Migration Toolkit for Runtimes 1.2

Visual Studio Code Extension Guide

Identify and resolve migration issues by analyzing your applications with the Migration Toolkit for Runtimes extension for Visual Studio Code.
Migration Toolkit for Runtimes 1.2 Visual Studio Code Extension Guide

Identify and resolve migration issues by analyzing your applications with the Migration Toolkit for Runtimes extension for Visual Studio Code.
Abstract

This guide describes how to use the Migration Toolkit for Runtimes extension for Visual Studio Code to simplify the migration of Java applications.
# Table of Contents

**MAKING OPEN SOURCE MORE INCLUSIVE**  ................................................................. 3

**CHAPTER 1. INTRODUCTION**  .................................................................................... 4  
1.1. ABOUT THE MTR EXTENSION FOR MICROSOFT VISUAL STUDIO CODE  .................................................. 4  
1.2. ABOUT THE MIGRATION TOOLKIT FOR RUNTIMES  ................................................................. 4  
   What is the Migration Toolkit for Runtimes?  ............................................................................. 4  
   How does the Migration Toolkit for Runtimes simplify migration?  ...................................... 4

**CHAPTER 2. INSTALLING THE MTR EXTENSION FOR VISUAL STUDIO CODE**  .................................................. 6

**CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTR EXTENSION**  .................................................. 7  
3.1. MTR EXTENSION INTERFACE  ......................................................................................... 7  
3.2. CONFIGURING A RUN CONFIGURATION  ............................................................................... 7

**CHAPTER 4. REVIEWING AND RESOLVING MIGRATION ISSUES**  .................................................. 9  
4.1. REVIEWING ISSUES  ......................................................................................................... 9  
4.2. RESOLVING ISSUES  ....................................................................................................... 9  
   4.2.1. Using a Quick Fix  .................................................................................................... 9  
   4.2.2. Editing the code of a file  .......................................................................................... 10
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
CHAPTER 1. INTRODUCTION

1.1. ABOUT THE MTR EXTENSION FOR MICROSOFT VISUAL STUDIO CODE

You can migrate and modernize applications by using the Migration Toolkit for Runtimes (MTR) extension for Microsoft Visual Studio Code.

The MTR extension analyzes your projects using customizable rulesets, marks issues in the source code, provides guidance to fix the issues, and offers automatic code replacement, if possible.

The MTR extension is also compatible with Visual Studio Codespaces, the Microsoft cloud-hosted development environment.

1.2. ABOUT THE MIGRATION TOOLKIT FOR RUNTIMES

What is the Migration Toolkit for Runtimes?

Migration Toolkit for Runtimes (MTR) accelerates large-scale application modernization efforts across hybrid cloud environments on Red Hat OpenShift. This solution provides insight throughout the adoption process, at both the portfolio and application levels: inventory, assess, analyze, and manage applications for faster migration to OpenShift via the user interface.

MTR uses an extensive questionnaire as the basis for assessing your applications, enabling you to estimate the difficulty, time, and other resources needed to prepare an application for containerization. You can use the results of an assessment as the basis for discussions between stakeholders to determine which applications are good candidates for containerization, which require significant work first, and which are not suitable for containerization.

MTR analyzes applications by applying one or more rulesets to each application considered to determine which specific lines of that application must be modified before it can be modernized.

MTR examines application artifacts, including project source directories and application archives, and then produces an HTML report highlighting areas needing changes. MTR supports many migration paths including the following:

- Upgrading to the latest release of Red Hat JBoss Enterprise Application Platform
- Migrating from Oracle WebLogic or IBM WebSphere Application Server to Red Hat JBoss Enterprise Application Platform
- Containerizing applications and making them cloud-ready
- Migrating from Java Spring Boot to Quarkus
- Upgrading from OpenJDK 8 to OpenJDK 11
- Upgrading from OpenJDK 11 to OpenJDK 17
- Migrating EAP Java applications to Azure App Service
- Migrating Spring Boot Java applications to Azure App Service

For more information about use cases and migration paths, see the MTR for developers web page.

How does the Migration Toolkit for Runtimes simplify migration?
The Migration Toolkit for Runtimes looks for common resources and known trouble spots when migrating applications. It provides a high-level view of the technologies used by the application.

MTR generates a detailed report evaluating a migration or modernization path. This report can help you to estimate the effort required for large-scale projects and to reduce the work involved.
CHAPTER 2. INSTALLING THE MTR EXTENSION FOR VISUAL STUDIO CODE

You can install the MTR extension for Visual Studio Code (VS Code).

Prerequisites

- Java Development Kit (JDK) installed. MTR supports the following JDKs:
  - OpenJDK 11
  - OpenJDK 17
  - Oracle JDK 11
  - Oracle JDK 17
  - Eclipse Temurin™ JDK 11
  - Eclipse Temurin™ JDK 17
- 8 GB RAM
- macOS installation: the value of `maxproc` must be **2048** or greater.

Procedure

1. Set the environmental variable `JAVA_HOME`:

   ```bash
   $ export JAVA_HOME=jdk11
   ```

2. In VS Code, click the Extensions icon on the Activity bar to open the Extensions view.

3. Enter **Migration Toolkit for Runtimes** in the Search field.

4. Select the **Migration Toolkit for Runtimes** extension and click Install.
   The MTR extension icon ( tamil ) is displayed on the Activity bar.
You can analyze your projects with the MTR extension by creating a run configuration and running an analysis.

3.1. MTR EXTENSION INTERFACE

The interface of the Migration Toolkit for Runtimes (MTR) extension is designed to make it easier for you to find information and perform actions:

- In the left pane, you can see a directory tree named **Analysis Results** with a report icon at its top. You can click the icon to open the MTR report in your browser. Beneath the report icon are the other elements of the tree: the applications analyzed by MTR, the rulesets used, and the issues discovered by the analysis.

- In the upper right pane, you can configure an analysis.

- In the lower right pane, you can see the settings of the configuration, including source, target, and advanced options. You can view the progress of an analysis in this pane. When the analysis is completed, you can click the **Open Report** button to open the MTR report, which describes any issues you need to address before you migrate or modernize your application. For more information, see **Reviewing the reports** in the **CLI Guide**.

3.2. CONFIGURING A RUN CONFIGURATION

You can configure multiple run configurations to run against each project you import to VS Code.

**Prerequisites**

- **windup-cli** executable installed. You can download the **windup-cli** executable from **mtr download**.

Figure 3.1. MTR extension interface

![MTR extension interface](image-url)
Procedure

1. In Extensions view, click the Migration Toolkit for Runtimes icon ( ) on the Activity bar.

2. Click the + (plus sign) next to Migration Toolkit for Runtimes to add a run configuration.

3. Complete the following configuration fields:
   - **Name**: Enter a meaningful name for the analysis configuration or accept the default.
   - **cli**: Enter the path to the cli executable. For example: $HOME/mtr-cli-1.2.3.GA-redhat-00001/bin/windup-cli.
   - **Input**: Set to the path of the project that you have open within your IDE by clicking Add and doing one of the following:
     - Enter the input file or directory and press Enter.
     - Click Open File Explorer and select the directory.
   - **Target**: Select one or more target migration paths.

4. Right-click the run configuration and select Run. When the analysis is completed, you can click the Open Report button to open the MTR report, which describes any issues you need to address before you migrate or modernize your application. For more information, see Reviewing the reports in the CLI Guide.
CHAPTER 4. REVIEWING AND RESOLVING MIGRATION ISSUES

You can review and resolve migration issues identified by the MTR extension in the left pane.

4.1. REVIEWING ISSUES

You can use the MTR extension icons to prioritize issues based on their severity. You can also see which issues have a Quick Fix automatic code replacement.

Procedure

1. Select a run configuration directory on the left pane.
2. Expand its folders to view the Hints generated for each application file.
3. Select a hint to view the source code.
4. Right-click a hint and select View Details to view the Rule ID and other information.
5. Prioritize issues based on the following icons, which are displayed next to each hint:
   - : The issue must be fixed for a successful migration.
   - : The issue might need to be addressed during migration.
   - : The issue is optional to fix for migration. This icon is also used to indicate any Quick Fixes available for an issue.

4.2. RESOLVING ISSUES

You can resolve issues by doing one of the following:

- Using a Quick Fix automatic code replacement
- Editing the code of a file with a hint

You can keep track of resolved issues by using the Delete or Mark as Complete options. Files marked deleted or completed will be analyzed again the next time you analyze a project that contains them.

4.2.1. Using a Quick Fix

You can use a Quick Fix automatic code replacement to save time and ensure consistency in resolving repetitive issues.

Procedure

1. In the left pane, right-click an issue that has the Quick Fix icon ( ) and select Preview Quick Fix.
2. To accept the suggested fix, right-click the issue again and select Apply Quick Fix.
3. Optional: Right-click the issue and select Mark as Complete or Delete.
4.2.2. Editing the code of a file

You can edit the file of a project imported to VS Code.

Procedure

1. In the left pane, right-click an issue and select Open Code.

2. Make any changes needed to the code and save the file.

3. Optional: Right-click the issue and select Mark as Complete or Delete.