



Migration Toolkit for Applications 5.3

Visual Studio Code Extension Guide

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

Migration Toolkit for Applications 5.3 Visual Studio Code Extension Guide

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

Legal Notice

Copyright © 2022 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.

Abstract

This guide describes how to use the Migration Toolkit for Applications 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 MTA EXTENSION FOR MICROSOFT VISUAL STUDIO CODE	4
1.2. ABOUT THE MIGRATION TOOLKIT FOR APPLICATIONS	4
What is the Migration Toolkit for Applications?	4
How does the Migration Toolkit for Applications simplify migration?	4
How do I learn more?	4
CHAPTER 2. INSTALLING THE MTA EXTENSION FOR VISUAL STUDIO CODE	5
CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA EXTENSION	6
3.1. MTA EXTENSION INTERFACE	6
3.2. CONFIGURING A RUN CONFIGURATION	6
CHAPTER 4. REVIEWING AND RESOLVING MIGRATION ISSUES	8
4.1. REVIEWING ISSUES	8
4.2. RESOLVING ISSUES	8
4.2.1. Using a Quick Fix	8
4.2.2. Editing the code of a file	9

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 MTA EXTENSION FOR MICROSOFT VISUAL STUDIO CODE

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

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

1.2. ABOUT THE MIGRATION TOOLKIT FOR APPLICATIONS

What is the Migration Toolkit for Applications?

The Migration Toolkit for Applications (MTA) is an extensible and customizable rule-based tool that simplifies the migration and modernization of Java applications.

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

- 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
- Updating from Oracle JDK to OpenJDK

For more information about use cases and migration paths, see the [MTA for developers](#) web page.

How does the Migration Toolkit for Applications simplify migration?

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

MTA 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.

How do I learn more?

See the [Introduction to the Migration Toolkit for Applications](#) to learn more about the features, supported configurations, system requirements, and available tools in the Migration Toolkit for Applications.

CHAPTER 2. INSTALLING THE MTA EXTENSION FOR VISUAL STUDIO CODE

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


Prerequisites

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

Procedure

1. Set the environmental variable **JAVA_HOME**:

```
$ 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 Applications** in the Search field.
4. Select the **Migration Toolkit for Applications** extension and click **Install**.
The MTA extension icon () is displayed on the Activity bar.

CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA EXTENSION

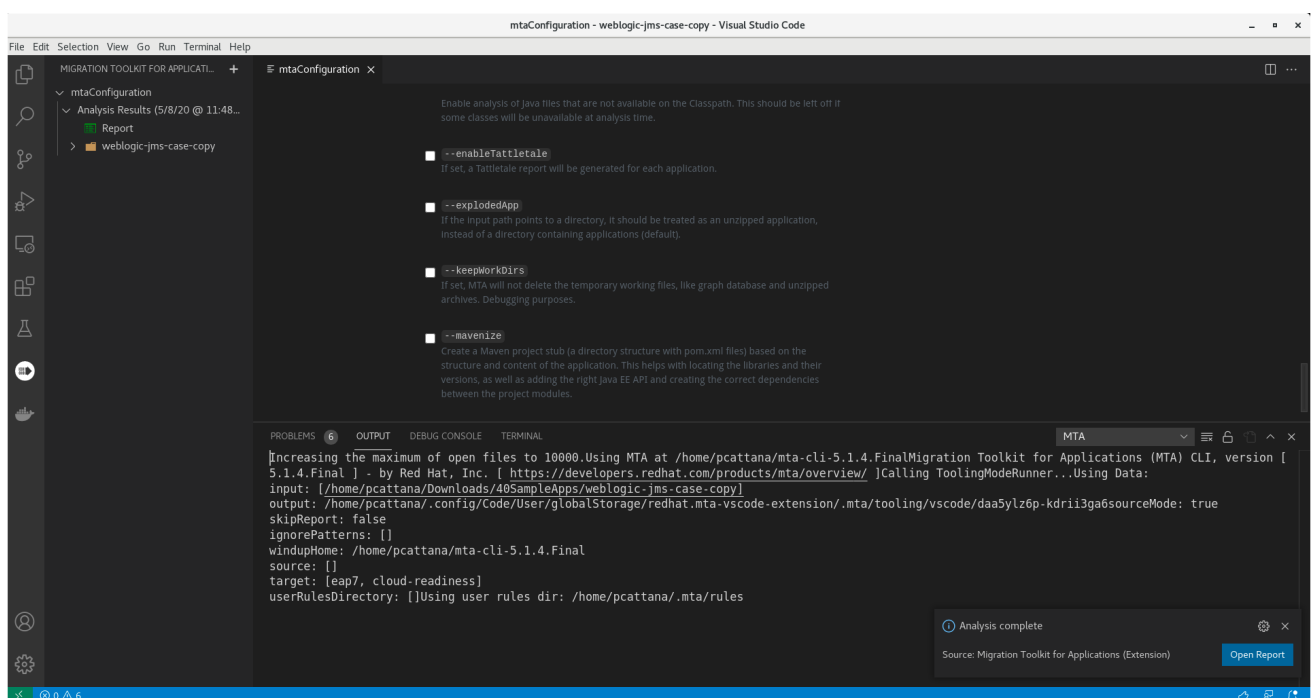
You can analyze your projects with the MTA extension by creating a run configuration and running an analysis.

3.1. MTA EXTENSION INTERFACE

The interface of the Migration Toolkit for Applications (MTA) 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 MTA report in your browser. Beneath the report icon are the other elements of the tree: the applications analyzed by MTA, 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 MTA report, which describes any issues you need to address before you migrate or modernize your application. For more information, see [MTA report](#) in the *CLI Guide*.

Figure 3.1. MTA extension interface




3.2. CONFIGURING A RUN CONFIGURATION

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

Prerequisites

- **mta-cli** executable installed. You can download the **mta-cli** executable from [MTA download](#).

Procedure

1. In **Extensions** view, click the Migration Toolkit for Applications icon () on the Activity bar.
2. Click the + (plus sign) next to **Migration Toolkit for Applications** to add a run configuration.
3. Complete the following configuration fields:
 - **Name:** Enter a meaningful name for the analysis configuration or accept the default.
 - **mta-cli:** Enter the path to the **mta-cli** executable. For example: **\$HOME/mta-cli-5.3.0.Final/bin/mta-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 MTA report, which describes any issues you need to address before you migrate or modernize your application. For more information, see [MTA report](#) in the *CLI Guide*.




CHAPTER 4. REVIEWING AND RESOLVING MIGRATION ISSUES

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

4.1. REVIEWING ISSUES

You can use the MTA 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**.