Red Hat Single Sign-On 7.1
Upgrading Guide

For Use with Red Hat Single Sign-On 7.1

Red Hat Customer Content Services

Red Hat Single Sign-On 7.1 Upgrading Guide

For Use with Red Hat Single Sign-On 7.1

Legal Notice

Abstract

This book is a guide to upgrading your application from a previous version of Red Hat Single Sign-On 7.1.

Table of Contents

CHAPTER 1. INTRODUCTION

Red Hat Single Sign-On (RH-SSO) 7.1 is based on the Keycloak project and provides security for your web applications by providing Web single sign-on capabilities based on popular standards such as SAML 2.0, OpenID Connect, and OAuth 2.0. The Red Hat Single Sign-On Server can act as a SAML or OpenID Connect–based identity provider, mediating with your enterprise user directory or third-party SSO provider for identity information and your applications using standards-based tokens.

RH-SSO provides two operating modes: standalone server or managed domain. The standalone server operating mode represents running RH-SSO as a single server instance. The managed domain operating mode allows for the management of multiple RH-SSO instances from a single control point. The upgrade process differs depending on which operating mode has been implemented. Specific instructions for each mode are provided where applicable.

The purpose of this guide is to document the steps that are required to successfully upgrade from Red Hat Single Sign-On 7.0 to Red Hat Single Sign-On 7.1.

1.1. ABOUT UPGRADES

1.1.1. Major Upgrades

A major upgrade or migration is required when RH-SSO is upgraded from one major release to another, for example, from Red Hat Single Sign-On 7.0 to Red Hat Single Sign-On 8.0. There may be breaking API changes between major releases that could require rewriting parts of applications or server extensions.

1.1.2. Minor Updates

Red Hat Single Sign-On periodically provides point releases, which are minor updates that include bug fixes, security fixes, and new features. If you plan to upgrade from one Red Hat Single Sign-On point release to another, for example, from Red Hat Single Sign-On 7.0 to Red Hat Single Sign-On 7.1, code changes should not be required for applications or custom server extensions as long as no private, unsupported, or tech preview APIs are used.

1.1.3. Micro Updates

Red Hat Single Sign-On 7 also periodically provides micro releases that contain bug and security fixes. Micro releases increment the minor release version by the last digit, for example from 7.1.0 to 7.1.1. These release do not require migration and should not impact the server configuration files. The patch management system for ZIP installations can also rollback the patch and server configuration.

A micro release only contains the artifacts that have changed. For example if Red Hat Single Sign-On 7.1.1 contains changes to the server and the JavaScript adapter, but not the EAP adapter, only the server and JavaScript adapter are released and require updating.

CHAPTER 2. ABOUT CHANGES FROM RH-SSO 7.0 TO RH-SSO 7.1

The following changes have occurred from RH-SSO 7.0 to RH-SSO 7.1. Review these changes carefully before upgrading.

2.1. REALM KEYS

For RH-SSO 7.0 only one set of keys could be associated with a realm. This meant that when changing the keys all current cookies and tokens would be invalidated and all users would have to re-authenticate. For RH-SSO 7.1 support for multiple keys for one realm has been added. At any given time one set of keys is the active keys used for signatures, but there can be multiple keys used to verify signatures. This means that old cookies and tokens can be verified, then refreshed with the new signatures, allowing users to remain authenticated when keys are changed. There are also some changes to how keys are managed through the Admin Console and Admin REST API; for more details see Realm Keys in the *Red Hat Single Sign-On Server Administration Guide*.

To allow seamless key rotation you must remove hard-coded keys from client adapters. The client adapters will automatically retrieve keys from the server as long as the realm key is not specified. Client adapters will also retrieve new keys automatically when keys are rotated.

2.2. CLIENT REDIRECT URI MATCHING

For RH-SSO 7.0 query parameters are ignored when matching valid redirect URIs for a client. For RH-SSO 7.1 query parameters are no longer ignored. If you need to include query parameters in the redirect URI you must specify the query parameters in the valid redirect URI for the client (for example, https://hostname/app/login?foo=bar) or use a wildcard (for example, https://hostname/app/login/*). Fragments are also no longer permitted in Valid Redirect URIs (that is, https://hostname/app#fragment).

2.3. AUTOMATICALLY REDIRECT TO IDENTITY PROVIDER

For RH-SSO 7.1, identity providers cannot be set as the default authentication provider. To automatically redirect to an identity provider for RH-SSO 7.1 you must now configure the identity provider redirector. For more information see Default Identity Provider in the *Red Hat Single Sign-On Server Administration Guide*. If you previously had an identity provider with the default authentication provider option set, this value is automatically used as the value for the identity provider redirector when the server is upgraded to RH-SSO 7.1.

2.4. ADMIN REST API

For RH-SSO 7.0 paginated endpoints in the Admin REST API return all results if the maxResults query parameter was not specified. This could cause issues with a temporary high load and requests timing out when a large number of results were returned (for example, users). For RH-SSO 7.1 a maximum of 100 results are returned if a value for maxResults is not specified. You can return all results by specifying maxResults as -1.

2.5. SERVER CONFIGURATION

For RH-SSO 7.0 server configuration is split between the keycloak-server.json file and the standalone/domain.xml or domain.xml file. For RH-SSO 7.1 the keycloak-server.json file has been removed and all server configuration is done through the standalone.xml or domain.xml file. The upgrading procedure for RH-SSO 7.1 automatically migrates the server configuration from the keycloak-server.json file to the standalone.xml or domain.xml file.

2.6. KEY ENCRYPTION ALGORITHM IN SAML ASSERTIONS

For RH-SSO 7.1, keys in SAML assertions and documents are now encrypted using the RSA-OAEP encryption scheme. To use encrypted assertions, ensure your service providers support this encryption scheme. In the event that you have service providers that do not support RSA-OAEP, RH-SSO can be configured to use the legacy RSA-v1.5 encryption scheme by starting the server with the system property “keycloak.saml.key_trans.rsa_v1.5” set to true. If you do this you should upgrade your service providers as soon as possible to be able to revert to the more secure RSA-OAEP encryption scheme.

CHAPTER 3. UPGRADING RED HAT SINGLE SIGN-ON SERVER

The upgrade process for the Red Hat Single Sign-On Server is different if you are upgrading to a different minor release or not.

If you are upgrading to a new minor release, for example from 7.0.0 to 7.1.0, follow the steps in Minor Upgrades.

If you are upgrading to a new micro release, for example from 7.1.0 to 7.1.1, follow the steps in Micro Upgrades.

3.1. MINOR UPGRADES

3.1.1. Preparing for Upgrading

Before you upgrade RH-SSO 7.0 to 7.1, be aware of the order in which you need to perform the upgrade steps. Also note potential issues that can occur within the upgrade process. In general, you must upgrade RH-SSO Server first, and then upgrade the adapters.

1. Prior to applying the upgrade, handle any open transactions and delete the data/tx-object-store/ transaction directory.
2. Back up RH-SSO 7.0 (configuration, themes, and so on).
3. Back up the database. For detailed information on how to back up the database, see the documentation for the relational database you’re using.
4. Upgrade RH-SSO Server.
    - Testing the upgrade in a non-production environment first, to prevent any installation issues from being exposed in production, is a best practice.
    - Be aware that after the upgrade the database will no longer be compatible with RH-SSO 7.0.
    - Ensure the upgraded server is functional before upgrading adapters in production.
5. If you need to revert the upgrade, first restore the RH-SSO 7.0 installation, and then restore the database from the backup copy.
6. Upgrade the adapters.

3.1.2. Upgrading RH-SSO Server

It is important that you upgrade RH-SSO Server before upgrading the adapters.

Note

If you are using RH-SSO 7.0 SAML adapters with the isPassive option set to "true" in combination with RH-SSO 7.1 Server, there is a known issue that prevents correct handling of the SAML response. When the user is not authenticated and tries to access a protected page it results in an exception rather than a redirect to authenticate. To prevent this issue from occurring, you must upgrade both RH-SSO Server and the SAML adapter at the same time.

To upgrade RH-SSO Server from 7.0 to 7.1, complete the following steps:

1. Prior to applying the upgrade, handle any open transactions and delete the data/tx-object-store/ transaction directory.
2. Download the ZIP or TAR archive: rh-sso-7.1.[zip|tar.gz]
3. Move the downloaded ZIP or TAR archive to the desired location.
4. Run the **`unzip`** or **`gunzip`** and **`tar`** utilities. This step installs a clean instance of the latest RH-SSO release.
5. For standalone installations, copy the RHSSO_HOME/standalone/ directory from the previous installation over the RH-SSO 7.1 installation directories.

   For domain installations, copy the RHSSO_HOME/domain/ directory from the previous installation over the RH-SSO 7.1 installation directories.
6. For standalone installations, create the empty directory RHSSO_HOME/standalone/deployments.

   For domain installations, create the empty directory RHSSO_HOME/domain/deployments.
7. Review the changes made to the bin directory of the RH-SSO 7.0 installation, and make the equivalent modifications to the RH-SSO 7.1 directory.

   NOTE: Files in the bin directory should not be overwritten by the files from previous versions. Changes should be made manually.
8. Copy any custom modules that have been added to the modules directory.
9. Run the applicable upgrade script below.

3.1.2.1. Running the Standalone Mode Upgrade Script

To run the upgrade script for standalone mode, complete the following steps:

1. If you are using a different configuration file than the default one, edit the migration script to specify the new file name.
2. Stop the server.
3. Run the upgrade script:

   ```
   bin/jboss-cli.sh --file=bin/migrate-standalone.cli
   ```

3.1.2.2. Running the Standalone-High Availability Mode Upgrade Script

For standalone-high availability (HA) mode, all instances must be upgraded at the same time.

To run the upgrade script for standalone-HA mode, complete the following steps:

1. If you are using a different configuration file than the default one, edit the migration script to specify the new file name.
2. Stop the server.
3. Run the upgrade script:

   ```
   bin/jboss-cli.sh --file=bin/migrate-standalone-ha.cli
   ```

3.1.2.3. Running the Domain Mode Upgrade Script

For domain mode, all instances must be upgraded at the same time.

To run the upgrade script for domain mode, complete the following steps:

1. If you have changed the profile name, you must edit the upgrade script to change a variable near the beginning of the script.
2. Edit the domain script to include the location of the keycloak-server.json file.
3. Stop the server.
4. Run the upgrade script on the domain controller only:

   ```
   bin/jboss-cli.sh --file=bin/migrate-domain.cli
   ```

3.1.2.4. Running the Domain-clustered Mode Upgrade Script

For domain-clustered mode, all instances must be upgraded at the same time.

To run the upgrade script for domain-clustered mode, complete the following steps:

1. If you have changed the profile name, you must edit the upgrade script to change a variable near the beginning of the script.
2. Edit the domain-clustered script to include the location of the keycloak-server.json file.
3. Stop the server.
4. Run the upgrade script on the domain controller only:

   ```
   bin/jboss-cli.sh --file=bin/migrate-domain-clustered.cli
   ```

3.1.3. Migrating the Database

Red Hat Single Sign-On can automatically migrate the database schema, or you can choose to do it manually. By default the database is automatically migrated when you start RH-SSO 7.1 for the first time.

3.1.3.1. Automatic Relational Database Migration

To enable automatic upgrading of the database schema, set the migrationStrategy property value to "update" for the default connectionsJpa provider:

```
 <spi name="connectionsJpa">
    <provider name="default" enabled="true">
        <properties>
            ...
            <property name="migrationStrategy" value="update"/>
        </properties>
    </provider>
</spi>
```

Or run this CLI command:

/subsystem=keycloak-server/spi=connectionsJpa/provider=default/:map-put(name=properties,key=migrationStrategy,value=update)

When you start the server with this setting your database is automatically migrated if the database schema has changed in the new version.

3.1.3.2. Manual Relational Database Migration

To enable manual upgrading of the database schema, set the migrationStrategy property value to "manual" for the default connectionsJpa provider:

```
 <spi name="connectionsJpa">
    <provider name="default" enabled="true">
        <properties>
            ...
            <property name="migrationStrategy" value="manual"/>
        </properties>
    </provider>
</spi>
```

Or run this CLI command:

/subsystem=keycloak-server/spi=connectionsJpa/provider=default/:map-put(name=properties,key=migrationStrategy,value=manual)

When you start the server with this configuration it checks if the database needs to be migrated. The required changes are written to an SQL file that you can review and manually run against the database. For further details on how to apply this file to the database, see the documentation for the relational database you’re using. After the changes have been written to the file, the server exits.

3.1.4. Migrating Themes

If you have created any custom themes they must be migrated to the RH-SSO 7.1 server. For RH-SSO 7.1 any changes to the built-in themes might need to be reflected in your custom themes, depending on which aspects you have customized.

You must copy your custom themes from the RH-SSO 7.0 “themes” directory to the “themes” directory in RH-SSO 7.1. After that you need to review the changes below and consider if the changes need to be applied to your custom theme. In summary:

- If you have customized any of the changed templates for RH-SSO 7.1 listed below you need to compare the template from the base theme to see if there are changes you need to apply.
- If you have customized any of the styles and are extending the Red Hat Single Sign-On themes you need to review the changes to the styles. If you are extending the base theme you can skip this step.
- If you have customized messages you might need to change the key or value or to add additional messages.

Each step is described in more detail below the list of changes.

The changes that have been made for RH-SSO 7.1 include:

Templates

- Account: account.ftl
- Account: federatedIdentity.ftl
- Account: totp.ftl
- Login: info.ftl
- Login: login-config-totp.ftl
- Login: login-reset-password.ftl
- Login: login.ftl

Messages

- Account: editAccountHtmlTtile renamed to editAccountHtmlTitle
- Account: role_uma_authorization added
- Login: loginTotpStep1 value changed
- Login: invalidPasswordGenericMessage added
- Login: invlidRequesterMessage renamed to invalidRequesterMessage
- Login: clientDisabledMessage added

Styles

- Account: account.css
- Login: login.css

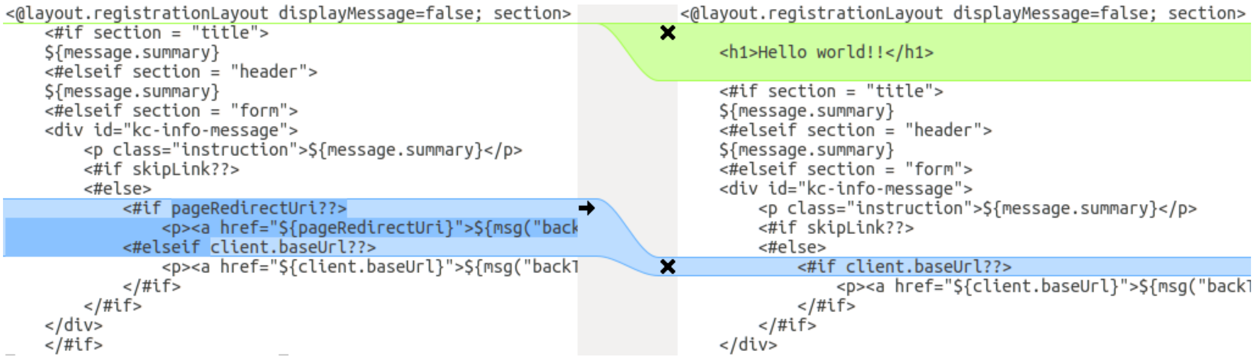
3.1.4.1. Migrating Templates

If you have customized any of the templates listed above you need to carefully review the changes that have been made to the templates for RH-SSO 7.1 to decide if you need to apply these changes to your customized templates. Most likely you will need to apply the same changes to your customized templates. If you have not customized any of the listed templates you can skip this section.

A best practice is to use a diff tool to compare the templates to see what changes you might need to make to your customized template. If you have only made minor changes it is simpler to compare the updated RH-SSO 7.1 template to your customized template. However, if you have made many changes it might be easier to compare the RH-SSO 7.1 template to your customized RH-SSO 7.0 template from the RH-SSO 7.0 installation, as this will show you what changes you need to make.

The following screenshot compares the info.ftl template from the Login theme and an example custom theme:

Comparison of the updated version of a Login theme template with an example custom Login theme template



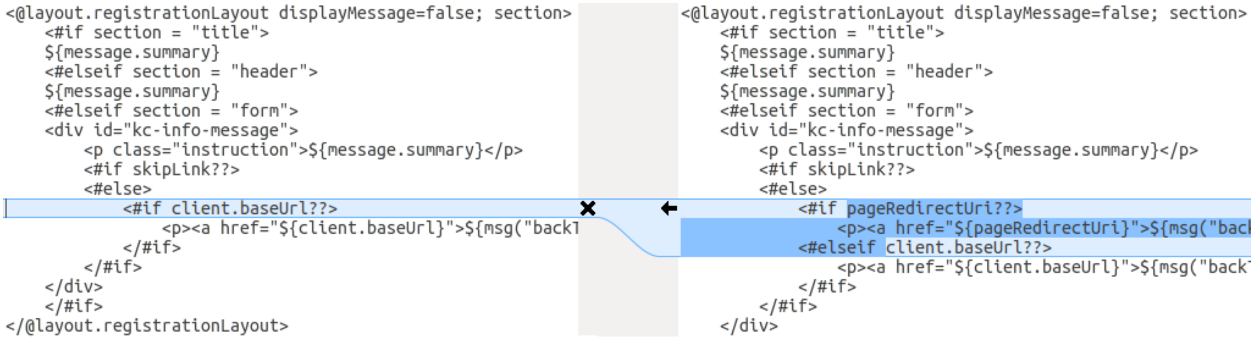
```
<@layout.registrationLayout displayMessage=false; section>
    <#if section = "title">
    ${message.summary}
    <#elseif section = "header">
    ${message.summary}
    <#elseif section = "form">
    <div id="kc-info-message">
        <p class="instruction">${message.summary}</p>
        <#if skipLink??>
        <#else>
            <#if pageRedirectUri??>
                <p><a href="${pageRedirectUri}">${msg("back
            <#elseif client.baseUrl??>
                <p><a href="${client.baseUrl}">${msg("backT
            </#if>
        </#if>
    </div>
    </#if>
```

```
<@layout.registrationLayout displayMessage=false; section>

    <h1>Hello world!!</h1>

    <#if section = "title">
    ${message.summary}
    <#elseif section = "header">
    ${message.summary}
    <#elseif section = "form">
    <div id="kc-info-message">
        <p class="instruction">${message.summary}</p>
        <#if skipLink??>
        <#else>
            <#if client.baseUrl??>
                <p><a href="${client.baseUrl}">${msg("backT
            </#if>
        </#if>
    </div>
```

From this comparison it is easy to identify that the first change (“Hello world!!”) was a customization, while the second change (“if pageRedirectUri”) is a change to the base theme. By copying the second change to your custom template, you have successfully updated your customized template.

For the alternative approach the following screenshot compares the info.ftl template from the RH-SSO 7.0 server installation with the updated info.ftl template from the RH-SSO 7.1 server installation:

Comparison of an example custom Login theme template with the updated version of the Login theme template



```
<@layout.registrationLayout displayMessage=false; section>
    <#if section = "title">
    ${message.summary}
    <#elseif section = "header">
    ${message.summary}
    <#elseif section = "form">
    <div id="kc-info-message">
        <p class="instruction">${message.summary}</p>
        <#if skipLink??>
        <#else>
            <#if client.baseUrl??>
                <p><a href="${client.baseUrl}">${msg("backT
            </#if>
        </#if>
    </div>
    </#if>
</@layout.registrationLayout>
```

```
<@layout.registrationLayout displayMessage=false; section>
    <#if section = "title">
    ${message.summary}
    <#elseif section = "header">
    ${message.summary}
    <#elseif section = "form">
    <div id="kc-info-message">
        <p class="instruction">${message.summary}</p>
        <#if skipLink??>
        <#else>
            <#if pageRedirectUri??>
                <p><a href="${pageRedirectUri}">${msg("back
            <#elseif client.baseUrl??>
                <p><a href="${client.baseUrl}">${msg("back
            </#if>
        </#if>
    </div>
```

From this comparison it is easy to identify what has been changed in the base template. You will then manually have to make the same changes to your modified template. Since this approach is not as simple as the first approach, only use this approach if the first one is not feasible.

3.1.4.2. Migrating Messages

If you have added support for another language, you need to apply all the changes listed above. If you have not added support for another language, you might not need to change anything; you only have to make changes if you have changed an affected message in your theme.

For added values, review the value of the message in the base theme to determine if you need to customize that message.

For renamed keys, rename the key in your custom theme.

For changed values, check the value in the base theme to determine if you need to make changes to your custom theme.

3.1.4.3. Migrating Styles

If you are inheriting styles from the keycloak or rh-sso themes you might need to update your custom styles to reflect changes made to the styles from the built-in themes.

A best practice is to use a diff tool to compare the changes to stylesheets between the RH-SSO 7.0 server installation and the RH-SSO 7.1 server installation.

For example, using the diff command: $ diff rh-sso-7.0/themes/keycloak/login/resources/css/login.css \

rh-sso-7.1/themes/keycloak/login/resources/css/login.css

Review the changes and determine if they affect your custom styling.

3.2. MICRO UPGRADES

3.2.1. Patching a ZIP/Installer Installation

Patches for a ZIP installation of RH-SSO are available to download from the Red Hat Customer Portal.

For multiple RH-SSO hosts in a managed domain environment, individual hosts can be patched from your RH-SSO domain controller.

In addition to applying a patch, you can also roll back the application of a patch.

3.2.1.1. Important Notes on ZIP Installation Patching

- If you apply a patch that updates a module, the new patched JARs that are used at runtime are stored in **`RHSSO_HOME/modules/system/layers/base/.overlays/`*`PATCH_ID`*`/`*`MODULE`***. The original unpatched files are left in **`RHSSO_HOME/modules/system/layers/base/`*`MODULE`***, but these JARs are **not** used at runtime.

- In order to significantly decrease the size of cumulative patch releases for RH-SSO 7 you cannot perform a partial roll back of a cumulative patch. For a patch that has been applied, you will only be able to roll back the whole patch.

  For example, if you apply CP03 to RH-SSO 7.0.0, you will not be able to roll back to CP01 or CP02. If you would like the ability to roll back to each cumulative patch release, each cumulative patch must be applied separately in the order they were released.

3.2.1.2. Applying a Patch

Note

RH-SSO servers that have been installed using the RPM method cannot be updated using these instructions. See the RPM instructions for applying a patch instead.

You can apply downloaded patches to a RH-SSO server using either the management CLI or the management console.

Applying a Patch to RH-SSO Using the Management CLI

1. Download the patch file from the Red Hat Customer Portal at https://access.redhat.com/downloads/.

2. From the management CLI, apply the patch using the following command, including the appropriate path to the patch file:

   ```
   patch apply /path/to/downloaded-patch.zip
   ```

   Note

   To patch another RH-SSO host in a managed domain, you can specify the RH-SSO host name using the **`--host=`** argument. For example:

   ```
   patch apply /path/to/downloaded-patch.zip --host=my-host
   ```

   The patch tool will warn if there are any conflicts in attempting to apply the patch. If there are conflicts, enter **`patch --help`** for the available arguments to re-run the command with an argument specifying how to resolve the conflicts.

3. Restart the RH-SSO server for the patch to take effect:

   ```
   shutdown --restart=true
   ```

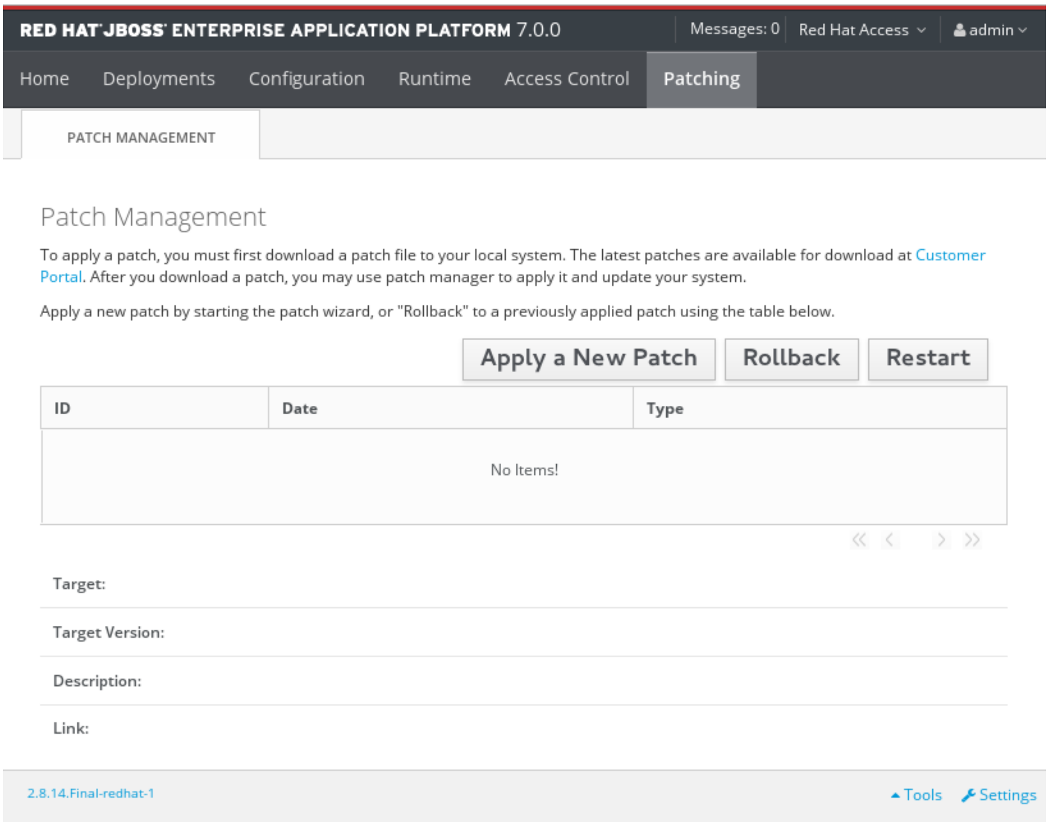
Applying a Patch to RH-SSO Using the Management Console

1. Download the patch file from the Red Hat Customer Portal at https://access.redhat.com/downloads/.

2. Open the management console and navigate to the **Patch Management** view.

   a. For a standalone server, click the **Patching** tab.

      The Patch Management Screen for a Standalone Server



Patch Management

To apply a patch, you must first download a patch file to your local system. The latest patches are available for download at Customer Portal. After you download a patch, you may use patch manager to apply it and update your system.

Apply a new patch by starting the patch wizard, or "Rollback" to a previously applied patch using the table below.

ID	Date	Type
No Items!		

Target:

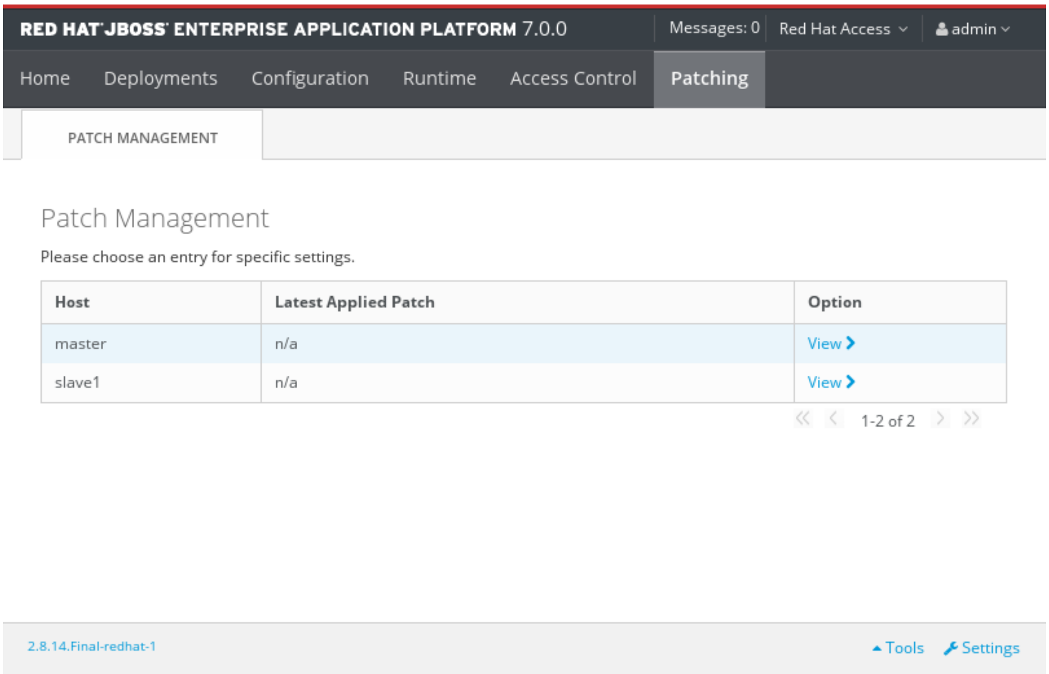
Target Version:

Description:

Link:

b. For a server in a managed domain, click the **Patching** tab, then select the host that you want to patch from the table, and click **View**.

The Patch Management Screen for a Managed Domain



Patch Management

Please choose an entry for specific settings.

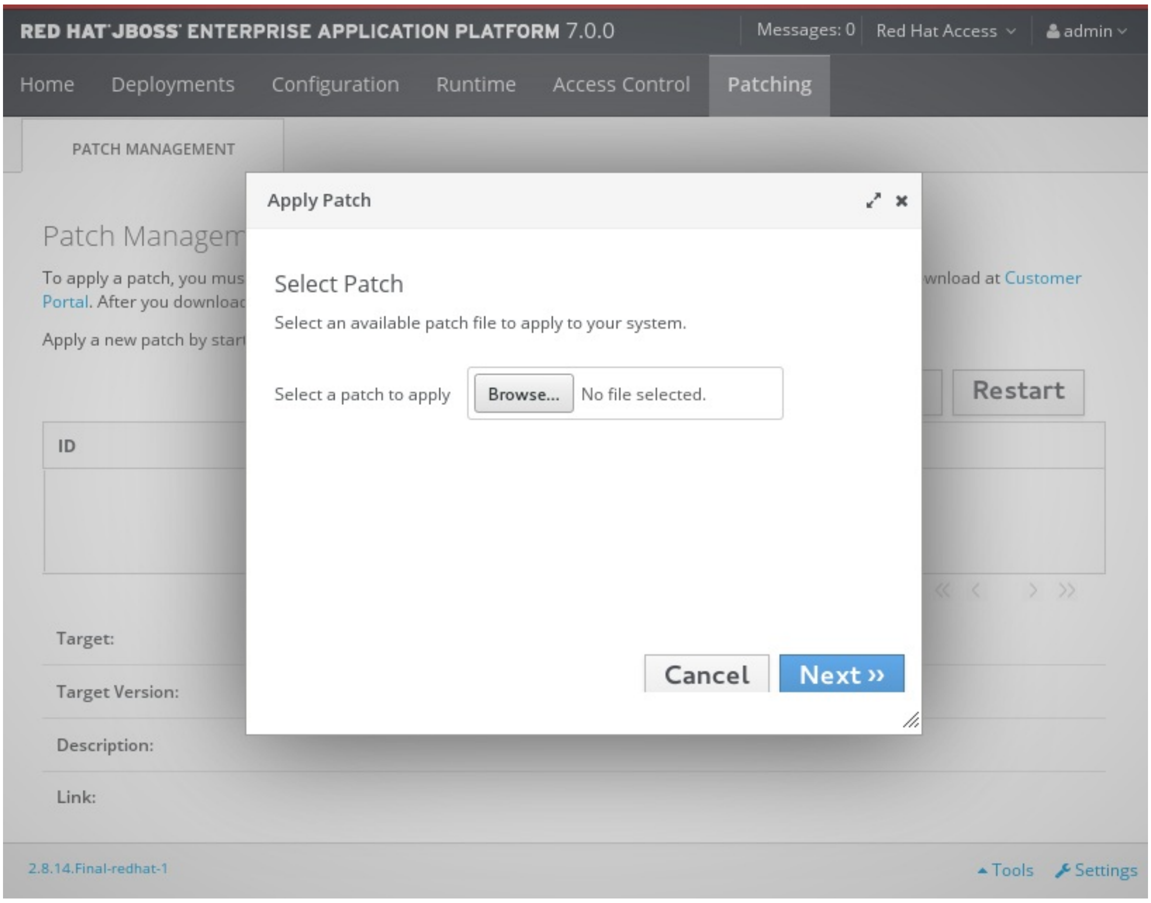
Host	Latest Applied Patch	Option
master	n/a	View ›
slave1	n/a	View ›

3. Click **Apply a New Patch**.

    a. If you are patching a managed domain host, on the next screen select whether to shutdown the servers on the host, and click **Next**.

4. Click the **Browse** button, select the downloaded patch you want to apply, and then click **Next**.

Apply Patch Screen



To apply a patch, you mus … wnload at Customer Portal. After you download

Apply a new patch by start

Select Patch

Select an available patch file to apply to your system.

Select a patch to apply [Browse...] No file selected.

.. If there are any conflicts in attempting to apply the patch, a warning will be displayed. Click **View error details** to see the detail of the conflicts. If there is a conflict, you can either cancel the operation, or select the **Override all conflicts** check box and click **Next**. Overriding conflicts will result in the content of the patch overriding any user modifications.

5. After the patch has been successfully applied, select whether to restart RH-SSO now for the patch to take effect, and click **Finish**.

3.2.1.3. Rolling Back a Patch

You can roll back a previously applied RH-SSO patch using either the management CLI or the management console.

Important

Rolling back a patch using the patch management system is not intended as a general uninstall functionality. It is only intended to be used immediately after the application of a patch that had undesirable effects.

Prerequisites

- A patch that was previously applied.

Warning

When following either procedure, use caution when specifying the value of the **`Reset Configuration`** option:

If set to **`TRUE`**, the patch rollback process will also roll back the RH-SSO server configuration files to their pre-patch state. Any changes that were made to the RH-SSO server configuration files after the patch was applied will be lost.

If set to **`FALSE`**, the server configuration files will not be rolled back. In this situation, it is possible that the server will not start after the rollback, as the patch may have altered configurations, such as namespaces, which may no longer be valid and will have to be fixed manually.

Rolling Back a Patch Using the Management CLI

1. From the management CLI, use the **`patch history`** command to find the ID of the patch that you want to roll back.

   Note

   If you are using a managed domain, you must add the **`--host=`*`HOSTNAME`*** argument to the commands in this procedure to specify the RH-SSO host.

2. Roll back the patch with the appropriate patch ID from the previous step.

   ```
   patch rollback --patch-id=PATCH_ID --reset-configuration=TRUE
   ```

   The patch tool will warn if there are any conflicts in attempting to roll back the patch. If there are conflicts, enter **`patch --help`** for the available arguments to re-run the command with an argument specifying how to resolve the conflicts.

3. Restart the RH-SSO server for the patch roll back to take effect:

   ```
   shutdown --restart=true
   ```

Rolling Back a Patch Using the Management Console

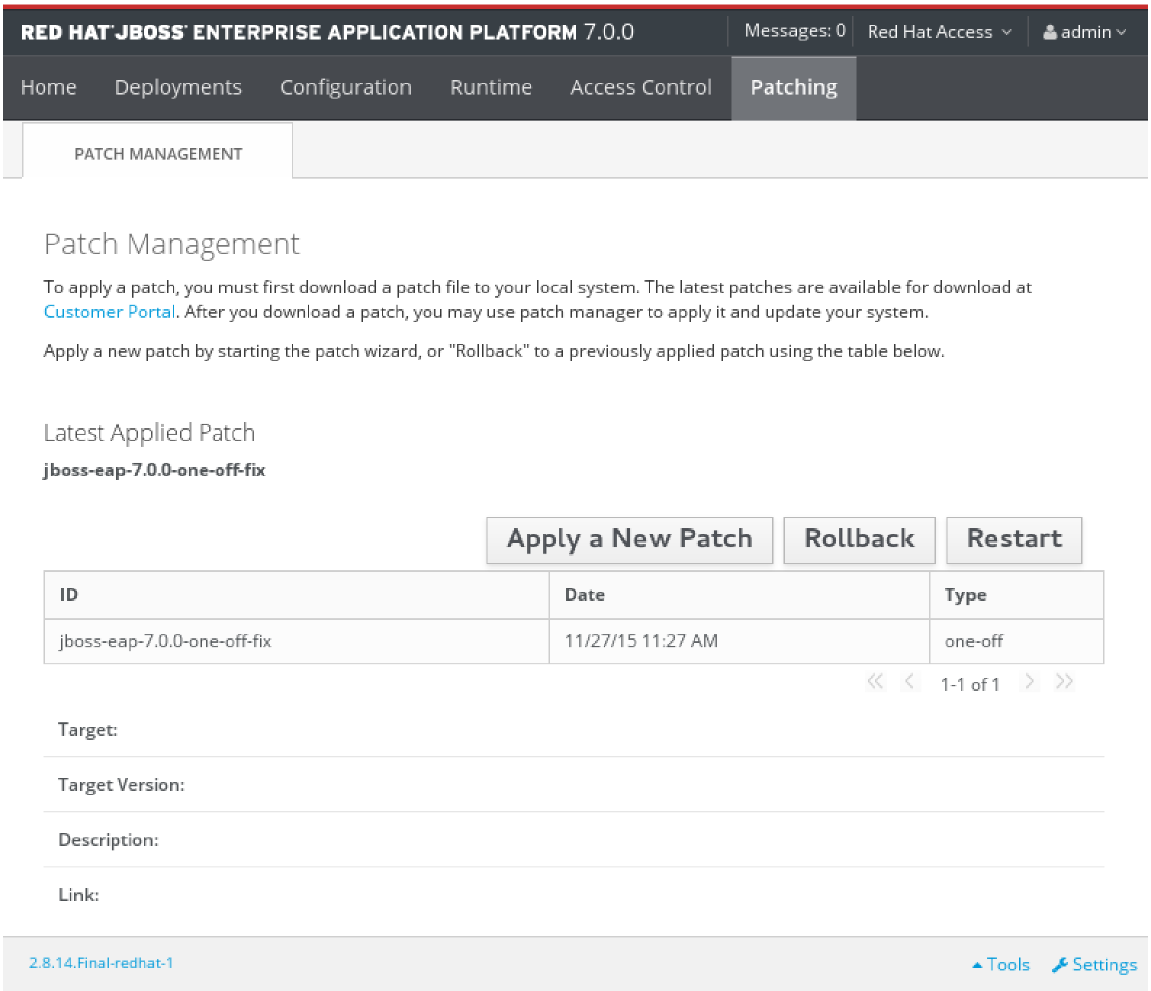
1. Open the management console and navigate to the **Patch Management** view.

   a. For a standalone server, click the **Patching** tab.

   b. For a server in a managed domain, click the **Patching** tab, then select the host that you want to patch from the table, and click **View.**

2. Select the patch that you want to rollback from those listed in the table, then click **Rollback**.

   Recent Patch History Screen



Patch Management

To apply a patch, you must first download a patch file to your local system. The latest patches are available for download at Customer Portal. After you download a patch, you may use patch manager to apply it and update your system.

Apply a new patch by starting the patch wizard, or "Rollback" to a previously applied patch using the table below.

Latest Applied Patch

jboss-eap-7.0.0-one-off-fix

ID	Date	Type
jboss-eap-7.0.0-one-off-fix	11/27/15 11:27 AM	one-off

Target:

Target Version:

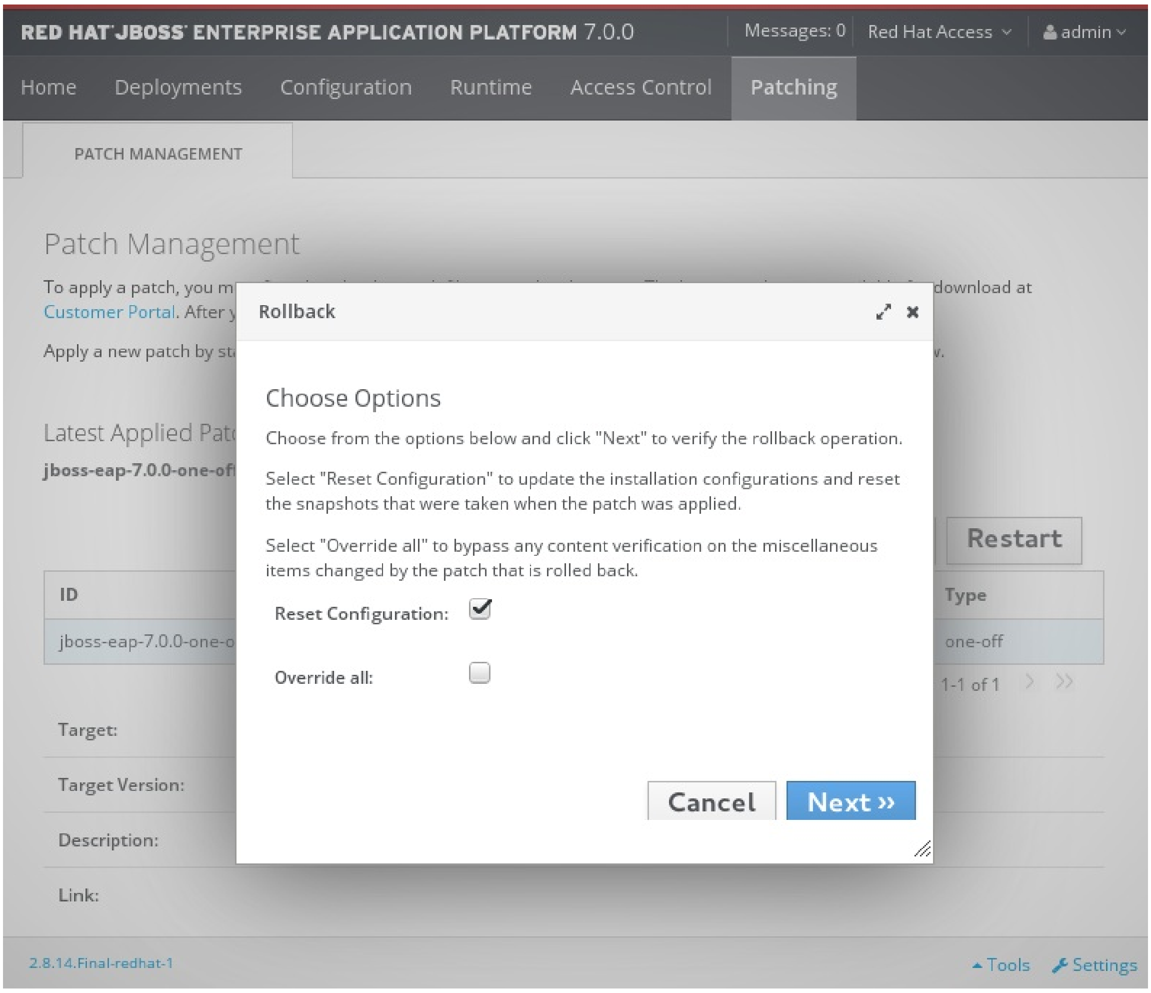
Description:

Link:

.. If you are rolling back a patch on a managed domain host, on the next screen select whether to shutdown the servers on the host, and click **Next**.

3. Choose your options for the rollback process, then click **Next**.

Patch Rollback Options



Patch Management

To apply a patch, you m download at Customer Portal. After y

Apply a new patch by st v.

Latest Applied Pat

jboss-eap-7.0.0-one-of

ID	Type
jboss-eap-7.0.0-one-o	one-off

Rollback

Choose Options

Choose from the options below and click "Next" to verify the rollback operation.

Select "Reset Configuration" to update the installation configurations and reset the snapshots that were taken when the patch was applied.

Select "Override all" to bypass any content verification on the miscellaneous items changed by the patch that is rolled back.

Reset Configuration: ☑

Override all: □

4. Confirm the options and the patch to be rolled back, then click **Next**.

    a. If there are any conflicts in attempting to rollback the patch and the **Override all** option was not selected, a warning will be displayed. Click **View error details** to see the detail of the conflicts. If there is a conflict, you can either cancel the operation, or click **Choose Options** and try the operation again with the **Override all** check box selected. Overriding conflicts will result in the rollback operation overriding any user modifications.

5. After the patch has been successfully rolled back, select whether to restart the RH-SSO server now for the changes to take effect, and click **Finish**.

3.2.1.4. Clearing Patch History

When patches are applied to a RH-SSO server, the content and history of the patches are preserved for use in rollback operations. If multiple cumulative patches are applied, the patch history may use a significant amount of disk space.

You can use the following management CLI command to remove all older patches that are not currently in use. When using this command, only the latest cumulative patch is preserved along with the GA release. This is only useful for freeing space if multiple cumulative patches have previously been applied.

```
/core-service=patching:ageout-history
```

Important

If you clear the patch history, you will not be able to roll back a previously applied patch.

3.2.2. Patching an RPM Installation

Prerequisites

- Ensure that the base operating system is up to date, and is subscribed and enabled to get updates from the standard Red Hat Enterprise Linux repositories.
- Ensure that you are subscribed to the relevant RH-SSO repository for the update.
- Back up all configuration files, deployments, and user data.

Important

For a managed domain, the RH-SSO domain controller should be updated first.

To install a RH-SSO patch via RPM from your subscribed repository, update your Red Hat Enterprise Linux system using the following command:

```
yum update
```

CHAPTER 4. UPGRADING RED HAT SINGLE SIGN-ON ADAPTERS

It is important that you upgrade Red Hat Single Sign-On Server first, and then upgrade the adapters. Earlier versions of the adapter might work with later versions of RH-SSO Server, but earlier versions of RH-SSO Server might not work with later versions of the adapter.

4.1. UPGRADING THE EAP ADAPTER

Note

If you are using RH-SSO 7.0 SAML adapters with the isPassive option set to "true" in combination with RH-SSO 7.1 Server, there is a known issue that prevents correct handling of the SAML response. When the user is not authenticated and tries to access a protected page it results in an exception rather than a redirect to authenticate. To prevent this issue from occurring, you must upgrade both RH-SSO Server and the SAML adapter at the same time.

To upgrade the EAP adapter, complete the following steps:

1. Download the rh-sso-VERSION-eap7-adapter.zip file for EAP 7.x, or rh-sso-VERSION-eap6-adapter.zip for EAP 6.4.
2. Remove the previous adapter modules by deleting the **`EAP_HOME/modules/system/add-ons/keycloak/`** directory.
3. Unzip the downloaded archive into EAP_HOME.

To upgrade on JBoss EAP 7:

```
$ cd $EAP_HOME
$ unzip -o rh-sso-VERSION-eap7-adapter.zip
```

To upgrade on JBoss EAP 6:

```
$ cd $EAP_HOME
$ unzip -o rh-sso-VERSION-eap6-adapter.zip
```

4.2. UPGRADING THE JAVASCRIPT ADAPTER

To upgrade a JavaScript adapter that has been copied to your web application, complete the following steps:

1. Download the rh-sso-VERSION-js-adapter.zip file.
2. Overwrite the keycloak.js file in your application with the keycloak.js file from the downloaded archive.

4.3. UPGRADING THE NODE.JS ADAPTER

To upgrade a Node.js adapter that has been copied to your web application, complete the following steps:

1. Download the rh-sso-VERSION-nodejs-adapter.zip file.
2. Remove the existing Node.js adapter directory
3. Unzip the updated file into its place
4. Change the dependency for keycloak-connect in the package.json of your application