



Red Hat Fuse 7.5

迁移指南

迁移到 Red Hat Fuse 7.5

Red Hat Fuse 7.5 迁移指南

迁移到 Red Hat Fuse 7.5

法律通告

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.

摘要

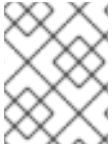
将 Fuse 安装升级到最新版本的 Red Hat Fuse 时，请使用本指南帮助您。

目录

前言	3
第 1 章 在 SPRING BOOT 独立升级 FUSE 应用程序	4
1.1. 关于 MAVEN 依赖项	4
1.2. 更新 FUSE 项目的 MAVEN 依赖项	4
第 2 章 在 JBOSS EAP 独立上升级 FUSE 应用程序	6
2.1. 关于 MAVEN 依赖项	6
2.2. 更新 FUSE 项目的 MAVEN 依赖项	6
第 3 章 在 KARAF 独立升级 FUSE 应用程序	8
3.1. 关于 MAVEN 依赖项	8
3.2. 更新 FUSE 项目的 MAVEN 依赖项	8
第 4 章 在 KARAF 上升级 FUSE STANDALONE	10
4.1. 将 FUSE 升级到 KARAF 的影响	10
4.2. 在 KARAF 上升级 FUSE STANDALONE	10
4.3. 在 KARAF 上回滚 FUSE 的升级	12

前言

本指南提供有关更新 Red Hat Fuse 和 Fuse 应用程序的信息：



注意

如果您要从 Fuse 6 迁移到最新的 Fuse 7 版本，请在本指南中的说明操作，请按照 [Red Hat Fuse 7.0 迁移](#) 指南中的说明进行操作。

[第 1 章 在 Spring Boot 独立升级 Fuse 应用程序](#)

[第 2 章 在 JBoss EAP 独立升级 Fuse 应用程序](#)

[第 3 章 在 Karaf 独立升级 Fuse 应用程序](#)

[第 4 章 在 Karaf 上升级 Fuse Standalone](#)

第 1 章 在 SPRING BOOT 独立升级 FUSE 应用程序

要在 Spring Boot 上升级 Fuse 应用程序，您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

通常，您可以使用 Maven 构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用的 **pom.xml** 文件中定义。在构建 Fuse 项目时，默认行为是 Maven 搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM) 的依赖项添加到 **pom.xml** 文件中，以便 Maven 构建过程获取正确的 Fuse 支持的工件集合。

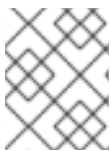
以下小节提供了有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

- [第 1.1 节 “关于 Maven 依赖项”](#)
- [第 1.2 节 “更新 Fuse 项目的 Maven 依赖项”](#)

1.1. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖项版本，这些版本可以很好地协同工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。

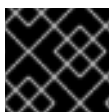


注意

您可以在此处找到这些 BOM 文件：<https://github.com/jboss-fuse/redhat-fuse>。或者，访问有关 BOM 文件更新的信息的[最新发行注记](#)。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 **pom.xml** 文件时不需要指定版本。
- 定义一组对特定版本的 Fuse 经过全面测试并支持的策展依赖关系。
- 简化 Fuse 的升级。



重要

红帽仅支持由 Fuse BOM 定义的一组依赖项。

1.2. 更新 FUSE 项目的 MAVEN 依赖项

要升级 Spring Boot 的 Fuse 应用程序，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 **pom.xml** 文件。
2. 在项目的 **pom.xml** 文件中添加 dependencies **Management** 元素（或者，可能在父 **pom.xml** 文件中），如下例所示：

```
<?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>7.5.0.fuse-750029-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>

```



注意

确保也更新您的 Spring Boot 版本。这通常位于 **pom.xml** 文件中的 Fuse 版本下：

```

<properties>
  <!-- configure the versions you want to use here -->
  <fuse.version>7.5.0.fuse-750029-redhat-00002</fuse.version>
  <spring-boot.version>1.5.19.RELEASE</spring-boot.version>
</properties>

```

3. 保存 **pom.xml** 文件。

在将 BOM 指定为 **pom.xml** 文件中的依赖项后，可以在没有指定工件版本 *的情况下* 将 Maven 依赖项添加到 **pom.xml** 文件中。例如，要为 **camel-velocity** 组件添加依赖项，您可以在 **pom.xml** 文件中的 **dependencies** 元素中添加以下 XML 片段：

```

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

```

注意如何在这个依赖项定义中省略 **version** 元素。

第 2 章 在 JBOSS EAP 独立上升级 FUSE 应用程序

要在 JBoss EAP 上升级 Fuse 应用程序，您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

通常，您可以使用 Maven 构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用的 **pom.xml** 文件中定义。在构建 Fuse 项目时，默认行为是 Maven 搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM) 的依赖项添加到 **pom.xml** 文件中，以便 Maven 构建过程获取正确的 Fuse 支持的工件集合。

以下小节提供了有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

- [第 2.1 节 “关于 Maven 依赖项”](#)
- [第 2.2 节 “更新 Fuse 项目的 Maven 依赖项”](#)

2.1. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖项版本，这些版本可以很好地协同工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。

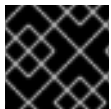


注意

您可以在此处找到这些 BOM 文件：<https://github.com/jboss-fuse/redhat-fuse>。或者，访问有关 BOM 文件更新的信息的[最新发行注记](#)。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 **pom.xml** 文件时不需要指定版本。
- 定义一组对特定版本的 Fuse 经过全面测试并支持的策展依赖关系。
- 简化 Fuse 的升级。



重要

红帽仅支持由 Fuse BOM 定义的一组依赖项。

2.2. 更新 FUSE 项目的 MAVEN 依赖项

若要升级用于 JBoss EAP 的 Fuse 应用，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 **pom.xml** 文件。
2. 在项目的 **pom.xml** 文件中添加 **dependencies Management** 元素（或者，可能在父 **pom.xml** 文件中），如下例所示：

```
<?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>7.5.0.fuse-750029-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. 保存 **pom.xml** 文件。

在将 BOM 指定为 **pom.xml** 文件中的依赖项后，可以在没有指定工件版本 *的情况下* 将 Maven 依赖项添加到 **pom.xml** 文件中。例如，要为 **camel-velocity** 组件添加依赖项，您可以在 **pom.xml** 文件中的 **dependencies** 元素中添加以下 XML 片段：

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

注意如何在这个依赖项定义中省略 **version** 元素。

第 3 章 在 KARAF 独立升级 FUSE 应用程序

要在 Karaf 上升级 Fuse 应用程序，您必须更新 Fuse 项目的 Maven 依赖项，以确保您使用正确的 Fuse 版本。

通常，您可以使用 Maven 构建 Fuse 应用程序。Maven 是 Apache 的免费开源构建工具。Maven 配置在 Fuse 应用的 **pom.xml** 文件中定义。在构建 Fuse 项目时，默认行为是 Maven 搜索外部存储库并下载所需的工件。您可以将 Fuse Bill of Materials (BOM) 的依赖项添加到 **pom.xml** 文件中，以便 Maven 构建过程获取正确的 Fuse 支持的工件集合。

以下小节提供了有关 Maven 依赖项的信息，以及如何在 Fuse 项目中更新它们。

- [第 3.1 节 “关于 Maven 依赖项”](#)
- [第 3.2 节 “更新 Fuse 项目的 Maven 依赖项”](#)

3.1. 关于 MAVEN 依赖项

Maven Bill of Materials (BOM) 文件的目的是提供一组策展的 Maven 依赖项版本，这些版本可以很好地协同工作，从而为您为每个 Maven 工件单独定义版本。

每个容器都有一个专用的 BOM 文件，在其中运行 Fuse。



注意

您可以在此处找到这些 BOM 文件：<https://github.com/jboss-fuse/redhat-fuse>。或者，访问有关 BOM 文件更新的信息的[最新发行注记](#)。

Fuse BOM 提供以下优点：

- 定义 Maven 依赖项的版本，以便在将依赖项添加到 **pom.xml** 文件时不需要指定版本。
- 定义一组对特定版本的 Fuse 经过全面测试并支持的策展依赖关系。
- 简化 Fuse 的升级。



重要

红帽仅支持由 Fuse BOM 定义的一组依赖项。

3.2. 更新 FUSE 项目的 MAVEN 依赖项

若要升级用于 Karaf 的 Fuse 应用，请更新项目的 Maven 依赖项。

流程

1. 打开项目的 **pom.xml** 文件。
2. 在项目的 **pom.xml** 文件中添加 **dependencies Management** 元素（或者，可能在父 **pom.xml** 文件中），如下例所示：

```
<?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>7.5.0.fuse-750029-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. 保存 **pom.xml** 文件。

在将 BOM 指定为 **pom.xml** 文件中的依赖项后，可以在没有指定工件版本 *的情况下* 将 Maven 依赖项添加到 **pom.xml** 文件中。例如，要为 **camel-velocity** 组件添加依赖项，您可以在 **pom.xml** 文件中的 **dependencies** 元素中添加以下 XML 片段：

```

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

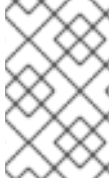
```

注意如何在这个依赖项定义中省略 **version** 元素。

第 4 章 在 KARAF 上升级 FUSE STANDALONE

通过 Apache Karaf 升级机制上的 Fuse，您可以将修复应用到 Apache Karaf 容器，而无需在 Karaf 上重新安装 Fuse 的更新版本。如果升级导致了与部署的应用程序相关的问题，它还允许您回滚升级。

升级安装程序文件是 *用来在 Apache Karaf 上安装 Fuse 的文件*。



注意

要获取升级安装程序文件，请进入红帽客户门户的 **Downloads** 页面，并下载 Apache Karaf 上 Fuse 的最新版本的安装存档（例如 **fuse-karaf-7.5.0.fuse-750035-redhat-00001.zip**）。

- [第 4.1 节 “将 Fuse 升级到 Karaf 的影响”](#)
- [第 4.2 节 “在 Karaf 上升级 Fuse Standalone”](#)
- [第 4.3 节 “在 Karaf 上回滚 Fuse 的升级”](#)

4.1. 将 FUSE 升级到 KARAF 的影响

升级机制可以对 **任何安装文件** 进行更新，包括 **捆绑包 JAR** 和 **静态文件**（如 **etc/** 目录下的配置文件）。Apache Karaf 升级过程上的 Fuse：

- 更新任何文件，包括捆绑 JAR、配置文件和任何静态文件。
- 对当前容器实例（及其在 **data/** 目录下的运行时存储）和底层安装进行补丁。因此，在删除容器实例后会保留补丁。
- 更新与 Karaf 功能相关的所有文件，包括功能存储库文件和功能本身。因此，推出补丁后安装的任何功能都将引用正确的补丁依赖项。
- 如有必要，更新配置文件（例如 **etc/** 下的文件），自动合并您已与补丁所做的配置更改所做的任何配置更改。如果发生合并冲突，请参阅补丁日志来了解如何处理它们的详细信息。
- 大多数合并冲突会自动解决。例如，补丁机制检测到属性文件的属性级别的冲突。它检测它是更改任何属性的用户或补丁。如果只更改属性，则更改会被保留。
- 跟踪对安装进行的所有更改（包括静态文件），以便可以回滚补丁。



注意

回滚补丁机制使用内部 git 存储库（位于 **patches/.management/history** 下）来跟踪所做的更改。

4.2. 在 KARAF 上升级 FUSE STANDALONE

以下说明指导您在 Apache Karaf 上升级 Fuse。在开始升级过程前，请确保所有先决条件都已完成。

先决条件

- 在升级前，确保您已在 Apache Karaf 安装上进行 Fuse 的完整备份。
- 启动容器（如果尚未运行）。

提示

如果容器在后台运行（或远程运行），请使用 SSH 控制台客户端 `bin/client` 连接到容器。

- 通过调用 `patch:add` 命令将升级安装程序文件添加到容器环境中。例如，要添加 `fuse-karaf-7.5.0.fuse-750035-redhat-00001.zip` 升级安装程序文件：

```
patch:add file:///path/to/fuse-karaf-7.5.0.fuse-750035-redhat-00001.zip
```

流程

1. 运行 `patch:update` 命令。不需要重启容器。

```
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. 调用 `patch:list` 命令来显示升级安装程序列表。在此列表中，`[name]` 标题下的条目是升级 ID。例如：

```
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. 通过调用 `patch:simulate` 命令并为您要应用的升级指定升级 ID 来模拟升级，如下所示：

```
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(>
```

这会生成执行升级时将对容器进行的更改日志，但不会对容器进行任何更改。检查模拟日志，以了解要对容器进行的更改。

4. 通过调用 `patch:install` 命令并为您要应用的升级指定升级 ID 来升级容器。例如：

```
karaf@root(> patch:install fuse-karaf-7.5.0.fuse-750035-redhat-00001
```

5. 通过搜索其中一个升级工件来验证升级。例如，如果您刚刚将 Fuse 7.1.0 升级到 Fuse 7.2.0，您可以搜索构建号 750035 的捆绑包，如下所示：

```
karaf@root(> bundle:list -l | grep 750035
 22 | Active | 80 | 7.5.0.fuse-750035-redhat-00001 | mvn:org.jboss.fuse.modules/fuse-
pax-transx-tm-narayana/7.5.0.fuse-750035-redhat-00001
188 | Active | 80 | 7.5.0.fuse-750035-redhat-00001 |
mvn:org.jboss.fuse.modules.patch/patch-commands/7.5.0.fuse-750035-redhat-00001
```



注意

升级后，在重启容器时，您还会在 Welcome banner 中看到新版本和构建号。

4.3. 在 KARAF 上回滚 FUSE 的升级

有时，升级可能无法正常工作，或者可能会给容器带来新的问题。在这些情况下，您可以使用 `patch:rollback` 命令轻松回滚升级并将系统恢复到之前的状态。这组指导您完成此流程。

先决条件

- 您最近在 Karaf 上升级了 Fuse。
- 您需要回滚升级。

流程

1. 调用 `patch:list` 命令，以获取最近安装的补丁的升级 ID `UPGRADE_ID`。
2. 调用 `patch:rollback` 命令，如下所示：

```
patch:rollback UPGRADE_ID
```



注意

在某些情况下，容器需要重启才能回滚升级。在这些情况下，容器会自动重启。由于 OSGi 运行时具有高度动态性质，在重启过程中，您可能会看到一些与不兼容类相关的严重错误。这些错误与刚才启动或停止且可以安全地忽略的 OSGi 服务相关。

