



Migration Toolkit for Applications 6.1

CLI ガイド

Migration Toolkit for Applications CLI を使用してアプリケーションを移行する方法を説明します。

Migration Toolkit for Applications 6.1 CLI ガイド

Migration Toolkit for Applications CLI を使用してアプリケーションを移行する方法を説明します。

法律上の通知

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.

概要

本ガイドでは、Migration Toolkit for Applications CLI を使用して Java アプリケーションの移行を単純化する方法を説明します。

目次

多様性を受け入れるオープンソースの強化	3
第1章 はじめに	4
1.1. CLI ガイドについて	4
1.2. MIGRATION TOOLKIT FOR APPLICATIONS の概要	4
1.3. CLI について	5
第2章 CLI のインストールおよび実行	6
2.1. CLI のインストール	6
2.2. CLI の実行	6
2.3. レポートへのアクセス	10
第3章 レポートの確認	11
3.1. アプリケーションレポート	12
3.2. TECHNOLOGIES レポート	18
3.3. 複数のアプリケーションで共有されるアーカイブ	19
3.4. ルールプロバイダー実行の概要	19
第4章 CSV 形式でのレポートのエクスポート	21
4.1. レポートのエクスポート	21
4.2. CSV ファイルのスプレッドシートプログラムへのインポート	21
4.3. CSV データ構造について	21
第5章 アプリケーションの MAVEN 化	23
5.1. MAVEN プロジェクト構造の生成	23
5.2. MAVEN プロジェクト構造の確認	23
第6章 MTA パフォーマンスの最適化	26
6.1. アプリケーションのデプロイおよび実行	26
6.2. ハードウェアのアップグレード	26
6.3. パッケージおよびファイルを除外する MTA の設定	26
付録A 参考資料	28
A.1. MTA コマンドライン引数	28
A.2. サポート対象のテクノロジータグ	35
A.3. ルールのストーリーポイントについて	48
A.4. 関連情報	49

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 はじめに

1.1. CLI ガイドについて

Java アプリケーションやその他のコンポーネントを移行するために、Migration Toolkit for Applications (MTA) を使用するエンジニア、コンサルタント、およびその他のユーザーを対象としています。CLI のインストールおよび実行方法、生成されたレポートを確認し、追加機能を活用する方法を説明します。

1.2. MIGRATION TOOLKIT FOR APPLICATIONS の概要

Migration Toolkit for Applications とは

Migration Toolkit for Applications (MTA) は、Red Hat OpenShift 上のハイブリッドクラウド環境全体で大規模なアプリケーションモダナイゼーションに対する取り組みを促進します。このソリューションは、ポートフォリオレベルとアプリケーションレベルの両方で、採用プロセス全体を通じて洞察を提供します。つまり、アプリケーションのインベントリ、評価、分析、および管理を行い、ユーザーインターフェイスを介して OpenShift への移行をより短い期間で行うことができます。

MTA は、アプリケーションを評価するための基礎として広範にわたる質問リストを使用して、コンテナ化のアプリケーションの準備に必要な難易度、時間、およびその他のリソースを見積もることができますようにします。ステークホルダー間の議論の基礎として評価の結果を使用して、どのアプリケーションがコンテナ化に適しているか、どのアプリケーションが最初に多大な作業を必要とするか、どのアプリケーションがコンテナ化に適していないかを判断できます。

MTA は、対象のアプリケーションごとに1つ以上のルールセットを適用してアプリケーションを分析し、モダナイゼーションする前に、そのアプリケーションに含まれるどの行を変更するかを判断します。

MTA は、プロジェクトソースディレクトリーやアプリケーションアーカイブを含むアプリケーションアーティファクトを検査し、変更を必要とするエリアを強調表示する HTML レポートを作成します。MTA は、以下を含む多くの移行パスをサポートします。

- Red Hat JBoss Enterprise Application Platform の最新リリースへのアップグレード
- Oracle WebLogic または IBM WebSphere Application Server から Red Hat JBoss Enterprise Application Platform への移行
- アプリケーションのコンテナ化とクラウド化
- Java Spring Boot から Quarkus への移行
- OpenJDK 8 から OpenJDK 11 へのアップグレード
- OpenJDK 11 から OpenJDK 17 へのアップグレード
- EAP Java アプリケーションの Azure App Service への移行
- Spring Boot Java アプリケーションの Azure App Service への移行

ユースケースおよび移行パスの詳細は、[開発者向け MTA Web ページ](#)を参照してください。

Migration Toolkit for Applications による移行を単純化する方法

Migration Toolkit for Applications は一般的なリソースを探し、アプリケーションを移行する際の既知の問題点を明らかにします。アプリケーションで使用されるテクノロジーの概要を示します。

MTA は、移行またはモダナイゼーションパスの評価に関する詳細なレポートを生成します。このレポートは、大規模なプロジェクトに必要な作業を見積もり、関係する作業を減らすのに役立ちます。

1.3. CLI について

CLI は、Migration Toolkit for Applications のコマンドラインツールで、アプリケーションの移行と最新化の取り組みを評価し、優先順位を付けることができます。他のツールのオーバーヘッドなしに分析を強調表示する多数のレポートが提供されます。CLI にはさまざまなカスタマイズオプションが含まれており、MTA 分析オプションを細かく調整したり、外部の自動化ツールと統合したりできます。

第2章 CLI のインストールおよび実行

2.1. CLI のインストール

CLI は、Linux、Windows、または macOS オペレーティングシステムにインストールできます。

前提条件

- Java Development Kit(JDK) がインストールされている。MTA が以下の JDK をサポートしている
 - OpenJDK 11
 - OpenJDK 17
 - Oracle JDK 11
 - Oracle JDK 17
 - Eclipse Temurin™ JDK 11
 - Eclipse Temurin™ JDK 17
- 8 GB RAM
- macOS のインストール: **maxproc** の値は **2048** 以上である。

手順

1. [MTA Download ページ](#) に移動し、**Migration Toolkit CLI** ファイルをダウンロードします。
2. **.zip** ファイルを任意のディレクトリーにデプロイメントします。



注記

Windows オペレーティングシステムにインストールする場合は、以下を行います。

1. **.zip** ファイルを **LC_PSN** という名前のディレクトリーに展開し、**Path too long** エラーを回避します。または、[7-Zip](#) のファイルを、任意の名前のディレクトリーに展開します。
2. 抽出中に **Confirm file replace** ウィンドウが表示されている場合は、**Yes to all** をクリックします。

本ガイドで **<MTA_HOME>** が発生した場合は、これを MTA インストールへの実際のパスに置き換えます。

2.2. CLI の実行

アプリケーションに対して MTA を実行できます。

手順

1. ターミナルを開き、`<MTA_HOME>/bin/` ディレクトリーに移動します。
2. Windows の場合は **windup-cli** スクリプトまたは **windup-cli.bat** を実行し、適切な引数を指定します。

```
$ ./windup-cli --input /path/to/jee-example-app-1.0.0.ear \
  --output /path/to/output --source weblogic --target eap:6 \
  --packages com.acme org.apache
```

- **--input**: 評価されるアプリケーション。
- **--output**: 生成されたレポートの出力ディレクトリー。
- **--source**: アプリケーション移行元のテクノロジー。
- **--target**: アプリケーション移行先のテクノロジー。
- **--packages**: 評価されるパッケージ。この引数は、パフォーマンスを改善するために強く推奨されます。

3. レポートにアクセスします。

2.2.1. MTA コマンドの例

アプリケーションアーカイブでの MTA の実行

次のコマンドは、JBoss EAP 5 から JBoss EAP 7 に移行するための サンプル EAR アーカイブ [jee-example-app-1.0.0.ear](#) の **com.acme** パッケージおよび **org.apache** パッケージを解析します。

```
$ <MTA_HOME>/bin/windup-cli \
  --input /path/to/jee-example-app-1.0.0.ear \
  --output /path/to/report-output/ --source eap:5 --target eap:7 \
  --packages com.acme org.apache
```

ソースコードでの MTA の実行

次のコマンドは、JBoss EAP 6 に移行するための [seam-booking-5.2](#) サンプルソースコードの **org.jboss.seam** パッケージを分析します。

```
$ <MTA_HOME>/bin/windup-cli --sourceMode --input /path/to/seam-booking-5.2/\
  --output /path/to/report-output/ --target eap:6 --packages org.jboss.seam
```

cloud-readiness ルールの実行

次のコマンドは、JBoss EAP 7 に移行するためのサンプル EAR アーカイブ [jee-example-app-1.0.0.ear](#) の **com.acme** パッケージおよび **org.apache** パッケージを解析します。また、クラウドの準備ができるかどうかを評価します。

```
$ <MTA_HOME>/bin/windup-cli --input /path/to/jee-example-app-1.0.0.ear \
  --output /path/to/report-output/\
  --target eap:7 --target cloud-readiness --packages com.acme org.apache
```

MTA プロパティーの上書き

デフォルトの Fernflower デコンパイラーを上書きするには、コマンドラインで **-Dwindup.decompiler** 引数を渡します。たとえば、Procyon デコンパイル機能を使用する場合は、以下の構文を使用します。

```
$ <MTA_HOME>/bin/windup-cli -Dwindup.decompiler=procyon \
  --input <INPUT_ARCHIVE_OR_DIRECTORY> --output <OUTPUT_REPORT_DIRECTORY> \
  --target <TARGET_TECHNOLOGY> --packages <PACKAGE_1> <PACKAGE_2>
```

2.2.2. MTA CLI Bash の完了

MTA CLI は、Linux システムの Bash 補完を有効にするオプションを提供し、コマンドを入力した時に Tab キーを押して MTA コマンドライン引数を自動化できます。たとえば、bash 補完を有効にすると、以下のように利用可能な引数のリストが表示されます。

```
$ <MTA_HOME>/bin/windup-cli [TAB]
```

bash 補完の有効化

現在のシェルで bash 補完を有効にするには、以下のコマンドを実行します。

```
$ source <MTA_HOME>/bash-completion/windup-cli
```

永続的な bash 補完の有効化

以下のコマンドを使用すると、再起動後も bash 補完が維持されます。

- システムを再起動しても特定のユーザーの bash 補完を有効にするには、そのユーザーの `~/.bashrc` ファイルに以下の行を追加します。

```
source <MTA_HOME>/bash-completion/windup-cli
```

- システムの再起動後すべてのユーザーの bash 補完を有効にするには、root ユーザーで Migration Toolkit for Applications CLI Bash 補完ファイルを `/etc/bash_completion.d/` ディレクトリにコピーします。

```
# cp <MTA_HOME>/bash-completion/windup-cli /etc/bash_completion.d/
```

2.2.3. MTA ヘルプへのアクセス

`windup-cli` コマンドで利用可能な引数の完全な一覧を表示するには、ターミナルを開き、`<MTA_HOME>` ディレクトリに移動して、以下のコマンドを実行します。

```
$ <MTA_HOME>/bin/windup-cli --help
```

2.2.4. OpenRewrite レシピの使用



重要

OpenRewrite レシピのサポートはテクノロジープレビュー機能としてのみ提供されません。テクノロジープレビュー機能は、Red Hat の実稼働環境でのサービスレベルアグリメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

テクノロジープレビュー機能のサポート範囲については、Red Hat カスタマーポータル [のテクノロジープレビュー機能のサポート範囲](#) を参照してください。

MTA CLI で [OpenRewrite](#) レシピを使用すると、Java アプリケーションのソースコードをリファクタリングできます。

たとえば、OpenRewrite レシピの `org.jboss.windup.JavaxToJakarta` は、インポートされた `javax` パッケージの名前を同等の `jakarta` に変更します。

手順

1. `windup-cli` を実行し、レシピ名、設定ファイルへのパス、およびアプリケーションを指定します。

```
$ ./windup-cli --openrewrite --input </path/to/source/project> \
  "-Drewrite.configLocation=<path/to/rewrite.yaml>" \
  "-DactiveRecipes=<recipe_name>" --goal dryRun
```

- `"-DactiveRecipes=<recipe name>":` OpenRewrite recipe を指定します (例: `org.jboss.windup.JavaxToJakarta`)。
- `--input:` リファクタリングするアプリケーションを指定します。アプリケーションは、Maven Project Object Model (POM) XML ファイル `pom.xml` を含むソースコードプロジェクトの最上位にある必要があります。
- `-Drewrite.configLocation=<path/to/rewrite.yaml>:` 使用する `rewrite.yaml` 設定ファイルの場所。同梱の `rewrite.yaml` 設定ファイルは、`<MTA_HOME>/rules/openrewrite` サブフォルダーにあります (例: `"-Drewrite.configLocation=<MTA_HOME>/rules/openrewrite/jakarta/javax/imports/rewrite.yaml"`)。
- `"-DactiveRecipes=<recipe name> >":` OpenRewrite recipe を指定します (例: `org.jboss.windup.JavaxToJakarta`)。
`activeRecipes` パラメーターでそれぞれを指定すると、複数のレシピを含めることができます。たとえば、レシピ `org.jboss.windup.JavaxInjectToJakartaInject` と `org.jboss.windup.JavaxEjbToJakartaEjb` を含めるには、`"-DactiveRecipes=<recipe name>"` で以下を入力します。

```
"-DactiveRecipes=org.jboss.windup.JavaxInjectToJakartaInject, \
  org.jboss.windup.JavaxEjbToJakartaEjb"
```

- `--goal:` オプション: 実行する OpenRewrite Maven ゴール。
 - `dryRun:` スクリプトは、提案される変更のリストを返します。"`Run 'mvn rewrite:run' to apply the recipes`" メッセージは無視します。
 - `run:` スクリプトは変更を適用します。
2. `--goal run` を指定して `windup-cli` を実行し、レシピを適用します。

```
$ ./windup-cli --openrewrite --input </path/to/source/project> \
  "-Drewrite.configLocation=<path/to/rewrite.yaml>" \
  "-DactiveRecipes=<recipe_name>" --goal run
```

2.2.4.1. 利用可能な OpenRewrite レシピ

表2.1 利用可能な OpenRewrite レシピ

移行パス	目的	rewrite.configLocation	activeRecipes
Java EE から Jakarta EE	<p>javax パッケージのインポートを同等の jakarta パッケージに置き換える</p> <p>pom.xml ファイル内で宣言された javax アーティファクトを jakarta と同等のものに置き換える</p>	<MTR_HOME>/rules/opensslrewrite/jakarta \ /javax/imports/rewrite.yml	org.jboss.windup.JavaxToJakarta
Java EE から Jakarta EE	ブートストラップファイルの名前を変更する	<MTR_HOME>/rules/opensslrewrite/jakarta \ /javax/bootstrapping/rewrite.yml	org.jboss.windup.java.rta.javax. \ BootstrappingFiles
Java EE から Jakarta EE	persistence.xml 設定を変換する	<MTR_HOME>/rules/opensslrewrite/jakarta \ /javax/xml/rewrite.yml	org.jboss.windup.java.x-jakarta. \ PersistenceXML
Spring Boot から Quarkus	<p>application*.properties に一致するファイル内の spring.jpa.hibernate.ddl-auto プロパティを置き換える</p>	<MTR_HOME>/rules/opensslrewrite/quarkus \ /springboot/properties/rewrite.yml	org.jboss.windup.sb-quarkus.Properties

2.3. レポートへのアクセス

Migration Toolkit for Applications を実行すると、コマンドラインの **--output** 引数を使用して指定する **<OUTPUT_REPORT_DIRECTORY>** にレポートが生成されます。

output ディレクトリーには、以下のファイルおよびサブディレクトリーが含まれます。

```

<OUTPUT_REPORT_DIRECTORY>/
├── index.html      // Landing page for the report
├── <EXPORT_FILE>.csv // Optional export of data in CSV format
├── archives/      // Archives extracted from the application
├── mavenized/     // Optional Maven project structure
├── reports/       // Generated HTML reports
└── stats/         // Performance statistics

```

手順

- MTA の実行後に表示される出力から、レポートの **index.html** ファイルのパスを取得します。

```

Report created: <OUTPUT_REPORT_DIRECTORY>/index.html
Access it at this URL: file:///<OUTPUT_REPORT_DIRECTORY>/index.html

```

- ブラウザーを使用して **index.html** ファイルを開きます。
生成されたレポートが表示されます。

第3章 レポートの確認

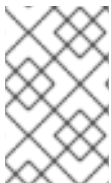
次のセクションに示すレポートの例は、MTA GitHub ソースリポジトリにある [jee-example-app-1.0.0.ear](#) サンプルアプリケーションの **com.acme** パッケージおよび **org.apache** パッケージを分析した結果です。

以下のコマンドを使用してレポートが生成されました。

```
$ <MTA_HOME>/bin/windup-cli --input /home/username/windup-cli-source/test-files/jee-example-app-1.0.0.ear/ --output /home/username/windup-cli-reports/jee-example-app-1.0.0.ear-report --target eap:6 --packages com.acme org.apache
```

ブラウザを使用して、レポート出力ディレクトリーにある **index.html** ファイルを開きます。これにより、処理されたアプリケーションのリストが表示されます。各行には、ストーリーポイント、インシデントの数、アプリケーションで発生したテクノロジーの概要が含まれます。

図3.1 アプリケーションリスト



注記

新しいルールが MTA に追加されると、インシデントと予測されるストーリーポイントが変わります。この値は、このアプリケーションをテストする際に表示される値と一致しない場合があります。

以下の表は、このメインの MTA ランディングページからアクセスできるレポートおよびページのリストです。アプリケーションの名前 **jee-example-app-1.0.0.ear** をクリックして、アプリケーションレポートを表示します。

ページ	アクセス方法
アプリケーション	アプリケーションの名前をクリックします。
Technologies レポート	ページ上部にある Technologies リンクをクリックします。
複数のアプリケーションで共有されるアーカイブ	Archives shared by multiple applications リンクをクリックします。このリンクは、複数のアプリケーションに共有アーカイブがある場合にのみ利用できることに注意してください。

ページ	アクセス方法
ルールプロバイダー実行の概要	ページの下部にある Rule providers execution overview リンクをクリックします。

アプリケーションが、他の分析済みアプリケーションとアーカイブを共有している場合は、共有アーカイブからのストーリーポイントの数と、このアプリケーションに固有のストーリーポイントの数が表示されることに注意してください。

図3.2 共有アーカイブ

Application: Archives shared by multip... ▾

Issues
This report provides a concise summary of all issues identified.

Search: Category ▾ Level effort ▾ Source ▾ Target ▾ [Export CSV](#) 1 - 2 of 2 < >

Issue	Category	Source technolo...	Target technolo...	Level of effort	Total incidents	Total storypoi...
> Embedded framework - AOP Alliance	information			Info	1	0
> Embedded library - Apache Commons Logging	information			Info	1	0

1 - 2 of 2 << < 1 of 1 > >>

アプリケーション間で共有されるアーカイブに関する情報は、複数のアプリケーションで共有されるアーカイブレポートを参照してください。

3.1. アプリケーションレポート

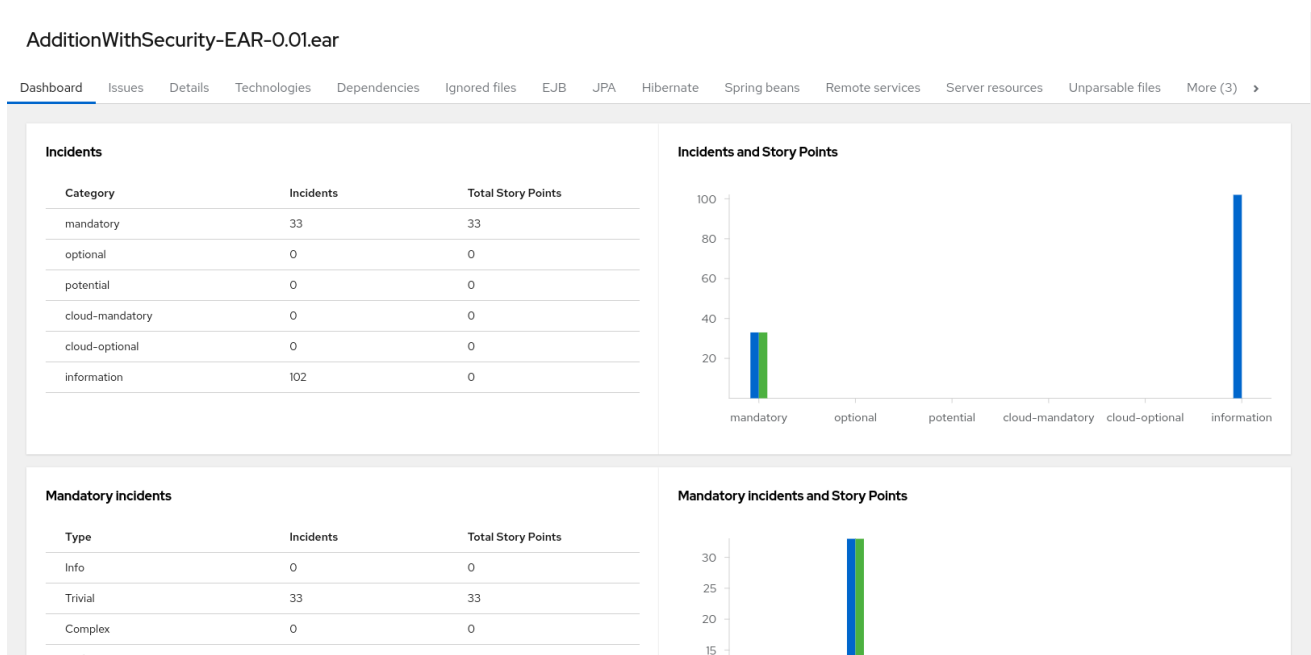
3.1.1. ダッシュボード

アプリケーションのリストでアプリケーション名をクリックして、レポートランディングページからこのレポートにアクセスします。

ダッシュボードでは、アプリケーションの移行作業全体の概要が表示されます。つまり、以下のようになります。

- カテゴリー別のインシデントおよびストーリーポイント
- 提案された変更の作業レベル別のインシデントおよびストーリーポイント
- パッケージ別のインシデント

図3.3 ダッシュボード



上部のナビゲーションバーには、このアプリケーションの移行に関する追加の詳細を含むさまざまなレポートがリスト表示されます。現在のアプリケーションに適用可能なレポートのみが利用できるように注意してください。

レポート	説明
Issues	注意が必要な問題すべてについて簡潔に説明します。
Application Details	移行時に注意する必要がある可能性のあるアプリケーション内で見つかったすべてのリソースの詳細を説明します。
Technologies	機能を基にグループ化されたすべての埋め込みライブラリーを表示します。これにより、各アプリケーションで使用されるテクノロジーを迅速に表示できます。
依存関係	アプリケーション内にある Java パッケージの依存関係をすべて表示します。
Unparsable	MTA が想定される形式で解析できなかったすべてのファイルを示しています。たとえば、 .xml または .wsdl 接尾辞が含まれるファイルは XML ファイルであると仮定します。XML パーサーが失敗すると、問題はここに報告され、また個々のファイルがリストされている場所も報告されます。
Remote Services	アプリケーション内で見つかったすべてのリモートサービス参照を表示します。
EJB	アプリケーション内の EJB のリストが含まれます。
JBPM	分析中に見つかった JBPM 関連のリソースすべてが含まれます。

レポート	説明
JPA	アプリケーションで見つかったすべての JPA 関連リソースの詳細が含まれます。
Hibernate	アプリケーションで検出されたすべての Hibernate 関連リソースの詳細が含まれています。
Server Resources	入力アプリケーションですべてのサーバーリソース (JNDI リソースなど) を表示します。
Spring Beans	分析中に見つかった Spring Bean の一覧が含まれます。
Hard-Coded IP Addresses	アプリケーションで見つかったすべてのハードコーディングされた IP アドレスの一覧を提供します。
Ignored Files	特定のルールおよび MTA 設定に基づいてアプリケーションに含まれるファイルを一覧表示し、処理されませんでした。詳細は、 --userIgnorePath オプションを参照してください。
About	現在のバージョンの MTA を説明し、詳細なヘルプリンクを提供します。

3.1.2. Issues レポート

Issues リンクをクリックして、Dashboard からこのレポートにアクセスします。

このレポートには、選択した移行パスによって発生したすべての問題に関する詳細情報が含まれます。発生した問題ごとに以下の情報が提供されます。

- 問題を要約するタイトル。
- インシデントの合計数、または問題の発生回数。
- 問題の1つのインスタンスを解決するルールのストーリーポイント。
- この問題を解決するための推定作業量レベル。
- 発生したすべてのインスタンスを解決するための全ストーリーポイント。これは、インシデントごとのストーリーポイントで検出されたインシデントの数を掛けて計算されます。

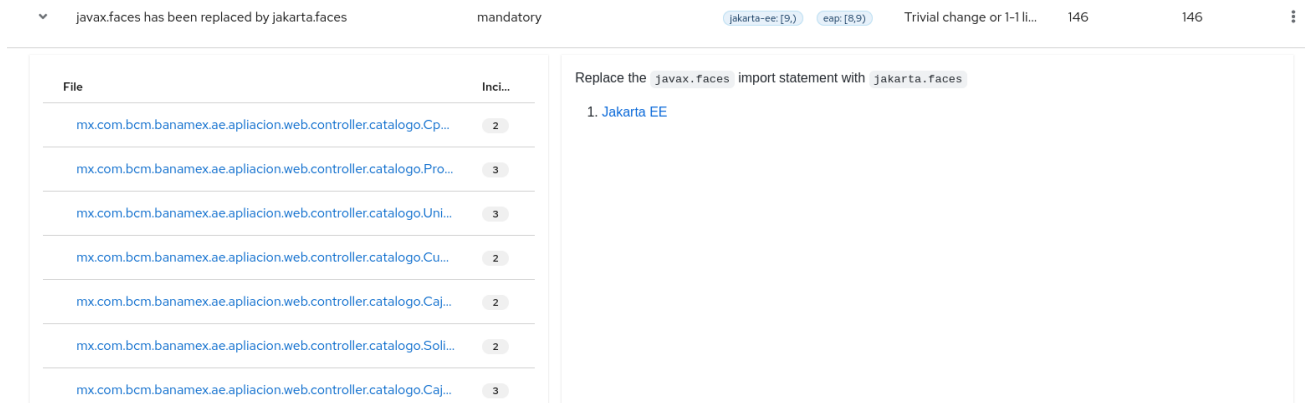
図3.4 Issues レポート

Issue	Category	Source technol...	Target technol...	Level of effort	Total incidents	Total storypoi...
> Common Annotations	information			Info	1	0
> Embedded framework - AOP Alliance	information			Info	1	0
> Embedded framework - Apache Aries	information			Info	4	0

報告された各問題は、タイトルをクリックして追加情報を取得することで拡張できます。以下の情報が含まれています。

- インシデントが発生したファイルの一覧と、各ファイル内のインシデントの数。ファイルが Java ソースファイルの場合は、ファイル名をクリックすると、対応するソースレポートが表示されます。
- 問題の詳細情報。この説明は問題の概要を示し、既知の解決策を提供し、問題または解決策に関するサポートドキュメントを参照します。
- 問題を生成したルールへの **Show Rule** というタイトルの直接リンク。

図3.5 問題の拡張



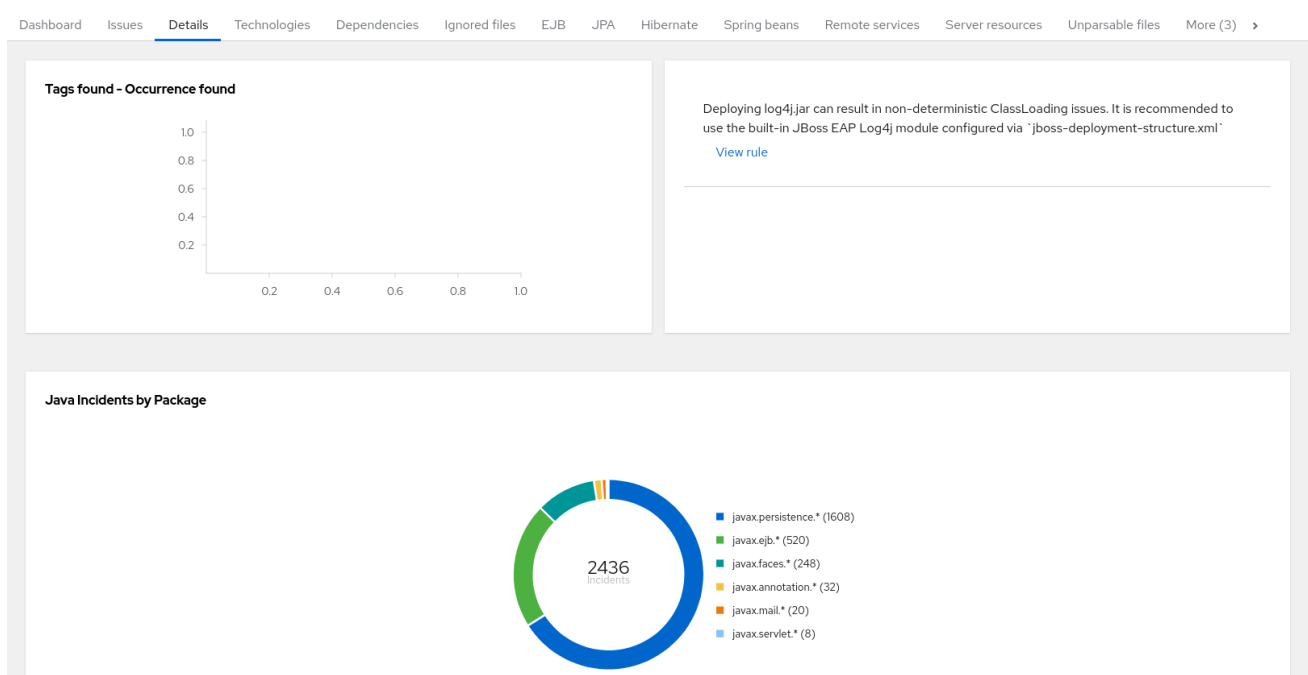
デフォルトでは、問題は4つのカテゴリーに分類されます。これらのカテゴリーに関する情報は、カテゴリーを参照してください。

3.1.3. Application Details レポート

Application Details リンクをクリックして、ダッシュボードからこのレポートにアクセスします。

レポートには、ストーリーポイント、パッケージごとの Java インシデント、およびアプリケーションで見つかったテクノロジーの発生回数が記載されます。以下は、移行プロセス中に生成されたアプリケーションメッセージの表示です。最後に、プロセス中に分析される各アーカイブにこの情報の内訳が表示されます。

図3.6 Application Details レポート

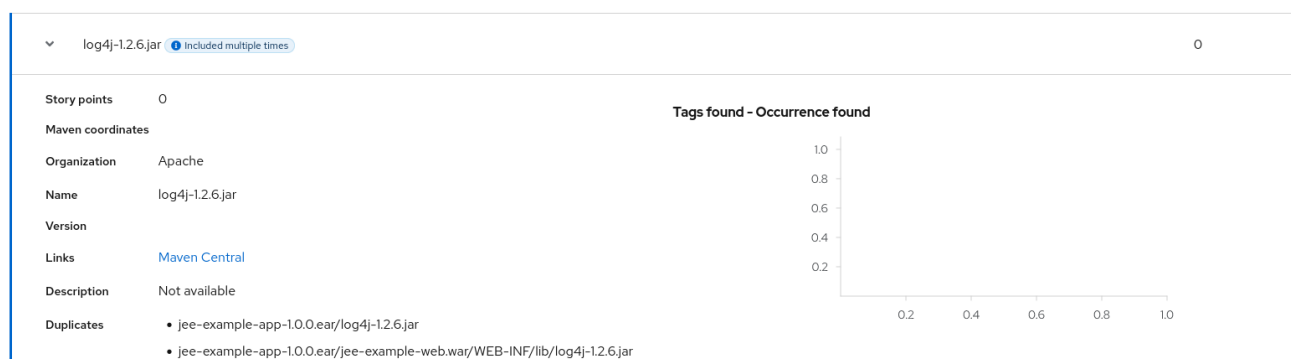


`jee-example-app-1.0.0.ear/jee-example-services.jar` をデプロイメントして、ストーリーポイント、パッケージごとの Java インシデント、およびこのアーカイブで見つかったテクノロジーの発生数を確認します。この概要は、移行に割り当てられたストーリーポイントの合計で始まり、アーカイブ内の各ファイルに必要な変更の詳細を示す表が続きます。レポートには以下のコラムが含まれます。

列名	説明
名前	分析されるファイルの名前。
Technology	分析するファイルのタイプ (例: Decompiled Java File または Properties)。
Issues	レビューまたは変更が必要なコードのエリアに関する警告。
ストーリーポイント	ファイルの移行に必要な作業レベル。

アーカイブがアプリケーションに複数回複製されると、そのアーカイブはレポートで1回だけリスト表示され、**[Included multiple times]** でタグ付けされます。

図3.7 アプリケーションでのアーカイブの重複



アプリケーション内で重複するアーカイブのストーリーポイントは、そのアプリケーションの合計ストーリーポイント数に1回だけカウントされます。

3.1.4. Technologies レポート

Technologies リンクをクリックして、ダッシュボードからこのレポートにアクセスします。

レポートには、解析されたアプリケーションで機能別にグループ化されたテクノロジーがリスト表示されます。これは、アプリケーションに含まれるテクノロジーの概要であり、各アプリケーションの目的を素早く理解できるように設計されています。

以下の図は、**jee-example-app** で使用されるテクノロジーを示しています。

図3.8 アプリケーションのテクノロジー

Category	Technology	Count
EJB Connect	Stateless (SLSB)	75
HTTP Connect	Servlet	2
Other Connect	Mail	26
	EAR Deployment	1
	Properties	5
	Common Annotations	1
Inversion of Control Execute	AOP Alliance	2
	Spring	11
	Spring DI	2
Processing Execute	Spring Scheduled	1
	Java EE XML	1
	Quartz	4
Database Store	JDBC	2
	JDBC XA datasources	1

3.1.5. トランザクションレポート

トランザクションレポートには、リレーショナルデータベーステーブルで操作を実行するコールスタックが表示されます。Enable Transaction Analysis 機能は、Spring Data JPA と、SQL ステートメント実行用の従来の **preparedStatement()** メソッドをサポートします。Hibernate などの ORM フレームワークはサポートしていません。

以下のイメージは、トランザクションレポートの例を示しています。

図3.9 トランザクションレポート

entry class	entry method
org.springframework.samples.petclinic.owner.PetController	populatePetTypes
SELECT ptype FROM PetType ptype ORDER BY ptype.name	
org.springframework.samples.petclinic.owner.PetController	findOwner
org.springframework.samples.petclinic.owner.VisitController	loadPetWithVisit
org.springframework.samples.petclinic.owner.PetController	initUpdateForm
org.springframework.samples.petclinic.owner.VisitController	processNewVisitForm
org.springframework.samples.petclinic.owner.PetController	processCreationForm
org.springframework.samples.petclinic.owner.PetController	processUpdateForm
org.springframework.samples.petclinic.owner.PetController	processUpdateOwnerForm
org.springframework.samples.petclinic.vet.VetController	showResourcesVetList
org.springframework.samples.petclinic.owner.OwnerController	processCreationForm
org.springframework.samples.petclinic.owner.OwnerController	initUpdateOwnerForm
org.springframework.samples.petclinic.owner.OwnerController	showOwner
org.springframework.samples.petclinic.owner.OwnerController	processFindForm
org.springframework.samples.petclinic.vet.VetController	showVetList

Page generated: 24 May 2022, 15:15:22

3.1.6. ソースレポート

ソースレポートには、移行の問題が検出されたソースファイルのコンテキストで表示されます。

図3.10 ソースレポート

File AdministracionEfectivo.ear/AdministracionEfectivo-web-0.0.1-SNAPSHOT.war/WEB-INF/classes/mx/com/bcm/banamex/ae/apliacion/web/controller/catalogo/...

```

5 import javax.faces.bean.ManagedBean;
6 import javax.faces.bean.RequestScoped;
7 import mx.com.bcm.banamex.ae.negocio.facade.CatalogoFacade;
8 import mx.com.bcm.banamex.ae.persistencia.exception.EfectivoAplicacionB0Exception;
9 import mx.com.bcm.banamex.ae.persistencia.vo.CajaVO;
10
11 @ManagedBean(
12     name = "cajaMB"
13 )
14 @RequestScoped
15 public class CajaMB implements Serializable {
16     private static final long serialVersionUID = 1L;
17     @EJB
18     private CatalogoFacade catalogoFacade;
19     private CajaVO cajaVO = new CajaVO();
20
21     public void consultCajas() throws EfectivoAplicacionB0Exception {
22     }
23
24     public void ConsultaEditCajas() throws EfectivoAplicacionB0Exception {
25     }
26
27     public void findByIpAddressCajaIdnTipoCaja(String cajaIpAddress, short cajaIdn, short cajaTipo
28     )
29
30     public void addCaja(CajaVO cajaVO) throws EfectivoAplicacionB0Exception {
31     }
32
33     public void formatoIp(CajaVO cajaVO) throws EfectivoAplicacionB0Exception {
34     }
35 }

```

Annotation `javax.faces.bean.RequestScoped` removed
Line:6

Annotation javax.faces.bean.RequestScoped removed. Use jakarta.enterprise.context.RequestScoped to replace it.

Close

3.2. TECHNOLOGIES レポート

Technologies リンクをクリックして、レポートランディングページからこのレポートにアクセスします。

このレポートは、解析されたアプリケーションに使用されたテクノロジーを機能別にまとめたリストを提供します。これは、テクノロジーがどのように分散されているかを示し、通常は、多数のアプリケーションを分析してアプリケーションをグループ化し、パターンを特定した後にレビューされます。また、各アプリケーションのサイズ、ライブラリー数、およびストーリーポイントの合計も表示されます。

Markup などの任意のヘッダーをクリックすると、結果が降順に並べ替えられます。同じヘッダーを再度選択すると、結果が昇順になります。現在選択されているヘッダーは、並べ替えの方向を示す方向矢印の横に太字で示されます。

図3.11 複数のアプリケーションで使用されるテクノロジー

Application:

Dependencies

This report is a statistic of technologies occurrences in the input applications. It shows how the technologies are distributed and is mostly useful when analysing many applications.

Application	Markup	MVC	Rich	Web
AdditionWithSecurity-EAR-0.01.ear	0	3	0	4
AdministracionEfectivo.ear	127	4	0	140
Archives shared by multiple applications	0	0	0	0

3.3. 複数のアプリケーションで共有されるアーカイブ

複数のアプリケーションによって共有されるアーカイブリンクをクリックして、レポートランディングページからこれらのレポートにアクセスします。このリンクは、適用可能な共有アーカイブがある場合にのみ利用できることに注意してください。

図3.12 複数のアプリケーションで共有されるアーカイブ

Applications > Archives shared by multiple applications

Archives shared by multiple applications

Dashboard Issues Details Technologies Dependencies Ignored files EJB JPA Hibernate Spring beans Remote services Server resources Unparsable files More (3)

Category	Incidents	Total Story Points
mandatory	0	0
optional	0	0
potential	0	0
cloud-mandatory	0	0
cloud-optional	0	0
information	2	0

Incidents and Story Points

Type	Incidents	Total Story Points
00020		

Mandatory Incidents and Story Points

これにより、複数のアプリケーション間で共有されるすべてのアーカイブの詳細なレポートを表示できます。

3.4. ルールプロバイダー実行の概要

Rule providers execution overview リンクをクリックして、レポートランディングページからこのレポートにアクセスします。

このレポートは、アプリケーションに対して MTA 移行コマンドを実行する際に実行するルールの一覧を提供します。

図3.13 ルールプロバイダー実行の概要

Rule providers execution overview

This report lists 'rule providers', or sets of Windup rules. They may originate from a '.windup.xml', a '.rhamt.xml', or a '.mta.xml' file or a Java class implementing 'RuleProvider'.

Phase	ID	Status	Result	
> PostReportGenerationPhase	CreateApplicationListReportRuleProvider_1	Condition met	Success	⋮
> PostReportGenerationPhase	CreateSourceReportRuleProvider_1	Condition met	Success	⋮
> DecompilationPhase	BeforeDecompileClassesRuleProvider_1	Condition met	Success	⋮
> DecompilationPhase	DecompileClassesRuleProvider_1	Condition met	Success	⋮
> DecompilationPhase	DecompileClassesRuleProvider_2	Condition met	Success	⋮
> ReportGenerationPhase	CreateUnparsableFilesReportRuleProvider_1	Condition met	Success	⋮
> ReportGenerationPhase	CreateAboutWindupReportRuleProvider_1	Condition met	Success	⋮

第4章 CSV 形式でのレポートのエクспорт

MTA には、分類やヒントなどのレポートデータをローカルファイルシステムのフラットファイルにエクспортする機能があります。export 関数は現在 CSV ファイル形式をサポートし、レポートデータはコンマ (,) で区切られたフィールドとして示されます。

CSV ファイルは、Microsoft Excel、OpenOffice Calc、LibreOffice Calc などのスプレッドシートソフトウェアでインポートおよび操作できます。スプレッドシートソフトウェアは、MTA レポートから結果データを並べ替え、分析、評価、および管理する機能を提供します。

4.1. レポートのエクспорт

レポートを CSV ファイルとしてエクспортするには、**--exportCSV** 引数を付けて MTA を実行します。CSV ファイルは、解析される各アプリケーションの、**--output** 引数で指定されたディレクトリーに作成されます。

解析されたすべてのアプリケーションにわたって発見されたすべての問題は、レポートのルートディレクトリーにエクспортされる **AllIssues.csv** ファイルに含まれます。

アプリケーションレポートからのレポートへのアクセス

CSV レポートをエクспортすると、問題レポートの CSV 問題をすべてダウンロードできます。これらの問題をダウンロードするには、Issues Report の **Download All Issues CSV** をクリックします。

図4.1 CSV ダウンロードに関する問題レポート

Issues
This report provides a concise summary of all issues identified.

Issue	Category	Source technol...	Target technolo...	Level of effort	Total incidents	Total storypoi...
> Caching - Ehcache embedded library	cloud-mandatory		cloud-readiness	Requires re-design or lib...	1	5
> Common Annotations	information			Info	1	0
> Embedded framework - AOP Alliance	information			Info	1	0
> Embedded framework - Apache Aries	information			Info	4	0

4.2. CSV ファイルのスプレッドシートプログラムへのインポート

1. スプレッドシートソフトウェア (例: Microsoft Excel) を起動します。
2. **File** → **Open** を選択します。
3. CSV でエクспортされるファイルを参照し、これを選択します。
4. これで、スプレッドシートソフトウェアでデータを分析できるようになりました。

4.3. CSV データ構造について

CSV 形式の出力ファイルには、以下のデータフィールドが含まれます。

ルール ID

指定の項目を生成したルールの ID。

問題のタイプ

ヒント または 分類。

件名

`classification` または `hint` の件名。このフィールドは、特定の項目の問題を要約します。

説明

指定項目の問題の詳細な説明。

リンク

問題に関する追加情報を提供する URL。リンクは、`リンク`、`およびリンクの説明`という 2 つの属性で設定されます。

アプリケーション

この項目が生成されたアプリケーションの名前。

ファイル名

指定項目のファイルの名前。

ファイルパス

指定項目のファイルパス。

行

指定項目のファイルの行番号。

ストーリーポイント

特定の項目に割り当てられた、努力のレベルを表すストーリーポイントの数。

第5章 アプリケーションの MAVEN 化

MTA は、提供されるアプリケーションに基づいて Apache Maven プロジェクト構造を生成する機能を提供します。これにより、適切な依存関係を指定する必要な Maven Project Object Model (POM) ファイルを使用してディレクトリ構造が作成されます。

この機能は、プロジェクトの最終ソリューションを作成する予定はありません。これは、開始点を示し、アプリケーションに必要な依存関係および API を特定することを目的としています。プロジェクトでは、さらにカスタマイズが必要になる場合があります。

5.1. MAVEN プロジェクト構造の生成

MTA の実行時に **--mavenize** フラグを渡すことにより、提供されたアプリケーションの Maven プロジェクト構造を生成できます。

次の例では、`jee-example-app-1.0.0.ear` テストアプリケーションを使用して MTA を実行します。

```
$ <MTA_HOME>/bin/windup-cli --input /path/to/jee-example-app-1.0.0.ear --output /path/to/output --target eap:6 --packages com.acme org.apache --mavenize
```

これにより、Maven プロジェクト構造が `/path/to/output/mavenized` ディレクトリに生成されます。



注記

--input 引数にコンパイルされたアプリケーションを指定する場合に限り、**--mavenize** オプションを使用できます。この機能は、ソースコードに対して MTA を実行する場合は利用できません。

--mavenizeGroupId オプションを使用して、POM ファイルに使用する `<groupId>` を指定することもできます。指定しないと、MTA はアプリケーションに適切な `<groupId>` の識別を試行するか、デフォルトで `com.mycompany.mavenized` になります。

5.2. MAVEN プロジェクト構造の確認

`/path/to/output/mavenized/<APPLICATION_NAME>/` ディレクトリには、以下の項目が含まれます。

- ルート **POM** ファイル。これは、最上位ディレクトリーの `pom.xml` ファイルです。
- BOM ファイル。これは、`-bom` で終わるディレクトリーの **POM** ファイルです。
- 1つ以上のアプリケーション **POM** ファイル。各モジュールには、アーカイブの名前が付けられたディレクトリーに **POM** ファイルがあります。

サンプルの `jee-example-app-1.0.0.ear` アプリケーションは、WAR と複数の JAR を含む EAR アーカイブです。これらのアーティファクトごとに個別のディレクトリーが作成されます。以下は、このアプリケーション用に作成された Maven プロジェクト構造です。

```
/path/to/output/mavenized/jee-example-app/
  jee-example-app-bom/pom.xml
  jee-example-app-ear/pom.xml
  jee-example-services2-jar/pom.xml
```

```
jee-example-services-jar/pom.xml
jee-example-web-war/pom.xml
pom.xml
```

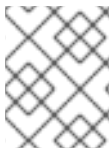
生成された各ファイルを確認し、プロジェクトに合わせてカスタマイズします。Maven POM ファイルの詳細は、Apache Maven ドキュメントの [Introduction to the POM](#) セクションを参照してください。

ルート POM ファイル

jee-example-app-1.0.0.ear アプリケーションのルート POM ファイルは `/path/to/output/mavenized/jee-example-app/pom.xml` にあります。このファイルは、すべてのプロジェクトモジュールのディレクトリーを特定します。

以下のモジュールは、サンプルの **jee-example-app-1.0.0.ear** アプリケーションのルート POM のリストに記載されています。

```
<modules>
  <module>jee-example-app-bom</module>
  <module>jee-example-services2-jar</module>
  <module>jee-example-services-jar</module>
  <module>jee-example-web-war</module>
  <module>jee-example-app-ear</module>
</modules>
```



注記

必要に応じて、モジュールのリストがプロジェクトの適切なビルド順序にリスト表示されるように、必ずモジュールのリストの順番を変更してください。

また、ルート POM は [Red Hat JBoss Enterprise Application Platform Maven リポジトリー](#) を使用してプロジェクトの依存関係をダウンロードするように設定されます。

BOM ファイル

BOM (Bill of Materials) ファイルは、**-bom** で終わるディレクトリーに生成されます。サンプルアプリケーション **jee-example-app-1.0.0.ear** の場合、BOM ファイルは `/path/to/output/mavenized/jee-example-app/jee-example-app-bom/pom.xml` にあります。この BOM の目的は、プロジェクトで使用されるサードパーティーの依存関係のバージョンを 1 か所で定義することです。BOM の使用に関する詳細は、Apache Maven ドキュメントの [Introduction to the dependencies mechanism](#) セクションを参照してください。

以下の依存関係は、**jee-example-app-1.0.0.ear** アプリケーションの BOM にリスト表示されています。

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>1.2.6</version>
    </dependency>
    <dependency>
      <groupId>commons-lang</groupId>
      <artifactId>commons-lang</artifactId>
      <version>2.5</version>
```

```
</dependency>  
</dependencies>  
</dependencyManagement>
```

アプリケーション POM ファイル

Maven が可能なアプリケーションモジュールには、POM ファイルが含まれる個別のディレクトリーがあります。ディレクトリー名には、アーカイブタイプに応じてアーカイブの名前と、接尾辞 **-jar**、**-war**、または **-ear** で終わるものが含まれます。

各アプリケーション POM ファイルには、以下を含むモジュールの依存関係がリスト表示されます。

- サードパーティーライブラリー
- Java EE API
- アプリケーションサブモジュール

たとえば、**jee-example-app-1.0.0.ear** EAR (/path/to/output/mavenized/jee-example-app/jee-example-app-ear/pom.xml) の POM ファイルには、以下の依存関係のリストが記載されます。

```
<dependencies>  
  <dependency>  
    <groupId>log4j</groupId>  
    <artifactId>log4j</artifactId>  
    <version>1.2.6</version>  
  </dependency>  
  <dependency>  
    <groupId>org.jboss.seam</groupId>  
    <artifactId>jee-example-web-war</artifactId>  
    <version>1.0</version>  
    <type>war</type>  
  </dependency>  
  <dependency>  
    <groupId>org.jboss.seam</groupId>  
    <artifactId>jee-example-services-jar</artifactId>  
    <version>1.0</version>  
  </dependency>  
  <dependency>  
    <groupId>org.jboss.seam</groupId>  
    <artifactId>jee-example-services2-jar</artifactId>  
    <version>1.0</version>  
  </dependency>  
</dependencies>
```

第6章 MTA パフォーマンスの最適化

MTA のパフォーマンスは、ハードウェア設定、アプリケーション内のファイルの数と種類、評価するアプリケーションのサイズと数、アプリケーションにソースコードとコンパイル済みコードのどちらが含まれているかなど、多くの要因に依存します。たとえば、10 MB を超えるファイルは、処理に時間がかかる場合があります。

一般に、MTA はクラスの逆コンパイルに約 40%、ルールの実行に 40% の時間を費やし、残りの時間は他のタスクの処理とレポートの生成に費やします。本セクションでは、MTA のパフォーマンスを改善するために実行できる内容を説明します。

6.1. アプリケーションのデプロイおよび実行

ハードウェアをアップグレードする前に、これらの提案を試してください。

- 可能であれば、アーカイブではなくソースコードに対して MTA を実行します。これにより、追加の JAR およびアーカイブをコンパイルする必要がなくなります。
- `<MTA_HOME>/bin/mta-cli` コマンドラインで `--packages` 引数を使用して、MTA によって評価されるパッケージのコンマ区切りリストを指定します。この引数を省略すると、MTA はすべてを逆コンパイルするため、パフォーマンスに大きな影響があります。
- `--excludeTags` 引数を指定すると、処理から除外できます。
- プロプライエタリーパッケージや含まれている依存関係などの不要なパッケージやファイルの逆コンパイルや解析は回避してください。
- 大規模なアプリケーションを分析する際に `ulimit` を増やします。Red Hat Enterprise Linux でこれを行う方法は、[この Red Hat ナレッジアーティクル](#) を参照してください。
- ラップトップまたはデスクトップマシンよりも優れたリソースを持つサーバーにアクセスできる場合は、そのサーバーで MTA を実行することを検討してください。

6.2. ハードウェアのアップグレード

上記のアプリケーションとコマンドラインの提案がパフォーマンスを改善しない場合は、ハードウェアをアップグレードが必要な場合があります。

- ラップトップまたはデスクトップよりも優れたリソースを持つサーバーにアクセスできる場合は、そのサーバーで MTA を実行することを検討してください。
- 逆コンパイルが必要な非常に大規模なアプリケーションには、大容量のメモリーが必要です。8 GB の RAM が推奨されます。これにより、3~4 GB の RAM が JVM で使用できるようになります。
- シングルコアまたはデュアルコアからクアッドコアの CPU プロセッサへのアップグレードにより、パフォーマンスが向上します。
- ディスク領域と断片化はパフォーマンスに影響を及ぼす可能性があります。高速ディスク、特にソリッドステートドライブ (SSD) で、4 GB を超える最適化されたディスク領域があると、パフォーマンスが向上します。

6.3. パッケージおよびファイルを除外する MTA の設定

6.3.1. パッケージの除外

逆コンパイル中および分析中にパッケージを除外して、パフォーマンスを向上させることができます。これらのパッケージへの参照はアプリケーションのソースコードに残りますが、その参照を除外すると、プロプライエタリークラスの逆コンパイルと解析が回避されます。

定義された値に一致するパッケージはすべて除外されます。たとえば、**com.acme** を使用して、**com.acme.example** および **com.acme.roadrunner** の両方を除外できます。

以下のいずれかの方法でパッケージを除外できます。

- **--excludePackages** 引数の使用
- 無視される場所のいずれかに含まれるファイルでパッケージを指定します。各パッケージは別々の行に含める必要があり、このファイルは **.package-ignore.txt** で終わる必要があります。たとえば、**<MTA_HOME>/ignore/proprietary.package-ignore.txt** を参照してください。

6.3.2. ファイルの除外

MTA は、スキャンやレポートの生成中に、含まれるライブラリーや依存関係などの特定のファイルを除外できます。除外されたファイルは、無視されたいずれかの場所で拡張子が **.mta-ignore.txt** または **.windup-ignore.txt** ファイルで定義されています。

これらのファイルには、除外する名前の詳細を示す正規表現文字列が含まれ、1行に1つのファイルがリストされます。たとえば、ライブラリー **ant.jar** と、次の内容を含むファイルを使用して、**Example** で始まる Java ソースファイルを除外できます。

```
.*ant.jar  
.*Example.*\.java
```

6.3.3. 除外の場所の検索

MTA は以下の場所を検索します。

- **~/.mta/ignore/**
- **~/.windup/ignore/**
- **<MTA_HOME>/ignore/**
- **--userIgnorePath** 引数で指定されたファイルおよびディレクトリー

この各ファイルは、除外するコンテンツのタイプに応じて、パッケージまたはファイルを除外するように指定されたルールに準拠する必要があります。

付録A 参考資料

A.1. MTA コマンドライン引数

以下は、利用可能な MTA コマンドライン引数の詳細な説明です。



注記

スクリプトから実行した場合など、プロンプトを表示せずに MTA コマンドを実行するには、以下の引数を使用する必要があります。

- **--batchMode**
- **--overwrite**
- **--input**
- **--target**

表A.1 MTA CLI 引数

引数	説明
--additionalClassPath	クラスパスに追加する追加の JAR ファイルまたはディレクトリーのスペース区切りリスト。逆コンパイルやその他の解析に使用できます。
--addonDir	指定したディレクトリーをカスタムアドオンリポジトリとして追加します。
--analyzeKnownLibraries	アプリケーション内に埋め込まれた既知のソフトウェアアーティファクトを分析するフラグ。デフォルトでは、MTA はアプリケーションコードのみを分析します。  注記 このオプションを使用すると実行時間が長くなり、多数の移行問題が報告される可能性があります。
--batchMode	MTA を非対話モードで実行すべきように指定するフラグ。確認をプロンプトなしで実行します。このモードは、コマンドラインに渡さないパラメーターのデフォルト値を取ります。
--debug	デバッグモードで MTA を実行するフラグ。
--disableTattletale	Tattletale レポートの生成を無効にするフラグ。 enableTattletale と disableTattletale の両方が true に設定されていると、 disableTattletale は無視されます。また、Tattletale レポートは引き続き生成されます。

引数	説明
--discoverPackages	入力バイナリアプリケーションで利用可能なパッケージをすべて表示するフラグ。
--enableClassNotFoundAnalysis	クラスパスで利用できない Java ファイルの分析を有効にするフラグ。分析時に一部のクラスが利用できない場合は、これを使用しないでください。
--enableCompatibleFilesReport	Compatible Files レポートの生成を有効にするフラグ。問題が検出されない状態ですべてのファイル进行处理するため、このレポートには大規模なアプリケーションの処理に時間がかかる場合があります。
--enableTattletale	各アプリケーションの Tattletale レポートの生成を有効にするフラグ。このオプションは、 eap が含まれるターゲットにある場合にデフォルトで有効になります。 enableTattletale と disableTattletale の両方が true に設定されていると、 disableTattletale は無視されます。また、Tattletale レポートは引き続き生成されます。
--enableTransactionAnalysis	<p>[テクノロジープレビュー] リレーショナルデータベーステーブルで操作を実行するコールスタックを表示するトランザクションレポートの生成を有効にするフラグ。Enable Transaction Analysis 機能は、Spring Data JPA と、SQL ステートメント実行用の従来の preparedStatement() メソッドをサポートします。Hibernate などの ORM フレームワークはサポートしていません。</p> <div data-bbox="687 1285 798 1664" style="float: left; width: 60px; height: 169px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); border: 1px solid #ccc; margin-bottom: 10px;"></div> <p>注記</p> <p>enableTransactionAnalysis はテクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。</p>
--excludePackages	評価から除外するパッケージのスペース区切りの一覧。たとえば、 com.mycompany.commonutilities と入力すると、パッケージ名が com.mycompany.commonutilities で始まるすべてのクラスが除外されます。
--excludeTags	除外するタグのスペースで区切られたリスト。指定されている場合は、これらのタグを持つルールは処理されません。タグの全リストを表示するには、 --listTags 引数を使用します。

引数	説明
--explodedApp	指定された入力ディクショナリーに1つのアプリケーションのソースファイルが含まれていることを示すフラグ。
--exportCSV	レポートデータをローカルファイルシステムの CSV ファイルにエクスポートするフラグ。MTA は、 --output 引数で指定されたディレクトリーにファイルを作成します。CSV ファイルは、データの操作および分析のためにスプレッドシートプログラムにインポートできます。
--exportSummary	MTA に、出力ディレクトリーに analysisSummary.json エクスポートファイルを生成するよう指示するフラグ。このファイルには、分析された各アプリケーションの分析概要情報 (カテゴリー別のインシデント数とストーリーポイント、分析されたアプリケーションに関連付けられたすべてのテクノロジータグなど) が含まれています。
--exportZipReport	この引数により、出力フォルダーに reports.zip ファイルが作成されます。ファイルには分析出力 (通常はレポート) が含まれます。要求に応じて、CSV エクスポートファイルを含めることもできます。
--help	MTA ヘルプメッセージを表示します。
--immutableAddonDir	指定したディレクトリーを、読み取り専用のアドオンディレクトリーとして追加します。
--includeTags	使用するタグのスペースで区切られたリスト。指定されると、これらのタグを持つルールのみが処理されます。タグの全リストを表示するには、 --listTags 引数を使用します。
--input	分析する1つ以上のアプリケーションを含むファイルまたはディレクトリーへのパスのスペースで区切られたリスト。この引数は必須です。
--install	インストールするアドオンを指定します。構文は <GROUP_ID>:<ARTIFACT_ID>[:<VERSION>] です。たとえば、 --install core-addon-x または --install org.example.addon:example:1.0.0 です。
--keepWorkDirs	グラフデータベースやデプロイメントされたアーカイブファイルなどの一時作業ファイルを削除しないように MTA に指示するフラグ。これはデバッグに役立ちます。
--legacyReports	新しい形式のレポートではなく古い形式のレポートを生成するように MTA に指示するフラグ。
--list	インストールされたアドオンをリスト表示するフラグ。

引数	説明
--listSourceTechnologies	利用可能なすべてのソーステクノロジーをリスト表示するフラグ。
--listTags	使用可能なタグをすべて表示するフラグ。
--listTargetTechnologies	利用可能なすべてのターゲットテクノロジーをリスト表示するフラグ。
--mavenize	アプリケーションの構造および内容に基づいて Maven プロジェクトディレクトリー構造を作成するフラグ。これにより、適切な Java EE API とプロジェクトモジュール間の正しい依存関係を使用して pom.xml ファイルが作成されます。-- mavenizeGroupId オプションも参照してください。
--mavenizeGroupId	-- mavenize オプションと使用すると、生成されるすべての pom.xml ファイルは <groupId> に提供された値を使用します。この引数を省略すると、MTA はアプリケーションに基づいて適切な <groupId> を判別しようとします。または、デフォルトでは com.mycompany.mavenized に設定されます。
--online	フラグは、それを必要とする機能のネットワークアクセスを許可します。現在、外部リソースに対する XML スキーマのみの検証は、インターネットアクセスに依存します。これには、パフォーマンスの低下があることに注意してください。
--output	MTA が生成したレポート情報を出力するディレクトリーへのパスを指定します。
--overwrite	<p>--output で指定された既存の出力ディレクトリーを強制的に削除するためのフラグ。この引数を指定せず、-output ディレクトリーが存在する場合は、内容を上書きするかどうかを選択するように求められます。</p> <div data-bbox="687 1563 794 1697" style="display: inline-block; vertical-align: middle;">  </div> <p style="margin-left: 20px;">重要</p> <p style="margin-left: 20px;">重要な情報を含むレポート出力ディレクトリーを上書きしないでください。</p>
--packages	MTA によって評価されるパッケージのスペース区切りのリスト。この引数を使用することは強く推奨されます。
--remove	指定したアドオンを削除します。構文は <GROUP_ID>:<ARTIFACT_ID>[:<VERSION>] です。たとえば、-- remove core-addon-x または -- remove org.example.addon:example:1.0.0 です。

引数	説明
--skipReports	HTML レポートが生成されないことを示すフラグ。この引数の一般的な用途は、 --exportCSV を使用してレポートデータを CSV ファイルにエクスポートすることです。
--source	移行元の1つ以上のソーステクノロジー、サーバー、プラットフォーム、またはフレームワークのスペース区切りのリスト。この引数は、 --target 引数とともに、使用されるルールセットを判断するのに役立ちます。 --listSourceTechnologies 引数を使用して、利用可能なソースをリスト表示します。
--sourceMode	評価するアプリケーションに、コンパイルされたバイナリーではなくソースファイルが含まれていることを示すフラグ。 sourceMode 引数は廃止されました。そのため、指定する必要がなくなりました。MTA は、提示された入力を直感的に処理できます。さらに、プロジェクトソースフォルダーは、同じ分析実行内でバイナリー入力を使用して分析できます。
--target	移行先の1つ以上のターゲットテクノロジー、サーバー、プラットフォーム、またはフレームワークのスペース区切りのリスト。この引数は、 --source 引数とともに、使用されるルールセットを判断するのに役立ちます。 --listTargetTechnologies 引数を使用して、利用可能なターゲットをリスト表示します。
--userIgnorePath	\${user.home}/.mta/ignore/ の他に、無視すべきファイルを MTA が特定できるように場所を指定します。
--userLabelsDirectory	MTA がカスタムターゲットランタイムラベルを探す場所を指定します。値には、複数または単数のラベルファイルを含むディレクトリーを使用できます。ターゲットランタイムラベルファイルでは、 .windup.label.xml の接尾辞を使用する必要があります。同梱されている Target Runtime Labels は \$(MTA_HOME)/rules/migration-core/core.windup.label.xml 内で定義されます。
--userRulesDirectory	<MTA_HOME>/rules/ および \$(user.home)/.mta/rules/ に加えて、MTA がカスタム MTA ルールを探す場所を指定します。値は、単数または複数のルールセットファイルを含むディレクトリーです。ルールセットファイルでは、 .windup.xml の接尾辞を使用する必要があります。
--version	MTA バージョンを表示します。
--skipSourceCodeReports	このオプションは、アプリケーション分析レポートの生成時に ソースコードレポート の生成をスキップします。アプリケーション分析に表示されるソースコード情報が心配な場合は、この詳細オプションを使用します。

A.1.1. 入力の指定

分析する1つ以上のアプリケーションを含むファイルまたはディレクトリーへのパスのスペースで区切られたリスト。この引数は必須です。

使用方法

```
--input <INPUT_ARCHIVE_OR_DIRECTORY> [...]
```

--input 引数に指定された入力ファイルタイプがファイルであるかディクショナリーであるかに応じて、指定された追加の引数に応じて次のように評価されます。

ディレクトリー

--explodedApp	--sourceMode	引数なし
ディレクトリーは1つのアプリケーションとして評価されます。	ディレクトリーは1つのアプリケーションとして評価されます。	各サブディレクトリーはアプリケーションとして評価されます。

ファイル

--explodedApp	--sourceMode	引数なし
引数は無視されます。ファイルは1つのアプリケーションとして評価されます。	ファイルは圧縮プロジェクトとして評価されます。	ファイルは1つアプリケーションとして評価されます。

A.1.2. 出力ディレクトリーの指定

MTA が生成したレポート情報を出力するディレクトリーへのパスを指定します。

使用方法

```
--output <OUTPUT_REPORT_DIRECTORY>
```

- 省略すると、レポートは <INPUT_ARCHIVE_OR_DIRECTORY>.report ディレクトリーに生成されます。
- 出力ディレクトリーが存在する場合は、次のメッセージが表示されます (デフォルトは N)。

```
Overwrite all contents of "/home/username/<OUTPUT_REPORT_DIRECTORY>" (anything already in the directory will be deleted)? [y,N]
```

ただし、--overwrite 引数を指定すると、MTA はディレクトリーの削除と再作成を続行します。詳細は、この引数の説明を参照してください。

A.1.3. ソーステクノロジーの設定

移行元の1つ以上のソーステクノロジー、サーバー、プラットフォーム、またはフレームワークのスペース区切りのリスト。この引数は、--target 引数とともに、使用されるルールセットを判断するのに役立ちます。--listSourceTechnologies 引数を使用して、利用可能なソースをリスト表示します。

使用方法

```
--source <SOURCE_1> <SOURCE_2>
```

--source 引数は、[Maven バージョン範囲の構文](#) に続くバージョンサポートを提供するようになりました。これにより、指定されたバージョンに一致するルールセットのみを実行するように MTA が指示されます。例: **--source eap:5**



警告

JBoss EAP に移行する場合は、バージョン (例: **eap:6**) を指定してください。**eap** のみを指定すると、移行パスに関連しないものを含め、すべてのバージョンの JBoss EAP にルールセットが実行されます。

JBoss EAP バージョンに適した [Migration Toolkit for Applications の概要のサポートされる移行パス](#) を参照してください。

A.1.4. ターゲットテクノロジーの設定

移行先の1つ以上のターゲットテクノロジー、サーバー、プラットフォーム、またはフレームワークのスペース区切りのリスト。この引数は、**--source** 引数とともに、使用されるルールセットを判断するのに役立ちます。このオプションを指定しないと、ターゲットを選択するように求められます。**--listTargetTechnologies** 引数を使用して、利用可能なターゲットをリスト表示します。

使用方法

```
--target <TARGET_1> <TARGET_2>
```

--target 引数は、[Maven バージョン範囲の構文](#) に続くバージョンサポートを提供するようになりました。これにより、指定されたバージョンに一致するルールセットのみを実行するように MTA が指示されます。例: **--target eap:7**



警告

JBoss EAP に移行する場合は、必ずバージョンをターゲットに指定してください (例: **eap:6**)。 **eap** のみを指定すると、移行パスに関連しないものを含め、すべてのバージョンの JBoss EAP にルールセットが実行されます。

JBoss EAP バージョンに適した [Migration Toolkit for Applications の概要のサポートされる移行パス](#) を参照してください。

A.1.5. パッケージの選択

MTA によって評価されるパッケージのスペース区切りのリスト。この引数を使用することは強く推奨されます。

使用方法

```
--packages <PACKAGE_1> <PACKAGE_2> <PACKAGE_N>
```

- 多くの場合、関心があるのは、カスタムアプリケーションクラスパッケージの評価で、標準の Java EE パッケージやサードパーティーのパッケージではありません。**<PACKAGE_N>** 引数はパッケージ接頭辞で、すべてのサブパッケージがスキャンされます。たとえば、パッケージ **com.mycustomapp** および **com.myotherapp** をスキャンするには、コマンドラインで **--packages com.mycustomapp com.myotherapp** 引数を使用します。
- **org.apache** などの標準の Java EE サードパーティーソフトウェアにパッケージ名を指定することはできますが、通常は移行作業に影響しないため、追加しないことが推奨されます。



警告

--packages 引数を省略すると、アプリケーションのすべてのパッケージがスキャンされ、パフォーマンスに影響を及ぼす可能性があります。

A.2. サポート対象のテクノロジータグ

以下のテクノロジープレビュータグは MTA 6.1.3 でサポートされています。

- OMQ Client
- 3scale
- Acegi Security
- AcrIS Security
- ActiveMQ library
- Airframe
- Airlift Log Manager
- AKKA JTA
- Akka Testkit
- Amazon SQS Client
- AMQP Client
- Anakia
- AngularFaces
- ANTLR StringTemplate
- AOP Alliance

- Apache Accumulo Client
- Apache Aries
- Apache Commons JCS
- Apache Commons Validator
- Apache Flume
- Apache Geronimo
- Apache Hadoop
- Apache HBase Client
- Apache Ignite
- Apache Karaf
- Apache Mahout
- Apache Meecrowave JTA
- Apache Sirona JTA
- Apache Synapse
- Apache Tapestry
- Apiman
- Applet
- Arquillian
- AspectJ
- Atomikos JTA
- Avalon Logkit
- Axion Driver
- Axis
- Axis2
- BabbageFaces
- Bean Validation
- BeanInject
- Blaze
- Blitz4j

-
- BootsFaces
 - Bouncy Castle
 - ButterFaces
 - Cache API
 - Cactus
 - Camel
 - Camel Messaging Client
 - Camunda
 - Cassandra Client
 - CDI
 - Cfg Engine
 - Chunk Templates
 - Cloudera
 - Coherence
 - Common Annotations
 - Composite Logging
 - Composite Logging JCL
 - Concordion
 - CSS
 - Cucumber
 - Dagger
 - DbUnit
 - Demoiselle JTA
 - Derby Driver
 - Drools
 - DVSL
 - Dynacache
 - EAR Deployment
 - Easy Rules

- EasyMock
- Eclipse RCP
- EclipseLink
- Ehcache
- EJB
- EJB XML
- Elasticsearch
- Entity Bean
- EtlUnit
- Eureka
- Everit JTA
- Evo JTA
- Feign
- File system Logging
- FormLayoutMaker
- FreeMarker
- Geronimo JTA
- GFC Logging
- GIN
- GlassFish JTA
- Google Guice
- Grails
- Grapht DI
- Guava Testing
- GWT
- H2 Driver
- Hamcrest
- Handlebars
- HavaRunner

-
- Hazelcast
 - Hdiv
 - Hibernate
 - Hibernate Cfg
 - Hibernate Mapping
 - Hibernate OGM
 - HighFaces
 - HornetQ Client
 - HSQLDB Driver
 - HTTP Client
 - HttpUnit
 - ICEfaces
 - Ickenham
 - Ignite JTA
 - Iksan
 - iLog
 - Infinispan
 - Injekt for Kotlin
 - Iroh
 - Istio
 - Jamon
 - Jasypt
 - Java EE Batch
 - Java EE Batch API
 - Java EE JACC
 - Java EE JAXB
 - Java EE JAXR
 - Java EE JSON-P
 - Java Transaction API

- JavaFX
- JavaScript
- Javax Inject
- JAX-RS
- JAX-WS
- JayWire
- JBehave
- JBoss Cache
- JBoss EJB XML
- JBoss logging
- JBoss Transactions
- JBoss Web XML
- JBossMQ Client
- JBPM
- JCA
- Jcabi Log
- JCache
- JUnit
- JDBC
- JDBC datasources
- JDBC XA datasources
- Jersey
- Jetbrick Template
- Jetty
- JFreeChart
- JFunk
- JGoodies
- JMock
- JMockit

-
- JMS Connection Factory
 - JMS Queue
 - JMS Topic
 - JMustache
 - JNA
 - JNI
 - JNLP
 - JPA entities
 - JPA Matchers
 - JPA named queries
 - JPA XML
 - JSecurity
 - JSF
 - JSF Page
 - JSilver
 - JSON-B
 - JSP Page
 - JSTL
 - JTA
 - Jukito
 - JUnit
 - Ka DI
 - Keyczar
 - Kibana
 - KLogger
 - Kodein
 - Kotlin Logging
 - Koulject
 - KumuluzEE JTA

- LevelDB Client
- Liferay
- LiferayFaces
- Lift JTA
- Log.io
- Log4J
- Log4s
- Logback
- Logging Utils
- Logstash
- Lumberjack
- Macros
- Magicgroupayout
- Mail
- Management EJB
- MapR
- MckoiSQLDB Driver
- Memcached
- Message (MDB)
- Micro DI
- Micrometer
- Microsoft SQL Driver
- MiGLayout
- MinLog
- Mixer
- Mockito
- MongoDB Client
- Monolog
- Morphia

-
- MRules
 - Mule
 - Mule Functional Test Framework
 - MultithreadedTC
 - Mycontainer JTA
 - MyFaces
 - MySQL Driver
 - Narayana Arjuna
 - Needle
 - Neo4j
 - NLOG4J
 - Nuxeo JTA/JCA
 - OACC
 - OAUTH
 - OCPsoft Logging Utils
 - OmniFaces
 - OpenFaces
 - OpenPojo
 - OpenSAML
 - OpenWS
 - OPS4J Pax Logging Service
 - Oracle ADF
 - Oracle DB Driver
 - Oracle Forms
 - Orion EJB XML
 - Orion Web XML
 - Oscache
 - OTR4J
 - OW2 JTA

- OW2 Log Util
- OWASP CSRF Guard
- OWASP ESAPI
- Peaberry
- Pega
- Persistence units
- Petals EIP
- PicketBox
- PicketLink
- PicoContainer
- Play
- Play Test
- Plexus Container
- Polyforms DI
- Portlet
- PostgreSQL Driver
- PowerMock
- PrimeFaces
- Properties
- Qpid Client
- RabbitMQ Client
- RandomizedTesting Runner
- Resource Adapter
- REST Assured
- Restito
- RichFaces
- RMI
- RocketMQ Client
- Rythm Template Engine

- SAML
- Santuario
- Scalate
- Scaldi
- Scribe
- Seam
- Security Realm
- ServiceMix
- Servlet
- ShiftOne
- Shiro
- Silk DI
- SLF4J
- Snippetory Template Engine
- SNMP4J
- Socket handler logging
- Spark
- Specs4j
- Spock
- Spring
- Spring Batch
- Spring Boot
- Spring Boot Actuator
- Spring Boot Cache
- Spring Boot Flo
- Spring Cloud Config
- Spring Cloud Function
- Spring Data
- Spring Data JPA

- spring DI
- Spring Integration
- Spring JMX
- Spring Messaging Client
- Spring MVC
- Spring Properties
- Spring Scheduled
- Spring Security
- Spring Shell
- Spring Test
- Spring Transactions
- Spring Web
- SQLite Driver
- SSL
- Standard Widget Toolkit (SWT)
- Stateful (SFSB)
- Stateless (SLSB)
- Sticky Configured
- Stripes
- Struts
- SubCut
- Swagger
- SwarmCache
- Swing
- SwitchYard
- Syringe
- Talend ESB
- Teiid
- TensorFlow

-
- Test Interface
 - TestNG
 - Thymeleaf
 - TieFaces
 - tinylog
 - Tomcat
 - Tornado Inject
 - Trimou
 - Trunk JGuard
 - Twirl
 - Twitter Util Logging
 - UberFire
 - Unirest
 - Unitils
 - Vaadin
 - Velocity
 - Vlad
 - Water Template Engine
 - Web Services Metadata
 - Web Session
 - Web XML File
 - WebLogic Web XML
 - Webmacro
 - WebSocket
 - WebSphere EJB
 - WebSphere EJB Ext
 - WebSphere Web XML
 - WebSphere WS Binding
 - WebSphere WS Extension

- Weka
- Weld
- WF Core JTA
- Wicket
- Winter
- WSDL
- WSO2
- WSS4J
- XACML
- XFire
- XMLUnit
- Zbus Client
- Zipkin

A.3. ルールのストーリーポイントについて

A.3.1. ストーリーポイントとは

ストーリーポイントは、アジャイルソフトウェア開発で一般的に使用される抽象メトリックで、機能や変更を実装するのに必要な作業量を予測します。

Migration Toolkit for Applications はストーリーポイントを使用して、特定のアプリケーションコンストラクトとアプリケーション全体を移行するために必要な作業のレベルを表現します。必ずしも工数に変換される訳ではありませんが、この値はタスク全体で一貫性を持たせる必要があります。

A.3.2. ルールにおけるストーリーポイントの見積方法

ルールのストーリーポイントの作業レベルを見積もることは複雑です。以下は、ルールに必要な作業レベルを見積もる際に MTA が使用する一般的なガイドラインです。

作業レベル	ストーリーポイント	説明
Information	0	移行の優先度が非常に低いか、優先度のない情報警告。
Trivial	1	移行は簡単な変更、または API の変更がないか最小限の変更を伴う単純なライブラリーの交換です。
Complex	3	移行タスクに必要な変更は複雑ですが、解決策が文書化されています。

作業レベル	ストーリーポイント	説明
Redesign	5	移行タスクでは、API が大幅に変更され、再設計または完全なライブラリーの変更が必要になります。
Rearchitecture	7	移行には、コンポーネントまたはサブシステムの完全な再アーカイブが必要です。
Unknown	13	移行ソリューションは不明なため、完全な再書き込みが必要になる場合があります。

A.3.3. タスクカテゴリー

作業量レベルに加えて、移行タスクを分類してタスクの重大度を示すことができます。次のカテゴリーは、移行作業の優先順位を行えるように、問題をグループ化するために使用されます。

Mandatory

移行を成功させるには、タスクを完了する必要があります。変更が行われないと、生成されるアプリケーションはビルドまたは実行に成功しません。たとえば、ターゲットプラットフォームでサポートされないプロプライエタリー API の置き換え例が含まれます。

Optional

移行タスクが完了しない場合、アプリケーションは動作しますが、結果が最適になるとは限りません。移行時に変更が行われなかった場合は、移行の完了後すぐにスケジュールに配置することが推奨されます。これには、EJB 2.x コードの EJB 3 へのアップグレードが挙げられます。

Potential

移行プロセス中にタスクを調べる必要があります。しかし、移行を成功させるためにタスクが必須かどうかを判断するのに十分な詳細情報がありません。この例は、直接互換性のあるタイプがないサードパーティーのプロプライエタリータイプの移行です。

Information

タスクは、特定のファイルの存在を通知するために含まれています。これらは、モダナイゼーション作業の一部として検証または変更する必要がある場合がありますが、通常は変更は必要ありません。これには、ロギング依存関係または Maven `pom.xml` があります。

タスクの分類に関する詳細は、[カスタムルールカテゴリーの使用](#) を参照してください。

A.4. 関連情報

A.4.1. ご協力をお願い

Migration Toolkit for Applications がほとんどのアプリケーション設定とサーバー設定 (お客様のものを含む) をカバーできるようにするために、次の項目のいずれかを支援してください。

- jboss-migration-feedback@redhat.com にメールを送信し、MTA 移行ルールが対象とすべき内容を知らせる。
- 移行ルールをテストするためのアプリケーションの例を指定する。
- 移行が困難なアプリケーションコンポーネントおよび問題の領域を特定する。

- これらの問題がある移行領域について簡単な説明を記入します。
- 問題の移行領域を解決する方法を説明する簡単な概要を記述する。
- アプリケーションで Migration Toolkit for Applications を試行する。発生している問題は必ず報告してください。
- Migration Toolkit for Applications ルールリポジトリに貢献する。
 - Migration Toolkit for Applications ルールを記述して、移行プロセスを識別または自動化する。
 - 新規ルールのテストを作成する。
 - 詳細は、[ルール開発ガイド](#) を参照してください。
- プロジェクトのソースコードに貢献する。
 - コアルールを作成する。
 - MTA のパフォーマンスまたは効率が向上する。

どのレベルの参加でも大歓迎です。

A.4.2. リソース

- MTA フォーク: <https://developer.jboss.org/en/windup>
- Jira 課題トラッカー: <https://issues.redhat.com/projects/TACKLE>
- MTA メーリングリスト: jboss-migration-feedback@redhat.com

A.4.3. 問題の報告

MTA は Jira を問題追跡システムとして使用します。MTA の実行で問題が発生した場合は、[Jira issue](#) を作成してください。

改訂日時: 2023-10-26