Red Hat Decision Manager 7.1

Migrating from Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1

Red Hat Decision Manager 7.1 Migrating from Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Abstract

This document describes how to migrate project data from Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1.

Table of Contents

PREFACE

As a system administrator or business rules developer, you can migrate your existing project data in Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1. Red Hat provides a Decision Central migration tool to facilitate migration, but in some cases, manual migration or asset re-creation may be required.

Prerequisites

- Red Hat Decision Manager 7.0 is installed and contains artifacts that you want to migrate to Red Hat Decision Manager 7.1.

- A clean Red Hat Decision Manager 7.1 installation exists that does not contain a **.niogit** folder. If the Red Hat Decision Manager 7.1 installation contains a **.niogit** folder, the migration will fail. For installation options, see *Planning a Red Hat Decision Manager installation* .

> **IMPORTANT**
>
> If you use Red Hat JBoss BRMS 6.4, you can migrate directly to Red Hat Decision Manager 7.1 without migrating to Red Hat Decision Manager 7.0. For more information, see *Migrating from Red Hat JBoss BRMS 6.4 to Red Hat Decision Manager 7.1* .

CHAPTER 1. MIGRATION OVERVIEW

If you use Red Hat Decision Manager 7.0 and install Red Hat Decision Manager 7.1, you must use the Decision Central migration tool provided with this release to migrate the applications that you created in Red Hat Decision Manager 7.0 to the new product. The reason for this migration requirement is that project data is restructured in Red Hat Decision Manager 7.1 so that each space (previously known as organizational unit) contains repositories that correspond to individual projects, instead of multiple projects in a single repository as in Red Hat Decision Manager 7.0. This improved structure means that you do not need to create and manage repositories in the new Decision Central and can focus on developing your business assets.

Red Hat Decision Manager 7.1 also contains Red Hat Business Optimizer updates that you may need to apply to the relevant Red Hat Business Optimizer assets in your project, if applicable.

For more information about updates and new features in this release, see the *Release notes for Red Hat Decision Manager 7.1*.

CHAPTER 2. DECISION CENTRAL DATA MIGRATION WITH THE MIGRATION TOOL

Red Hat Decision Manager 7.1 contains a modified data structure and other feature changes that do not support a direct data migration from Red Hat Decision Manager 7.0. Therefore, Red Hat provides a Decision Central migration tool that enables you to migrate project data and configurations from Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1. You can run the migration tool on the **.niogit** directory of your Decision Central distribution or on an external Git repository. The migration tool prepares your Decision Central data for the new data structure and features supported in Red Hat Decision Manager 7.1.

2.1. USING THE MIGRATION TOOL ON A **.NIOGIT** DIRECTORY

The **.niogit** directory of your Decision Central distribution contains all Decision Central data. You can use the Decision Central migration tool on your **.niogit** directory to prepare project data and system configurations for migration to Red Hat Decision Manager 7.1.

Prerequisites

- Decision Central is not running for either Red Hat Decision Manager 7.0 or Red Hat Decision Manager 7.1.

- Java 8 is installed and is available on the path where the project migration tool will be installed.

Procedure

1. Locate the **bin/.niogit** directory in your current Decision Central distribution for Red Hat Decision Manager 7.0 and locate the same directory in your new Red Hat Decision Manager 7.1 installation.
   Example:

   ```
   $ ~/$RHDM_7.0_HOME/bin/.niogit
   ```

   ```
   $ ~/$RHDM_7.1_HOME/bin/.niogit
   ```

2. In your command terminal, copy the **.niogit** directory from your current Decision Central distribution for Red Hat Decision Manager 7.0 to the **bin** folder of the new Red Hat Decision Manager 7.1 installation directory:

   ```
   cp -r /$RHDM_7.0_HOME/bin/.niogit  /$RHDM_7.1_HOME/bin/
   ```

   The copied **.niogit** folder is the directory that the migration tool will prepare for Red Hat Decision Manager 7.1. The original **.niogit** folder remains in your Red Hat Decision Manager 7.0 installation and will not be modified by the migration tool in this procedure.

3. In the Red Hat Decision Manager 7.1 **bin** directory, delete the **.index** directory. You must delete this folder because Red Hat Decision Manager 7.1 comes with certain updates that are not compatible with Red Hat Decision Manager 7.0. The **bin** directory will be re-indexed after the migration when you start Red Hat Decision Manager 7.1.

4. Note the path to the copied **.niogit** directory for Red Hat Decision Manager 7.1. The path will be required when you run the migration tool.

5. If you have specific **dependencies**, **repositories**, or **pluginRepositories** that you want to add, remove, or update as part of the project migration, create a **pom-migration.json** file containing these elements. When you run the migration tool, the tool will prompt you if you want to include this file in the migration.
   The **pom-migration.json** file requires the following sections:

   - **"dependencies":[]**

   - **"repositories-add":[]**

   - **"repositories-remove":[]**

   - **"repositories-update-urls":[]**

   - **"pluginRepositories-add":[]**

   - **"pluginRepositories-remove":[]**

   - **"pluginRepositories-update-urls":[]**

   If any of these elements do not apply to your project, include them as empty sections to prevent parsing errors.

   Example **pom-migration.json** file:

   ```
   {
     "dependencies":[

       {"groupId":"junit", "artifactId":"junit", "version":"4.12", "scope":"test"},
       {"groupId":"com.thoughtworks.xstream", "artifactId":"xstream", "version":"4.12",
   "scope":"test"}

     ],

     "repositories-add":[
       {
         "id":"jboss-public-repository-group",
         "name":"JBoss Public Repository Group",
         "url":"http://repository.jboss.org/nexus/content/groups/public/",
         "releasesEnabled":true,
         "releasesUpdatePolicy":"never",
         "snapshotEnabled":true,
         "snapshotUpdatePolicy":"never"
       },
       {
         "id":"kie-internal-staging-repository-group",
         "name":"KIE Internal Staging Repositories",
         "url":"https://origin-repository.jboss.org/nexus/content/groups/kie-internal-group",
         "releasesEnabled":true,
         "releasesUpdatePolicy":"never",
         "snapshotEnabled":false,
         "snapshotUpdatePolicy":"never"
       }
     ],

     "repositories-remove":[
   ```

```
      {
        "id":"productization-repository",
  "url":"http://download.lab.bos.redhat.com/brewroot/repos/jb-ip-6.1-build/latest/maven/"
      }
    ],

    "repositories-update-urls":[
      {
        "id":"guvnor-m2-repo", "url":"http://localhost:8080/decision-central/maven3/"
      }
    ],

    "pluginRepositories-add":[

      {
        "id":"jboss-public-repository-group",
        "name":"JBoss Public Repository Group",
        "url":"http://repository.jboss.org/nexus/content/groups/public/",
        "releasesEnabled":true,
        "releasesUpdatePolicy":"never",
        "snapshotEnabled":true,
        "snapshotUpdatePolicy":"never"
      },
      {
        "id":"kie-internal-staging-repository-group",
        "name":"KIE Internal Staging Repositories",
        "url":"https://origin-repository.jboss.org/nexus/content/groups/kie-internal-group",
        "releasesEnabled":true,
        "releasesUpdatePolicy":"never",
        "snapshotEnabled":false,
        "snapshotUpdatePolicy":"never"
      }

    ],
    "pluginRepositories-remove":[],
    "pluginRepositories-update-urls":[]
  }
```

Example **pom-migration.json** file with **dependencies** only:

```
  {
    "dependencies":[

      {"groupId":"junit", "artifactId":"junit", "version":"4.12", "scope":"test"},
      {"groupId":"com.thoughtworks.xstream", "artifactId":"xstream", "version":"4.12",
  "scope":"test"}

    ],
    "repositories-add":[],
    "repositories-remove":[],
    "repositories-update-urls":[],
    "pluginRepositories-add":[],
    "pluginRepositories-remove":[],
    "pluginRepositories-update-urls":[]
  }
```

6. Navigate to the Software Downloads page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:

   - **Product:** Decision Manager

   - **Version:** 7.1

7. Download **Red Hat Decision Manager 7.1.0 Add-Ons** and extract the downloaded **rhdm-7.1.0-add-ons.zip** file to a temporary directory.

8. In the extracted **rhdm-7.1.0-add-ons** folder, extract the **rhdm-7.1-migration-tool.zip** sub-folder. The migration tool is in the **bin** directory.

9. In your command terminal, navigate to the temporary directory where you extracted the **rhdm-7.1-migration-tool** folder and run the migration tool. The **$RHDM_NIOGIT_DIR** portion is the path to the **.niogit** directory that you previously copied to the Red Hat Decision Manager 7.1 installation.
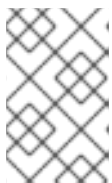   On Linux or UNIX-based systems:

   ```
   $ cd $INSTALL_DIR/rhdm-7.1-migration-tool/bin
   $ ./migration-tool.sh -t $RHDM_NIOGIT_DIR
   ```

   On Windows:

   ```
   $ cd $INSTALL_DIR\rhdm-7.1-migration-tool\bin
   $ migration-tool.bat -t $RHDM_NIOGIT_DIR
   ```

   In the command prompt that appears, the following options are displayed:

   - **Project structure migration**: Migrates the Red Hat Decision Manager 7.0 project repository structure to the new project-oriented structure used in Red Hat Decision Manager 7.1.

   - **System configuration directory structure migration**: Migrates the **system.git** repository structure used in Red Hat Decision Manager 7.0 to the new structure used in Red Hat Decision Manager 7.1. This migration option requires the project structure migration to be executed first.

   - **POMs migration:** Updates **pom.xml** files with dependencies required for Red Hat Decision Manager 7.1. This migration option requires the project structure migration and system configuration directory structure migration to be executed first.

   - **All:** Runs all migration options in sequence.

   - **Exit:** Exits the migration tool.

10. Select the option to run **All** migrations in sequence.

    > **NOTE**
    >
    > If you prefer to run one migration option at a time, select and run the first individual migration option. After the tool runs, re-run the Decision Central migration tool and select the next individual migration option in the sequence.

11. Enter **yes** each time you are prompted to run a specific migration option.

For the POMs migration option, if you want to include a path to an external **pom-migration.json** file that you created previously, enter **yes** when prompted and enter the path.

12. After the tool finishes running, enter the option to **Exit** the migration tool.
The **.niogit** directory structure is now compatible with Decision Central in Red Hat Decision Manager 7.1. Project directories are in separate repositories and all other related configurations have been migrated. You can navigate to the new **.niogit** directory to inspect the restructured contents.

2.2. USING THE MIGRATION TOOL ON AN EXTERNAL GIT REPOSITORY

If you store Decision Central project data in a Git repository outside of your **.niogit** directory, you can also use the Decision Central migration tool on the external repository to prepare project data for Red Hat Decision Manager 7.1. Only files in project directories in the Git repository will be restructured by the migration tool.

Prerequisite

Java 8 is installed and is available on the path where the project migration tool will be installed.

Procedure

1. In a local directory, create a clone of the Git repository containing the projects to be migrated (if the repository has not been cloned already).

   ```
   $ cd $REPO_DIR/
   $ git clone $GIT_REPO_URL
   ```

2. Note the path to the cloned Git repository. The path will be required when you run the migration tool.

3. Create an output directory where a copy of the newly migrated repository or repositories will be placed after the migration tool runs. You can also use an existing directory as an output location. Note the path to this output directory. The migration tool operates on a copy of the cloned repository and will prompt you for the new output directory path when you run the tool.

4. If you have specific **dependencies**, **repositories**, or **pluginRepositories** that you want to add, remove, or update as part of the project migration, create a **pom-migration.json** file containing these elements. When you run the migration tool, the tool will prompt you if you want to include this file in the migration.
The **pom-migration.json** file requires the following sections:

   - **"dependencies":[]**

   - **"repositories-add":[]**

   - **"repositories-remove":[]**

   - **"repositories-update-urls":[]**

   - **"pluginRepositories-add":[]**

   - **"pluginRepositories-remove":[]**

   - **"pluginRepositories-update-urls":[]**

If any of these elements do not apply to your project, include them as empty sections to prevent parsing errors.

Example **pom-migration.json** file:

```json
{
  "dependencies":[

    {"groupId":"junit", "artifactId":"junit", "version":"4.12", "scope":"test"},
    {"groupId":"com.thoughtworks.xstream", "artifactId":"xstream", "version":"4.12",
"scope":"test"}

  ],

  "repositories-add":[
    {
      "id":"jboss-public-repository-group",
      "name":"JBoss Public Repository Group",
      "url":"http://repository.jboss.org/nexus/content/groups/public/",
      "releasesEnabled":true,
      "releasesUpdatePolicy":"never",
      "snapshotEnabled":true,
      "snapshotUpdatePolicy":"never"
    },
    {
      "id":"kie-internal-staging-repository-group",
      "name":"KIE Internal Staging Repositories",
      "url":"https://origin-repository.jboss.org/nexus/content/groups/kie-internal-group",
      "releasesEnabled":true,
      "releasesUpdatePolicy":"never",
      "snapshotEnabled":false,
      "snapshotUpdatePolicy":"never"
    }
  ],

  "repositories-remove":[
    {
      "id":"productization-repository",
"url":"http://download.lab.bos.redhat.com/brewroot/repos/jb-ip-6.1-build/latest/maven/"
    }
  ],

  "repositories-update-urls":[
    {
      "id":"guvnor-m2-repo", "url":"http://localhost:8080/decision-central/maven3/"
    }
  ],

  "pluginRepositories-add":[

    {
      "id":"jboss-public-repository-group",
      "name":"JBoss Public Repository Group",
      "url":"http://repository.jboss.org/nexus/content/groups/public/",
      "releasesEnabled":true,
      "releasesUpdatePolicy":"never",
```

```
      "snapshotEnabled":true,
      "snapshotUpdatePolicy":"never"
    },
    {
      "id":"kie-internal-staging-repository-group",
      "name":"KIE Internal Staging Repositories",
      "url":"https://origin-repository.jboss.org/nexus/content/groups/kie-internal-group",
      "releasesEnabled":true,
      "releasesUpdatePolicy":"never",
      "snapshotEnabled":false,
      "snapshotUpdatePolicy":"never"
    }

  ],
  "pluginRepositories-remove":[],
  "pluginRepositories-update-urls":[]
}
```

Example **pom-migration.json** file with **dependencies** only:

```
{
  "dependencies":[

    {"groupId":"junit", "artifactId":"junit", "version":"4.12", "scope":"test"},
    {"groupId":"com.thoughtworks.xstream", "artifactId":"xstream", "version":"4.12",
"scope":"test"}

  ],
  "repositories-add":[],
  "repositories-remove":[],
  "repositories-update-urls":[],
  "pluginRepositories-add":[],
  "pluginRepositories-remove":[],
  "pluginRepositories-update-urls":[]
}
```

5. Navigate to the Software Downloads page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:

   - **Product:** Decision Manager

   - **Version:** 7.1

6. Download **Red Hat Decision Manager 7.1.0 Add-Ons** and extract the downloaded **rhdm-7.1.0-add-ons.zip** file to a temporary directory.

7. In the extracted **rhdm-7.1.0-add-ons** folder, extract the **rhdm-7.1-migration-tool.zip** sub-folder. The migration tool is in the **bin** directory.

8. In your command terminal, navigate to the temporary directory where you extracted the **rhdm-7.1-migration-tool** folder and run the migration tool. The **$GIT_REPO_PATH** portion is the path to the cloned Git repository.
   On Linux or UNIX-based systems:

```
$ cd $INSTALL_DIR/rhdm-7.1-migration-tool/bin
$ ./migration-tool.sh -t $GIT_REPO_PATH
```

On Windows:

```
$ cd $INSTALL_DIR\rhdm-7.1-migration-tool\bin
$ migration-tool.bat -t $GIT_REPO_PATH
```

9. In the command prompt that appears, enter the path to the output directory where the migrated copy of the repository will be placed. The migration tool operates on a copy of the cloned repository and will place the new repository or repositories in the output location that you specify.
   After you enter the output location, the migration tool prepares the repository copy and restructures all project directories to be compatible with Red Hat Decision Manager 7.1.

   In the command prompt, the following additional migration options are displayed:

   - **POMs migration:** Updates **pom.xml** files with dependencies required for Red Hat Decision Manager 7.1.

   - **All:** Runs all remaining migration options in sequence.

   - **Exit:** Exits the migration tool.

   > **NOTE**
   >
   > The **Project structure migration** option is not displayed because that option was run automatically when you entered the output location to initiate the migration tool.

10. Select the option to run **ALL** migrations in sequence.

    > **NOTE**
    >
    > If you prefer to run one migration option at a time, select and run the first individual migration option. After the tool runs, re-run the Decision Central migration tool and select the next individual migration option in the sequence.

11. Enter **yes** each time you are prompted to run a specific migration option.
    For the POMs migration option, if you want to include a path to an external **pom-migration.json** file that you created previously, enter **yes** when prompted and enter the path.

12. After the tool finishes running, enter the option to **Exit** the migration tool.
    The project directories in the specified output location are now separated into individual repositories compatible with Decision Central in Red Hat Decision Manager 7.1. The new project repositories are bare repositories with no working directory, and therefore do not show any content files. You can clone each repository to create non-bare repositories and inspect directory contents.

13. Log in to Decision Central for Red Hat Decision Manager 7.1.

14. For each project, create or select the space for the project and click **Import Project**.

15. Enter the **Repository URL** for the newly structured project repository. This URL may be the local file path to the output location if you are importing directly from the workstation where you ran the migration tool, or a GitHub URL or other file hosting location where you have uploaded

the repository.
Example: Local file location

```
file:///$OUTPUT_DIR/loan-application.git
```

Example: GitHub repository location

```
https://github.com/$REPO/loan-application.git
```

> **NOTE**
>
> If you use Git **hooks** with your project data, you may need to update your **hooks** scripts to accommodate the migration.

2.3. VERIFY THE MIGRATION IN DECISION CENTRAL

After you have migrated all Decision Central data, verify that the migration was successful in the new Decision Central for Red Hat Decision Manager 7.1.

Prerequisite

Project data from Red Hat Decision Manager 7.0 has been migrated using the Decision Central migration tool.

Procedure

1. Start Red Hat Decision Manager 7.1 depending on your installation.
   Example: Installation with Red Hat JBoss EAP

   ```
   $ ~/$EAP_HOME/bin/standalone.sh -c standalone-full.xml
   ```

2. Log in to Decision Central with your credentials and navigate to Menu → **Design** → **Projects**.

3. Verify that all projects have been migrated and select each project to verify the migrated project assets.

4. For each project, click **Build** and then **Deploy** in the upper-right corner of the project window to validate the migrated project data and the configured Decision Server.

2.3.1. Troubleshooting Decision Central migration problems

If you encounter problems with your Decision Central migration to Red Hat Decision Manager 7.1, review the following troubleshooting suggestions:

- If any project data is missing from Decision Central, ensure that the **.niogit** directory for Red Hat Decision Manager 7.1 contains the restructured data and is in the correct installation location for Red Hat Decision Manager 7.1 (for example, ~/**$EAP_HOME/bin/**).

- If projects fail to build, open the project **pom.xml** file and remove the **http://<SERVER>:<PORT>/decision-central/maven2/** repository, if present. The use of this repository can prevent projects from building properly in Red Hat Decision Manager 7.1. If this repository contains project data that you want to preserve, create a new repository for the data and add the repository as a dependency in the **pom.xml** file.

Example repository to be removed from **pom.xml** file:

```
<repository>
  <id>guvnor-m2-repo</id>
  <name>Guvnor M2 Repo</name>
  <url>http://localhost:8080/decision-central/maven2/</url>
</repository>
```

- If projects fail to deploy to Decision Server, review your Decision Server installation and configuration. For more information, see *Planning a Red Hat Decision Manager installation* .

- If you cannot resolve migration problems, complete the migration process again on a new Red Hat Decision Manager 7.1 installation.

CHAPTER 3. RED HAT BUSINESS OPTIMIZER CHANGES IN RED HAT DECISION MANAGER 7.1

Red Hat Business Optimizer is an embeddable planning engine in Red Hat Decision Manager that optimizes planning problems. Red Hat Business Optimizer is based on the community OptaPlanner project that is regularly updated and in some cases requires code changes for the latest Red Hat Business Optimizer features. For an overview of the latest OptaPlanner changes and migration requirements, see the OptaPlanner upgrade recipe archive . OptaPlanner upgrade information for versions 7.6 through 7.11, inclusive, is relevant for upgrading from Red Hat Decision Manager 7.0 to Red Hat Decision Manager 7.1.

CHAPTER 4. NEXT STEPS

- *Getting started with decision services*

- *Designing a decision service using guided decision tables*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Wednesday, March 27, 2019.