以下のコマンドを実行し、廃止された **cmp**、**jaxr**、および **threads** サブシステムを削除します。

```

/subsystem=cmp:remove
/extension=org.jboss.as.cmp:remove
/subsystem=jaxr:remove
/extension=org.jboss.as.jaxr:remove
/subsystem=threads:remove
/extension=org.jboss.as.threads:remove

```

重要

サーバーを再起動する前に **messaging**、**jacorb**、および **web** サブシステムを移行し、**cmp**、**jaxr**、および **threads** 拡張およびサブシステムを削除する必要があります。この作業の完了前にサーバーを再起動する必要がある場合は、サーバー起動のコマンドラインで **--admin-only** 引数を使用するようにしてください。これにより、サーバーの変更を継続できます。

4.3. ロギングメッセージの接頭辞の変更

ログメッセージには、メッセージを報告するサブシステムのプロジェクトコードが接頭辞として付けられます。JBoss EAP 7 では、すべてのログメッセージの接頭辞が変更になりました。

For a complete list of the new log message project code prefixes used in JBoss EAP 7, see [Project Codes Used in JBoss EAP](#) in the JBoss EAP **Development Guide**.

4.4. WEB サーバー設定の変更

4.4.1. Undertow による Web サブシステムの置換

JBoss EAP 7 の web サーバーは、JBoss Web から Undertow に変更になりました。そのため、**web** サブシステム設定を新しい JBoss EAP 7 **undertow** サブシステム設定に移行する必要があります。

- サーバー設定ファイルの **urn:jboss:domain:web:2.2** サブシステム設定ネームスペースは **urn:jboss:domain:undertow:3.1** ネームスペースに置き換えられました。
- **EAP_HOME/modules/system/layers/base/** にあった **org.jboss.as.web** 拡張モジュールは、**org.wildfly.extension.undertow** 拡張モジュールに置き換えられました。

管理 CLI **migrate** 操作を使うと、**web** サブシステムをサーバー設定ファイル内の **undertow** に移行することができます。ただし、この操作では JBoss Web サブシステムのすべての設定が移行できるわけではないことに注意してください。"migration-warning" エントリーが表示される場合は、追加の管理 CLI コマンドを実行して設定を Undertow に移行する必要があります。管理 CLI **migrate** 操作についての詳細情報は、[管理 CLI の移行操作](#) を参照してください。

以下の例は、JBoss EAP 6 のデフォルトの **web** サブシステム設定を示しています。

```
<subsystem xmlns="urn:jboss:domain:web:2.2" default-virtual-server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
</subsystem>
```

以下の例は、JBoss EAP 7 のデフォルトの **undertow** サブシステム設定を示しています。

```
<subsystem xmlns="urn:jboss:domain:undertow:3.1">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http" redirect-socket="https"/>
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
    <websockets/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="{jboss.home.dir}/welcome-content"/>
  </handlers>
</subsystem>
```

```

    </handlers>
    <filters>
      <response-header name="server-header" header-name="Server" header-
value="JBoss-EAP/7"/>
      <response-header name="x-powered-by-header" header-name="X-
Powered-By" header-value="Undertow/1"/>
    </filters>
  </subsystem>

```

4.4.2. JBoss Web リライト条件の移行

管理 CLI の **migrate** 操作はリライト条件を自動的に移行できません。"migration-warnings" として報告され、手作業で移行する必要があります。Undertow の述語属性およびハンドラーを使用すると ([Undertow Predicates Attributes and Handlers](#) を参照)、JBoss EAP 7 で同等の設定を作成できます。

以下の例は、**rewrite** 設定を含む JBoss EAP 6 の **web** サブシステム設定を示しています。

```

<subsystem xmlns="urn:jboss:domain:web:2.2" default-virtual-
server="default" native="false">
  <virtual-server name="default" enable-welcome-root="true">
    <alias name="localhost"/>
    <rewrite name="test" pattern="(.*)/toberewritten/(.*)"
substitution="$1/rewritten/$2" flags="NC"/>
    <rewrite name="test2" pattern="(.*)" substitution="-" flags="F">
      <condition name="get" test="%{REQUEST_METHOD}"
pattern="GET"/>
      <condition name="andCond" test="%{REQUEST_URI}"
pattern=".*index.html" flags="NC"/>
    </rewrite>
  </virtual-server>
</subsystem>

```

管理 CLI の移行操作にある手順にしたがってサーバーと管理 CLI を開始し、以下のコマンドを使用して **web** サブシステム設定ファイルを移行します。

```
/subsystem=web:migrate
```

上記の設定で **migrate** 操作を実行すると、以下の "migration-warnings" がレポートされます。

```

/subsystem=web:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYWEB0002: Could not migrate resource {
\"pattern\" => \"(.*)\",
\"substitution\" => \"-\",
\"flags\" => \"F\",
\"operation\" => \"add\",
\"address\" => [
  (\"subsystem\" => \"web\"),
  (\"virtual-server\" => \"default-host\"),
  (\"rewrite\" => \"test2\")
]
}"],
}

```

```

        "WFLYWEB0002: Could not migrate resource {
\"test\" => \"%{REQUEST_METHOD}\",
\"pattern\" => \"GET\",
\"flags\" => undefined,
\"operation\" => \"add\",
\"address\" => [
    (\"subsystem\" => \"web\"),
    (\"virtual-server\" => \"default-host\"),
    (\"rewrite\" => \"test2\"),
    (\"condition\" => \"get\")
]
}],
        "WFLYWEB0002: Could not migrate resource {
\"test\" => \"%{REQUEST_URI}\",
\"pattern\" => \".*index.html\",
\"flags\" => \"NC\",
\"operation\" => \"add\",
\"address\" => [
    (\"subsystem\" => \"web\"),
    (\"virtual-server\" => \"default-host\"),
    (\"rewrite\" => \"test2\"),
    (\"condition\" => \"andCond\")
]
}
    ]}
}

```

サーバー設定ファイルを確認すると、**undertow** サブシステムが以下の設定になっています。



注記

リライト設定は削除されています。

```

<subsystem xmlns="urn:jboss:domain:undertow:3.1">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="http" socket-binding="http"/>
    <host name="default-host" alias="localhost, example.com">
      <location name="/" handler="welcome-content"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
  </handlers>
</subsystem>

```

管理 CLI を使用してフィルターを作成し、**undertow** サブシステムのリライト設定を置き換えます。各コマンドで "{"outcome" => "success"}" が表示されるはずですが。

```

# Create the filters
/subsystem=undertow/configuration=filter/expression-

```

```

filter="test1":add(expression="path('(.*)/toberewritten/(.*)') ->
rewrite('$1/rewritten/$2')")
/subsystem=undertow/configuration=filter/expression-
filter="test2":add(expression="method('GET') and path('.*index.html') ->
response-code(403)")

# Add the filters to the default server
/subsystem=undertow/server=default-server/host=default-host/filter-
ref="test1":add
/subsystem=undertow/server=default-server/host=default-host/filter-
ref="test2":add

```

更新されたサーバー設定ファイルを確認します。JBoss Web サブシステムは完全に移行され、**undertow** サブシステムに設定されます。

```

<subsystem xmlns="urn:jboss:domain:undertow:3.1">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="http" socket-binding="http"/>
    <host name="default-host" alias="localhost, example.com">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="test1"/>
      <filter-ref name="test2"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="${jboss.home.dir}/welcome-
content"/>
  </handlers>
  <filters>
    <expression-filter name="test1"
expression="path('(.*)/toberewritten/(.*)') ->
rewrite('$1/rewritten/$2')"/>
    <expression-filter name="test2" expression="method('GET') and
path('.*index.html') -> response-code(403)"/>
  </filters>
</subsystem>

```

For more information about how to configure filters and handlers using the management CLI, see [Configuring the Web Server](#) in the JBoss EAP 7 **Configuration Guide**.

4.4.3. Migrate JBoss Web System Properties

In the previous release of JBoss EAP, system properties could be used to modify the default JBoss Web behavior. For information about how to configure the same behavior in Undertow, see [JBoss Web System Properties Migration Reference](#)

4.4.4. Migrate JBoss Web `jboss-web.xml` Overlay

In JBoss EAP 6, JBoss Web supported the ability to specify additional static files for a web application using the **overlay** element in the `jboss-web.xml` file. This feature did not actually overlay existing files, but instead added static files to a deployment outside of the WAR archive.

In the following `jboss-web.xml` file example, `/tmp/mycontents/test.html` was returned by the JBoss EAP 6 server when you accessed the URL <http://localhost:8080/example/test.html>.

```
<jboss-web>
  <context-root>/example</context-root>
  <overlay>/tmp/mycontents/</overlay>
</jboss-web>
```

Undertow, which replaces JBoss Web in JBoss EAP 7, does not support the `jboss-web.xml` file `overlay` feature.

4.4.5. グローバルバルブの移行

以前のリリースの JBoss EAP ではバルブがサポートされました。バルブとは、リクエストの変更や追加処理を実行するために、サーブレットフィルターの前にアプリケーションのリクエスト処理パイプラインへ挿入されるカスタムクラスのことです。

- グローバルバルブは、デプロイされたすべてのアプリケーションのリクエスト処理パイプラインへ挿入され、サーバー設定ファイルで設定されます。
- オーセンティケーターバルブはリクエストのクレデンシャルを認証します。
- カスタムアプリケーションバルブは、`org.apache.catalina.valves.ValveBase` クラスを拡張して作成され、`jboss-web.xml` 記述子ファイルの `<valve>` 要素で設定されます。これらのバルブは手作業で移行する必要があります。

この項では、グローバルバルブの移行方法について説明します。カスタムおよびオーセンティケーターバルブの移行については、本ガイドの[カスタムアプリケーションバルブの移行](#)を参照してください。

JBoss EAP 7 で JBoss Web の代わりに導入された Undertow はグローバルバルブをサポートしませんが、Undertow ハンドラーを使用すると同様の機能を実現できます。Undertow には共通の機能を提供する複数のビルトインハンドラーが含まれています。また、カスタムハンドラーを作成する機能も含まれ、カスタムバルブの機能を置き換えるために使用することができます。

アプリケーションがバルブを使用する場合、JBoss EAP 7 へ移行するときに適切な Undertow ハンドラーコードに置き換え、同様の機能を実現する必要があります。

For more information about how to configure handlers, see [Configuring Handlers](#) in the JBoss EAP 7 **Configuration Guide**.

For more information about how to configure filters, see [Configuring Filters](#) in the JBoss EAP 7 **Configuration Guide**.

JBoss Web バルブの移行

以下の表では、JBoss EAP の前リリースで JBoss Web が提供していたバルブとそれに対応する Undertow ビルトインハンドラーを示しています。JBoss Web バルブは、`org.apache.catalina.valves` パッケージにあります。

表4.2 バルブとハンドラーのマッピング

バルブ	ハンドラー
AccessLogValve	io.undertow.server.handlers.accesslog.AccessLogHandler

バルブ	ハンドラー
CrawlerSessionManagerValve	io.undertow.servlet.handlers.CrawlerSessionManagerHandler
ExtendedAccessLogValve	io.undertow.server.handlers.accesslog.AccessLogHandler
JDBCAccessLogValve	手順については JDBCAccessLogValve の手動移行の手順 を参照してください。
RemoteAddrValve	io.undertow.server.handlers.IPAddressAccessControlHandler
RemoteHostValve	io.undertow.server.handlers.AccessControlListHandler
RemoteIpValve	io.undertow.server.handlers.ProxyPeerAddressHandler
RequestDumperValve	io.undertow.server.handlers.RequestDumpingHandler
RewriteValve	これらのバルブを手作業で移行する手順は、 JBoss Web リライト条件の移行 を参照してください。
StuckThreadDetectionValve	io.undertow.server.handlers.StuckThreadDetectionHandler

管理 CLI の **migrate** 操作を使用すると、以下の基準を満たすグローバルバルブを自動的に移行できます。

- 前述の表にリストされている、手動処理の必要がないバルブに限定されます。
- サーバー設定ファイルの **web** サブシステムに定義されている必要があります。

管理 CLI の **migrate** 操作の詳細は、[管理 CLI の移行操作](#) を参照してください。

JDBCAccessLogValve の手動移行の手順

org.apache.catalina.valves.JDBCAccessLogValve バルブはルールの例外で、**io.undertow.server.handlers.JDBCLogHandler** へ自動的に移行することができません。以下の手順に従って、次のバルブ例を移行します。

```
<valve name="jdbc" module="org.jboss.as.web" class-
name="org.apache.catalina.valves.JDBCAccessLogValve">
  <param param-name="driverName" param-value="com.mysql.jdbc.Driver" />
  <param param-name="connectionName" param-value="root" />
  <param param-name="connectionPassword" param-value="password" />
  <param param-name="connectionURL" param-
value="jdbc:mysql://localhost:3306/wildfly?
zeroDateTimeBehavior=convertToNull" />
  <param param-name="format" param-value="combined" />
</valve>
```

1. ログエントリを保存するデータベースのドライバーモジュールを作成します。
2. データベースのデータソースを設定し、ドライバーを **datasources** サブシステムの利用可能なドライバーのリストに追加します。

```

<datasources>
  <datasource jndi-name="java:jboss/datasources/accessLogDS" pool-
name="accessLogDS" enabled="true" use-java-context="true">
    <connection-url>jdbc:mysql://localhost:3306/wildfly?
zeroDateTimeBehavior=convertToNull</connection-url>
    <driver>mysql</driver>
    <security>
      <user-name>root</user-name>
      <password>Password1!</password>
    </security>
  </datasource>
  ...
<drivers>
  <driver name="mysql" module="com.mysql">
    <driver-class>com.mysql.jdbc.Driver</driver-class>
  </driver>
  ...
</drivers>
</datasources>

```

3. 式 `jdbc-access-log(datasource=DATASOURCE_JNDI_NAME)` を使用して、`undertow` サブシステムの `expression-filter` を設定します。

```

<filters>
  <expression-filter name="jdbc-access" expression="jdbc-access-
log(datasource='java:jboss/datasources/accessLogDS')" />
  ...
</filters>

```

4.5. JGROUPS サーバー設定の変更

4.5.1. JGroups はデフォルトでプライベートネットワークインターフェースを使用

JBoss EAP 6 のデフォルト設定では、JGroups はサーバー設定ファイルの `<interfaces>` セクションに定義された `public` インターフェースを使用しました。

専用のネットワークインターフェースを使用することが推奨されるため、JBoss EAP 7 では JGroups はサーバー設定ファイルの `<interfaces>` セクションに定義された新しい `private` インターフェースを使用するようになりました。

4.5.2. JGroups チャンネルの変更

JGroups は JGroups チャンネルで HA サービスのグループ通信サポートを提供します。JBoss EAP 7 では、サーバー設定ファイルの `jgroups` サブシステムに `<channel>` 要素が導入されました。管理 CLI を使用して JGroups チャンネル設定を追加、削除、および変更できます。

For more information about how to configure JGroups, see [Cluster Communication with JGroups](#) in the JBoss EAP Configuration Guide.

4.6. INFINISPAN サーバー設定の変更

4.6.1. Infinispan のデフォルトキャッシュ設定の変更

In JBoss EAP 6, the default clustered caches for web session replication and EJB replication were replicated **ASYNC** caches. This has changed in JBoss EAP 7. The default clustered caches are now distributed **ASYNC** caches. The replicated caches are no longer even configured by default. See [Configure the Cache Mode](#) in the JBoss EAP **Configuration Guide** for information about how to add a replicated cache and make it the default.

これは、新しい JBoss EAP 7 のデフォルト設定を使用する場合のみ影響します。JBoss EAP 6 から設定を移行する場合は、**infinispan** サブシステムの設定は保持されます。

4.6.2. Infinispan のキャッシュストラテジーの変更

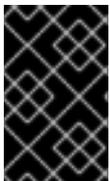
ASYNC キャッシュストラテジーの動作が JBoss EAP 7 では変更になりました。

JBoss EAP 6 では、**ASYNC** キャッシュの読み取りにロックはありませんでした。ブロックは発生しませんでした。フェイルオーバーなどで陳腐データのダーティリードが発生する傾向にありました。これは、リクエストの完了前に同じユーザーの次のリクエストを開始できたためです。これは、クラスタートポロジーの変更がセッションアフィニティーに影響し、容易にデータが陳腐化することがあるため、分散モードを使用するときは許可されません。

JBoss EAP 7 では、**ASYNC** キャッシュの読み取りにロックが必要になりました。レプリケーションが終了するまで同じユーザーの新しいリクエストをブロックするようになったため、ダーティリードが発生しないようになりました。

4.7. EJB SERVER CONFIGURATION CHANGES

If you use the management CLI **migrate** operation to upgrade your existing configuration, you might see exceptions in the server log when you deploy your EJB applications.



重要

If you use the JBoss Server Migration Tool to update your server configuration, the **ejb3** subsystem should be configured correctly and you should not see any issues when you deploy your EJB applications.

DuplicateServiceException

The following **DuplicateServiceException** is caused by caching changes in JBoss EAP 7.

DuplicateServiceException in Server Log

```
ERROR [org.jboss.msc.service.fail] (MSC service thread 1-3) MSC000001:
Failed to start service jboss.deployment.unit."mdb-1.0-
SNAPSHOT.jar".cache-dependencies-installer:
org.jboss.msc.service.StartException in service
jboss.deployment.unit."mdb-1.0-SNAPSHOT.jar".cache-dependencies-installer:
Failed to start service
...
Caused by: org.jboss.msc.service.DuplicateServiceException: Service
jboss.infinispan.ejb."mdb-1.0-SNAPSHOT.jar".config is already registered
```

You must reconfigure the cache to resolve this error.

1. Follow the instructions to [Start the Server and the Management CLI](#)
2. Issue the following commands to reconfigure caching in the **ejb3** subsystem.

```

/subsystem=ejb3/file-passivation-store=file:remove
/subsystem=ejb3/cluster-passivation-store=infinispan:remove
/subsystem=ejb3/passivation-store=infinispan:add(cache-
container=ejb, max-size=10000)

/subsystem=ejb3/cache=passivating:remove
/subsystem=ejb3/cache=clustered:remove
/subsystem=ejb3/cache=distributable:add(passivation-
store=infinispan, aliases=[passivating, clustered])

```

4.8. メッセージングサーバー設定の変更

JBoss EAP 7 では、JMS サポートプロバイダーが HornetQ から ActiveMQ Artemis に変更になりました。ここでは、設定と関連するメッセージングデータの移行方法を説明します。

4.8.1. メッセージングサブシステムサーバー設定の変更

EAP_HOME/modules/system/layers/base/ にあった **org.jboss.as.messaging** モジュール拡張は、**org.wildfly.extension.messaging-activemq** 拡張モジュールに置き換えられました。

urn:jboss:domain:messaging:3.0 サブシステム設定ネームスペースは
urn:jboss:domain:messaging-activemq:1.0 ネームスペースに置き換えられました。

管理モデル

ほとんどの場合で、要素名と属性名は以前のリリースとできる限り同じになるよう努力しています。以下の表は変更の一部を表しています。

表4.3 メッセージング属性のマッピング

HornetQ での名前	ActiveMQ での名前
hornetq-server	server
hornetq-serverType	serverType
connectors	connector
discovery-group-name	discovery-group

新しい **messaging-activemq** サブシステム上で呼び出される管理操作は、**/subsystem=messaging/hornetq-server=** から **/subsystem=messaging-activemq/server=** に変更になりました。

migrate 操作を呼び出すと、既存の JBoss EAP 6 の **messaging** サブシステム設定を JBoss EAP 7 の **messaging-activemq** サブシステムに移行できます。

```

/subsystem=messaging:migrate

```

Before you execute the **migrate** operation, you can invoke the **describe-migration** operation to review the list of management operations that will be performed to migrate from the existing JBoss EAP 6 **messaging** subsystem configuration to the **messaging-activemq** subsystem on the JBoss EAP 7

server.

```
/subsystem=messaging:describe-migration
```

migrate および **describe-migration** 操作は、自動的に移行されないリソースや属性の **migration-warnings** のリストも表示します。

Messaging サブシステムの移行および前方互換性

messaging サブシステムの **describe-migration** および **migrate** 操作は、追加の設定引数を提供します。メッセージングを設定してレガシー JBoss EAP 6 クライアントが JBoss EAP 7 サーバーに接続できるようにするには、以下のようにブール値の **add-legacy-entries** 引数を **describe-migration** または **migrate** 操作に追加します。

```
/subsystem=messaging:describe-migration(add-legacy-entries=true)
/subsystem=messaging:migrate(add-legacy-entries=true)
```

ブール値の引数 **add-legacy-entries** が **true** に設定されていると、**messaging-activemq** サブシステムが **legacy-connection-factory** リソースを作成し、**legacy-entries** を **jms-queue** および **jms-topic** リソースに追加します。

ブール値の引数 **add-legacy-entries** が **false** に設定されていると、**messaging-activemq** サブシステムにはレガシーリソースが作成されず、レガシー JMS クライアントは JBoss EAP 7 サーバーと通信できません。これがデフォルト値になります。

For more information about forward and backward compatibility see the [Backward and Forward Compatibility](#) in **Configuring Messaging** for JBoss EAP.

管理 CLI の **migrate** および **describe-migration** 操作の詳細は、[管理 CLI の移行操作](#) を参照してください。

Change in Behavior of forward-when-no-consumers Attribute

The behavior of the **forward-when-no-consumers** attribute has changed in JBoss EAP 7.

In JBoss EAP 6, when **forward-when-no-consumers** was set to **false** and there were no consumers in a cluster, messages were redistributed to all nodes in a cluster.

This behavior has changed in JBoss EAP 7. When **forward-when-no-consumers** is set to **false** and there are no consumers in a cluster, messages are not redistributed. Instead, they are kept on the original node to which they were sent.

Messaging サブシステムの XML 設定

新しい **messaging-activemq** サブシステムによって、他の JBoss EAP サブシステムとより一貫性のある XML スキームが提供されるようになり、XML の設定は大幅に変更されました。

It is strongly advised that you do not attempt to modify the JBoss EAP **messaging** subsystem XML configuration to conform to the new **messaging-activemq** subsystem. Instead, invoke the legacy subsystem **migrate** operation. This operation will write the XML configuration of the new **messaging-activemq** subsystem as a part of its execution.

4.8.2. メッセージングデータの移行

JMS サポートプロバイダーが HornetQ から ActiveMQ Artemis に変更になったため、JBoss EAP 7 ではメッセージングデータの形式と場所が変更になりました。

以下の 2 つの方法のいずれかを使用してデータを移行できます。

- 管理 CLI の **import-journal** 操作を使用すると、以前のリリースから HornetQ のメッセージングデータをエクスポートし、インポートすることができます。詳細は、[エクスポートおよびインポートを使用したメッセージングデータの移行](#)を参照してください。
- JMS ブリッジを設定すると、以前のリリースからデータを移行できます。この手順は、[JMS ブリッジを使用したメッセージングデータの移行](#)に記載されています。

リリース間で行われたメッセージングデータフォルダー名の変更についての詳細は、[メッセージングフォルダー名のマッピング](#)を参照してください。

4.8.2.1. エクスポートおよびインポートを使用したメッセージングデータの移行

この方法では、HornetQ の **exporter** ユーティリティを使用して以前のリリースからデータをエクスポートします。その後、**import-journal** 操作を使用して XML 形式の出力ファイルをインポートします。



警告

現在、この方法には既知の問題が存在します。この問題の最新状況を確認するには、[JBPEAP-4407 - Consumer crashes with IndexOutOfBoundsException when reading large messages from imported journal](#) (インポートされたジャーナルから大型のメッセージを読み取ると、IndexOutOfBoundsException によってコンシューマーがクラッシュする) を参照してください。

前リリースからのメッセージングデータのエクスポート



重要

メッセージングデータをエクスポートする前に、JBoss EAP 6 サーバーを停止する必要があります。

HornetQ **exporter** ユーティリティは、メッセージングデータを生成し、JBoss EAP 6 から XML 形式のファイルへエクスポートします。このコマンドでは、JBoss EAP 6 に同梱されている必須の HornetQ JAR へのパスを指定し、以前のリリースからの **messagingbindings/**、**messagingjournal/**、**messagingpaging/**、および **messaginglargemessages/** フォルダーへのパスを引数として渡し、エクスポートされる XML データを書き込む出力ファイルを指定する必要があります。

以下は HornetQ **exporter** ユーティリティで必要となる構文です。

```
java -jar -mp MODULE_PATH org.hornetq.exporter MESSAGING_BINDINGS_DIRECTORY
MESSAGING_JOURNAL_DIRECTORY MESSAGING_PAGING_DIRECTORY
MESSAGING_LARGE_MESSAGES_DIRECTORY > OUTPUT_DATA.xml
```

カスタムモジュールを作成し、パッチやアップグレードでインストールされた JAR を含む HornetQ JAR の正しいバージョンがロードされ、**exporter** ユーティリティで利用可能になるようにします。好みのエディターを使用して、**EAP6_HOME/modules/org/hornetq/exporter/main/** ディレクト

リー内に `module.xml` ファイルを作成し、以下のコンテンツをコピーします。

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.hornetq.exporter">
  <main-class
name="org.hornetq.jms.persistence.impl.journal.XmlDataExporter"/>
  <properties>
    <property name="jboss.api" value="deprecated"/>
  </properties>
  <dependencies>
    <module name="org.hornetq"/>
  </dependencies>
</module>
```



注記

カスタムモジュールは `modules/system/layers/base/` ディレクトリーではなく、`modules/` ディレクトリー内に作成します。

以下の手順に従い、データをエクスポートします。

1. JBoss EAP 6 サーバーを停止します。
2. 上記の説明にあるようにカスタムモジュールを作成します。
3. 以下のコマンドを実行してデータをエクスポートします。

```
$ java -jar jboss-modules.jar -mp modules/ org.hornetq.exporter
standalone/data/messagingbindings/ standalone/data/messagingjournal/
standalone/data/messagingpaging
standalone/data/messaginglargemessages/ >
OUTPUT_DIRECTORY/OldMessagingData.xml
```

4. コマンド完了後に、ログにエラーや警告メッセージがないことを確認します。
5. 使用中のオペレーティングシステムで利用可能なツールを使用して、生成された出力ファイルの XML を検証します。

XML 形式のメッセージングデータのインポート

その後は、以下のように `import-journal` 操作で XML 形式の出力ファイルをインポートします。

```
/subsystem=messaging-activemq/server=default:import-
journal(file=OUTPUT_DIRECTORY/OldMessagingData.xml)
```

4.8.2.2. JMS ブリッジを使用したメッセージングデータの移行

この方法では、JBoss EAP 6 の HornetQ キューから JBoss EAP 7 の ActiveMQ Artemis キューへメッセージを移行する JMS ブリッジを JBoss EAP 7 サーバーに設定およびデプロイします。

JMS ブリッジはソースの JMS キューまたはトピックからメッセージを消費し、通常は異なるサーバーにあるターゲットの JMS キューまたはトピックへ送信します。JMS 1.1 に準拠する JMS サーバーの間でメッセージをブリッジするために使用できます。送信元および宛先の JMS リソースは、JNDI を使用してロックアップされ、JNDI ロックアップのクライアントクラスはモジュールでバンドルされる必要があります。モジュール名は JMS ブリッジ設定で宣言されます。

ここでは、メッセージングデータを JBoss EAP 6 から JBoss EAP 7 に移動するためにサーバーを設定し、JMS ブリッジをデプロイする方法を説明します。

JBoss EAP 6 サーバーの設定

1. JBoss EAP 6 サーバーを停止します。
2. HornetQ のジャーナルおよび設定ファイルをバックアップします。
 - デフォルトでは、HornetQ ジャーナルは **EAP6_HOME/standalone/data/** ディレクトリーにあります。
 - 各リリースにおけるメッセージングフォルダーのデフォルトの場所は、[メッセージングフォルダー名のマッピング](#)を参照してください。
3. JMS メッセージが含まれる **InQueue** JMS キューが JBoss EAP 6 サーバーで定義されていることを確認してください。
4. **messaging** サブシステムの設定に以下と似た **RemoteConnectionFactory** のエントリーが含まれていることを確認してください。

```
<connection-factory name="RemoteConnectionFactory">
  <entries>
    <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
  </entries>
  ...
</connection-factory>
```

エントリーが含まれていない場合は、以下の管理 CLI コマンドを使用して作成します。

```
/subsystem=messaging/hornetq-server=default/connection-
factory=RemoteConnectionFactory:add(factory-type=XA_GENERIC,
connector=[netty], entries=
[java:jboss/exported/jms/RemoteConnectionFactory],ha=true,block-on-
acknowledge=true,retry-interval=1000,retry-interval-
multiplier=1.0,reconnect-attempts=-1)
```

JBoss EAP 7 サーバーの設定

1. 以前のリリースでは、JMS ブリッジ設定が HornetQ サーバーに接続するには、**org.hornetq** モジュールが必要です。このモジュールと直接の依存関係は JBoss EAP 7 では存在しないため、以前のリリースから以下のモジュールをコピーする必要があります。
 - **org.hornetq** モジュールを JBoss EAP 7 **EAP_HOME/modules/org/** ディレクトリーにコピーします。
 - If you did not apply patches to this module, copy this folder from the JBoss EAP 6.4 server: **EAP6_HOME/modules/system/layers/base/org/hornetq/**
 - If you did apply patches to this module, copy this folder from the JBoss EAP 6.4 server: **EAP6_HOME/modules/system/layers/base/.overlays/layer-base-jboss-eap-6.4.x.CP/org/hornetq/**
 - Remove the **<resource-root>** for the HornetQ **lib** path from the JBoss EAP 7 **EAP_HOME/modules/org/hornetq/main/module.xml** file.

- If you did not apply patches to the JBoss EAP 6.4 **org.hornetq** module, remove the following line from the file:

```
<resource-root path="lib"/>
```

- If you did apply patches to the JBoss EAP 6.4 **org.hornetq** module, remove the following lines from the file:

```
<resource-root path="lib"/>
<resource-root path="../../../../../../org/hornetq/main/lib"/>
```



警告

Failure to remove the HornetQ **lib** path **resource-root** will cause the bridge to fail with the following error in the log file.

```
2016-07-15 09:32:25,660 ERROR
[org.jboss.as.controller.management-operation]
(management-handler-thread - 2) WFLYCTL0013:
Operation ("add") failed - address: ([
  ("subsystem" => "messaging-activemq"),
  ("jms-bridge" => "myBridge")
]) - failure description: "WFLYMSGAMQ0086:
Unable to load module org.hornetq"
```

- Copy the **org.jboss.netty** module into the JBoss EAP 7 **EAP_HOME/modules/org/jboss/** directory.
 - If you did not apply patches to this module, copy this folder from the JBoss EAP 6.4 server: **EAP6_HOME/modules/system/layers/base/org/jboss/netty/**
 - If you did apply patches to this module, copy this folder from the JBoss EAP 6.4 server: **EAP6_HOME/modules/system/layers/base/.overlays/layer-base-jboss-eap-6.4.x.CP/org/jboss/netty**
2. JBoss EAP 6 サーバーから受信したメッセージを格納するために JSM キューを作成します。以下は、**MigratedMessagesQueue** JMS キューを作成してメッセージを受信する管理 CLI コマンドの例になります。

```
jms-queue add --queue-address=MigratedMessagesQueue --entries=
[jms/queue/MigratedMessagesQueue
java:jboss/exported/jms/queue/MigratedMessagesQueue]
```

このコマンドにより、JBoss EAP 7 サーバーの **messaging-activemq** サブシステムに以下のデフォルトサーバー用の **jms-queue** 設定が作成されます。

```
<jms-queue name="MigratedMessagesQueue"
entries="jms/queue/MigratedMessagesQueue
java:jboss/exported/jms/queue/MigratedMessagesQueue"/>
```

3. **messaging-activemq** サブシステムの **default** サーバーに以下と似た **InVmConnectionFactory connection-factory** の設定が含まれるようにしてください。

```
<connection-factory name="InVmConnectionFactory" factory-
type="XA_GENERIC" entries="java:/ConnectionFactory" connectors="in-
vm"/>
```

エントリーが含まれていない場合は、以下の管理 CLI コマンドを使用して作成します。

```
/subsystem=messaging-activemq/server=default/connection-
factory=InVmConnectionFactory:add(factory-type=XA_GENERIC,
connectors=[in-vm], entries=[java:/ConnectionFactory])
```

4. JBoss EAP 6 サーバーで設定された **InQueue** JMS キューからメッセージを読み取る JMS ブリッジを作成およびデプロイし、JBoss EAP 7 サーバーで設定された **MigratedMessagesQueue** に転送します。

```
/subsystem=messaging-activemq/jms-bridge=myBridge:add(add-messageID-
in-header=true,max-batch-time=100,max-batch-size=10,max-retries=-
1,failure-retry-interval=1000,quality-of-
service=AT_MOST_ONCE,module=org.hornetq,source-
destination=jms/queue/InQueue,source-connection-
factory=jms/RemoteConnectionFactory,source-context=
[("java.naming.factory.initial"=>"org.jboss.naming.remote.client.Ini-
tialContextFactory"),
("java.naming.provider.url"=>"remote://127.0.0.1:4447")],target-
destination=jms/queue/MigratedMessagesQueue,target-connection-
factory=java:/ConnectionFactory)
```

これにより、JBoss EAP 7 サーバーの **messaging-activemq** サブシステムに以下の **jms-bridge** 設定が作成されます。

```
<jms-bridge name="myBridge" add-messageID-in-header="true" max-
batch-time="100" max-batch-size="10" max-retries="-1" failure-retry-
interval="1000" quality-of-service="AT_MOST_ONCE"
module="org.hornetq">
  <source destination="jms/queue/InQueue" connection-
factory="jms/RemoteConnectionFactory">
    <source-context>
      <property name="java.naming.factory.initial"
value="org.jboss.naming.remote.client.InitialContextFactory"/>
      <property name="java.naming.provider.url"
value="remote://127.0.0.1:4447"/>
    </source-context>
  </source>
  <target destination="jms/queue/MigratedMessagesQueue"
connection-factory="java:/ConnectionFactory"/>
</jms-bridge>
```

5. また、セキュリティーが JBoss EAP 6 に設定されている場合、接続の作成時に JNDI ルックアップで使用される正しいユーザー名およびパスワードを指定する **source-context** が含まれるよう、JMS ブリッジ設定 **<source>** 要素を設定する必要があります。

データの移行

1. 以下の設定に提供する情報が正しいことを確認します。
 - キューおよびトピック名
 - JUDI ルックアップの `java.naming.provider.url`
2. ターゲットの JMS 宛先が JBoss EAP 7 サーバーにデプロイされているようにしてください。
3. JBoss EAP 6 サーバーと JBoss EAP 7 サーバーを両方起動します。

4.8.2.3. メッセージングフォルダー名のマッピング

以下の表では、JBoss EAP の以前および現在のリリースで対応するメッセージングディレクトリーの名前を示しています。ディレクトリーは `jboss.server.data.dir` ディレクトリーに相対的で、指定がない場合はデフォルトで `EAP_HOME/standalone/data/` になります。

JBoss EAP 6 の ディレクトリー名	JBoss EAP 7 のディレクトリー名
<code>messagingbindings/</code>	<code>activemq/bindings/</code>
<code>messagingjournal/</code>	<code>activemq/journal/</code>
<code>messaginglargemessages/</code>	<code>activemq/largemessages/</code>
<code>messagingpaging/</code>	<code>activemq/paging/</code>



注記

The `messaginglargemessages/` and `messagingpaging/` directories might not be present if there are no large messages or if paging is disabled.

4.8.3. JMS 宛先の移行

JBoss EAP のこれまでのリリースでは、JMS 宛先キューは `messaging` サブシステムの `<hornetq-server>` 要素下にある `<jms-destinations>` 要素に設定されました。

```
<hornetq-server>
...
<jms-destinations>
  <jms-queue name="testQueue">
    <entry name="queue/test"/>
    <entry name="java:jboss/exported/jms/queue/test"/>
  </jms-queue>
</jms-destinations>
...
</hornetq-server>
```

JBoss EAP 7 では、JMS 宛先キューは **messaging-activemq** サブシステムのデフォルトの **<server>** 要素に設定されます。

```
<server name="default">
  ...
  <jms-queue name="testQueue" entries="queue/test
java:jboss/exported/jms/queue/test"/>
  ...
</server>
```

4.8.4. メッセージングインターセプターの移行

JBoss EAP 7 では、JMS メッセージングプロバイダーが HornetQ から ActiveMQ Artemis に変更されたため、メッセージングインターセプターが大幅に変更されました。

以前の JBoss EAP リリースに含まれていた HornetQ **messaging** サブシステムでは、HornetQ インターセプターを JAR に追加し、HornetQ **module.xml** ファイルを修正するというインストール方法が必要でした。

The **messaging-activemq** subsystem included in JBoss EAP 7 does not require modification of a **module.xml** file. User interceptor classes, which now implement the Apache ActiveMQ Artemis [Interceptor](#) interface, can now be loaded from any server module. You specify the module from which the interceptor should be loaded in the **messaging-activemq** subsystem of the server configuration file.

インターセプター設定の例

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">
  <server name="default">
    ...
    <incoming-interceptors>
      <class name="com.mycompany.incoming.myInterceptor"
module="com.mycompany" />
      <class name="com.othercompany.incoming.myOtherInterceptor"
module="com.othercompany" />
    </incoming-interceptors>
    <outgoing-interceptors>
      <class name="com.mycompany.outgoing.myInterceptor"
module="com.mycompany" />
      <class name="com.othercompany.outgoing.myOtherInterceptor"
module="com.othercompany" />
    </outgoing-interceptors>
  </server>
</subsystem>
```

4.8.5. Netty サーブレット設定の変更

JBoss EAP 6 では、Netty サーブレットトランスポートと動作するようサーブレットエンジンを設定することができました。JBoss EAP 7 では、ビルトインメッセージングプロバイダーが HornetQ から ActiveMQ Artemis に変更になったため、この設定は使用できなくなりました。新しいビルトインメッセージング HTTP コネクターおよび HTTP アクセプターを使用するよう、サーブレット設定を変更する必要があります。

4.9. JMX MANAGEMENT CHANGES

The HornetQ component in JBoss EAP 6 provided its own JMX management; however, it was not recommended and is now deprecated and no longer supported. If you relied on this feature in JBoss EAP 6, you must migrate your management tooling to use either the JBoss EAP management CLI or the JMX management provided with JBoss EAP 7.

You must also upgrade your client libraries to use the `jboss-client.jar` that ships with JBoss EAP 7.

The following is an example of HornetQ JMX management code that was used in JBoss EAP 6.

```
JMXConnector connector = null;
try {
    HashMap environment = new HashMap();
    String[] credentials = new String[]{"admin", "Password123!"};
    environment.put(JMXConnector.CREDENTIALS, credentials);

    // HornetQ used the protocol "remoting-jmx" and port "9999"
    JMXServiceURL beanServerUrl = new JMXServiceURL("service:jmx:remoting-
jmx://127.0.0.1:9999");

    connector = JMXConnectorFactory.connect(beanServerUrl, environment);
    MBeanServerConnection mbeanServer =
connector.getMBeanServerConnection();

    // The JMX object name pointed to the HornetQ JMX management
    ObjectName objectName = new
ObjectName("org.hornetq:type=Server,module=JMS");

    // The invoked method name was "listConnectionIDs"
    String[] connections = (String[]) mbeanServer.invoke(objectName,
"listConnectionIDs", new Object[] {}, new String[] {});
    for (String connection : connections) {
        System.out.println(connection);
    }
} finally {
    if (connector != null) {
        connector.close();
    }
}
```

The following is an example of the equivalent code needed for ActiveMQ Artemis in JBoss EAP 7.

```
JMXConnector connector = null;
try {
    HashMap environment = new HashMap();
    String[] credentials = new String[]{"admin", "Password123!"};
    environment.put(JMXConnector.CREDENTIALS, credentials);

    // ActiveMQ Artemis uses the protocol "remote+http" and port "9990"
    JMXServiceURL beanServerUrl = new
JMXServiceURL("service:jmx:remote+http://127.0.0.1:9990");

    connector = JMXConnectorFactory.connect(beanServerUrl, environment);
    MBeanServerConnection mbeanServer =
connector.getMBeanServerConnection();
```

```

    // The JMX object name points to the new JMX management in the
    `messaging-activemq` subsystem
    ObjectName objectName = new ObjectName("jboss.as:subsystem=messaging-
activemq,server=default");

    // The invoked method name is now "listConnectionIds"
    String[] connections = (String[]) mbeanServer.invoke(objectName,
"listConnectionIds", new Object[] {}, new String[] {});
    for (String connection : connections) {
        System.out.println(connection);
    }
} finally {
    if (connector != null) {
        connector.close();
    }
}
}

```

Notice that the method names and parameters have changed in the new implementation. You can find the new method names in the JConsole by following these steps.

1. Connect to the JConsole using the following command.

```
EAP_HOME/bin/jconsole.sh
```

2. Connect to JBoss EAP local process. Note that it should start with "jboss-modules.jar".
3. In the **MBeans** tab, choose **jboss.as** → **messaging-activemq** → **default** → **Operations** to display the list of method names and attributes.

4.10. ORB サーバー設定の変更

JBoss EAP 7 では、JacORB 実装が OpenJDK ORB のダウンストリームブランチに変更になりました。

`EAP_HOME/modules/system/layers/base/` にあった `org.jboss.as.jacorb` 拡張モジュールは、`org.wildfly.iiop-openjdk` 拡張モジュールに置き換えられました。

サーバー設定ファイルの `urn:jboss:domain:jacorb:1.4` サブシステム設定ネームスペースは `urn:jboss:domain:iiop-openjdk:1.0` ネームスペースに置き換えられました。

以下の例は、JBoss EAP 6 のデフォルトの `jacorb` システム設定を示しています。

```

<subsystem xmlns="urn:jboss:domain:jacorb:1.4">
  <orb socket-binding="jacorb" ssl-socket-binding="jacorb-ssl">
    <initializers security="identity" transactions="spec"/>
  </orb>
</subsystem>

```

以下の例は、JBoss EAP 7 のデフォルトの `iiop-openjdk` サブシステム設定を示しています。

```

<subsystem xmlns="urn:jboss:domain:iiop-openjdk:1.0">
  <orb socket-binding="jacorb" ssl-socket-binding="jacorb-ssl" />
  <initializers security="identity" transactions="spec" />
</subsystem>

```

新しい **iiop-openjdk** サブシステム設定は、レガシー要素および属性のサブセットのみを受け入れません。以下は、有効な要素および属性がすべて含まれる、前リリースの JBoss EAP の **jacorb** サブシステム設定例になります。

```
<subsystem xmlns="urn:jboss:domain:jacorb:1.4">
  <orb name="JBoss" print-version="off" use-imr="off" use-bom="off"
cache-typecodes="off"
  cache-poa-names="off" giop-minor-version="2" socket-
binding="jacorb" ssl-socket-binding="jacorb-ssl">
    <connection retries="5" retry-interval="500" client-timeout="0"
server-timeout="0"
      max-server-connections="500" max-managed-buf-size="24" outbuf-
size="2048"
        outbuf-cache-timeout="-1"/>
    <initializers security="off" transactions="spec"/>
  </orb>
  <poa monitoring="off" queue-wait="on" queue-min="10" queue-max="100">
    <request-processors pool-size="10" max-threads="32"/>
  </poa>
  <naming root-context="JBoss/Naming/root" export-corbaloc="on"/>
  <interop sun="on" comet="off" iona="off" chunk-custom-rmi-
valuetypes="on"
    lax-boolean-encoding="off" indirection-encoding-disable="off"
strict-check-on-tc-creation="off"/>
  <security support-ssl="off" add-component-via-interceptor="on" client-
supports="MutualAuth"
    client-requires="None" server-supports="MutualAuth" server-
requires="None"/>
  <properties>
    <property name="some_property" value="some_value"/>
  </properties>
</subsystem>
```

以下の要素属性はサポート対象外になったため、削除する必要があります。

表4.4 削除する属性

要素	サポートされない属性
----	------------

要素	サポートされない属性
<orb>	<ul style="list-style-type: none"> ● client-timeout ● max-managed-buf-size ● max-server-connections ● outbuf-cache-timeout ● outbuf-size ● connection retries ● retry-interval ● name ● server-timeout
<poa>	<ul style="list-style-type: none"> ● queue-min ● queue-max ● pool-size ● max-threads

以下の **on/off** 属性はサポート対象外となり、管理 CLI の **migrate** 操作を実行しても移行されません。これらの属性が **on** に設定されていると移行の警告が表示されます。**<security support-ssl="on|off">** などの、この表に記載されていない **on/off** 属性のサポートは継続され、正常に移行されます。唯一の違いは、値が **on/off** から **true/false** に変更されることです。

表4.5 off に設定するまたは削除する属性

要素	off に設定する属性
<orb>	<ul style="list-style-type: none"> ● cache-poa-names ● cache-typecodes ● print-version ● use-bom ● use-imr

要素	off に設定する属性
<interop>	(sun 以外すべて) <ul style="list-style-type: none"> • comet • iona • chunk-custom-rmi-valuetypes • indirection-encoding-disable • lax-boolean-encoding • strict-check-on-tc-creation
<poa/>	<ul style="list-style-type: none"> • monitoring • queue-wait

4.11. THREADS サブシステム設定の移行

JBoss EAP 6 のサーバー設定には、サーバーの異なるサブシステムにまたがってスレッドプールを管理するために使用される **threads** が含まれていました。

threads サブシステムは JBoss EAP 7 では利用できないようになりました。この代わりに、各サブシステムが独自のスレッドプールを管理します。

For information about how to configure thread pools for the **infinispan** subsystem, see [Configure Infinispan Thread Pools](#) in the JBoss EAP **Configuration Guide**.

For information about how to configure thread pools for the **jgroups** subsystem, see [Configure JGroups Thread Pools](#) in the JBoss EAP **Configuration Guide**.

In JBoss EAP 6, you configured thread pools for connectors and listeners for the **web** subsystem by referencing an **executor** that was defined in the **threads** subsystem. In JBoss EAP 7, you now configure thread pools for the **undertow** subsystem by referencing a **worker** that is defined in the **io** subsystem. For more information, see [Configuring the IO Subsystem](#) in the JBoss EAP **Configuration Guide**.

For information about about changes to thread pool configuration in the **remoting** subsystem, see [Migrate the Remoting Subsystem Configuration](#) in this guide, and [Configuring the Endpoint](#) in the JBoss EAP **Configuration Guide**.

4.12. REMOTING サブシステム設定の移行

JBoss EAP 6 では、複数の **worker-*** 属性を設定して **remoting** サブシステムのスレッドプールを設定しました。JBoss EAP 7 では **remoting** サブシステムでワーカースレッドプールを設定しないようになりました。既存の設定を変更しようとすると、以下のメッセージが表示されます。

```
WFLYRMT0022: Worker configuration is no longer used, please use endpoint worker configuration
```

JBoss EAP 7 では、ワーカーレッドプールは、**io** サブシステムで定義される **worker** を参照するエンドポイント設定に置き換えられました。

For information about how to configure the endpoint, see [Configuring the Endpoint](#) in the JBoss EAP Configuration Guide.

4.13. WEBSOCKET サーバー設定の変更

JBoss EAP 6 で WebSockets を使用するには、以下と同様のコマンドを使用して JBoss EAP サーバー設定ファイルの **web** サブシステムにある **http** コネクタに対して非ブロッキング Java NIO2 コネクタを有効にする必要がありました。

```
/subsystem=web/connector=http/:write-attribute(name=protocol,value=org.apache.coyote.http11.Http11NioProtocol)
```

アプリケーションで WebSockets を使用するには、アプリケーションの **WEB-INF/jboss-web.xml** ファイル内で **<enable-websockets>** 要素を作成し、これを **true** に設定する必要もありました。

JBoss EAP 7 では、デフォルトの WebSocket サポートに対してサーバーを設定したり、アプリケーションがそれを使用するように設定したりする必要がなくなりました。WebSocket は Java EE 7 では必須で、必要なプロトコルはデフォルトで設定されています。さらに複雑な WebSocket の設定は、JBoss EAP サーバー設定ファイルの **undertow** サブシステムにある **servlet-container** で行います。以下のコマンドを実行すると、使用できる設定を表示できます。

```
/subsystem=undertow/servlet-container=default/setting=websockets:read-resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "buffer-pool" => "default",
    "dispatch-to-worker" => true,
    "worker" => "default"
  }
}
```

また、WebSocket のコードサンプルは JBoss EAP に同梱されるクイックスタートに含まれています。

4.14. シングルサインオンサーバーの変更

The **infinispan** subsystem still provides distributed caching support for HA services in the form of Infinispan caches in JBoss EAP 7; however the caching and distribution of authentication information is handled differently than in previous releases.

JBoss EAP 6 では、シングルサインオン (SSO) が Infinispan キャッシュに提供されないと、キャッシュが分散されませんでした。

JBoss EAP 7 では、HA プロファイルを選択すると SSO が自動的に分散されるようになりました。HA プロファイルを実行している場合、各ホストは web キャッシュコンテナのデフォルトキャッシュを基にした独自の Infinispan キャッシュを持ちます。このキャッシュは関連するセッションとホストの SSO クッキーの情報を格納します。JBoss EAP は、各キャッシュ情報をすべてのホストに伝搬する処理を行います。JBoss EAP 7 では、明確に Infinispan キャッシュを SSO に割り当てる方法はありません。

JBoss EAP 7 では、SSO はサーバー設定ファイルの **undertow** サブシステムで設定されます。

JBoss EAP 7 に移行する際、アプリケーションコードを変更する必要はありません。

4.15. データソース設定の変更

4.15.1. JDBC データソースドライバー名

以前のリリースの JBoss EAP でデータソースを設定した場合、ドライバー名に指定される値は JDBC ドライバー JAR に含まれる **META-INF/services/java.sql.Driver** ファイルにリストされたクラスの数によって決まりました。

単一クラスを含むドライバー

META-INF/services/java.sql.Driver ファイルで指定されたクラスが 1 つのみであった場合、ドライバー名は単に JDBC ドライバー JAR の名前でした。これは JBoss EAP 7 でも変更ありません。

複数クラスを含むドライバー

JBoss EAP 6 では、**META-INF/services/java.sql.Driver** ファイルに複数のクラスが記載されている場合、以下の形式で JAR 名にドライバークラスにするクラスの名前とメジャーおよびマイナーバージョンを追加してクラスを指定していました。

```
JAR_NAME + DRIVER_CLASS_NAME + "_" + MAJOR_VERSION + "_" + MINOR_VERSION
```

JBoss EAP 7 ではこれが変更され、以下の形式でドライバー名を指定します。

```
JAR_NAME + "_" + DRIVER_CLASS_NAME + "_" + MAJOR_VERSION + "_" + MINOR_VERSION
```



注記

JAR_NAME と DRIVER_CLASS_NAME の間にアンダースコアが加えられています。

2 つのクラスが含まれるドライバーの例としては、MySQL 5.1.31 JDBC ドライバーが挙げられます。このドライバークラス名は **com.mysql.jdbc.Driver** となります。以下の 2 つの例では、JBoss EAP の以前のリリースと現行リリースにおけるドライバー名の指定方法の違いが示されています。

JBoss EAP 6 ドライバー名の例

```
mysql-connector-java-5.1.31-bin.jarcom.mysql.jdbc.Driver_5_1
```

JBoss EAP 7 ドライバー名の例

```
mysql-connector-java-5.1.31-bin.jar_com.mysql.jdbc.Driver_5_1
```

4.16. セキュリティーサーバー設定の変更

For information about Java Security Manager server configuration changes, see [Considerations Moving from Previous Versions](#) in [How To Configure Server Security](#) for JBoss EAP.

4.17. トランザクションサブシステムの変更

JBoss EAP 6 の **transactions** サブシステムで使用できた Transaction Manager 設定属性の一部が JBoss EAP 7 で変更になりました。

Removed Transactions Subsystem Attributes

以下の表は、JBoss EAP 7 の **transactions** サブシステムから削除された JBoss EAP 6 の属性と、それらの代替となる同等の属性を示しています。

JBoss EAP 6 の属性	JBoss EAP 7 で代替となる属性
パス	object-store-path
relative-to	object-store-relative-to

非推奨となったトランザクションサブシステム属性

The following attributes that were available in the **transactions** subsystem in JBoss EAP 6 are deprecated in JBoss EAP 7. The deprecated attributes might be removed in a future release of the product. The following table lists the equivalent replacement attributes.

JBoss EAP 6 の属性	JBoss EAP 7 で代替となる属性
use-hornetq-store	use-journal-store
hornetq-store-enable-async-io-to	journal-store-enable-async-io
enable-statistics	statistics-enabled

4.18. MOD_CLUSTER 設定の変更

JBoss EAP 7 では、`mod_cluster` の静的プロキシの設定が変更になりました。

JBoss EAP 6 では、**hostname:port** の形式で指定された `httpd` プロキシアドレスのカンマ区切りリストである **proxy-list** 属性を設定しました。

proxy-list 属性は JBoss EAP 7 では非推奨になりました。この属性は、アウトバウンドソケットバインディング名のリストである **proxies** 属性に置き換えられました。

This change impacts how you define a static proxy list, for example, when disabling advertising for `mod_cluster`. For information about how to disable advertising for `mod_cluster`, see [Disable Advertising for mod_cluster](#) in the JBoss EAP **Configuration Guide**.

For more information about `mod_cluster` attributes, see [ModCluster Subsystem Attributes](#) in the JBoss EAP **Configuration Guide**.

第5章 アプリケーション移行の変更

5.1. WEB サービスアプリケーションの変更

主に [Apache CXF](#)、[Apache WSS4J](#)、および [Apache Santuario](#) コンポーネントがアップグレードされ、JBossWS 5 は JBoss EAP 7 の Web サービスに新機能と改良されたパフォーマンスを提供します。

5.1.1. JAX-RPC サポートの変更

XML ベースの RPC (JAX-RPC) は Java EE 6 で非推奨となり、Java EE 7 ではオプションとなりました。JAX-RPC は JBoss EAP 7 では使用できず、サポートされません。JAX-RPC を使用するアプリケーションは、現在の Java EE 標準の Web サービスフレームワークである [JAX-WS](#) に移行する必要があります。

JAX-RPC web サービスの使用は、以下のいずれかの方法で特定できます。

- JAX-RPC マッピングファイルの存在。これは、root 要素 `<java-wsdl-mapping>` のある XML ファイルです。
- `webservices.xml` XML 記述子ファイルの存在。このファイルには `<jaxrpc-mapping-file>` 子要素が含まれる `<webservice-description>` 要素があります。以下に JAX-RPC Web サービスを定義する `webservices.xml` 記述子ファイルの例を示します。

```
<webservices xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://www.ibm.com/webservices/xsd/j2ee_web_services_1_1.xsd"
  version="1.1">
  <webservice-description>
    <webservice-description-name>HelloService</webservice-
description-name>
    <wsdl-file>WEB-INF/wsdl/HelloService.wsdl</wsdl-file>
    <jaxrpc-mapping-file>WEB-INF/mapping.xml</jaxrpc-mapping-file>
    <port-component>
      <port-component-name>Hello</port-component-name>
      <wsdl-port>HelloPort</wsdl-port>
      <service-endpoint-
interface>org.jboss.chap12.hello.Hello</service-endpoint-interface>
      <service-impl-bean>
        <servlet-link>HelloWorldServlet</servlet-link>
      </service-impl-bean>
    </port-component>
  </webservice-description>
</webservices>
```

- `ejb-jar.xml` ファイルの存在。これには、JAX-RPC マッピングファイルを参照する `<service-ref>` が含まれます。

5.1.2. Apache CXF Spring Web サービスの変更

以前のリリースの JBoss EAP では、エンドポイントデプロイメントアーカイブを `jbossws-cxf.xml` 設定ファイルに含め、JBossWS と Apache CXF の統合をカスタマイズすることができました。このユースケースの 1 つが、Apache CXF バスで Web サービスクライアントおよびサーバーエンドポイン

トのインターセプターチェーンを設定することでした。この統合には、JBoss EAP サーバーに Spring をデプロイする必要がありました。

JBoss EAP 7 では、Spring の統合がサポートされなくなりました。`jbossws-cxf.xml` 記述子設定ファイルが含まれるアプリケーションを編集し、このファイルに定義されているカスタム設定を置き換える必要があります。JBoss EAP 7 でも Apache CXF API に直接アクセスすることはできますが、アプリケーションは移植できないことに注意してください。

可能な場合は、Spring のカスタム設定を新しい JBossWS 記述子設定オプションに置き換えることが推奨されます。この JBossWS 記述子ベースの方法では、クライアントエンドポイントコードを編集する必要がなく、同様の機能を提供できます。場合によっては、Spring を Context and Dependency Injection (コンテキストと依存性の注入、CDI) に置き換えることができます。

Apache CXF インターセプター

JBossWS 記述子は、クライアントエンドポイントコードを編集せずにインターセプターを宣言できる新しい設定オプションを提供します。`cxf.interceptors.in` および `cxf.interceptors.out` プロパティーのインターセプタークラス名のリストを指定して、事前定義されたクライアントおよびエンドポイント設定内でインターセプターを宣言します。

以下は、これらのプロパティーを使用してインターセプターを宣言する `jaxws-endpoint-config.xml` ファイルの例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<jaxws-config xmlns="urn:jboss:jbossws-jaxws-config:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:javase="http://java.sun.com/xml/ns/javase"
  xsi:schemaLocation="urn:jboss:jbossws-jaxws-config:4.0 schema/jbossws-
jaxws-config_4_0.xsd">
  <endpoint-config>
    <config-
name>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointImpl</config-name>
    <property>
      <property-name>cxf.interceptors.in</property-name>
      <property-
value>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointInterceptor,org.jbo
ss.test.ws.jaxws.cxf.interceptors.FooInterceptor</property-value>
    </property>
    <property>
      <property-name>cxf.interceptors.out</property-name>
      <property-
value>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointCounterInterceptor<
/property-value>
    </property>
    </endpoint-config>
  </jaxws-config>
```

Apache CXF の機能

JBossWS 記述子を使用すると、`cxf.features` プロパティーの機能クラス名のリストを指定して、事前定義のクライアントおよびエンドポイント設定内で機能を宣言できます。

以下は、このプロパティーを使用して機能を宣言する `jaxws-endpoint-config.xml` ファイルの例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<jaxws-config xmlns="urn:jboss:jbossws-jaxws-config:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xmlns:javaee="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="urn:jboss:jbossws-jaxws-config:4.0 schema/jbossws-
jaxws-config_4_0.xsd">
  <endpoint-config>
    <config-name>Custom FI Config</config-name>
    <property>
      <property-name>cxf.features</property-name>
      <property-value>org.apache.cxf.feature.FastInfoSetFeature</property-
value>
    </property>
  </endpoint-config>
</jaxws-config>

```

Apache CXF HTTP トランスポート

Apache CXF では、`org.apache.cxf.transport.http.HTTPConduit` オプションを指定すると HTTP トランスポートの設定を実行できます。JBossWS の統合により、以下のように Apache CXF API を使用して conduit がプログラマ的に編集されます。

```

import org.apache.cxf.frontend.ClientProxy;
import org.apache.cxf.transport.http.HTTPConduit;
import org.apache.cxf.transports.http.configuration.HTTPClientPolicy;

// Set chunking threshold before using a JAX-WS port client
...
HTTPConduit conduit =
(HTTPConduit)ClientProxy.getClient(port).getConduit();
HTTPClientPolicy client = conduit.getClient();

client.setChunkingThreshold(8192);
...

```

また、システムプロパティを設定すると Apache CXF `HTTPConduit` のデフォルト値を制御および上書きできます。

プロパティ	タイプ	説明
<code>cxf.client.allowChunking</code>	ブール値	チャンキングを使用してリクエストを送信するかどうかを指定します。
<code>cxf.client.chunkingThreshold</code>	整数	非チャンキングからチャンキングモードに切り替えるしきい値を設定します。
<code>cxf.client.connectionTimeout</code>	Long	接続タイムアウトをミリ秒単位で設定します。
<code>cxf.client.receiveTimeout</code>	Long	受信タイムアウトをミリ秒単位で設定します。
<code>cxf.client.connection</code>	文字列	Keep-Alive または close 接続タイプを使用するかどうかを指定します。
<code>cxf.tls-client.disableCNCheck</code>	ブール値	CN ホスト名チェックを無効化するかどうかを指定します。

5.1.3. WS-Security の変更

- アプリケーションに、**org.apache.ws.security.WSPasswordCallback** クラスにアクセスするカスタムコールバックハンドラーが含まれている場合、このクラスは **org.apache.wss4j.common.ext** パッケージに移動されたため注意してください。
- ほとんどの SAML Bean オブジェクトは **org.apache.ws.security.saml.ext** パッケージから **org.apache.wss4j.common.saml package** に移動されました。
- RSA v1.5キートランスポートおよび関連するすべてのアルゴリズムの使用はデフォルトで禁止されています。
- これまで、セキュリティトークンサービス (STS) は **onBehalfOf** トークンのみを検証しましたが、**ActAs** トークンも検証するようになりました。そのため、**ActAs** トークンに提供される **UsernameToken** に有効なユーザー名とパスワードを指定する必要があります。
- SAML Bearer トークンには内部署名が必要になりました。署名の検証を有効または無効にするため、**org.apache.wss4j.dom.validate.SamlAssertionValidator** クラスに **setRequireBearerSignature()** メソッドが含まれるようになりました。

5.1.4. JBoss モジュール構造の変更

cxfr-api および **cxfr-rt-core** JAR が 1つの **cxfr-core** JAR に統合されました。そのため、JBoss EAP の **org.apache.cxf** モジュールに **cxfr-core** JAR が含まれるようになり、これまでのリリースよりも多いクラスが公開されました。

5.1.5. Bouncy Castle の要件の変更

XML/WS-Security での対称暗号化で Galois/Counter Mode (GCM) を用いた AES 暗号化を使用したい場合、BouncyCastle Security Provider が必要になります。

JBoss EAP 7 には **org.bouncycastle** モジュールが同梱されているため、JBossWS はそのクラスローダーに依存して BouncyCastle Security Provider を入手および使用できるようになりました。そのため、現在の JVM に BouncyCastle を静的にインストールする必要がなくなりました。コンテナの外部で実行しているアプリケーションの場合、BouncyCastle ライブラリーをクラスパスに追加すると JBossWS はこのセキュリティープロバイダーを使用できます。

この動作を無効にするには、**jaxws-endpoint-config.xml** デプロイメント記述子ファイル (サーバーの場合) または **jaxws-client-config.xml** 記述子ファイル (クライアントの場合) で **org.jboss.ws.cxf.noLocalBC** プロパティの値を **true** に設定します。

JBoss EAP に同梱されるバージョンでないものを使用したい場合は、BouncyCastle を静的に JVM にインストールできます。この場合、静的にインストールされた BouncyCastle Security Provider はクラスパスに存在するプロバイダーよりも優先的に選択されます。この問題を回避するには、BouncyCastle 1.49、1.51、またはそれ以降のバージョンを使用する必要があります。

5.1.6. Apache CXF バス選択ストラテジー

コンテナ内で実行されているクライアントのデフォルトのバス選択ストラテジーが **THREAD_BUS** から **TCCL_BUS** に変更されました。コンテナ外部で実行されているクライアントのデフォルトストラテジーは **THREAD_BUS** で、変更はありません。以下の方法の 1 つを使用すると、以前のリリースでの動作を復元できます。

- `org.jboss.ws.cxf.jaxws-client.bus.strategy` システムプロパティーの値を `THREAD_BUS` に設定して JBoss EAP サーバーを起動します。
- クライアントコードで選択ストラテジーを明示的に設定します。

5.1.7. WebServiceRef の JAX-WS 2.2 要件

コンテナは [WebServiceFeature](#) を使用して JAX-WS 2.2 スタイルのコンストラクターを使用し、Web サービス参照ヘインジェクトされるクライアントを構築する必要があります。よって、コンテナによってインジェクトされるユーザー提供のサービスクラスは JAX-WS 2.2 以上を実装する必要があります。

5.1.8. IgnoreHttpsHost CN チェックの変更

以前のリリースでは、システムプロパティー `org.jboss.security.ignoreHttpsHost` を `true` に設定すると、証明書にあるサービスの一般名 (CN) に対する HTTPS URL ホスト名チェックを無効にすることができました。このシステムプロパティー名は `cxf.tls-client.disableCNCheck` に置き換えられました。

5.1.9. サーバー側設定およびクラスローディング

As a consequence of enabling injections into service endpoint and service client handlers, it is no longer possible to automatically load handler classes from the `org.jboss.as.webservices.server.integration` JBoss module. If your application depends on a given predefined configuration, you might need to explicitly define new module dependencies for your deployment. For more information, see [Migrate Explicit Module Dependencies](#)

5.1.10. Java Endorsed Standards Override Mechanism の非推奨

[Java Endorsed Standards Override Mechanism](#) は JDK 1.8_40 では非推奨になり、JDK 9 では削除される予定です。これは、JAR を JRE 内の endorsed ディレクトリーに置くことで、デプロイされたアプリケーションすべてがライブラリーを利用できるメカニズムです。

アプリケーションが Apache CXF の JBossWS 実装を使用する場合、JBoss EAP 7 では必要な依存関係が正しい順序で追加されるため、この変更による影響はないはずです。アプリケーションが Apache CXF に直接アクセスする場合、アプリケーションデプロイメントの一部として JBossWS の依存関係の後に Apache CXF の依存関係を提供する必要があります。

5.1.11. EAR アーカイブでの記述子の仕様

以前のリリースの JBoss EAP では、EJB Web サービスデプロイメントの `jboss-webservices.xml` デプロイメント記述子ファイルを JAR アーカイブの `META-INF/` ディレクトリーまたは POJO Web サービスデプロイメントの `WEB-INF/` ディレクトリーと、WAR アーカイブにバンドル化された EJB Web サービスエンドポイントに設定することができました。

JBoss EAP 7 では、`jboss-webservices.xml` デプロイメント記述子ファイルを EAR アーカイブの `META-INF/` ディレクトリーで設定できるようになりました。`jboss-webservices.xml` ファイルが EAR アーカイブと JAR (または WAR) アーカイブの両方で見つかった場合、JAR または WAR の `jboss-webservices.xml` ファイルにある設定データによって EAR 記述子ファイルの対応するデータが上書きされます。

5.2. リモート URL コネクターおよびポートの更新

JBoss EAP 7 では、デフォルトのコネクターが **remote** から **http-remoting** に変更になり、デフォルトのリモート接続ポートが **4447** から **8080** に変更になりました。デフォルト設定の JNDI プロバイダー URL は **remote://localhost:4447** から **http-remoting://localhost:8080** に変更になりました。

If you use the JBoss EAP 7 **migrate** operation to update your configuration, you do not need to modify the remote connector, remote port, or JNDI provider URLs because the migration operation preserves the JBoss EAP 6 remoting connector and **4447** port configuration settings in the subsystem configuration. For more information about the **migrate** operation, see [Management CLI Migration Operation](#).

migrate 操作を使用せずに、新しい JBoss EAP 7 のデフォルト設定を使用して実行する場合は、新しい設定を使用するようにリモートコネクター、リモートポート、および JNDI プロバイダー URL を変更する必要があります。

5.3. メッセージングアプリケーションの変更

5.3.1. JMS デプロイメント記述子の置き換えおよび更新

ネーミングパターン - **jms.xml** によって識別されたプロプラエタリーの JMS リソースデプロイメント記述子ファイルは JBoss EAP 7 で非推奨になりました。以下は、JBoss EAP 6 での JMS リソースデプロイメント記述子ファイルの例になります。

```
<?xml version="1.0" encoding="UTF-8"?>
<messaging-deployment xmlns="urn:jboss:messaging-deployment:1.0">
  <hornetq-server>
    <jms-destinations>
      <jms-queue name="testQueue">
        <entry name="queue/test"/>
        <entry name="java:jboss/exported/jms/queue/test"/>
      </jms-queue>
      <jms-topic name="testTopic">
        <entry name="topic/test"/>
        <entry name="java:jboss/exported/jms/topic/test"/>
      </jms-topic>
    </jms-destinations>
  </hornetq-server>
</messaging-deployment>
```

以前のリリースで **-jms.xml** JMS デプロイメント記述子をアプリケーションで使用した場合、[Java EE 7 仕様](#)の EE.5.18 で指定されたとおりに標準の Java EE 7 デプロイメント記述子を使用するようアプリケーションを変換するか、JBoss EAP 7 の **messaging-activemq** サブシステムを使用するようデプロイメント記述子を更新してください。

記述子の更新を選択した場合、以下の変更を加える必要があります。

- ネームスペースを "urn:jboss:messaging-deployment:1.0" から "urn:jboss:messaging-activemq-deployment:1.0" に変更します。
- **<hornetq-server>** 要素名を **<server>** に変更します。

編集後のファイルは以下の例のようになるはずです。

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<messaging-deployment xmlns="urn:jboss:messaging-activemq-deployment:1.0">
  <server>
    <jms-destinations>
      <jms-queue name="testQueue">
        <entry name="queue/test"/>
        <entry name="java:jboss/exported/jms/queue/test"/>
      </jms-queue>
      <jms-topic name="testTopic">
        <entry name="topic/test"/>
        <entry name="java:jboss/exported/jms/topic/test"/>
      </jms-topic>
    </jms-destinations>
  </server>
</messaging-deployment>
```

メッセージングに関するサーバー設定の変更については、[メッセージングサーバー設定の変更](#)を参照してください。

5.3.2. 外部 JMS クライアントの更新

JBoss EAP 7 は JMS 1.1 API をサポートするため、コードを変更する必要はありません。

JBoss EAP 7 ではデフォルトのリモートコネクタおよびポートが変更になりました。この変更の詳細は [リモート URL コネクタおよびポートの更新](#)を参照してください。

migrate 操作を使用してサーバー設定を移行する場合、これまでの設定は保持され、**PROVIDER_URL** を更新する必要はありません。しかし、新しい JBoss EAP 7 のデフォルト設定を使用して実行する場合は、新しい **http-remoting://localhost:8080** 設定を使用するよう **PROVIDER_URL** を変更する必要があります。詳細は [リモートネーミングクライアントコードの移行](#)を参照してください。

JMS 2.0 API を使用するためにコードを移行する計画がある場合は、作業例を **helloworld-jms** クリックスタートで確認してください。

5.3.3. HornetQ API の置換

JBoss EAP 6 には **org.hornetq** モジュールが含まれ、これによりアプリケーションソースコードで **HornetQ API** が使用できました。

JBoss EAP 7 では HornetQ が Apache ActiveMQ Artemis に置き換えられたため、HornetQ API を使用したコードは **Apache ActiveMQ Artemis API** を使用するように移行する必要があります。この API のライブラリーは、**org.apache.activemq.artemis** モジュールに含まれています。

ActiveMQ Artemis は HornetQ の進化版なので、概念の多くは継続して適用されます。

5.4. JAX-RS および RESTEasy アプリケーションの変更

JBoss EAP の以前のリリースでは、JAX-RS 1.x の実装である RESTEasy 2 がバンドルされていました。JBoss EAP 7 には、[JSR 339: JAX-RS 2.0: The Java API for RESTful Web Services](#) 仕様で定義されている JAX-RS 2.0 の実装となる RESTEasy 3 が含まれています。RESTful Web サービスの Java API に関する詳細情報は、[JAX-RS 2.0 API Specification](#) を参照してください。

JBoss EAP に含まれる Jackson のバージョンは変更されました。JBoss EAP の以前のリリースに含まれていた Jackson は 1.9.9 でしたが、JBoss EAP 7 には Jackson 2.6.3 またはそれ以降が含まれています。

This section describes how these changes might impact applications that use RESTEasy or JAX-RS.

5.4.1. 非推奨の RESTEasy クラス

インターセプターおよび `MessageBody` クラス

JSR 311: JAX-RS: The Java™ API for RESTful Web Services did not include an interceptor framework, so RESTEasy 2 provided one. JSR 339: JAX-RS 2.0: The Java API for RESTful Web Services introduced an official interceptor and filter framework, so the interceptor framework included in RESTEasy 2 is now deprecated, and is replaced by the JAX-RS 2.0 compliant interceptor facility in RESTEasy 3.x. The relevant interfaces are defined in the `javax.ws.rs.ext` package of the `jaxrs-api` module.

- RESTEasy 3.x では、以下のインターセプターインターフェースが非推奨になりました。
 - `org.jboss.resteasy.spi.interception.PreProcessInterceptor`
 - `org.jboss.resteasy.spi.interception.PostProcessInterceptor`
 - `org.jboss.resteasy.spi.interception.ClientExecutionInterceptor`
 - `org.jboss.resteasy.spi.interception.ClientExecutionContext`
 - `org.jboss.resteasy.spi.interception.AcceptedByMethod`
- `org.jboss.resteasy.spi.interception.PreProcessInterceptor` インターフェースは RESTEasy 3.x の `javax.ws.rs.container.ContainerRequestFilter` インターフェースに置き換えられました。
- RESTEasy 3.x では、以下のインターフェースとクラスも非推奨になりました。
 - `org.jboss.resteasy.spi.interception.MessageBodyReaderInterceptor`
 - `org.jboss.resteasy.spi.interception.MessageBodyWriterInterceptor`
 - `org.jboss.resteasy.spi.interception.MessageBodyWriterContext`
 - `org.jboss.resteasy.spi.interception.MessageBodyReaderContext`
 - `org.jboss.resteasy.core.interception.InterceptorRegistry`
 - `org.jboss.resteasy.core.interception.InterceptorRegistryListener`
 - `org.jboss.resteasy.core.interception.ClientExecutionContextImpl`
- The `org.jboss.resteasy.spi.interception.MessageBodyWriterInterceptor` インターフェースは `javax.ws.rs.ext.WriterInterceptor` インターフェースに置き換えられました。
- In addition, some changes to the `javax.ws.rs.ext.MessageBodyWriter` interface might not be backward compatible with respect to JAX-RS 1.x. If your application used JAX-RS 1.x, review your application code to make sure you define `@Produces` or `@Consumes` for your endpoints. Failure to do so might result in an error similar to the following.

```
org.jboss.resteasy.core.NoMessageBodyWriterFoundFailure: Could not find MessageBodyWriter for response object of type: <OBJECT> of media type:
```

以下に、このエラーの原因となる REST エンドポイントの例を示します。

```
@Path("dates")
public class DateService {

    @GET
    @Path("daysuntil/{targetdate}")
    public long showDaysUntil(@PathParam("targetdate") String
targetDate) {
        DateLogger.LOGGER.logDaysUntilRequest(targetDate);
        final long days;

        try {
            final LocalDate date = LocalDate.parse(targetDate,
DateTimeFormatter.ISO_DATE);
            days = ChronoUnit.DAYS.between(LocalDate.now(), date);
        } catch (DateTimeParseException ex) {
            // ** DISCLAIMER **. This example is contrived.
            throw new
WebApplicationException(Response.status(400).entity(ex.getLocalizedMessage())
.type(MediaType.TEXT_PLAIN)
                .build());
        }
        return days;
    }
}
```

この問題を修正するには、次のように `javax.ws.rs.Produces` のインポートと `@Produces` アノテーションを追加します。

```
...
import javax.ws.rs.Produces;
...

@Path("dates")
public class DateService {

    @GET
    @Path("daysuntil/{targetdate}")
    @Produces(MediaType.TEXT_PLAIN)
    public long showDaysUntil(@PathParam("targetdate") String
targetDate) {
        DateLogger.LOGGER.logDaysUntilRequest(targetDate);
        final long days;

        try {
            final LocalDate date = LocalDate.parse(targetDate,
DateTimeFormatter.ISO_DATE);
            days = ChronoUnit.DAYS.between(LocalDate.now(), date);
        } catch (DateTimeParseException ex) {
            // ** DISCLAIMER **. This example is contrived.
            throw new
WebApplicationException(Response.status(400).entity(ex.getLocalizedMessage())
.type(MediaType.TEXT_PLAIN)
                .build());
        }
    }
}
```

```

    }
    return days;
}
}

```



注記

RESTEasy の以前のリリースからのインターセプターはすべて、新規の JAX-RS 2.0 フィルターおよびインターセプターインターフェースと並行して実行することが可能です。

For more information about interceptors, see [RESTEasy Interceptors](#) in **Developing Web Services Applications** for JBoss EAP.

新しい代替の API については、[RESTEasy JAX-RS 3.0.13.Final API](#) もしくはそれ以降を参照してください。

クライアント API

`resteasy-jaxrs` の RESTEasy クライアントフレームワークは、JAX-RS 2.0 準拠の `resteasy-client` モジュールに置き換えられました。そのため、RESTEasy クライアント API クラスおよびメソッドの中には非推奨となっているものもあります。

- 以下のクラスは非推奨になりました。
 - [org.jboss.resteasy.client.ClientRequest](#)
 - [org.jboss.resteasy.client.ClientRequestFactory](#)
 - [org.jboss.resteasy.client.ClientResponse](#)
 - [org.jboss.resteasy.client.ProxyBuilder](#)
 - [org.jboss.resteasy.client.ProxyConfig](#)
 - [org.jboss.resteasy.client.ProxyFactory](#)
- [org.jboss.resteasy.client.ClientResponseFailure](#) 例
外、[org.jboss.resteasy.client.ClientExecutor](#) インターフェースおよび
[org.jboss.resteasy.client.EntityTypeFactory](#) インターフェースも非推奨になりました。
- [org.jboss.resteasy.client.ClientRequest](#) および
[org.jboss.resteasy.client.ClientResponse](#) クラスを、それぞれ
[org.jboss.resteasy.client.jaxrs.ResteasyClient](#) および
[javax.ws.rs.core.Response](#) に置き換える必要があります。
以下は RESTEasy 2.3.x の RESTEasy クライアントでリンクヘッダーを送信する例です。

```

ClientRequest request = new
ClientRequest(generateURL("/linkheader/str"));
request.addLink("previous chapter", "previous",
"http://example.com/TheBook/chapter2", null);
ClientResponse response = request.post();
LinkHeader header = response.getLinkHeader();

```

以下は RESTEasy 3 の RESTEasy クライアントで上記と同じタスクを実行する例です。

```
ResteasyClient client = new ResteasyClientBuilder().build();
Response response =
client.target(generateURL("/linkheader/str")).request()
    .header("Link", "<http://example.com/TheBook/chapter2>";
    rel="previous");
title="previous chapter");
javax.ws.rs.core.Link link = response.getLink("previous");
```

JAX-RS Web サービスと対話する外部 JAX-RS RESTEasy クライアントの例は、**resteasy-jaxrs-client** クイックスタートを参照してください。

- **org.jboss.resteasy.client.cache** パッケージのクラスおよびインターフェースも非推奨になりました。これらのクラスとインターフェースは、**org.jboss.resteasy.client.jaxrs.cache** パッケージの同等のクラスおよびインターフェースに置き換えられました。



注記

org.jboss.resteasy.client.jaxrs API クラスの詳細については、[The RESTEasy JAX-RS JavaDoc](#) を参照してください。

StringConverter

org.jboss.resteasy.spi.StringConverter クラスは RESTEasy 3.x では非推奨になりました。この機能は JAX-RS 2.0 の **jax.ws.rs.ext.ParamConverterProvider** クラスを使用して置き換えられます。

5.4.2. 削除または保護されている RESTEasy クラス

ResteasyProviderFactory Add メソッド

org.jboss.resteasy.spi.ResteasyProviderFactory add() メソッドのほとんどは、RESTEasy 3.0 で削除または保護されました。たとえば、**addBuiltInMessageBodyReader()** および **addBuiltInMessageBodyWriter()** メソッドは削除され、**addMessageBodyReader()** および **addMessageBodyWriter()** メソッドは保護されました。

現時点では、**registerProvider()** と **registerProviderInstance()** のメソッドを使用してください。

RESTEasy 3 から削除された他のクラス

@org.jboss.resteasy.annotations.cache.ServerCached アノテーションは、JAX-RS メソッドへの応答をサーバーでキャッシュすることを指定するものでしたが、これは RESTEasy 3 から削除されたので、アプリケーションコードから削除する必要があります。

5.4.3. 他の RESTEasy 変更点

SignedInput および SignedOutput

- **resteasy-crypto** の **SignedInput** および **SignedOutput** は、**Content-Type** を **Request** または **Response** オブジェクトのいずれかから **multipart/signed** に設定する必要があります。そうでない場合は、**@Consumes** または **@Produces** アノテーションを使用する必要があります。

- **SignedOutput** および **SignedInput** を使用すると、**@Produces** または **@Consumes** アノテーションで **application/pkcs7-signature** MIME タイプを設定して、そのタイプの形式をバイナリ形式で返すことができます。
- **@Produces** または **@Consumes** が **text/plain** MIME タイプの場合、**SignedOutput** は base64 でエンコードされ、文字列として送信されます。

セキュリティーフィルター

@RolesAllowed、**@PermitAll**、および **@DenyAll** のセキュリティーフィルターは、"401 Unauthorized" ではなく "403 Forbidden" を返すようになりました。

クライアント側のフィルター

以前のリリースからの RESTEasy クライアント API を使用している場合は、新しい JAX-RS 2.0 クライアント側のフィルターはバインドされず、実行されません。

非同期 HTTP サポート

Because the JAX-RS 2.0 specification adds asynchronous HTTP support using the **@Suspended** annotation and the **AsyncResponse** interface, the RESTEasy proprietary API for asynchronous HTTP has been deprecated and might be removed as soon as RESTEasy 3.1. The asynchronous Tomcat and asynchronous JBoss Web modules have also been removed from the server installation. If you are not using the Servlet 3.0 container or higher, asynchronous HTTP server-side processing will be simulated and run synchronously in same request thread.

サーバー側のキャッシュ

サーバー側のキャッシュ設定が変更されました。詳細は、[RESTEasy Documentation](#) を参照してください。

5.4.4. RESTEasy SPI の変更点

SPI 例外

すべての SPI 失敗例外は非推奨となり、内部的には使用されません。これら是对応する JAX-RS 2.0 例外に置き換えられました。

非推奨の例外	jaxrs-api モジュールでの代替の例外
org.jboss.resteasy.spi.ForbiddenException	javax.ws.rs.ForbiddenException
org.jboss.resteasy.spi.MethodNotAllowedException	javax.ws.rs.NotAllowedException
org.jboss.resteasy.spi.NotAcceptableException	javax.ws.rs.NotAcceptableException
org.jboss.resteasy.spi.NotFoundException	javax.ws.rs.NotFoundException
org.jboss.resteasy.spi.UnauthorizedException	javax.ws.rs.NotAuthorizedException
org.jboss.resteasy.spi.UnsupportedMediaTypeException	javax.ws.rs.NotSupportedException

InjectorFactory および Registry

InjectorFactory および **Registry** SPI が変更されました。ドキュメントおよびサポートに沿って RESTEasy を使用してる場合は、問題はありません。

5.4.5. Jackson プロバイダーの変更

JBoss EAP に含まれる Jackson のバージョンは変更されました。JBoss EAP の以前のリリースに含まれていた Jackson は 1.9.9 でしたが、JBoss EAP 7 には Jackson 2.6.3 またはそれ以降が含まれています。このため、Jackson プロバイダーは **resteasy-jackson-provider** から **resteasy-jackson2-provider** に変更されました。

resteasy-jackson2-provider へのアップグレードにはいくつかのパッケージ変更が必要になります。たとえば、Jackson アノテーションパッケージは **org.codehaus.jackson.annotate** から **com.fasterxml.jackson.annotation** に変更されました。

To switch your application to use the default provider that was included in the previous release of JBoss EAP, see [Switching the Default Jackson Provider](#) in **Developing Web Services Applications** for JBoss EAP.

5.4.6. Spring RESTEasy 統合の変更

Spring 4.0 フレームワークは、Java 8 のサポートを導入しました。Spring と RESTEasy 3.x 統合を使用する場合は、使用するデプロイメントで最小 Spring バージョンに 4.2.x を指定してください。これは JBoss EAP 7 がサポートする安定性のある最も早期のバージョンです。

5.4.7. RESTEasy Jettison JSON プロバイダーの変更

RESTEasy Jettison JSON プロバイダーは JBoss EAP 7 では非推奨となり、デフォルトでデプロイメントに追加されなくなりました。推奨される RESTEasy Jackson プロバイダーに切り替えるようにしてください。Jettison プロバイダーの使用継続を希望する場合は、以下の例で示すように **jboss-deployment-descriptor.xml** ファイルでその明示的な依存関係を定義する必要があります。

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.jboss.resteasy.resteasy-jackson2-provider"/>
      <module name="org.jboss.resteasy.resteasy-jackson-provider"/>
    </exclusions>
    <dependencies>
      <module name="org.jboss.resteasy.resteasy-jettison-provider"
services="import"/>
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

For more information about how to define explicit dependencies, see [Add an Explicit Module Dependency to a Deployment](#) in the JBoss EAP **Development Guide**.

5.5. CDI 1.2 アプリケーションの変更

JBoss EAP 7 includes support for CDI 1.2. As a result, applications written using CDI 1.0 might see some changes in behavior when migrated to JBoss EAP 7. This section summarizes only a few of those changes.

Weld および CDI 1.2 についての追加情報は、以下の参照先で確認できます。

- [Contexts and Dependency Injection for the Java EE platform](#)

- [CDI 1.2 Javadoc](#)
- [Weld 2.3.3.Final - CDI Reference Implementation](#)

Bean アーカイブ

CDI によってスキャンされ、bean クラスを検索および処理するようにするため、有効な bean の bean クラスを bean アーカイブにデプロイする必要があります。

CDI 1.0 では、アプリケーションクライアント、EJB、またはライブラリー JAR の **META-INF/** ディレクトリーに **beans.xml** ファイルが含まれた場合や、WAR の **WEB-INF/** ディレクトリーに **beans.xml** ファイルが含まれた場合には、アーカイブが **明示的** bean アーカイブとして定義されました。

CDI 1.1 には **暗黙的** bean アーカイブが導入されました。これは bean を定義するアノテーションを持つ bean クラスが 1 つ以上またはセッション bean が 1 つ以上含まれるアーカイブです。暗黙的な bean アーカイブは CDI によってスキャンされ、型検索中に bean を定義するアノテーションを持つクラスのみが検出されます。詳細は、**Contexts and Dependency Injection for the Java EE platform** の [Type and Bean Discovery](#) を参照してください。

bean アーカイブは **all**、**annotated**、または **none** の bean 検索モードを持ちます。バージョンのない **beans.xml** ファイルが含まれる bean アーカイブは、デフォルトの bean 検索モード **all** を持ちます。バージョンが **1.1** 以上の **beans.xml** ファイルが含まれる bean アーカイブは **bean-discovery-mode** 属性を指定する必要があります。この属性のデフォルト値は **annotated** です。

以下の場合、アーカイブは bean アーカイブではありません。

- **bean-discovery-mode** が **none** である **beans.xml** ファイルが含まれる場合。
- **beans.xml** ファイルがない CDI 拡張が含まれる場合。

以下の場合、アーカイブは **明示的** bean アーカイブになります。

- バージョン番号が 1.1 以上で、**bean-discovery-mode** が **all** の **beans.xml** ファイルがアーカイブに含まれる場合。
- アーカイブにバージョン番号のない **beans.xml** ファイルが含まれる場合。
- アーカイブに空の **beans.xml** ファイルが含まれる場合。

以下の場合、アーカイブは **暗黙的** bean アーカイブになります。

- アーカイブに **beans.xml** ファイルが含まれなくても、bean を定義するアノテーションを持つ bean クラスが 1 つ以上含まれるか、セッション bean が 1 つ以上含まれる場合。
- アーカイブに **bean-discovery-mode** が **annotated** **beans.xml** ファイルが含まれる場合。

CDI 1.2 は [bean を定義するアノテーション](#) を以下に限定します。

- **@ApplicationScoped**、**@SessionScoped**、**@ConversationScoped**、および **@RequestScoped** アノテーション
- その他すべての通常スコープタイプ
- **@Interceptor** および **@Decorator** アノテーション
- **@Stereotype** アノテーションが付けられた **stereotype** アノテーションすべて

- **@Dependent** スコープアノテーション

bean アーカイブの詳細は、**Contexts and Dependency Injection for the Java EE platform** の [Bean Archives](#) を参照してください。

会話解決の明確化

会話コンテキストのライフサイクルは、[CDI Specification Issue CDI-411](#) で説明されているサーブレット仕様との競合を回避するために変更されました。会話スコープはすべてのサーブレットリクエストの間はアクティブで、他のサーブレットやサーブレットフィルターによるリクエストボディーや文字エンコードの設定を妨げないはずで

オブザーバー解決

イベント解決は、CDI 1.2 で部分的に書き換えられました。CDI 1.0 では、オブザーバーメソッドにすべてのイベント修飾子がある場合にイベントはオブザーバーメソッドに配信されました。CDI 1.2 では、オブザーバーメソッドにイベント修飾子がないかイベント修飾子のサブセットがある場合にイベントがオブザーバーメソッドに配信されます。

5.6. 明示的なモジュール依存関係の移行

前リリースの JBoss EAP ではモジュラークラスローディングシステムと JBoss モジュールが導入され、アプリケーションが使用できるクラスを細かに制御することができました。この機能によって、アプリケーションの **MANIFEST.MF** ファイルまたは **jboss-deployment-structure.xml** デプロイメント記述子ファイルを使用して暗示的なモジュール依存関係を設定できました。

アプリケーションで明示的なモジュール依存関係を定義した場合、JBoss EAP 7 で変更になった以下の項目に注意してください。

利用可能な依存関係の確認

JBoss EAP に含まれるモジュールが変更されました。アプリケーションを JBoss EAP 7 に移行する場合、**MANIFEST.MF** および **jboss-deployment-structure.xml** ファイルエントリを確認し、これらのエントリが本リリースで削除されたモジュールを参照しないようにしてください。

アノテーションのスキャンに必要な依存関係

以前のリリースの JBoss EAP では、EJB インターセプターの宣言時など、アノテーションのスキャン中に処理される必要があるアノテーションが依存関係に含まれていると、Jandex インデックスを生成して新しい JAR ファイルに含まれるようにし、**MANIFEST.MF** または **jboss-deployment-structure.xml** デプロイメント記述子ファイルにフラグを設定する必要がありました。

JBoss EAP 7 では、静的モジュールに対するアノテーションインデックスの自動ランタイム生成が提供されるため、手動で生成する必要がなくなりました。しかし、JBoss EAP 7 でも以下のように **annotations** フラグを **MANIFEST.MF** ファイルまたは **jboss-deployment-structure.xml** デプロイメント記述子ファイルに追加する必要があります。

MANIFEST.MF ファイルのアノテーションフラグの例

```
Dependencies: com.company.my-ejb annotations, com.company.other
```

jboss-deployment-structure.xml ファイルのアノテーションフラグの例

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="com.company.my-ejb" annotations="true"/>
      <module name="com.company.other"/>
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

```
</deployment>  
</jboss-deployment-structure>
```

5.7. HIBERNATE および JPA の移行の変更

5.7.1. Hibernate ORM 3.0

The integration classes that made it easier to use Hibernate ORM 3 in the previous release were removed from JBoss EAP 7. If your application still uses Hibernate ORM 3 libraries, it is strongly recommended that you migrate your application to use Hibernate ORM 5 as Hibernate ORM 3 will no longer work in JBoss EAP without a lot of effort. If you can not migrate to Hibernate ORM 5, you must define a custom JBoss Module for the Hibernate ORM 3 JARs and exclude the Hibernate ORM 5 classes from your application.

5.7.2. Hibernate ORM 4.0 - 4.3

If your application needs second-level cache enabled, you should migrate to Hibernate ORM 5, which is integrated with Infinispan 8.x.

Applications written with Hibernate ORM 4.x can still use Hibernate ORM 4.x. You must define a custom JBoss module for the Hibernate ORM 4.x JARs and exclude the Hibernate ORM 5 classes from your application. However, it is strongly recommended that you rewrite your application code to use Hibernate ORM 5. For information about migrating to Hibernate ORM 5, see [Migrating to Hibernate ORM 5](#).

5.7.3. Hibernate ORM 5

If your application contains a `persistence.xml` file or the code uses the `@PersistenceContext` or `@PersistenceUnit` annotations, JBoss EAP 7 detects this during deployment and assumes the application uses JPA. It implicitly adds the Hibernate ORM 5 libraries plus a few other dependencies to your application class path and defaults to using these libraries.

5.7.4. Migrating to Hibernate ORM 5

This section highlights the changes you need to make when migrating from Hibernate ORM version 4.3 to version 5. For more information about the changes implemented between Hibernate ORM 4 and Hibernate ORM 5, see the [Hibernate ORM 5.0 Migration Guide](#).

Removed and Deprecated Classes

The following deprecated classes were removed from Hibernate ORM 5.

- [org.hibernate.cfg.AnnotationConfiguration](#)
- [org.hibernate.id.TableGenerator](#)
- [org.hibernate.id.TableHiLoGenerator](#)
- [org.hibernate.id.SequenceGenerator](#)

Other Changes to Classes and Packages

- The [org.hibernate.integrator.spi.Integrator](#) interface changed to account for bootstrap redesign.

- A new package `org.hibernate.engine.jdbc.env` package was created. It contains the `org.hibernate.engine.jdbc.env.spi.JdbcEnvironment` interface, which was extracted from `org.hibernate.engine.jdbc.spi.JdbcServices` interface.
- A new `org.hibernate.boot.model.relational.ExportableProducer` interface was introduced that will affect `org.hibernate.id.PersistentIdentifierGenerator` implementations.
- The signature of `org.hibernate.id.Configurable` was changed to accept `org.hibernate.service.ServiceRegistry` rather than just `org.hibernate.dialect.Dialect`.
- The `org.hibernate.metamodel.spi.TypeContributor` interface has migrated to `org.hibernate.boot.model.TypeContributor`.
- The `org.hibernate.metamodel.spi.TypeContributions` interface has migrated to `org.hibernate.boot.model.TypeContributions`.

Type Handling

- Built-in `org.hibernate.type.descriptor.sql.SqlTypeDescriptor` implementations no longer auto-register themselves with `org.hibernate.type.descriptor.sql.SqlTypeDescriptorRegistry`. Applications using custom `SqlTypeDescriptor` implementations that extend the built-in implementations and rely on that behavior must be updated to call `SqlTypeDescriptorRegistry.addDescriptor()` themselves.
- For IDs defined as generated UUIDs, some databases require you to explicitly set the `@Column(length=16)` in order to generate `BINARY(16)` so that comparisons work properly.
- For `EnumType` mappings defined in the `hbm.xml`, where you want `javax.persistence.EnumType.STRING` name-mapping, this configuration must be explicitly stated by using either the `useNamed(true)` setting or by specifying a `VARCHAR` value of `12`.

Transaction Management

- The transaction SPI underwent a major redesign in Hibernate ORM 5. In Hibernate ORM 4.3, you used the `org.hibernate.Transaction` API to directly access different back-end transaction strategies. Hibernate ORM 5 introduced a level of indirection. On the back end, the `org.hibernate.Transaction` implementation now talks to a `org.hibernate.resource.transaction.TransactionCoordinator`, which represents the transactional context for a given session according to the back-end strategy. While this does not have a direct impact on developers, it could affect the bootstrap configuration. Previously applications would specify `hibernate.transaction.factory_class` property, which is now deprecated, and refer to a `org.hibernate.engine.transaction.spi.TransactionFactory` FQN (fully qualified name). With Hibernate ORM 5, you specify the `hibernate.transaction.coordinator_class` setting and refer to a `org.hibernate.resource.transaction.TransactionCoordinatorBuilder`. See `org.hibernate.cfg.AvailableSettings.TRANSACTION_COORDINATOR_STRATEGY` for additional details.
- The following short names are now recognized.
 - `jdbc`: Manage transactions using the JDBC `java.sql.Connection`. This is the default for

non-JPA transactions.

- o **jta**: Manage transactions using JTA.



重要

If a JPA application does not provide a setting for the **hibernate.transaction.coordinator_class** property, Hibernate will automatically build the proper transaction coordinator based on the transaction type for the persistence unit.

If a non-JPA application does not provide a setting for the **hibernate.transaction.coordinator_class** property, Hibernate will default to **jdbc** to manage the transactions. This default will cause problems if the application actually uses JTA-based transactions. A non-JPA application that uses JTA-based transactions should explicitly set the **hibernate.transaction.coordinator_class** property value to **jta** or provide a custom **org.hibernate.resource.transaction.TransactionCoordinatorBuilder** that builds a **org.hibernate.resource.transaction.TransactionCoordinator** that properly coordinates with JTA-based transactions.

Other Hibernate ORM 5 Changes

- The **cfg.xml** files are again fully parsed and integrated with events, security, and other functions.
- The properties loaded from the **cfg.xml** using the **EntityManagerFactory** did not previously prefix names with **hibernate**. This has now been made consistent.
- The configuration is no longer serializable.
- The **org.hibernate.dialect.Dialect.getQuerySequencesString()** method now retrieves catalog, schema, and increment values.
- The **AuditConfiguration** modifier was removed from **org.hibernate.envers.boot.internal.EnversService**.
- The **AuditStrategy** method parameters were changed to remove the obsolete **AuditConfiguration** and use the new **EnversService**.
- Various classes and interfaces in the **org.hibernate.hql.spi** package and subpackages have been moved to the new **org.hibernate.hql.spi.id** package. This includes the **MultiTableBulkIdStrategy** class and the **AbstractTableBasedBulkIdHandler**, **TableBasedDeleteHandlerImpl**, and **TableBasedUpdateHandlerImpl** interfaces and their subclasses.
- There was a complete redesign of property access contracts.
- Valid **hibernate.cache.default_cache_concurrency_strategy** setting values are now defined using the **org.hibernate.cache.spi.access.AccessType.getExternalName()** method rather than the **org.hibernate.cache.spi.access.AccessType** enum constants. This is more consistent with other Hibernate settings.

5.8. HIBERNATE SEARCH の変更

JBoss EAP 7 に同梱される Hibernate Search のバージョンが変更になりました。以前のリリースの JBoss EAP には Hibernate Search 4.6.x が含まれていましたが、JBoss EAP 7 には Hibernate Search 5.5.x が同梱されます。

Hibernate Search 5.5 is built upon Apache Lucene 5.3.1. If you use any native Lucene APIs, be sure to align with this version. The [Hibernate Search 5.5 API](#) wraps and hides the complexity of many of the Lucene API changes made between version 3 and version 5, however, some classes are now deprecated, renamed, or repackaged. This section describes how these changes might impact your application code.

Hibernate Search マッピングの変更

埋め込み関係の ID フィールドのインデックス化

@IndexedEmbedded アノテーションを使用して関連エントリーからのフィールドを含む場合、関連エントリーの **id** が含まれなくなりました。**id** が含まれるようにするには、**@IndexedEmbedded** アノテーションの **includeEmbeddedObjectId** 属性を使用します。

```
@IndexedEmbedded(includeEmbeddedObjectId=true)
```

番号および日付のインデックス形式の変更

番号や日付はデフォルトで数字フィールドとしてインデックス化されるようになりました。タイプ **int**、**long**、**float**、**double** のプロパティおよびこれらのラッパークラスは文字列としてインデックス化されないようになりました。代わりに、これらは Lucene の適切な数字エンコーディングを使用してインデックス化されるようになりました。**id** フィールドはこのルールの例外で、数字タイプによって表されても、デフォルトで文字列キーワードとしてインデックス化されます。**@NumericField** の使用は、数字エンコーディングのカスタム精度を指定する場合以外は廃止されました。文字列エンコーディングフィールドブリッジを明示的に指定すると、これまでの文字列ベースのインデックス形式を保持できます。整数の場合は [org.hibernate.search.bridge.builtin.IntegerBridge](#) になります。公開されているその他の使用可能なフィールドブリッジは、[org.hibernate.search.bridge.builtin](#) パッケージを確認してください。

Date および **Calendar** は文字列としてインデックス化されないようになりました。代わりに、インスタンスは 1970 年 1 月 1 日グリニッジ標準時 00:00:00 からの期間 (ミリ秒) を表す長整数としてエンコードされます。新しい [EncodingType](#) 列挙を使用するとインデックス形式を切り替えられます。以下に例を示します。

```
@DateBridge(encoding=EncodingType.STRING)
@CalendarBridge(encoding=EncodingType.STRING)
```

数字と日付のエンコーディングの変更は重要で、アプリケーションの動作に大きく影響する可能性があります。以前は文字列にエンコードされ、現在は数字にエンコードされるフィールドをターゲットとするクエリーがある場合、クエリーを更新する必要があります。**NumericRangeQuery** で数字フィールドを検索する必要があります。また、ファセットがターゲットとするすべてのフィールドが文字列でエンコードされるようにする必要があります。Search クエリー DSL を使用する場合、適切なクエリーが自動的に作成されるはずですが。

その他の Hibernate Search の変更

- ソートオプションが改良され、ソートオプションに対してフィールドエンコーディングが誤って指定されるとランタイム例外が発生するようになりました。また、あらかじめソートに使用されるフィールドが分かる場合、Lucene によってより高性能なソート機能が提供されます。Hibernate Search 5.5 は新しい **@SortableField** および **@SortableFields** アノテーションを提供します。詳細は、[Migration Guide from Hibernate Search 5.4 to 5.5](#) を参照してください。

- Lucene の **SortField** API には、以下のアプリケーションコードの変更を適用する必要があります。
以前のリリースの JBoss EAP では、以下のようにクエリーでソートフィールドのタイプを設定しました。

```
fulltextQuery.setSort(new Sort(new SortField("title",
SortField.STRING)));
```

JBoss EAP 7 での設定例は次のとおりです。

```
fulltextQuery.setSort(new Sort(new SortField("title",
SortField.Type.STRING)));
```

- **SearchFactory** は ORM の統合でのみ使用される必要があるため、**hibernate-search-engine** モジュールから **hibernate-search-orm** モジュールに移動されました。他のインタグレーターは **SearchIntegrator** のみに依存する必要があります。**SearchIntegrator** は、非推奨となった **SearchFactoryIntegrator** の代わりに使用されます。
- 列挙値 **SpatialMode.GRID** の名前が **SpatialMode.HASH** に変更になりました。
- **FullTextIndexEventListener** が最終クラスになりました。現在このクラスを拡張する場合、同じ機能を実現できる他の方法を見つける必要があります。
- **hibernate-search-analyzers** モジュールが削除されました。**org.apache.lucene:lucece-analyzers-common** などの適切な Lucene アーティファクトを直接使用する方法が推奨されます。
- The JMS controller API has changed. The JMS back-end dependency on Hibernate ORM was removed so that it could be used in other non-ORM environments. A consequence is that implementors of **org.hibernate.search.backend.impl.jms.AbstractJMSHibernateSearchController** must adjust to the new signature. This class is an internal class and it is recommended to use it as an example instead of extending it.
- **org.hibernate.search.spi.ServiceProvider** SPI がリファクターされました。古いサービスコントラクトで統合している場合は、**ServiceManager**、**Service**、**Startable**、および **Stoppable** の [Hibernate Search 5.5 Javadoc](#) で新しいコントラクトの詳細を確認してください。
- Lucene 3.x によって生成されたインデックスを保持し、これらのインデックスを Hibernate Search 5.0 以上で再構築していない場合、**IndexFormatTooOldException** が発生します。マスインデクサーでインデックスを再構築することが推奨されます。再構築できない場合は、Lucene の **IndexUpgrader** を使用してください。デフォルトの動作が変更になった場合があるため、注意して Hibernate Search のマッピングを更新する必要があります。詳細は [Apache Lucene Migration Guide](#) を参照してください。
- JBoss EAP 7 では、Apache Lucene が 3.6 から 5.3 にアップグレードされました。ご使用のコードが直接 Lucene のコードをインポートする場合は、[Apache Lucene Migration Guide](#) で変更の詳細を確認してください。[Lucene Change Log](#) にも追加の情報が記載されています。
- **@Field(indexNullAs=)** を使用してインデックスの null マーカー値をエンコードする場合、同じフィールドでインデックス化されるその他の値すべてに適合するマーカーのタイプである必要があります。たとえば、これまでは文字列「null」を使用して数字フィールドの null 値を

エンコードすることが可能でしたが、本リリースではこれができなくなりました。この代わりに、`-1`などの `null` 値を表す数字を選択する必要があります。

- ファセティングエンジンに大幅な改良が加えられました。ほとんどの変更は API には影響しませんが、ファセティングに使用する予定のフィールドすべてに `@Facet` または `@Facets` アノテーションを付ける必要があります。

Hibernate Search の名前変更および再パッケージされたクラス

以下は、名前変更または再パッケージされた Hibernate Search クラスのリストになります。

以前のパッケージおよびクラス	新しいパッケージおよびクラス
<code>org.hibernate.search.Environment</code>	<code>org.hibernate.search.cfg.Environment</code>
<code>org.hibernate.search.FullTextFilter</code>	<code>org.hibernate.search.filter.FullTextFilter</code>
<code>org.hibernate.search.ProjectionConstants</code>	<code>org.hibernate.search.engine.ProjectionConstants</code>
<code>org.hibernate.search.SearchException</code>	<code>org.hibernate.search.exception.SearchException</code>
<code>org.hibernate.search.Version</code>	<code>org.hibernate.search.engine.Version</code>

Lucene - 名前変更および再パッケージされたクラス

クエリーパーサーが新しいモジュールに移動されたため、パッケージが

`org.apache.lucene.queryParser.QueryParser` から

`org.apache.lucene.queryparser.classic.QueryParser` に変更になりました。

Lucene アナライザーの多くがリファクタされたため、パッケージが変更になりました。 [Apache Lucene Documentation](#) を参照してください。

`TokenizerFactory` や `TokenFilterFactory` などの Apache Solr ユーティリティークラスの一部が Apache Lucene に移動されました。これらのユーティリティーやカスタムアナライザーがアプリケーションによって使用される場合は、Apache Lucene で新パッケージ名を探す必要があります。

詳細は [Apache Lucene Migration Guide](#) を参照してください。

Hibernate Search で非推奨となった API

Hibernate Search で非推奨となったインターフェース、クラス、列挙、アノテーションタイプ、メソッド、コンストラクター、および列挙定数の完全リストは、 [Hibernate Search Deprecated API](#) ドキュメントを参照してください。

Hibernate Search で非推奨となったインターフェース

インターフェース	説明
<code>org.hibernate.search.store.IndexShardingStrategy</code>	Hibernate Search 4.4 で非推奨となりました。Hibernate Search 5 で削除される可能性があります。代わりに ShardIdentifierProvider を使用してください。

インターフェース	説明
org.hibernate.search.store.Workspace	This interface will be moved and should be considered non-public API. For more information, see HSEARCH-1915 .

Hibernate Search で非推奨となったクラス

クラス	説明
org.hibernate.search.filter.FilterKey	カスタムフィルターキーは非推奨となり、Hibernate Search 6 で削除される予定です。Hibernate Search 5.1 では、Lucene フィルターをキャッシュするキーは指定のフィルターパラメーターを基に自動的に算出されます。
org.hibernate.search.filter.StandardFilterKey	カスタムフィルターキーは非推奨となり、Hibernate Search 6 で削除される予定です。Hibernate Search 5.1 では、Lucene フィルターをキャッシュするキーは指定のフィルターパラメーターを基に自動的に算出されます。

Hibernate Search で非推奨となった列挙

列挙	説明
org.hibernate.search.annotations.FieldCacheType	非推奨となったため CacheFromIndex アノテーションを削除します。 Hibernate Search Deprecated Annotations を参照してください。

Hibernate Search で非推奨となったアノテーション

アノテーション	説明
org.hibernate.search.annotations.CacheFromIndex	アノテーションを削除してください。代替はありません。
org.hibernate.search.annotations.Key	カスタムフィルターキャッシュキーは非推奨機能となり、Hibernate Search 6 で削除される予定です。Hibernate Search 5.1 では、フィルターキャッシュキーはフィルターパラメーターを基に自動的に判断されるため、キーオブジェクトを提供する必要がなくなりました。

Hibernate Search で非推奨となったメソッド

メソッド	説明
<code>org.hibernate.search.FullTextSharedSessionBuilder.autoClose()</code>	代替はありません。
<code>org.hibernate.search.FullTextSharedSessionBuilder.autoClose(boolean)</code>	代替はありません。
<code>org.hibernate.search.cfg.IndexedMapping.cacheFromIndex(FieldCacheType...)</code>	今後削除される予定で、代替はありません。
<code>org.hibernate.search.cfg.EntityDescriptor.getCacheInMemory()</code>	今後削除される予定で、代替はありません。
<code>org.hibernate.search.cfg.ContainedInMapping.numericField()</code>	代わりに、 <code>field().numericField()</code> を呼び出します。
<code>org.hibernate.search.cfg.EntityDescriptor.setCacheInMemory(Map<String, Object>)</code>	今後削除される予定で、代替はありません。
<code>org.hibernate.search.MassIndexer.threadsForSubsequentFetching(int)</code>	このメソッドは削除される予定です。
<code>org.hibernate.search.query.dsl.FuzzyContext.withThreshold(float)</code>	<code>FuzzyContext.withEditDistanceUpTo(int)</code> を使用します。

Hibernate Search で非推奨となったコンストラクター

コンストラクター	説明
<code>org.hibernate.search.cfg.NumericFieldMapping(PropertyDescriptor, EntityDescriptor, SearchMapping)</code>	代わりに <code>NumericFieldMapping.NumericFieldMapping(String, PropertyDescriptor, EntityDescriptor, SearchMapping)</code> を使用します。

上級インテグレーターに影響する変更

ここでは、パブリック API の一部ではない変更について説明します。これらのアーティファクトは、Hibernate Search フレームワークを拡張するインテグレーターのみがアクセスできる必要があるため、通常の開発者には影響はないはずです。

- **`IndexWriterSetting.MAX_THREAD_STATES`** および **`IndexWriterSetting.TERM_INDEX_INTERVAL`** 列挙定数は非推奨になりました。これらは設定から読み取るプロパティに影響します。実際にこれらの列挙定数がないと、**`hibernate.search.Animals.2.indexwriter.term_index_interval = default`** などの設定プロパティが無視されます。これによる影響は、プロパティが適用されないことのみです。

- **SearchFactoryIntegrator** インターフェースが非推奨になりました。 **SearchIntegrator** を使用するよう、即座にすべてのコードを移行する必要があります。
- **SearchFactoryBuilder** クラスが非推奨になりました。代わりに **SearchIntegrationBuilder** を使用してください。
- The **HSQuery.getExtendedSearchIntegrator()** method has been deprecated. It might be possible to use **SearchIntegrator**, but it is preferable to remove it altogether.
- **DocumentBuilderIndexedEntity.getFieldCacheOption()** メソッドが非推奨になりました。代替メソッドはありません。
- **BuildContext.getIndexingStrategy()** メソッドが非推奨になりました。代わりに **BuildContext.getIndexingMode()** を使用してください。
- **DirectoryHelper.getVerifiedIndexDir(String, Properties, boolean)** メソッドが非推奨になりました。代わりに **DirectoryHelper.getVerifiedIndexPath(java.lang.String, java.util.Properties, boolean)** を使用してください。
- 以下は、名前変更または再パッケージされた Hibernate Search クラスのリストになります。

以前のパッケージおよびクラス	新しいパッケージおよびクラス
org.hibernate.search.engine.impl.SearchMappingBuilder	org.hibernate.search.engine.spi.SearchMappingHelper
org.hibernate.search.indexes.impl.DirectoryBasedIndexManager	org.hibernate.search.indexes.spi.DirectoryBasedIndexManager
org.hibernate.search.spi.MassIndexerFactory	org.hibernate.search.batchindexing.spi.MassIndexerFactory
org.hibernate.search.spi.SearchFactoryBuilder	org.hibernate.search.spi.SearchIntegratorBuilder
org.hibernate.search.spi.SearchFactoryIntegrator	org.hibernate.search.spi.SearchIntegrator

5.9. エンティティ BEAN の JPA への移行

Java EE 7 では EJB エンティティ bean のサポートが任意となり、JBoss EAP 7 ではサポート対象外になりました。そのため、JBoss EAP 7 への移行時に CMP (Container-Managed Persistence) および BMP (Bean-Managed Persistence) エンティティ bean を書き直し、Java Persistence API (JPA) エンティティを使用するようにする必要があります。

以前のリリースの JBoss EAP では、**javax.ejb.EntityBean** クラスを拡張し、必要なメソッドを実装してエンティティ bean がアプリケーションのソースコードに作成されました。作成後、エンティティ bean は **ejb-jar.xml** ファイルで設定されました。CMP エンティティ bean は、値が **Container** の **<persistence-type>** 子要素が含まれる **<entity>** 要素を使用して指定されました。BMP エンティティ bean は、値が **Bean** の **<persistence-type>** 子要素が含まれる **<entity>** 要素を使用して指定されました。

JBoss EAP 7では、コードの CMP および BMP エンティティ bean を Java Persistence API (JPA) エンティティに置き換える必要があります。JPA エンティティは `javax.persistence.*` クラスを使用して作成され、`persistence.xml` ファイルに定義されます。

以下は JPA エンティティクラスの例になります。

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
// User is a keyword in some SQL dialects!
@Table(name = "MyUsers")
public class MyUser {
    @Id
    @GeneratedValue
    private Long id;

    @Column(unique = true)
    private String username;
    private String firstName;
    private String lastName;

    public Long getId() {
        return id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

以下は `persistence.xml` ファイルの例になります。

```
<persistence version="2.1"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="my-unique-persistence-unit-name">
```

```
<properties>
  // properties...
</properties>
</persistence-unit>
</persistence>
```

JAP エンティティの実施例は、JBoss EAP 7 に同梱される **bmt**、**cmt**、および **hibernate5** クイックスタートを参照してください。

5.10. JPA 永続プロパティの変更

これまでの JBoss EAP リリースでの永続性動作と互換性を維持するため、新しい永続プロパティ **jboss.as.jpa.deferdetach** が追加されました。

jboss.as.jpa.deferdetach プロパティは、非 JTA トランザクションスレッドで使用されるトランザクションスコープの永続コンテキストが各 **EntityManager** 呼び出しの後にロードされたエントリーをデタッチするかどうか、または永続コンテキストが閉じられるまで待機するかどうか (セッション bean が終了するときなど) を制御します。このプロパティのデフォルト値は **false** で、各 **EntityManager** 呼び出しの後にエンティティがデタッチまたは消去されます。これは、**JPA 仕様** で定義されている正しいデフォルト動作です。プロパティの値が **true** に設定されると、永続コンテキストが閉じられるまでエンティティはデタッチされません。

JBoss EAP 5 では、**jboss.as.jpa.deferdetach** プロパティが **true** に設定された場合と同様に永続性が動作しました。JBoss EAP 5 から JBoss EAP 7 に移行するときにこの動作を保持するには、以下の例のように **persistence.xml** で **jboss.as.jpa.deferdetach** プロパティの値を **true** に設定する必要があります。

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">
  <persistence-unit name="EAP5_COMPAT_PU">
    <jta-data-source>java:jboss/datasources/ExampleDS</jta-data-source>
    <properties>
      <property name="jboss.as.jpa.deferdetach" value="true" />
    </properties>
  </persistence-unit>
</persistence>
```

In JBoss EAP 6 では、**jboss.as.jpa.deferdetach** プロパティが **false** に設定された場合と同様に永続性が動作しました。これは JBoss EAP 7 での動作と同じであるため、アプリケーションの移行時に変更を加える必要はありません。

5.11. EJB クライアントコードの移行

JBoss EAP 7 ではデフォルトのリモートコネクターおよびポートが変更になりました。この変更の詳細は [リモート URL コネクターおよびポートの更新](#) を参照してください。

migrate 操作を使用してサーバー設定を移行した場合は、旧設定は保持されるため以下の変更を行う必要はありません。しかし、新しい JBoss EAP 7 のデフォルト設定で実行する場合は以下の変更を行う必要があります。

5.11.1. デフォルトのリモート接続ポートの更新

`jboss-ejb-client.properties` ファイルのリモート接続ポートの値を **4447** から **8080** に変更します。

以下は、前リリースと本リリースの `jboss-ejb-client.properties` ファイルの例になります。

例: JBoss EAP 6 の `jboss-ejb-client.properties` ファイル

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=4447
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NONYMOUS=false
```

例: JBoss EAP 7 の `jboss-ejb-client.properties` ファイル

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=8080
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NONYMOUS=false
```

5.11.2. デフォルトコネクタの更新

新しい JBoss EAP 7 設定で実行している場合、デフォルトのコネクタは **remote** から **http-remoting** に変更になりました。この変更は、使用するライブラリーと接続するサーバーの JBoss EAP リリースが異なるクライアントに影響します。

- クライアントアプリケーションが JBoss EAP 6 の EJB クライアントライブラリーを使用し、JBoss EAP 7 サーバーへ接続する場合、**8080** 以外のポートで **remote** コネクタを公開するようサーバーを設定する必要があります。
- JBoss EAP 7 の EJB クライアントライブラリーを使用し、JBoss EAP 6 サーバーへ接続するクライアントアプリケーションは、サーバーインスタンスによって **http-remoting** コネクタは使用されず、**remote** コネクタが使用されることを認識する必要があります。これは、新しいクライアント側接続プロパティを定義することで実現されます。

```
remote.connection.default.protocol=remote
```

5.11.3. リモートネーミングクライアントコードの移行

新しいデフォルトの JBoss EAP 7 設定で実行している場合、新しいデフォルトのリモートポートおよびコネクタを使用するようクライアントコードを変更する必要があります。

以下の例は、リモートネーミングプロパティが JBoss EAP 6 のクライアントコードでどのように指定されていたかを示しています。

```
java.naming.factory.initial=org.jboss.naming.remote.client.InitialContextFactory
java.naming.provider.url=remote://localhost:4447
```

以下は、JBoss EAP 7 のクライアントコードでリモートネーミングプロパティを指定する方法の例になります。

```
java.naming.factory.initial=org.jboss.naming.remote.client.InitialContextFactory
java.naming.provider.url=http-remoting://localhost:8080
```

5.12. デプロイメント計画設定の移行

The Java EE Application Deployment specification (JSR-88) was intended to define a standard contract to enable tools from multiple providers to configure and deploy applications on any Java EE platform product. The contract required Java EE Product Providers to implement the **DeploymentManager** and other **javax.enterprise.deploy.spi** interfaces to be accessed by the Tool Providers. In case of JBoss EAP 6, a deployment plan is identified by an XML descriptor named **deployment-plan.xml** that is bundled in a ZIP or JAR archive.

ほとんどのアプリケーションサーバー製品は、より機能の充実した独自のデプロイメントソリューションを提供するため、この仕様の採用は限定的でした。そのため、JSR-88 のサポートは JBoss EE 7 より廃止されたため、JBoss EAP 7 でも廃止されました。

If you used JSR-88 to deploy your application, you must now use another method to deploy the application. The JBoss EAP management CLI **deploy** command provides a standard way to deploy archives to standalone servers or to server groups in a managed domain. For more information about the management CLI, see the [Management CLI Guide](#).

5.13. カスタムアプリケーションバルブの移行

カスタムバルブまたは **jboss-web.xml** XML ファイルで定義されたバルブは、手動で移行する必要があります。これには、**org.apache.catalina.valves.ValveBase** クラスを拡張して作成されたバルブや、**jboss-web.xml** 記述子ファイルの **<valve>** 要素で設定されたバルブが含まれます。



重要

jboss-web.xml ファイルに定義されたカスタムバルブおよびバルブは、対応する Undertow のビルトインハンドラーで書き直すか、置き換える必要があります。Undertow ハンドラーへのバルブのマッピングに関する詳細は [JBoss Web バルブの移行](#) を参照してください。

認証バルブは、Undertow ビルトイン認証メカニズムを使用して手動で置き換える必要があります。

デプロイメントで設定されたバルブの移行

JBoss EAP 6 では、カスタムバルブを **jboss-web.xml** web アプリケーション記述子ファイルで設定するとアプリケーションレベルで定義することができました。JBoss EAP 7 では、Undertow ハンドラーを使用して定義することもできます。

以下は、JBoss EAP 6 の **jboss-web.xml** ファイルに設定されたバルブの例になります。

```
<jboss-web>
  <valve>
    <class-name>org.jboss.examples.MyValve</class-name>
    <param>
      <param-name>myParam</param-name>
```

```

        <param-value>foobar</param-value>
    </param>
</valve>
</jboss-web>

```

For more information about how to create and configure custom handlers in JBoss EAP, see [Creating Custom Handlers](#) in the JBoss EAP **Development Guide**.

カスタムオーセンティケーターバルブの移行

オーセンティケーターバルブを移行する方法については、[セキュリティアプリケーションの変更](#)を参照してください。

5.14. セキュリティアプリケーションの変更

JBoss Web を Undertow で置き換えるには、JBoss EAP 7 のセキュリティ設定における変更が必要になります。

5.14.1. オーセンティケーターバルブの移行

オーセンティケーターバルブは、Undertow ビルトイン認証メカニズムを使用して手動で置き換える必要があります。オーセンティケーターバルブを移行する方法は、以下の項を参照してください。

5.14.2. PicketLink の変更

For information about the changes required for SSO with SAML v2 configuration, see [Changes from Previous Versions of JBoss EAP](#) in [How to Set Up SSO with SAML v2](#) for JBoss EAP.

5.14.3. その他のセキュリティアプリケーションの変更

For information about the differences in SSO configuration with Kerberos, see [Differences from Configuring Previous Versions JBoss EAP](#) in [How to Set Up SSO with Kerberos](#) for JBoss EAP.

5.15. JBOSS LOGGING の変更

アプリケーションが JBoss Logging を使用する場合、**org.jboss.logging** パッケージのアノテーションは JBoss EAP 7 では非推奨になったことに注意してください。アノテーションは **org.jboss.logging.annotations** パッケージに移動されたため、ソースコードを更新して新しいパッケージをインポートする必要があります。

また、アノテーションは個別の Maven **groupId:artifactId:version** (GAV) ID に移動されたため、プロジェクトの **pom.xml** ファイルで **org.jboss.logging:jboss-logging-annotations** の新しいプロジェクト依存関係を追加する必要があります。



注記

ロギングアノテーションのみが移動されました。**org.jboss.logging.BasicLogger** および **org.jboss.logging.Logger** は、これまでどおり **org.jboss.logging** パッケージにあります。

非推奨となったアノテーションクラスと代替クラスを以下の表に示します。

表5.1 非推奨となったロギングアノテーションの代替

非推奨となったクラス	代替クラス
org.jboss.logging.Cause	org.jboss.logging.annotations.Cause
org.jboss.logging.Field	org.jboss.logging.annotations.Field
org.jboss.logging.FormatWith	org.jboss.logging.annotations.FormatWith
org.jboss.logging.LoggingClass	org.jboss.logging.annotations.LoggingClass
org.jboss.logging.LogMessage	org.jboss.logging.annotations.LogMessage
org.jboss.logging.Message	org.jboss.logging.annotations.Message
org.jboss.logging.MessageBundle	org.jboss.logging.annotations.MessageBundle
org.jboss.logging.MessageLogger	org.jboss.logging.annotations.MessageLogger
org.jboss.logging.Param	org.jboss.logging.annotations.Param
org.jboss.logging.Property	org.jboss.logging.annotations.Property

5.16. JAVASERVER FACES (JSF) のコード変更

JSF 1.2 のサポート停止

JBoss EAP 6 では、`jboss-deployment-structure.xml` ファイルを作成して、アプリケーションデプロイメントで JSF 1.2 の使用を継続することができました。

JBoss EAP 7 には JSF 2.2 が含まれ、JSF 1.2 API はサポート対象外になりました。アプリケーションが JSF 1.2 を使用する場合、JSF 2.2 を使用するようアプリケーションを書き直す必要があります。

JSF 2.1 および JSF 2.2 の互換性の問題

JSF 2.1 API と JSF 2.2 API は完全に互換性があるわけではありません。リリース間で、`FACELET_CONTEXT_KEY` 定数値が `com.sun.faces.facelets.FACELET_CONTEXT` から `javax.faces.FACELET_CONTEXT` に変更されました。この値はコンパイラによってインライン化され、どちらかのリリースに対してコンパイルされたコードは他のリリースでは動作しません。

以下の例に似たコードが含まれ、JSF 2.1 API でコンパイルされたアプリケーションが JSF 2.2 API を使用する JBoss EAP 7 で実行されると、`NullPointerException` が発生します。この問題を修正するには、アプリケーションを JSF 2.2 API に対してコンパイルする必要があります。

```
Object obj =
FacesContext.getCurrentInstance().getAttributes().get(FaceletContext.FACELET_CONTEXT_KEY);
```

詳細は、[JAVASERVERFACES_SPEC_PUBLIC-1257](#) を参照してください。

5.17. モジュールクラスローティングの変更

JBoss EAP 7 では、複数のモジュールに同じクラスまたはパッケージが含まれる場合のクラスローディングの動作が変更になりました。

お互いに依存し、一部同じパッケージが含まれる **MODULE_A** と **MODULE_B** の 2 つのモジュールがあるとします。JBoss EAP 6 では、依存関係からロードされたクラスまたはパッケージは **module.xml** ファイルの **resource-root** に指定されたものよりも優先されました。これは、**MODULE_A** は **MODULE_B** のパッケージを認識し、**MODULE_B** は **MODULE_A** のパッケージを認識したことを意味しますが、この動作は複雑で競合が発生することがありました。この動作は JBoss EAP 7 では変更になり、**module.xml** ファイルの **resource-root** によって指定されたクラスまたはパッケージは依存関係によって指定されたものよりも優先されるようになりました。これは、**MODULE_A** は **MODULE_A** のパッケージを認識し、**MODULE_B** は **MODULE_B** のパッケージを認識することを意味します。これにより、競合の発生を防ぎ、より適切な動作を提供します。

If you have defined custom modules that include **resource-root** libraries or packages that contain classes that are duplicated in their module dependencies, you might see **ClassCastException**, **LinkageError**, class loading errors, or other changes in behavior when you migrate to JBoss EAP 7. To resolve these issues, you must configure your **module.xml** file to ensure only one version of a class is used. This can be accomplished by using either of the following approaches.

- モジュール依存関係のクラスを複製する **resource-root** を指定しないようにします。
- **imports** および **exports** 要素の **include** および **exclude** サブ要素を使用して **module.xml** ファイルでクラスローディングを制御できます。以下は、指定されたパッケージでクラスを除外する **export** 要素になります。

```
<exports>
  <exclude path="com/mycompany/duplicateclassespath"/>
</exports>
```

既存の動作を保持するには、**filter** 要素を使用して **module.xml** ファイルの依存する **resource-root** から依存パッケージをフィルターする必要があります。これにより、JBoss EAP 6 で見られる odd loop なしで既存の動作を保持できます。以下は、指定のパッケージのクラスをフィルターする **root-resource** の例になります。

```
<resource-root path="mycompany.jar">
  <filter>
    <exclude path="com/mycompany/duplicateclassespath"/>
  </filter>
</resource-root>
```

For more information about modules and class loading, see [Class Loading and Modules](#) in the JBoss EAP Development Guide.

5.18. アプリケーションクラスタリングの変更

5.18.1. 新しいクラスタリング機能の概要

以下のリストは、アプリケーションを JBoss EAP 6 から JBoss EAP 7 へ移行するときに注意する必要がある新しいクラスタリング機能の一部を示しています。

- JBoss EAP 7 introduces a new public API for building singleton services that significantly simplifies the process. For more information, see [Implement an HA Singleton](#) in **Developing EJB Applications** for JBoss EAP.

- A singleton deployment can be configured to deploy and start on only a single node in the cluster at a time. For more information, see [HA Singleton Deployments](#) in the JBoss EAP **Development Guide**.
- You can now define clustered singleton MDBs. For more information, see [Clustered Singleton MDBs](#) in **Developing EJB Applications** for JBoss EAP.
- JBoss EAP 7 includes the Undertow `mod_cluster` implementation. This offers a pure Java load balancing solution that does not require an `httpd` web server. For more information, see [Configuring JBoss EAP as a Front-end Load Balancer](#) in the JBoss EAP **Configuration Guide**.

The remainder of this section describes how clustering changes might impact the migration of your applications to JBoss EAP 7.

5.18.2. Web セッションクラスタリングの変更

JBoss EAP 7 では、新しい Web セッションクラスタリング実装が導入されました。これは、レガシーの JBoss Web サブシステムソースコードに密に結合された以前の実装に代わる実装です。

新しい Web セッションクラスタリング実装は、JBoss プロプライエタリー Web アプリケーションの XML 記述子ファイルである `jboss-web.xml` にアプリケーションが設定される方法に影響します。以下は、このファイルのクラスタリング設定要素のみになります。

```
<jboss-web>
  ...
  <max-active-sessions>...</max-active-sessions>
  ...
  <replication-config>
    <replication-granularity>...</replication-granularity>
    <cache-name>...</cache-name>
  </replication-config>
  ...
</jboss-web>
```

以下の表は、`jboss-web.xml` ファイルの廃止された要素と同様の動作を実現する方法を説明しています。

設定要素	変更の説明
<code><max-active-sessions/></code>	<p>これまでの実装では、アクティブなセッションの数が <code><max-active-sessions/></code> によって指定された値を超えるとセッションの作成に失敗しました。</p> <p>新しい実装では、<code><max-active-sessions/></code> を使用してセッションパッシベーションが有効化されます。セッションの作成によって、アクティブなセッションの数が <code><max-active-sessions/></code> を超える場合、新しいセッションが作成されるよう、セッションマネージャーが認識する最も古いセッションがパッシベートされます。</p>
<code><passivation-config/></code>	この設定要素とサブ要素は JBoss EAP 7 では使用されなくなりました。

設定要素	変更の説明
<use-session-passivation/>	<p>以前の実装では、この属性を使用してパッシベーションが有効化されました。</p> <p>新しい実装では、<max-active-sessions/> に負でない値を指定するとパッシベーションが有効化されます。</p>
<passivation-min-idle-time/>	<p>以前の実装では、パッシベーションの候補となる前にセッションは最小時間アクティブである必要がありました。そのため、パッシベーションが有効であってもセッションの作成に失敗することがありました。</p> <p>新しい実装はこの論理をサポートしないため、サービス拒否 (DoS) 攻撃の発生を防ぎます。</p>
<passivation-max-idle-time/>	<p>以前の実装では、セッションは指定期間アイドル状態であった後にパッシベートされました。</p> <p>新しい実装はレイジー (Lazy) パッシベーションのみをサポートします。イーガー (Eager) パッシベーションはサポートしません。セッションは、<max-active-sessions/> に準拠する必要があるときのみパッシベートされます。</p>
<replication-config/>	新しい実装では複数のサブ要素が非推奨になりました。
<replication-trigger/>	Previously, this element was used to determine when session replication was triggered. The new implementation replaces this configuration option with a single, robust strategy. For more information, see Immutable Session Attributes in the JBoss EAP Development Guide .
<use-jk/>	<p>以前の実装では、<use-jk/> に指定された値に応じて、指定のリクエストを処理するノードの instance-id が jsessionId の末尾に追加され、mod_jk、mod_proxy_balancer、mod_cluster などのロードバランサーによって使用されました。</p> <p>新しい実装では、instance-id が定義されている場合は常に jsessionId の末尾に追加されるようになりました。</p>
<max-unreplicated-interval/>	<p>以前の実装では、この設定オプションは変更されたセッション属性がない場合にセッションのタイムスタンプがレプリケートされないようにする最適化を目的としていました。しかし、セッション属性が変更されたかどうかに関係なく、セッションのアクセスにはキャッシュトランザクション RPC が必要であるため、実際は RPC の実行を防ぐことはできません。</p> <p>新しい実装では、リクエストごとにセッションのタイムスタンプがレプリケートされるようになりました。これにより、フェイルオーバーの後にセッションメタデータが陳腐化されないようにします。</p>
<snapshot-mode/>	Previously, one could configure <snapshot-mode/> as INSTANT or INTERVAL . Infinispan's asynchronous replication makes this configuration option obsolete.

設定要素	変更の説明
<snapshot-interval/>	これは <snapshot-mode>INTERVAL</snapshot-mode> にのみ関係しました。<snapshot-mode/> は廃止されたため、このオプションも廃止されました。
<session-notification-policy/>	<p>以前の実装では、この属性によって指定された値はセッションイベントをトリガーするポリシーを定義しました。</p> <p>新しい実装ではこの動作は仕様に応じて判断されるようになり、設定不可能になりました。</p>

This new implementation also supports write-through cache stores as well as passivation-only cache stores. Typically, a write-through cache store is used in conjunction with an invalidation cache. The web session clustering implementation in JBoss EAP 6 did not operate correctly when used with an invalidation cache.

5.18.3. ステートフルセッション EJB クラスタリングの変更

JBoss EAP 6 では、以下の方法の 1 つでステートフルセッション Bean (SFSB) のクラスタリング動作を有効化する必要がありました。

- セッション Bean に `org.jboss.ejb3.annotation.Clustered` アノテーションを追加。

```
@Stateful
@Clustered
public class MyBean implements MySessionInt {

    public void myMethod() {
        //
    }
}
```

- <clustered> 要素を `jboss-ejb3.xml` ファイルに追加。

```
<c:clustering>
  <ejb-name>DDBasedClusteredSFSB</ejb-name>
  <c:clustered>true</c:clustered>
</c:clustering>
```

JBoss EAP 7 では、クラスタリング動作を有効にする必要がなくなりました。デフォルトでは、HA プロファイルを使用してサーバーが起動された場合は SFSB の状態が自動的にレプリケートされます。

以下の方法の 1 つを使用すると、このデフォルト動作を無効にできます。

- EJB 3.2 仕様に新たに導入された `@Stateful(passivationCapable=false)` を使用して単一のステートフルセッション Bean のデフォルト動作を無効化します。
- サーバー設定の `ejb3` サブシステムの設定で、この動作をグローバルに無効化します。



注記

@**Clustered** アノテーションがアプリケーションから削除されると、このアノテーションは無視され、アプリケーションのデプロイメントには影響しません。

5.18.4. クラスタリングサービスの変更

JBoss EAP 6 では、クラスタリングサービスの API はプライベートモジュールでサポートされませんでした。

JBoss EAP 7 には、アプリケーションによって使用されるパブリッククラスタリングサービス API が導入されました。新しいサービスは、ライトウェイトで簡単にインジェクトできるように設計されています。外部の依存関係は必要ありません。

- 新しい **org.wildfly.clustering.group.Group** インターフェースを使用すると、現在のクラスター状態へアクセスでき、クラスターメンバーシップの変更をリッスンできます。
- 新しい **org.wildfly.clustering.dispatcher.CommandDispatcher** インターフェースを使用すると、すべてのノードまたはノードの選択されたサブセットのクラスターでコードを実行できます。

これらのサービスは、JBoss EAP 5 の **HAPartition**、JBoss EAP 6 の **GroupCommunicationService**、**GroupMembershipNotifier**、および **GroupRpcDispatcher** に代わるサービスです。これらの API は以前のリリースで利用できました。

For more information, see [Public API for Clustering Services](#) in the JBoss EAP **Development Guide**.

5.18.5. クラスタリング HA シングルトンの移行

JBoss EAP 6 では、クラスター全体の HA シングルトンサービスに使用できるパブリック API がありませんでした。プライベートの **org.jboss.as.clustering.singleton.*** クラスを使用した場合、アプリケーションを JBoss EAP 7 に移行するときに新しいパブリックの **org.wildfly.clustering.singleton.*** パッケージを使用するようコードを変更する必要があります。

For more information about how to implement an HA singleton, see [HA Singleton Service](#) in the **Development Guide** for JBoss EAP.

第6章 その他の変更

6.1. JBOSS EAP ネイティブおよび APACHE HTTP SERVER の提供に関する変更

JBoss EAP 7 ネイティブは、これまでのリリースとは異なる方法で提供されます。一部のネイティブは、多くの Red Hat JBoss ミドルウェア製品に共通する補完ソフトウェアセットである、新規の Red Hat JBoss Core Services 製品に同梱されるようになりました。新製品では更新のより迅速な配布と一貫性のある更新が可能になります。JBoss Core Services 製品は、Red Hat カスタマーポータル別の場所からダウンロードできます。

- 以下の表では、リリースごとの提供方法の違いを示しています。

パッケージ	JBoss EAP 6	JBoss EAP 7
メッセージング用 AIO ネイティブ	別個の "ネイティブユーティリティ" ダウンロードで製品とともに提供。	JBoss EAP ディストリビューションに含まれます。他のダウンロードは必要ありません。
Apache HTTP Server	別個の "Apache HTTP Server" ダウンロードで製品とともに提供。	新しい JBoss Core Services 製品とともに提供されます。
mod_cluster、mod_jk、isapi、および nsapi コネクタ	別個の "Webserver コネクタネイティブ" ダウンロードで製品とともに提供。	新しい JBoss Core Services 製品とともに提供されます。
JSVC	別個の "ネイティブユーティリティ" ダウンロードで製品とともに提供。	新しい JBoss Core Services 製品とともに提供されます。
OpenSSL	別個の "ネイティブユーティリティ" ダウンロードで製品とともに提供。	これは JBoss EAP 7 では廃止されました
tcnatives	別個の "ネイティブコンポーネント" ダウンロードで製品とともに提供。	これは JBoss EAP 7 では廃止されました

- 以下の変更点にも注意してください。
 - Red Hat Enterprise Linux RPM チャンネルからの Apache HTTP Server と使用される mod_cluster および mod_jk コネクタのサポートは廃止されました。Red Hat Enterprise Linux RPM チャンネルから Apache HTTP Server を実行し、JBoss EAP 7 サーバーの負荷分散を設定する必要がある場合は、以下の 1 つを行います。
 - JBoss Core Services によって提供される Apache HTTP Server を使用します。
 - You can configure JBoss EAP 7 to act as a front-end load balancer. For more information, see [Configuring JBoss EAP as a Front-end Load Balancer](#) in the JBoss EAP Configuration Guide.
 - You can deploy Apache HTTP Server on a machine that is supported and certified and then run the load balancer on that machine. For the list of supported configurations, see [Overview of HTTP Connectors](#) in the JBoss EAP 7 Configuration Guide.

- HP-UX Web Server Suites からの Apache HTTP Server と使用される `mod_cluster` および `mod_jk` コネクタのサポートは廃止されました。HP-UX Web Server Suites から Apache HTTP Server を実行し、JBoss EAP 7 サーバーの負荷分散を設定する必要がある場合は、以下の 1 つを行います。
 - You can configure JBoss EAP 7 to act as a front-end load balancer. For more information, see [Configuring JBoss EAP as a Front-end Load Balancer](#) in the JBoss EAP **Configuration Guide**.
 - You can deploy Apache HTTP Server on a machine that is supported and certified and then run the load balancer on that machine. For the list of supported configurations, see [Overview of HTTP Connectors](#) in the JBoss EAP **Configuration Guide**.
- You can find more information about [JBoss Core Services](#) in the **Apache HTTP Server Installation Guide**.

6.2. AMAZON EC2 でのデプロイメントの変更

A number of changes have been made to the Amazon Machine Images (AMI) in JBoss EAP 7. This section briefly summarizes some of those changes.

- 非クラスターおよびクラスターの JBoss EAP インスタンスおよびドメインを Amazon EC2 で起動する方法が大幅に変更されました。
- JBoss EAP 6 では JBoss EAP の設定に **User Data:** フィールドが使用されました。JBoss EAP 7 では、**User Data:** フィールドの設定を分析した AMI スクリプトと、インスタンスの起動時に自動的にサーバーを起動した AMI スクリプトが削除されました。
- JBoss EAP のこれまでのリリースでは、Red Hat JBoss Operations Network エージェントがインストールされましたが、JBoss EAP 7 では個別にインストールする必要があります。

For details on deploying JBoss EAP 7 on Amazon EC2, see [Deploying Red Hat JBoss Enterprise Application Platform on Amazon EC2](#).

6.3. CHANGES TO JBOSS EAP SCRIPTS

The **add-user** script behavior has changed in JBoss EAP 7 due to a change in password policy. JBoss EAP 6 had a strict password policy. As a result, the **add-user** script rejected weak passwords that did not satisfy the minimum requirements. In JBoss EAP 7, weak passwords are accepted and a warning is issued. For more information, see [Setting Add-User Utility Password Restrictions](#) in the JBoss EAP **Configuration Guide**.

6.4. REMOVAL OF OSGI SUPPORT

When JBoss EAP 6.0 GA was first released, JBoss OSGi, an implementation of the OSGi specification, was included as a Technology Preview feature. With the release of JBoss EAP 6.1.0, JBoss OSGi was demoted from Technology Preview to Unsupported.

In JBoss EAP 6.1.0, the **configadmin** and **osgi** extension modules and subsystem configuration for a standalone server were moved to a separate **EAP_HOME/standalone/configuration/standalone-osgi.xml** configuration file. Because you should not migrate this unsupported configuration file, the removal of JBoss OSGi support should not impact the migration of a standalone server configuration. If you modified any of the other standalone configuration files to configure **osgi** or **configadmin**, those configurations must be removed.

For a managed domain, the **osgi** extension and subsystem configuration were removed from the **EAP_HOME/domain/configuration/domain.xml** file in the JBoss EAP 6.1.0 release. However, the **configadmin** module extension and subsystem configuration remain in the **EAP_HOME/domain/configuration/domain.xml** file. This configuration is no longer supported in JBoss EAP 7 and must be removed.

第7章 MIGRATING FROM OLDER RELEASES OF JBOSS EAP

7.1. MIGRATING FROM JBOSS EAP 5 TO JBOSS EAP 7

This guide focuses on the changes that are required to successfully run and deploy JBoss EAP 6 applications on JBoss EAP 7. If you plan to migrate your applications directly from JBoss EAP 5 to JBoss EAP 7, there are a number of resources available to help you plan and execute your migration. We suggest you take the following approach.

1. See [Summary of Changes Made to Each Release](#) in this guide for a quick, high-level overview of the changes made to each release of JBoss EAP.
2. Read through the JBoss EAP 6 [Migration Guide](#) and this guide to become familiar with the contents of each one.
3. Use the [JBoss EAP 5 Component Upgrade Reference](#) as a quick reference to migration information about specific components and features.
4. The Windup rule-based migration tool continues to add rules to help you migrate directly from JBoss EAP 5 to JBoss EAP 7. You can use this tool to analyze your application and to generate detailed reports about the changes needed to migrate to JBoss EAP 7. For more information, see [Use Windup to Analyze Applications for Migration](#).
5. The [Customer Portal Knowledgebase](#) currently contains articles and solutions to help with migration from JBoss EAP 5 to JBoss EAP 6. There are plans in place to add additional content for migration from JBoss EAP 5 to JBoss EAP 7 over time.

7.2. SUMMARY OF CHANGES MADE TO EACH RELEASE

Before you plan your migration, you should be aware of the changes that were made to JBoss EAP 6 and JBoss EAP 7.

The [JBoss EAP 6 Migration Guide](#) covers changes that were made between JBoss EAP 5 and JBoss EAP 6. The following is a condensed list of the most significant changes made in JBoss EAP 6.

- Implemented a new architecture built on the Modular Service Container
- Was a certified implementation of the Java Enterprise Edition 6 specification
- Introduced domain management, new deployment configuration, and a new file directory structure and scripts
- Standardized on new portable JNDI namespaces

See [Review What's New and Different in JBoss EAP 6](#) in the JBoss EAP 6 [Migration Guide](#) for a detailed list of changes made in that release.

JBoss EAP 7 is built on the same modular structure as JBoss EAP 6 and includes the same domain management, deployment configuration, file directory structure, and scripts. It also still uses the same standardized JNDI namespaces. However, JBoss EAP 7 introduces the following changes.

- Adds support for the Java Enterprise Edition 7 specification
- Replaces the web server with Undertow
- Replaces the JacORB IIOP implementation with a downstream branch of the OpenJDK ORB

- Includes Apache ActiveMQ Artemis as the new messaging provider
- Removes the **cmp**, **jaxr**, and **threads** subsystems
- Removes support for EJB entity beans

For a more complete list of changes, see [Review What's New in JBoss EAP 7](#)

7.3. REVIEW THE CONTENT IN THE MIGRATION GUIDES

Review the entire contents of the Migration Guide for each release to become aware of the features that were added or deprecated, and to understand the server configuration and the application changes required to run existing applications for that release.

Because the underlying architecture was not changed between JBoss EAP 6 and JBoss EAP 7, many of the changes documented in the JBoss EAP 6 [Migration Guide](#) still apply. For example, changes documented under [Changes Required by Most Applications](#) are related to the underlying architectural changes made in JBoss EAP 6, which still apply to this release. The change to the new modular class loading system is significant and impacts the packaging and dependencies of almost every JBoss EAP 5 application. Many of the changes listed under [Changes Dependent on Your Application Architecture and Components](#) are also still valid. However, because JBoss EAP 7 replaced the web server, ORB, and messaging provider, removed the **cmp**, **threads**, and **jaxr** subsystems, and removed support for EJB entity beans, you must consult this guide for any changes related to those component areas. Pay particular attention to the [Server Configuration Changes](#) and [Application Migration Changes](#) detailed in this guide before you begin.

7.4. JBOSS EAP 5 COMPONENT UPGRADE REFERENCE

Use the following table to find information about how to migrate a particular feature or component from JBoss EAP 5 to JBoss EAP 7.

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
Application Packaging and Class Loading	<p>In JBoss EAP 6, the previous hierarchical class loading structure was replaced with a modular architecture based on JBoss Modules. Application packaging also changed due to the new modular class loading structure. This architecture is still used in JBoss EAP 7. For information about the new modular architecture, see the following chapter in the JBoss EAP 7 Development Guide.</p> <ul style="list-style-type: none"> • Class Loading and Modules <p>For information about how to update and repackage applications for the new modular architecture, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • Class Loading Changes

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
Application Configuration Files	<p>Due to the changes in JBoss EAP 6 to use modular class loading, you might need to create or modify one or more application configuration files to add dependencies or to prevent automatic dependencies from loading. This has not changed in JBoss EAP 7. For details, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • Configuration File Changes
Caching and Infinispan	<p>JBoss Cache was replaced by Infinispan for internal use by the server only in JBoss EAP 6. See the following sections in the JBoss EAP 6 Migration Guide for information about how to replace JBoss Cache in application code.</p> <ul style="list-style-type: none"> • Cache Changes <p>Infinispan caching strategy and configuration changes for JBoss EAP 7 are documented in the following section of this guide.</p> <ul style="list-style-type: none"> • Infinispan Server Configuration Changes
Data Sources and Resource Adapters	<p>JBoss EAP 6 consolidated configuration of data sources and resource adapters into mainly one file and this is still true in JBoss EAP 7. See the following section in the JBoss EAP 6 Migration Guide for more information.</p> <ul style="list-style-type: none"> • Datasource and Resource Adapter Configuration Changes <p>In addition, the ability to configure a generic JMS resource adapter to connect to a JMS provider is no longer supported in JBoss EAP 7. See the following section in this guide.</p> <ul style="list-style-type: none"> • Review The List of Deprecated and Unsupported Features
Directory Structure, Scripts, and Deployment Configuration	<p>In JBoss EAP 6, the directory structure, scripts, and deployment configuration changed. These changes are still valid in JBoss EAP 7. See the following section of the JBoss EAP 6 Migration Guide for more information.</p> <ul style="list-style-type: none"> • Review What's New and Different in JBoss EAP 6

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
EJB	<p>The Java EE 7 specification made EJB 2.x and earlier features optional, so it is strongly recommended that you rewrite your application code to use the EJB 3.x specification and JPA. For information about deprecated features and changes required to run EJB 2.x, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • EJB 2.x and Earlier Changes <p>In JBoss EAP 6, stateful EJB cache and stateless session bean pool size is configured in the ejb3 subsystem of the server configuration file. The jboss-ejb3.xml deployment descriptor replaces the jboss.xml deployment descriptor file. For more information about these changes, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • EJB Changes <p>The default remote connector and port has changed in JBoss EAP 7. For more information about this and server configuration changes, see the following sections in this guide.</p> <ul style="list-style-type: none"> • EJB Server Configuration Changes • Migrate EJB Client Code <p>EJB entity beans are not supported in JBoss EAP 7. For information about how to migrate entity beans to JPA, see the following section in this guide.</p> <ul style="list-style-type: none"> • Migrate Entity Beans to JPA
Hibernate および JPA	<p>In JBoss EAP 6, Hibernate was updated from version 3 to version 4. This version of JBoss EAP also implemented the JPA 2.0 specification and changes were made to JPA persistence properties. For information about how to modify your application for these changes, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • Hibernate and JPA Changes <p>JBoss EAP 7 implements JPA 2.1 and includes Hibernate 5. It also updates Hibernate Search from version 4.6.x to version 5.5.x. Other changes include removal of support for EJB entity beans and additional updates to JPA persistence properties. For information about how these changes impact your applications, see the following sections in this guide.</p> <ul style="list-style-type: none"> • Hibernate and JPA Migration Changes • Hibernate Search Changes • Migrate Entity Beans to JPA • JPA Persistence Property Changes

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
JAX-RS and RESTEasy	<p>JBoss EAP 6 bundled RESTEasy 2, which automatically configured RESTEasy and required changes in application configuration. See the following section in the JBoss EAP 6 Migration Guide for information.</p> <ul style="list-style-type: none"> • JAX-RS and RESTEasy Changes <p>JBoss EAP 7 includes RESTEasy 3 and many classes have been deprecated. The version of Jackson changed from version 1.9.9 to version 2.6.3 or greater. For details about these changes, see the following section in this guide.</p> <ul style="list-style-type: none"> • JAX-RS and RESTEasy Application Changes
JBoss AOP	<p>JBoss AOP (Aspect Oriented Programming) was removed in JBoss EAP 6. For information about how to refactor applications that use JBoss AOP, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • JBoss AOP Changes
JGroups and Clustering	<p>The way you enable clustering and specify bind addresses changed in JBoss EAP 6. See the following section in the JBoss EAP 6 Migration Guide for more information.</p> <ul style="list-style-type: none"> • Clustering Changes <p>In JBoss EAP 7, JGroups now defaults to using a private network interface instead of a public network interface and also introduces <channel> elements to the jgroups subsystem. JBoss EAP 7 also includes the Undertow <code>mod_cluster</code> implementation, introduces a new API for building singleton services, and other new clustering features. These changes are documented in the following sections of this guide.</p> <ul style="list-style-type: none"> • JGroups Server Configuration Changes • Application Clustering Changes
JNDI	<p>JBoss EAP 6 implemented a new standardized global JNDI namespace and a series of related namespaces that map to the various scopes of a Java EE application. See the following section of the JBoss EAP 6 Migration Guide for information about application changes needed to use the new JNDI namespace rules.</p> <ul style="list-style-type: none"> • JNDI Changes

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
JSF	<p>JBoss EAP 6 included JSF 2.0 and allowed you to configure your application to use an older version. This is no longer possible in JBoss EAP 7, which now includes JSF 2.2. See the following section in this guide for more information.</p> <ul style="list-style-type: none"> • JavaServer Faces (JSF) Code Changes
ロギング	<p>JBoss EAP 6 introduced a new JBoss Logging framework that is still used in JBoss EAP 7. Applications that use third-party logging frameworks might be impacted by the modular class loading changes. Review the following section in the JBoss EAP 6 Migration Guide for information about these changes.</p> <ul style="list-style-type: none"> • Logging Changes <p>In JBoss EAP 7, annotations in the org.jboss.logging package are now deprecated, which impacts source code and Maven GAVs (groupId:artifactId:version). The prefixes for all log messages were also changed. For more information about these changes, see the following sections in this guide.</p> <ul style="list-style-type: none"> • JBoss Logging Changes • Logging Message Prefix Changes
Messaging and JMS	<p>In JBoss EAP 6, HornetQ replaced JBoss Messaging as the default JMS implementation. Then in JBoss EAP 7, ActiveMQ Artemis replaced HornetQ as the built-in messaging provider.</p> <p>The best approach to migrating your messaging configuration is to start with the JBoss EAP 7 default server configuration and use the following guide to apply your current messaging configuration changes.</p> <ul style="list-style-type: none"> • Configuring Messaging for JBoss EAP 7 <p>If you want to understand the changes required to move from JBoss Messaging to HornetQ, review the following section of the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • HornetQ Changes <p>Then review the following information about how to migrate the HornetQ configuration and related messaging data in this guide.</p> <ul style="list-style-type: none"> • Messaging Server Configuration Changes • Messaging Application Changes

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
ORB	<p>In JBoss EAP 6, JacORB configuration was moved from the EAP_HOME/server/production/conf/jacorb.properties file to the server configuration file. JBoss EAP 7 then replaced the JacORB IIOP implementation with a downstream branch of the OpenJDK ORB.</p> <p>The best approach to migrating your ORB configuration is to start with the JBoss EAP 7 default server configuration and use the following section in the JBoss EAP 7 Configuration Guide to apply the your current ORB configuration changes.</p> <ul style="list-style-type: none"> • ORB Configuration
Remote Invocation	<p>A new EJB client API was introduced in JBoss EAP 6 for remote invocations; however, if you preferred not to rewrite your application code to use the new API, you could modify your existing code to use the ejb:BEAN_REFERENCE for remote access to EJBs. See the following section in the JBoss EAP 6 Migration Guide for more information.</p> <ul style="list-style-type: none"> • Remote Invocation Changes <p>In JBoss EAP 7, the default connector and default remote connection port changed. For more information, see the following sections in this guide.</p> <ul style="list-style-type: none"> • Update the Remote URL Connector and Port • Update External Clients • Migrate EJB Client Code
Seam 2.x	<p>While official support for Seam 2.2 applications was dropped in JBoss EAP 6, it was still possible to configure dependencies for JSF 1.2 and Hibernate 3 to allow Seam 2.2 applications to run on that release. JBoss EAP 7, which now includes JSF 2.2 and Hibernate 5, does not support Seam 2.2 or Seam 2.3 due to end of life of Red Hat JBoss Web Framework Kit. It is recommended that you rewrite your Seam components using Weld CDI beans.</p>
セキュリティー	<p>Security updates in JBoss EAP 6 included changes to security domain names and changes to how to configure security for basic authentication. The LDAP security realm configuration was moved to the server configuration file. See the following sections in the JBoss EAP 6 Migration Guide for more information.</p> <ul style="list-style-type: none"> • Security Changes • LDAP Security Realm Changes <p>Updates that impact security in JBoss EAP 7 include server configuration changes and application changes. Information can be found in the following sections of this guide.</p> <ul style="list-style-type: none"> • Security Server Configuration Changes • Security Application Changes

JBoss EAP 5 Feature or Component	Summary of Changes and Where to Find Migration Information
Spring Applications	<p>Spring 4.2.x is the earliest stable Spring version supported by JBoss EAP 7. For information about Apache CXF Spring web services and Spring RESTEasy integration changes, see the following sections in this guide.</p> <ul style="list-style-type: none"> • Apache CXF Spring Web Services Changes • Spring RESTEasy Integration Changes
トランザクション	<p>JBoss EAP 6 consolidated transaction configuration and moved it to the server configuration file. Other updates included changes to JTA node identifier settings and how to enable JTS. For details, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • JTS and JTA Changes <p>Some Transaction Manager configuration attributes that were available in the transactions subsystem in JBoss EAP 6 have changed in JBoss EAP 7. For more information, see the following section in this guide.</p> <ul style="list-style-type: none"> • Migrate Transactions Subsystem Changes
バルブ	<p>Undertow replaced JBoss Web in JBoss EAP 7 and valves are no longer supported. See the following sections in this guide.</p> <ul style="list-style-type: none"> • Migrate Global Valves • Migrate Custom Application Valves • Migrate Authenticator Valves
Web Services	<p>JBoss EAP 6 included JBossWS 4. For information about the changes required by that version update, see the following section in the JBoss EAP 6 Migration Guide.</p> <ul style="list-style-type: none"> • Web Services Changes <p>JBoss EAP 7 introduced JBossWS 5. See the following section in this guide for required updates.</p> <ul style="list-style-type: none"> • Web Services Applications Changes

付録A リファレンス資料

A.1. JACORB サブシステム移行操作の警告

The **migrate** operation is not able to process all resources and attributes. The following table lists some of the warnings you might see when you run either the **migrate** or **describe-migration** operation for the **jacorb** subsystem.



注記

migrate 操作の出力に「Could not migrate」または「Can not migrate」エントリが記録された場合、サーバー設定の移行は正常に完了したにも関わらず、すべての要素および属性を自動的に移行できなかったことを表します。「migration-warnings」の提案に従ってこれらの設定を変更する必要があります。

警告メッセージ	メッセージの意味 / 修正方法
The iiop migration can be performed when the server is in admin-only mode (iiop の移行はサーバーが admin-only モードであるときに実行できます)	<p>migrate 操作を実行するには、--admin-only パラメーターをサーバー起動コマンドに追加し、サーバーを admin-only モードで起動する必要があります。</p> <pre>\$ EAP_HOME/bin/standalone.sh --admin-only</pre>
Properties X cannot be emulated using OpenJDK ORB and are not supported (プロパティー X は OpenJDK ORB を使用してエミュレートできず、サポートされていません)	<p>指定プロパティーの設定はサポートされず、新しい iiop-openjdk サブシステム設定には含まれていません。以前のリリースの JBoss EAP でこのプロパティーによって示された動作は移行されないため、管理者は JBoss EAP 7 の新しい iiop-openjdk サブシステムがこの動作を示さずに正しく操作できることを検証する必要があります。</p> <p>サポートされないプロパティーには以下が含まれます: cache-poa-names、cache-typecodes、chunk-custom-rmi-valuetypes、client-timeout、comet、indirection-encoding-disable、iona、lax-boolean-encoding、max-managed-buf-size、max-server-connections、max-threads、outbuf-cache-timeout、outbuf-size、queue-max、queue-min、poa-monitoring、print-version、retries、retry-interval、queue-wait、server-timeout、strict-check-on-tc-creation、use-bom、use-imr。</p>

警告メッセージ	メッセージの意味 / 修正方法
<p>The properties X use expressions. Configuration properties that are used to resolve those expressions should be transformed manually to the new iiop-openjdk subsystem format (プロパティ X は式を使用します。これらの式を解決するために使用される設定プロパティは新しい iiop-openjdk サブシステム形式に手動で変換する必要があります)</p>	<p>式を使用するプロパティは管理者が手作業で設定する必要があります。</p> <p>たとえば、JBoss EAP 6 の jacorb サブシステムは giop-minor-version を定義しました。JBoss EAP 7 の iiop-openjdk サブシステムは giop-version プロパティを定義します。jacorb サブシステムのマイナーバージョン属性が #{iiop-giop-minor-version} に設定され、システムプロパティが -Diiop-giop-minor-version=1 として standalone.conf ファイルに設定されている場合、管理者は migrate 操作の後にシステムプロパティの値を 1.1 に変更し、新しいサブシステムが適切に設定されるようにする必要があります。</p>
<p>Can not migrate: the new iiop-openjdk subsystem is already defined (移行できません: 新しい iiop-openjdk サブシステムはすでに定義されています)</p>	<p>メッセージには説明が含まれています。</p>

A.2. MESSAGING サブシステム移行操作の警告

The **migrate** operation is not able to process all resources and attributes. The following table lists some of the warnings you might see when you run either the **migrate** or **describe-migration** operation for the **messaging** subsystem.



注記

migrate 操作の出力に「Could not migrate」または「Can not migrate」エントリが記録された場合、サーバー設定の移行は正常に完了したにもかかわらず、すべての要素および属性を自動的に移行できなかったことを表します。「migration-warnings」の提案に従ってこれらの設定を変更する必要があります。

警告メッセージ	メッセージの意味 / 修正方法
<p>The migrate operation can not be performed: the server must be in admin-only mode (migrate 操作は実行できません。サーバーは admin-only モードである必要があります)</p>	<p>migrate 操作を実行するには、--admin-only パラメーターをサーバー起動コマンドに追加し、サーバーを admin-only モードで起動する必要があります。</p> <pre>\$ EAP_HOME/bin/standalone.sh --admin-only</pre>

警告メッセージ	メッセージの意味 / 修正方法
<p>Can not migrate attribute local-bind-address from resource X. Use instead the socket-binding attribute to configure this broadcast-group. (リソース X から属性 local-bind-address を移行できません。代わりに socket-binding 属性を使用してこの broadcast-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute local-bind-port from resource X. Use instead the socket-binding attribute to configure this broadcast-group. (リソース X から属性 local-bind-port を移行できません。代わりに socket-binding 属性を使用してこの broadcast-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute group-address from resource X. Use instead the socket-binding attribute to configure this broadcast-group. (リソース X から属性 group-address を移行できません。代わりに socket-binding 属性を使用してこの broadcast-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute group-port from resource X. Use instead the socket-binding attribute to configure this broadcast-group. (リソース X から属性 group-port を移行できません。代わりに socket-binding 属性を使用してこの broadcast-group を設定してください。)</p>	<p>broadcast-group リソースは local-bind-address、local-bind-port、group-address、または group-port 属性を許可しないようになりました。このリソースは socket-binding 属性のみを許可します。この警告は、リソース X にサポートされない属性があることを通知します。リソースの socket-binding 属性を手作業で設定し、定義された socket-binding リソースに対応するようにする必要があります。</p>
<p>Classes providing the X are discarded during the migration. To use them in the new messaging-activemq subsystem, you will have to extend the Artemis-based Interceptor. (移行中に X を提供するクラスは破棄されます。新しい messaging-activemq サブシステムでこれらを使用するには、Artemis ベースのインターセプターを拡張する必要があります。)</p>	<p>JBoss EAP 7 のメッセージングインターセプターのサポートは大幅に異なります。以前のバージョンのサブシステムに設定されたインターセプターはすべて移行中に破棄されます。詳細は メッセージングインターセプターの移行 を参照してください。</p>

警告メッセージ	メッセージの意味 / 修正方法
<p>Can not migrate the HA configuration of X. Its shared-store and backup attributes holds expressions and it is not possible to determine unambiguously how to create the corresponding ha-policy for the messaging-activemq's server. (X の HA 設定は移行できません。この shared-store および backup 属性は式を保持し、messaging-activemq のサーバーの対応する ha-policy を作成する方法を明確に決定することができません。)</p>	<p>これは、hornetq-server X の shared-store または backup 属性に $\\${xxx}$ などの式が含まれ、移行操作が具体的な式に解決できなかったことを意味します。値は破棄され、messaging-activemq の ha-policy を手作業で更新する必要があります。</p>
<p>Can not migrate attribute local-bind-address from resource X. Use instead the socket-binding attribute to configure this discovery-group. (リソース X から属性 local-bind-address を移行できません。代わりに socket-binding 属性を使用してこの discovery-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute local-bind-port from resource X. Use instead the socket-binding attribute to configure this discovery-group. (リソース X から属性 local-bind-port を移行できません。代わりに socket-binding 属性を使用してこの discovery-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute group-address from resource X. Use instead the socket-binding attribute to configure this discovery-group. (リソース X から属性 group-address を移行できません。代わりに socket-binding 属性を使用してこの discovery-group を設定してください。)</p>	<p>メッセージには説明が含まれています。</p>
<p>Can not migrate attribute group-port from resource X. Use instead the socket-binding attribute to configure this discovery-group. (リソース X から属性 group-port を移行できません。代わりに socket-binding 属性を使用してこの discovery-group を設定してください。)</p>	<p>discovery-group リソースは local-bind-address、local-bind-port、group-address、または group-port 属性を許可しないようになりました。socket-binding のみを許可します。この警告は、リソース X にサポートされない属性があることを通知します。リソースの socket-binding 属性を手作業で設定し、定義された socket-binding リソースに対応するようになる必要があります。</p>

警告メッセージ	メッセージの意味 / 修正方法
<p>Can not create a legacy-connection-factory based on connection-factory X. It uses a HornetQ in-vm connector that is not compatible with Artemis in-vm connector (connection-factory X を基にして legacy-connection-factory を作成できません。これは Artemis in-vm コネクタと互換性のない HornetQ in-vm コネクタを使用します。)</p>	<p>JBoss EAP 6 クライアントが JBoss EAP 7 に接続できるようにするため、レガシーの HornetQ リモート connection-factory リソースは legacy-connection-factory リソースに移行されました。しかし、legacy-connection-factory リソースは connection-factory がリモートコネクタを使用しているときのみ作成されません。in-vm クライアントは JBoss EAP 6 ではなく JBoss EAP 7 をベースにしているため、in-vm を使用する connection-factory は移行されません。この警告は、in-vm connection-factory が移行されなかったことを通知します。</p>
<p>Can not migrate attribute X from resource Y. The attribute uses an expression that can be resolved differently depending on system properties. After migration, this attribute must be added back with an actual value instead of the expression. (リソース Y から属性 X を移行できません。属性はシステムプロパティに応じて異なって解決される式を使用します。移行後、式の代わりに実際の値を用いてこの属性を戻す必要があります。)</p>	<p>この警告は、移行処理中に属性 X を具体的な値に解決できないときに表示されます。値は破棄され、属性を手作業で移行する必要があります。これは以下の場合に発生します。</p> <ul style="list-style-type: none"> • cluster-connection forward-when-no-consumers: このブール値属性は、列挙で OFF、STRICT、または ON_DEMAND を値として持つ message-load-balancing-type 属性に置き換えられました。 • broadcast-group および discovery-group の jgroups-stack および jgroups-channel 属性 これらの属性は他のリソースを参照し、JBoss EAP 7 ではこれらの式が許可されないようになりました。
<p>Can not migrate attribute X from resource Y. This attribute is not supported by the new messaging-activemq subsystem. (リソース Y から属性 X を移行できません。この属性は新しい messaging-activemq サブシステムによってサポートされません。)</p>	<p>一部の属性は新しい messaging-activemq サブシステムではサポート対象外となり、破棄されます。</p> <ul style="list-style-type: none"> • hornetq-server の failback-delay • http-connector の use-nio 属性 • http-acceptor の use-nio 属性 • remote-connector の use-nio 属性 • remote-acceptor の use-nio 属性

警告メッセージ	メッセージの意味 / 修正方法
<p>Can not migrate attribute failback-delay from resource X. Artemis detects failback deterministically and it no longer requires to specify a delay for failback to occur. (リソース X から 属性 failback-delay を移行できません。フェイルバックは Artemis によって確定的に検出され、フェイルバックの遅延を指定する必要がなくなりました。)</p>	<p>メッセージには説明が含まれています。</p>

非推奨の **broadcast-group** または **discovery-group** 属性の置き換え

非推奨の **broadcast-group** または **discovery-group** 属性を **socket-binding** 属性に置き換えるよう通知された場合、管理 CLI を使用して **socket-binding** 属性を追加できます。

以下の例では、**messaging** サブシステムに以下の **discovery-group** 設定が含まれるスタンドアロンサーバーを移行することを前提とします。

```
<discovery-groups>
  <discovery-group name="my-discovery-group">
    <group-address>224.0.1.105</group-address>
    <group-port>56789</group-port>
  </discovery-group>
</discovery-groups>
```

messaging サブシステムに対して **migrate** 操作を実行すると、以下の出力および警告が表示されます。

```
[standalone@localhost:9999 /] /subsystem=messaging:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYMSG0084: Can not migrate attribute group-address from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"discovery-group\\" => \\"my-discovery-group\\" )
]. Use instead the socket-binding attribute to configure this discovery-
group.",
    "WFLYMSG0084: Can not migrate attribute group-port from resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"discovery-group\\" => \\"my-discovery-group\\" )
]. Use instead the socket-binding attribute to configure this discovery-
group."
  ]}
}
```

migrate 操作によって、"my-discovery-group" という名前の **discovery-group** が新しい **messaging-activemq** サブシステムに作成され、以下のように設定されます。

```
<discovery-group name="my-discovery-group"/>
```

ここで、以下の管理 CLI コマンドを使用して、"my-discovery-group-socket-binding" という名前のサーバー設定ファイルに **socket-binding** 要素を作成する必要があります。

```
/socket-binding-group=standard-sockets/socket-binding=my-discovery-group-socket-binding:add(multicast-address=224.0.1.105, multicast-port=56789)
```

次に、以下の管理 CLI コマンドを使用して、新たに作成された **socket-binding** をサーバー設定ファイルにある **messaging-activemq** サブシステムの "my-discovery-group" という名前の **discovery-group** に追加します。

```
/subsystem=messaging-activemq/server=default/discovery-group=my-discovery-group:write-attribute(name=socket-binding,value=my-discovery-group-socket-binding)
```

これらのコマンドによって、サーバー設定ファイルに以下の XML が作成されます。

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">
  <server name="default">
    ...
    <discovery-group name="my-discovery-group" socket-binding="my-discovery-group-socket-binding"/>
    ...
  </server>
</subsystem>
...
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">
  ...
  <socket-binding name="my-discovery-group-socket-binding" multicast-address="224.0.1.105" multicast-port="56789"/>
  ...
</socket-binding-group>
```

A.3. WEB サブシステム移行操作の警告

The **migrate** operation is not able to process all resources and attributes. The following table lists some of the warnings you might see when you run either the **migrate** or **describe-migration** operation for the **web** subsystem.



注記

migrate 操作の出力に「Could not migrate」または「Can not migrate」エントリーが記録された場合、サーバー設定の移行は正常に完了したにも関わらず、すべての要素および属性を自動的に移行できなかったことを表します。「migration-warnings」の提案に従ってこれらの設定を変更する必要があります。

警告メッセージ

メッセージの意味 / 修正方法

警告メッセージ	メッセージの意味 / 修正方法
<p>Migrate operation only allowed in admin only mode (移行操作は管理専用モードのみで許可されます)</p>	<p>migrate 操作を実行するには、--admin-only パラメーターをサーバー起動コマンドに追加し、サーバーを admin-only モードで起動する必要があります。</p> <pre>\$ EAP_HOME/bin/standalone.sh --admin-only</pre>
<p>Could not migrate resource X (リソース X を移行できませんでした)</p>	<p>以前のリリースの JBoss EAP でこのリソースによって示された動作は移行されませんでした。管理者は JBoss EAP 7 の新しい undertow サブシステムがこの動作を示さずに正しく操作できるかどうか、またはこの動作を手作業で移行する必要があるかどうかを検証する必要があります。</p>
<p>Could not migrate attribute X from resource Y. (リソース Y から属性 X を移行できません。)</p>	<p>以前のリリースの JBoss EAP でこのリソース属性によって示された動作は移行されませんでした。管理者は JBoss EAP 7 の新しい undertow サブシステムがこの動作を示さずに正しく操作できるかどうか、またはこの動作を手作業で移行する必要があるかどうかを検証する必要があります。</p> <p>See Web Subsystem Migration Operation Attribute Warnings for the list of attributes that are not migrated.</p>
<p>Could not migrate SSL connector as no SSL config is defined (定義された SSL 設定がないため SSL コネクターを移行できませんでした)</p>	<p>メッセージには説明が含まれています。</p>
<p>Could not migrate verify-client attribute X to the Undertow equivalent (verify-client 属性 X を Undertow の同等の属性に移行できませんでした)</p>	<p>メッセージには説明が含まれています。</p>
<p>Could not migrate verify-client expression X (verify-client 式 X を移行できませんでした)</p>	<p>メッセージには説明が含まれています。</p>

警告メッセージ	メッセージの意味 / 修正方法
<p>Could not migrate valve X (バルブ X を移行できませんでした)</p>	<p>以前のリリースの JBoss EAP でこのバルブによって示された動作は移行されませんでした。管理者は JBoss EAP 7 の新しい undertow サブシステムがこの動作を示さずに正しく操作できるかどうか、またはこの動作を手作業で移行する必要があるかどうかを検証する必要があります。</p> <p>This warning can occur for the following valves:</p> <ul style="list-style-type: none">● org.apache.catalina.valves.RemoteAddrValve 許可または拒否される値が 1 つ以上必要です。● org.apache.catalina.valves.RemoteHostValve 許可または拒否される値が 1 つ以上必要です。● org.apache.catalina.authenticator.BasicAuthenticator● org.apache.catalina.authenticator.DigestAuthenticator● org.apache.catalina.authenticator.FormAuthenticator● org.apache.catalina.authenticator.SSLAuthenticator● org.apache.catalina.authenticator.SpnegoAuthenticator● カスタムバルブ

警告メッセージ	メッセージの意味 / 修正方法
<p>Could not migrate attribute X from valve Y (バルブ Y から属性 X を移行できませんでした)</p>	<p>The behavior exhibited by this valve attribute in the previous release of JBoss EAP was not migrated. The administrator must verify if the new undertow subsystem in JBoss EAP 7 is able to operate correctly without that behavior or whether the behavior must be migrated manually. This warning can occur for the following valve attributes:</p> <ul style="list-style-type: none"> • org.apache.catalina.valves.AccessLogValve <ul style="list-style-type: none"> ◦ resolveHosts ◦ fileDateFormat ◦ renameOnRotate ◦ encoding ◦ locale ◦ requestAttributesEnabled ◦ buffered • org.apache.catalina.valves.ExtendedAccessLogValve <ul style="list-style-type: none"> ◦ resolveHosts ◦ fileDateFormat ◦ renameOnRotate ◦ encoding ◦ locale ◦ requestAttributesEnabled ◦ buffered • org.apache.catalina.valves.RemoteIpValve <ul style="list-style-type: none"> ◦ httpServerPort ◦ httpsServerPort ◦ remoteIpHeader 定義されていても "x-forwarded-for" に設定されていない場合 ◦ protocolHeader 定義されていても "x-forwarded-proto" に設定されていない場合

Web Subsystem Migration Operation Attribute Warnings

The **migrate** operation is not able to process all JBoss Web attributes. See the following reference tables for information about how to migrate the unprocessed attributes manually.

Web SSL Connector Attributes

The following attributes were used in JBoss EAP 6 to configure the SSL connector. OpenSSL native libraries are not supported in JBoss EAP 7 so there are no equivalent settings.

Attribute	説明	Undertow Equivalent
ca-revocation-url	The file or URL that contains the revocation list.	No equivalent in Undertow.
certificate-file	When using OpenSSL encryption, the path to the file containing the server certificate.	No equivalent in Undertow.
ssl-protocol	The SSL protocol string.	No equivalent in Undertow.
verify-depth	The maximum number of intermediate certificate issuers checked before deciding that the clients do not have a valid certificate.	No equivalent in Undertow.

Web Static Resource Attributes

The following **static-resources** element attributes were used to describe how static resources were handled by the **DefaultServlet** or by the **WebdavServlet**. There are no equivalents for these attributes because WebDAV is not supported by Undertow. For more information, see <https://issues.jboss.org/browse/JBEAP-1036>.

Attribute	説明	Undertow Equivalent
disabled	Enable the default Servlet mapping.	No equivalent setting in Undertow.
file-encoding	File encoding to be used when reading static files.	No equivalent setting in Undertow.
max-depth	Maximum recursion for PROPFIND .	This is a WebDAV setting and WebDAV is not supported by Undertow.
read-only	Allow write HTTP methods (PUT, DELETE).	This is a WebDAV setting and WebDAV is not supported by Undertow.
secret	Secret for WebDAV locking operations.	This is a WebDAV setting and WebDAV is not supported by Undertow.

Attribute	説明	Undertow Equivalent
sendfile	Enable sendfile if possible, for files bigger than the specified byte size.	This is set to a sensible default value in Undertow and is not configurable.
webdav	Enable WebDAV functionality.	WebDAV is not supported by Undertow.

Web SSO Resource Attributes

SSO is handled differently than in the previous release and there are no equivalent attribute settings in JBoss EAP 7.

JBoss Web Attribute	説明	Undertow Equivalent
cache-container	Name of the cache container to use for clustered SSO.	This setting is no longer needed in Undertow. This works by default across a distributed clustered environment.
cache-name	Name of the cache to use for clustered SSO.	This setting is no longer needed in Undertow. This works by default across a distributed clustered environment.
reauthenticate	Whether each request should cause a reauthentication.	There is no equivalent setting in Undertow, which behaves similarly to the reauthenticate=true setting in JBoss EAP 6. While reauthenticate=false could possibly improve performance, it could also create security issues.

Web Access Log Attributes

JBoss Web Attribute	説明	Undertow Equivalent
resolve-hosts	Whether to enable resolving hosts for access logging.	Use the setting on the connector to accomplish the same behavior.

Web Connector Attributes

JBoss Web Attribute	説明	Undertow Equivalent
executor	The name of the executor that should be used to process the threads of this connector.	You now reference a worker that is defined in the io subsystem. See Migrate the Threads Subsystem Configuration for more information.

JBoss Web Attribute	説明	Undertow Equivalent
proxy-binding	The socket binding to define the host and port that is used when sending a redirect.	There is no direct equivalent. See https-listener Attributes in the JBoss EAP Configuration Guide for available configuration options.
redirect-port	The port for redirection to a secure connector.	This attribute was deprecated in JBoss EAP 6 and replaced with redirect-binding . Undertow provides the redirect-socket attribute on the http-listener element, which is a replacement for redirect-binding . See https-listener Attributes in the JBoss EAP Configuration Guide for more information.

A.4. MIGRATE JBOSS WEB SYSTEM PROPERTIES REFERENCE

This reference describes how to map system properties previously used for JBoss Web configuration to the equivalent configuration for Undertow in JBoss EAP 7.

- [Map Servlet Container and Connectors System Properties](#)
- [Map EL System Properties](#)
- [Map JSP System Properties](#)
- [Map Security System Properties](#)

表A.1 Map Servlet Container and Connectors System Properties

JBoss EAP 6 System Property	description
	Equivalent in JBoss EAP 7
jvmRoute	Provides a default value for the jvmRoute attribute. It does not override the automatically generated value when using the standalone-ha.xml configuration file. It supports reload . Management CLI command: <pre> /subsystem=undertow:write-attribute(name=instance-id,value=VALUE) </pre>
org.apache.tomcat.util.buf.StringCache.byte.enabled	If true , the String cache is enabled for ByteChunk . If the value is not specified, the default value of false is used.

	No equivalent configuration
org.apache.tomcat.util.buf.StringCache.char.enabled	If true , the String cache is enabled for CharChunk . If the value is not specified, the default value of false is used.
	No equivalent configuration
org.apache.tomcat.util.buf.StringCache.cacheSize	The size of the String cache. If the value is not specified, the default value of 5000 is used.
	No equivalent configuration
org.apache.tomcat.util.buf.StringCache.maxLength	The maximum length of String that will be cached. If the value is not specified, the default value of 128 is used.
	No equivalent configuration
org.apache.tomcat.util.http.FastDateFormat.CACHE_SIZE	The size of the cache to use parsed and formatted date value. If the value is not specified, the default value of 1000 is used.
	No equivalent configuration
org.apache.catalina.core.StandardService.DELAY_CONNECTOR_STARTUP	If true , the connector startup is not done automatically. It is useful in embedded mode.
	No equivalent configuration
org.apache.catalina.connector.Request.SESSION_ID_CHECK	If true , the Servlet container verifies that a session exists in a context with the specified session ID before creating a session with that ID.
	No equivalent configuration
org.apache.coyote.Constants.USE_CUSTOM_STATUS_MSG_IN_HEADER	If true , custom HTTP status messages are used within HTTP headers. Users must ensure that any such message is ISO-8859-1 encoded, particularly if user provided input is included in the message, to prevent a possible XSS vulnerability. If value is not specified the default value of false is used.
	Must be enabled programmatically by implementing a custom io.undertow.servlet.ServletExtension . Then use the extension to call setSendCustomReasonPhraseOnError(true) on the io.undertow.servlet.api.DeploymentInfo structure instance.
org.apache.tomcat.util.http.Parameters.MAX_COUNT	The maximum number of parameters that can be parsed in a post body. If exceeded, parsing fails using an IllegalStateException . The default value is 512 parameters.

	<p>Management CLI command:</p> <pre> /subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=max-parameters,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=max-parameters,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=max-parameters,value=VALUE) </pre>
<p>org.apache.tomcat.util.http.MimeHeaders. MAX_COUNT</p>	<p>The maximum number of headers that can be sent in the HTTP request. If exceeded, parsing will fail using an IllegalStateException. The default value is 128 headers.</p> <p>Management CLI command:</p> <pre> /subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=max-headers,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=max-headers,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=max-headers,value=VALUE) </pre>
<p>org.apache.tomcat.util.net.MAX_THREADS</p>	<p>The maximum number of threads a connector is going to use to process requests. The default value is 32 x 512. (512 x Runtime.getRuntime().availableProcessors() for the JIO connector)</p> <p>Management CLI command:</p> <pre> /subsystem=io/worker=default:write-attribute(name=task-max-threads,value=VALUE) </pre>
<p>org.apache.coyote.http11.Http11Protocol. MAX_HEADER_SIZE</p>	<p>The maximum size of the HTTP headers, in bytes. If exceeded, parsing will fail using an ArrayOutOfBoundsException. The default value is 8192 bytes.</p>

	<p>Management CLI command:</p> <pre> /subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=max-header-size,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=max-header-size,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=max-header-size,value=VALUE) </pre>
<p>org.apache.coyote.http11.Http11Protocol.COMPRESSION</p>	<p>Allows using simple compression with the HTTP connector. The default value is off, and compression can be enabled using the value on to enable it conditionally, or force to always enable it.</p> <p>Configure a filter using the management CLI:</p> <pre> # Create a filter /subsystem=undertow/configuration=filter/gzip=gzipfilter:add() /subsystem=undertow/server=default-server/host=default-host/filter-ref=gzipfilter:add() </pre>
<p>org.apache.coyote.http11.Http11Protocol.COMPRESSION_RESTRICTED_UA</p>	<p>圧縮されたコンテンツを受け取らないユーザーエージェント regexprs。デフォルト値は空です。</p> <p>Configure a predicate in a filter using the management CLI:</p> <pre> # Use a predicate in a filter /subsystem=undertow/configuration=filter/gzip=gzipfilter:add() /subsystem=undertow/server=default-server/host=default-host/filter-ref=gzipfilter:add(predicate="regex[pattern='AppleWebKit',value=%{i>User-Agent}]") </pre>
<p>org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIME_TYPES</p>	<p>Content type prefixes of compressible content. The default value is text/html, text/xml, text/plain.</p>

	<p>Configure a predicate in a filter using the management CLI:</p> <pre># Use a predicate in a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add(predicate="regex[patte rn='text/html',value=%{o,Content-Type}]")</pre>
<p>org.apache.coyote.http11.Http11Protocol. COMPRESSION_MIN_SIZE</p>	<p>Minimum size of content that will be compressed. The default value is 2048 bytes.</p> <p>Configure a predicate in a filter using the management CLI:</p> <pre># Use a predicate in a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add(predicate="max- content-size[value=MIN_SIZE]")</pre>
<p>org.apache.coyote.http11.DEFAULT_CO NNECTION_TIMEOUT</p>	<p>Default socket timeout. The default value is 60000 ms.</p> <p>Management CLI command:</p> <pre>/subsystem=undertow/server=default- server/http-listener=default:write- attribute(name=no-request- timeout,value=VALUE) /subsystem=undertow/server=default- server/https-listener=default:write- attribute(name=no-request- timeout,value=VALUE) /subsystem=undertow/server=default- server/ajp-listener=default:write- attribute(name=no-request- timeout,value=VALUE)</pre>
<p>org.jboss.as.web.deployment.DELETE_W ORK_DIR_ONCONTEXTDESTROY</p>	<p>Use this property to remove .java and .class files to ensure that JSP sources are recompiled. The default value is false. Default socket timeout for keep-alive. The default value is -1 ms, which means it will use the default socket timeout.</p> <p>No equivalent configuration</p>
<p>org.apache.tomcat.util.buf.StringCache.tra inThreshold</p>	<p>Specifies the number of times toString() must be invoked before activating cache. The default value is 100000.</p>

No equivalent configuration

表A.2 Map EL System Properties

JBoss EAP 6 System Property	description
	Equivalent in JBoss EAP 7
org.apache.el.parser.COERCE_TO_ZERO	<p>If true, when coercing expressions to numbers, empty strings ("") and null will be coerced to zero as required by the specification. If a value is not specified, the default value of true is used.</p> <p>System property is still valid and processed by the EL</p>

表A.3 Map JSP System Properties

JBoss EAP 6 System Property	description
	Equivalent in JBoss EAP 7
org.apache.jasper.compiler.Generated.VARIABLE_EXPRESSIONFACTORY	<p>The name of the variable to use for the expression language expression factory. If value is not specified, the default value of _el_expressionfactory is used.</p> <p>System property has not changed</p>
org.apache.jasper.compiler.Generated.VARIABLE_INSTANCEMANAGER	<p>The name of the variable to use for the instance manager factory. If value is not specified, the default value of _jsp_instancemanager is used.</p> <p>System property has not changed</p>
org.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING	<p>If false, the requirements for escaping quotes in JSP attributes are relaxed so that a missing required quote does not cause an error. If value is not specified, the specification compliant default of true is used.</p> <p>System property has not changed</p>
org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE	<p>Any tag buffer that expands beyond org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE is destroyed and a new buffer is created of the default size. If value is not specified, the default value of 512 is used.</p> <p>System property has not changed</p>

org.apache.jasper.runtime.JspFactoryImpl. .USE_POOL	If true , a ThreadLocal PageContext pool is used. If value is not specified, the default value of true is used. System property has not changed
org.apache.jasper.runtime.JspFactoryImpl. .POOL_SIZE	The size of the ThreadLocal PageContext. If value is not specified, the default value of 8 is used. System property has not changed
org.apache.jasper.Constants.JSP_SERVLET_BASE	The base class of the Servlets generated from the JSPs. If value is not specified, the default value of org.apache.jasper.runtime.HttpJspBase is used. System property has not changed
org.apache.jasper.Constants.SERVICE_METHOD_NAME	The name of the service method called by the base class. If value is not specified, the default value of _jspService is used. System property has not changed
org.apache.jasper.Constants.SERVLET_CLASSPATH	The name of the ServletContext attribute that provides the class path for the JSP. If value is not specified, the default value of org.apache.catalina.jsp_classpath is used. System property has not changed
org.apache.jasper.Constants.JSP_FILE	The name of the request attribute for <jsp-file> element of a servlet definition. If present on a request, this overrides the value returned by request.getServletPath() to select the JSP page to be executed. If value is not specified, the default value of org.apache.catalina.jsp_file is used. System property has not changed
org.apache.jasper.Constants.PRECOMPILE	The name of the query parameter that causes the JSP engine to just pregenerate the servlet but not invoke it. If value is not specified, the default value of org.apache.catalina.jsp_precompile is used. System property has not changed
org.apache.jasper.Constants.JSP_PACKAGE_NAME	The default package name for compiled JSP pages. If value not specified, the default value of org.apache.jsp is used. System property has not changed

org.apache.jasper.Constants.TAG_FILE_PACKAGE_NAME	The default package name for tag handlers generated from tag files. If value is not specified, the default value of org.apache.jsp.tag is used.
	System property has not changed
org.apache.jasper.Constants.TEMP_VARIABLE_NAME_PREFIX	Prefix to use for generated temporary variable names. If value is not specified, the default value of __jspx_temp is used.
	System property has not changed
org.apache.jasper.Constants.USE_INSTANCE_MANAGER_FOR_TAGS	If true , the instance manager is used to obtain tag handler instances. If value is not specified, true is used.
	System property has not changed
org.apache.jasper.Constants.INJECT_TAGS	If true , annotations specified in tags will be processed and injected. This can have a performance impact when using simple tags, or if tag pooling is disabled. If value is not specified, false is used.
	System property has not changed

表A.4 Map Security System Properties

JBoss EAP 6 System Property	description
	Equivalent in JBoss EAP 7
org.apache.catalina.connector.RECYCLE_FACADES	If this is true or if a security manager is in use a new facade object is created for each request. If value is not specified, the default value of false is used.
	No equivalent configuration
org.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH	If this is true the '\' character is permitted as a path delimiter. If value is not specified, the default value of false is used.
	No equivalent configuration
org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH	If this is true , '%2F' and '%5C' is permitted as path delimiters. If value is not specified, the default value of false is used.

	<p>Management CLI command:</p> <pre> /subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=allow-encoded-slash,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=allow-encoded-slash,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=allow-encoded-slash,value=VALUE) </pre>
<p>org.apache.catalina.STRICT_SERVLET_COMPLIANCE</p>	<p>If value is not specified, true is used. If this is true the following actions will occur: any wrapped request or response object passed to an application dispatcher is checked to ensure that it has wrapped the original request or response. (SRV.8.2 / SRV.14.2.5.1) a call to Response.getWriter() if no character encoding has been specified results in subsequent calls to Response.getCharacterEncoding() returning ISO-8859-1 and the Content-Type response header will include a charset=ISO-8859-1 component. (SRV.15.2.22.1) every request that is associated with a session causes the session's last accessed time to be updated regardless of whether or not the request explicitly accesses the session. (SRV.7.6)</p> <p>Compliant by default</p>
<p>org.apache.catalina.core.StandardWrapperValve.SERVLET_STATS</p>	<p>If true or if org.apache.catalina.STRICT_SERVLET_COMPLIANCE is true, the wrapper will collect the JSR-77 statistics for individual servlets. If value is not specified, the default value of false is used.</p> <p>No equivalent configuration</p>
<p>org.apache.catalina.session.StandardSession.ACTIVITY_CHECK</p>	<p>If this is true or if org.apache.catalina.STRICT_SERVLET_COMPLIANCE is true Tomcat tracks the number of active requests for each session. When determining if a session is valid, any session with at least one active request is always be considered valid. If value is not specified, the default value of false is used.</p> <p>No equivalent configuration</p>

A.5. リリース間の互換性および相互運用性

ここでは、JBoss EAP 5、JBoss EAP 6、および JBoss EAP 7 リリース間での、クライアントおよびサーバー EJB とメッセージングコンポーネントの互換性および相互運用性について説明します。

IIOp 上の EJB リモータリング

以下の設定では問題が発生しません。

- JBoss EAP 5 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 5 サーバーへの接続

JNDI を使用した EJB リモータリング

以下の設定では問題が発生しません。

- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続

JBoss EAP 6 では、EJB 3.1 仕様のサポートが提供され、標準化されたグローバル JNDI ネームスペースが導入されました。JBoss EAP 7 でも標準化されたグローバル JNDI ネームスペースは使用されますが、JNDI ネームスペースの名前が変更になったため、以下の設定は互換性がありません。

- JBoss EAP 5 クライアントから JBoss EAP 7 または JBoss EAP 6 サーバーへの接続
- JBoss EAP 7 または JBoss EAP 6 クライアントから JBoss EAP 5 サーバーへの接続

標準化された JNDI ネームスペースの変更に関する詳細は、JBoss EAP 6 [移行ガイド](#)の [JNDI の変更](#)を参照してください。

@WebService を使用した EJB リモータリング

以下の設定では問題が発生しません。

- JBoss EAP 5 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 5 サーバーへの接続

メッセージングスタンドアロンクライアント

以下の設定では問題が発生しません。

- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続

以下の設定では、クライアントが汎用 JMS API ではなくメッセージングブローカー専用の HornetQ API を使用すれば接続が可能です。しかし、JBoss EAP 7 に同梱される JBoss EAP レガシー JNDI ネーミング拡張を使用して JNDI ルックアップに対応する必要があります。

- JBoss EAP 5 クライアントから JBoss EAP 7 サーバーへの接続

プロトコル互換性の問題があるため、JBoss EAP 7 のビルトインメッセージングは JBoss EAP 5 に同梱された HornetQ 2.2.x へは接続できません。そのため、以下の設定は互換性がありません。

- JBoss EAP 7 クライアントから JBoss EAP 5 サーバーへの接続

メッセージング MDB

以下の設定では問題が発生しません。

- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続

以下の設定では、クライアントが汎用 JMS API ではなくメッセージングブローカー専用の HornetQ API を使用すれば接続が可能です。しかし、JBoss EAP 7 に同梱される JBoss EAP レガシー JNDI ネーミング拡張を使用して JNDI ルックアップに対応する必要があります。

- JBoss EAP 5 クライアントから JBoss EAP 7 サーバーへの接続

プロトコル互換性の問題があるため、JBoss EAP 7 のビルトインメッセージングは JBoss EAP 5 に同梱された HornetQ 2.2.x へは接続できません。そのため、以下の設定は互換性はありません。

- JBoss EAP 7 クライアントから JBoss EAP 5 サーバーへの接続

JMS ブリッジ

以下の設定では問題が発生しません。

- JBoss EAP 5 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 6 クライアントから JBoss EAP 7 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 6 サーバーへの接続
- JBoss EAP 7 クライアントから JBoss EAP 5 サーバーへの接続

Revised on 2018-01-12 05:22:35 EST