



Migration Toolkit for Applications 6.1

Eclipse and Red Hat CodeReady Studio Guide

Identify and resolve migration issues by analyzing your applications with the MTA plugin for Eclipse or Red Hat CodeReady Studio.

Migration Toolkit for Applications 6.1 Eclipse and Red Hat CodeReady Studio Guide

Identify and resolve migration issues by analyzing your applications with the MTA plugin for Eclipse or Red Hat CodeReady Studio.

Legal Notice

Copyright © 2023 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 MTA plugin for Eclipse or Red Hat CodeReady Studio 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 ECLIPSE AND RED HAT CODEREADY STUDIO	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	6
2.1. INSTALLING IN A CONNECTED ENVIRONMENT	6
2.2. INSTALLING IN A CONNECTED ENVIRONMENT	7
2.3. ACCESSING THE MTA TOOLS	7
CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA PLUGIN	9
3.1. CREATING A RUN CONFIGURATION	9
3.2. ANALYZING PROJECTS	9
3.3. REVIEWING ISSUES	10
3.4. RESOLVING ISSUES	11
CHAPTER 4. MANAGING RULES	12
4.1. VIEWING RULES	12
4.2. CREATING A CUSTOM RULESET	12
4.3. IMPORTING A CUSTOM RULESET	13
4.4. SUBMITTING A CUSTOM RULESET	13

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 ECLIPSE AND RED HAT CODEREADY STUDIO

You can migrate and modernize applications by using the Migration Toolkit for Applications (MTA) plugin for Eclipse and Red Hat CodeReady Studio.

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.

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

You can install the MTA plugin in a connected or a restricted network environment.

2.1. INSTALLING IN A CONNECTED ENVIRONMENT

You need a connected environment to install the MTA plugin.

The MTA plugin has been tested with the Eclipse IDE for Java Enterprise Developers 2022-03 and Red Hat CodeReady Studio 12.21.3.GA.

Prerequisites

- Java Development Kit (JDK) 11 is installed and applied as the default. MTA supports the following JDKs:
 - OpenJDK 11
 - Oracle JDK 11
 - Eclipse Temurin™ JDK 11
- The Java Compiler compliance level is set to 11.
- 8 GB RAM
- macOS installation: the value of **maxproc** must be **2048** or greater.
- [Red Hat CodeReady Studio](#) or [Eclipse IDE for Java Enterprise Developers 2022-03](#)
- JBoss Tools, installed from the [Eclipse Marketplace](#)
- [Mylyn SDK and frameworks](#), installed with Eclipse

Procedure

1. Launch Eclipse or CodeReady Studio.
2. From the menu bar, select **Help** → **Install New Software**.
3. Next to the **Work with** field, click **Add**.
4. In the **Name** field, enter **MTA**.
5. In the **Location** field, enter <https://marketplace.eclipse.org/content/migration-toolkit-applications-mta> and click **OK**.
6. Select all the **JBoss Tools - MTA** check boxes and click **Next**.
7. Review the installation details and click **Next**.
8. Accept the terms of the license agreement and click **Finish**.
9. Restart Eclipse or CodeReady Studio.

2.2. INSTALLING IN A CONNECTED ENVIRONMENT

You need a connected environment to install the MTA plugin.

The MTA plugin has been tested with the Eclipse IDE for Java Enterprise Developers 2022-03 and Red Hat CodeReady Studio 12.21.3.GA.

Prerequisites

- Java Development Kit (JDK) 11 is installed and applied as the default. MTA supports the following JDKs:
 - OpenJDK 11
 - Oracle JDK 11
 - Eclipse Temurin™ JDK 11
- The Java Compiler compliance level is set to 11.
- 8 GB RAM
- macOS installation: the value of **maxproc** must be **2048** or greater.
- [Red Hat CodeReady Studio](#) or [Eclipse IDE for Java Enterprise Developers 2022-03](#)
- JBoss Tools, installed from the [Eclipse Marketplace](#)
- [Mylyn SDK and frameworks](#), installed with Eclipse

Procedure

1. Launch Eclipse or CodeReady Studio.
2. From the menu bar, select **Help → Install New Software**.
3. Next to the **Work with** field, click **Add**.
4. In the **Name** field, enter **MTA**.
5. In the **Location** field, enter <https://marketplace.eclipse.org/content/migration-toolkit-applications-mta> and click **OK**.
6. Select all the **JBoss Tools - MTA** check boxes and click **Next**.
7. Review the installation details and click **Next**.
8. Accept the terms of the license agreement and click **Finish**.
9. Restart Eclipse or CodeReady Studio.

2.3. ACCESSING THE MTA TOOLS

You can access the MTA plugin tools in the **MTA** perspective.

Prerequisites

- You must restart the Eclipse IDE or Red Hat CodeReady Studio after installing the MTA plugin.

Procedure

1. Click **Window** → **Perspective** → **Open Perspective** → **Other**.

2. Select **MTA** and click **OK**.

The following components are displayed:

- **Issue Explorer** displays the migration issues identified by the MTA plugin.
- **MTA Server** is a separate process that analyzes projects, flags migration issues, and generates reports.
You can start, stop, and view the status of the MTA server in the **Issue Explorer**.
- **Issue Details** displays detailed information about a selected issue, including the hint, severity, and any additional resources.
- **MTA Report** is an HTML report generated by the MTA plugin. From the report landing page you can navigate to detailed reports, such as Application Details, Issues, and Dependencies.



NOTE

The report is not generated by default. You must select the **Generate Report** option in the run configuration.

CHAPTER 3. ANALYZING YOUR PROJECTS WITH THE MTA PLUGIN

You can analyze your projects with the MTA plugin by creating a run configuration, running an analysis, and then reviewing and resolving migration issues detected by the MTA plugin.

3.1. CREATING A RUN CONFIGURATION


You can create a run configuration in the **Issue Explorer**. A run configuration specifies the project to analyze, migration path, and additional options.

You can create multiple run configurations. Each run configuration must have a unique name.

Prerequisite

- You must import your projects into the Eclipse IDE or CodeReady Studio.

Procedure

1. In the **Issue Explorer**, click the MTA icon () to create a run configuration.
2. On the **Input** tab, complete the following fields:
 - a. Select a migration path.
 - b. Beside the **Projects** field, click **Add** and select one or more projects.
 - c. Beside the **Packages** field, click **Add** and select one or more the packages.



NOTE


Specifying the packages for analysis reduces the run time. If you do not select any packages, all packages in the project are scanned.

3. On the **Options** tab, you can select **Generate Report** to generate an HTML report. The report is displayed in the **Report** tab and saved as a file. Other options are displayed. See [About MTA command-line arguments](#) in the *CLI Guide* for details.
4. On the **Rules** tab, you can select custom rulesets that you have imported or created for the MTA plugin.
5. Click **Run** to start the analysis.

3.2. ANALYZING PROJECTS

You can analyze your projects by running the MTA plugin with a saved run configuration.

Procedure

1. In the **MTA** perspective, click the **Run** button () and select a run configuration.


The MTA plugin analyzes your projects. The **Issue Explorer** displays migration issues that are detected with the ruleset.

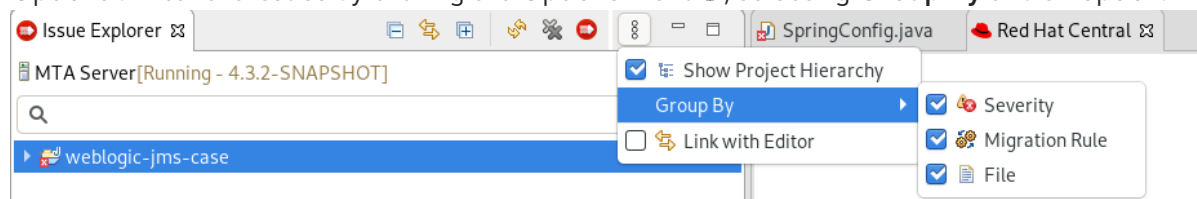
- When you have finished analyzing your projects, stop the MTA server in the **Issue Explorer** to conserve memory.

3.3. REVIEWING ISSUES

You can review issues identified by the MTA plugin.

Procedure








- Click **Window** → **Show View** → **Issue Explorer**.
- Optional: Filter the issues by clicking the Options menu , selecting **Group By** and an option.




- Right-click and select **Issue Details** to view information about the issue, including its severity and how to address it.

The following icons indicate the severity and state of an issue:

Table 3.1. Issue icons





Icon	Description
	The issue must be fixed for a successful migration.
	The issue is optional to fix for migration.
	The issue might need to be addressed during migration.
	The issue was resolved.
	The issue is stale. The code marked as an issue was modified since the last time that MTA identified it as an issue.
	A quick fix is available for this issue, which is mandatory to fix for a successful migration.
	A quick fix is available for this issue, which is optional to fix for migration.

Icon	Description
	A quick fix is available for this issue, which may potentially be an issue during migration.

4. Double-click an issue to open the associated line of code in an editor.

3.4. RESOLVING ISSUES

You can resolve issues detected by the MTA plugin by performing one of the following actions:

- You can double-click the issue to open it in an editor and edit the source code.
The issue displays a Stale icon () until the next time you run the MTA plugin.
- You can right-click the issue and select **Mark as Fixed**.
- If the issue displays a Quick Fix icon (  ), you can right-click the issue and select **Preview Quick Fix** and then **Apply Quick Fix**.

CHAPTER 4. MANAGING RULES

The MTA plugin comes with a core set of **System** rules for analyzing projects and identifying migration and modernization issues.

You can create and import custom rulesets.

4.1. VIEWING RULES

You can view system and custom rules, if any, for the MTA plugin.

Prerequisites

- To view system rules, the MTA server must be running.

Procedure


1. Click the **Rulesets** tab.
2. Expand **System** to view system rulesets or **Custom** to view custom rulesets.
3. Expand a ruleset.
4. Double-click a rule to open it in a viewer.
5. Click the **Source** tab to view the XML source of the rule.

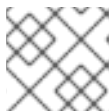
4.2. CREATING A CUSTOM RULESET

You can create a custom ruleset in the **MTA** perspective.

See the [Rules Development Guide](#) to learn more about creating custom XML rules.

Procedure

1. Click the **Rulesets** tab.
2. Click the Create Ruleset icon ().
3. Select a project and a directory for the ruleset.
4. Enter the file name.



NOTE

The file must have the extension **.windup.xml**.

5. Enter a ruleset ID, for example, **my-ruleset-id**.
6. Optional: Select **Generate quickstart template** to add basic rule templates to the file.
7. Click **Finish**.
8. The ruleset file opens in an editor and you can add and edit rules in the file.

9. Click the **Source** tab to edit the XML source of the ruleset file.

You can select the new ruleset when you create a run configuration.


4.3. IMPORTING A CUSTOM RULESET

You can import a custom ruleset into the MTA plugin to analyze your projects.

Prerequisites

- Custom ruleset file with a **.windup.xml** extension.
See the [Rules Development Guide](#) for information about creating rulesets.


Procedure

1. Click the **Rulesets** tab.
2. Click the Import Ruleset icon ().
3. Browse to and select the XML rule file to import.
The custom ruleset is displayed when you expand **Custom** on the **Rulesets** tab.

4.4. SUBMITTING A CUSTOM RULESET

You can submit your custom ruleset for inclusion in the official MTA rule repository. This allows your custom rules to be reviewed and included in subsequent releases of MTA.

Procedure

1. Click the **Rulesets** tab.
2. Click the Arrow icon () and select **Submit Ruleset**.
3. Complete the following fields:
 - **Summary:** Describe the purpose of the rule. This becomes the title of the submission.
 - **Code Sample:** Enter an example of the source code that the rule should run against.
 - **Description:** Enter a brief description of the rule.
4. Click **Choose Files** and select the ruleset file.
5. Click **Submit**.