



Migration Toolkit for Runtimes 1.2

Visual Studio Code 扩展指南

使用 Visual Studio Code 的 Migration Toolkit for Runtimes 扩展来分析应用程序并解决迁移问题。

Migration Toolkit for Runtimes 1.2 Visual Studio Code 扩展指南

使用 Visual Studio Code 的 Migration Toolkit for Runtimes 扩展来分析应用程序并解决迁移问题。

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

本指南介绍了如何对 Visual Studio Code 使用 Migration Toolkit for Runtimes extension 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 MICROSOFT VISUAL STUDIO CODE 的 MTR 扩展	4
1.2. 关于 MIGRATION TOOLKIT FOR RUNTIMES	4
第 2 章 为 VISUAL STUDIO CODE 安装 MTR 扩展	5
第 3 章 使用 MTR 扩展分析项目	6
3.1. MTR 扩展接口	6
3.2. 配置运行配置	6
第 4 章 检查并解决迁移问题	8
4.1. 检查问题	8
4.2. 解决问题	8

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 MICROSOFT VISUAL STUDIO CODE 的 MTR 扩展

您可以使用 Microsoft Visual Studio Code 的 Migration Toolkit for Runtimes (MTR) 扩展来迁移和现代化应用程序。

MTR 扩展利用自定义规则集来分析您的项目，标记源代码中的问题，提供指导来修复问题，并提供自动代码替换（如果可能）。

该 MTR 扩展还可与 Microsoft 托管的开发环境 Visual Studio Code 空间兼容。

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 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。

第 2 章 为 VISUAL STUDIO CODE 安装 MTR 扩展

您可以为 Visual Studio Code (VS Code) 安装 MTR 扩展。

先决条件

以下是 Migration Toolkit for Runtimes (MTR) 安装的先决条件：

- 安装了 Java Development Kit (JDK)。MTR 支持以下 JDK：
 - OpenJDK 11
 - OpenJDK 17
 - Oracle JDK 11
 - Oracle JDK 17
 - Eclipse Temurin™ JDK 11
 - Eclipse Temurin™ JDK 17
- 8 GB RAM
- macOS 安装：**maxproc** 的值必须是 **2048** 或更高版本。

流程

1. 设置环境变量 **JAVA_HOME**：

```
$ export JAVA_HOME=jdk11
```

2. 在 VS Code 中，单击 Activity 栏中的 **Extensions** 图标，以打开 **Extensions** 视图。
3. 在 Search 字段中输入 **Migration Toolkit for Runtimes**。
4. 选择 **Migration Toolkit for Runtimes** 扩展，然后点 **Install**。

MTR 扩展图标()显示在 Activity 栏中。

第 3 章 使用 MTR 扩展分析项目

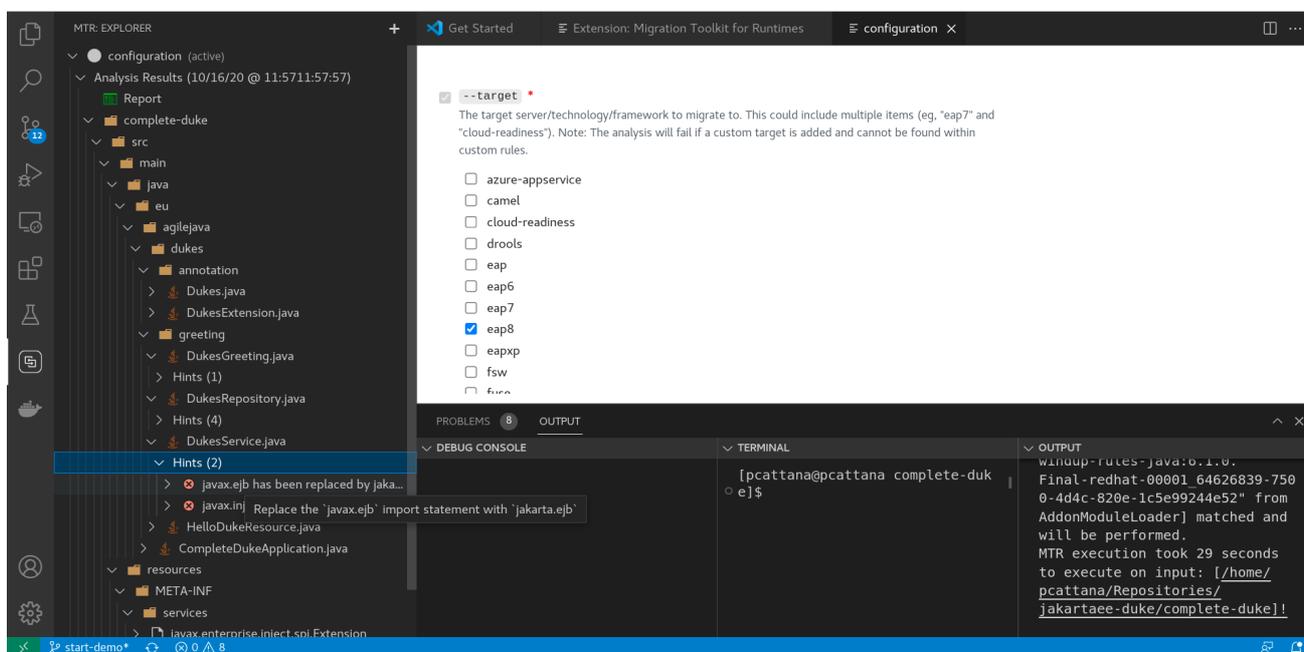
您可以通过创建运行配置和运行分析，通过 MTR 扩展来分析您的项目。

3.1. MTR 扩展接口

Migration Toolkit for Runtimes (MTR) 扩展的接口旨在让您更轻松地查找信息并执行操作：

- 在左侧窗格中，您可以使用其顶部的报告图标查看名为 **Analysis Results** 的目录树。您可以点击图标在浏览器中打开 MTR 报告。报告图标下是树的其它元素：MTR 分析的应用程序、使用的规则集以及分析发现的问题。
- 在右窗格中，您可以配置分析。
- 在右下角的窗格中，您可以看到配置的设置，包括源、目标和高级选项。您可以在此窗格中查看分析的进度。分析完成后，您可以点 **Open Report** 按钮以打开 MTR 报告，该报告描述了迁移或现代化应用程序前需要解决的所有问题。如需更多信息，请参阅 *CLI 指南* 中的 [检查报告](#)。

图 3.1. MTR 扩展接口



3.2. 配置运行配置

您可以配置多个运行配置，以针对您导入到 VS Code 的每个项目运行。

先决条件

- 已安装了 **mta-cli** 可执行文件。您可以从 [mtr](#) 下载 **mta-cli** 可执行文件。

流程

1. 在 **Extensions** 视图中，点 Activity 栏中的 Migration Toolkit for Runtimes 图标 ()。
2. 点 **Migration Toolkit for Runtimes** 旁边的 + (加号) 来添加运行配置。
3. 完成以下配置字段：

- **名称** : 为分析配置输入一个有意义的名称, 或者接受默认设置。
 - **cli**: 输入到 cli 执行文件的路径。例如: `$HOME/mta-cli-1.2.6.GA-redhat-00001/bin/mta-cli`。
 - **输入** : 点 **Add** 并执行以下操作之一来设置为您在 IDE 中打开的项目的路径:
 - 输入输入文件或目录, 然后按 Enter 键。
 - 点 **Open File Explorer**, 再选择目录。
 - **目标** : 选择一个或多个目标迁移路径。
4. 右键单击运行配置并选择 **运行**。
- 分析完成后, 您可以点 **Open Report** 按钮以打开 MTR 报告, 该报告描述了迁移或现代化应用程序前需要解决的所有问题。如需更多信息, 请参阅 *CLI 指南* 中的 [检查报告](#)。

第 4 章 检查并解决迁移问题

您可以在左侧窗格中查看并解决由 MTR 扩展确定的迁移问题。

4.1. 检查问题

您可以使用 MTR 扩展图标根据问题的严重性确定优先级。

流程

1. 选择左侧窗格中的运行配置目录。
2. 展开其文件夹以查看为每个应用程序文件生成的 **Hints**。
3. 选择一个 hint 来查看源代码。
4. 右键点提示并选择 **View Details** 以查看规则 ID 和其他信息。
5. 根据以下图标排序问题，这些图标会显示在每个提示旁边：
 -  : 必须解决这个问题才能成功迁移。
 -  : 此问题可能需要在迁移过程中解决。

4.2. 解决问题

您可以使用 hint 编辑文件代码来解决这个问题

4.2.1. 编辑文件代码

您可以编辑导入到 VS Code 的项目的文件。

流程

1. 在左侧窗格中，在问题上单击鼠标右键，然后选择**打开代码**。
2. 进行代码所需的任何更改，并保存文件。
3. 可选：在问题右键并选择 **Mark as Complete** 或 **Delete**。