



Red Hat Fuse 7.7

移行ガイド

Red Hat Fuse 7.7 への移行

Red Hat Fuse 7.7 移行ガイド

Red Hat Fuse 7.7 への移行

法律上の通知

Copyright © 2023 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, the Red Hat 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 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

本ガイドは、Fuse インストールを Red Hat Fuse の最新バージョンにアップグレードする際に役立ちます。

目次

はじめに	3
第1章 FUSE ONLINE のアップグレード	4
1.1. OPENSIFT ONLINE で実行されている FUSE ONLINE インテグレーションのアップグレード	4
1.2. OCP での FUSE ONLINE のアップグレード	4
第2章 SPRING BOOT スタンドアロンでの FUSE アプリケーションのアップグレード	7
2.1. MAVEN 依存関係について	7
2.2. FUSE プロジェクトの MAVEN 依存関係の更新	7
第3章 JBOSS EAP スタンドアロンでの FUSE アプリケーションのアップグレード	10
3.1. MAVEN 依存関係について	10
3.2. FUSE プロジェクトの MAVEN 依存関係の更新	10
3.3. 既存の FUSE ON JBOSS EAP インストールのアップグレード	11
第4章 KARAF スタンドアロンでの FUSE アプリケーションのアップグレード	13
4.1. MAVEN 依存関係について	13
4.2. FUSE プロジェクトの MAVEN 依存関係の更新	13
第5章 KARAF での FUSE スタンドアロンのアップグレード	15
5.1. FUSE ON KARAF のアップグレードによる影響	15
5.2. KARAF での FUSE スタンドアロンのアップグレード	16
5.3. FUSE ON KARAF のアップグレードのロールバック	17

はじめに

本ガイドでは、Red Hat Fuse および Fuse アプリケーションの更新に関する情報を提供します。



注記

Fuse 6 から最新の Fuse 7 リリースに移行する場合は、本ガイドの手順に従う前に、[Red Hat Fuse 7.0 Migration Guide](#) の手順に従う必要があります。

[1章 Fuse Online のアップグレード](#)

[2章 Spring Boot スタンドアロンでの Fuse アプリケーションのアップグレード](#)

[3章 JBoss EAP スタンドアロンでの Fuse アプリケーションのアップグレード](#)

[4章 Karaf スタンドアロンでの Fuse アプリケーションのアップグレード](#)

[5章 Karaf での Fuse スタンドアロンのアップグレード](#)

第1章 FUSE ONLINE のアップグレード

Fuse Online のアップグレードプロセスは、Fuse Online が Red Hat OpenShift Online または OpenShift Container Platform (OCP) にインストールされているかによって異なります。


- **OpenShift Online** の場合: Fuse 7.6 のリリース時に、OpenShift Online の Fuse Online インフラストラクチャーは自動的にアップグレードされます。Upgrading Fuse Online integrations that are running on OpenShift Online の説明にしたがって、稼働中のインテグレーションを再パブリッシュする必要があります。
- **OCP** の場合: オンサイトの OpenShift Container Platform で稼働している Fuse Online 環境をアップグレードするには、最新の Fuse Online リリースをダウンロードし、スクリプトを更新する必要があります。さらに [Upgrading Fuse Online on OCP](#) の説明どおりに稼働中のインテグレーションを再パブリッシュする必要があります。

1.1. OPENSIFT ONLINE で実行されている FUSE ONLINE インテグレーションのアップグレード

Fuse 7.2 のリリース時、OpenShift Online の Fuse Online インフラストラクチャーは自動的にアップグレードされます。インフラストラクチャーのアップグレード中、OpenShift Online で実行されている既存のインテグレーションはすべてアップグレード中とアップグレード後の実行が継続されます。しかし、既存のインテグレーションは古いバージョンの Fuse ライブラリーおよび依存関係で実行を継続します。

Fuse Online のインフラストラクチャーが新しいリリースにアップグレードされたことを知らせる電子メールのメッセージを受け取った後、既存のインテグレーションを再パブリッシュ (再起動だけではなく) してアップグレードします。これはできるだけ早期に行ってください。

Fuse Online 環境でインテグレーションを再パブリッシュするには、左側のナビゲーションパネルで **Integrations** をクリックします。その後、インテグレーションごとに以下を行います。

1. インテグレーションエントリーの右側で  をクリックし、**Edit** を選択します。
2. Fuse Online で編集するインテグレーションが表示されたら、右上の **Publish** をクリックします。

パブリッシュを行うと、最新の Fuse Online 依存関係を使用して再ビルドが強制されます。



注記

インテグレーションの要素に、更新の必要がある新しい依存関係がある場合、Fuse Online ユーザーインターフェイスに警告が表示されます。

1.2. OCP での FUSE ONLINE のアップグレード

オンサイトで OCP で Fuse Online をアップグレードするには、最新の Fuse Online リリースをダウンロードして、クラスター管理者から Fuse Online をアップグレードする権限を取得し、更新スクリプトを実行します。

時々、パッチとセキュリティ修正が組み込まれた新しいアプリケーションイメージが Fuse Online に対してリリースされます。これらの更新は、Red Hat のエラータ更新チャンネルを介して通知されます。その後、Fuse Online イメージをアップグレードできます。

下記条件のアップグレード手順は同じです。

- Fuse Online 7.6 から Fuse Online 7.7
- Fuse Online 7.7 の古いバージョンから Fuse Online 7.7 の新しいバージョン

前提条件

- オンサイトで Fuse Online バージョン 7.6 が OCP にインストールされ、稼働している。または、オンサイトで Fuse Online 7.7 のあるバージョンが OCP にインストールされ、稼働しており、新しいアプリケーションイメージへのアップグレードを考えている。
以前のバージョンの場合は以下が必要になります。
 - OCP で Fuse Online バージョン 7.5 を実行している場合は、[7.6 にアップグレード](#)してから 7.7 にアップグレードする必要があります。
 - OCP で Fuse Online バージョン 7.4 を実行している場合は、[7.5 にアップグレード](#)してから 7.6 にアップグレードする必要があります。
 - OCP で Fuse Online バージョン 7.3 を実行している場合は、[7.4 にアップグレード](#)してから 7.5 にアップグレードする必要があります。
 - OCP で Fuse Online バージョン 7.2 を実行している場合は、[7.3 にアップグレード](#)する必要があります。
 - OCP で Fuse Online バージョン 7.1 を実行している場合は、[7.2 にアップグレード](#)する必要があります。
- **oc** クライアントツールをインストール済みであり、Fuse Online がインストールされている OCP クラスタに接続されている。
- クラスタ管理者権限がある (この手順の最初の 5 つの手順で必要です)。

手順

1. クラスタ管理者は、Fuse Online パッケージをダウンロードしてして、特定のプロジェクトでユーザーに対して Fuse Online のアップグレード権限を割り当てる必要があります。
 - a. 以下の場所から Fuse Online インストールスクリプトが含まれるパッケージをダウンロードします。
<https://github.com/syndesisio/fuse-online-install/releases/tag/1.14>

ファイルシステムの任意の場所で、ダウンロードしたアーカイブをデプロイメントします。**fuse-online-install-1.13** ディレクトリーには、Fuse Online のアップグレード用のスクリプトとサポートファイルが含まれます。
 - b. デプロイメントしたアーカイブが含まれるディレクトリーに移動します。以下に例を示します。
cd fuse-online-install-1.13
 - c. 以下のように、クラスタ管理者アカウントで OpenShift にログインします。
oc login -u admin -p admin
 - d. Fuse Online をアップグレードする必要がある OpenShift プロジェクトに切り替えます。
oc project fuse-online-project
 - e. そのプロジェクト内だけで、Fuse Online のアップグレード権限を割り当てます。たとえば、以下のコマンドは、**developer** ユーザーに、Fuse Online のアップグレード権限を割り

当てます。クラスター管理者がこのコマンドを実行した後に、**developer** ユーザーは対象のプロジェクトでのみ (**fuse-online-project**) Fuse Online をアップグレードできます。

```
bash install_ocp.sh --grant developer
```

2. Fuse Online のアップグレード権限を割り当てられたユーザーは、アプリケーションを実行できます。

- a. 以下のように、OpenShift にログインします。

```
oc login -u developer
```

- b. 以下のように、Fuse Online をアップグレードするプロジェクトに切り替えます。

```
oc project fuse-online-project
```

- c. アップグレードするバージョンを確認するには、以下のように **--version** オプションを指定して更新スクリプトを実行します。

```
bash update_ocp.sh --version
```

- d. 以下のような更新スクリプトを実行します。

```
bash update_ocp.sh
```

このスクリプトの詳細を確認するには、**\$ bash update_ocp.sh --help** を実行します。

インフラストラクチャーのアップグレード中およびアップグレード後も、既存のインテグレーションは引き続き Fuse Online ライブラリーおよび依存関係の古いバージョンで実行されます。

3. 次のように稼働中の Fuse Online インテグレーションをアップグレードします。

- a. Fuse Online で、アップグレードするインテグレーションを選択します。

- b. **Edit** を選択します。

- c. **Publish** を選択してインテグレーションを再パブリッシュします。

インテグレーションの再パブリッシュを行うと、最新の Fuse Online 依存関係を使用して再ビルドが強制されます。

第2章 SPRING BOOT スタンドアロンでの FUSE アプリケーションのアップグレード

Spring Boot で Fuse アプリケーションをアップグレードするには、Fuse プロジェクトの Maven 依存関係を更新し、Fuse の正しいバージョンを使用していることを確認する必要があります。

通常は、Maven を使用して Fuse アプリケーションを構築します。Maven は、Apache の無料のオープンソースビルドツールです。Maven 設定は Fuse アプリケーションプロジェクトの **pom.xml** ファイルで定義されます。Fuse プロジェクトのビルド中、Maven が外部リポジトリを探し、必要なアーティファクトをダウンロードするのがデフォルトの動作になります。Maven ビルドプロセスで Fuse がサポートするアーティファクトの正しいセットを選択できるように、Fuse Bill of Materials (BOM) の依存関係を **pom.xml** ファイルに追加します。

以下のセクションでは、Maven の依存関係と Fuse プロジェクトでの更新方法を説明します。

- 「[Maven 依存関係について](#)」
- 「[Fuse プロジェクトの Maven 依存関係の更新](#)」

2.1. MAVEN 依存関係について

Maven BOM (Bill of Materials) ファイルの目的は、正常に動作する Maven 依存関係バージョンのセットを提供し、各 Maven アーティファクトに対して個別にバージョンを定義する必要をなくすることです。

Fuse が実行される各コンテナには専用の BOM ファイルがあります。



注記

これらの BOM ファイルは <https://github.com/jboss-fuse/redhat-fuse> にあります。または、BOM ファイル更新の詳細は latest Release Notes を参照してください。

Fuse BOM には以下の利点があります。

- Maven 依存関係のバージョンを定義するため、依存関係を **pom.xml** ファイルに追加するときにバージョンを指定する必要がありません。
- 特定バージョンの Fuse に対して完全にテストされ、完全にサポートする依存関係のセットを定義します。
- Fuse のアップグレードを簡素化します。



重要

Fuse BOM によって定義される依存関係のセットのみが Red Hat によってサポートされます。

2.2. FUSE プロジェクトの MAVEN 依存関係の更新

Spring Boot の Fuse アプリケーションをアップグレードするには、プロジェクトの Maven 依存関係を更新します。

手順

1. プロジェクトの **pom.xml** ファイルを開きます。
2. 以下の例のように、プロジェクトの **pom.xml** ファイル (または、場合によっては親 **pom.xml** ファイル) に **dependencyManagement** 要素を追加します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
  ...
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <!-- configure the versions you want to use here -->
    <fuse.version>7.11.1.fuse-sb2-7_11_1-00022-redhat-00002</fuse.version>

  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.jboss.redhat-fuse</groupId>
        <artifactId>fuse-springboot-bom</artifactId>
        <version>${fuse.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  ...
</project>
```



注記

Spring Boot バージョンも更新するようにしてください。これは通常、**pom.xml** ファイルの Fuse バージョンにあります。

```
<properties>
  <!-- configure the versions you want to use here -->
  <fuse.version>7.11.1.fuse-sb2-7_11_1-00022-redhat-00002</fuse.version>
  <spring-boot.version>2.5.13.RELEASE</spring-boot.version>
</properties>
```

3. **pom.xml** ファイルを保存します。

BOM を **pom.xml** ファイルで依存関係として指定した後、アーティファクトのバージョンを **指定せず** に Maven 依存関係を **pom.xml** ファイルに追加できるようになります。たとえば、**camel-velocity** コンポーネントの依存関係を追加するには、以下の XML フラグメントを **pom.xml** ファイルの **dependencies** 要素に追加します。

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

この依存関係の定義では、**version** 要素が省略されることに注意してください。

第3章 JBOSS EAP スタンドアロンでの FUSE アプリケーションのアップグレード

JBoss EAP で Fuse アプリケーションをアップグレードするには、Fuse プロジェクトの Maven 依存関係を更新し、Fuse の正しいバージョンを使用するようにする必要があります。

通常は、Maven を使用して Fuse アプリケーションを構築します。Maven は、Apache の無料のオープンソースビルドツールです。Maven 設定は Fuse アプリケーションプロジェクトの **pom.xml** ファイルで定義されます。Fuse プロジェクトのビルド中、Maven が外部リポジトリを探し、必要なアーティファクトをダウンロードするのがデフォルトの動作になります。Maven ビルドプロセスで Fuse がサポートするアーティファクトの正しいセットを選択できるように、Fuse Bill of Materials (BOM) の依存関係を **pom.xml** ファイルに追加します。

以下のセクションでは、Maven の依存関係と Fuse プロジェクトでの更新方法を説明します。

- 「[Maven 依存関係について](#)」
- 「[Fuse プロジェクトの Maven 依存関係の更新](#)」

3.1. MAVEN 依存関係について

Maven BOM (Bill of Materials) ファイルの目的は、正常に動作する Maven 依存関係バージョンのセットを提供し、各 Maven アーティファクトに対して個別にバージョンを定義する必要をなくすることです。

Fuse が実行される各コンテナには専用の BOM ファイルがあります。



注記

これらの BOM ファイルは <https://github.com/jboss-fuse/redhat-fuse> にあります。または、BOM ファイル更新の詳細は [latest Release Notes](#) を参照してください。

Fuse BOM には以下の利点があります。

- Maven 依存関係のバージョンを定義するため、依存関係を **pom.xml** ファイルに追加するときにバージョンを指定する必要がありません。
- 特定バージョンの Fuse に対して完全にテストされ、完全にサポートする依存関係のセットを定義します。
- Fuse のアップグレードを簡素化します。



重要

Fuse BOM によって定義される依存関係のセットのみが Red Hat によってサポートされます。

3.2. FUSE プロジェクトの MAVEN 依存関係の更新

JBoss EAP の Fuse アプリケーションをアップグレードするには、プロジェクトの Maven 依存関係を更新します。

手順

1. プロジェクトの **pom.xml** ファイルを開きます。
2. 以下の例のように、プロジェクトの **pom.xml** ファイル (または、場合によっては親 **pom.xml** ファイル) に **dependencyManagement** 要素を追加します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
...
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

  <!-- configure the versions you want to use here -->
  <fuse.version>7.11.1.fuse-sb2-7_11_1-00022-redhat-00002</fuse.version>

</properties>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.jboss.redhat-fuse</groupId>
      <artifactId>fuse-eap-bom</artifactId>
      <version>${fuse.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
...
</project>
```

3. **pom.xml** ファイルを保存します。

BOM を **pom.xml** ファイルで依存関係として指定した後、アーティファクトのバージョンを **指定せず** に Maven 依存関係を **pom.xml** ファイルに追加できるようになります。たとえば、**camel-velocity** コンポーネントの依存関係を追加するには、以下の XML フラグメントを **pom.xml** ファイルの **dependencies** 要素に追加します。

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

この依存関係の定義では、**version** 要素が省略されることに注意してください。

3.3. 既存の FUSE ON JBOSS EAP インストールのアップグレード

以下の手順で、既存の Fuse on JBoss EAP インストールをアップグレードする方法を説明します。

手順

1. ある JBoss EAP マイナーリリースから別のマイナーリリースにアップグレードするには、[JBoss EAP Patching and Upgrading Guide](#) の手順に従う必要があります。

2. Fuse を更新するには、[Installing on JBoss EAP](#) ガイドの説明に従って、Fuse on JBoss EAP インストーラーを実行する必要があります。



注記

Fuse アプリケーションの再コンパイルまたは再デプロイは必要ありません。

第4章 KARAF スタンドアロンでの FUSE アプリケーションのアップグレード

Karaf で Fuse アプリケーションをアップグレードするには、Fuse プロジェクトの Maven 依存関係を更新し、Fuse の正しいバージョンを使用するようにする必要があります。

通常は、Maven を使用して Fuse アプリケーションを構築します。Maven は、Apache の無料のオープンソースビルドツールです。Maven 設定は Fuse アプリケーションプロジェクトの **pom.xml** ファイルで定義されます。Fuse プロジェクトのビルド中、Maven が外部リポジトリを探し、必要なアーティファクトをダウンロードするのがデフォルトの動作になります。Maven ビルドプロセスで Fuse がサポートするアーティファクトの正しいセットを選択できるように、Fuse Bill of Materials (BOM) の依存関係を **pom.xml** ファイルに追加します。

以下のセクションでは、Maven の依存関係と Fuse プロジェクトでの更新方法を説明します。

- 「[Maven 依存関係について](#)」
- 「[Fuse プロジェクトの Maven 依存関係の更新](#)」

4.1. MAVEN 依存関係について

Maven BOM (Bill of Materials) ファイルの目的は、正常に動作する Maven 依存関係バージョンのセットを提供し、各 Maven アーティファクトに対して個別にバージョンを定義する必要をなくすることです。

Fuse が実行される各コンテナには専用の BOM ファイルがあります。



注記

これらの BOM ファイルは <https://github.com/jboss-fuse/redhat-fuse> にあります。または、BOM ファイル更新の詳細は latest Release Notes を参照してください。

Fuse BOM には以下の利点があります。

- Maven 依存関係のバージョンを定義するため、依存関係を **pom.xml** ファイルに追加するときにバージョンを指定する必要がありません。
- 特定バージョンの Fuse に対して完全にテストされ、完全にサポートする依存関係のセットを定義します。
- Fuse のアップグレードを簡素化します。



重要

Fuse BOM によって定義される依存関係のセットのみが Red Hat によってサポートされます。

4.2. FUSE プロジェクトの MAVEN 依存関係の更新

Karaf の Fuse アプリケーションをアップグレードするには、プロジェクトの Maven 依存関係を更新します。

手順

1. プロジェクトの **pom.xml** ファイルを開きます。
2. 以下の例のように、プロジェクトの **pom.xml** ファイル (または、場合によっては親 **pom.xml** ファイル) に **dependencyManagement** 要素を追加します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project ...>
  ...
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <!-- configure the versions you want to use here -->
    <fuse.version>7.11.1.fuse-sb2-7_11_1-00022-redhat-00002</fuse.version>

  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.jboss.redhat-fuse</groupId>
        <artifactId>fuse-karaf-bom</artifactId>
        <version>${fuse.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  ...
</project>
```

3. **pom.xml** ファイルを保存します。

BOM を **pom.xml** ファイルで依存関係として指定した後、アーティファクトのバージョンを **指定せず** に Maven 依存関係を **pom.xml** ファイルに追加できるようになります。たとえば、**camel-velocity** コンポーネントの依存関係を追加するには、以下の XML フラグメントを **pom.xml** ファイルの **dependencies** 要素に追加します。

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-velocity</artifactId>
  <scope>provided</scope>
</dependency>
```

この依存関係の定義では、**version** 要素が省略されることに注意してください。

第5章 KARAF での FUSE スタンドアロンのアップグレード

Fuse on Apache Karaf のアップグレードメカニズムを使用すると、更新バージョンの Fuse on Karaf を再インストールする必要なく、Apache Karaf コンテナに修正を適用できます。アップグレードによりデプロイされたアプリケーションで問題が発生した場合に、アップグレードをロールバックすることもできます。

アップグレードインストーラーファイルは、Fuse on Apache Karaf をインストールのに使用するファイルと同じものです。



注記

アップグレードのインストーラーファイルを取得するには、Red Hat カスタマーポータル [Downloads](#) ページに移動し、Fuse on Apache Karaf のインストールアーカイブの最新バージョンをダウンロードします (例: `fuse-karaf-7.12.0.fuse-7_12_0-00019-redhat-00001.zip`)。

- [「Fuse on Karaf のアップグレードによる影響」](#)
- [「Karaf での Fuse スタンドアロンのアップグレード」](#)
- [「Fuse on Karaf のアップグレードのロールバック」](#)

5.1. FUSE ON KARAF のアップグレードによる影響

アップグレードのメカニズムは、**バンドル JAR** および **静的ファイル** (たとえば `etc/` ディレクトリー下の設定ファイルなど) を含む、**すべての** インストールファイルへの更新を行うことができます。Fuse on Apache Karaf のアップグレードプロセス:

- バンドル JAR、設定ファイル、および静的ファイルを含むすべてのファイルを更新します。
- 現在のコンテナインスタンス (および `data/` ディレクトリー下のランタイムストレージ) とベースのインストール両方にパッチを適用します。したがって、パッチはコンテナインスタンスの削除後に維持されます。
- 機能リポジトリファイルおよび機能自体を含む、Karaf 機能に関連するすべてのファイルを更新します。そのため、ロールアップパッチ後にインストールされる機能はすべて、パッチが適用された正しい依存関係を参照します。
- 必要な場合は、設定ファイル (例: `etc/` にあるファイル) を更新し、パッチによる設定変更に伴う設定変更を自動的にマージします。マージの競合が発生した場合は、処理方法の詳細についてパッチログを確認してください。
- マージの競合のほとんどは、自動的に解決されます。たとえば、パッチメカニズムはプロパティファイルのプロパティレベルで競合を検出します。プロパティを変更したのがユーザーなのかパッチなのかを検出します。一方だけがプロパティを変更している場合、変更が保持されます。
- (静的ファイルを含む) インストールに加えられた **すべての** 変更を追跡し、パッチをロールバックできるようにします。



注記

ロールアウトパッチメカニズムは、内部の git リポジトリ (`patches/.management/history` の場所にある) を使用して、変更を追跡します。

5.2. KARAF での FUSE スタンドアロンのアップグレード

以下の手順で、Fuse on Apache Karaf をアップグレードする方法を順を追って説明します。アップグレード手順を開始する前に、すべての前提条件が満たされていることを確認します。

前提条件

- アップグレードする前に、Fuse on Apache Karaf インストールの完全バックアップを作成している。
- コンテナがまだ実行されていない場合は起動している。

ヒント

コンテナがバックグラウンド (またはリモート) で実行されている場合は、SSH コンソールクライアント `bin/client` を使用してコンテナに接続します。

- `patch:add` コマンドを呼び出して、アップグレードのインストーラーファイルをコンテナの環境に追加します。たとえば、`fuse-karaf-7.12.0.fuse-7_12_0-00019-redhat-00001.zip` アップグレードインストーラーファイルを追加するには、次のようにします。

```
patch:add file:///path/to/fuse-karaf-7.11.1.fuse-7_11_1-00013-redhat-00003.zip
```

手順

1. `patch:update` コマンドを実行します。コンテナを再起動する必要はありません。

```
karaf@root(>) patch:update
Current patch mechanism version: 7.1.0.fuse-710023-redhat-00001
New patch mechanism version detected: 7.2.0.fuse-720035-redhat-00001
Uninstalling patch features in version 7.1.0.fuse-710023-redhat-00001
Installing patch features in version 7.2.0.fuse-720035-redhat-00001
```

2. `patch:list` コマンドを呼び出し、アップグレードインストーラーのリストを表示します。このリストで、`[name]` 見出しの下にあるエントリはアップグレード ID です。以下に例を示します。

```
karaf@root(>) patch:list
[name] [installed] [rollup] [description]
fuse-karaf-7.2.0.fuse-720035-redhat-00001 false true fuse-karaf-7.2.0.fuse-720035-redhat-00001
```

3. 以下のように、`patch:simulate` コマンドを呼び出し、適用するアップグレードのアップグレード ID を指定して、アップグレードをシミュレートします。

```
karaf@root(>) patch:simulate fuse-karaf-7.2.0.fuse-720035-redhat-00001
INFO : org.jboss.fuse.modules.patch.patch-management (226): Installing rollup patch "fuse-
```

```

karaf-7.2.0.fuse-720035-redhat-00001"
===== Repositories to remove (9):
- mvn:io.hawt/hawtio-karaf/2.0.0.fuse-710018-redhat-00002/xml/features
...
===== Repositories to add (9):
- mvn:io.hawt/hawtio-karaf/2.0.0.fuse-720044-redhat-00001/xml/features
...
===== Repositories to keep (10):
- mvn:org.apache.activemq/artemis-features/2.4.0.amq-711002-redhat-1/xml/features
...
===== Features to update (100):
[name]                [version]                [new version]
aries-blueprint      4.2.0.fuse-710024-redhat-00002  4.2.0.fuse-720061-redhat-00001
...
===== Bundles to update as part of features or core bundles (100):
[symbolic name]                [version]                [new location]
io.hawt.hawtio-log              2.0.0.fuse-710018-redhat-00002
mvn:io.hawt/hawtio-log/2.0.0.fuse-720044-redhat-00001
...
===== Bundles to reinstall as part of features or core bundles (123):
[symbolic name]                [version]                [location]
com.fasterxml.jackson.core.jackson-annotations      2.8.11
mvn:com.fasterxml.jackson.core/jackson-annotations/2.8.11
...
Simulation only - no files and runtime data will be modified.
karaf@root(>

```

これにより、アップグレードの実行時にコンテナに加えられる変更のログが生成されますが、実際にはコンテナに何の変更も加えません。シミュレーションログを確認し、コンテナに加えられる変更を確認します。

4. **patch:install** コマンドを呼び出し、適用するアップグレードのアップグレード ID を指定して、コンテナをアップグレードします。以下に例を示します。

```
karaf@root(> patch:install fuse-karaf-7.11.1.fuse-7_11_1-00013-redhat-00003
```

5. アップグレードアーティファクトのいずれかを検索して、アップグレードを検証します。たとえば、Fuse 7.1.0 を Fuse 7.2.0 にアップグレードしたばかりの場合、ビルド番号 7_11_0-00023-redhat-00001 のバンドルを次のように検索できます。

```

karaf@root(> bundle:list -l | grep 7_11_1-00017-redhat-00001
 22 | Active | 80 | 7.11.1.fuse-7_11_1-00013-redhat-00003 |
mvn:org.jboss.fuse.modules/fuse-pax-transx-tm-narayana/7.11.1.fuse-7_11_1-00013-redhat-
00003
188 | Active | 80 | 7.11.1.fuse-7_11_1-00013-redhat-00003 |
mvn:org.jboss.fuse.modules.patch/patch-commands/7.11.1.fuse-7_11_1-00013-redhat-
00003

```



注記

アップグレード後にコンテナを再起動すると、新しいバージョンとビルド番号が Welcome バナーにも表示されます。

5.3. FUSE ON KARAF のアップグレードのロールバック

アップグレードが機能しない場合や、コンテナに新たな問題が発生することがあります。このような場合は、**patch:rollback** コマンドを使用して、簡単にアップグレードをロールバックし、システムを以前の状態に復元することができます。この一連の手順で、そのステップを順を追って説明します。

前提条件

- 最近 Fuse on Karaf がアップグレードされている。
- アップグレードをロールバックする必要がある。

手順

1. **patch:list** コマンドを呼び出し、最近インストールされたパッチのアップグレード ID **UPGRADE_ID** を取得します。
2. 以下のように **patch:rollback** コマンドを実行します。

```
patch:rollback UPGRADE_ID
```



注記

場合によっては、アップグレードをロールバックするためにコンテナを再起動する必要があります。そのような場合には、コンテナが自動的に再起動します。OSGi ランタイムの非常に動的な性質上、再起動時に互換性のないクラスに関連するエラーが表示される場合があります。これらのエラーは、起動または停止したばかりの OSGi サービスに関連しており、無視しても問題はありません。