Identify and resolve migration issues by analyzing your applications with the Migration Toolkit for Applications plugin for IntelliJ IDEA.
Identify and resolve migration issues by analyzing your applications with the Migration Toolkit for Applications plugin for IntelliJ IDEA.
Abstract

This guide describes how to use the Migration Toolkit for Applications plugin for IntelliJ IDEA to accelerate large-scale application modernization efforts across hybrid cloud environments on Red Hat OpenShift.
Table of Contents

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

CHAPTER 1. INTRODUCTION .............................................................................. 4
  1.1. ABOUT THE MTA PLUGIN FOR INTELLIJ IDEA ...................................... 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

CHAPTER 2. INSTALLING THE MTA PLUGIN FOR INTELLIJ IDEA ...................... 6

CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA PLUGIN ................. 7
  3.1. CREATING A RUN CONFIGURATION ..................................................... 7

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 PLUGIN FOR INTELLIJ IDEA

You can migrate and modernize applications by using the Migration Toolkit for Applications (MTA) plugin for IntelliJ IDEA.

The MTA plugin 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 plugin supports both the Community Edition and the Ultimate version of IntelliJ IDEA.

1.2. ABOUT THE MIGRATION TOOLKIT FOR APPLICATIONS

What is the Migration Toolkit for Applications?

Migration Toolkit for Applications (MTA) 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.

MTA 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.

MTA 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.

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:

- 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 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.
CHAPTER 2. INSTALLING THE MTA PLUGIN FOR INTELLIJ IDEA

You can install the MTA plugin in the Ultimate and the Community Edition releases of IntelliJ IDEA.

Prerequisites

- Java Development Kit (JDK) installed. MTA 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.
- The latest version of `mta-cli` from the MTA download page

Procedure

1. In IntelliJ IDEA, click the Plugins tab on the Welcome screen.
2. Enter Migration Toolkit for Applications in the Search field on the Marketplace tab.
3. Select the Migration Toolkit for Applications (MTA) by Red Hat plugin and click Install. The plugin is listed on the Installed tab.
CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA PLUGIN

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

3.1. CREATING A RUN CONFIGURATION

You can create multiple run configurations to run against each project you import to IntelliJ IDEA.

Procedure

1. In the Projects view, click the project you want to analyze.

2. On the left side of the screen, click the Migration Toolkit for Applications tab. If this is your first configuration, the run configuration panel is displayed on the right.

3. If this is not your first configuration, right-click configuration in the list and select New configuration. The run configuration panel is displayed on the right.

4. Complete the following configuration fields:
   - **cli**: Enter the path to the cli executable. For example: $HOME/mta-cli-6.2.1.GA-redhat/bin/windup-cli.
   - **Input**: Click Add and enter the input file or directory.
   - **Target**: Select one or more target migration paths.

      NOTE

      The location shown in the Output is set by the plugin.

5. In the list of configurations, right-click the new configuration and select Run Analysis. The Console (MTA) terminal emulator opens, displaying information about the progress of the analysis.

When the analysis is completed, you can click either Report or Results below the name of the configuration file you ran.

- **Reports** opens the MTA 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.

- **Results** opens a directory displaying hints (issues) per application.
CHAPTER 4. REVIEWING AND RESOLVING MIGRATION ISSUES

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

4.1. REVIEWING ISSUES

You can use the MTA plugin icons to prioritize issues based on their severity. You can see which issues have a Quick Fix automatic code replacement and which do not.

The results of an analysis are displayed in a directory format, showing the hints and classifications for each application analyzed.

A hint is a read-only snippet of code that contains a single issue that you should or must address before you can modernize or migrate an application. Often a Quick Fix is suggested, which you can accept or ignore.

A classification is a file that has an issue but does not have any suggested Quick Fixes. You can edit a classification.

Procedure

1. In the Migration Toolkit for Applications view, select a run configuration directory in the left pane.

2. Click Results.
   The modules and applications of the run configuration are displayed, with hints and classifications beneath each application.

3. Prioritize issues based on the following icons, which are displayed next to each hint:
   - !: You must fix this issue in order to migrate or modernize the application.
   - ⚠: You might need to fix this issue in order to migrate or modernize the application.

4. Optional: To learn more about a hint, right-click it and select Show More Details.

4.2. RESOLVING ISSUES

You can resolve issues by doing one of the following:

- Using a Quick Fix to fix a code snippet that has a hint
- Editing the code of a file that appears in a classification

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. Quick Fixes are available for many issues displayed in the Hints section of the Results directory.

Procedure
1. In the left pane, click a hint that has an error indicator. Any Quick Fixes are displayed as child folders with the Quick Fix icon (ileges) on their left side.

2. Right-click a Quick Fix and select **Preview Quick Fix**. The current code and the suggested change are displayed in the **Preview Quick Fix** window.

3. To accept the suggested fix, click **Apply Quick Fix**.

4. Optional: Right-click the issue and select **Mark As Complete**. A green check ( GameManager) is displayed by the hint, replacing the error indicator.

**4.2.2. Editing the code of a file**

You can directly edit a file displayed in the **Classifications** section of the **Results** directory. These files do not have any Quick Fixes.

**Procedure**

1. In the left pane, click the file you want to edit.

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**. If you select **Mark as Complete**, a green check ( GameManager) is displayed by the hint, replacing the error indicator.