Red Hat Decision Manager 7.6

Configuring Business Central settings and properties
Abstract

This document describes how to customize and manage the various features of Business Central in Red Hat Decision Manager 7.6.
# Table of Contents

PREFACE ................................................................. 4

CHAPTER 1. USER MANAGEMENT ........................................... 5
1.1. CREATING USERS ........................................... 5
1.2. EDITING USERS ........................................... 6
1.3. CREATING GROUPS ........................................... 6
1.4. EDITING GROUPS ........................................... 6

CHAPTER 2. SECURITY MANAGEMENT ...................................... 8
2.1. CHANGING PERMISSIONS FOR GROUPS AND ROLES ............. 8

CHAPTER 3. ARTIFACT MANAGEMENT .................................... 9
3.1. VIEWING AN ARTIFACT ....................................... 9
3.2. DOWNLOADING AN ARTIFACT ................................ 9
3.3. UPLOADING AN ARTIFACT .................................... 9

CHAPTER 4. DATA SOURCE MANAGEMENT ................................. 11
4.1. ADDING A DATABASE DRIVER .................................. 11
4.2. EDITING A DATABASE DRIVER ................................ 11
4.3. DELETING A DATABASE DRIVER ................................ 11
4.4. ADDING A DATA SOURCE ...................................... 12
4.5. EDITING A DATA SOURCE ..................................... 12
4.6. DELETING A DATA SOURCE .................................... 12

CHAPTER 5. DATA SETS AUTHORIZATION .................................. 14
5.1. ADDING DATA SETS ........................................... 14
5.2. EDITING DATA SETS ........................................... 14
5.3. DATA REFRESH .................................................. 15
5.4. CACHING ....................................................... 15
   Client cache ..................................................... 16
   Back-end cache ............................................... 16

CHAPTER 6. CUSTOMIZING PROJECT PREFERENCES ....................... 17

CHAPTER 7. CUSTOMIZING ARTIFACT REPOSITORY PROPERTIES ........ 19

CHAPTER 8. CUSTOMIZING LANGUAGE SETTINGS ........................ 20

CHAPTER 9. CUSTOMIZING PROCESS ADMINISTRATION .................. 21

CHAPTER 10. CUSTOMIZING THE PROCESS DESIGNER ..................... 22

CHAPTER 11. SSH KEYS ................................................ 23
11.1. DEFAULT SSH KEYSTORE ..................................... 23
11.2. CUSTOM SSH KEYSTORE ..................................... 23
11.3. CREATING AN SSH KEY ...................................... 24
11.4. REGISTERING YOUR SSH PUBLIC KEY WITH THE SSH KEYSTORE .................................................................................. 24
11.5. DELETING AN SSH KEY ....................................... 25

CHAPTER 12. MANAGING SERVICE TASKS IN BUSINESS CENTRAL ........ 26

CHAPTER 13. EXPORTING AND IMPORTING DASHBUILDER DATA .......... 30
13.1. EXPORTING DASHBUILDER DATA ................................ 30
13.2. IMPORTING DASHBUILDER DATA ................................ 30
PREFACE

As an administrator, you can customize the following on the admin **Settings** page:

- **Roles**: Set the home page, priority, and permissions of a role.
- **Groups**: Set the home page, priority, and permissions of a group as well as create and delete groups.
- **Users**: Create and delete users, add or remove groups and roles from users, and view user permissions.
- **Artifacts**: View M2 repository artifacts, upload artifacts, view, and download JAR files.
- **Data Sources**: Add, update, or delete data sources and database drivers.
- **Data Sets**: Create, modify, or delete data sets.
- **Projects**: View and edit project preferences such as file export properties, space properties, default values, and advanced GAV properties.
- **Artifact Repository**: Manage artifact repository properties.
- **Languages**: Set the Business Central language.
- **Process Administration**: Set the default pagination option in Business Central.
- **Process Designer**: Set diagram editor properties.
- **SSH Keys**: Add or delete SSH keys.
- **Service Tasks Administration**: Enable or disable default service tasks and upload custom service tasks.
- **Profiles**: Set the workbench profile as **Planner and Rules** or **Full**.

**Prerequisites**

- Red Hat JBoss Enterprise Application Platform 7.2.0 is installed. For more information, see [Red Hat JBoss Enterprise Application Platform 7.2 Installation Guide](#).
- Red Hat Decision Manager is installed and running. For more information, see [Installing and configuring Red Hat Decision Manager on Red Hat JBoss EAP 7.2](#).
- You are logged in to Business Central with the **admin** user role.
CHAPTER 1. USER MANAGEMENT

Business Central defines three types of entities for security management: users, groups, and roles. You can assign permissions to both roles and groups. You can assign the following roles in Business Central:

- process-admin
- manager
- admin
- analyst
- developer
- user

NOTE

User roles in the application Role Registry have a role identifier, whereas user groups do not.

Use Business Central to create and manage as many users and groups as you require. A user must be assigned to at least one user-specific role to log in to Business Central. User privileges depend on permissions from the groups and roles that the user is a member of. Note that the role or group priority is considered if a user has several roles or groups assigned to it.

1.1. CREATING USERS

User privileges and settings are controlled by the roles assigned to a user and the groups that a user belongs to. You can create any number of users in Business Central.

NOTE

Do not create a user called unknown in process engine or Decision Server. The unknown user account is a reserved system name with superuser access. The unknown user account performs tasks related to the SLA violation listener when there are no users logged in.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Users.

2. Click New user, enter a user name, and then click Next.

3. To assign roles to the user, click the Roles tab, click Add Roles, select the desired roles, and click Add to selected roles.

4. Optionally, to assign groups to the user, click the Groups tab, click Add to groups, select the desired groups, and click Add to selected groups.

5. Click Create.

6. Click Yes to set a password for the user, enter a desired password, and click Change.
NOTE
The user must have at least one role to access Business Central.

1.2. EDITING USERS

You can change the group and role of a user using the Users option on the Business Central Settings page. All user permissions are based on the group and role permissions of the user. You can view the user permissions from the Permissions tab.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Users.
2. From the All users list, click the user you want to edit. The user details display in the right pane.
3. Click Edit to perform any of the following tasks:
   - To change the groups of a user, click the Groups tab, click Add to groups, select the groups you want the user to be part of, click Add to selected groups, and click Save.
   - To change the roles of a user, click the Roles tab, click Add roles, select the roles you want to assign to the user, click Add to selected roles, and click Save.
   - To view the user permissions, click the Permissions tab and expand the attributes.
   - To change the password, click Change Password, enter the new password, and click Change.
   - To delete the user, click Delete and then click Yes to confirm removal.

1.3. CREATING GROUPS

In Business Central, you can use groups to control permissions for a collection of users. You can create as many groups as you want but a group must have at least one user.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Groups.
2. Click New group, enter a group name, and then click Next.
3. Select the users that you want to add to this group, and then click Add selected users. The newly created group is listed under All groups.

1.4. EDITING GROUPS

You can edit the attribute of a group such as home page, priority, and permissions according to your requirements. From the Groups option on the Business Central Settings page, you can modify or delete a group.

Procedure
1. In Business Central, select the Admin icon in the top-right corner of the screen and select Groups.

2. From the All groups list, click the group that you want to edit. The user details display in the right pane.

3. Select the home page from the Home Page list.

4. Select the priority from the Priority list.

5. In the Permissions section, expand the resource attribute and change its permission.

   **NOTE**

   You can add exceptions to Pages, Editor, Spaces, and Projects permissions.

6. Click Save to apply the changes.
CHAPTER 2. SECURITY MANAGEMENT

You can perform security and permission management tasks from the Security management page. To access the Security management page, select the Admin icon in the top-right corner of the screen and click Roles, Groups, or Users on the Business Central Settings page. Depending on which icon you click, the Security management page opens for the respective tab. Use roles and groups to manage privileges and settings for users. Any changes apply to all users with the affected role or group.

2.1. CHANGING PERMISSIONS FOR GROUPS AND ROLES

You cannot change permissions for an individual user. However, you can change permissions for groups and roles. The changed permissions apply to users that have the role or belong to the group that you changed.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen.
2. Click Roles or Groups to open the corresponding list.
3. From the list, click the role or group you want to edit. All details are displayed in the right pane.
4. Set the Home Page or Priority under the Settings section.
5. Set the Business Central, page, editor, space, and project permissions under the Permissions section.
   You can expand the resource type whose permissions you want to change.
6. Optional: To add an item not listed, click Add Exception and then set the permissions as required.

   **NOTE**
   You cannot add an exception to the Business Central resource type.
7. Click Save.

   **NOTE**
   Any changes you make to roles or groups affect all of the users associated with that role or group.
CHAPTER 3. ARTIFACT MANAGEMENT

You can manage artifacts from the **Artifacts** page in Business Central. The artifact repository is a local Maven repository and there is only one Maven repository for each installation. Business Central recommends using Maven repository solutions like **Sonatype Nexus™**, **Apache Archiva™**, or **JFrog Artifactory™**.

The **Artifacts** page lists all the artifacts in the Maven repository. You can upload artifacts to the Maven repository.

**NOTE**

You can only upload JAR, KJAR, and **pom.xml** files to the **Artifacts** repository.

### 3.1. VIEWING AN ARTIFACT

You can view all the content of the local maven repository from the **Artifacts** page.

**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Artifacts**.
2. Click **Open** to view the artifact details.
3. Click **Ok** to go back to the **Artifacts** page.

### 3.2. DOWNLOADING AN ARTIFACT

You can download and save an artifact from Business Central repository to the local storage of a project.

**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Artifacts**.
2. Click **Download**.
3. Browse to the directory where you want to save the artifact.
4. Click **Save**.

### 3.3. UPLOADING AN ARTIFACT

You can upload an artifact from the local storage to a project in Business Central.

**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Artifacts**.
2. Click **Upload**.
3. Click **Choose File** and browse to the directory from where you want to upload the artifact.

4. Click **Upload**.

**NOTE**

If you are using a non-Maven artifact, first deploy the artifact to the Maven repository using the `mvn deploy` command and then refresh the artifact list in Business Central.
CHAPTER 4. DATA SOURCE MANAGEMENT

Business Central provides data source management features that enable you to define data sources for accessing a database. These data sources are then used by other Business Central components such as the data sets. A database driver enables communication between a data source and the targeted database.

From the Data Source Authoring page you can add data sources and database drivers to Business Central.

NOTE

Business Central provides a default data source that can be used but cannot be edited or deleted.

4.1. ADDING A DATABASE DRIVER

You can add a new database driver to Business Central.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Data Sources.

2. In the DataSource Explorer pane, click Add Driver. The New driver window opens.

3. In the New driver window, enter the Name, Driver Class Name, Group Id, Artifact Id, and Version of the database driver.

4. Click Finish to add the driver to Business Central.

4.2. EDITING A DATABASE DRIVER

You can update the properties of a database driver from the Driver Definition pane.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Data Sources.

2. In the DataSource Explorer pane, select the driver you want to edit.

3. In the Driver Definition pane, make the necessary changes to the Name, Driver Class Name, Group Id, Artifact Id, and the Version fields.

4. Click Update.

5. Click Yes to save the changes to the driver.

4.3. DELETING A DATABASE DRIVER

You can remove database drivers from the Data Source Definition pane of Business Central.
**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Data Sources**.

2. In the **DataSource Explorer** pane, select the driver you want to delete.

3. Click **Remove**.

4. Click **Delete** to delete the driver.

**4.4. ADDING A DATA SOURCE**

You can add a new data source to Business Central from the **Data Sources Authoring** page.

**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Data Sources**.

2. In the **DataSource Explorer** pane, click **Add DataSource**. The **New data source** window opens.

3. In the **New data source** window, enter the data source **Name**, database **Connection URL**, **User** and **Password**, and **Driver**.

4. Click **Test Connection** to verify the connection to the database.

5. Click **Finish** to add the data source to Business Central.

**4.5. EDITING A DATA SOURCE**

You can edit the properties of a data source and also test its connection to the database in Business Central.

**Procedure**

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Data Sources**.

2. In the **DataSource Explorer** pane, click the data source you want to edit.

3. In the **Data Source Definition** pane, make the necessary changes to the **Name**, **Connection URL**, **User**, **Password**, and the **Driver** fields.

4. Click **Test Connection** to verify the connection to the database.

5. Click **Update**.

6. Click **Save** to confirm the changes to the data source.

**4.6. DELETING A DATA SOURCE**

You can delete an existing data source from the **DataSource Explorer** pane in Business Central.

**Procedure**

Red Hat Decision Manager 7.6 Configuring Business Central settings and properties
1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Data Sources**.

2. In the **DataSource Explorer** pane, click the data source you want to delete.

3. Click **Remove**.

4. Click **Delete** to confirm the deletion of the data source.
CHAPTER 5. DATA SETS AUTHORING

A data set is a collection of related sets of information. It can be stored in many ways, such as in a database, in a Microsoft Excel file, or in memory. A data set definition instructs Business Central methods to access, read, and parse a data set. Business Central does not store data. It enables you to define access to a data set regardless of where the data is stored.

For example, if data is stored in a database, a valid data set could contain the entire database or a subset of the database as a result of an SQL query. In both cases, the data is used as input for the reporting components of Business Central which then displays the information.

To access a data set, you must create and register a data set definition. This data set definition specifies the location of the data set, the ways to access it, read it, and parse it, and the columns that it contains.

NOTE
The Data Sets page is visible only to users with the admin role.

5.1. ADDING DATA SETS

You can create a data set to fetch data from an external data source and use that data for the reporting components.

Procedure

1. In Business Central, go to Admin → Data Sets. The Data Set Explorer page opens.
2. Click New Data Set and select one of the following provider types:
   - Bean: Use to generate a data set from a Java class
   - CSV: Use to generate a data set from a remote or local CSV file
   - SQL: Use to generate a data set from an ANSI-SQL compliant database
   - Elastic Search: Use to generate a data set from Elastic Search nodes
   - Execution Server: Use to generate a data set using the custom query feature of an Execution Server

   NOTE
   KIE Server must be configured with this option.

3. Complete the Data Set Creation Wizard and click Test.

   NOTE
   The configuration steps differ based on the provider you chose.

4. Click Save.

5.2. EDITING DATA SETS
You can edit existing data sets to ensure that the data fetched to the reporting components is up-to-date.

**Procedure**

1. In Business Central, go to **Admin → Data Sets**. The **Data Set Explorer** page opens.
2. In the **Data Set Explorer** pane, search for the data set you want to edit and click **Edit**.
3. In the **Data Set Editor** pane, use the appropriate tab to edit the data as required. The tabs differ based on the data set provider type you chose. For example, the following changes are applicable for editing a CSV data provider:
   - **CSV Configuration**: Enables you to change the name of the data set definition, the source file, the separator, and other properties.
   - **Preview**: Enables you to preview the data. After you click **Test** in the **CSV Configuration** tab, the system executes the data set lookup call and if the data is available, a preview appears. Note that the **Preview** tab has two sub-tabs:
     - **Data columns**: Enables you to specify what columns are part of your data set definition.
     - **Filter**: Enables you to add a new filter.
   - **Advanced**: Enables you to manage the following configurations:
     - **Caching**: See Section 5.4, “Caching” for more information.
     - **Cache life-cycle**: Enables you to specify an interval of time after which a data set (or data) is refreshed. The **Refresh on stale data** feature refreshes the cached data when the back-end data changes.
4. After making the required changes, click **Validate**.
5. Click **Save**.

### 5.3. DATA REFRESH

The data refresh feature enables you to specify an interval of time after which a data set (or data) is refreshed. The **Refresh on stale data** feature refreshes the cached data when the back-end data changes.

### 5.4. CACHING

Business Central provides caching mechanisms for storing data sets and performing data operations using in-memory data. Caching data reduces network traffic, remote system payload, and processing time. To avoid performance issues, configure the cache settings in Business Central.

For any data lookup call that results in a data set, the caching method determines where the data lookup call is executed and where the resulting data set is stored. An example of a data lookup call would be all the mortgage applications whose locale parameter is set as "Urban".

Business Central data set functionality provides two cache levels:

- **Client level**
- Back-end level

**Client cache**
When the cache is turned on, the data set is cached in a web browser during the lookup operation and further lookup operations do not perform requests to the back-end. Data set operations like grouping, aggregations, filtering, and sorting are processed in the web browser. Enable client caching only if the data set size is small, for example, for data sets with less than 10 MB of data. For large data sets, browser issues such as slow performance or intermittent freezing can occur. Client caching reduces the number of back-end requests including requests to the storage system.

**Back-end cache**
When the cache is enabled, the decision engine caches the data set. This reduces the number of back-end requests to the remote storage system. All data set operations are performed in the decision engine using in-memory data. Enable back-end caching only if the data set size is not updated frequently and it can be stored and processed in memory. Using back-end caching is also useful in cases with low latency connectivity issues with the remote storage.

**NOTE**
Back-end cache settings are not always visible in the Advanced tab of the **Data Set Editor** because Java and CSV data providers rely on back-end caching (data set must be in the memory) in order to resolve any data lookup operation using the in-memory decision engine.
CHAPTER 6. CUSTOMIZING PROJECT PREFERENCES

A project stores assets and is part of a space. A space can hold multiple projects.

For example, an organization has many departments, such as HR, Payroll, Engineering, R&D, and so on. Each department maps to a space and every department can have its own projects.

You can create a new project from scratch or clone projects from an existing Git repository.

**Procedure**

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Projects. The Projects page opens.

2. In the Project Preferences pane, click the preference you want to modify. The following preferences are available:
   - **File exporting**: This preference has the following properties:
     
     **Table 6.1. File exporting properties**

     | Field            | Description                                                                 |
     |------------------|-----------------------------------------------------------------------------|
     | PDF orientation  | Determines whether the PDF orientation is portrait or landscape.            |
     | PDF units        | Determines whether the PDF unit is PT, MM, CN or IN.                        |
     | PDF page format  | Determines whether the PDF page format is A[0-10], B[0-10], or C[0-10].     |

   - **Spaces**: This preference has the following properties:
     
     **Table 6.2. Spaces properties**

     | Field            | Description                                                                 |
     |------------------|-----------------------------------------------------------------------------|
     | Name             | The default name of the space that is created automatically if none exists. |
     | Owner            | The default owner of the space that is created automatically if none exists.|
     | Group ID         | The default group ID of the space that is created automatically if none exists.|
     | Alias (in singular) | Determines the customized alias (singular) of the space.                  |
### Default values

This preference has the following properties:

**Table 6.3. Default values properties**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>The default version number of a project when creating projects using the Quick setup option.</td>
</tr>
<tr>
<td>Description</td>
<td>The default description of a project when creating projects using the Quick setup option.</td>
</tr>
<tr>
<td>Branch</td>
<td>The default branch to be used when using a Git repository.</td>
</tr>
</tbody>
</table>

### Advanced GAV preferences

This preference has the following properties:

**Table 6.4. Advanced GAV preference properties**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable GAV conflict check?</td>
<td>Determines whether to enable or disable the GAV conflict check. Disabling this feature allows projects to have the same GAV (Group ID, Artifact, Version).</td>
</tr>
<tr>
<td>Allow child GAV edition?</td>
<td>Determines whether to allow child/sub-projects to have GAV edition.</td>
</tr>
</tbody>
</table>

**NOTE**

Duplicate GAV detection is disabled for projects in **Development Mode**. To enable duplicate GAV detection for a project in Business Central, go to project **Settings → General Settings → Version** and toggle the **Development Mode** option to **OFF** (if applicable).

3. Click **Save**.
CHAPTER 7. CUSTOMIZING ARTIFACT REPOSITORY PROPERTIES

In some cases, projects need to resolve external dependencies to build domain model JAR files. A repository contains the needed artifacts and has the following features:

- The repository is a Maven repository.
- All snapshots are time stamped.
- Assets are stored mostly in the local hard drive.

By default, the artifact repository is in `${WORKING_DIRECTORY}/repositories/kie`.

**Procedure**

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Artifact Repository. The Artifact Repository page opens.

2. Make selections and enter information in the Properties section.

3. Click Save.
CHAPTER 8. CUSTOMIZING LANGUAGE SETTINGS

You can change the language on the Business Central Settings page. Business Central supports the following languages:

- English
- German
- Spanish
- French
- Japanese
- Portuguese
- Chinese (Simplified)

The default language is English.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Languages. The Language Selector window opens.

2. Select the desired language from the Language list.

3. Click Ok.
CHAPTER 9. CUSTOMIZING PROCESS ADMINISTRATION

You can customize the default pagination option in Business Central by editing the Default items per page property on the Process Administration page.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Process Administration.

2. From the Properties section, update the Default items per page property and click Save.

NOTE

You can specify 10, 20, 50, or 100 items to display on each page.
CHAPTER 10. CUSTOMIZING THE PROCESS DESIGNER

You can customize the process designer in Business Central by editing the properties of the diagram editor on the Business Central Settings page.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Process Designer.

2. In the Properties section, update any of the following properties:
   - Select the Auto hide category panel check box to automatically hide a category toolbar panel.
   - In the Drawing area width field, enter an integer value between 2800 and 5600 to set the width of the drawing area.
   - In the Drawing area height field, enter an integer value between 1400 and 2800 to set the height of the drawing area.
   - Select the Enable HiDPI check box if you are using a high resolution display and are seeing blurry text and objects. This option is disabled by default.

3. Click Save.
CHAPTER 11. SSH KEYS

Business Central provides an SSH keystore service to enable user SSH authentication. Business Central provides a configurable default SSH keystore, extensible APIs (for custom implementations), and support for multiple SSH public key formats.

You can access the SSH Keys option on the Business Central Settings page to register your SSH public keys.

11.1. DEFAULT SSH KEYSTORE

The default SSH keystore included with Business Central provides a file-based storage mechanism to store a user’s public keys. By default, Business Central uses the *.security folder as the root directory. However, you can also use a custom storage path by setting the value of the appformer.ssh.keys.storage.folder system property to point to a different folder.

The SSH public keys are stored in the {securityFolderPath}/pkeys/{userName}/ folder structure.

Each SSH public key consists of the following files, located in the storage folder:

- `{keyId}.pub`: This file contains the SSH public key content. As the file name determines the logic key ID on the system, ensure that the file name is not modified during run time.
  
  For example:

  ```
  ssh-rsa
  AAAAB3NzaC1yc2EAAAADAQABAAABAADmak4Wu23RZ6 XmN94bOsqecZxuTa4RRhhQmHMfTzMB7HM57/90u/B/g/B/GhsPEx1nAXL0npY56IT/MPQ8vRm2C2W9A7CnN5+Z5yyL3W01YZy3zk77CjULIfhrfQSL3b2sPG5j5vE5/nyC/swSytcwT/PE7aXTS9H6cHIKUdYPzl94SHoBxWRK7PjI9d-eLB+hmd2v/b/va1ezu5a8yuy2kchH6Nxxfl5iRj2rscdTfp0imC1jmOC6ZDfBvZSxL9Fv
  TMwFdNnmTJveBtv9nAbnAvlWllS0V0Okgdj1s3GxBxeYAcKbcsK9sJzusptk5dxGszGZ8vlnaqfN6oaQ7b7lcomzCYYwviGOq9gRX8sgsVrw39ggsDIGYP2tA4bRr7ecHn lng1b0HCchA5+QCDk4Hbzu1unHmPA2Lg9c3WGM2qedvQdVJ XuSMiwYoqL40aXPs6890PvFUtpyVSznF50djPnwsMxJZEf1HdTXgdBDhS4ogZ17czyUNlkNkE69yJDbThJqDq0cKUqn6tVxqmBZhX31yF4VcsMeADc1L28wlA3n4LZnC/Gwony/95/G93zJpF0kPhme8c2XuPuCXF795lsxyJ8SB/Aw/PAJhEm0y0s0l114eWqxsDxKBOGniU0czrVMssHJEJb4o0FL7iHhOW56/iMdD9w== userName
  ```

- `{keyId}.pub.meta`: This file contains the key metadata in JSON format. A new metadata file is dynamically generated if a key has no metadata.
  
  For example:

  ```
  {
  "name":"Key",
  "creationDate":"Oct 10, 2018 10:10:50 PM",
  "lastTimeUsed":"Oct 11, 2018 12:11:23 PM"
  }
  ```

11.2. CUSTOM SSH KEYSTORE

You can extend and customize the default SSH keystore according to your requirements. Use the appformer.ssh.keystore system property to specify the Java class name of the SSH service to use. If this property is not defined or it contains an incorrect value, then the default SSH keystore is loaded.
NOTE

To create a custom implementation of the SSH keystore, your Java class must implement the `org.uberfire.ssh.service.backend.keystore.SSHKeyStore` class defined in the `uberfire-ssh-api` module.

11.3. CREATING AN SSH KEY

Before you can add or register SSH keys to Business Central, you must generate an SSH key on your system.

Procedure

1. Open a command terminal on your system.

2. Run the `ssh-keygen` command to create the SSH key as shown in the following example, where `<user_login>` is your user name:

   ```bash
   ssh-keygen -t rsa -b 4096 -C "<user_login>"
   ```

   **NOTE**
   The SSH key formats supported by Business Central keystore are `ssh-rsa`, `ssh-dss`, `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp384`, and `ecdsa-sha2-nistp521`.

3. When prompted, press Enter and accept the default key file location as shown in the following example, where `<user_login>` is your user name:

   ```text
   Enter a file in which to save the key (/home/<user_login>/.ssh/id_rsa): [Press enter]
   ```

4. At the command prompt, enter and confirm the passphrase:

   ```text
   Enter passphrase (empty for no passphrase): [Type a passphrase]
   Enter same passphrase again: [Type passphrase again]
   ```

5. Start the `ssh-agent`:

   ```bash
   eval "$(ssh-agent -s)"
   Agent pid <any-number-here>
   ```

6. Add the new SSH private key to the `ssh-agent`. If you have used a different key name, replace `id_rsa` in that code:

   ```bash
   ssh-add ~/.ssh/id_rsa
   ```

11.4. REGISTERING YOUR SSH PUBLIC KEY WITH THE SSH KEystore

You must register your newly created SSH public key with the Business Central keystore.

Procedure

1. Open a command terminal on your system.
2. Run the `cat` command as shown in the following example, where `id_rsa` is your key name:

   ```bash
   cat ~/.ssh/id_rsa.pub
   ```

3. Copy the contents of your SSH public key.

4. In Business Central, select the `Admin` icon in the top-right corner of the screen and select `SSH Keys`.

5. On the `SSH Keys` page, click `Add SSH Key`.

6. In the `Add SSH Key` window, enter a name in the `Name` field and copy the contents of the SSH public key to the `Key` field.

   **NOTE**

   The `Name` and the `Key` fields are mandatory.

7. Click `Add SSH Key` to register the key.

### 11.5. DELETING AN SSH KEY

You can delete an SSH key from Business Central by from the `SSH Keys` page.

**Procedure**

1. In Business Central, select the `Admin` icon in the top-right corner of the screen and select `SSH Keys`.

2. On the `SSH Keys` page, click the delete icon of the SSH key you want to delete.

3. Click `Delete SSH Key` to confirm the deletion.
CHAPTER 12. MANAGING SERVICE TASKS IN BUSINESS CENTRAL

Service tasks (work items) are tasks that you can customize and reuse across multiple business processes or across all projects in Business Central. Red Hat Decision Manager provides a set of service tasks within the service task repository in Business Central. You can enable or disable the default service tasks and upload custom service tasks into Business Central to implement the tasks in the relevant processes.

NOTE

Red Hat Decision Manager includes a limited set of supported custom tasks. Custom tasks that are not included in Red Hat Decision Manager are not supported.

Procedure

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Service Tasks Administration. This page lists the service task installation settings and available service tasks for processes in projects throughout Business Central. The service tasks that you enable on this page become available in the project-level settings where you can then install each service task to be used in processes. The way in which the service tasks are installed in a project is determined by the global settings that you enable or disable under Settings on this Service Tasks Administration page.

2. Under Settings, enable or disable each setting to determine how the available service tasks will be implemented when a user installs them at the project level. The following service task settings are available:

   - **Install as Maven artifact** Uploads the service task JAR file to the Maven repository that is configured with Business Central, if the file is not already present.

   - **Install service task dependencies into project** Adds any service task dependencies to the pom.xml file of the project where the task is installed.

   - **Use version range when installing service task into project** Uses a version range instead of a fixed version of a service task that is added as a project dependency. Example: [7.16,) instead of 7.16.0.Final

3. Enable or disable (set to ON or OFF) any available service tasks as needed. Service tasks that you enable will be displayed in project-level settings for all projects in Business Central.
To add a custom service task, click Add Service Task, browse to the relevant JAR file, and click the Upload icon. The JAR file must contain work item handler implementations annotated with @Wid.

After you configure all required service tasks, navigate to a project in Business Central and go to the project Settings → Service Tasks page to view the available service tasks that you enabled.

For each service task, click Install to make the task available to the processes in that project or click Uninstall to exclude the task from the processes in the project.
7. If you are prompted for additional information when you install a service task, enter the required information and click **Install** again. The required parameters for the service task depend on the type of task. For example, rule and decision tasks require artifact GAV information (Group ID, Artifact ID, Version), email tasks require host and port access information, and REST tasks require API credentials. Other service tasks might not require any additional parameters.

**Figure 12.2. Install service tasks for use in processes**

8. Click **Save**.

9. Return to the project page, select or add a business process in the project, and in the process designer palette, select the **Service Tasks** option to view the available service tasks that you enabled and installed:
Figure 12.3. Access installed service tasks in process designer
CHAPTER 13. EXPORTING AND IMPORTING DASHBUILDER DATA

Dashbuilder is a dashboard and reporting tool integrated in Business Central and is used by the Datasets editor and Content Manager page. There are three data types:

- Datasets
- Perspectives
- Navigation

You can import and export Dashbuilder data as ZIP files in Business Central.

### 13.1. EXPORTING DASHBUILDER DATA

You can export all Dashbuilder data from Business Central as a ZIP file.

**Procedure**

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Dashbuilder Data Transfer.

2. On the Dashbuilder Data Transfer page, click Export.

   An export.zip file containing all Dashbuilder data is downloaded. The export.zip file structure is separated by data type. For example:

   ```
   dashbuilder/datasets/definitions/dataset-example1.csv
   dashbuilder/datasets/definitions/dataset-example1.dset
   dashbuilder/datasets/definitions/dataset-example2.csv
   dashbuilder/datasets/definitions/dataset-example2.dset
   dashbuilder/datasets/readme.md
   dashbuilder/perspectives/page1/perspective_layout
   dashbuilder/perspectives/page1/perspective_layout.plugin
   dashbuilder/perspectives/page2/perspective_layout
   dashbuilder/perspectives/page2/perspective_layout.plugin
   dashbuilder/perspectives/readme.md
   dashbuilder/navigation/navigation/navtree.json
   dashbuilder/navigation/readme.md
   ```

### 13.2. IMPORTING DASHBUILDER DATA

You can import Dashbuilder data to Business Central from a ZIP file if the archive is structured in the same way as the following example:

```
 dashbuilder/datasets/definitions/dataset-example1.csv
 dashbuilder/datasets/definitions/dataset-example1.dset
 dashbuilder/datasets/definitions/dataset-example2.csv
```
Procedure

1. In Business Central, select the **Admin** icon in the top-right corner of the screen and select **Dashbuilder Data Transfer**.

   ![WARNING]
   You should only import Dashbuilder data to a clean installation of Red Hat Decision Manager in order to avoid overwriting data on an existing system.

2. On the **Dashbuilder Data Transfer** page, click the **Choose File** icon.

3. Navigate to the ZIP file you want to import and select the file.

4. Click the **Upload** icon.

5. Click **Import**.
CHAPTER 14. LDAP CONNECTION

Business Central provides a dedicated UserGroupCallback implementation for LDAP servers with Red Hat Decision Manager to enable the user task service to retrieve information on users, groups, and roles directly from an LDAP service.

You can configure the following LDAP UserGroupCallback implementation properties:

Table 14.1. LDAP UserGroupCallback properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap.bind.user</td>
<td>User name for connecting to the LDAP server. This property is optional if it is not specified and the LDAP server accepts anonymous access.</td>
</tr>
<tr>
<td>ldap.bind.pwd</td>
<td>Password for connecting to the LDAP server. This property is optional if it is not specified and the LDAP server accepts anonymous access.</td>
</tr>
<tr>
<td>ldap.user.ctx</td>
<td>Context in LDAP with user information.</td>
</tr>
<tr>
<td>ldap.role.ctx</td>
<td>Context in LDAP with group and role.</td>
</tr>
<tr>
<td>ldap.user.roles.ctx</td>
<td>Context in LDAP with user group and role membership information.</td>
</tr>
<tr>
<td></td>
<td>This property is optional if it is not specified and the ldap.role.ctx property is used instead.</td>
</tr>
<tr>
<td>ldap.user.filter</td>
<td>Filter for searching user information.</td>
</tr>
<tr>
<td></td>
<td>This property usually contains substitution keys {0} that are replaced with parameters.</td>
</tr>
<tr>
<td>ldap.role.filter</td>
<td>Filter for searching group and role information.</td>
</tr>
<tr>
<td></td>
<td>This property usually contains substitution keys {0} that are replaced with parameters.</td>
</tr>
<tr>
<td>ldap.user.roles.filter</td>
<td>Filter for searching user group and role membership information.</td>
</tr>
<tr>
<td></td>
<td>This property usually contains substitution keys {0} that are replaced with parameters.</td>
</tr>
<tr>
<td>ldap.user.attr.id</td>
<td>Attribute name of the user ID in LDAP.</td>
</tr>
<tr>
<td></td>
<td>This property is optional if it is not specified and the uid property is used instead.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ldap.roles.attr.id</td>
<td>Attribute name of the group and role ID in LDAP. This property is optional if it is not specified and the cn property is used instead.</td>
</tr>
<tr>
<td>ldap.user.id.dn</td>
<td>User ID in a DN, instructs the callback to query for user DN before searching for roles. This is optional and is false by default.</td>
</tr>
<tr>
<td>java.naming.factory.initial</td>
<td>Initial context factory class name; is com.sun.jndi.ldap.LdapCtxFactory by default.</td>
</tr>
<tr>
<td>java.naming.security.authentication</td>
<td>Authentication type where the possible values are none, simple, and strong. This is simple by default.</td>
</tr>
<tr>
<td>java.naming.security.protocol</td>
<td>Security protocol to be used, for example, ssl.</td>
</tr>
<tr>
<td>java.naming.provider.url</td>
<td>LDAP url (by default ldap://localhost:389; if the protocol is set to ssl then ldap://localhost:636)</td>
</tr>
</tbody>
</table>

### 14.1. USING LDAP USERGROUPCALLBACK IMPLEMENTATION

You can use the LDAP UserGroupCallback implementation by configuring the respective LDAP properties in one of the following ways:

- Programmatically: build a properties object with the respective LDAPUserGroupCallbackImpl properties and create LDAPUserGroupCallbackImpl with the properties object as its parameter.
  
  For example:

  ```java
  import org.kie.api.PropertiesConfiguration;
  import org.kie.api.task.UserGroupCallback;
  ...
  Properties properties = new Properties();
  properties.setProperty(LDAPUserGroupCallbackImpl.USER_CTX, "ou=People,dc=my-domain,dc=com");
  properties.setProperty(LDAPUserGroupCallbackImpl.ROLE_CTX, "ou=Roles,dc=my-domain,dc=com");
  properties.setProperty(LDAPUserGroupCallbackImpl.USER_ROLES_CTX, "ou=Roles,dc=my-domain,dc=com");
  properties.setProperty(LDAPUserGroupCallbackImpl.USER_FILTER, "(uid={0})");
  properties.setProperty(LDAPUserGroupCallbackImpl.ROLE_FILTER, "(cn={0})");
  properties.setProperty(LDAPUserGroupCallbackImpl.USER_ROLES_FILTER, "(member={0})");
  UserGroupCallback ldapUserGroupCallback = new LDAPUserGroupCallbackImpl(properties);
  ```
LDAPUserGroupCallbackImpl(properties);
UserGroupCallbackManager.getInstance().setCallback(ldapUserGroupCallback);

- Declaratively: create the `jbpm.usergroup.callback.properties` file in the root of your application or specify the file location as a system property. For example:

  ```
  -Djbpm.usergroup.callback.properties=FILE_LOCATION_ON_CLASSPATH
  ```

  Ensure that you register the LDAP callback when starting the user task server. For example:

  ```
  #ldap.bind.user=
  #ldap.bind.pwd=
  ldap.user.ctx=ou=People,dc=my-domain,dc=com
  ldap.role.ctx=ou=Roles,dc=my-domain,dc=com
  ldap.user.roles.ctx=ou=Roles,dc=my-domain,dc=com
  ldap.user.filter=(uid={0})
  ldap.role.filter=(cn={0})
  ldap.user.roles.filter=(member={0})
  #ldap.user.attr.id=
  #ldap.roles.attr.id=
  ```

Additional resources

- Roles and users
- Red Hat Single Sign-On Server Administration Guide
- Defining LDAP login domain
- LDAP login module
- LDAPExtended login module
- AdvancedLDAP login module
- AdvancedAdLDAP login module
- LDAP connectivity options
- LDAPUsers login module
Java application development uses the Apache Maven build automation tool to build and manage software projects. Maven uses Project Object Model (POM) configuration XML files to define both, the project properties and the build process.

Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, such as http:// when located on an HTTP server, or file:// when located on a file server. The default repository is the public remote Maven 2 Central Repository. Configuration of Maven is performed by modifying the settings.xml file. You can either configure global Maven settings in the M2_HOME/conf/settings.xml file, or user-level settings in the USER_HOME/.m2/settings.xml file.

Additional resources

- Configuring an external Maven repository for Business Central and Decision Server
- Packaging and deploying a Red Hat Decision Manager project in Maven
- Maven settings and repositories for Red Hat Decision Manager
- System integration with Maven
- Welcome to Apache Maven
- Apache Maven Project – Introduction to Repositories
- Apache Maven Parent POMs Reference
CHAPTER 16. GAV CHECK MANAGEMENT

In Business Central, projects are identified by the Group ID, Artifact ID, and Version (GAV) Maven naming convention. GAV values differentiate projects and project versions as well as identify dependencies with particular projects.

By default, Business Central detects duplicate GAVs. This feature can be disabled by users with the admin role.

16.1. CONFIGURING GAV CHECKS AND CHILD GAV EDITION

This procedure describes how to configure GAV checks in Business Central.

Procedure

1. In Business Central, go to Menu → Design → Projects and click the project name.

2. In the project window, click the Settings tab.

3. In the General Settings tab, perform any of the following tasks:
   - To enable other projects to have the same GAV, select the Disable GAV conflict check check box.
   - To enable child projects to have GAV edition, select the Allow child GAV edition check box.

4. Click Save.

   NOTE

   You can click Reset to undo all changes.

5. Click Save to confirm the changes.

   NOTE

   Duplicate GAV detection is disabled for projects in Development Mode. To enable duplicate GAV detection in Business Central, go to project Settings → General Settings → Version and toggle the Development Mode option to OFF (if applicable).

16.2. CONFIGURING GAV CHECKS FOR ALL PROJECTS

This procedure describes how to configure GAV checks for all projects in Business Central. You can also disable GAV checks at system startup.

Procedures

1. In Business Central, select the Admin icon in the top-right corner of the screen and select Projects. The Projects window opens.

2. In the Advanced GAV preferences tab, perform any of the following tasks:
   - To enable other projects to have the same GAV, select the Disable GAV conflict check check box.

   NOTE

   You can click Reset to undo all changes.
check box.

- To enable child projects to have GAV edition, select the **Allow child GAV edition** check box.

3. Click **Save**.

**NOTE**

You can also disable the duplicate GAV detection feature by setting the `org.guvnor.project.gav.check.disabled` system property to `true` for Business Central at system startup:

```
$ ~/EAP_HOME/bin/standalone.sh -c standalone-full.xml
   -Dorg.guvnor.project.gav.check.disabled=true
```
CHAPTER 17. CONFIGURING THE ENVIRONMENT MODE IN DECISION SERVER AND BUSINESS CENTRAL

You can set Decision Server to run in **production** mode or in **development** mode. Development mode provides a flexible deployment policy that enables you to update existing deployment units (KIE containers) while maintaining active process instances for small changes. It also enables you to reset the deployment unit state before updating active process instances for larger changes. Production mode is optimal for production environments, where each deployment creates a new deployment unit.

In a development environment, you can click **Deploy** in Business Central to deploy the built KJAR file to a Decision Server without stopping any running instances (if applicable), or click **Redeploy** to deploy the built KJAR file and replace all instances. The next time you deploy or redeploy the built KJAR, the previous deployment unit (KIE container) is automatically updated in the same target Decision Server.

In a production environment, the **Redeploy** option in Business Central is disabled and you can click only **Deploy** to deploy the built KJAR file to a new deployment unit (KIE container) on a Decision Server.

Procedure

1. To configure the Decision Server environment mode, set the `org.kie.server.mode` system property to `org.kie.server.mode=development` or `org.kie.server.mode=production`.

2. To configure the deployment behavior for a project in Business Central, go to project **Settings** → **General Settings** → **Version** and toggle the **Development Mode** option.

   **NOTE**

   By default, Decision Server and all new projects in Business Central are in development mode.

   You cannot deploy a project with **Development Mode** turned on or with a manually added **SNAPSHOT** version suffix to a Decision Server that is in production mode.
CHAPTER 18. GIT HOOKS AND REMOTE GIT REPOSITORY INTEGRATION

Git hooks are bash scripts that execute before or after Git events such as `git commit` or `git push`. In Business Central, you can use Git hooks to configure repositories to trigger specified actions every time events happen. For more information about Git hooks, see Customizing Git Hooks.

You can integrate remote Git repositories with Business Central by using post-commit Git hooks. This enables you to automate content replication between Business Central and remote repositories. For example, you can implement a real-time backup strategy where changes you make to your Business Central projects are replicated to your remote Git repositories.

NOTE

Business Central only supports post-commit Git hooks.

A post-commit Git hook executes after every commit as a sync operation. Business Central waits for the post-commit bash to complete and no other write operation occurs in the repository.

18.1. CREATING POST-COMMIT GIT HOOKS

You can create a post-commit Git hook bash script file that executes code contained in that file or execute code from a different file such as a Java program.

Procedure

1. Create a `post-commit` Git hook file:
   ```bash
   $ touch post-commit
   ```

2. Set the permissions of the `post-commit` file to `755`:
   ```bash
   $ chmod 755 post-commit
   ```

3. Add `#!/bin/bash` and any required code to the `post-commit` file, for example:
   - To push all changes to a remote repository:
     ```bash
     #!/bin/bash
git push origin +master
     ```
   - To log a message:
     ```bash
     #!/bin/bash
echo 'Hello World'
     ```
   - To execute code of another file:
     ```bash
     #!/bin/bash
java -jar _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks/git-push.jar
     ```
NOTE

To use post-commit Git hooks that execute Java code, you must use the following Java libraries:

- JGit: Used to interact with internal Business Central Git repositories.
- GitHub API for Java: Used to communicate with GitHub.

For more information about post-commit Git hook and Java code examples, see Business Central post-commit Git Hooks Integration.

18.2. IMPORTING REMOTE GIT REPOSITORIES

You can import a remote Git repository into Business Central and configure a post-commit Git hook to automatically push changes to that remote repository.

Prerequisites

- Red Hat Decision Manager is installed in a Red Hat JBoss EAP 7.2 server instance.
- Red Hat Decision Manager projects exist in an external Git repository.
- Read access credentials for the external Git repository.
- (For Windows) Cygwin is installed with the Git package added during installation and the path to the Cygwin /bin folder is added to your environment PATH variable. For example, C:\cygwin64\bin. For more information about Cygwin installation, see Installing and Updating Cygwin Packages.

Procedure

1. In Business Central, go to Menu → Projects.

2. Select or create the space that you want to import the Git projects into.

3. Click on the right side of the screen and select Import Project.

4. In the Import Project window, enter the URL of your Git repository, for example, https://github.com/USERNAME/REPOSITORY_NAME.git, and the credentials for the Git repository.

5. Click Import.
   The project is added to the Business Central Git repository and is then available in the space.
IMPORTANT

Use the HTTPS or Git protocol instead of a SCP-style SSH URL. Business Central does not support the basic SSH URL and an error appears if you use this URL.

You must have your public ssh key configured in your Git provider.

The Git repository must be a KJAR project, containing only a single KJAR that is compatible with the Red Hat Decision Manager version. The KJAR content must be in the root of the repository.

6. In a command terminal, navigate to the hooks folder located in the repository Git folder of the project. For example:

   $ cd _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks

7. Create a post-commit file that pushes changes to the remote Git repository. For example:

   #!/bin/sh
   git push origin +master

   For more information about creating post-commit Git hooks, see Section 18.1, “Creating post-commit Git hooks”.

8. Optional: To check that the configuration was successful, create a guided rule in Business Central:

   b. On the Create new Guided Rule page, enter the required information.
   c. Click Ok.
      Business Central automatically pushes all changes to the remote repository.

Additional resources

- Customizing Git - Git Hooks

18.3. CONFIGURING GIT HOOKS FOR EXISTING REMOTE GIT PROJECT REPOSITORIES

If you have an existing remote Git repository project you can create a post-commit Git hook in a remote Git repository of that existing project and integrate the remote Git repository with Business Central.

Prerequisites

- Red Hat Decision Manager is installed in a Red Hat JBoss EAP 7.2 server instance.
- Red Hat Decision Manager projects exist in an external Git repository.
- Read access credentials for the external Git repository.
- (For Windows operating system) Cygwin is installed with the Git package added during
installation and the path to the Cygwin /bin folder is added to your environment PATH variable. For example, C:\cygwin64\bin. For more information about Cygwin installation, see Installing and Updating Cygwin Packages.

Procedure

1. In a command terminal, navigate to the hooks folder located in the repository Git folder of the project. For example:

   $ cd _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks

2. Create a post-commit file that pushes changes to the remote Git repository. For example:

   #!/bin/sh
   git push origin +master

   For more information about creating post-commit Git hooks, see Section 18.1, "Creating post-commit Git hooks".

3. Optional: To check that the configuration was successful, create a guided rule in Business Central:
   b. On the Create new Guided Rule page, enter the required information.
   c. Click Ok.
   Business Central automatically pushes all changes to the remote repository.

18.4. CONFIGURING GIT HOOKS AS A SYSTEM PROPERTY FOR BUSINESS CENTRAL

If you do not have an existing Git repository project or if you want to apply post-commit Git hooks to a large number of project repositories you can specify a directory containing a hook file for the value of the org.uberfire.nio.git.hooks system property. This directory is copied to the Git repositories.

NOTE

If you specify the org.uberfire.nio.git.hooks system property, all Business Central internal repositories and project repositories use the post-commit Git hook. You should only use fully qualified paths in your script.

Prerequisites

- Red Hat Decision Manager is installed in a Red Hat JBoss EAP 7.2 server instance.
- (For Windows operating system) Cygwin is installed with the Git package added during installation and the path to the Cygwin /bin folder is added to your environment PATH variable. For example, C:\cygwin64\bin. For more information about Cygwin installation, see Installing and Updating Cygwin Packages.

Procedure

1. Create a post-commit Git hook in a directory on your local system.
For more information about creating post-commit Git hooks, see Section 18.1, "Creating post-commit Git hooks".

2. To specify the directory with the hook file for the value of the `org.uberfire.nio.git.hooks` system property, do one of the following tasks:

   - Add the `org.uberfire.nio.git.hooks` system property to the `standalone.xml` file. For example:

     ```xml
     <system-properties>
     <property name="org.uberfire.nio.git.hooks" value="_EAP_HOME_/hooks">
     </property>
     ...
     </system-properties>
     ```

   - Use the `-Dorg.uberfire.nio.git.hooks` environment variable when executing Business Central. For example:

     ```bash
     $ ./standalone.sh -c standalone-full.xml -Dorg.uberfire.nio.git.hooks=_EAP_HOME_/hooks
     ```

   The post-commit Git hook is copied to all Business Central internal repositories and project repositories.

Additional resources

- Customizing Git - Git Hooks

18.5. INTEGRATING REMOTE GIT REPOSITORIES

In the following example, you use a post-commit Git hook and Java code to integrate Business Central with a remote Git repository. For the Java code example, see Business Central post-commit Git Hooks Integration. The example provides the following functionality:

   - Automatic generation of the template `.gitremote` configuration file
   - Validation of the `.gitremote` configuration file for required parameters
   - Patterns defined in the ignore parameter of the `.gitremote` file are ignored by Git
   - Message and notification output to users
   - Support for GitLab and GitHub token authentication
   - Support for GitLab group and subgroup project creation
   - Support for GitHub organization repository creation

Prerequisites

- Red Hat Decision Manager is installed in a Red Hat JBoss EAP 7.2 server instance.
- Java Development Kit (JDK) 8 is installed.
Maven is installed.

Procedure

1. In a terminal window, clone the GitHub repository to your system:

   ```
   $ git clone https://github.com/kiegroup/bc-git-integration-push.git
   ```

2. Navigate to the cloned repository:

   ```
   $ cd bc-git-integration-push
   ```

3. Execute a Maven clean install:

   ```
   $ mvn clean install
   ```

4. Create a `/hooks` folder in your `EAP_HOME` directory:

   ```
   $ mkdir -p _EAP_HOME_/hooks/
   ```

5. Copy the `git-push-2.1-SNAPSHOT.jar` to the `EAP_HOME/hooks/` folder:

   ```
   $ cp bc-git-integration-push/target/git-push-2.1-SNAPSHOT.jar _EAP_HOME_/hooks/
   ```

6. Optional: To create a template `.gitremote` configuration file, run `git-push-2.1-SNAPSHOT.jar`:

   ```
   $ java -jar git-push-2.1-SNAPSHOT.jar
   ```

   **Example template `.gitremote` configuration file**

   ```
   #This is an auto generated template empty property file
   provider=GIT_HUB
   login=
   password=
   token=
   remoteGitUrl=https://api.github.com/
   useSSH=false
   ignore=.*demo.*, test.*
   githubOrg=OrgName
   gitlabGroup=Group/subgroup
   ```

7. Modify the `.gitremote` configuration file parameters.

   **Table 18.1. Example `.gitremote` parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>provider</code></td>
<td>The Git provider. Only two values are accepted: GIT_HUB and GIT_LAB. Required</td>
</tr>
<tr>
<td><code>login</code></td>
<td>The user name for the Git provider. Required</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td>A plain text password. Not required if a <strong>token</strong> is provided.</td>
</tr>
<tr>
<td>token</td>
<td>A generated token to replace the <strong>username</strong> and <strong>password</strong> based unsecured connection. Note: If this is not set a warning is displayed that you are using an unsecured connection. Not required if a <strong>password</strong> is provided. Note: GitLab only supports token authentication.</td>
</tr>
<tr>
<td>remoteGitUrl</td>
<td>A public provider URL or a locally hosted enterprise for any provider. Required. Note: The public GitHub URL should be the API URL. For example, api.github.com.</td>
</tr>
<tr>
<td>useSSH</td>
<td>Boolean to allow the SSH protocol to push changes to the remote repository. Optional. Default = false. Note: This parameter uses the local ~/.ssh/ directory to obtain the SSH configuration.</td>
</tr>
<tr>
<td>ignore</td>
<td>A comma separated regular expressions to ignore project names that match any of these expressions. Optional.</td>
</tr>
<tr>
<td>githubOrg</td>
<td>Defines the repository organization if GitHub is used as the provider. Optional.</td>
</tr>
<tr>
<td>gitlabGroup</td>
<td>Defines the repository group and subgroup if GitLab is used as the provider. Optional.</td>
</tr>
</tbody>
</table>

8. Create a **post-commit** Git hook file in **EAP_HOME/hooks**:

   $ touch post-commit

9. Set the permissions of the **post-commit** file to **755**:

   $ chmod 755 post-commit

10. Add #!/bin/bash and code to execute **git-push-2.1-SNAPSHOT.jar** to the **post-commit** file:

    $ echo "#!/bin/bash\n java -jar $APP_SERVER_HOME/hooks/git-push-2.1-SNAPSHOT.jar" > hooks/post-commit

11. Start Business Central with the **-Dorg.uberfire.nio.git.hooks** environment variable set. For example:

    $ ./standalone.sh -c standalone-full.xml -Dorg.uberfire.nio.git.hooks=_EAP_HOME_/hooks
NOTE

To use post-commit Git hooks that execute Java code, you must use the following Java libraries:

- **JGit**: Used to interact with internal Business Central Git repositories.
- **GitHub API for Java**: Used to communicate with GitHub.

For more information about post-commit Git hook and Java code examples, see Business Central post-commit Git Hooks Integration.

18.6. GIT HOOK EXIT CODES

When a Git hook exits an integer value is returned which determines the status of the Git hook execution. This integer value is known as a Git hook exit code. The execution status can be a success (1), warning (2 to 30) or error (31 to 255).

18.7. CUSTOMIZING GIT HOOK NOTIFICATIONS

Business Central provides a mechanism that enables users to receive customized Git hook notifications based on the hook exit codes.

To enable the notification mechanism you must create a *.properties file containing the custom messages and then specify the path to that file as the value of the appformer.git.hooks.bundle system property.

Procedure

1. Create the *.properties file and add a line for each exit code with a corresponding message in the following format:

   `<exit_code>=<display_message>`

   The `<exit_code>` is the Git hook exit code and the `<display_message>` is the custom message that is displayed to a user.

   For example:

   ```
   0=Success! All working as expected.
   1=Warning! Please check the logs and advise your admin.
   .
   .
   31=Error! Please advise your admin immediately.
   ```

NOTE

It is not necessary to define all the possible exit codes in the *.properties file. Notifications appear only for the exit codes defined in the *.properties file.
The notification service only supports the ISO 8859-1 (LATIN 1) character set in the properties file. If you want to use extended characters, please use their escaped Unicode character code sequences.

2. To enable Git hook notifications, specify the path to the file as the value of the appformer.git.hooks.bundle system property. See the following example of a standalone.xml file with the setting that points to a Messages.properties file:

```xml
<system-properties>
  <property name="appformer.git.hooks.bundle" value="/opt/jboss-as/git-hooks-messages/Messages.properties">
  </property>
  ...
</system-properties>
```

18.7.1. Git hook notifications in Business Central

You can view Git hook notifications in Business Central. There are three Git hook exit code notification types.

Table 18.2. Git hook UI notification types

<table>
<thead>
<tr>
<th>Exit code</th>
<th>Customized message</th>
<th>UI notification color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success! All working as expected.</td>
<td>Green</td>
</tr>
<tr>
<td>1 to 30</td>
<td>Warning! Please check the logs and advise your admin.</td>
<td>Orange</td>
</tr>
<tr>
<td>31 to 255</td>
<td>Error! Please advise your admin immediately.</td>
<td>Red</td>
</tr>
</tbody>
</table>

**IMPORTANT**

UNIX machines only support error codes between 0 (success) to 255 (error), any exit code outside of this range will end up being converted into a different code which may cause showing a wrong notification message.

Windows machines don’t have this limitation and support a wide range of exit codes.

18.7.2. Git hook notification internationalization support

You can internationalize notification messages by placing additional properties files in the same path as the original properties file specified as the appformer.git.hooks.bundle system property.

The name of the different localized files must be `<filename>_lang.properties`, where the `<filename>` is the same as the original. For example, where the system property points to Messages.properties, you can create Messages_en.properties for English, Messages_fr.properties for French, or Messages_it.properties for Italian.
The notification service will choose the properties file based on the user’s language, if there are no available translations for that language it will use the entries from the original `Messages.properties` file.
CHAPTER 19. ROLE-BASED ACCESS CONTROL FOR BRANCHES IN BUSINESS CENTRAL

Business Central provides the option for users to restrict the access for a target branch for a specific collaborator type. The security check uses both the Security Management screen and contributors sources to grant or deny permissions to spaces and projects. For example, if a user has the security permission to update a project and has write permission on that branch, based on the contributor type, then they are able to create new assets.

19.1. CUSTOMIZING ROLE-BASED BRANCH ACCESS

You can customize contributor role permissions for each branch of a project in Business Central. For example, you can set Read, Write, Delete, and Deploy access for each role assigned to a branch.

Procedure

1. In Business Central, go to Menu → Design → Projects.

2. If needed, add a new contributor:
   a. Click the project name and then click the Contributors tab.
   b. Click Add Contributor.
   c. Enter user name in the text field.
   d. Select the Contributor role type from the drop-down list.
   e. Click Ok.

3. Customize role-based branch access for the relevant contributor:
   a. Click Settings → Branch Management.
   b. Select the branch name from the drop-down list.
   c. In the Role Access section, select or deselect the permissions check boxes to specify role-based branch access for each available role type.
   d. Click Save and click Save again to confirm your changes.
CHAPTER 20. VIEWING PROCESS INSTANCE LOGS

You can view all the process events of an instance from its Logs tab. The instance logs list all the current and previous process states. Business Central has two types of logs for process instances, Business and Technical logs.

Procedure

1. In Business Central, go to Menu → Manage → Process Instances.

2. On the Manage Process Instances page, click the process instance whose log you want to view.

3. Select the Logs tab:
   - Click Business to view the business events log.
   - Click Technical to view the technical events log.
   - Click Asc or Desc to change the order of the log files.
CHAPTER 21. BUSINESS CENTRAL SYSTEM PROPERTIES

The Business Central system properties listed in this section are passed to standalone*.xml files. To install standalone Business Central, you can use the listed properties in the following command:

```
java -jar rhdm-7.6.0-decision-central-standalone.jar -s application-config.yaml -D<property>=<value> -D<property>=<value>
```

In this command, `<property>` is a property from list and `<value>` is a value that you assign to that property.

- `org.uberfire.nio.git.dir`: Location of the Decision Server Git directory.
- `org.uberfire.nio.git.proxy.ssh.over.http`: Specifies whether SSH should use an HTTP proxy. Default value: `false`.
- `http.proxyHost`: Defines the host name of the HTTP proxy. Default value: `null`.
- `http.proxyPort`: Defines the host port (integer value) of the HTTP proxy. Default value: `null`.
- `org.uberfire.nio.git.proxy.ssh.over.https`: Specifies whether SSH should use an HTTPS proxy. Default value: `false`.
- `https.proxyHost`: Defines the host name of the HTTPS proxy. Default value: `null`.
- `https.proxyPort`: Defines the host port (integer value) of the HTTPS proxy. Default value: `null`.
- `org.uberfire.nio.git.http.enabled`: Enables or disables the HTTP daemon. Default value: `true`.
- `org.uberfire.nio.git.http.host`: If the HTTP daemon is enabled, it uses this property as the host identifier. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default value: `localhost`.
- `org.uberfire.nio.git.http.hostname`: If the HTTP daemon is enabled, it uses this property as the host name identifier. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default value: `localhost`.
- `org.uberfire.nio.git.http.port`: If the HTTP daemon is enabled, it uses this property as the port number. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default value: `8080`.
- `org.uberfire.nio.git.https.enabled`: Enables or disables the HTTPS daemon. Default value: `false`.
- `org.uberfire.nio.git.https.host`: If the HTTPS daemon is enabled, it uses this property as the host identifier. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default value: `localhost`.
- `org.uberfire.nio.git.https.hostname`: If the HTTPS daemon is enabled, it uses this property as the host name identifier. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default value: `localhost`.
• **org.uberfire.nio.git.https.port**: If the HTTPS daemon is enabled, it uses this property as the port number. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default value: **8080**.

• **org.uberfire.nio.git.daemon.enabled**: Enables or disables the Git daemon. Default value: **true**.

• **org.uberfire.nio.git.daemon.host**: If the Git daemon is enabled, it uses this property as the local host identifier. Default value: **localhost**.

• **org.uberfire.nio.git.daemon.hostname**: If the Git daemon is enabled, it uses this property as the local host name identifier. Default value: **localhost**.

• **org.uberfire.nio.git.daemon.port**: If the Git daemon is enabled, it uses this property as the port number. Default value: **9418**.

• **org.uberfire.nio.git.http.sslVerify**: Enables or disables SSL certificate checking for Git repositories. Default value: **true**.

**NOTE**
If the default or assigned port is already in use, a new port is automatically selected. Ensure that the ports are available and check the log for more information.

• **org.uberfire.nio.git.ssh.enabled**: Enables or disables the SSH daemon. Default value: **true**.

• **org.uberfire.nio.git.ssh.host**: If the SSH daemon enabled, it uses this property as the local host identifier. Default value: **localhost**.

• **org.uberfire.nio.git.ssh.hostname**: If the SSH daemon is enabled, it uses this property as local host name identifier. Default value: **localhost**.

• **org.uberfire.nio.git.ssh.port**: If the SSH daemon is enabled, it uses this property as the port number. Default value: **8001**.

**NOTE**
If the default or assigned port is already in use, a new port is automatically selected. Ensure that the ports are available and check the log for more information.

• **org.uberfire.nio.git.ssh.cert.dir**: Location of the `.security` directory where local certificates are stored. Default value: Working directory.

• **org.uberfire.nio.git.ssh.passphrase**: Pass phrase used to access the public key store of your operating system when cloning git repositories with SCP style URLs. Example: `git@github.com:user/repository.git`.

• **org.uberfire.nio.git.ssh.algorithm**: Algorithm used by SSH. Default value: **RSA**.

• **org.uberfire.nio.git.ssh.ciphers**: A comma-separated string of ciphers. The available ciphers are `aes128-ctr, aes192-ctr, aes256-ctr, arcfour128, arcfour256, aes192-cbc, aes256-cbc`. If the property is not used, all available ciphers are loaded.
• **org.uberfire.nio.git.ssh.macs**: A comma-separated string of message authentication codes (MACs). The available MACs are **hmac-md5**, **hmac-md5-96**, **hmac-sha1**, **hmac-sha1-96**, **hmac-sha2-256**, **hmac-sha2-512**. If the property is not used, all available MACs are loaded.

**NOTE**
If you plan to use RSA or any algorithm other than DSA, make sure you set up your application server to use the Bouncy Castle JCE library.

• **org.uberfire.metadata.index.dir**: Place where the Lucene `.index` directory is stored. Default value: Working directory.

• **org.uberfire.ldap.regex.role_mapper**: Regex pattern used to map LDAP principal names to the application role name. Note that the variable role must be part of the pattern because it is substituted by the application role name when matching a principal value to a role name.

• **org.uberfire.sys.repo.monitor.disabled**: Disables the configuration monitor. Do not disable unless you are sure. Default value: `false`.

• **org.uberfire.secure.key**: Password used by password encryption. Default value: `org.uberfire.admin`.

• **org.uberfire.secure.alg**: Crypto algorithm used by password encryption. Default value: `PBEWithMD5AndDES`.

• **org.uberfire.domain**: Security-domain name used by uberfire. Default value: `ApplicationRealm`.

• **org.guvnor.m2repo.dir**: Place where the Maven repository folder is stored. Default value: `<working-directory>/repositories/kie`.

• **org.guvnor.project.gav.check.disabled**: Disables group ID, artifact ID, and version (GAV) checks. Default value: `false`.

• **org.kie.build.disable-project-explorer**: Disables automatic build of a selected project in Project Explorer. Default value: `false`.

• **org.kie.verification.disable-dtable-realtime-verification**: Disables the real-time validation and verification of decision tables. Default value: `false`.

• **org.kie.server.controller**: The URL is used to connect to the Decision Manager controller. For example, `ws://localhost:8080/decision-central/websocket/controller`.

• **org.kie.server.user**: User name used to connect to the Decision Server nodes from the Decision Manager controller. This property is only required when using this Business Central installation as a Decision Manager controller.

• **org.kie.server.pwd**: Password used to connect to the Decision Server nodes from the Decision Manager controller. This property is only required when using this Business Central installation as a Decision Manager controller.

• **kie.maven.offline.force**: Forces Maven to behave as if offline. If true, disables online dependency resolution. Default value: `false`.
NOTE

Use this property for Business Central only. If you share a runtime environment with any other component, isolate the configuration and apply it only to Business Central.

- **org.uberfire.gzip.enable**: Enables or disables Gzip compression on the GzipFilter compression filter. Default value: `true`.

- **org.kie.workbench.profile**: Selects the Business Central profile. Possible values are FULL or PLANNER_AND_RULES. A prefix FULL_ sets the profile and hides the profile preferences from the administrator preferences. Default value: FULL.

- **org.appformer.m2repo.url**: Business Central uses the default location of the Maven repository when looking for dependencies. It directs to the Maven repository inside Business Central, for example, `http://localhost:8080/business-central/maven2`. Set this property before starting Business Central. Default value: File path to the inner m2 repository.

- **appformer.ssh.keystore**: Defines the custom SSH keystore to be used with Business Central by specifying a class name. If the property is not available, the default SSH keystore is used.

- **appformer.ssh.keys.storage.folder**: When using the default SSH keystore, this property defines the storage folder for the user’s SSH public keys. If the property is not available, the keys are stored in the Business Central .security folder.

- **appformer.experimental.features**: Enables the experimental features framework. Default value: `false`.

- **org.kie.demo**: Enables an external clone of a demo application from GitHub.

- **org.kie.workbench.controller**: The URL used to connect to the Decision Manager controller, for example, `ws://localhost:8080/kie-server-controller/websocket/controller`.

- **org.kie.workbench.controller.user**: The Decision Manager controller user. Default value: kieserver.

- **org.kie.workbench.controller.pwd**: The Decision Manager controller password. Default value: kieserver1!.

- **org.kie.workbench.controller.token**: The token string used to connect to the Decision Manager controller.

- **kie.keystore.keyStoreURL**: The URL used to load a Java Cryptography Extension KeyStore (JCEKS). For example, `file:///home/kie/keystores/keystore.jceks`.

- **kie.keystore.keyStorePwd**: The password used for the JCEKS.

- **kie.keystore.key.ctrl.alias**: The alias of the key for the default REST Decision Manager controller.

- **kie.keystore.key.ctrl.pwd**: The password of the alias for the default REST Decision Manager controller.

- **org.jbpm.wb.forms.renderer.ext**: Switches the form rendering between Business Central and Decision Server. By default, the form rendering is performed by Business Central. Default value: `false`. 
- **org.jbpm.wb.forms.renderer.name**: Enables you to switch between Business Central and Decision Server rendered forms. Default value: `workbench`. 
CHAPTER 22. PERFORMANCE TUNING CONSIDERATIONS WITH BUSINESS CENTRAL

The following key concepts or suggested practices can help you optimize Business Central configuration and Red Hat Decision Manager performance. These concepts are summarized in this section as a convenience and are explained in more detail in the cross-referenced documentation, where applicable. This section will expand or change as needed with new releases of Red Hat Decision Manager.

Ensure that development mode is enabled during development

You can set Decision Server or specific projects in Business Central to use production mode or development mode. By default, Decision Server and all new projects in Business Central are in development mode. This mode provides features that facilitate your development experience, such as flexible project deployment policies, and features that optimize Decision Server performance during development, such as disabled duplicate GAV detection. Use development mode until your Red Hat Decision Manager environment is established and completely ready for production mode. For more information about configuring the environment mode or duplicate GAV detection, see the following resources:

- Chapter 17, Configuring the environment mode in Decision Server and Business Central
- Packaging and deploying a Red Hat Decision Manager project

Disable verification and validation of complex guided decision tables

The decision table verification and validation feature of Business Central is enabled by default. This feature helps you validate your guided decision tables, but with complex guided decision tables, this feature can hinder decision engine performance. You can disable this feature by setting the org.kie.verification.disable-dtable-realtime-verification system property value to true.

For more information about guided decision table validation, see Designing a decision service using guided decision tables.

Disable automatic builds if you have many large projects

In Business Central, when you navigate between projects in the Project Explorer side panel, the selected project is built automatically so that the Alerts window is updated to show any build errors for the project. If you have large projects or frequently switch between many projects that are under active development, this feature can hinder Business Central and decision engine performance.

To disable automatic project builds, set the org.kie.build.disable-project-explorer system property to true.
APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Tuesday, June 02, 2020.