



Migration Toolkit for Runtimes 1.2

Maven 插件指南

将运行时的 Migration Toolkit for Runtimes 集成到 Maven 构建过程中。

Migration Toolkit for Runtimes 1.2 Maven 插件指南

将运行时的 Migration Toolkit for Runtimes 集成到 Maven 构建过程中。

法律通告

Copyright © 2024 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

本指南论述了如何使用 Migration Toolkit for Runtimes Maven 插件来简化 Java 应用程序的迁移。

目录

使开源包含更多	3
第 1 章 简介	4
1.1. 关于 MAVEN 插件指南	4
1.2. 关于 MIGRATION TOOLKIT FOR RUNTIMES	4
1.3. 关于 MAVEN 插件	4
第 2 章 开始使用	5
2.1. 运行 MAVEN 插件	5
2.2. 使用多个模块运行 MAVEN 插件	6
2.3. 访问报告	7
第 3 章 以 CSV 格式导出报告	8
3.1. 导出报告	8
3.2. 将 CSV 文件导入到电子表格程序中	8
3.3. 关于 CSV 数据结构	8
附录 A. 参考材料	10
A.1. 关于 MAVEN 插件参数	10
A.2. 默认日志属性	14
A.3. 关于规则故事点	15
A.4. 其他资源	16

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

第 1 章 简介

1.1. 关于 MAVEN 插件指南

本指南适用于希望使用 Migration Toolkit for Runtimes (MTR) 迁移 Java 应用程序或其他组件的工程师、顾问和其他组件。它描述了如何安装和运行 Maven 插件，检查生成的报告，并利用额外的功能。

1.2. 关于 MIGRATION TOOLKIT FOR RUNTIMES

什么是 Migration Toolkit for Runtimes?

Migration Toolkit for Runtimes (MTR) 是一个可扩展、可自定义的基于规则的工具，简化了 Java 应用程序的迁移和现代化。

MTR 检查应用程序工件，包括项目源目录和应用程序存档，然后生成 HTML 报告突出显示需要更改的区域。MTR 支持许多迁移路径，包括以下示例：

- 升级至 Red Hat JBoss Enterprise Application Platform 的最新版本
- 从 Oracle WebLogic 或 IBM WebSphere Application Server 迁移到 Red Hat JBoss Enterprise Application Platform
- 容器化应用程序并使之成为云就绪
- 从 Java Spring Boot 迁移到 Quarkus
- 从 Oracle JDK 更新至 OpenJDK
- 从 OpenJDK 8 升级到 OpenJDK 11
- 从 OpenJDK 11 升级到 OpenJDK 17
- 从 OpenJDK 17 升级到 OpenJDK 21
- 将 EAP Java 应用程序迁移到 Azure
- 将 Spring Boot Java 应用程序迁移到 Azure

有关用例和迁移路径的更多信息，请参阅[开发人员网页的 MTR](#)。

Migration Toolkit for Runtimes 如何简化迁移？

Migration Toolkit for Runtimes 会查找常见资源和在迁移应用程序时的已知问题。它为应用程序使用的技术提供了高级视图。

MTR 生成详细的报告，评估迁移或现代化路径。此报告可帮助您估算大型项目所需的工作量，并减少涉及的工作。

如何了解更多信息？

请参阅 [Migration Toolkit for Runtimes 简介](#)，了解有关 Migration Toolkit for Runtimes 中的功能、支持的配置、系统要求以及可用工具的更多信息。

1.3. 关于 MAVEN 插件

Migration Toolkit for Runtimes 的 Maven 插件集成到 Maven 构建流程中，允许开发人员每次执行源代码时持续评估迁移和现代化工作。它提供了大量报告，突出显示分析结果，专为希望更新每个构建的开发人员而设计。

第 2 章 开始使用

2.1. 运行 MAVEN 插件

Maven 插件通过包含对应用的 **pom.xml** 文件中的插件的引用来运行。构建应用程序时，运行 Maven 插件并生成报告进行分析。

先决条件

- 安装了 Java 开发套件 (JDK)。
 - MTR 支持以下 JDK :
 - OpenJDK 11
 - Oracle JDK 11
- 8 GB RAM
- macOS 安装：**maxproc** 的值必须是 **2048** 或更高版本。
- Maven **settings.xml** 文件配置为 Maven 使用 JBoss EAP Maven 存储库。
- 要在 OpenJDK 17 或 Oracle JDK17 上运行 Maven 插件，首先需要在命令行上设置 MAVEN_OPTS :

```
export MAVEN_OPTS="--add-modules=java.se --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.util=ALL-UNNAMED --add-opens java.base/java.util.stream=ALL-UNNAMED"
```

流程

1. 将以下 **<plugin>** 加入到应用程序的 **pom.xml** 文件：

```
[...]
<plugin>
  <groupId>org.jboss.windup.plugin</groupId>
  <artifactId>mtr-maven-plugin</artifactId>
  <version>1.2.6.GA-redhat-00001</version>
  <executions>
    <execution>
      <id>run-windup</id>
      <phase>package</phase>
      <goals>
        <goal>windup</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <target>eap:7</target> 1
  </configuration>
</plugin>
[...]
```

- 1 指定迁移目标。在配置中，必须至少提供一个迁移目标。

- 将 `--add-modules=java.se` 添加到 `MAVEN_OPTS` 环境变量中。

```
export MAVEN_OPTS="--add-modules=java.se
```

- 构建项目：

```
$ mvn clean install
```

您可以访问生成的报告。

2.2. 使用多个模块运行 MAVEN 插件

要在带有多个模块的项目中使用 Maven 插件，请将配置放在父的 `pom.xml` 中。Maven 插件将生成单个报告，其中包含父模块和任何子模块的分析。



注意

强烈建议在多模块项目中将 `inherited` 设置为 `false`；否则，当每个子编译时，Maven 插件将会在编译每个子时运行，从而导致对子模块进行多个 Maven 插件执行。将 `inherited` 设置为 `false` 时，每个项目中都分析一次，并大大减少了启动时间。

若要在带有多个模块的项目中运行 Maven 插件，请执行以下步骤：

- 在父项目的 `pom.xml` 中包含以下插件。以下是父模块的 `pom.xml` 示例。

```
<plugin>
  <groupId>org.jboss.windup.plugin</groupId>
  <artifactId>mtr-maven-plugin</artifactId>
  <version>1.2.6.GA-redhat-00001</version>
  <inherited>>false</inherited>
  <executions>
    <execution>
      <id>run-windup</id>
      <phase>package</phase>
      <goals>
        <goal>windup</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <input>${project.basedir}</input>
    <target>eap:7</target> ①
    <windupHome>>/PATH/TO/CLI<</windupHome>
  </configuration>
</plugin>
```

- ① 指定迁移目标。在配置中，必须至少提供一个迁移目标。

这个 `pom.xml` 文件与以下属性中的默认设置不同：

- inherited**：在插件级别上定义，此属性指示是否应在子模块中使用此配置。设置为 `false` 以提高性能。

- **input** : 指定包含要分析的项目的目录的路径。此属性默认为 `{project.basedir}/src/main`，并且应该定义父项目没有要分析的源代码。
 - **windupHome** : MTR CLI 提取副本的路径。此属性为可选，但建议以提高性能。上面的示例演示了一组推荐的参数。
2. 构建父项目。在构建过程中，Maven 插件针对项目中的所有子对象运行，而不进行进一步配置。

```
$ mvn clean install
```

3. 完成后，您可以访问生成的报告。此报告包含父项和所有子项的分析。

2.3. 访问报告

当您运行 Migration Toolkit for Runtimes 时，报告会在使用 `pom.xml` 中的 `outputDirectory` 参数指定的 `OUTPUT_REPORT_DIRECTORY` 中生成。在构建完成后，您会在构建日志中看到以下消息。

```
Windup report created: <OUTPUT_REPORT_DIRECTORY>/index.html
```

输出目录包含以下文件和子目录：

```
<OUTPUT_REPORT_DIRECTORY>/
├── index.html      // Landing page for the report
├── <EXPORT_FILE>.csv // Optional export of data in CSV format
├── graph/         // Generated graphs used for indexing
├── reports/       // Generated HTML reports
└── stats/         // Performance statistics
```

有关 MTR 报告和使用方法评估迁移或现代化工作的信息，请参阅 MTR CLI 指南的[审查报告](#)部分。

第 3 章 以 CSV 格式导出报告

Migration Toolkit for Runtimes (MTR) 提供将报告数据（包括分类和提示）导出到本地文件系统上的平面文件的功能。导出功能当前支持 CSV 文件格式，其报告数据以逗号分开的字段显示(,)。

可以通过电子表格软件（如 Microsoft Excel、OpenOffice Calc 或 libreoffice Calc）导入和操作 CSV 文件。电子表格软件提供了从 MTR 报告排序、分析、评估和管理结果数据的功能。

3.1. 导出报告

要将报告导出为 CSV 文件，请运行将 **exportCSV** 参数设置为 **true** 的 MTR。

在由分析的每个应用程序的 **--output** 参数指定的目录中创建一个 CSV 文件。所有发现的问题（跨越所有分析的应用程序）都包含在 **AllIssues.csv** 文件中。

CSV 文件导出到由 **outputDirectory** 参数指定的目录中。

3.2. 将 CSV 文件导入到电子表格程序中

1. 启动电子表格软件，如 Microsoft Excel。
2. 选择 **File** → **Open**。
3. 浏览 CSV 导出的文件并选择它。
4. 数据现在已准备好在电子表格软件中进行分析。

3.3. 关于 CSV 数据结构

CSV 格式化输出文件包含以下数据字段：

规则 Id

生成给定项目的规则 ID。

问题类型

hint 或 *classification*

标题

classification 或 *hint* 的标题。此字段总结了给定项目的问题。

描述

给定项目问题的详细描述。

links

提供有关此问题的其他信息的 URL。链接由两个属性组成：链接和描述。

Application

生成此项目的应用程序的名称。

文件名

给定项的文件名。

文件路径

给定项目的文件路径。

行

给定项的文件行号。

故事点

代表给定项目的工作水平的故事点数。


附录 A. 参考材料

A.1. 关于 MAVEN 插件参数

以下是可用 MTR Maven 插件参数的详细描述。

表 A.1. MTR Maven 插件参数

参数	描述
analyzeKnownLibraries	<p>用于分析应用程序中嵌入的已知软件工件的标志。默认情况下，MTR 仅分析应用程序代码。</p> <div style="display: flex; align-items: center;">  <div> <p>注意</p> <p>这个选项可能会导致执行时间较长，并报告大量迁移问题。</p> </div> </div>
customLoggingPropertiesFile	<p>一个到包含 java.util.logging.LogManager 日志记录配置的 logging.properties 文件的绝对路径。如果指定路径无效，或者未指定选项，则日志记录将恢复为使用 Maven 插件中包含的 logging.properties 文件。</p>
disableTattletale	<p>可禁用 Tattletale 报告的生成标志。如果同时将 enableTattletale 和 disableTattletale 设定为 true，则 disableTattletale 会被忽略，但仍会生成 Tattletale 报告。</p>
enableCompatibleFilesReport	<p>启用生成可组合文件报告的标志。由于处理的所有文件都未发现任何问题，对于大型应用程序，此报告可能需要很长时间才能完成。</p>
enableTattletale	<p>用于为每个应用程序生成 Tattletale 报告的标志。当 eap 位于包含的目标中时，默认启用这个选项。如果同时将 enableTattletale 和 disableTattletale 设定为 true，则 disableTattletale 会被忽略，但仍会生成 Tattletale 报告。</p>
enableTransactionAnalysis	<p>[技术预览] 标记来启用生成交易报告，该报告显示调用堆栈，该堆栈对关系数据库表执行操作。Enable Transaction Analysis 功能支持 Spring Data JPA，以及用于 SQL 语句执行的传统的 preparedStatement() 方法。它不支持 ORM 框架，如 Hibernate。</p> <div style="display: flex; align-items: center;">  <div> <p>注意</p> <p>enableTransactionAnalysis 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。</p> </div> </div>

参数	描述
excludePackages	要从评估中排除的软件包列表。例如，输入 "com.mycompany.commonutilities" 将排除所有软件包名称以 "com.mycompany.commonutilities" 开头的类。
excludeTags	要排除的标签列表。指定后，不会处理具有这些标签的规则。
explodedApps	指定提供的输入目录包含单个应用程序的源文件的标志。
exportCSV	标志，将报告数据导出到本地文件系统中的 CSV 文件。MTR 在 outputDirectory 参数指定的目录中创建文件。CSV 文件可导入到电子表格程序，以便进行数据操作和分析。
includeTags	要使用的标签列表。指定后，仅处理具有这些标签的规则。
inputDirectory	指定包含要分析的应用程序的目录路径。此参数默认为 {project.basedir}/src/main/ 。
keepWorkDirs	标记指示 MTR 不删除临时工作文件，如图形数据库和解压缩存档。这对于调试非常有用。
legacyReports	指示 MTR 生成旧格式报告而不是新的格式报告的标记。
软件包	MTR 要评估的软件包列表。此参数是必需的。
offlineMode	以离线模式运行的标记，禁用网络访问功能，如验证方案。用于提高性能。
outputDirectory	指定输出 MTR 生成的报告信息的目录路径。此参数默认为 {project.build.directory}/windup-report 。
覆盖	<p>用于强制删除 outputDirectory 指定的现有输出目录的标志。默认值为 true。</p> <div data-bbox="684 1624 1428 1915" style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> <div style="display: flex; align-items: center;">  <div> <p>警告</p> <p>请注意不要指定包含重要信息的报告输出目录！</p> </div> </div> </div>
sourceTechnologies	从迁移的一个或多个源技术、服务器、平台或框架的列表。这个参数与 targetTechnologies 参数结合使用有助于确定使用哪个规则集。

参数	描述
sourceMode	用于表示要评估的应用包含源文件而非编译的二进制文件的标志。默认值为 true 。
targetTechnologies	要迁移至的一个或多个目标技术、服务器、平台或框架的列表。这个参数与 sourceTechnologies 参数结合使用有助于确定使用哪个规则集。
userIgnorePath	指定 MTR 的位置，以识别应忽略的文件。
userRulesDirectory	为 MTR 指定寻找自定义 MTR 规则的位置。该值可以是包含规则集文件或单个规则集文件的目录。ruleset 文件必须使用 .windup.xml 后缀。
windupHome	一个可选参数，指向提取的 MTR CLI 的根目录。通过引用 CLI 的本地安装，Maven 插件可直接访问所有索引，从而提高了性能。
windupVersion	指定要运行的 MTR 版本。默认情况下，这是 Maven 插件的构建版本。

A.1.1. 指定输入目录

包含要分析的一个或多个应用程序的文件或目录的路径。默认为 **{project.basedir}/src/main/**。

使用方法

```
<inputDirectory>
  <INPUT_ARCHIVE_OR_DIRECTORY>
</inputDirectory>
```

A.1.2. 评估输入文件

根据为 **inputDirectory** 参数提供的输入文件类型，它会根据提供的附加参数进行评估，如下所示。

目录

--explodedApp	--sourceMode	没有参数
目录作为单个应用进行评估。	目录作为单个应用进行评估。	每个子目录都作为应用进行评估。

File

--explodedApp	--sourceMode	没有参数
参数将被忽略；该文件将评估为单个应用。	该文件作为压缩的项目进行评估。	该文件作为单个应用进行评估。

A.1.3. 指定输出目录

指定输出 MTR 生成的报告信息的目录路径。

使用方法

```
<outputDirectory>
  <OUTPUT_REPORT_DIRECTORY>
</outputDirectory>
```

- 如果省略，则报告将在 `{project.build.directory}/windup-report` 目录中生成。
- 如果存在输出目录，它将根据 `overwrite` 参数的值覆盖。此参数默认为 `true`，并导致 MTR 删除并重新创建目录。

A.1.4. 设置源技术

从迁移的一个或多个源技术、服务器、平台或框架的列表。这个参数与 `targetTechnologies` 参数结合使用有助于确定使用哪个规则集。

使用方法

```
<sourceTechnologies>
  <source>eap:6</source>
</sourceTechnologies>
```

现在，`SourceTechnologies` 参数提供版本支持，它遵循 [Maven 版本范围语法](#)。这指示 MTR 只运行与指定版本匹配的规则集。例如，`<source>eap:5</source>`。

A.1.5. 设置目标参数

要迁移至的一个或多个目标技术、服务器、平台或框架的列表。这个参数与 `sourceTechnologies` 参数结合使用有助于确定使用哪个规则集。这个参数是必需的

使用方法

```
<targetTechnologies>
  <target>eap:7</target>
</targetTechnologies>
```

`targetTechnologies` 参数现在提供版本支持，它遵循 [Maven 版本范围规范](#)。这指示 MTR 只运行与指定版本匹配的规则集。例如，`<target>eap:7</target>`。



警告

迁移到 JBoss EAP 时，请务必在目标中指定版本，例如 **eap:6**。仅指定 **eap** 将针对所有版本的 JBoss EAP 运行规则集，包括与您的迁移路径无关。

有关相应 JBoss EAP 版本的 *Migration Toolkit for Runtimes* 简介，请参阅[支持的迁移路径](#)。

A.1.6. 选择软件包

MTR 要评估的软件包列表。强烈建议您使用此参数。

使用方法

```
<packages>
  <package>
    <PACKAGE_1>
  </package>
  <package>
    <PACKAGE_2>
  </package>
</packages>
```

- 在大多数情况下，您只想评估自定义应用程序类软件包而不是标准 Java EE 或第三方软件包。<PACKAGE_N> 参数是一个软件包前缀；所有子软件包都会被扫描。例如，要扫描软件包 **com.mycustomapp** 和 **com.myotherapp**，请在 **pom.xml** 中使用下列代码片段：

```
<packages>
  <package>com.mycustomapp</package>
  <package>com.myotherapp</package>
</packages>
```

- 虽然您可以为标准 Java EE 第三方软件（如 **org.apache**）提供软件包名称，但通常最好不要包含它们，因为它们不会影响迁移工作。

A.2. 默认日志属性

Maven 插件包括的默认 **logging.properties** 文件如下。此配置省略了许多额外的消息，同时允许您查看 Maven 插件的进度。

默认 logging.properties 文件

```
# Licensed under the Eclipse Public License version 1.0, available at
# http://www.eclipse.org/legal/epl-v10.html
#

# Additional loggers to configure (the root logger is always configured)
#loggers=
handlers=java.util.logging.ConsoleHandler
.level=INFO
```

```
#java.util.logging.ConsoleHandler.level=INFO

#loggers=org.jboss.forge,org.jboss.weld,org.xnio,org.jboss.forge,org.ocpsoft.rewrite,org.jboss.windup.graph.GraphModelScanner,org.jboss.windup.reporting.xml.ClassificationHandler,org.jboss.windup.graph.GraphType$
org.jboss.forge.level=SEVERE
org.janusgraph.level=SEVERE
org.janusgraph.diskstorage.berkeleyje.BerkeleyJEKeyValueStore.level=SEVERE
org.janusgraph.diskstorage.berkeleyje.level=SEVERE
org.jboss.weld.level=SEVERE
org.xnio.level=SEVERE
org.jboss.forge.level=SEVERE
org.ocpsoft.rewrite.level=SEVERE
org.jboss.windup.graph.GraphModelScanner.level=SEVERE
org.jboss.windup.reporting.xml.ClassificationHandler.level=SEVERE
org.jboss.windup.graph.GraphTypeManager.level=SEVERE
org.jboss.windup.graph.GraphContextImpl.level=SEVERE
org.jboss.windup.rules.files.FileMapping.level=SEVERE
org.jboss.windup.exec.level=SEVERE
org.jboss.windup.config.level=SEVERE
com.thinkaurelius.level=SEVERE
org.jboss.windup=INFO
```

A.3. 关于规则故事点

A.3.1. 什么是故事点？

故事点是敏捷软件开发中常用的抽象指标，用于估算实施功能或更改所需的工作量水平。

应用的 Migration Toolkit for Runtimes 使用故事点来代表迁移特定应用程序构造所需的工作程度，以及整个应用程序。它不一定转换为“人-小时”，但该值在不同的任务间应保持一致。

A.3.2. 故事点如何在规则中估计

估算一个规则的故事点的工作量水平可能很棘手。以下是估算规则所需工作量时的一般准则 MTR。

努力级别	故事点	描述
信息	0	迁移过程中具有非常低或没有优先级的信息。
微小	1	迁移是一个微小的变化或一个简单的库交换，没有或有最小的 API 更改。
复杂	3	迁移任务所需的更改比较复杂，但有一个已包括在文档中的解决方案。
重新设计	5	迁移任务需要重新设计或完整的库更改，并有显著的 API 更改。
架构重组	7	迁移会需要组件或子系统的完全的架构重组。
Unknown	13	迁移解决方案并非已知的，可能需要进行彻底的重写。

A.3.3. 任务类别

除了工作程度外，您还可以对迁移任务进行分类，以指明任务的严重性。下列类别用于对问题进行分组，以帮助确定迁移的工作量。

Mandatory (必需)

必须成功完成该任务才能成功迁移。如果没有进行任何更改，则生成的应用不会成功构建或运行。例如，替换在目标平台中不支持的专有 API。

选填

如果没有完成迁移任务，应用程序应该可以正常工作，但结果可能不是最佳。如果迁移时没有进行任何更改，建议在迁移完成后尽快按计划设置。

Potential

应在迁移过程中检查该任务，但没有足够的详细信息来确定任务是否成功完成。当没有直接兼容类型时，这将迁移第三方专有类型。

信息

该任务会包括告知您存在某些文件。可能需要将它们检查或修改为现代化工作的一部分，但通常不需要进行更改。

有关分类任务的更多信息，请参阅[使用自定义规则类别](#)。

A.4. 其他资源

A.4.1. 对项目贡献

为了帮助 Migration Toolkit for Runtimes 包括大多数应用程序结构和服务器配置，您可以使用以下项目帮助：

- 发送电子邮件到 jboss-migration-feedback@redhat.com，并告知我们必须覆盖哪些 MTR 迁移规则。
- 提供示例应用程序来测试迁移规则。
- 识别迁移可能很难迁移的应用程序组件和问题区域：
 - 编写问题迁移区域的简短描述。
 - 编写简短概述，了解如何解决问题迁移区域。
- 在应用程序上尝试 Migration Toolkit for Runtimes。确保报告您满足的任何问题。
- 贡献 Migration Toolkit for Runtimes 规则存储库：
 - 编写 Migration Toolkit for Runtimes 规则以识别或自动迁移过程。
 - 为新规则创建一个测试。

如需更多信息，[请参阅规则开发指南](#)。

- 贡献项目源代码：
 - 创建核心规则。
 - 提高 MTR 性能或效率。

任何级别的参与都非常感谢！

A.4.2. Migration Toolkit for Runtimes 开发资源

使用以下资源来学习并贡献 Migration Toolkit for Runtimes 开发：

- MTR论坛：<https://developer.jboss.org/en/windup>
- JIRA issue tracker: <https://issues.redhat.com/projects/WINDUP>
- MTR 邮件列表：jboss-migration-feedback@redhat.com

A.4.3. 报告问题

MTR 使用 JIRA 作为其问题跟踪系统。如果您在执行 MTR 时遇到问题，请提交 [JIRA 问题](#)。

更新于 2024-06-12