



# Red Hat Enterprise Linux 8

## Using the desktop environment in RHEL 8

Configuring and customizing the GNOME 3 desktop environment on RHEL 8



# Red Hat Enterprise Linux 8 Using the desktop environment in RHEL 8

---

Configuring and customizing the GNOME 3 desktop environment on RHEL 8

## Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document describes how to customize and use GNOME 3, which is the only desktop environment available in RHEL 8. The basics of using GNOME Shell and displaying the graphics are given, as well as the instructions for system administrators for configuring GNOME on a low level and customizing the desktop environment for multiple users. The documentation also describes how to handle selected system administration tasks using the desktop environment.

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b>	<b>7</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b>	<b>8</b>
<b>CHAPTER 1. GETTING STARTED WITH GNOME</b>	<b>9</b>
1.1. WHAT GNOME 3 IS	9
1.2. GNOME ENVIRONMENTS, BACK ENDS, AND DISPLAY PROTOCOLS	9
1.3. LAUNCHING APPLICATIONS IN GNOME	9
1.3.1. Launching an application in GNOME Standard	9
1.3.2. Launching an application in GNOME Classic	10
1.3.3. Launching an application in GNOME using a command	11
1.4. MANAGING DESKTOP ICONS	11
1.4.1. Desktop icons in RHEL 8	11
1.4.2. Enabling desktop icons in GNOME Standard	12
1.4.3. Creating a desktop icon for a file	12
1.5. DISABLING THE HOT CORNER FUNCTIONALITY ON GNOME SHELL	13
1.5.1. Disabling the hot corner functionality for a single user	13
1.5.1.1. Disabling hot corner using dconf Editor	13
1.5.1.2. Disabling hot corner using gsettings	15
1.5.2. Disabling the hot corner functionality for all users	15
1.6. CONFIGURING SOUND IN GNOME	16
1.6.1. Sound configuration tools in GNOME	16
1.6.2. Accessing sound configuration in GNOME Control Center	18
1.6.3. Sound options in GNOME Control Center	19
1.7. HANDLING GRAPHICS AND PHOTOS	20
1.8. HANDLING PRINTING	21
1.8.1. Starting the Settings application for setting up printing	21
1.8.2. Adding a new printer in Settings	22
1.8.3. Configuring a printer in the Settings application	24
Displaying printer's settings menu	24
Displaying and modifying printer's details	25
Setting the default printer	26
Removing a printer	26
1.8.4. Printing a test page in the Setting application	27
1.8.5. Setting print options using the Settings application	27
1.9. SHARING MEDIA BETWEEN APPLICATIONS	28
1.10. CUSTOMIZING GNOME SHELL ENVIRONMENT WITH THE TWEAKS TOOL	28
<b>CHAPTER 2. OVERVIEW OF GNOME ENVIRONMENTS</b>	<b>31</b>
2.1. GNOME ENVIRONMENTS, BACK ENDS, AND DISPLAY PROTOCOLS	31
2.2. GNOME STANDARD	31
2.3. GNOME CLASSIC	32
2.4. ENABLING WINDOW OVERVIEW IN GNOME CLASSIC	34
2.5. GRAPHICS BACK ENDS IN RHEL 8	36
2.6. KEY DIFFERENCES BETWEEN THE WAYLAND AND X11 PROTOCOL	37
2.7. CURRENT WAYLAND LIMITATIONS	38
2.8. SELECTING GNOME ENVIRONMENT AND DISPLAY PROTOCOL	39
<b>CHAPTER 3. INSTALLING APPLICATIONS IN GNOME</b>	<b>41</b>
3.1. THE GNOME SOFTWARE APPLICATION	41
3.2. INSTALLING AN APPLICATION USING GNOME SOFTWARE	41
3.3. INSTALLING AN APPLICATION TO OPEN A FILE TYPE	43

3.4. INSTALLING AN RPM PACKAGE IN GNOME	43
3.5. INSTALLING AN APPLICATION FROM THE APPLICATION SEARCH IN GNOME	44
3.6. ADDITIONAL RESOURCES	45
<b>CHAPTER 4. REGISTERING THE SYSTEM FOR UPDATES USING GNOME</b>	<b>46</b>
4.1. REGISTERING A SYSTEM USING RED HAT ACCOUNT ON GNOME	46
4.2. REGISTERING A SYSTEM USING AN ACTIVATION KEY ON GNOME	47
4.3. UNREGISTERING THE SYSTEM USING GNOME	48
4.4. ADDITIONAL RESOURCES	49
<b>CHAPTER 5. ACCESSING THE DESKTOP REMOTELY</b>	<b>50</b>
5.1. REMOTE DESKTOP ACCESS OPTIONS	50
5.2. REMOTELY ACCESSING THE DESKTOP AS A SINGLE USER	50
5.2.1. Enabling desktop sharing on the server using GNOME	50
5.2.2. Connecting to a shared desktop using GNOME	54
5.2.3. Disabling encryption in GNOME VNC	56
5.3. REMOTELY ACCESSING THE DESKTOP AS MULTIPLE USERS	57
5.3.1. The mapping of port and display numbers to users in VNC	57
5.3.2. VNC server configuration files	58
5.3.3. Enabling multi-user VNC access on the server	59
5.3.4. Connecting to the VNC server as multiple users	60
5.4. REMOTELY ACCESSING AN INDIVIDUAL APPLICATION	61
5.4.1. Enabling X11 forwarding on the server	61
5.4.2. Launching an application remotely using X11 forwarding	61
<b>CHAPTER 6. CONFIGURING GNOME AT LOW LEVEL</b>	<b>63</b>
6.1. INTRODUCTION TO CONFIGURING GNOME	63
6.2. GSETTINGS SCHEMAS AND KEYS	64
6.3. GSETTINGS VALUES	65
6.4. MANAGING USER AND SYSTEM GNOME SETTINGS	66
6.5. DCONF PROFILES	66
6.6. CONFIGURING CUSTOM DEFAULT VALUES	67
6.7. LOCKING DOWN SPECIFIC SETTINGS	68
6.8. STORING USER SETTINGS OVER NFS	69
6.9. WORKING WITH GSETTINGS KEYS ON COMMAND LINE	69
6.9.1. Setting the value of a key	70
6.9.2. Monitoring key changes	70
6.9.3. Checking whether a key is writable	70
6.9.4. Listing the valid values for a key	71
6.9.5. Displaying the description of a key	71
6.9.6. Displaying the value of a key	71
6.9.7. Resetting the value of a key	71
6.9.8. Resetting a whole schema	72
6.9.9. Listing installed non-relocatable schemas	72
6.9.10. Listing all keys in a schema	72
6.9.11. Listing all schemas nested in a schema	73
6.9.12. Listing the keys and values in a schema recursively	73
6.10. ACKNOWLEDGEMENTS	74
<b>CHAPTER 7. CONFIGURING FILE ASSOCIATIONS</b>	<b>75</b>
7.1. MULTIPURPOSE INTERNET MAIL EXTENSION TYPES	75
MIME database	75
7.2. ADDING A CUSTOM MIME TYPE FOR ALL USERS	75
7.3. ADDING A CUSTOM MIME TYPE FOR INDIVIDUAL USERS	77

7.4. OPTIONS TO OVERRIDE DEFAULT MIME TYPES	78
7.5. OVERRIDING DEFAULT REGISTERED APPLICATION FOR ALL THE USERS	79
7.6. OVERRIDING DEFAULT REGISTERED APPLICATION FOR INDIVIDUAL USERS	79
<b>CHAPTER 8. CUSTOMIZING DESKTOP APPEARANCE</b>	<b>81</b>
8.1. CUSTOMIZING DESKTOP BACKGROUNDS	81
8.1.1. Customizing the default desktop background	81
8.1.2. Adding extra backgrounds	82
8.1.3. Frequently used background schema keys	82
8.1.4. Setting the screen shield	84
8.2. CUSTOMIZING THE BRANDING OF THE LOGIN SCREEN	85
8.3. USING GNOME SHELL EXTENSIONS TO CUSTOMIZE DESKTOP ENVIRONMENT	86
8.3.1. Overview of GNOME Shell extensions	86
8.3.2. Enabling machine-wide extensions	88
8.3.3. Locking down enabled extensions	88
8.3.4. Setting up mandatory extensions	89
<b>CHAPTER 9. CUSTOMIZING GNOME DESKTOP FEATURES</b>	<b>90</b>
9.1. CHANGING THE LANGUAGE USING DESKTOP GUI	90
9.2. ENABLING THE CTRL+ALT+BACKSPACE SHORTCUT	92
9.3. DISABLING COMMAND-LINE ACCESS	93
9.3.1. Setting the org.gnome.desktop.lockdown.disable-command-line Key	93
9.3.2. Disabling virtual terminal switching on X.Org	94
9.4. PREVENTING THE COMPUTER FROM SUSPENDING WHEN CLOSING THE LID	94
9.5. CHANGING BEHAVIOR WHEN PRESSING THE POWER BUTTON IN GRAPHICAL TARGET MODE	95
<b>CHAPTER 10. DISPLAYING THE SYSTEM SECURITY CLASSIFICATION</b>	<b>97</b>
10.1. ENABLING SYSTEM SECURITY CLASSIFICATION BANNERS	97
10.2. NOTIFYING OF THE SYSTEM SECURITY CLASSIFICATION	98
<b>CHAPTER 11. ENABLING CHINESE, JAPANESE, OR KOREAN TEXT INPUT</b>	<b>100</b>
11.1. INPUT METHODS	100
11.2. AVAILABLE INPUT METHOD ENGINES	100
11.3. INSTALLING INPUT METHOD ENGINES	100
11.4. SWITCHING THE INPUT METHOD IN GNOME	100
<b>CHAPTER 12. USING SPECIAL CHARACTERS IN GNOME</b>	<b>106</b>
12.1. ENABLING THE COMPOSE KEY FOR AN INDIVIDUAL USER WITH THE TWEAKS APPLICATION	106
12.2. ENABLING THE COMPOSE KEY FOR ANOTHER USER	107
12.3. ENABLING THE COMPOSE KEY FOR ALL USERS	107
12.4. COMPOSE KEY SEQUENCES FOR SPECIAL CHARACTERS	108
<b>CHAPTER 13. TYPING EMOJI CHARACTERS</b>	<b>110</b>
13.1. TYPING EMOJI CHARACTERS IN GTK APPLICATIONS	110
13.2. TYPING EMOJI CHARACTERS IN ANY APPLICATIONS	110
<b>CHAPTER 14. MANAGING STORAGE VOLUMES IN GNOME</b>	<b>112</b>
14.1. THE GVFS SYSTEM	112
14.2. THE FORMAT OF THE GVFS URI STRING	112
14.3. MOUNTING A STORAGE VOLUME IN GNOME	112
14.4. UNMOUNTING A STORAGE VOLUME IN GNOME	113
14.5. ACCESS TO GVFS MOUNTS IN THE FILE SYSTEM	113
14.6. AVAILABLE GIO COMMANDS	113
14.7. SAMPLE GIO COMMANDS	115
14.8. OVERVIEW OF GVFS METADATA	115

14.9. SETTING CUSTOM GIO METADATA ATTRIBUTE	116
14.10. PASSWORD MANAGEMENT OF GVFS MOUNTS	117
14.11. GVFS BACK ENDS	117
<b>CHAPTER 15. TROUBLESHOOTING VOLUME MANAGEMENT IN GNOME</b>	<b>119</b>
15.1. TROUBLESHOOTING ACCESS TO GVFS LOCATIONS FROM NON-GIO CLIENTS	119
15.2. TROUBLESHOOTING AN INVISIBLE CONNECTED USB DISK	119
15.3. TROUBLESHOOTING UNKNOWN OR UNWANTED PARTITIONS LISTED IN FILES	120
15.4. TROUBLESHOOTING IF A CONNECTION TO THE REMOTE GVFS FILE SYSTEM IS UNAVAILABLE	120
15.5. TROUBLESHOOTING A BUSY DISK IN GNOME	120
<b>CHAPTER 16. MANAGING BOOKMARKS IN GNOME</b>	<b>122</b>
16.1. BOOKMARKS IN GNOME	122
16.2. ADDING A BOOKMARK IN FILES	122
16.3. ADDING A BOOKMARK FOR ALL USERS	122
<b>CHAPTER 17. CUSTOMIZING DEFAULT FAVORITE APPLICATIONS</b>	<b>124</b>
17.1. SETTING DIFFERENT FAVORITE APPLICATIONS FOR INDIVIDUAL USERS	124
17.2. SETTING THE SAME FAVORITE APPLICATIONS FOR ALL USERS	124
<b>CHAPTER 18. AUTHENTICATING THE USER IN THE DESKTOP ENVIRONMENT</b>	<b>126</b>
18.1. USING ENTERPRISE CREDENTIALS TO AUTHENTICATE IN GNOME	126
18.1.1. Logging in with Enterprise Credentials in GNOME	126
18.1.2. Configuring enterprise credentials at the GNOME welcome screen	126
18.1.3. Adding an authenticated user with enterprise credentials in GNOME	127
18.1.4. Troubleshooting enterprise login in GNOME	127
18.2. ENABLING SMART CARD AUTHENTICATION	128
18.2.1. Configuring smart card authentication in GDM using the GUI	128
18.2.2. Configuring smart card authentication in GDM using the command line	128
18.2.3. Enabling the smart card authentication method in the system	129
18.3. FINGERPRINT AUTHENTICATION	129
<b>CHAPTER 19. LOCKING DOWN SELECTED TASKS</b>	<b>130</b>
19.1. LOCKING DOWN PRINTING	130
19.2. LOCKING FILE SAVING ON DISK	131
19.3. LOCKING REPARTITIONING	131
19.4. LOCKING DOWN USER LOGOUT AND USER SWITCHING	132
<b>CHAPTER 20. RESTRICTING THE SESSION TO A SINGLE APPLICATION</b>	<b>135</b>
20.1. SINGLE-APPLICATION MODE	135
20.2. ENABLING SINGLE-APPLICATION MODE	135
<b>CHAPTER 21. MANAGING USER SESSIONS</b>	<b>137</b>
21.1. WHAT GDM IS	137
21.2. RESTARTING GDM	137
21.3. ADDING AN AUTOSTART APPLICATION FOR ALL USERS	137
21.4. CONFIGURING AUTOMATIC LOGIN	138
21.5. CONFIGURING AUTOMATIC LOGOUT	138
21.6. SETTING A DEFAULT DESKTOP SESSION FOR ALL USERS	140
21.7. SETTING SCREEN BRIGHTNESS AND IDLE TIME	140
Configuring the drop in the brightness level	140
Setting brightness level	141
Setting idle time	141
21.8. LOCKING THE SCREEN WHEN THE USER IS IDLE	141
21.9. SCREENCAST RECORDING	142



---

<b>CHAPTER 22. CONFIGURING THE SCREEN ROTATION</b> .....	<b>144</b>
22.1. CONFIGURING THE SCREEN ROTATION FOR A SINGLE USER	144
22.2. CONFIGURING THE SCREEN ROTATION FOR ALL USERS	145
<b>CHAPTER 23. ENABLING ACCESSIBILITY FOR VISUALLY IMPAIRED USERS</b> .....	<b>146</b>
23.1. COMPONENTS THAT PROVIDE ACCESSIBILITY FEATURES	146
23.2. ENABLING THE UNIVERSAL ACCESS MENU	146
23.3. ENABLING THE SCREEN READER	147
23.4. ENABLING A BRAILLE DISPLAY DEVICE	148
23.4.1. Supported types of Braille display device	148
23.4.2. Enabling the brltty service	148
23.4.3. Authorizing users of a Braille display device	149
23.4.4. Setting the driver for a Braille display device	149
23.4.5. Connecting a Braille display device	150
23.4.6. Setting the text table	151
23.4.7. Setting the contraction table	151
<b>CHAPTER 24. TABLETS</b> .....	<b>152</b>
24.1. PREPARING A TABLET DEFINITION FILE	152
24.2. ADDING SUPPORT FOR A NEW TABLET	154
24.3. WHERE IS THE WACOM TABLET CONFIGURATION STORED	155
24.4. LISTING AVAILABLE WACOM TABLET CONFIGURATION PATHS	155
<b>CHAPTER 25. INSTALLING APPLICATIONS USING FLATPAK</b> .....	<b>157</b>
25.1. THE FLATPAK TECHNOLOGY	157
25.2. SETTING UP FLATPAK	157
25.3. ENABLING THE RED HAT FLATPAK REMOTE	157
25.4. SEARCHING FOR FLATPAK APPLICATIONS	158
25.5. INSTALLING FLATPAK APPLICATIONS	159
25.6. LAUNCHING FLATPAK APPLICATIONS	159
25.7. UPDATING FLATPAK APPLICATIONS	160
25.8. INSTALLING FLATPAK APPLICATIONS IN THE GRAPHICAL INTERFACE	160
25.9. UPDATING FLATPAK APPLICATIONS IN THE GRAPHICAL INTERFACE	161



## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

### Submitting comments on specific passages

1. View the documentation in the **Multi-page HTML** format and ensure that you see the **Feedback** button in the upper right corner after the page fully loads.
2. Use your cursor to highlight the part of the text that you want to comment on.
3. Click the **Add Feedback** button that appears near the highlighted text.
4. Add your feedback and click **Submit**.

### Submitting feedback through Jira (account required)

1. Log in to the [Jira](#) website.
2. Click **Create** in the top navigation bar
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

# CHAPTER 1. GETTING STARTED WITH GNOME

## 1.1. WHAT GNOME 3 IS

Red Hat Enterprise Linux 8 is shipped with the default desktop environment GNOME 3.

GNOME 3 represents a presentation layer that provides a graphical user interface as well as the focused working environment, which enables you to access all your work from one place.

## 1.2. GNOME ENVIRONMENTS, BACK ENDS, AND DISPLAY PROTOCOLS

GNOME 3 provides two user environments:

- GNOME Standard
- GNOME Classic

Both environments can use two different protocols as their graphical back ends:

- The **X11** protocol, which uses **X.Org** as the display server.
- The **Wayland** protocol, which uses **GNOME Shell** as the **Wayland** compositor and display server.

This solution of display server is further referred as **GNOME Shell on Wayland**

The default combination in RHEL 8 is GNOME Standard environment using **GNOME Shell on Wayland** as the display server. However, due to certain **Wayland** limitations, you might want to switch the graphics protocol stack to **X11**. You may also want to switch from GNOME Standard to GNOME Classic.

### Additional resources

- For more information about graphics based on the **Wayland** protocol, see [Section 2.6, “Key differences between the Wayland and X11 protocol”](#).
- For information about how to switch the environments, see [Section 2.8, “Selecting GNOME environment and display protocol”](#).

## 1.3. LAUNCHING APPLICATIONS IN GNOME

This section describes various approaches that you can use to launch available applications in GNOME 3.

### 1.3.1. Launching an application in GNOME Standard

This procedure launches a graphical application in the GNOME Standard user environment.

#### Prerequisites

- You are using the GNOME Standard environment.

#### Procedure

1. Go to the **Activities Overview** screen by moving the cursor to the area of the upper-left corner and push the cursor to the screen corner. The **Activities Overview** menu opens automatically.
2. Find the application using either of the following ways:
  - Click the **Show Applications** icon in the vertical bar on the left side. Optionally, you can choose between displaying all or just the frequent applications by using the **Frequent** or **All** switch at the bottom of the screen.



- Start typing the name of the required application in the search entry.
3. Click an application from the displayed list.

### 1.3.2. Launching an application in GNOME Classic

This procedure launches a graphical application in the GNOME Classic user environment.

#### Prerequisites

- You are using the GNOME Classic environment.

#### Procedure

1. Go to the **Applications** menu.
2. Choose the required application from the available categories, which include:
  - Favorites
  - Accessories
  - Internet
  - Office

- Sound & Video
- Sundry
- System Tools
- Utilities

### 1.3.3. Launching an application in GNOME using a command

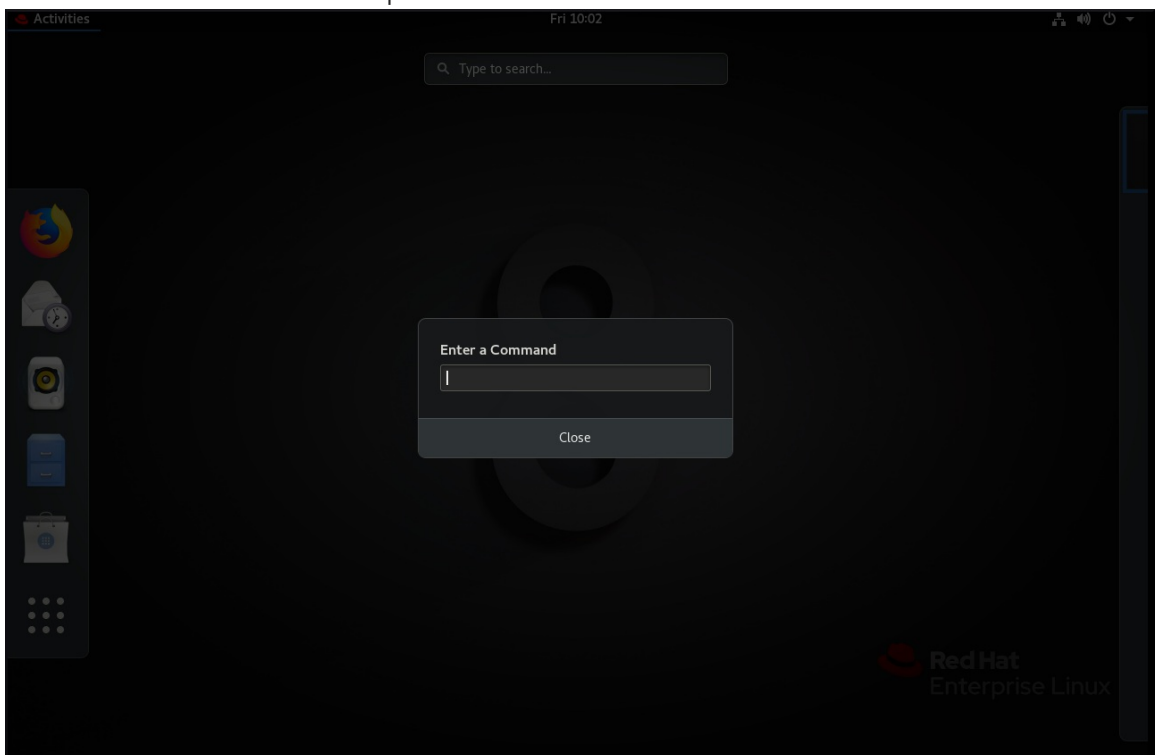
This procedure launches a graphical application in GNOME by entering a command.

#### Prerequisites

- You know the command that starts the application.

#### Procedure

1. Open a command prompt using either of the following ways:
  - Open a terminal.
  - Press the **Alt+F2** shortcut to open the **Enter a Command** screen.



2. Write the command in the command prompt.
3. Confirm the command by pressing **Enter**.

## 1.4. MANAGING DESKTOP ICONS

You can enable the desktop icons functionality and move files to the desktop.

### 1.4.1. Desktop icons in RHEL 8

In RHEL 8, the **Files** application no longer provides the desktop icons functionality. Instead, desktop icons are provided by the **Desktop icons** GNOME Shell extension, which is available from the **gnome-shell-extension-desktop-icons** package.

## Desktop icons in GNOME Classic

The GNOME Classic environment includes the **gnome-shell-extension-desktop-icons** package by default. Desktop icons are always on, and you cannot turn them off.

## Desktop icons in GNOME Standard

In GNOME Standard, desktop icons are disabled by default.

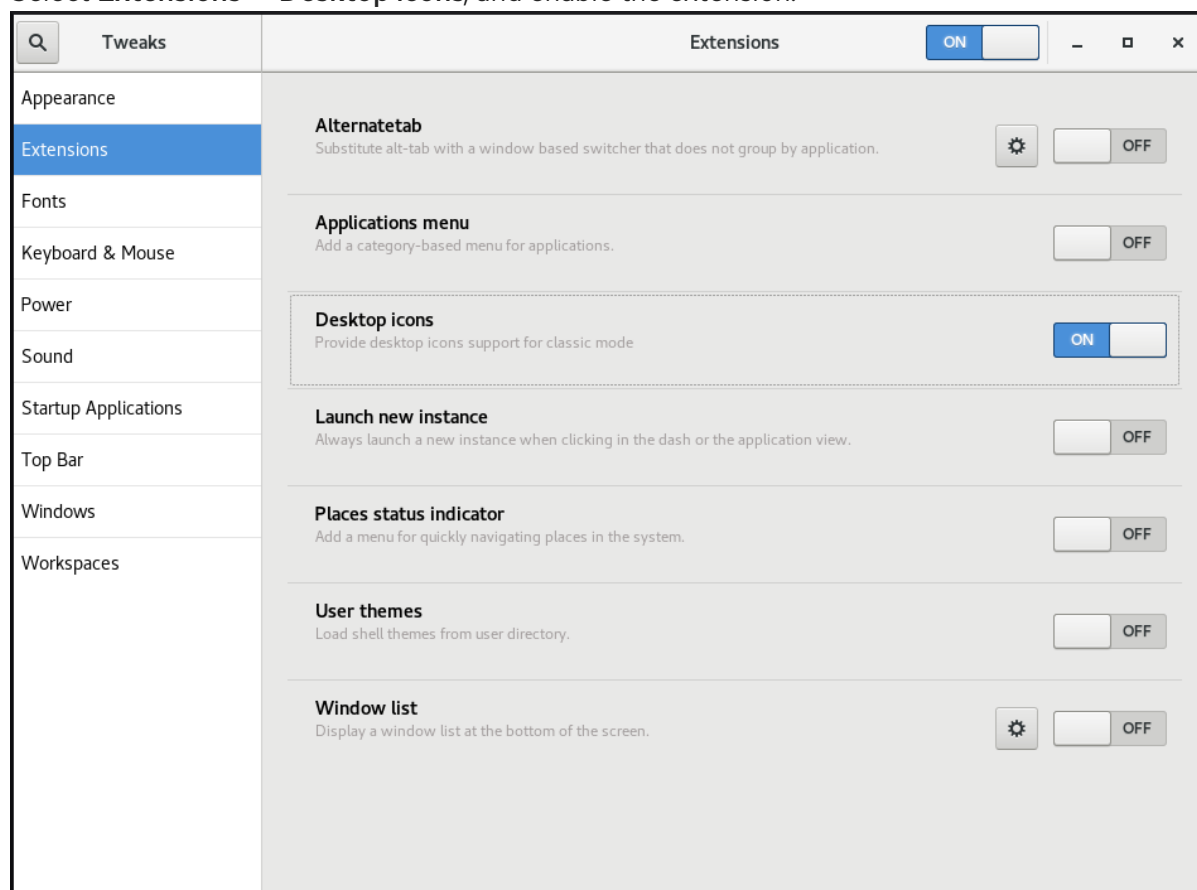
If you have only the GNOME Standard environment available, and not GNOME Classic, you must install the **gnome-shell-extension-desktop-icons** package.

### 1.4.2. Enabling desktop icons in GNOME Standard

This procedure enables the desktop icons functionality in the GNOME Standard environment.

#### Procedure

1. Open the **Tweaks** application.
2. Select **Extensions → Desktop icons**, and enable the extension.



### 1.4.3. Creating a desktop icon for a file

This procedure creates a desktop icon for an existing file.

#### Prerequisites



- The **Desktop icons** extension is enabled.

### Procedure

- Move the selected file into the `~/Desktop/` directory.

### Verification steps

- Check that the icon for the file appears on the desktop.

## 1.5. DISABLING THE HOT CORNER FUNCTIONALITY ON GNOME SHELL

The GNOME environment provides the hot corner functionality, which is enabled by default. This means that when you move the cursor to the area of the upper-left corner and push the cursor to the screen corner, the **Activities Overview** menu opens automatically.

However, you may want to disable this feature to not open **Activities Overview** unintentionally.

To do so, you can use the following tools:

- The **dconf Editor** application
- The **gsettings** command-line utility
- The **No topleft hot corner** extension

The selection of the tool might depend on whether you want to disable the hot corner functionality for a single user or for all users on the system. By using **dconf Editor** or **gsettings**, you can disable hot corner only for a single user. To disable hot corner system-wide, use the **No topleft hot corner** extension.

### 1.5.1. Disabling the hot corner functionality for a single user

To disable the hot corner functionality for a single user, you can use either the **dconf Editor** application or the **gsettings** command-line utility.

#### 1.5.1.1. Disabling hot corner using dconf Editor

To disable the hot corner functionality using the **dconf Editor** application, follow this procedure.



#### NOTE

This procedure disables the hot corner functionality for a **single** user.

### Prerequisites

- The **dconf Editor** application is installed on the system:

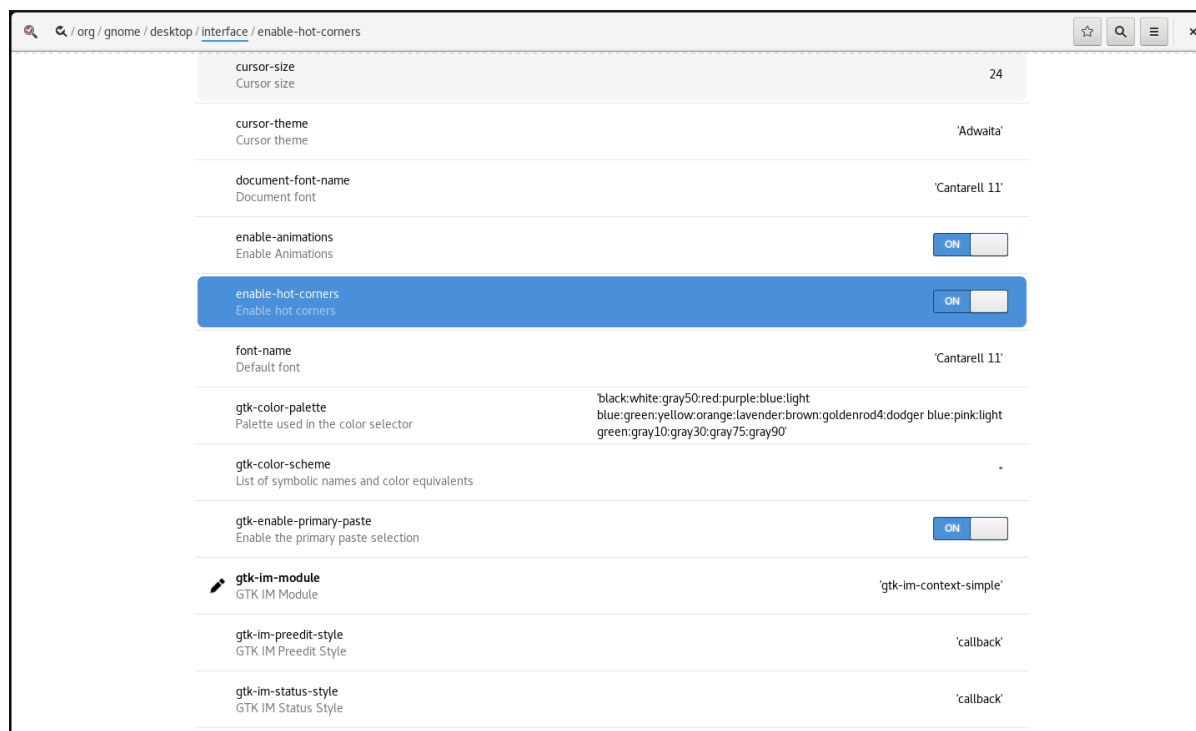
```
# yum install dconf-editor
```

### Procedure

1. Open the **dconf Editor** application.

2. Choose the **org.gnome.desktop.interface** menu.
3. Find the **enable-hot-corners** option.  
This option is by default set to **On**.

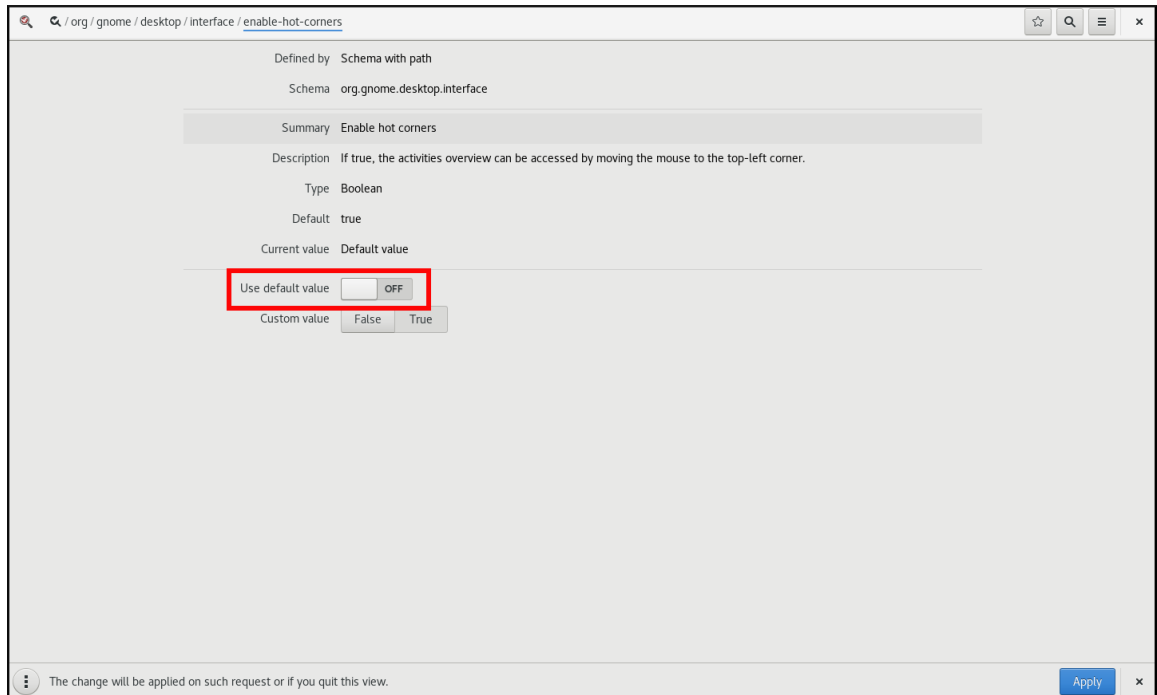
### Default settings of enable-hot-corners



4. Set **enable-hot-corners** to **Off**.  
You can do this either by:

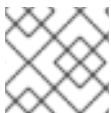
- Setting **enable-hot-corners** to **Off** in the same window.
- Clicking the line with **enable-hot-corners**, and proceeding to a new window. In the new window, you can switch the hot corner feature off.

### Switching the hot corner functionality off



### 1.5.1.2. Disabling hot corner using gsettings

To disable the hot corner functionality using the **gsettings** command-line utility, follow this procedure.



#### NOTE

This procedure disables the hot corner functionality for a **single** user.

#### Procedure

- Disable the hot corner feature:

```
$ gsettings set org.gnome.desktop.interface enable-hot-corners false
```

#### Verification steps

- Optionally, verify that the hot corner feature is disabled:

```
$ gsettings get org.gnome.desktop.interface enable-hot-corners
false
```

### 1.5.2. Disabling the hot corner functionality for all users

With the GNOME Shell extension called **No topleft hot corner** provided by the **gnome-shell-extension-no-hot-corner** package, you can disable the hot corner feature system-wide.

#### Prerequisites

- The **gnome-shell-extension-no-hot-corner** package is installed on the system:

```
# yum install gnome-shell-extension-no-hot-corner
```

## Procedure

1. Enable the **No topleft hot corner** extension by switching it on in the **Tweaks** tool.  
For more information about how to use **Tweaks**, see [Section 1.10, "Customizing GNOME Shell environment with the Tweaks tool"](#).
2. Log out, and restart the user session so that the extension can take effect.

## 1.6. CONFIGURING SOUND IN GNOME

You can configure sound volume and other sound options in GNOME.

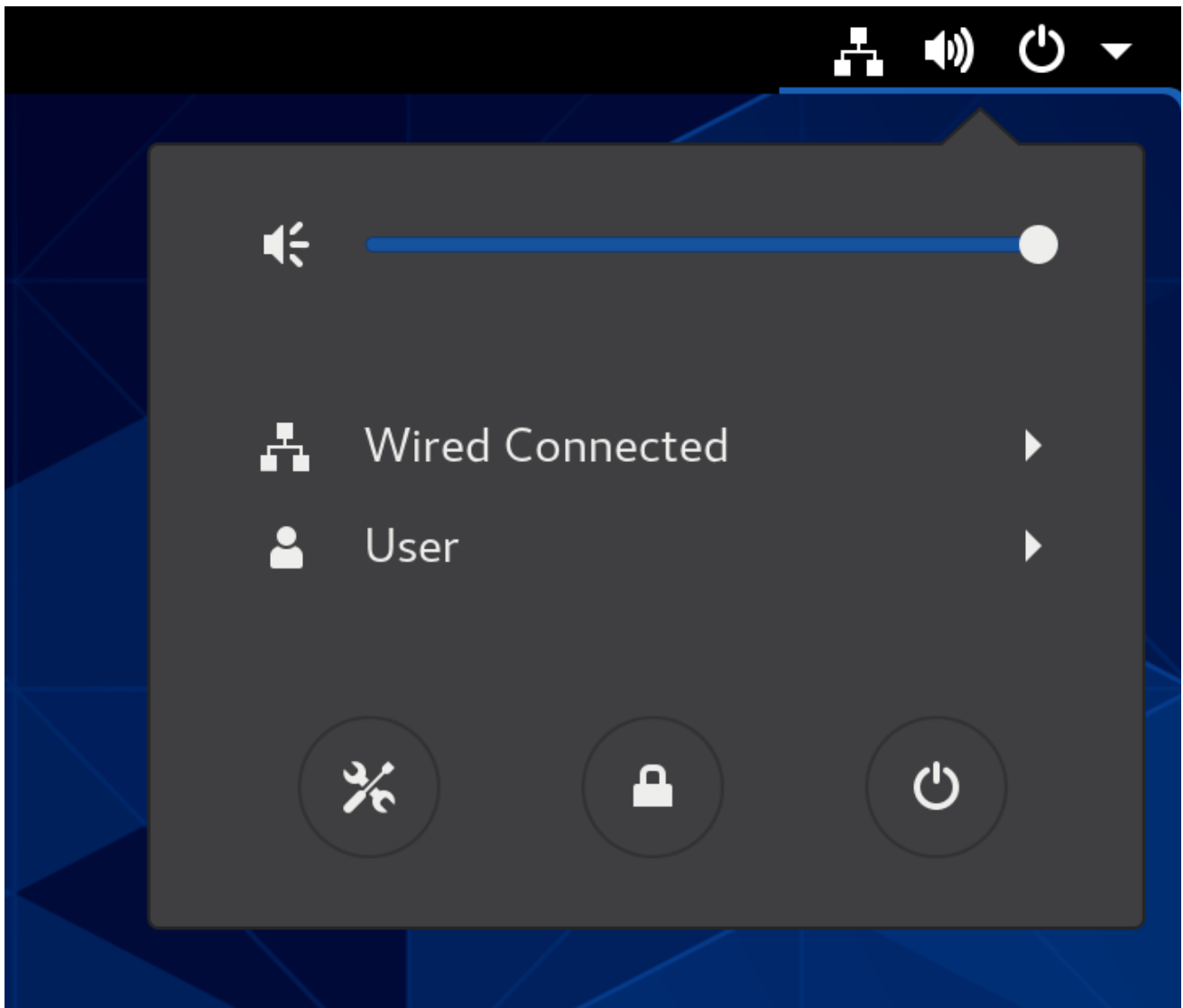
### 1.6.1. Sound configuration tools in GNOME

In RHEL 8, the **PulseAudio** sound server handles sound output and input. **PulseAudio** lets programs output the audio using the **pulseaudio** daemon.

To configure sound, you can use one of the following graphical applications in GNOME:

#### System menu

The **system menu** is located in the top-right screen corner. It enables you only to set the intensity of the sound output or sound input through the sound bar. The sound bar for input sound is available only if you are running an application that is using an internal microphone (built-in audio), such as some teleconference tools.

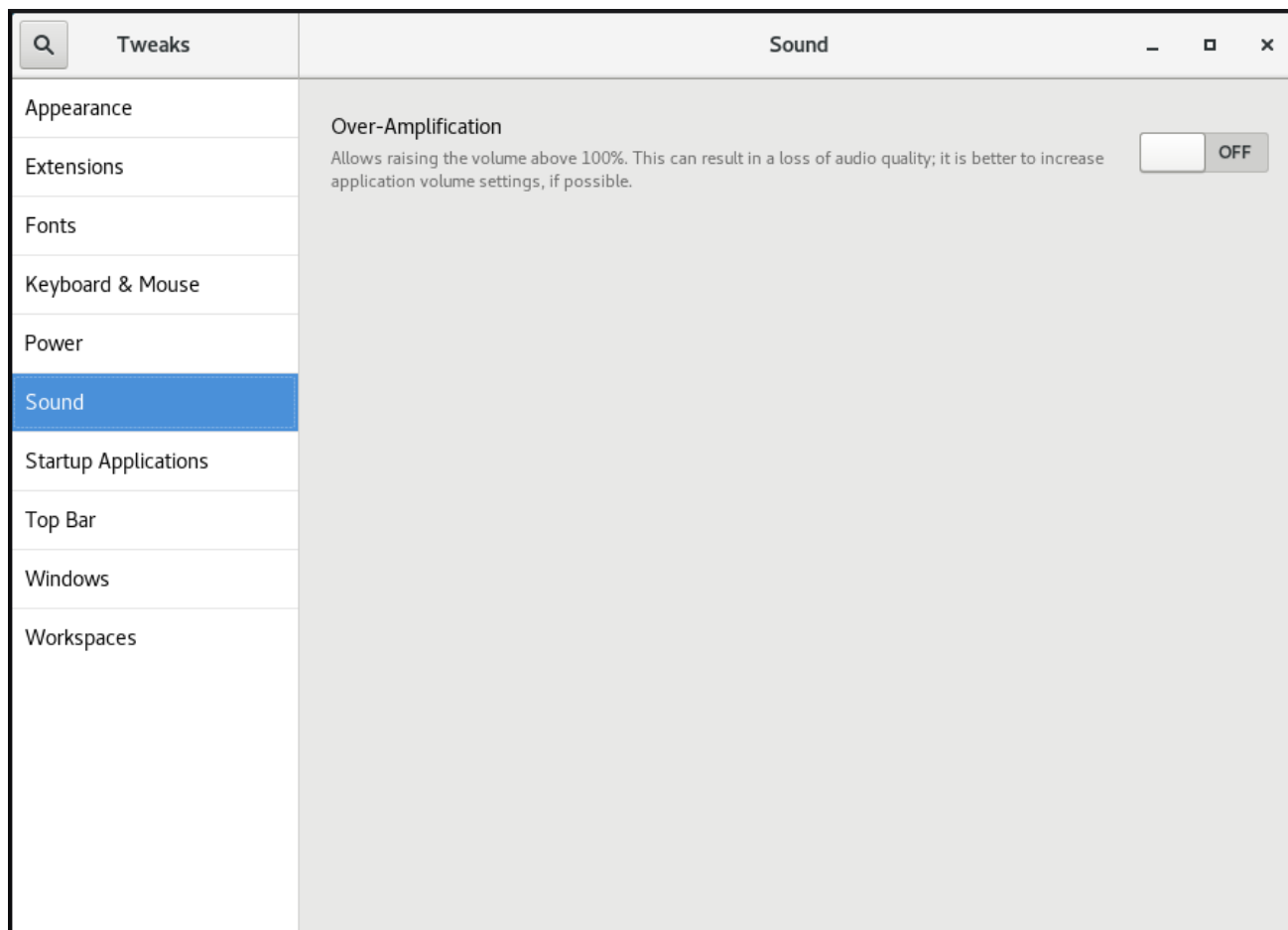


### GNOME Control Center

The **GNOME Control Center** (**Settings**) application provides other general options to configure sound.

### Tweaks

The **Tweaks** application enables you to configure only volume over-amplification.



### Additional resources

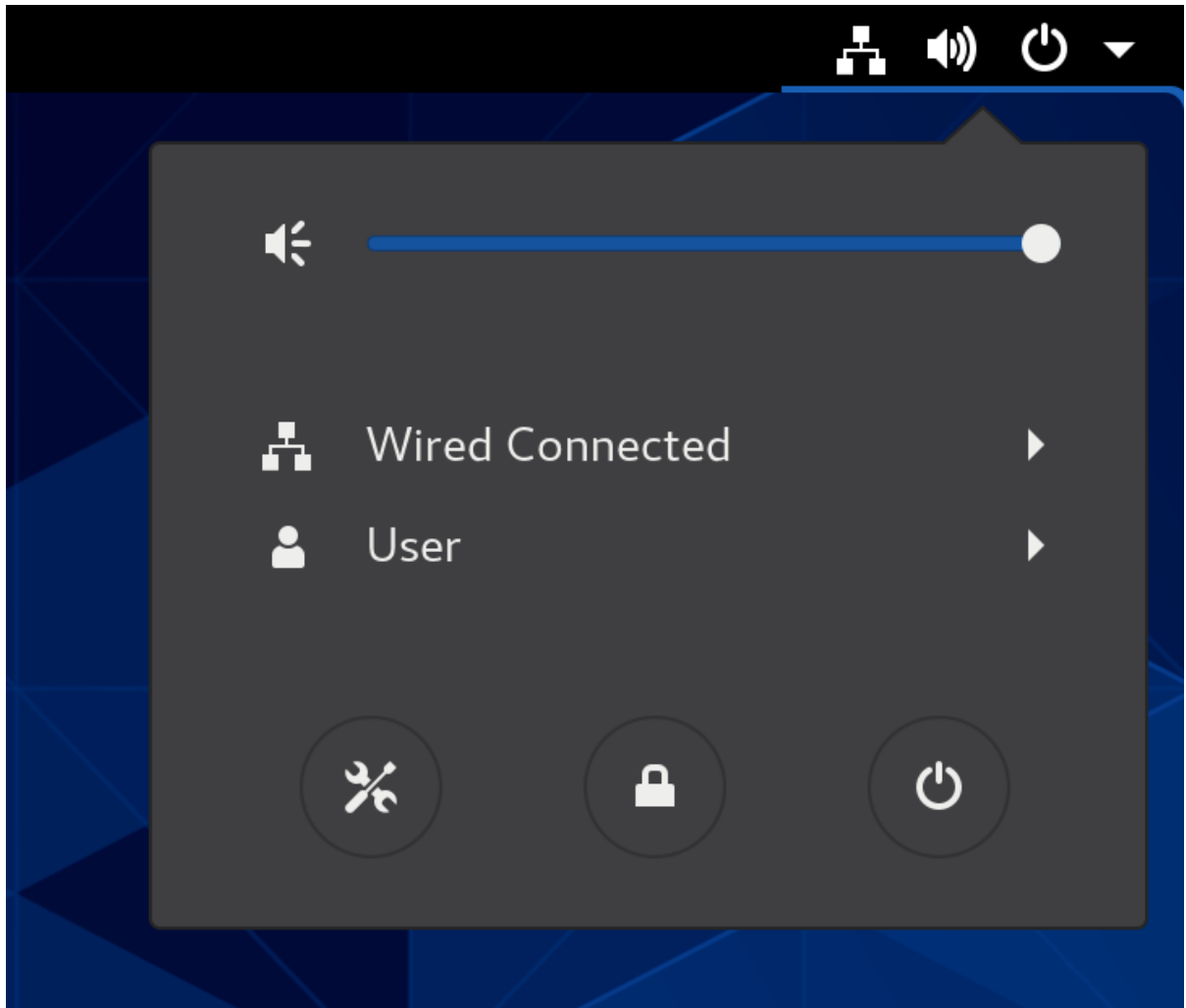
- For more information about **PulseAudio**, see the **pulseaudio** man page.

### 1.6.2. Accessing sound configuration in GNOME Control Center

This procedure opens the sound configuration screen in the **GNOME Control Center** application.

#### Procedure

1. Launch **GNOME Control Center**.  
You can use one of the approaches described in [Section 1.3, "Launching applications in GNOME"](#). Alternatively, you can also launch it from the **system menu** by clicking on its icon.



2. In **GNOME Control Center**, choose **Sound** from the left vertical bar.

### 1.6.3. Sound options in GNOME Control Center

Through the **Sound** menu in **GNOME Control Center**, you can configure the following sound options:

#### Output and Input

The **Output** and **Input** menus show only the built-in audio devices unless you connect any external device that can handle sound.

The **Output** menu enables you to select the required profile from available analog or digital profiles that are displayed depending on available output devices.

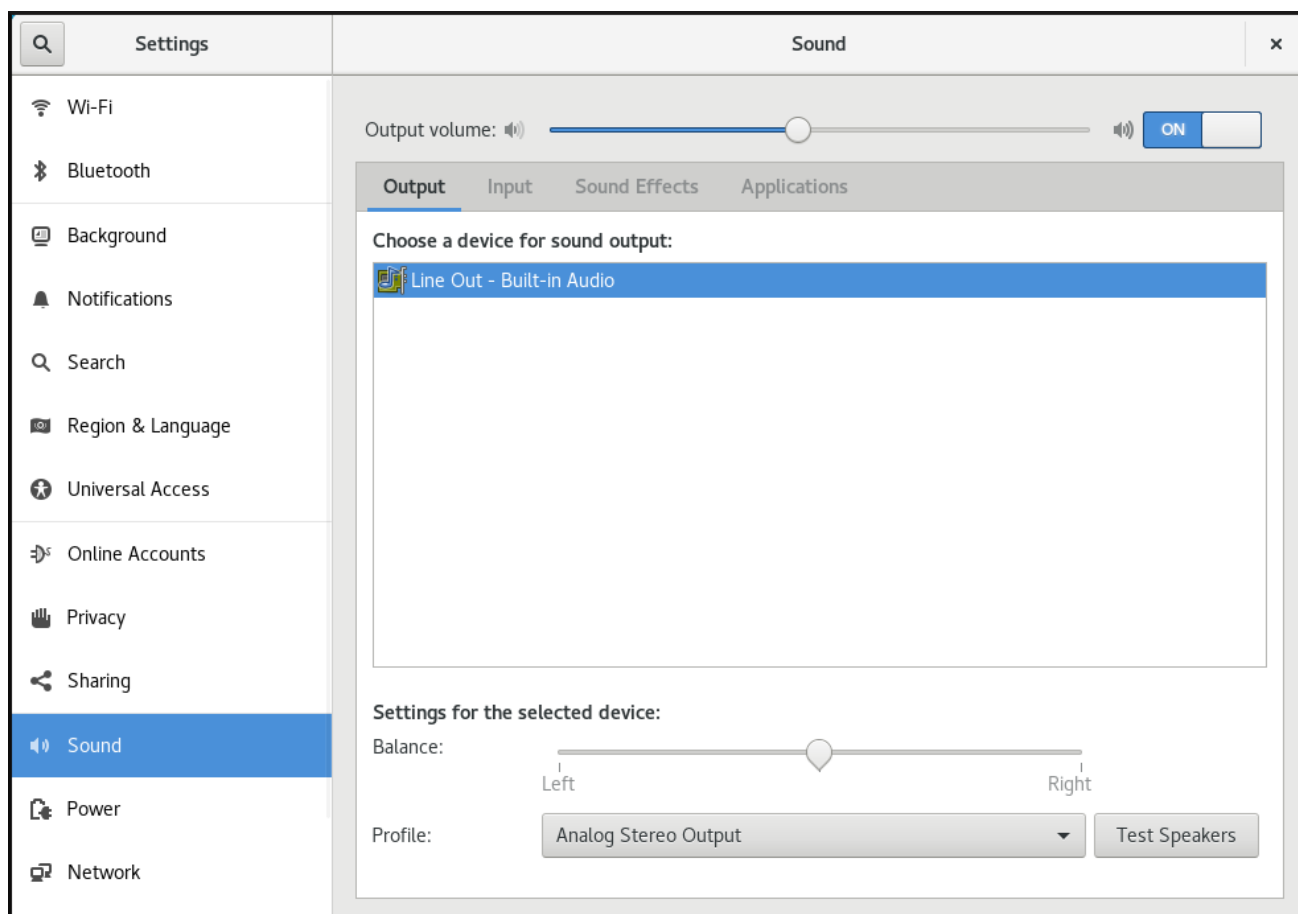
#### Sound Effects

The **Sound Effects** menu sets the volume and theme of system audio alerts.

#### Applications

The **Applications** menu shows all currently running applications that can process sound, and allows you to amplify or lower the sound of a particular application.

#### The Output tab on the sound configuration screen



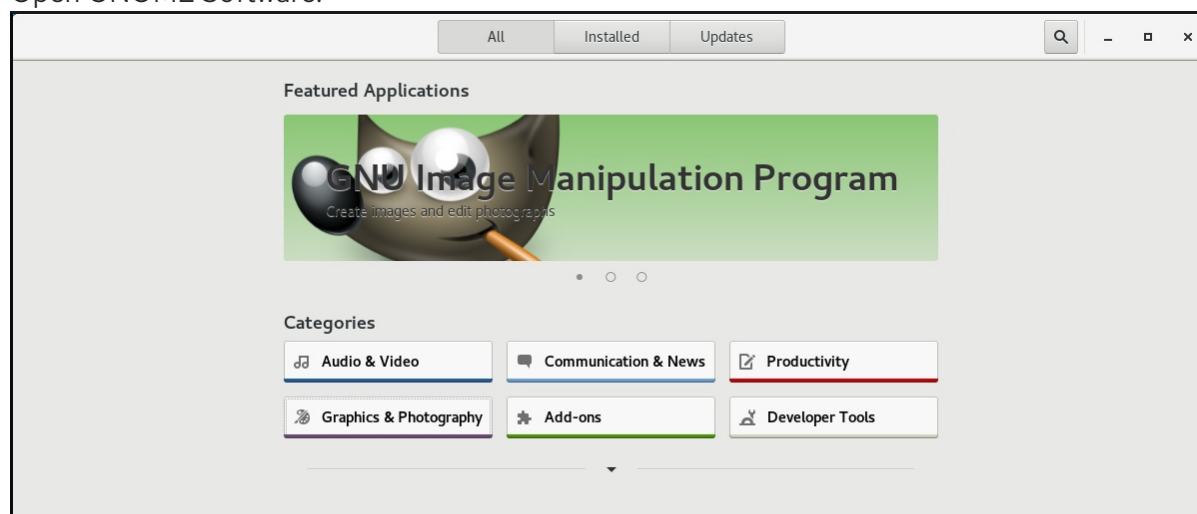
## 1.7. HANDLING GRAPHICS AND PHOTOS

GNOME provides multiple tools to handle graphics and photography.

You can check the available tools under the **Graphics & Photography** menu in GNOME Software.

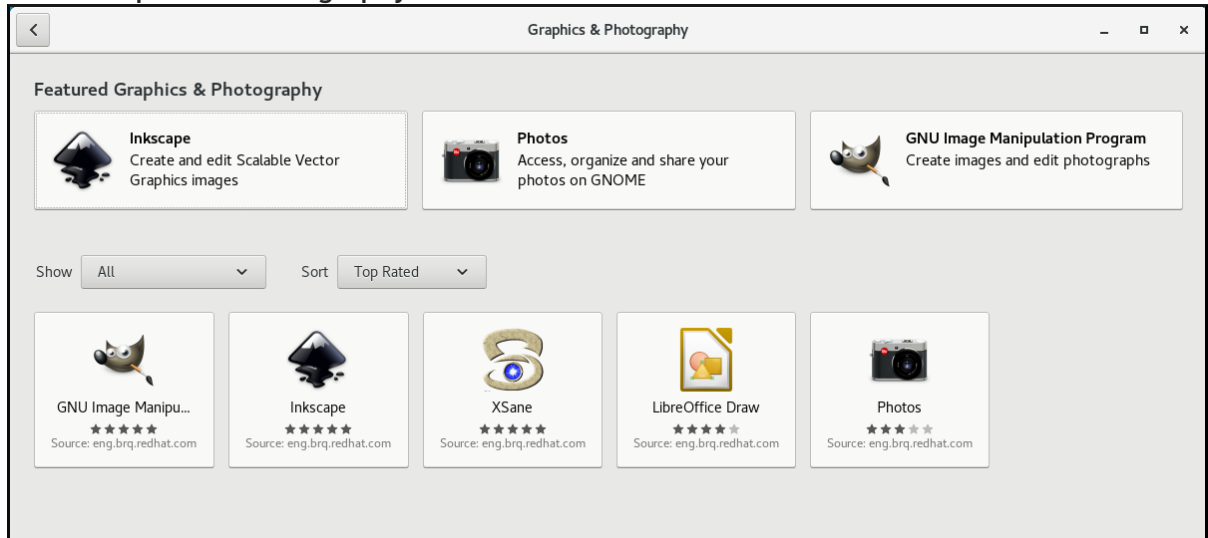
### Procedure

1. Open GNOME Software.





## 2. Go to **Graphics & Photography**.



## 3. Browse the available tools, such as the following:

### **Photos**

For accessing, organizing and sharing your photos.

### **GNU Image Manipulation Program**

For creating images and editing photographs.

### **Inkspace**

For creating and editing scalable vector graphics images.

### **XSane**

For scanning images with a scanner.

### **LibreOffice Draw**

For create and editing drawings, flow charts, and logos.

## 1.8. HANDLING PRINTING

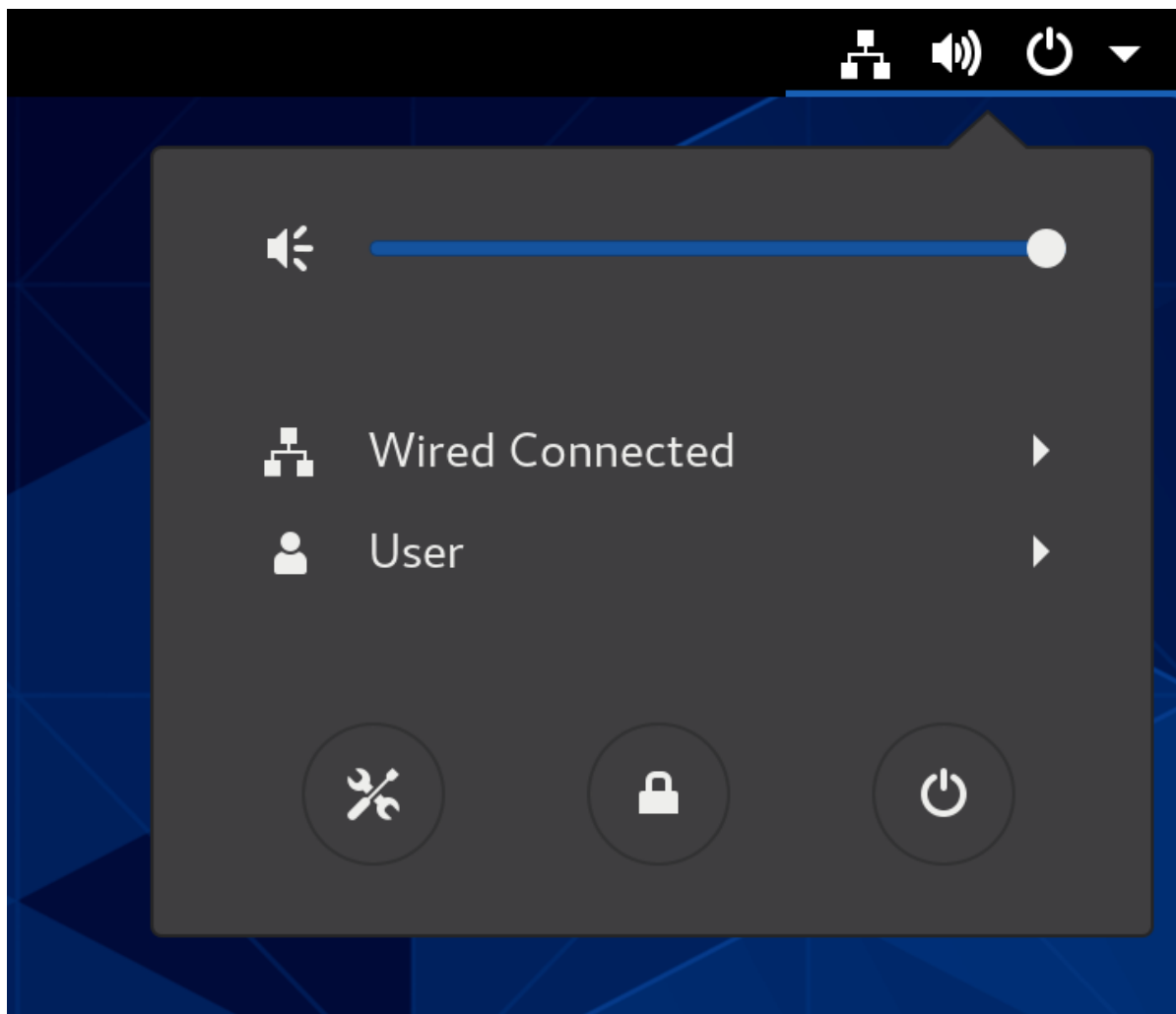
In GNOME, you can set up printing using the **Settings** application.

### 1.8.1. Starting the Settings application for setting up printing

#### **Procedure**

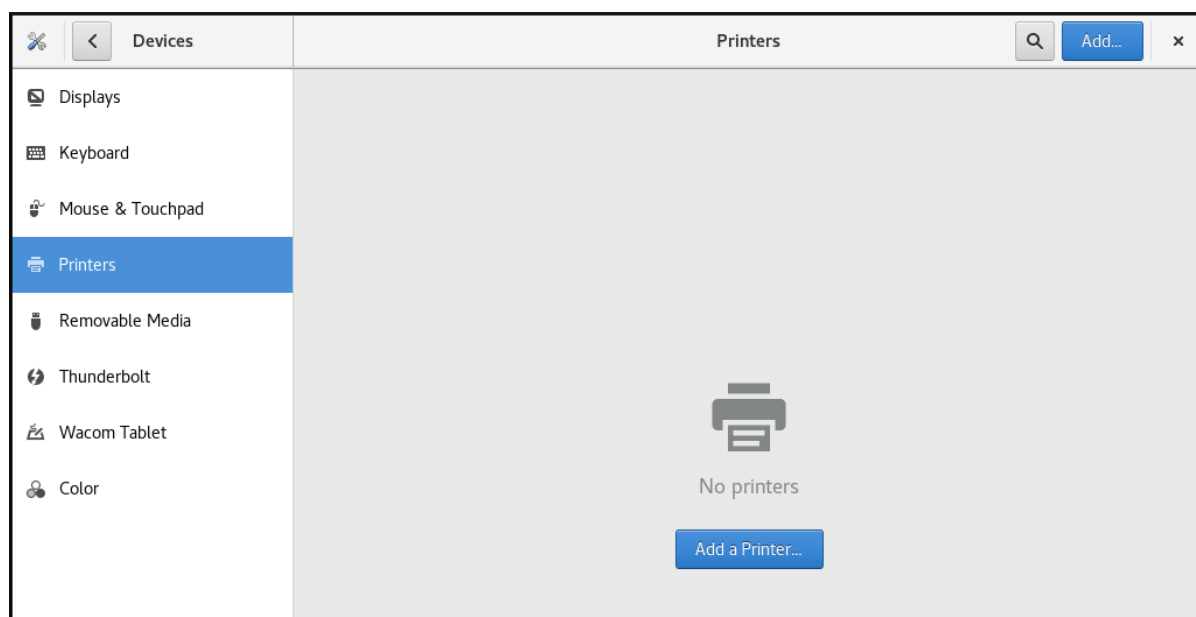
1. Use one of the approaches described in [Section 1.3, "Launching applications in GNOME"](#) to start the **Settings** application.

Moreover, you can also start the **Settings** application from the **system menu** in the top right corner by clicking on the "Settings" icon.



2. When the **Settings** application appears, go to **Devices → Printers**.

Figure 1.1. Settings configuration tool

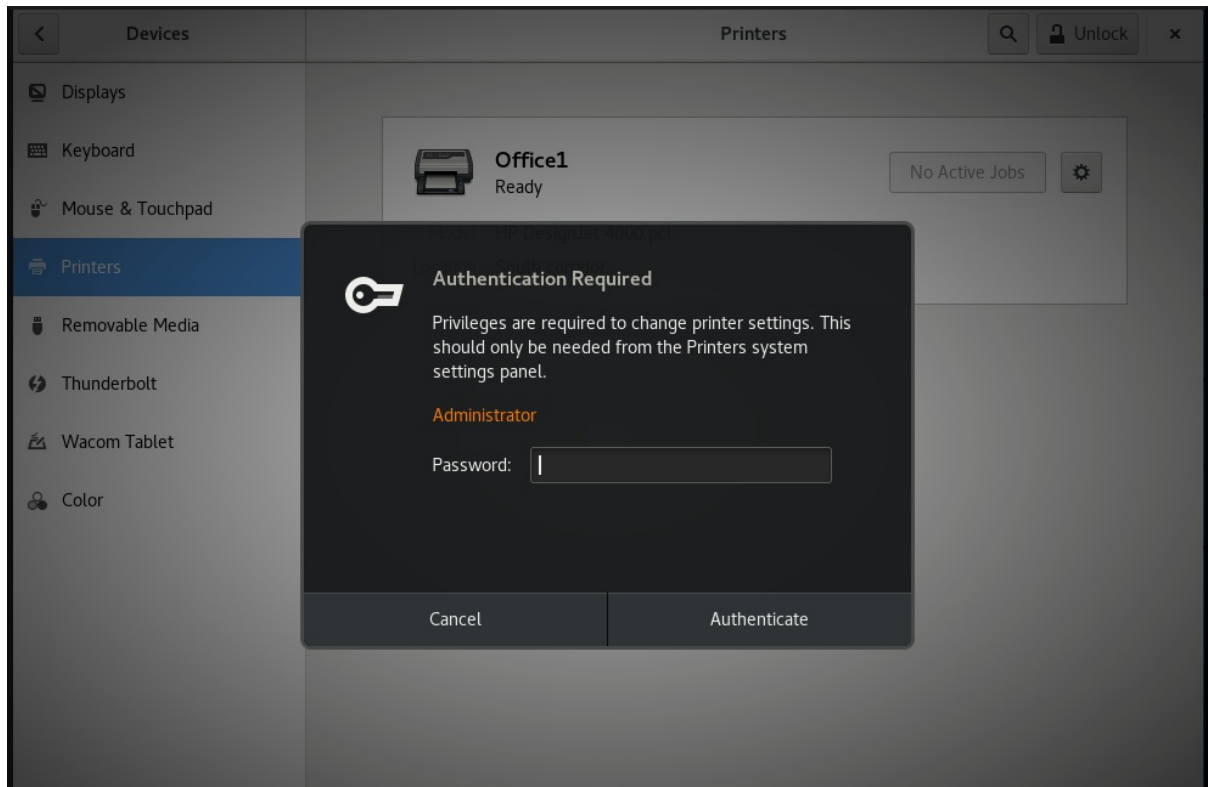


### 1.8.2. Adding a new printer in Settings

You can add a new printer using the **Settings** application.

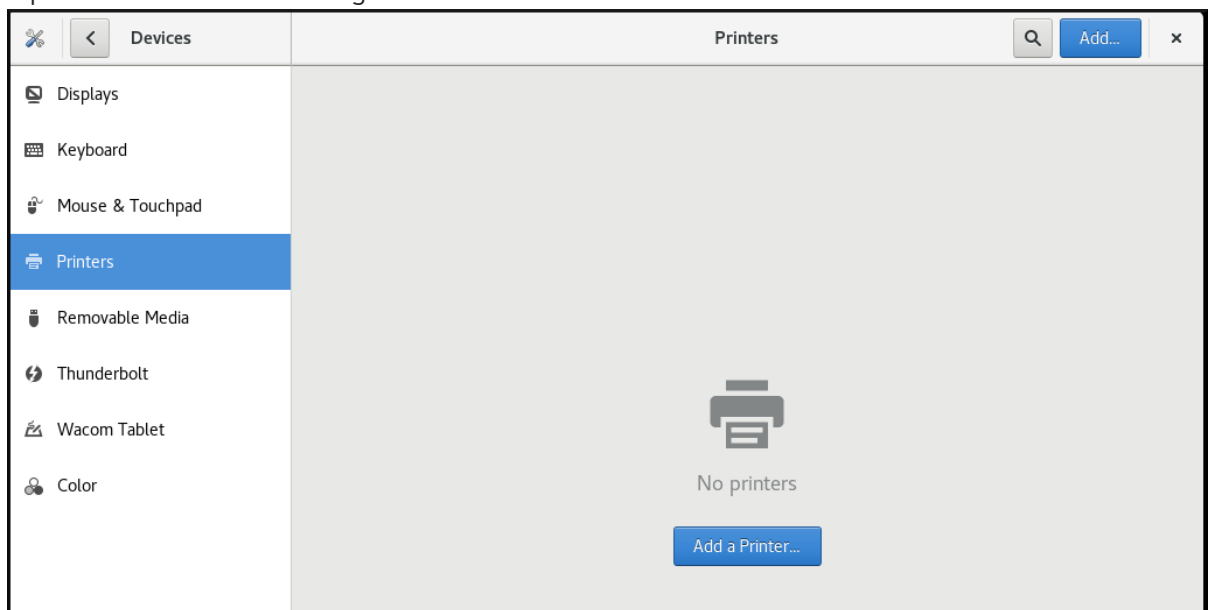
## Prerequisites

- Click the **Unlock** button, which appears near the upper-right corner of the **Printers** screen, and authenticate as one of the following users:
  - Superuser
  - Any user with the administrative access provided by **sudo** (users listed within **/etc/sudoers**)
  - Any user belonging to the **printadmin** group in **/etc/group**

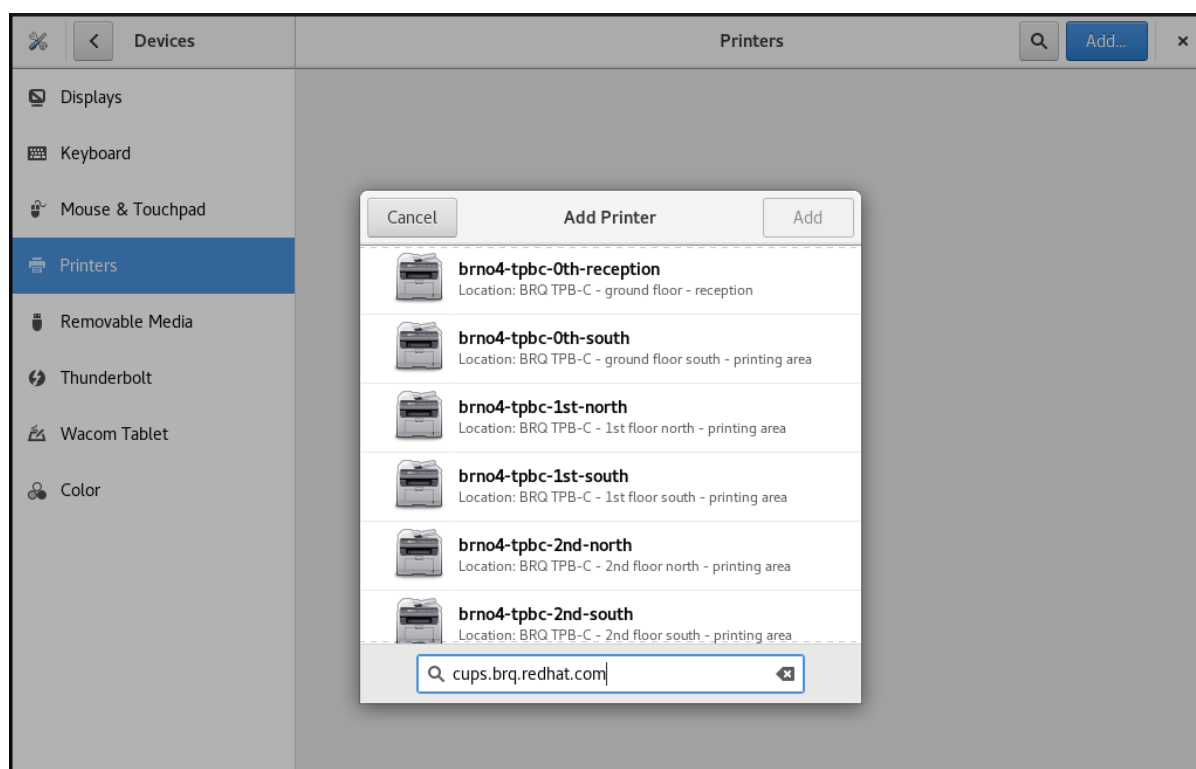
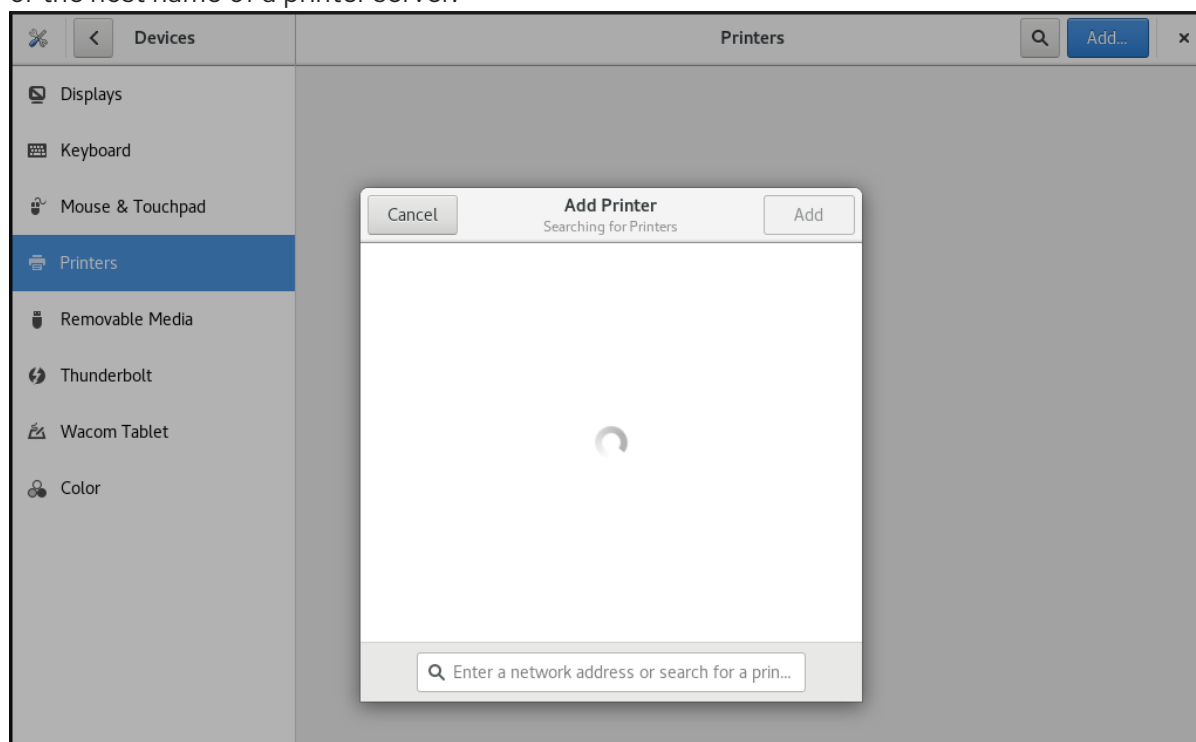


## Procedure

1. Open the **Add Printer** dialog.



2. Select one of the available printers (including also network printers), or enter printer IP address or the host name of a printer server.



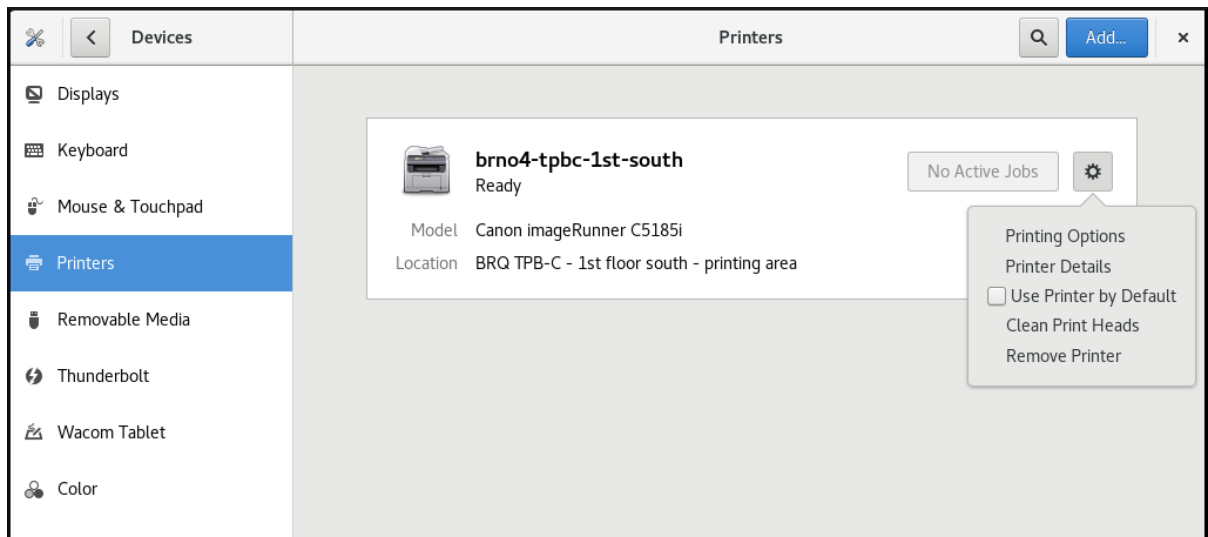
### 1.8.3. Configuring a printer in the Settings application

You can configure a new printer and maintain a configuration of a printer using the **Settings** application.

#### Displaying printer's settings menu

##### Procedure

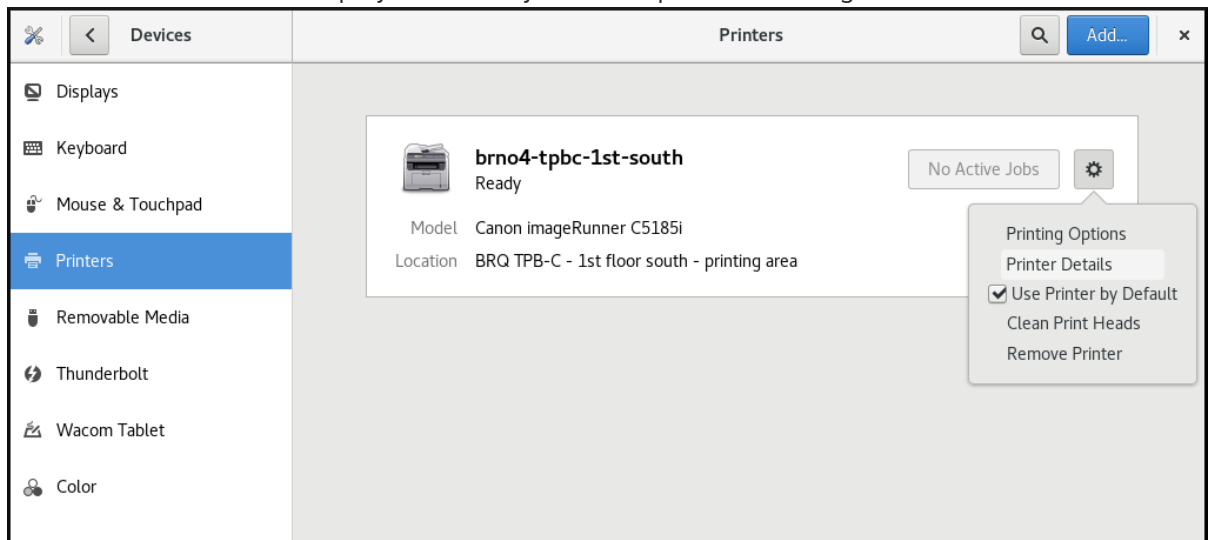
- Click the settings (⚙️) button on the right to display a settings menu for the selected printer:



## Displaying and modifying printer's details

### Procedure

- Click **Printer Details** to display and modify selected printer's settings:



In this menu, you can select the following actions:

### Search for Drivers

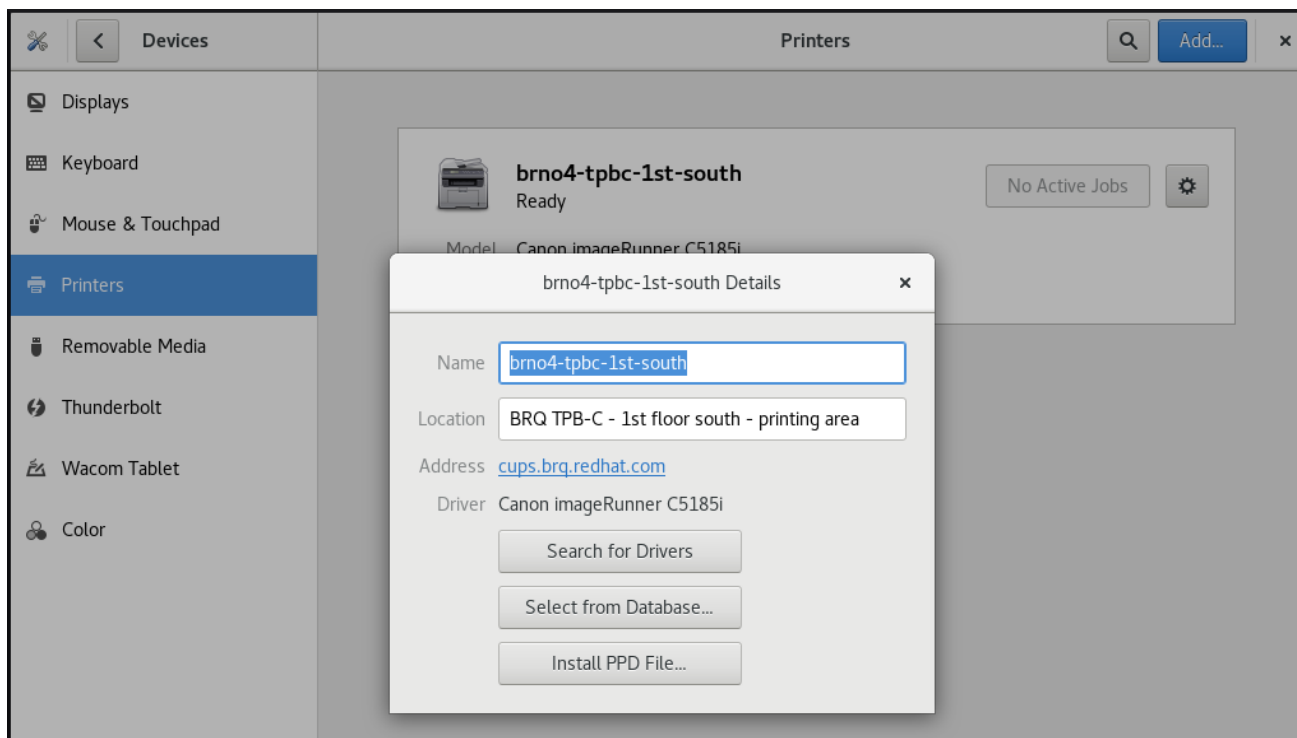
GNOME Control Center communicates with **PackageKit** that searches for a suitable driver suitable in available repositories.

### Select from Database

This option enables you to select a suitable driver from databases that have already been installed on the system.

### Install PPD File

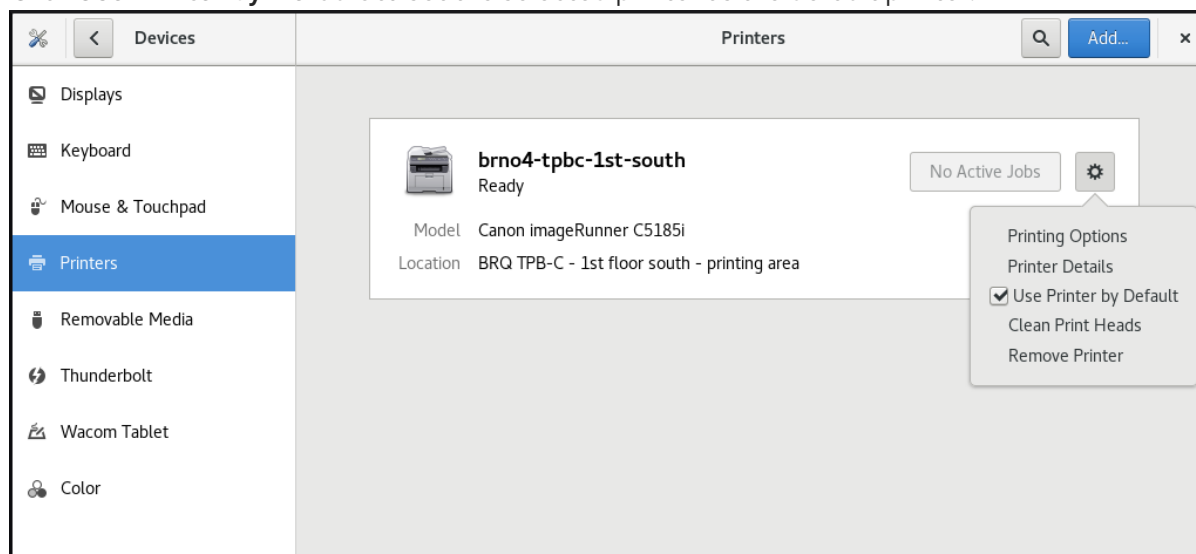
This option enables you to select from a list of available postscript printer description (PPD) files that can be used as a driver for your printer.



## Setting the default printer

### Procedure

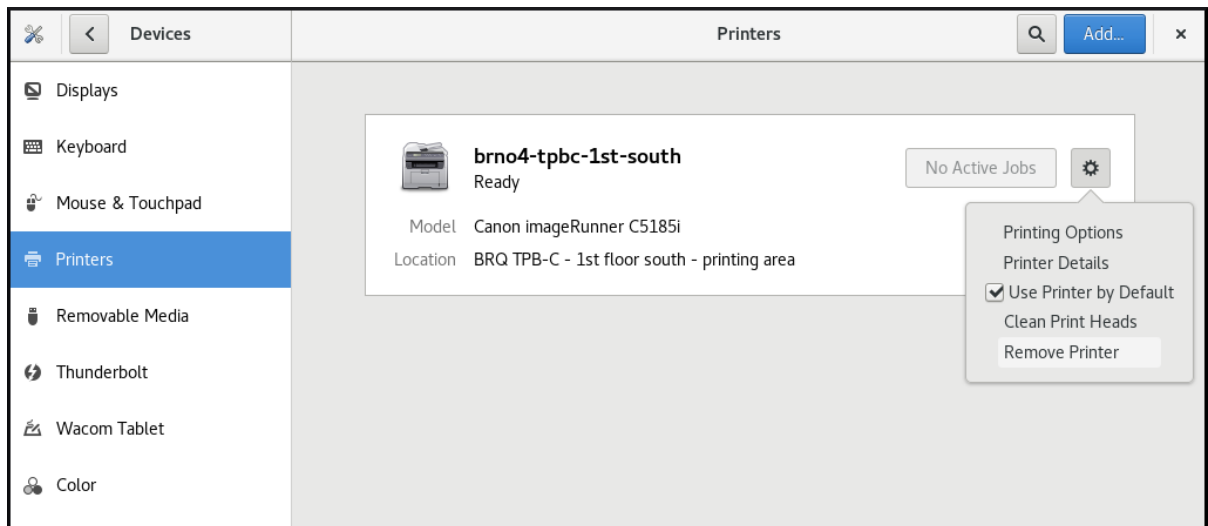
- Click **Use Printer by Default** to set the selected printer as the default printer:



## Removing a printer

### Procedure

- Click **Remove Printer** to remove the selected printer:



### 1.8.4. Printing a test page in the Setting application

You can print a test page to make sure that the printer functions properly.

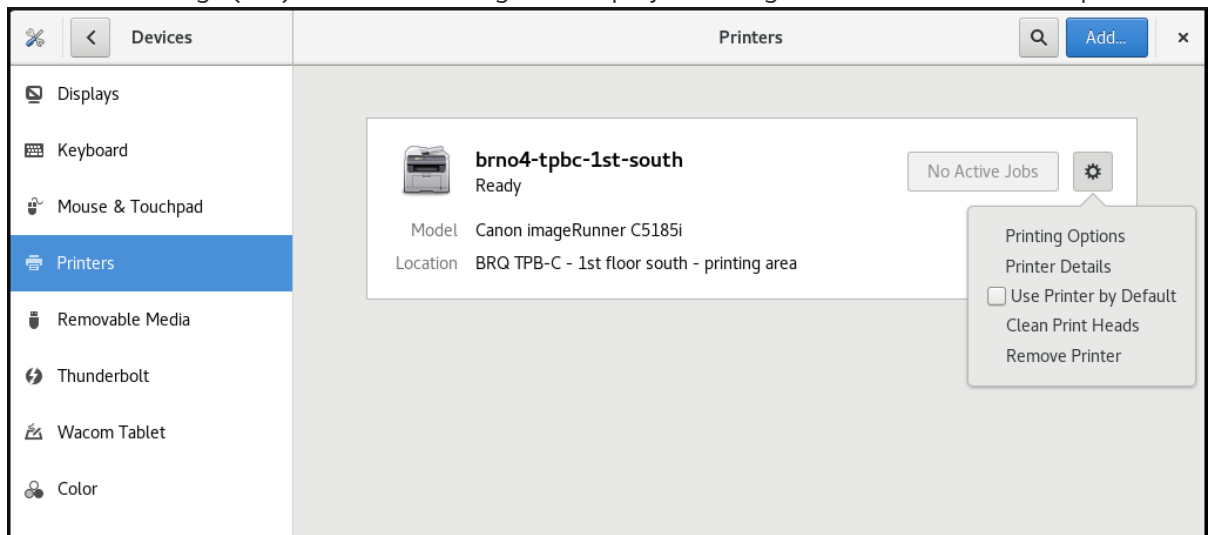
You might want to print a test page if one of the below prerequisites is met.

#### Prerequisites

- A printer has been set up.
- A printer configuration has been changed.

#### Procedure

1. Click the settings (⚙️) button on the right to display a settings menu for the selected printer:



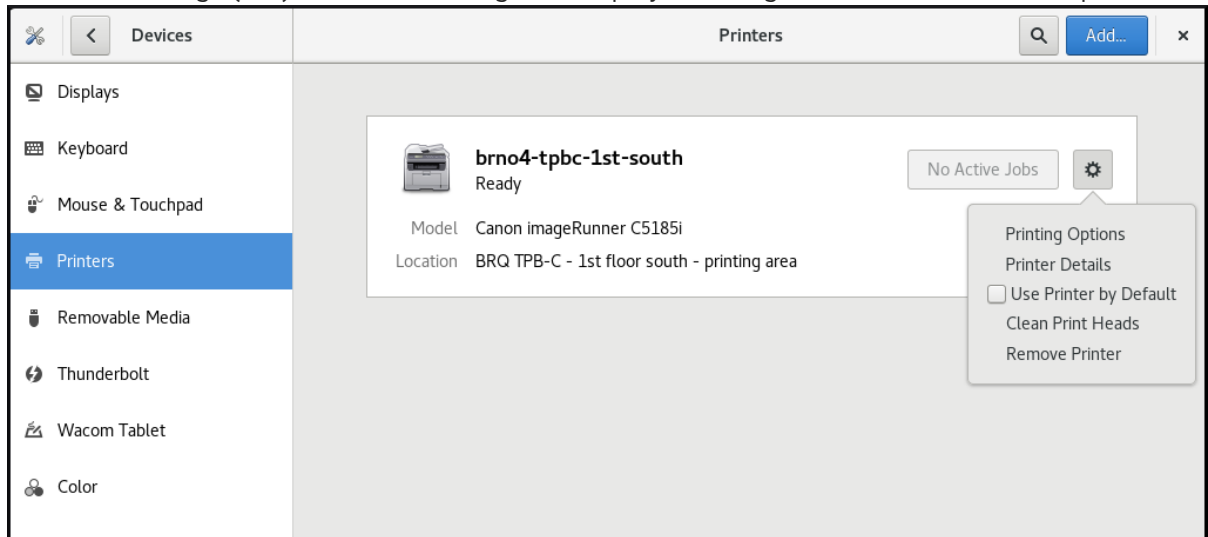
2. Click **Printing Options** → **Test Page**.

### 1.8.5. Setting print options using the Settings application

You can set print options using the **Settings** application.

#### Procedure

1. Click the settings (⚙️) button on the right to display a settings menu for the selected printer:



2. Click **Printing Options**.

## 1.9. SHARING MEDIA BETWEEN APPLICATIONS

Red Hat Enterprise Linux 8 includes the PipeWire media server, which ensures access to multimedia devices and media sharing between applications.

When running a remote desktop session on **GNOME Shell on Wayland**, PipeWire and the VNC server is used. The functionality of remote desktop session is provided by the **gnome-remote-desktop** and **pipewire** packages.

On **X.Org**, just VNC is needed to run a remote desktop session. This functionality on **X.Org** is provided by the **vin** package.

PipeWire is used also with teleconference tools such as BlueJeans when running on **GNOME Shell on Wayland**. In such case, the **pipewire service** is activated automatically when you start sharing your screen within the teleconference tool.

To check the status of the **pipewire service**, run:

```
~]$ systemctl --user status pipewire
```

## 1.10. CUSTOMIZING GNOME SHELL ENVIRONMENT WITH THE TWEAKS TOOL

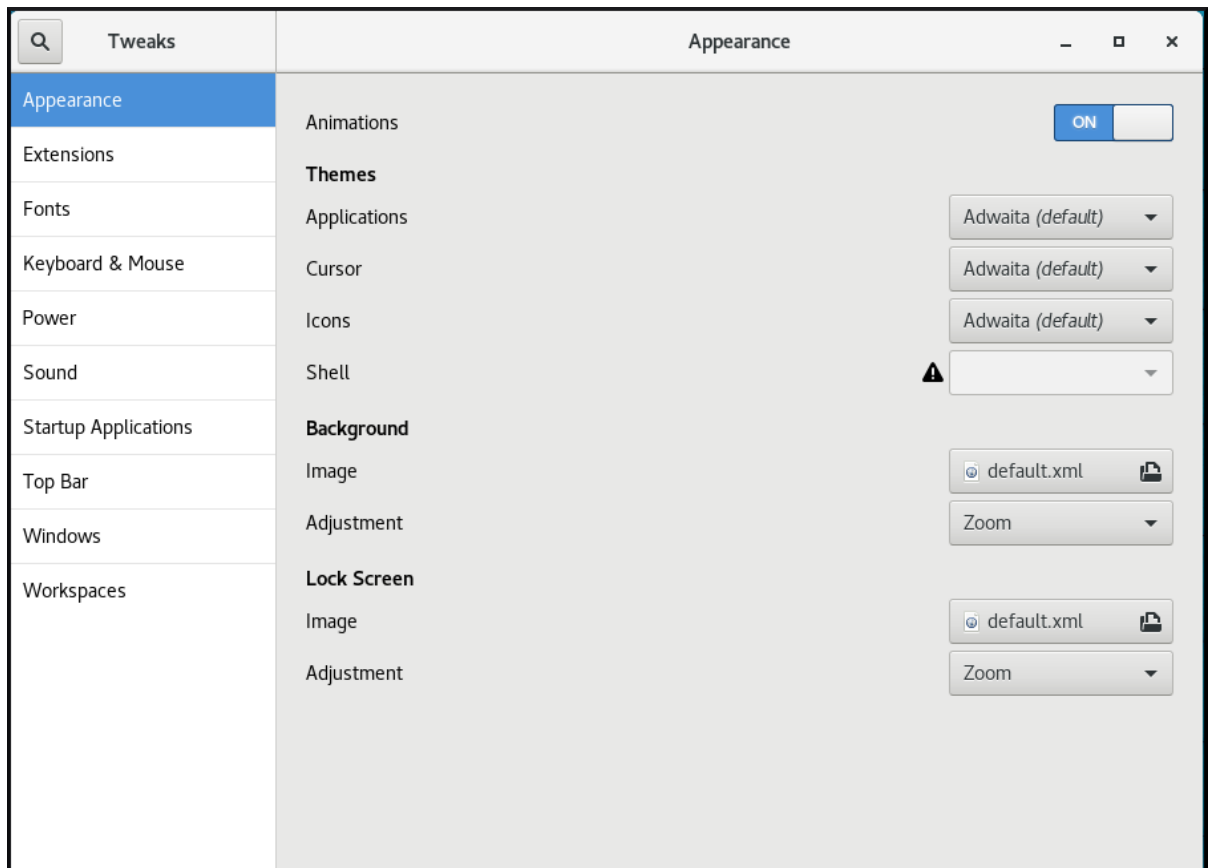
You can customize the GNOME Shell environment for a particular user by using the **Tweaks** tool.

### Procedure

1. Open **Tweaks**.

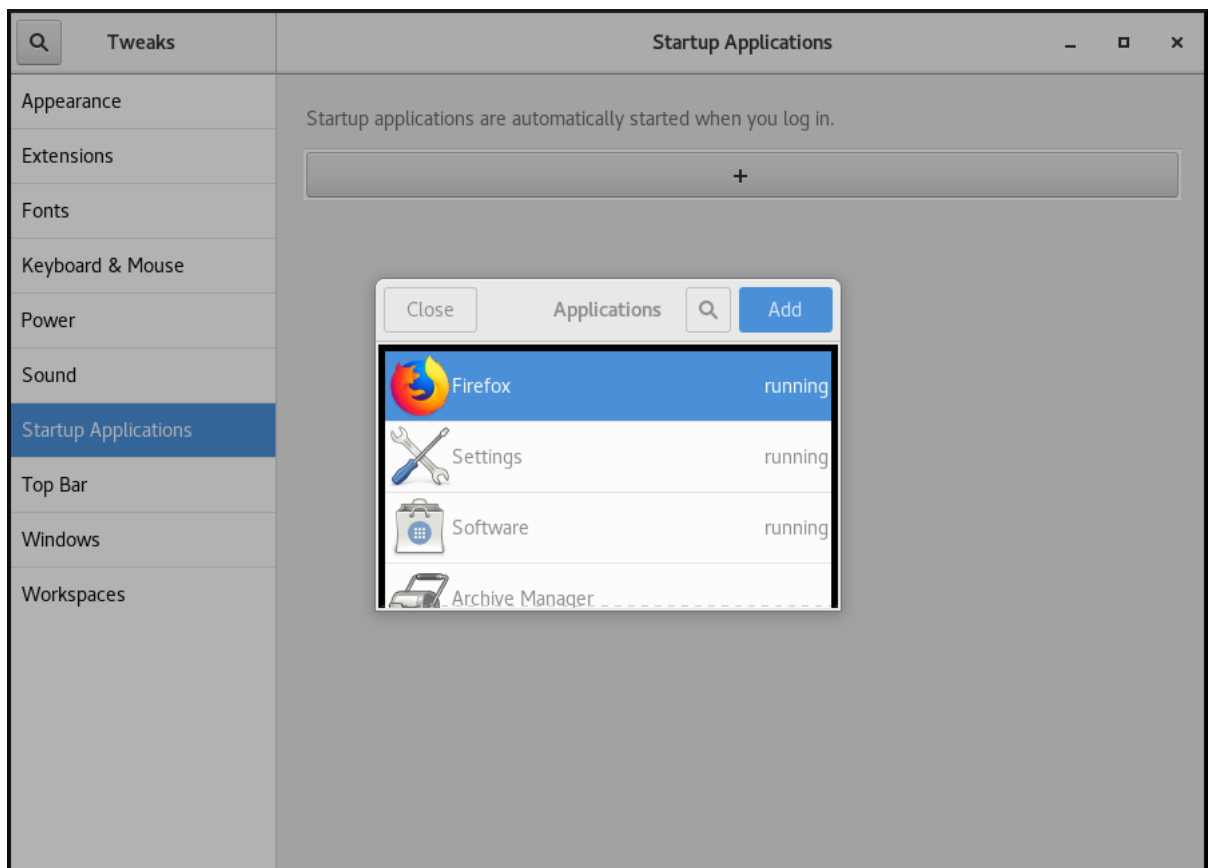
### The Tweaks tool



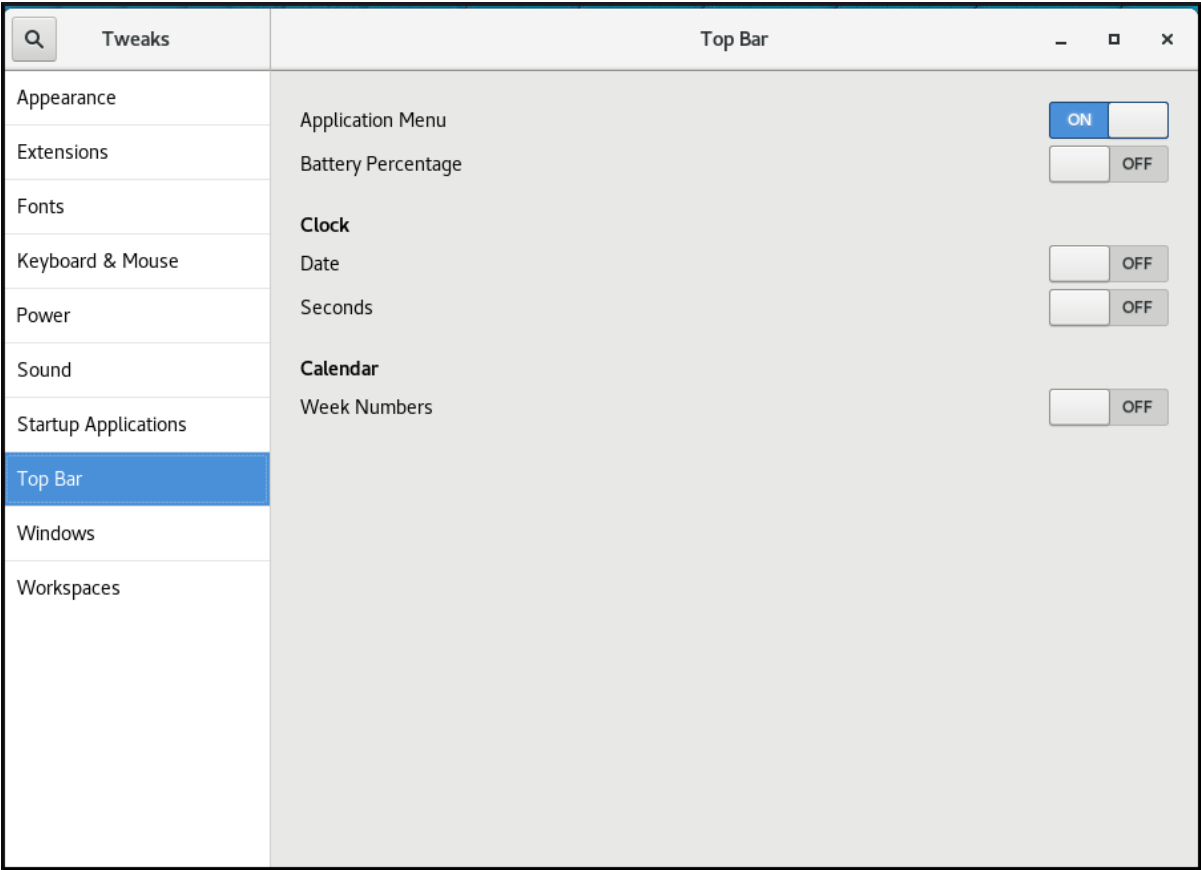


2. Select the item that you want to customize in the left side bar.  
For example, you can set applications to start automatically when you log in by using the **Startup Applications** menu, or you can customize your top bar appearance by using the **Top Bar** menu.

### Customizing startup applications in Tweaks



Customizing the appearance of your top bar in Tweaks



## CHAPTER 2. OVERVIEW OF GNOME ENVIRONMENTS

You can switch between several user interfaces in GNOME. Additionally, GNOME can run on several different graphics back ends.

### 2.1. GNOME ENVIRONMENTS, BACK ENDS, AND DISPLAY PROTOCOLS

GNOME 3 provides two user environments:

- GNOME Standard
- GNOME Classic

Both environments can use two different protocols as their graphical back ends:

- The **X11** protocol, which uses **X.Org** as the display server.
- The **Wayland** protocol, which uses **GNOME Shell** as the **Wayland** compositor and display server.

This solution of display server is further referred as **GNOME Shell on Wayland**

The default combination in RHEL 8 is GNOME Standard environment using **GNOME Shell on Wayland** as the display server. However, due to certain **Wayland** limitations, you might want to switch the graphics protocol stack to **X11**. You may also want to switch from GNOME Standard to GNOME Classic.

#### Additional resources

- For more information about graphics based on the **Wayland** protocol, see [Section 2.6, “Key differences between the Wayland and X11 protocol”](#).
- For information about how to switch the environments, see [Section 2.8, “Selecting GNOME environment and display protocol”](#).

### 2.2. GNOME STANDARD

GNOME Standard user interface includes these major components:

#### Top bar

The horizontal bar at the top of the screen provides access to some of the basic functions of GNOME Standard, such as the **Activities Overview**, clock and calendar, system status icons, and the **system menu**.

#### System menu

The **system menu** is located in the top right corner, and enables you:

- Updating settings
- Controlling the sound bar
- Finding information about your Wi-Fi connection
- Switching user
- Logging out

- Turning off the computer

## Activities Overview

The **Activities Overview** features windows and applications views that let the user run applications and windows and switch between them.

The **search entry** at the top allows for searching various items available on the desktop, including applications, documents, files, and configuration tools.

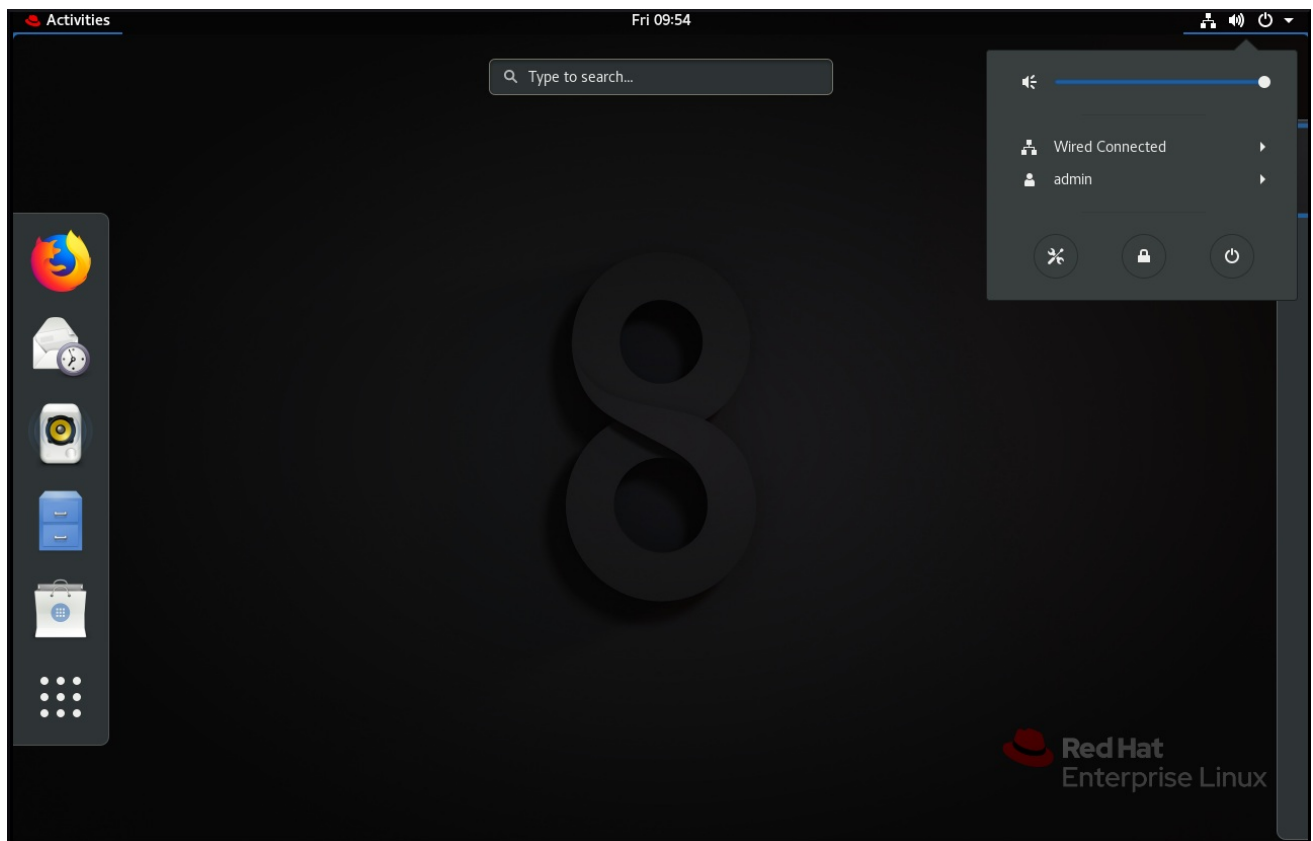
The vertical bar on the left side contains a list of favorite and running applications. You can add or remove applications from the default list of favorites according to your needs.

The workspace list displayed on the right side allows the user to switch between multiple workspaces, or move applications and windows from one workspace to another.

## Message tray

The **message tray** provides access to pending notifications. The **message tray** shows when the user presses **Super+M**.

## The GNOME 3 Standard Desktop



## 2.3. GNOME CLASSIC

GNOME Classic represents a mode for users who prefer a more traditional desktop experience that is similar to GNOME 2 environment used with Red Hat Enterprise Linux 6. It is based on GNOME 3 technologies, and at the same time it includes multiple features similar to GNOME 2.

GNOME Classic user interface consists of these major components:

### Applications and Places

The **Applications** menu is displayed at the top left of the screen. It gives the user access to applications organized into categories. If you enable window overview, you can also open the **Activities Overview** from that menu.

The **Places** menu is displayed next to the **Applications** menu on the top bar. It gives the user quick access to important folders, for example **Downloads** or **Pictures**.

## Taskbar

The **taskbar** is displayed at the bottom of the screen, and features:

- A window list
- A notification icon displayed next to the window list
- A short identifier for the current workspace and total number of available workspaces displayed next to the notification icon

## Four available workspaces

In GNOME Classic, the number of workspaces available to the user is by default set to 4.

## Minimize and maximize buttons

Window titlebars in GNOME Classic feature the minimize and maximize buttons that let the user quickly minimize the windows to the window list, or maximize them to take up all of the space on the desktop.

## A traditional **Super+Tab** window switcher

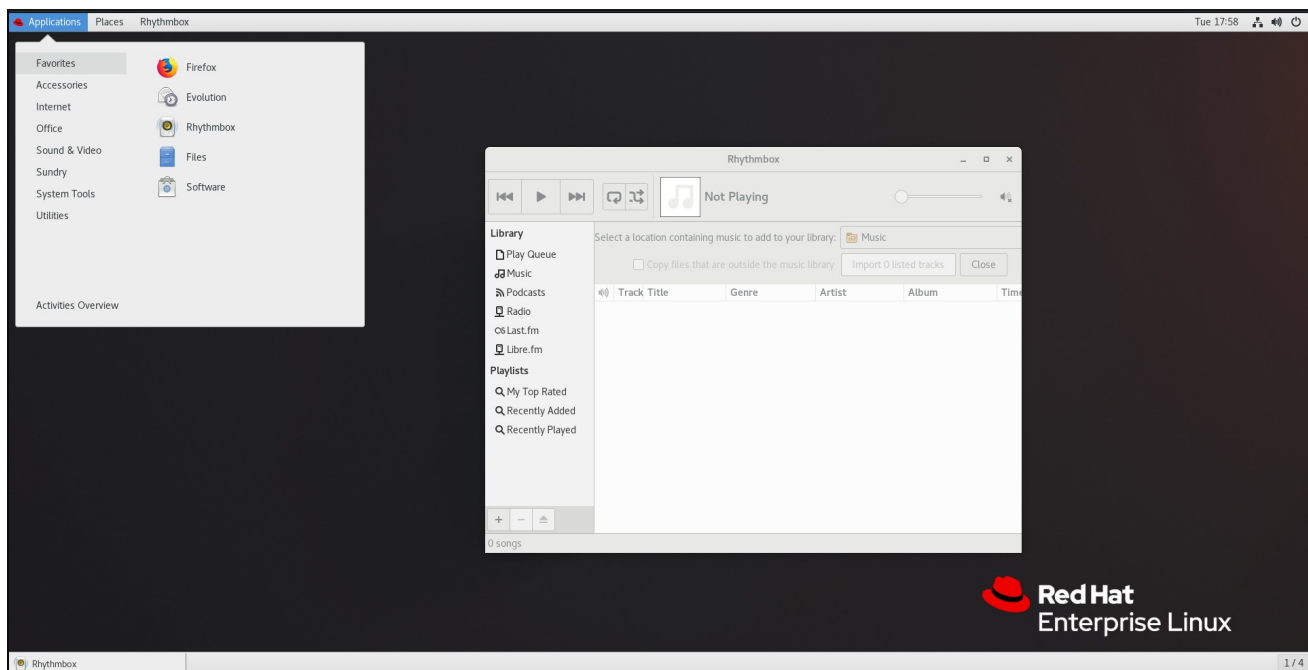
In GNOME Classic, windows in the **Super+Tab** window switcher are not grouped by application.

## System menu

The **system menu** is located in the top right corner, and enables the following actions:

- Updating settings
- Controlling the sound bar
- Finding information about your Wi-Fi connection
- Switching user
- Logging out
- Turning off the computer

## The GNOME 3 Classic desktop with the Rhythmbox application and the Favorites submenu of the Applications menu



In GNOME Classic, the overview of windows that are open is not available by default. You can see the list of all open windows in the **taskbar** at the bottom of the screen. However, you can enable the windows overview similar to what is by default available in GNOME Standard by changing the default settings of the GNOME Classic environment as described in [Section 2.4, “Enabling window overview in GNOME Classic”](#).

## 2.4. ENABLING WINDOW OVERVIEW IN GNOME CLASSIC

In GNOME Classic, the overview of open windows is not available by default. This procedure enables the window overview for all users on the system.



### IMPORTANT

Enabling the window overview by this procedure is not a permanent change. Each update of the **gnome-classic-session** package overwrites the configuration file to the default settings, which disable the window overview.

To keep the window overview enabled, apply the procedure after each update of **gnome-classic-session**.

### Procedure

1. Open the `/usr/share/gnome-shell/modes/classic.json` file as the **root** user.
2. Find the following line in the file:  

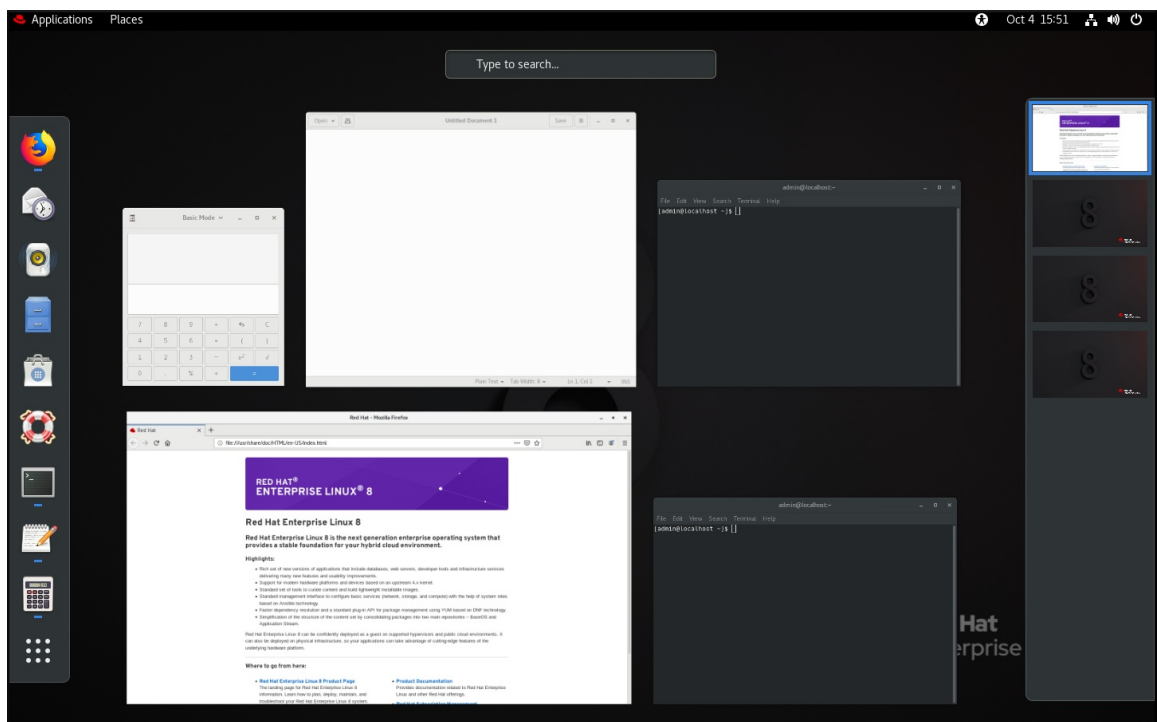
```
"hasOverview": false
```
3. Change the line to the following:  

```
"hasOverview": true
```
4. Save changes, and close the `/usr/share/gnome-shell/modes/classic.json` file.
5. Restart the user session.

## Verification steps

1. In your GNOME Classic session, open multiple windows.
2. Press the **Super** key to open the window overview.
3. In the overview, check that:
  - The **Dash** (the vertical panel on the left side of the screen) is displayed.
  - The bottom panel is not displayed.
  - The workspace switcher is displayed on the right side of the screen.

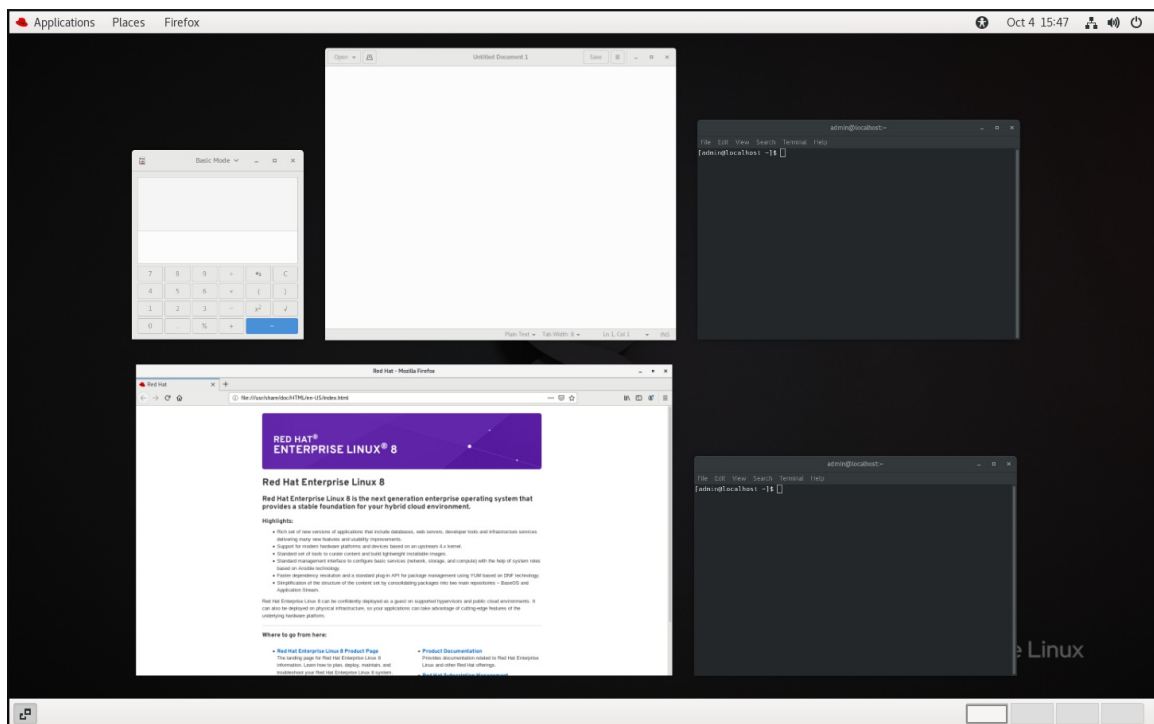
## Window overview with "hasOverview": true



With the default settings ("**hasOverview**": **false**), the overview has the following features:

- The **Dash** is not displayed.
- The bottom panel is displayed. It includes the **Window picker** button in its left part and the workspace switcher in its right part.

## Window overview with "hasOverview": false



## 2.5. GRAPHICS BACK ENDS IN RHEL 8

In RHEL 8, you can choose between two protocols to build a graphical user interface:

### X11

The **X11** protocol uses **X.Org** as the display server. Displaying graphics based on this protocol works the same way as in RHEL 7, where this was the only option.

### Wayland

The **Wayland** protocol on RHEL 8 uses **GNOME Shell** as its compositor and display server, which is further referred as **GNOME Shell on Wayland**. Displaying graphics based on the **Wayland** protocol has some differences and limitation compared to **X11**.

New installations of RHEL 8 automatically select **GNOME Shell on Wayland**. However, you can switch to **X.Org**, or select the required combination of GNOME environment and display server as described in [Section 2.8, “Selecting GNOME environment and display protocol”](#).

Note that there are also a few environments where **X.Org** is preferred over **GNOME Shell on Wayland** such as:

- Cirrus graphics used in a VM environment
- Matrox graphics
- Aspeed graphics
- QXL graphics used in a VM environment
- Nvidia graphics when used with the proprietary driver





## IMPORTANT

The Nvidia graphics by default use **nouveau**, which is an open source driver. **nouveau** is supported on **Wayland**, hence you can use Nvidia graphics with **nouveau** on **GNOME Shell on Wayland** without any limitations. However, using Nvidia graphics with proprietary Nvidia binary drivers is not supported on **GNOME Shell on Wayland**. In this case, you need to switch to **X.Org** as described in [Section 2.8, “Selecting GNOME environment and display protocol”](#).

### Additional resources

- You can find the current list of environments for which Wayland is not available in the `/usr/lib/udev/rules.d/61-gdm.rules` file.
- For additional information about the **Wayland** project, see [Wayland documentation](#).

## 2.6. KEY DIFFERENCES BETWEEN THE WAYLAND AND X11 PROTOCOL

### X11 applications

Client applications need to be ported to the **Wayland** protocol or use a graphical toolkit that has a **Wayland** backend, such as GTK, to be able to work natively with the compositor and display server based on **Wayland**.

Legacy **X11** applications that cannot be ported to **Wayland** automatically use **Xwayland** as a proxy between the **X11** legacy clients and the **Wayland** compositor. **Xwayland** functions both as an **X11** server and a **Wayland** client. The role of **Xwayland** is to translate the **X11** protocol into the **Wayland** protocol and reversely, so that **X11** legacy applications can work with the display server based on **Wayland**.

On **GNOME Shell on Wayland**, **Xwayland** is started automatically at startup, which ensures that most **X11** legacy applications work as expected when using **GNOME Shell on Wayland**. However, the **X11** and **Wayland** protocols are different, and hence some clients relying on **X11**-specific features may behave differently under **Xwayland**. For such specific clients, you can switch to the **X.Org** display server as described in [Section 2.8, “Selecting GNOME environment and display protocol”](#).

### libinput

Red Hat Enterprise Linux 8 uses a new unified input stack, **libinput**, which manages all common device types, such as mice, touchpads, touchscreens, tablets, trackballs and pointing sticks. This unified stack is used both by the **X.Org** and by the **GNOME Shell on Wayland** compositor.

**GNOME Shell on Wayland** uses **libinput** directly for all devices, and no switchable driver support is available. Under **X.Org**, **libinput** is implemented as the **X.Org libinput** driver, and driver support is outlined below.

### Mice, touchscreens, trackballs, pointing sticks

Red Hat Enterprise Linux 8 uses the **X.Org libinput** driver for these devices. The **X.Org evdev** driver, which was used in Red Hat Enterprise Linux 7, is available as fallback where required.

### Touchpads

Red Hat Enterprise Linux 8 uses the **X.Org libinput** driver for touchpads. The **X.Org synaptics** driver, which was used for touchpads in Red Hat Enterprise Linux 7, is no longer available.

### Graphics tablets

Red Hat Enterprise Linux 8 continues using the **X.Org wacom** driver, which was used for tablet devices in Red Hat Enterprise Linux 7. However, the **X.Org libinput** driver is available where required.

### Other input devices

Red Hat Enterprise Linux 7 used the **X.Org evdev** driver for other input devices that are not included in the above categories. Red Hat Enterprise Linux 8 uses the **X.Org libinput** driver by default but can fall back to the **X.Org evdev** driver if a device is incompatible with **libinput**.

### Gestures

**GNOME Shell on Wayland** supports new touchpad and touchscreen gestures. These gestures include:

- Switching workspaces by dragging up or down with four fingers.
- Opening the **Activities** overview by bringing three fingers closer together.

## 2.7. CURRENT WAYLAND LIMITATIONS

### Nvidia drivers

Proprietary Nvidia binary drivers are not supported with **GNOME Shell on Wayland**. To avoid any complications while using the Nvidia GPU, **GNOME Shell** automatically falls back to **X.Org**, which means that the login screen does not provide any option based on the **Wayland** protocol.



#### NOTE

The **nouveau** driver is still supported and is the default driver for Nvidia graphics.

### Remote desktop

With **GNOME Shell on Wayland**, VNC support is provided by the **gnome-remote-desktop** package. Remote access using VNC via **gnome-remote-desktop** currently requires an already logged in session, and only the primary monitor is accessible. Screen sharing with **GNOME Shell on Wayland** is possible using the PipeWire media server. For more details on the PipeWire media server, see [PipeWire project](#).

For more advanced VNC usage, you need to switch to **X.org**, where traditional VNC tools are available. For more information, see [Overview of GNOME environments](#).

### X Display Manager

The X Display Manager Control Protocol (XDMCP) is not supported with **GNOME Shell on Wayland**.

Hence, it is not possible to use the X display manager to start a session on the **X.Org** display server from the same or another computer.

### Additional limitations

The following additional limitations related to the **Wayland** protocol should be noted:

- **X.Org** screen manipulation utilities are not available.
- The **xrandr** utility is not supported because **Wayland** handles layout, rotations, and resolutions differently.
- **GNOME Shell** cannot be restarted using the **Alt+F2 r** shortcut.

- Due to stability issues, using **X.org** instead of **Wayland** is recommended in virtual environments. The graphics based on the **Wayland** protocol are not available for virtual machines that use the **qxl** driver.
- **Wayland** does not support the custom or niche input devices that cannot be handled by the **libinput** driver.

### Additional resources

- You can find the current list of environments for which Wayland-based graphics are unavailable in the `/usr/lib/udev/rules.d/61-gdm.rules` file.

## 2.8. SELECTING GNOME ENVIRONMENT AND DISPLAY PROTOCOL

The default desktop environment for Red Hat Enterprise Linux 8 is GNOME Standard with **GNOME Shell on Wayland** as the display server. However, due to certain limitations of **Wayland**, you might want to switch the graphics protocol stack. You might also want to switch from GNOME Standard to GNOME Classic.

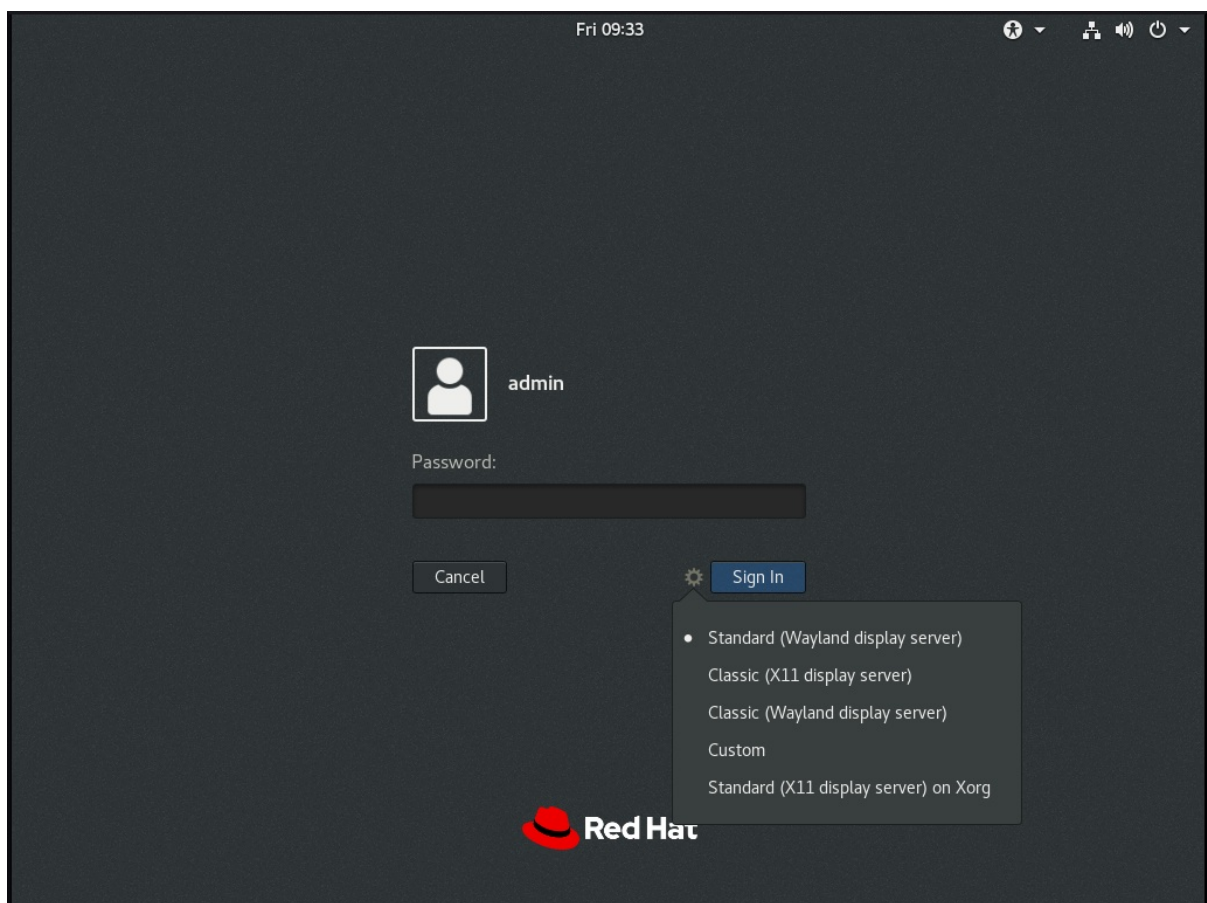
### Procedure

1. From the login screen (GDM), click the gear button next to the **Sign In** button.



### NOTE

You cannot access this option from the lock screen. The login screen appears when you first start Red Hat Enterprise Linux 8 or when you log out of your current session.



2. From the drop-down menu that appears, select the option that you prefer.



### NOTE

Note that in the menu that appears on the login screen, the **X.Org** display server is marked as **X11** display server.



### IMPORTANT

The change of GNOME environment and graphics protocol stack resulting from the above procedure is persistent across user logouts, and also when powering off or rebooting the computer.

## CHAPTER 3. INSTALLING APPLICATIONS IN GNOME

This section describes various approaches that you can use to install a new application in GNOME 3.

### Prerequisites

- Administrator privileges.

### 3.1. THE GNOME SOFTWARE APPLICATION

GNOME Software is an utility that enables you to install and update applications, software components, and GNOME Shell extensions in a graphical interface.

GNOME Software provides a catalog of graphical applications, which are the applications that include a **\*.desktop** file. The available applications are grouped into multiple categories according to their purpose.

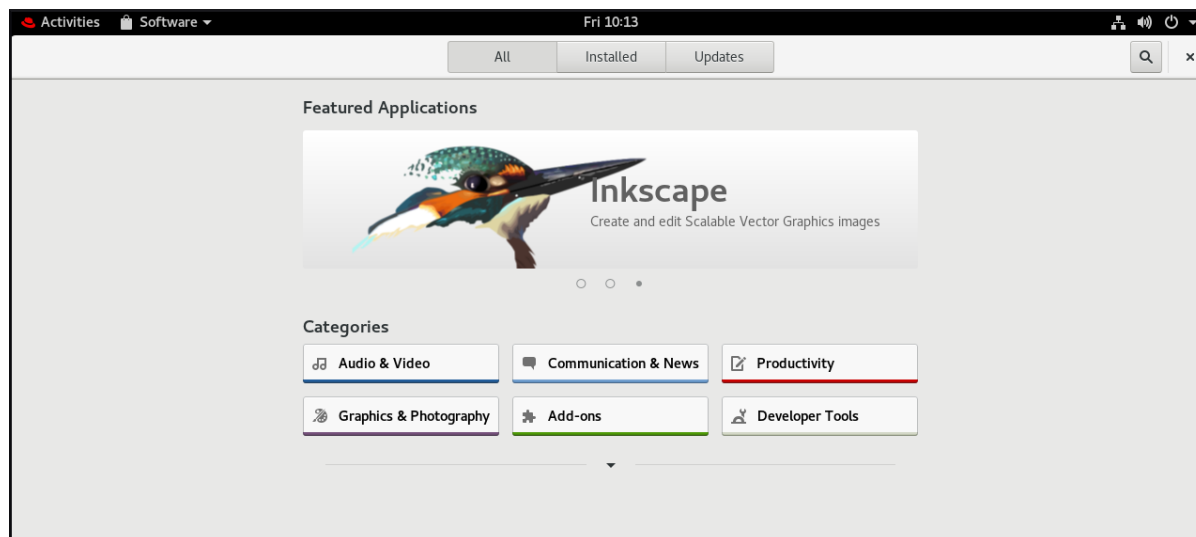
GNOME Software uses the PackageKit and Flatpak technologies as its back ends.

### 3.2. INSTALLING AN APPLICATION USING GNOME SOFTWARE

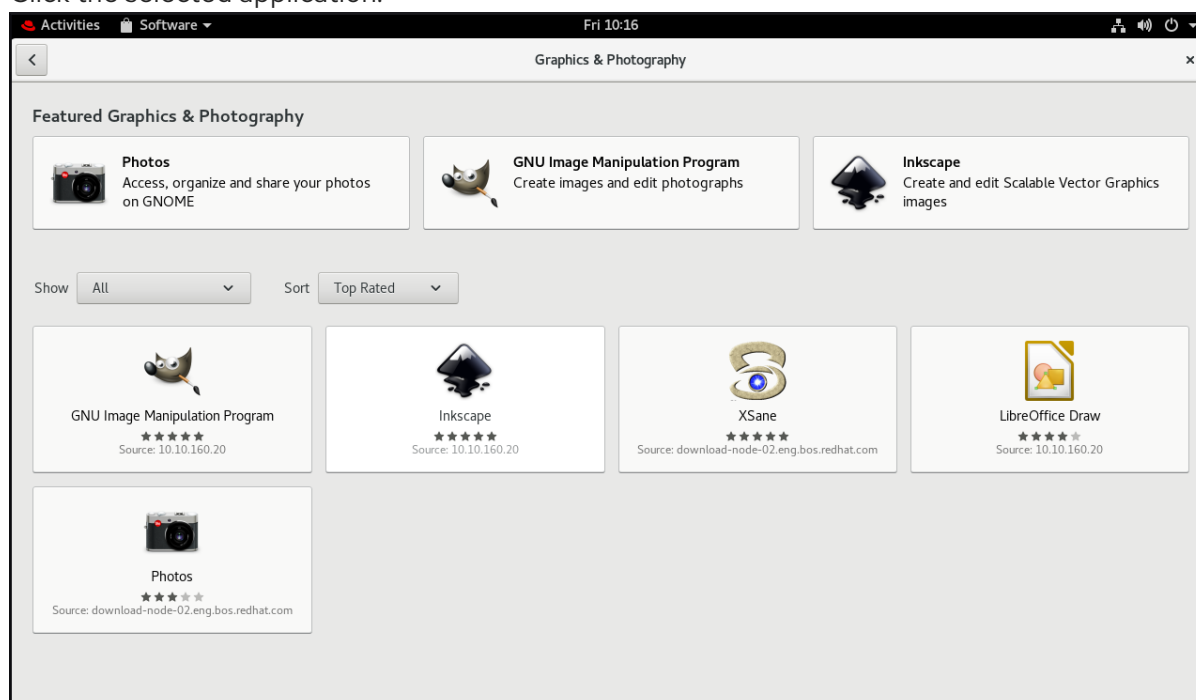
This procedure installs a graphical application using the GNOME Software installer.

#### Procedure

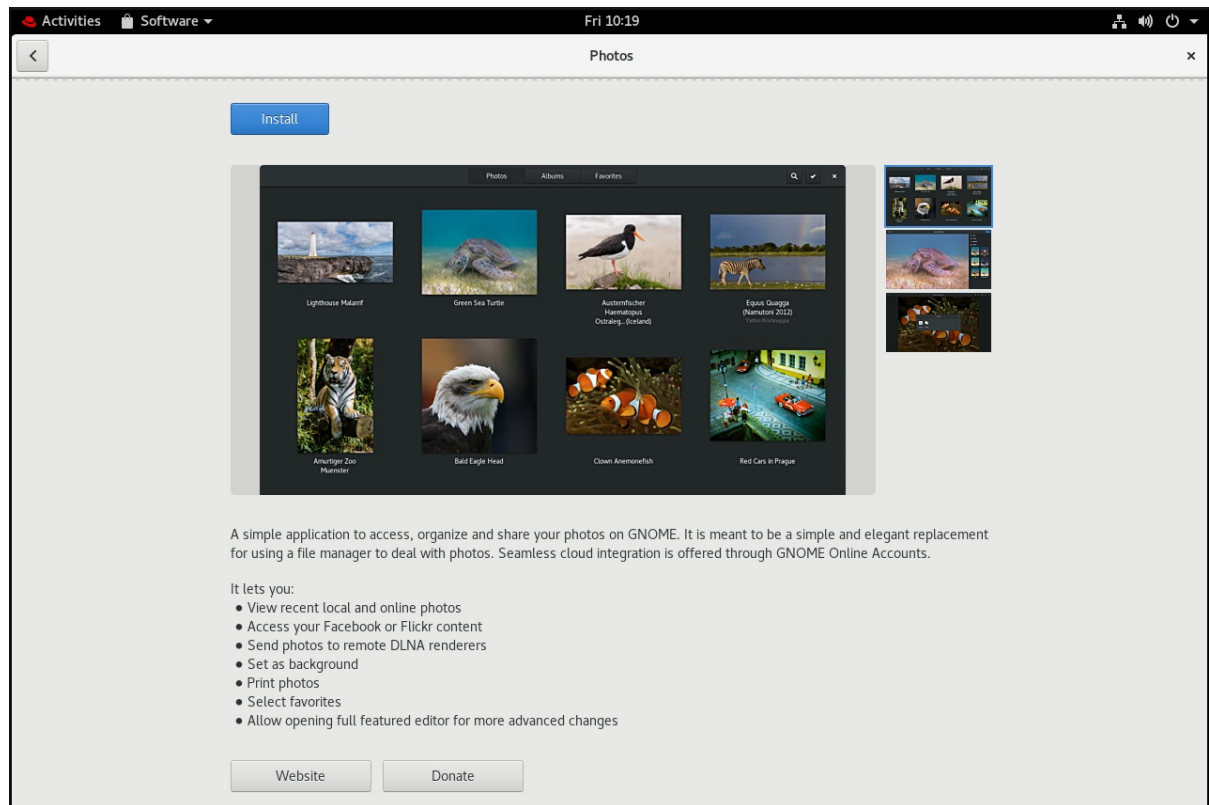
1. Launch the **GNOME Software** application.
2. Find the application to be installed in the available categories:
  - Audio & Video
  - Communication & News
  - Productivity
  - Graphics & Photography
  - Add-ons  
Add-ons include for example GNOME Shell extensions, codecs, or fonts.
  - Developer Tools
  - Utilities



3. Click the selected application.



4. Click the **Install** button.



### 3.3. INSTALLING AN APPLICATION TO OPEN A FILE TYPE

This procedure installs an application that can open a given file type.

#### Prerequisites

- You can access a file of the required file type in your file system.

#### Procedure

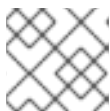
1. Try opening a file that is associated with an application that is currently not installed on your system.
2. GNOME automatically identifies the suitable application that can open the file, and offers to download the application.

### 3.4. INSTALLING AN RPM PACKAGE IN GNOME

This procedure installs an RPM software package that you manually downloaded as a file.

#### Procedure

1. Download the required RPM package.
2. In the **Files** application, open the directory that stores the downloaded RPM package.



#### NOTE

By default, downloaded files are stored in the **/home/user/Downloads/** directory.

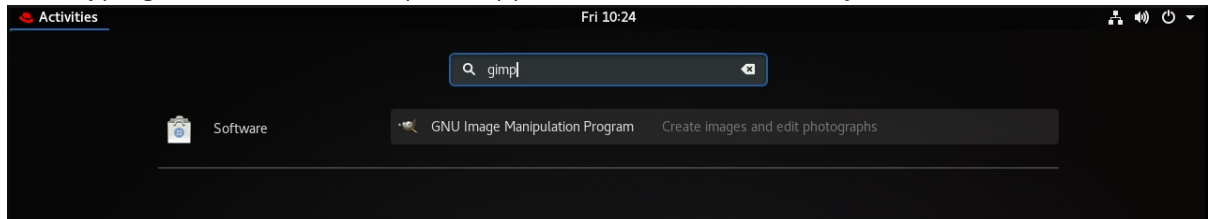
3. Double-click the icon of the RPM package to install it.

### 3.5. INSTALLING AN APPLICATION FROM THE APPLICATION SEARCH IN GNOME

This procedure installs a graphical application that you find in the GNOME application search.

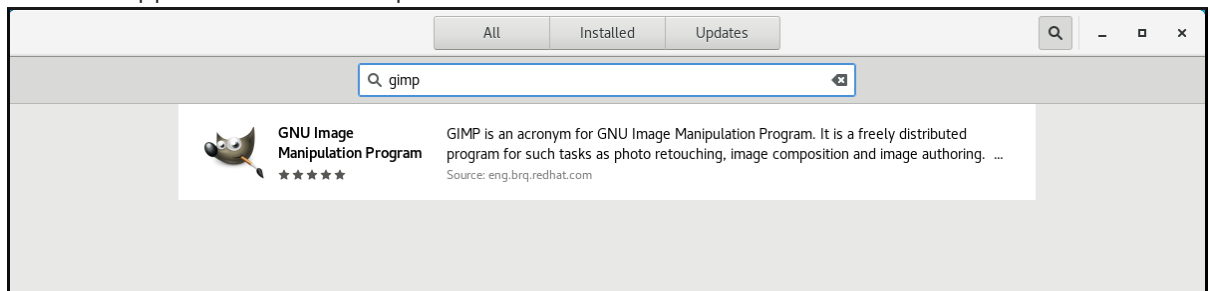
#### Procedure

1. Open the **Activities Overview** screen.
2. Start typing the name of the required application in the search entry.

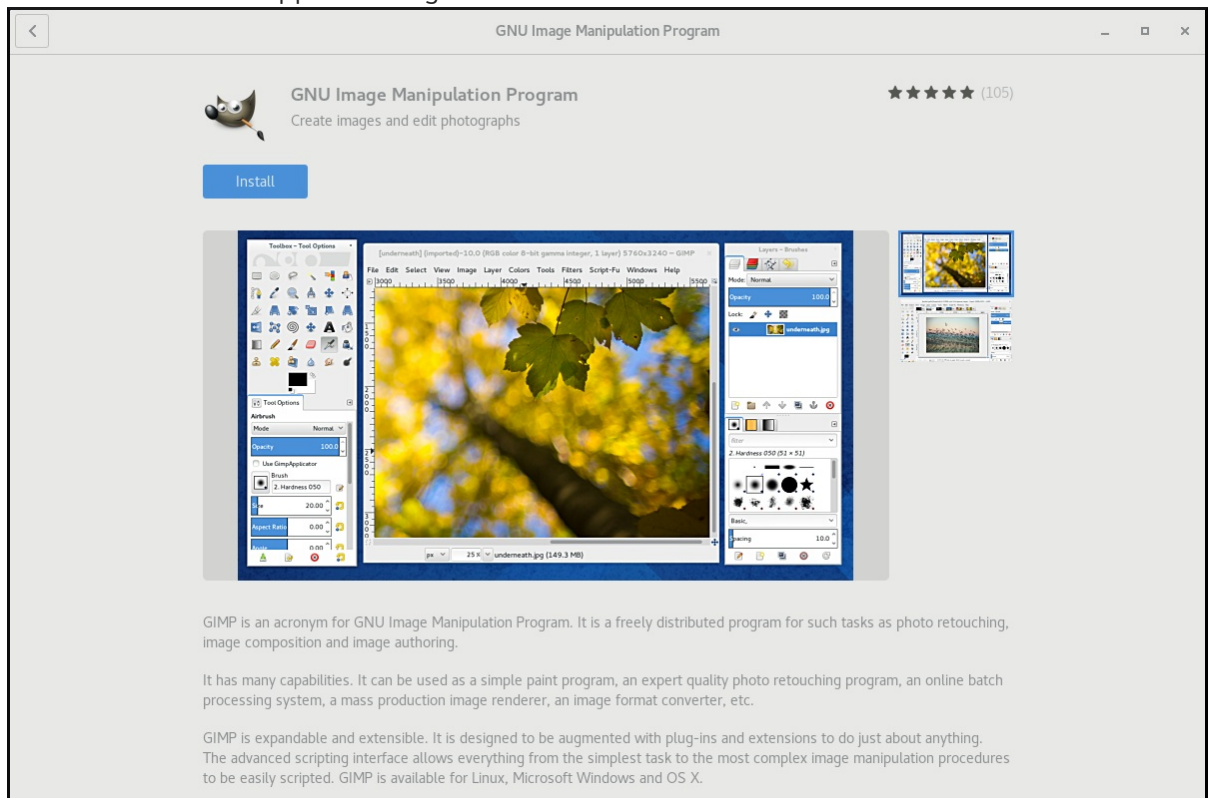


GNOME automatically finds the application in a repository, and displays the application's icon.

3. Click the application's icon to open **GNOME Software**.



4. Click the icon of the application again.





5. Click **Install** to finish the installation in **GNOME Software**.

## 3.6. ADDITIONAL RESOURCES

- For installing software on the command line, see [Installing software with yum](#).

## CHAPTER 4. REGISTERING THE SYSTEM FOR UPDATES USING GNOME

You must register your system to get software updates for your system. This section explains how you can register your system using GNOME.

### Prerequisites

- A valid account with Red Hat customer portal  
See the [Create a Red Hat Login](#) page for new user registration.
- Activation Key or keys, if you are registering the system with activation key
- A registration server, if you are registering system using the registration server

### 4.1. REGISTERING A SYSTEM USING RED HAT ACCOUNT ON GNOME

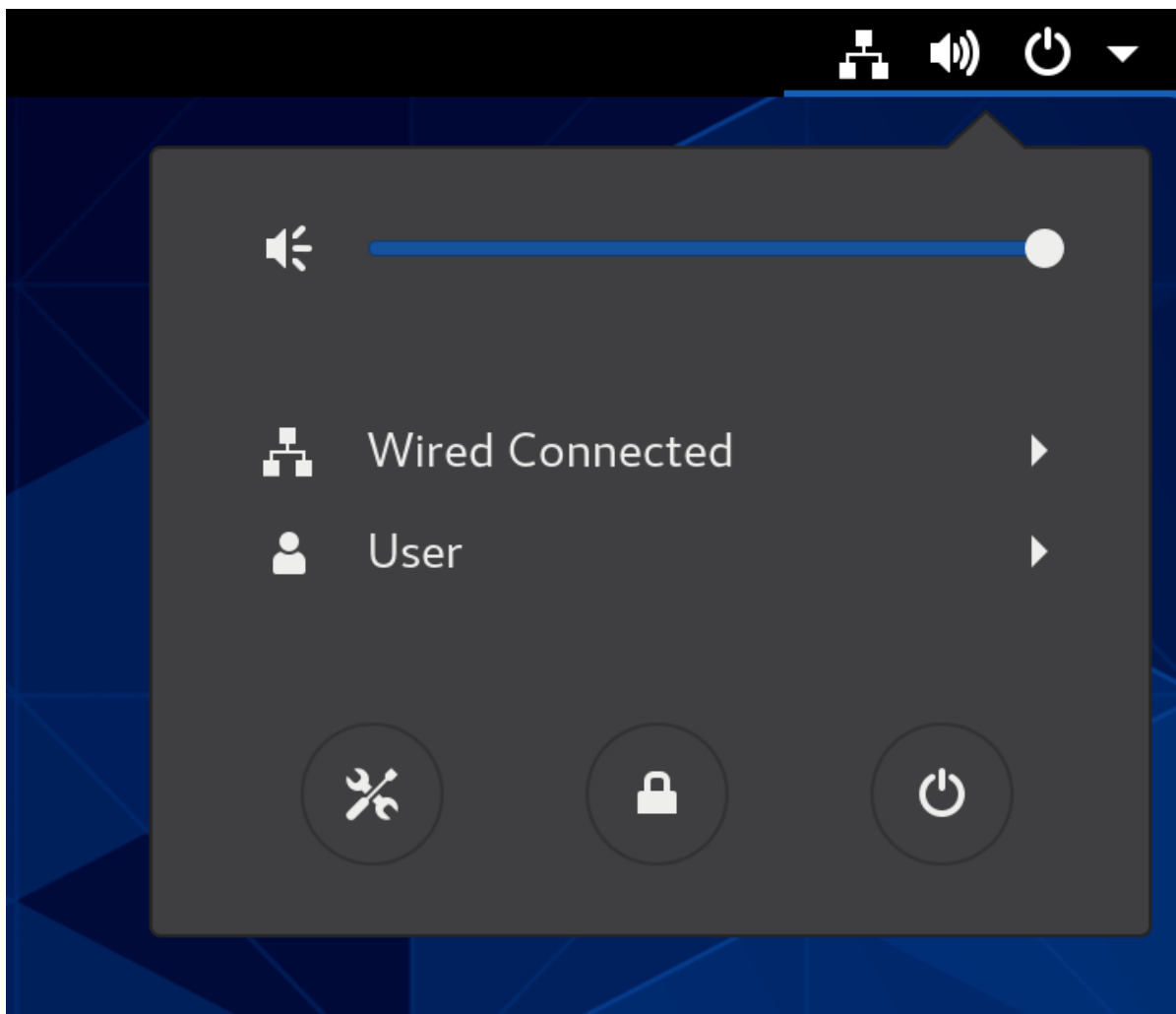
Follow the steps in this procedure to enroll your system with your Red Hat account.

#### Prerequisites

- A valid account on Red Hat customer portal.  
See the [Create a Red Hat Login](#) page for new user registration.

#### Procedure

1. Open the **system menu**, which is accessible from the upper-right screen corner, and click the **Settings** icon.



2. In the **Details** → **About** section, click **Register**.
3. Select **Registration Server**.
4. If you are not using the Red Hat server, enter the server address in the **URL** field.
5. In the **Registration Type** menu, select **Red Hat Account**
6. Under **Registration Details**:
  - Enter your Red Hat account user name in the **Login** field.
  - Enter your Red Hat account password in the **Password** field.
  - Enter the name of your organization in the **Organization** field.
7. Click **Register**.

## 4.2. REGISTERING A SYSTEM USING AN ACTIVATION KEY ON GNOME

Follow the steps in this procedure to register your system with an activation key. You can get the activation key from your organization administrator.

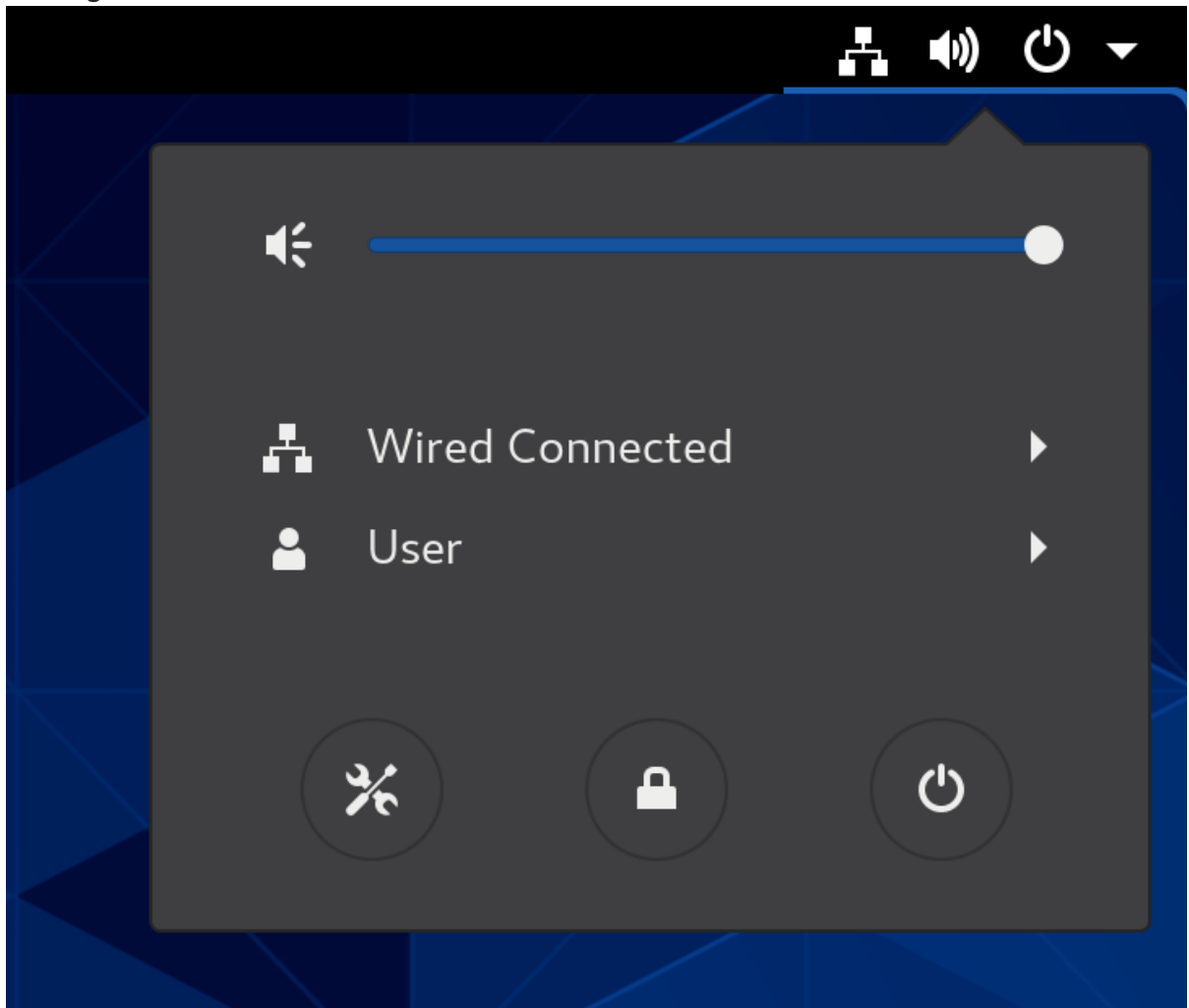
### Prerequisites

- Activation key or keys.

See the [Activation Keys](#) page for creating new activation keys.

### Procedure

1. Open the **system menu**, which is accessible from the upper-right screen corner, and click the **Settings** icon.



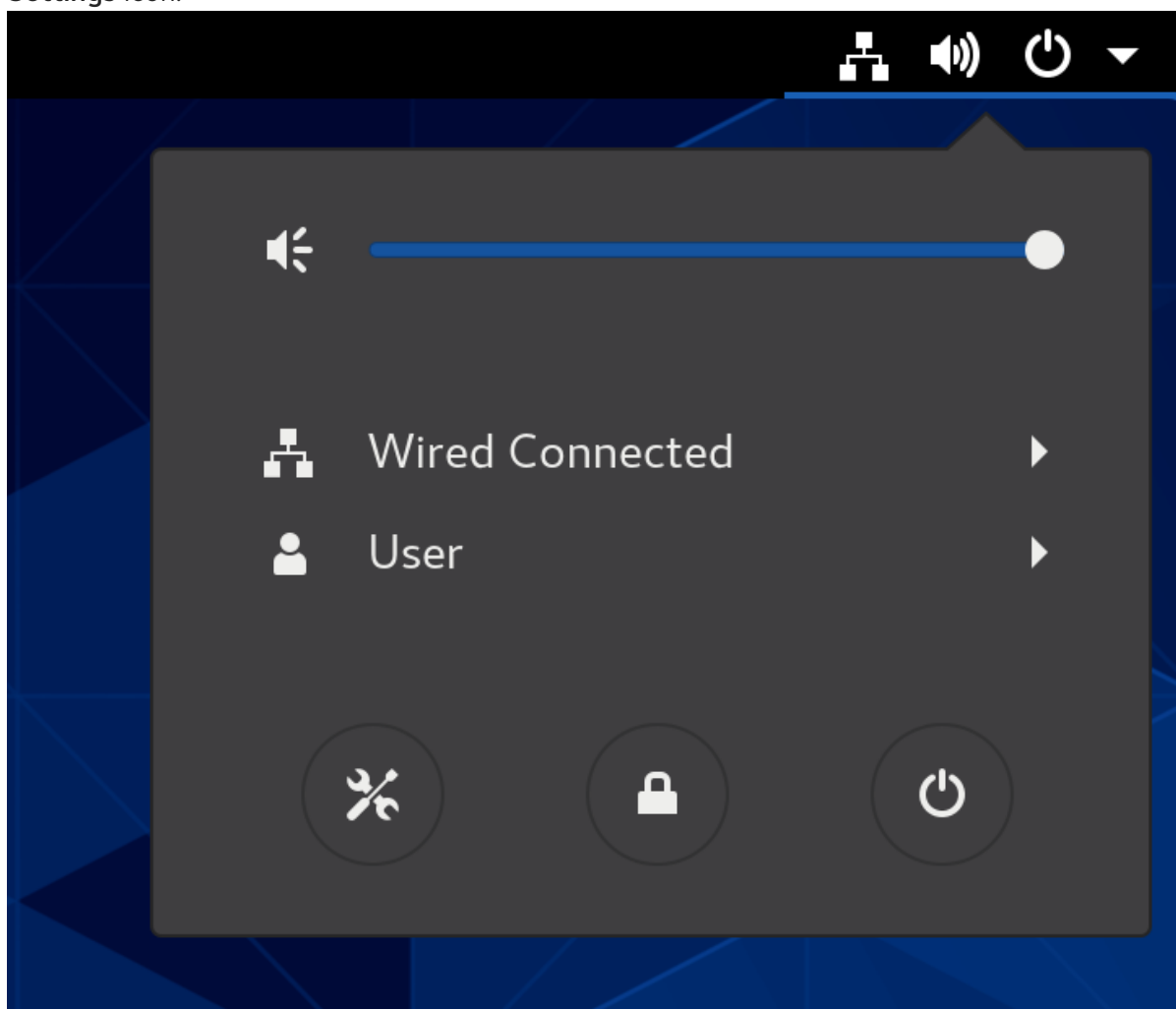
2. In the **Details** → **About** section, click **Register**.
3. Select **Registration Server**.
4. If you are not using the Red Hat server, enter the server address in the **URL** field.
5. In the **Registration Type** menu, select **Activation Keys**.
6. Under **Registration Details**:
  - Enter your activation keys in the **Activation Keys** field. Separate your keys by a comma (,).
  - Enter the name or ID of your organization in the **Organization** field.
7. Click **Register**.

## 4.3. UNREGISTERING THE SYSTEM USING GNOME

Follow the steps in this procedure to unregister your system. After unregistering, your system no longer receives software updates.

### Procedure

1. Open the **system menu**, which is accessible from the upper-right screen corner, and click the **Settings** icon.



2. In the **Details** → **About** section, click **Details**.  
The **Registration Details** screen appears.
3. Click **Unregister**.  
A warning appears about the impact of unregistering the system.
4. Click **Unregister**.

## 4.4. ADDITIONAL RESOURCES

- [Registering the system and managing subscriptions](#)
- [Creating Red Hat Customer Portal Activation Keys](#)
- [Understanding Activation Keys](#)
- [Registering Systems with Activation keys](#)

## CHAPTER 5. ACCESSING THE DESKTOP REMOTELY

You can connect to the desktop on a RHEL server from a remote client.

### 5.1. REMOTE DESKTOP ACCESS OPTIONS

RHEL provides several options for remotely connecting to the desktop. Each option fits a different use case:

#### Single-user access using GNOME tools

This method enables remote access on the client and the server using graphical GNOME applications. It configures a Virtual Network Computing (VNC) session so that only a single user can connect to the desktop on the server at a given time.

Depending on the session type, this method uses different components to implement screen sharing:

- In an X11 session, it uses the **vino** component.
- In a Wayland session, it uses the **gnome-remote-desktop** component.  
This method always uses display number 0. As a consequence, the VNC session always connects to the user who is logged into the server system.

The VNC client application must support **tls\_anon** connections. For example, you can use the **Remote Desktop Viewer (vinagre)** application on Linux systems. Before you can connect from Microsoft Windows clients, such as **RealVNC**, you must disable the VNC encryption on the server.

#### Multi-user access using command-line tools

This method configures a VNC session so that multiple remote clients can connect to the server in parallel. You must first enable the VNC access on the client and the server using command-line tools. Any VNC client application can connect to a server configured using this method.

#### Accessing a single application using X11 forwarding over SSH

This method executes an SSH command on the client that launches an individual graphical on the server. The application window opens on the client.

This method is useful when you do not require a full remote desktop session.

### 5.2. REMOTELY ACCESSING THE DESKTOP AS A SINGLE USER

You can remotely connect to the desktop on a RHEL server using graphical GNOME applications. Only a single user can connect to the desktop on the server at a given time.

#### 5.2.1. Enabling desktop sharing on the server using GNOME

This procedure configures a RHEL server to enable a remote desktop connection from a single client.

##### Prerequisites

- The GNOME Remote Desktop service is installed:

```
# yum install gnome-remote-desktop
```

## Procedure

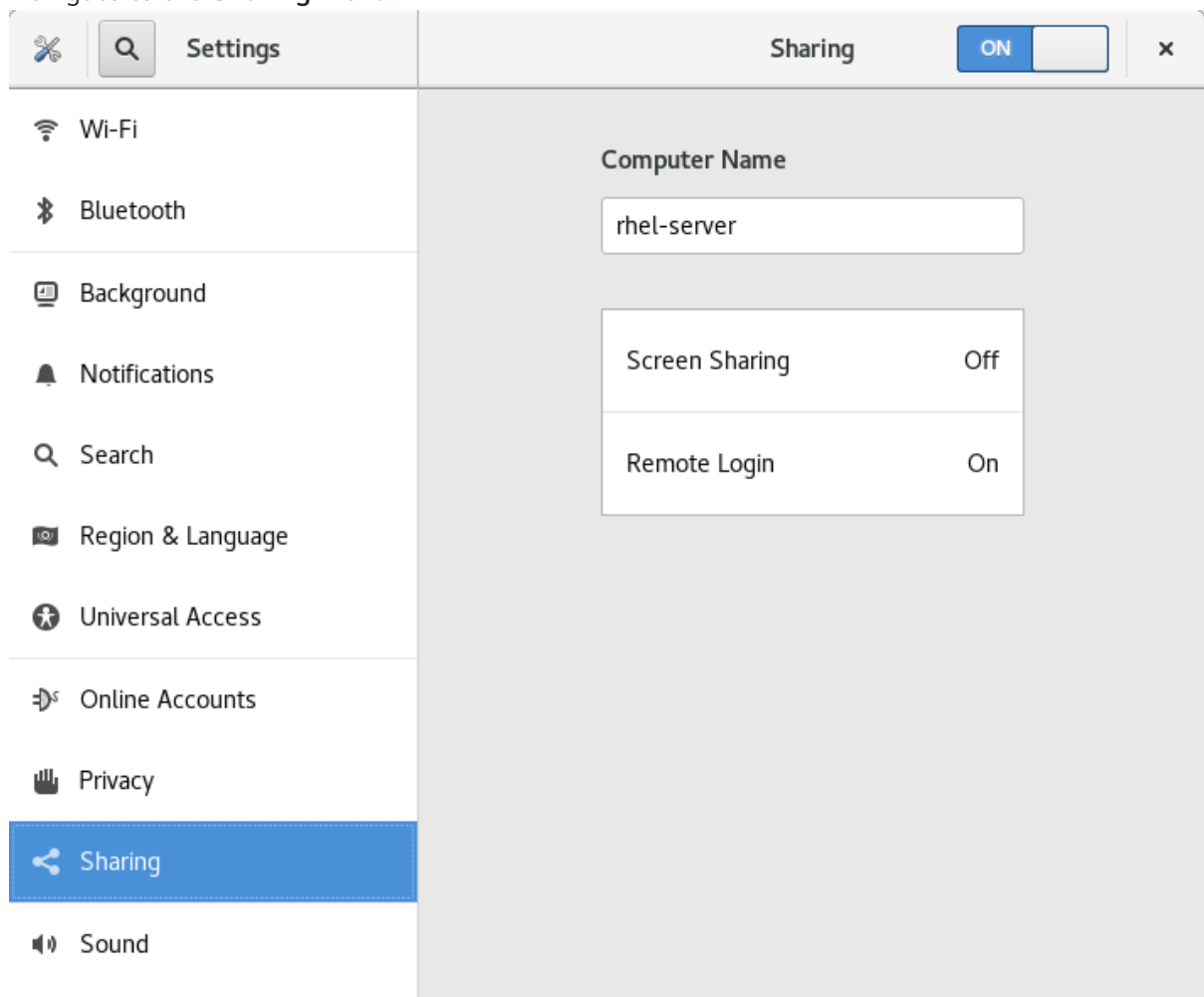
1. Configure a firewall rule to enable VNC access to the server:

```
# firewall-cmd --permanent --add-service=vnc-server
success
```

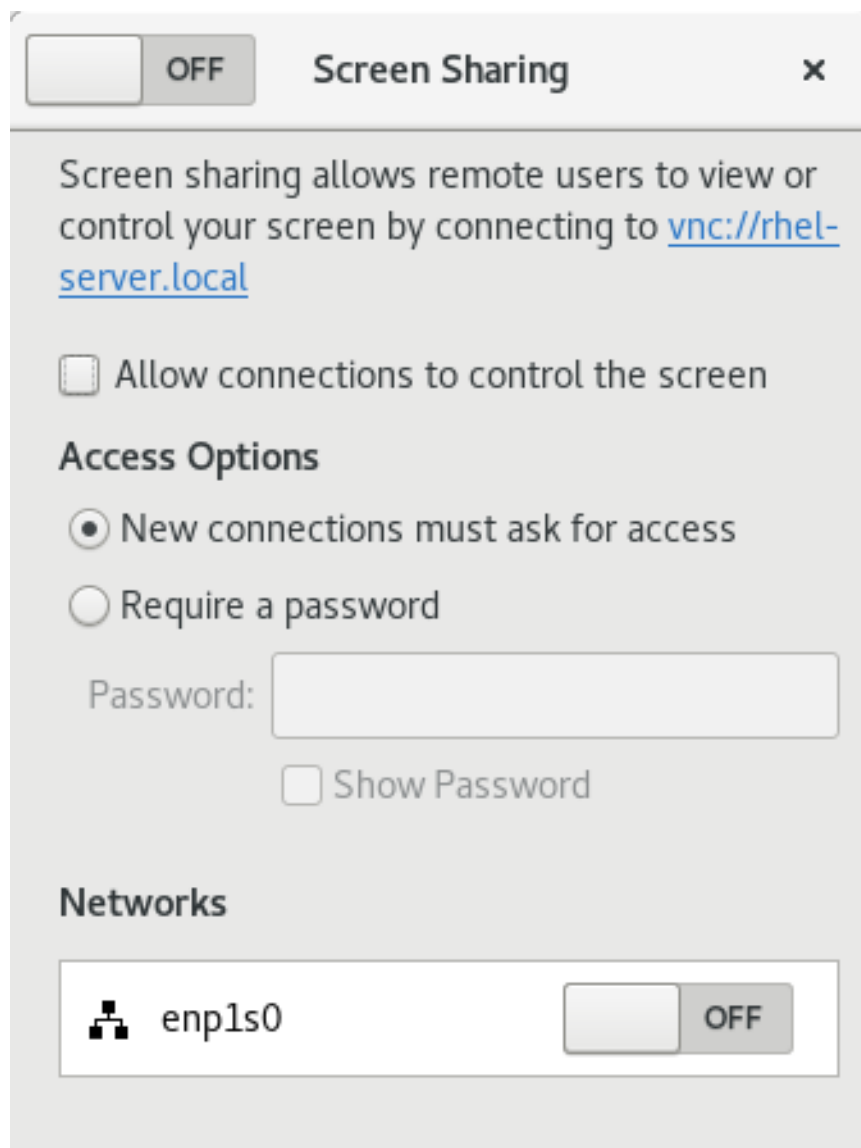
2. Reload firewall rules:

```
# firewall-cmd --reload
success
```

3. Open **Settings** in GNOME.
4. Navigate to the **Sharing** menu:

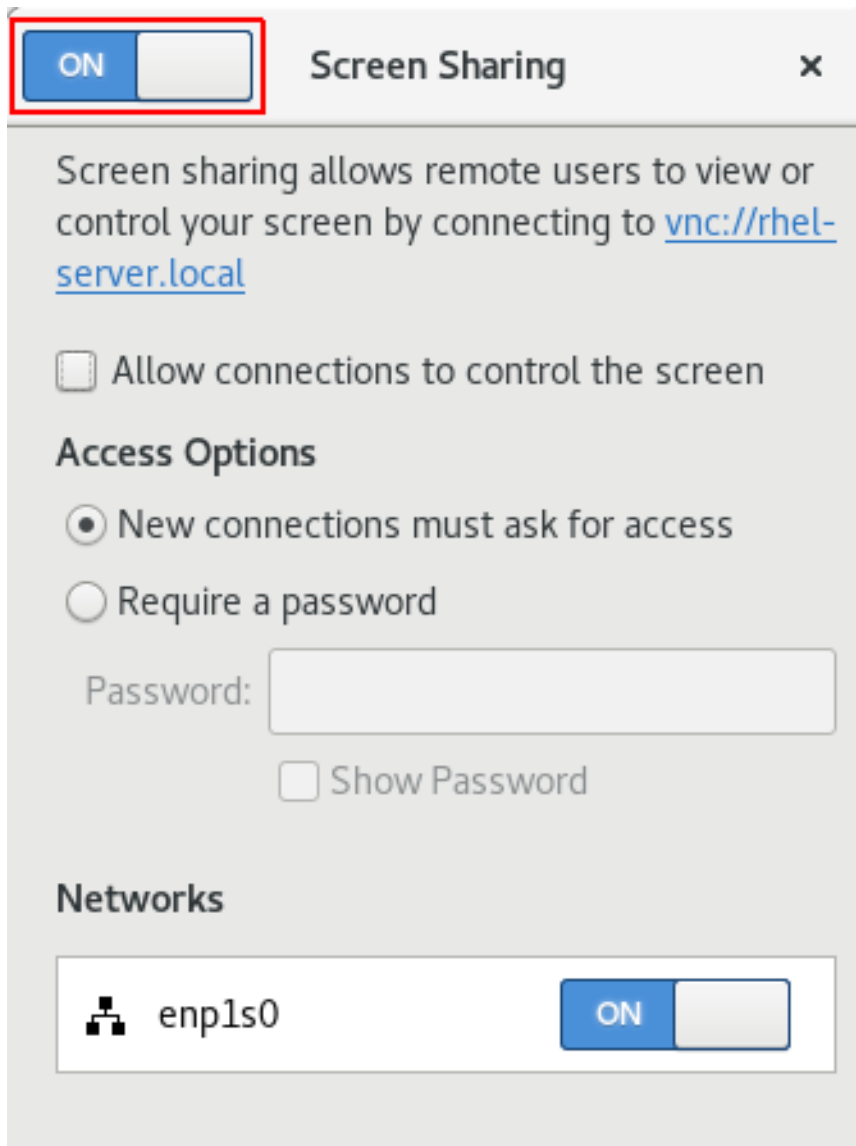


5. Click **Screen Sharing**.  
The screen sharing configuration opens:

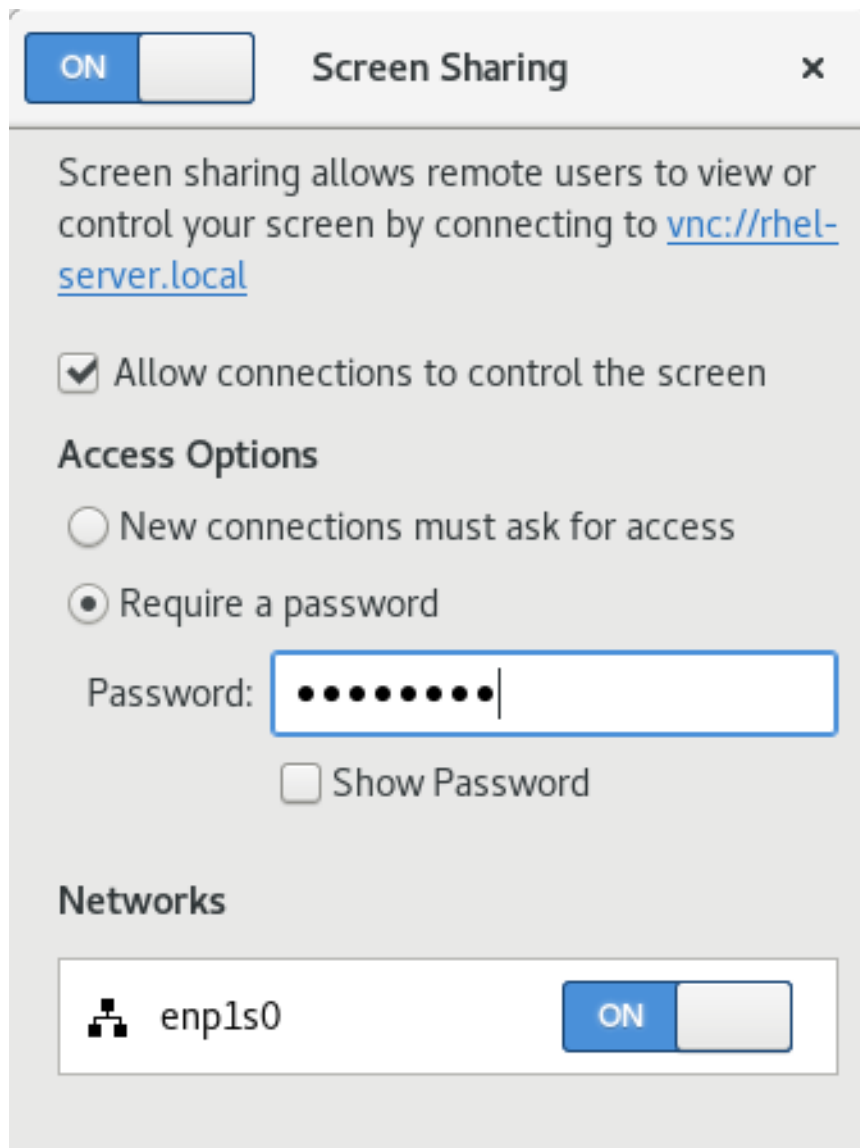


6. Click the switch button in the window header to enable screen sharing:





7. Select the **Allow connections to control the screen** check box.
8. Under **Access Options**, select the **Require a password** option.
9. Set a password in the **Password** field.  
Remote clients must enter this password when connecting to the desktop on the server.



### 5.2.2. Connecting to a shared desktop using GNOME

This procedure connects to a remote desktop session using the **Remote Desktop Viewer** application (**vinagre**). It connects to the graphical session of the user that is currently logged in on the server.

#### Prerequisites

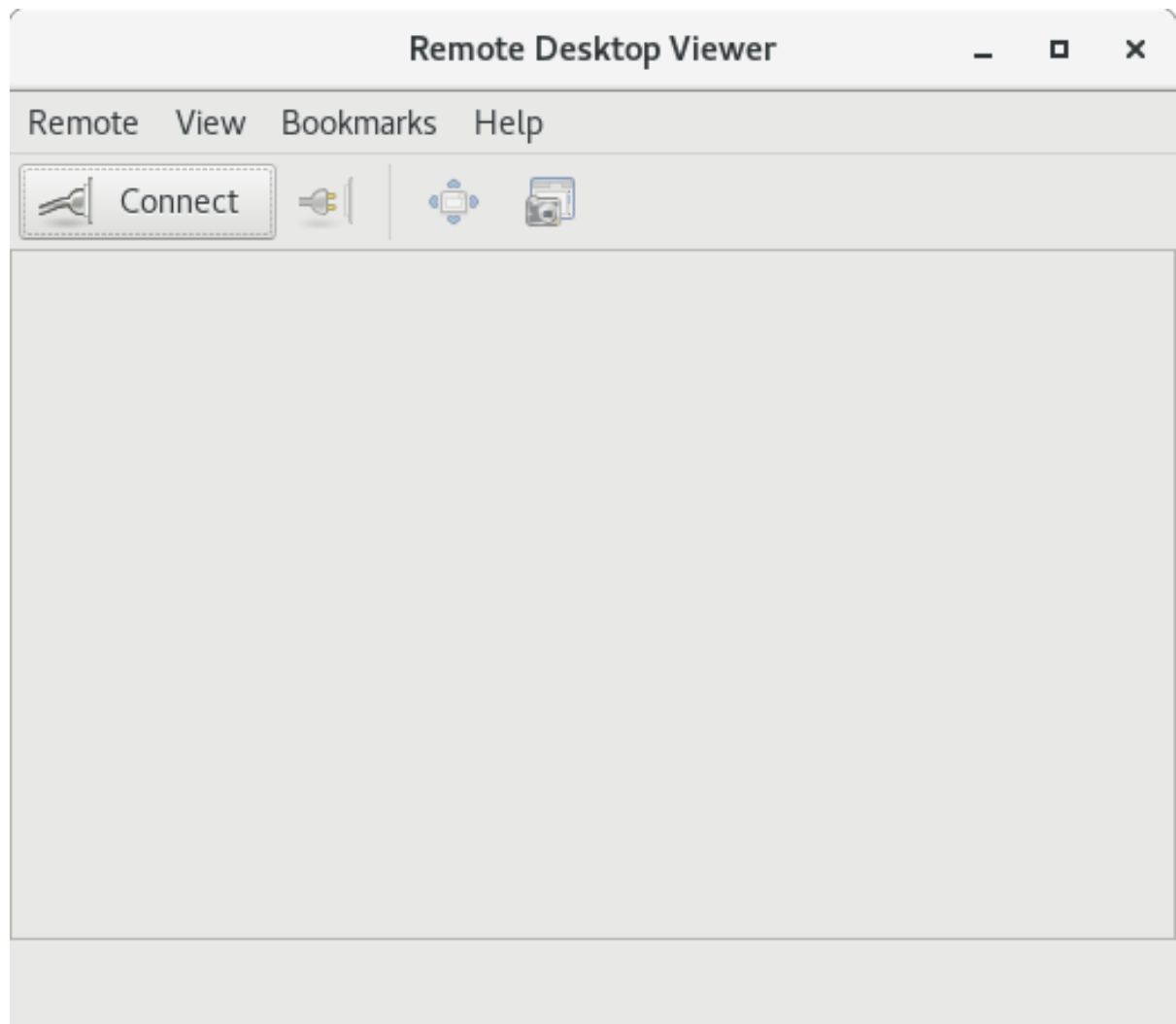
- A user is logged into the GNOME graphical session on the server.
- The desktop sharing is enabled on the server. For details, see [Section 5.2.1, “Enabling desktop sharing on the server using GNOME”](#).

#### Procedure

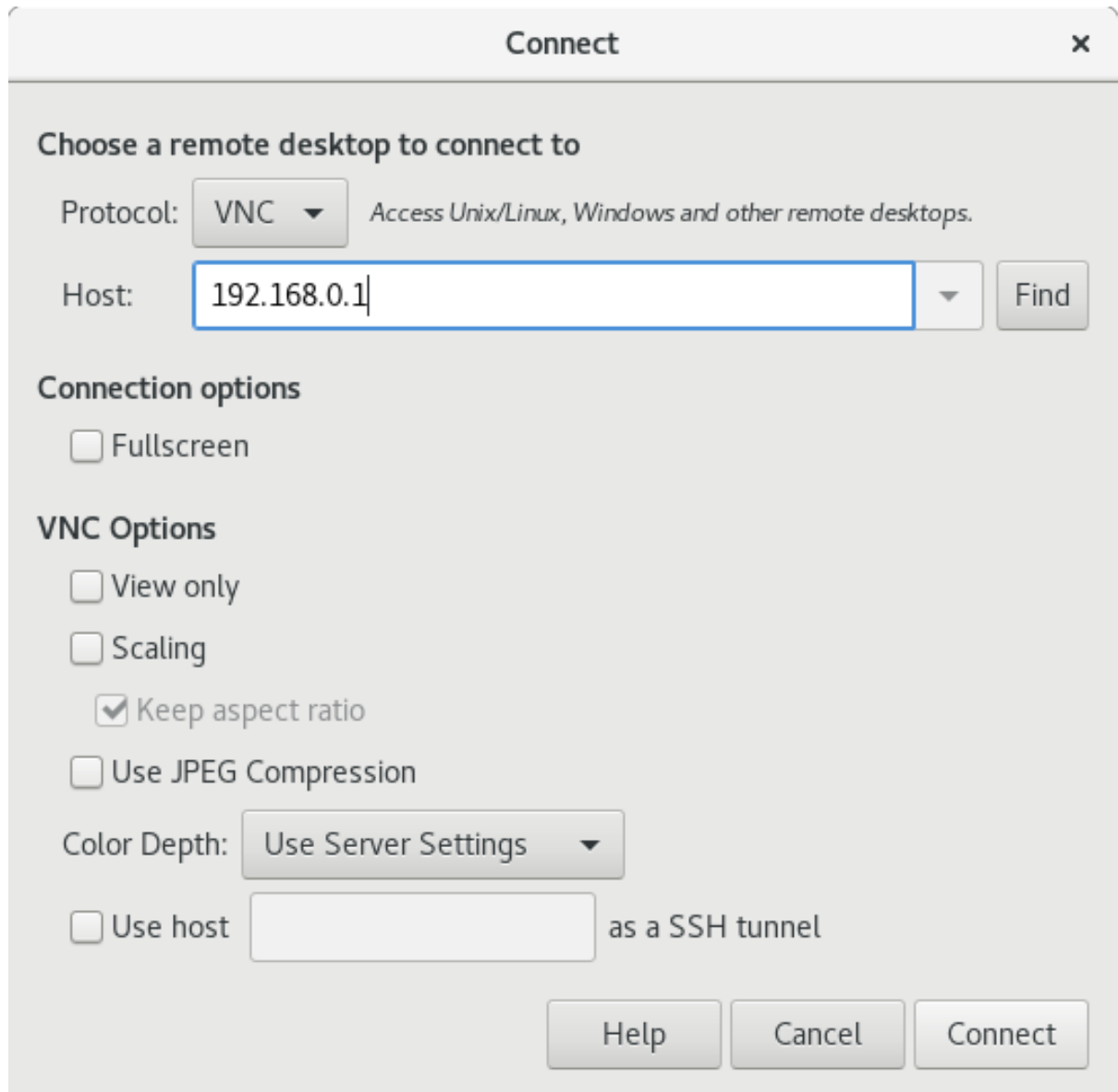
1. Install the **Remote Desktop Viewer** application on the client:

```
# yum install vinagre
```

2. Launch the **Remote Desktop Viewer** application (**vinagre**).
3. Click **Connect**.



4. In the **Protocol** menu, select **VNC**.
5. In the **Host** field, enter the IP address of the server.



6. Click **Connect**.

### Verification steps

1. On the client, check that you can see the shared server desktop.
2. On the server, a screen sharing indicator appears on the right side of the top panel:



You can control the screen sharing in the system menu.

### 5.2.3. Disabling encryption in GNOME VNC

You can disable encryption in the GNOME remote desktop solution. This enables VNC clients that do not support the encryption to connect to the server.

#### Procedure

1. As the server user, set the **encryption** key of **org.gnome.desktop.remote-desktop.vnc** GSettings schema to **['none']**.

```
$ gsettings set org.gnome.desktop.remote-desktop.vnc encryption "[none]"
```

2. Optional: Red Hat recommends that you tunnel the VNC connection over SSH to your VNC port. As a result, the SSH tunnel keeps the connection encrypted.  
For example:

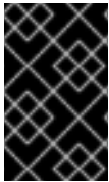
- a. On the client, configure the port forwarding.

```
# ssh -N -T -L 5901:server-ip-address:5901 user@server-ip-address
```

- b. Connect to the VNC session on the **localhost:5901** address.

## 5.3. REMOTELY ACCESSING THE DESKTOP AS MULTIPLE USERS

You can remotely connect to the desktop on a RHEL server and open multiple sessions as different users at the same time.



### IMPORTANT

The following instructions apply to RHEL 8.3 and later. If you are using RHEL 8.0 to 8.2, see [How to configure user specific VNC Server in Red Hat Enterprise Linux 8.0 – 8.2?](#) instead.

### Prerequisites

- Install the VNC server:

```
# yum install tigervnc-server
```

- Install the VNC client:

```
# yum install tigervnc
```

### 5.3.1. The mapping of port and display numbers to users in VNC

With VNC, the client can connect to the desktop sessions of different users on the server. A display number and a TCP port number are attached to each server user that exports a VNC session. The client uses the port number to specify which server user it connects to.

If several clients connect using the same port number, they all open a VNC session to the same server user.

You must configure a mapping for each server user that exports a VNC session. For every such user, you must pick a unique port and display number.

### The recommended mapping

Red Hat recommends that you start with port number 5902 and display number 2 for the first user, and increment the numbers by one for each additional server user.

Port number 5900 and display number 0 represent the server user that is currently logged into the graphical session. You cannot start a VNC server for the user who is already logged into the graphical session.

Table 5.1. Port and display number pairs

Port number	Display number	Note
5900	0	The logged-in user
5901	1	
5902	2	The first recommended VNC user
5903	3	
...	...	

**IMPORTANT**

Red Hat recommends that you do not configure the **root** user to export a VNC session. A **root** VNC session is unsafe and certain elements of the session might not work as expected.

**Firewall rules**

You must open the selected ports in your firewall configuration. Allowing the **vnc-server** service in your firewall opens ports from 5900 to 5903. If you need to enable access to additional server users, you must open ports above 5903 by manually specifying the port numbers.

**5.3.2. VNC server configuration files**

Several configuration files affect the behavior of the VNC server. You can configure the user mapping and various global options.

**General options**

You can configure general options of the VNC server in the **/etc/tigervnc/vncserver-config-defaults** configuration file. The file uses the following format:

```
option1=value
option2
```

For example:

```
session=gnome
alwaysshared
securitytypes=vncauth,tlsvnc
desktop=sandbox
geometry=2000x1200
```

**The priority of configuration files**

The VNC server reads the following files for general options, in order from most important to least important:

1. **/etc/tigervnc/vncserver-config-mandatory**

This file replaces the default configuration and has a higher priority than the per-user configuration. It is intended for system administrators who want to enforce particular VNC options.

2. **\$HOME/.vnc/config**

Individual users can override the default VNC configuration in this file.

3. **/etc/tigervnc/vncserver-config-defaults**

This file stores the default VNC configuration.

## User mapping

You can configure the mapping between users and their associated port and display numbers in the **/etc/tigervnc/vncserver.users** configuration file. The file uses the following format:

```
:number=user
```

For example:

```
:2=test
:3=vncuser
```

## Additional resources

- For a list of available configuration options, see the **Xvnc(1)** man page.

### 5.3.3. Enabling multi-user VNC access on the server

This procedure configures a RHEL server so that multiple users can open VNC sessions on it at the same time.

#### Prerequisites

- If you previously configured VNC using **systemd** unit files, remove any outdated VNC configuration:

```
[root]# rm /etc/systemd/system/vncserver@.service
```

#### Procedure

1. Map users to display and port numbers.

In the **/etc/tigervnc/vncserver.users** configuration file, add a line for each server user that will export a VNC session:

```
:user-number=user-name
```

- Replace *user-number* with the port and display number mapped to the selected existing user.
- Replace *user-name* with the user name of the selected existing user.

For example:

```
-
```

```
:2=vncuser
```

2. Open TCP ports 5900 to 5903 in the firewall:

```
[root]# firewall-cmd --permanent --add-service=vnc-server
```

3. Reload the firewall rules:

```
[root]# firewall-cmd --reload
```

4. Add the following lines to the **/etc/tigervnc/vncserver-config-defaults** configuration file:

```
session=gnome
alwaysshared
```

This configuration has the following effects:

- The VNC server starts the GNOME session when a remote user logs in.
- Multiple users can connect to the VNC server at the same time.

5. As each server user that exports a VNC session, set the VNC password for the user:

```
[regular-user]$ vncpasswd
```

Remote clients must enter this password when connecting to the desktop on the server.

6. If you previously configured VNC for the user, ensure that the configuration files have the correct **SELinux** context:

```
[regular-user]$ restorecon -RFv ~/.vnc
```

7. Enable and start the VNC server unit for the regular user:

```
[root]# systemctl enable --now vncserver@:user-number
```

8. If the server uses the proprietary Nvidia driver, disable Wayland:

- a. Uncomment the **WaylandEnable=False** line in the **/etc/gdm/custom.conf** configuration file.
- b. Add the **DefaultSession=gnome-xorg.desktop** option to the **[daemon]** section of the configuration file.
- c. Reboot the server.

### Additional resources

- To enable VNC access to more than two server users, open TCP ports above 5903. For details, see [Opening a port using CLI](#) or [Opening ports using GUI](#).

### 5.3.4. Connecting to the VNC server as multiple users



This procedure connects to a remote desktop session using the **vncviewer** application. You can open multiple connections to the remote desktop at the same time.

### Prerequisites

- Remote desktop access for multiple users is enabled on the server. For details, see [Section 5.3.3, “Enabling multi-user VNC access on the server”](#).

### Procedure

- Connect to the VNC server:

```
$ vncviewer --shared server-ip:display
```

- Replace *server-ip* with the IP address of the server that you are connecting to.
- Replace *display* with the display number where the server user exports the VNC session.

## 5.4. REMOTELY ACCESSING AN INDIVIDUAL APPLICATION

You can remotely launch a graphical application on a RHEL server and use it from the remote client using X11 forwarding.

### 5.4.1. Enabling X11 forwarding on the server

Configure a RHEL server so that remote clients can use graphical applications on the server over SSH.

### Procedure

1. Install basic X11 packages:

```
# yum install xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils dbus-x11
```



#### NOTE

Your applications might rely on additional graphical libraries.

2. Enable the **X11Forwarding** option in the `/etc/ssh/sshd_config` configuration file:

```
X11Forwarding yes
```

The option is disabled by default in RHEL.

1. Restart the **sshd** service:

```
# systemctl restart sshd.service
```

### 5.4.2. Launching an application remotely using X11 forwarding

Access a graphical application on a RHEL server from a remote client using SSH.

### Prerequisites

- X11 forwarding over SSH is enabled on the server. For details, see [Section 5.4.1, “Enabling X11 forwarding on the server”](#).
- Ensure that an X11 display server is running on your system:
  - On RHEL, X11 is available by default in the graphical interface.
  - On Microsoft Windows, install an X11 server such as Xming.
  - On macOS, install the XQuartz X11 server.
- You have configured and restarted an OpenSSH server. For details, see [Configuring and starting an OpenSSH server](#).

## Procedure

1. Log in to the server using SSH:

```
[local-user]$ ssh -X -Y remote-server
The authenticity of host 'remote-server (192.168.122.120)' can't be established.
ECDSA key fingerprint is SHA256:uYwFlgtP/2YABMHKv5BtN7nHK9SHRL4hdYxAPJVK/kY.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

2. Confirm that a server key is valid by checking its fingerprint.



### NOTE

If you plan to log in to the server on a regular basis, add the user's public key to the server using the **ssh-copy-id** command.

3. Continue connecting by typing **yes**.

```
Warning: Permanently added 'remote-server' (ECDSA) to the list of known hosts.
```

4. When prompted, type the server password.

```
local-user's password:
[local-user ~]$
```

5. Launch the application from the command line:

```
[remote-user]$ application-binary
```

## TIP

To skip the intermediate terminal session, use the following command:

```
$ ssh user@server -X -Y -C binary_application
```

## CHAPTER 6. CONFIGURING GNOME AT LOW LEVEL

### 6.1. INTRODUCTION TO CONFIGURING GNOME

The following components store settings in the GNOME desktop environment:

#### GSettings

A high-level API for GNOME application settings.

#### The **dconf** database

A key-based Binary Large Object (BLOB) database for storing GNOME configurations. It manages user settings such as GDM, application, and proxy settings, and serves as the back end for **GSettings**.

You can view and edit **GSettings** values using either of the following tools:

#### The **dconf-editor** graphical application

The **dconf-editor** provides a GUI for browsing the settings and their editing. It presents the hierarchy of settings in a tree-view and also displays additional information about each setting, including the description, type and default value.

#### The **gsettings** command-line utility

The **gsettings** utility can be used to display and set **dconf** values. The **gsettings** utility supports Bash completion for commands and settings. This tool also enables you to automate configuration in shell scripts.

#### The **dconf** command-line utility

A command-line utility which is used for reading and writing individual values or entire directories from and to a **dconf** database.



#### WARNING

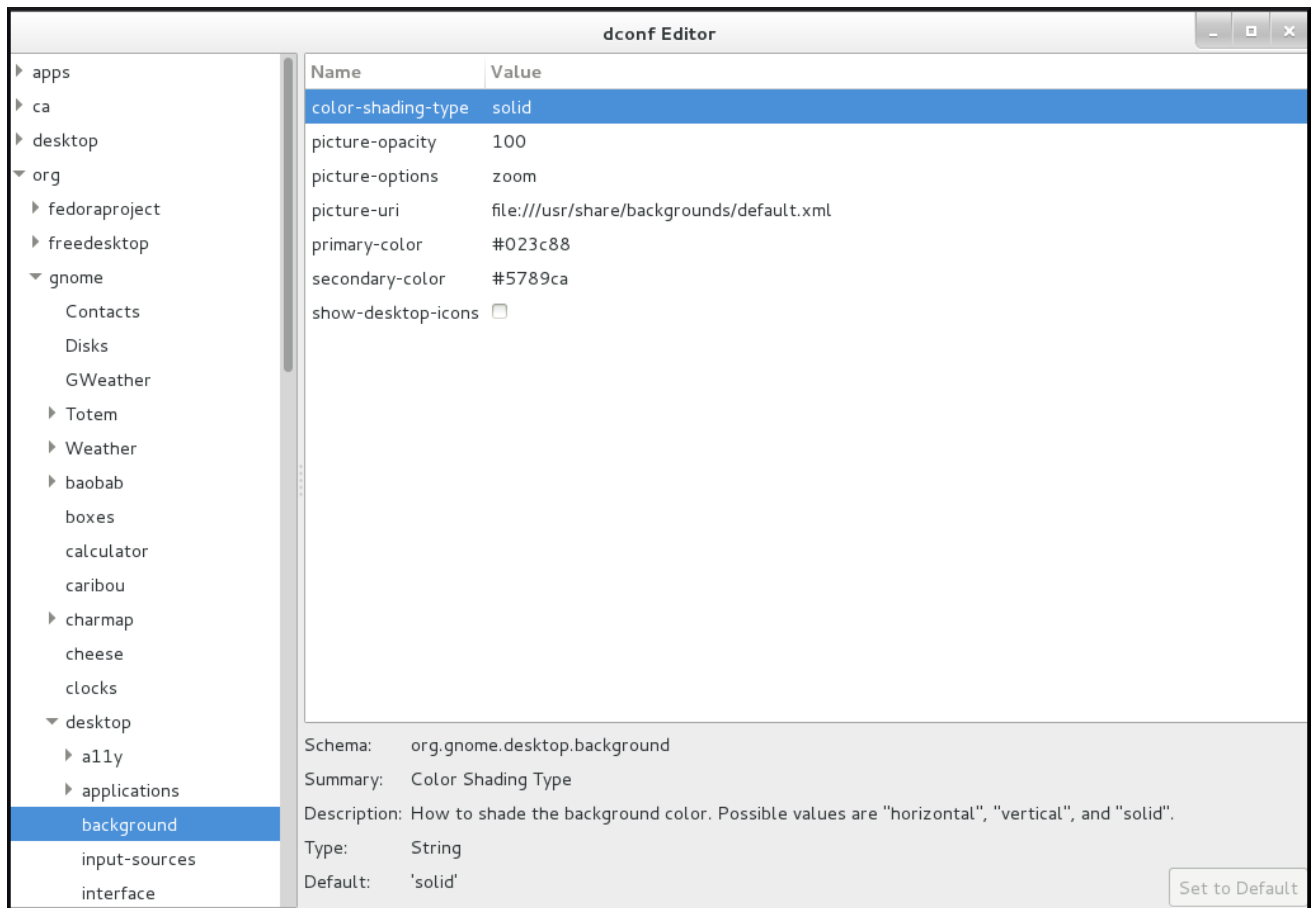
As a user or application developer, do not use the **dconf** utility directly.

To manipulate **dconf** values, always use the **dconf editor** application or the **gsettings** utility.

The only exception to use **dconf** directly is when setting system-wide default configurations, because the other tools do not allow to manipulate such configurations.

These tools share the following features:

- Browsing and changing options for system and application preferences
- Changing preferences
- Running by regular users, because both tools are intended to browse and modify the current user's **GSettings** database

Figure 6.1. dconf editor showing the `org.gnome.desktop.background` keys

## 6.2. GSETTINGS SCHEMAS AND KEYS

The GSettings database stores all settings in a tree of schemas and keys:

- A key stores a particular value of a setting.
- A schema defines a group and subgroups of related settings, including a summary and description of each setting and its default value.

Usually, there is a specific, well-known path associated with each schema in the GSettings database. However, in some situations schemas are *relocatable*, so that they can be applied to more than one path in the GSettings database. A relocatable schema can have an arbitrary number of instances. For example, this is useful if the schema describes an account, and you want to be able to store an arbitrary number of accounts in GSettings.

A schema path usually refers to a domain name associated with the configured program or library.

The **gsettings** utility specifies schemas and keys in a format where the key is separated with a space:

*domain.name key*

If a schema is relocatable, you must also specify a path after the schema name:

*domain.name:path key*

The **dconf** utility specifies schemas and keys as a single slash-separated string:

```
/domain/name/key
```

### Example 6.1. The folder view setting in Files

You can access the setting that controls the folder view in the **Files** application using the following format.

With **gsettings**:

```
org.gnome.nautilus.preferences default-folder-viewer
```

With **dconf**:

```
/org/gnome/nautilus/preferences/default-folder-viewer
```

## 6.3. GSETTINGS VALUES

Each GSettings key stores the value of a particular setting as a GVariant data type. To modify the value of a key, you must specify the value in the GVariant text format.

GVariant supports the following data types:

Data type	GVariant text format examples
Boolean	<ul style="list-style-type: none"> <li><b>true</b></li> <li><b>false</b></li> </ul>
String	<ul style="list-style-type: none"> <li><b>'option1'</b></li> <li><b>"option2"</b></li> </ul>
Number	<ul style="list-style-type: none"> <li><b>5</b></li> <li><b>37.5</b></li> <li><b>3.75e1</b></li> </ul>
Tuple	<ul style="list-style-type: none"> <li><b>("hello", 42)</b></li> </ul>
Array	<ul style="list-style-type: none"> <li><b>[1, 2, 3]</b></li> </ul>

Data type	GVariant text format examples
Dictionary	<ul style="list-style-type: none"> <li>• <code>{1: "one", 2: "two", 3: "three"}</code></li> </ul>
Variant	<ul style="list-style-type: none"> <li>• <code>[&lt;"hello"&gt;, &lt;42&gt;]</code></li> </ul>
Maybe	<ul style="list-style-type: none"> <li>• <code>just 'hello'</code></li> <li>• <code>nothing</code></li> </ul>
Byte string	<ul style="list-style-type: none"> <li>• <code>b'abc'</code></li> <li>• <code>b"abc"</code></li> </ul>

#### Additional resources

- [GVariant Text Format](#)
- [GVariant Format Strings](#)

## 6.4. MANAGING USER AND SYSTEM GNOME SETTINGS

The **dconf** database provides several levels of control over GNOME configuration for system administrators and users:

- Administrators can define default settings that apply to all users.
- Users can override the defaults with their own settings.
- Administrators can also lock settings to prevent users from overriding them.

## 6.5. DCONF PROFILES

A **dconf** profile is a list of system's hardware and software configuration databases, which the **dconf** system collects.

The **dconf** profiles enable you to compare identical systems to troubleshoot hardware or software problems.

The **dconf** system stores its profiles in the text files that can be located either within the `/etc/dconf/profile/` directory or elsewhere. The `$DCONF_PROFILE` environment variable can specify a relative path to the file from `/etc/dconf/profile/`, or an absolute path, such as in a user's home directory.

Note that key pairs that are set in a **dconf** profile override the default settings.

On startup, **dconf** consults the `$DCONF_PROFILE` environment variable to find the name of the **dconf** profile to open. The result depends on whether the variable is set or not:

- If set, **dconf** attempts to open the profile named in the variable and aborts if this step fails.
- If not set, **dconf** attempts to open the profile named **user** and uses an internal hard-wired configuration if this step fails.

Each line in a **dconf** profile specifies one **dconf** database.

The first line indicates the database used to write changes. The remaining lines show read-only databases.

The following is a sample profile stored in **/etc/dconf/profile/user**:

```
user-db:user
system-db:local
system-db:site
```

In this example, the **dconf** profile specifies three databases:

#### **user**

The name of the user database that can be found in **~/.config/dconf**

#### **local and site**

System databases, located in **/etc/dconf/db/**



#### **NOTE**

To apply a new **dconf** user profile to the user's session, you need to log out and log in, because the **dconf** profile for a session is determined at login.

## 6.6. CONFIGURING CUSTOM DEFAULT VALUES

You can set machine-wide default settings by providing a default for a key in a **dconf** profile. Users can override these defaults.

#### **Procedure**

1. Identify the GSettings schemas, keys, and allowed values of the settings that you want to configure.

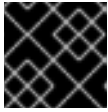
You can either use the **dconf editor** application or the **gsettings** utility. For details on **gsettings**, see [Working with GSettings keys on command line](#).

2. Ensure that the **/etc/dconf/profile/user** configuration file enables the user profile:

```
user-db:user
system-db:local
```

3. Create a plain text keyfile for the default settings in the **/etc/dconf/db/local.d/** directory. Choose a file name that represents the settings. The file name usually starts with a number, such as **/etc/dconf/db/local.d/01-background**.
4. In the keyfile, specify the default settings in the following format:

```
[path/to/schema]
key-name=value
another-key-name=another-value
```



## IMPORTANT

Separate the schema components with a slash (/), not a dot (.) here.

### Example 6.2. A default background

For example, to set a default desktop background, specify the following configuration in **/etc/dconf/db/local.d/01-background**:

```
[org/gnome/desktop/background]
picture-uri='file:///usr/local/share/backgrounds/wallpaper.jpg' 1
picture-options='scaled' 2
primary-color='000000' 3
secondary-color='FFFFFF' 4
```

- 1 URI to use for the background image. Note that the backend only supports local **file://** URIs.
- 2 Determines how the image set by **wallpaper\_filename** is rendered.
- 3 Left or Top color when drawing gradients, or the solid color.
- 4 Right or Bottom color when drawing gradients, not used for solid color.

5. Update the system databases:

```
# dconf update
```

6. The user must log out and log in again for the changes to take effect.

### Additional resources

- If you want to avoid creating a user profile, you can use the **dconf** utility to read and write individual values or entire directories from and to a **dconf** database. For more information, see the **dconf(1)** man page.

## 6.7. LOCKING DOWN SPECIFIC SETTINGS

By default, any settings that users make take precedence over the system settings. Therefore, users can override the system settings with their own. Using the lockdown mode in **dconf**, you can prevent users from changing specific settings.

### Procedure

- Create the **/etc/dconf/db/local.d/locks/** directory.



2. In this directory, add any number of files listing keys that you want to lock.

### Example 6.3. Locking the settings for the default wallpaper

1. Set a default wallpaper.
2. Create the `/etc/dconf/db/local.d/locks/` directory.
3. Create the `/etc/dconf/db/local.d/locks/00-default-wallpaper` file with the following content, listing one key per line:

```
# Prevent users from changing values for the following keys:
/org/gnome/desktop/background/picture-uri
/org/gnome/desktop/background/picture-options
/org/gnome/desktop/background/primary-color
/org/gnome/desktop/background/secondary-color
```

4. Update the system databases:

```
# dconf update
```

## 6.8. STORING USER SETTINGS OVER NFS

If you use GNOME on a system with NFS home directories, you must set the **keyfile** back end for the **dconf** database. Otherwise, **dconf** might not work correctly. With this configuration, **dconf** stores settings in the `~/.config/dconf-keyfile/user` file.

### Procedure

1. Ensure that the **glib2-fam** package is installed on the system:

```
# yum install glib2-fam
```

Without this package, notifications on configuration changes made on remote machines are not displayed properly.

2. Create or edit the `/etc/dconf/profile/user` file on every client.
3. At the very beginning of the `/etc/dconf/profile/user` file, add the following line:

```
service-db:keyfile/user
```

4. Users must log out and log back in.  
**dconf** polls the **keyfile** back end to determine whether updates have been made, so settings might not be updated immediately.

## 6.9. WORKING WITH GSETTINGS KEYS ON COMMAND LINE

You can use the **gsettings** utility to configure, manipulate and manage GSettings keys. Look at the examples of use cases that involve using the **gsettings** utility:

### 6.9.1. Setting the value of a key

```
$ gsettings set SCHEMA[:PATH] KEY VALUE
```

Note that the value is specified as a serialised GVariant.

If the operation succeeds, the command displays no output. The change is valid immediately.

#### Example 6.4. Adding selected applications among your favorite applications

```
$ gsettings set org.gnome.shell favorite-apps \
    ["firefox.desktop", 'evolution.desktop', 'rhythmbox.desktop']"
```

### 6.9.2. Monitoring key changes

To monitor a key for changes and print values that change:

```
$ gsettings monitor SCHEMA[:PATH] [KEY]
```

If you do not specify the **KEY** argument, all keys in the schema are monitored. Monitoring continues until you stop the process.

#### Example 6.5. Monitoring changes of the favorite applications key

To monitor the changes in your favorite applications:

1. Monitor the **favorite-apps** key:

```
$ gsettings monitor org.gnome.shell favorite-apps
```

2. In another terminal, change the **favorite-apps** key:

```
$ gsettings set org.gnome.shell favorite-apps \
    ["firefox.desktop", 'evolution.desktop', 'rhythmbox.desktop']"
```

3. As a result, the command in the first terminal displays a notification that the **favorite-apps** key has changed:

```
favorite-apps: ['firefox.desktop', 'evolution.desktop', 'rhythmbox.desktop']
```

### 6.9.3. Checking whether a key is writable

```
$ gsettings writable SCHEMA[:PATH] KEY
```

#### Example 6.6. Checking whether the favorite applications key is writable

The following key is writable:

```
$ gsettings writable org.gnome.shell favorite-apps
True
```

#### 6.9.4. Listing the valid values for a key

```
$ gsettings range SCHEMA[:PATH] KEY
```

##### Example 6.7. Checking the range of valid values for the remember-mount-password key

The following key stores a Boolean value:

```
$ gsettings range org.gnome.shell remember-mount-password
type b
```

For more information, see [GVariant Format Strings](#).

#### 6.9.5. Displaying the description of a key

```
$ gsettings describe SCHEMA[:PATH] KEY
```

##### Example 6.8. Displaying the description of the picture-uri key

```
$ gsettings describe org.gnome.desktop.screensaver picture-uri
URI to use for the background image. Note that the backend only supports local `file://` URIs.
```

#### 6.9.6. Displaying the value of a key

```
$ gsettings get SCHEMA[:PATH] KEY
```

Note that the value is displayed as a serialised GVariant.

##### Example 6.9. Displaying value of the remember-mount-password key

```
$ gsettings get org.gnome.shell remember-mount-password
false
```

#### 6.9.7. Resetting the value of a key

```
$ gsettings reset SCHEMA[:PATH] KEY
```

If resetting succeeds, the command displays no output.

Default values are stored in **dconf** and **gsettings-desktop-schemas** files.

#### Example 6.10. Resetting the lock-delay key to its default value

The default value of the **lock-delay** key is 0, and it is stored in the **/usr/share/glib-2.0/schemas/org.gnome.desktop.screensaver.gschema.xml** file.

```
$ gsettings reset org.gnome.desktop.screensaver lock-delay
```

As a result, the value of **lock-delay** value is set to 0.

### 6.9.8. Resetting a whole schema

```
$ gsettings reset-recursively SCHEMA[:PATH]
```

#### Example 6.11. Resetting the **org.gnome.desktop.screensaver** schema to its default values

```
$ gsettings reset-recursively org.gnome.desktop.screensaver
```

As a result, the **lock-delay** value is reset to 0, and other keys within the **org.gnome.desktop.screensaver** schema that were changed by user are reset to their defaults as well.

### 6.9.9. Listing installed non-relocatable schemas

```
$ gsettings list-schemas
```

#### Example 6.12. Listing installed non-relocatable schemas

To list all schemas installed on your system that are non-relocatable:

```
$ gsettings list-schemas

org.gnome.rhythmbox.library
org.gnome.shell.overrides
org.gnome.system.proxy.https
org.gnome.clocks
org.gnome.eog.fullscreen
org.gnome.login-screen
org.gnome.eog.view
...
```

### 6.9.10. Listing all keys in a schema

```
$ gsettings list-keys SCHEMA[:PATH]
```

#### Example 6.13. Listing all keys in the **org.gnome.shell** schema

```
-
```

```
$ gsettings list-keys org.gnome.shell

enabled-extensions
command-history
remember-mount-password
always-show-log-out
had-bluetooth-devices-setup
looking-glass-history
disable-user-extensions
app-picker-view
disable-extension-version-validation
development-tools
favorite-apps
...
```

### 6.9.11. Listing all schemas nested in a schema

```
$ gsettings list-children SCHEMA[:PATH]
```

Note that the list is empty if the schema contains no other schemas.

#### Example 6.14. Listing schemas in the org.gnome.shell schema

```
$ gsettings list-children org.gnome.shell

keyboard org.gnome.shell.keyboard
keybindings org.gnome.shell.keybindings
```

### 6.9.12. Listing the keys and values in a schema recursively

```
$ gsettings list-recursively [SCHEMA[:PATH]]
```

If you do not specify any schema, the command lists all keys within all schemas.

#### Example 6.15. Listing all keys and values on the system recursively

```
$ gsettings list-recursively

org.gnome.nautilus.desktop network-icon-visible false
org.gnome.nautilus.desktop font "
org.gnome.nautilus.desktop network-icon-name 'Network Servers'
org.gnome.nautilus.desktop home-icon-name 'Home'
org.gnome.nautilus.desktop volumes-visible true
org.gnome.Vinagre always-enable-listening false
org.gnome.Vinagre always-show-tabs false
org.gnome.Vinagre show-accels false
org.gnome.Vinagre history-size 15
org.gnome.Vinagre shared-flag true
...
```

## 6.10. ACKNOWLEDGEMENTS

Certain portions of this text first appeared in the [GNOME Desktop System Administration Guide](#) . Copyright © 2014 The GNOME Project, Michael Hill, Jim Campbell, Jeremy Bicha, Ekaterina Gerasimova, minnie\_eg, Aruna Sankaranarayanan, Sindhu S, Shobha Tyagi, Shaun McCance, David King, and others. Licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

The editors of this Red Hat Enterprise Linux 8 documentation would like to thank the GNOME community members for their valuable contributions to the GNOME Desktop System Administration Guide.

## CHAPTER 7. CONFIGURING FILE ASSOCIATIONS

This section provides information about configuring RHEL to open or access files with different formats.

In GNOME, MIME (Multipurpose Internet Mail Extension) types help to identify the format of a file and applications to use to open these files.

### 7.1. MULTIPURPOSE INTERNET MAIL EXTENSION TYPES

The GNOME desktop uses MIME types to:

- Determine which application should open a specific file format by default.
- Register other applications that can open files of a specific format.
- Set a string describing the type of a file, for example, in a file properties dialog of the files application.
- Set an icon representing a specific file format, for example, in a file properties dialog of the files application.

MIME type names follow a given format:

**media-type/subtype-identifier**

For example, **image/jpeg**.

Here **image** is a media type and **jpeg** is the subtype identifier.

GNOME follows Multipurpose Internet Mail Extension (MIME) info specification from the [Freedesktop.org](http://freedesktop.org) to determine:

- The machine-wide and user-specific location to store all the MIME type specification files.
- How to register a MIME type so that the desktop environment knows which application you can use to open a specific file format.
- How users can change which applications should open with what file formats.

#### MIME database

The MIME database is a collection of all the MIME type specification files that GNOME uses to store information about known MIME types.

The most important part of the MIME database from the system administrator's point of view is the **/usr/share/mime/packages/** directory, where the MIME type-related files specifying information about known MIME types are stored. One example of such a file is **/usr/share/mime/packages/freedesktop.org.xml**, specifying information about the standard MIME types available on the system, by default. The shared-mime-info package provides this file.

#### Additional resources

- For detailed information describing the MIME type system, see Shared MIME Info specifications on the [Freedesktop website](http://freedesktop.org).

### 7.2. ADDING A CUSTOM MIME TYPE FOR ALL USERS

You must create a new MIME type specification file in the `/usr/share/mime/packages/` directory and a `.desktop` file in the `/usr/share/applications/` directory to add a custom MIME type for all the users on the system and register a default application for that MIME type.

## Procedure

1. Create the `/usr/share/mime/packages/application-x-newtype.xml` file with following content:

```
<?xml version="1.0" encoding="UTF-8"?>

<mime-info xmlns="http://www.freedesktop.org/standards/shared-mime-info">
  <mime-type type="application/x-newtype">
    <comment>new mime type</comment>
    <glob pattern="*.xyz"/>
  </mime-type>
</mime-info>
```

The sample `application-x-newtype.xml` file here defines a new MIME type `application/x-newtype` and assigns file names with the `.xyz` extension to that MIME type.

2. Create a new `.desktop` file named, for example, `myapplication1.desktop` in the `/usr/share/applications/` with following content:

```
[Desktop Entry]
Type=Application
MimeType=application/x-newtype
Name=My Application 1
Exec=myapplication1 field_code
```

The sample `myapplication1.desktop` file here associates the `application/x-newtype` MIME type with an application named **My Application 1**. It is run by a command `myapplication1`.

Based on how **myapplication1** gets started, choose one respective field code from [Desktop Entry Specification](#). For example, for an application capable of opening multiple files, use:

```
Exec=myapplication1 %F
```

3. As a root user, update the MIME database for your changes to take effect.

```
# update-mime-database /usr/share/mime
```

4. As a root user, update the application database.

```
# update-desktop-database /usr/share/applications
```

## Verification steps

1. To verify that you have successfully associated `*.xyz` files with the `application/x-newtype` MIME type, first create an empty file, for example `test.xyz` and execute the following commands:

```
$ touch test.xyz

$ gvfs-info test.xyz | grep "standard::content-type"
standard::content-type: application/x-newtype
```



- 2. To verify **myapplication1.desktop** is correctly set as the default registered application for the **application/x-newtype** MIME type, execute following command:

```
$ gio mime --query application/x-newtype
Default application for 'application/x-newtype': myapplication1.desktop
Registered applications:
  myapplication1.desktop
Recommended applications:
  myapplication1.desktop
```

## 7.3. ADDING A CUSTOM MIME TYPE FOR INDIVIDUAL USERS

You must create a new MIME type specification file in the `~/.local/share/mime/packages/` directory and a **.desktop** file in the `~/.local/share/applications/` directory to add a custom MIME type for individual users and register a default application for that MIME type.

### Procedure

1. Create the `~/.local/share/mime/packages/application-x-newtype.xml` file with following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<mime-info xmlns="http://www.freedesktop.org/standards/shared-mime-info">
  <mime-type type="application/x-newtype">
    <comment>new mime type</comment>
    <glob pattern="*.xyz"/>
  </mime-type>
</mime-info>
```

The sample **application-x-newtype.xml** file here defines a new MIME type **application/x-newtype** and assigns file names with the **.xyz** extension to that MIME type.

2. Create a new **.desktop** file named, for example, **myapplication1.desktop**, and place it in the `~/.local/share/applications/` directory with following content:

```
[Desktop Entry]
Type=Application
MimeType=application/x-newtype
Name=My Application 1
Exec=myapplication1 field_code
```

The sample **myapplication1.desktop** file above associates the **application/x-newtype** MIME type with an application named My Application 1. It is run by a command **myapplication1**.

Based on how **myapplication1** gets started, choose one respective field code from [Desktop Entry Specification](#). For example, for an application capable of opening multiple files, use:

```
Exec=myapplication1 %F
```

3. Update the MIME database for your changes to take effect:

```
$ update-mime-database ~/.local/share/mime
```

4. Update the application database:

```
$ update-desktop-database ~/.local/share/applications
```

### Verification steps

1. To verify that you have successfully associated **\*.xyz** files with the **application/x-newtype** MIME type, first create an empty file, for example, test.xyz and execute the following commands:

```
$ touch test.xyz  
  
$ gvfs-info test.xyz | grep "standard::content-type"  
standard::content-type: application/x-newtype
```

2. To verify that **myapplication1.desktop** is correctly set as the default registered application for the **application/x-newtype** MIME type, execute following command:

```
$ gio mime --query application/x-newtype  
Default application for 'application/x-newtype': myapplication1.desktop  
Registered applications:  
myapplication1.desktop  
Recommended applications:  
myapplication1.desktop
```

## 7.4. OPTIONS TO OVERRIDE DEFAULT MIME TYPES

By default, the package-installed **/usr/share/applications/mimeapps.list** and **/usr/share/applications/gnome-mimeapps.list** files specify which application is registered to open specific MIME types.

System administrators can create the **/etc/xdg/mimeapps.list** or **/etc/xdg/gnome-mimeapps.list** file with a list of MIME types they want to override with the default registered application.

Local users can create the **~/.local/share/applications/mimeapps.list** or **~/.local/share/applications/gnome-mimeapps.list** file with a list of MIME types for which they want to override the default registered application.

Configurations are applied in the following order:

1. **/usr/share/applications/**
2. **/etc/xdg/**
3. **~/.local/share/application/**

Within a particular location, the configurations are applied in the following order:

1. mimeapps.list
2. gnome-mimeapps.list

## 7.5. OVERRIDING DEFAULT REGISTERED APPLICATION FOR ALL THE USERS

As a system administrator, you can update the configuration based on the requirements. System administrator's configuration takes precedence over default package configuration. Within each, the desktop-specific configuration takes precedence over the configuration that does not specify the desktop environment.

### Procedure

1. Consult the **/usr/share/applications/mimeapps.list** file to determine the MIME types for which you want to change the default registered application. For example, the following sample of the **mimeapps.list** file specifies the default registered application for the **text/html** and **application/xhtml+xml** MIME types:

```
[Default Applications]
text/html=firefox.desktop
application/xhtml+xml=firefox.desktop
```

This example above specifies default application (Firefox) by specifying its corresponding **.desktop** file (**firefox.desktop**). You can find **.desktop** files for other applications in the **/usr/share/applications/** directory.

2. Create the **/etc/xdg/mimeapps.list** file and specify the MIME types and their corresponding default registered applications in this file.

```
[Default Applications]
text/html=myapplication1.desktop
application/xhtml+xml=myapplication2.desktop
```

This example above sets the default registered application for the **text/html** MIME type to **myapplication1.desktop** and **application/xhtml+xml** MIME type to **myapplication2.desktop**.

### Verification steps

- For these settings to function correctly, ensure that both the **myapplication1.desktop** and **myapplication2.desktop** files are placed in the **/usr/share/applications/** directory.
- Execute the **gio mime query** command to verify that the default registered application is set correctly.

```
$ gio mime text/html
Default application for 'text/html': myapplication1.desktop
Registered applications:
myapplication1.desktop
firefox.desktop
Recommended applications:
myapplication1.desktop
firefox.desktop
```

## 7.6. OVERRIDING DEFAULT REGISTERED APPLICATION FOR INDIVIDUAL USERS

Individual users can also update the configuration based on the requirements. This configuration takes precedence over the system administrator's configuration and the system administrator's configuration takes precedence over the package configuration. Within each, the desktop-specific configuration takes precedence over the configuration that does not specify the desktop environment.

## Procedure

1. Consult the **/usr/share/applications/mimeapps.list** file to determine the MIME types for which you want to change the default registered application. For example, the following sample of the **mimeapps.list** file specifies the default registered application for the **text/html** and **application/xhtml+xml** MIME types:

```
[Default Applications]
text/html=firefox.desktop
application/xhtml+xml=firefox.desktop
```

This example above specifies default application (Firefox) by specifying its corresponding **.desktop** file (**firefox.desktop**). You can find **.desktop** files for other applications in the **/usr/share/applications/** directory.

2. Create the **~/.local/share/applications/mimeapps.list** file and specify the MIME types and their corresponding default registered applications in this file.

```
[Default Applications]
text/html=myapplication1.desktop
application/xhtml+xml=myapplication2.desktop
```

This example above sets the default registered application for the **text/html** MIME type to **myapplication1.desktop** and **application/xhtml+xml** MIME type to **myapplication2.desktop**.

## Verification steps

- For these settings to function correctly, ensure that both the **myapplication1.desktop** and **myapplication2.desktop** files are placed in the **/usr/share/applications/** directory.
- Execute the **gio mime query** command to verify that the default registered application is set correctly.

```
$ gio mime text/html
Default application for 'text/html': myapplication1.desktop
Registered applications:
myapplication1.desktop
firefox.desktop
Recommended applications:
myapplication1.desktop
firefox.desktop
```

## CHAPTER 8. CUSTOMIZING DESKTOP APPEARANCE

This section explains how system administrators can customize appearance of the desktop environment for different users of the system.

### 8.1. CUSTOMIZING DESKTOP BACKGROUNDS

As a system administrator, you can configure the default desktop background, add extra backgrounds, or add multiple backgrounds available to all users of the system.

If the users are not permitted to change background from the defaults, you must lock the settings using the **locks** directory. Otherwise, each user can customize the background according to their preferences.

#### 8.1.1. Customizing the default desktop background

You can configure the default desktop background and its appearance by setting the relevant GSettings keys in the **org.gnome.desktop.background** schema.

##### Procedure

1. Create a local database for machine-wide settings in **/etc/dconf/db/local.d/00-background**:

```
[org/gnome/desktop/background]
```

```
picture-uri='file:///usr/local/share/backgrounds/wallpaper.jpg' 1
picture-options='scaled' 2
primary-color='000000' 3
secondary-color='FFFFFF' 4
```

- 1 Specify the path to the desktop background image file.
- 2 Specify one of the rendering options for the background image:
  - **none**
  - **wallpaper**
  - **centered**
  - **scaled**
  - **stretched**
  - **zoom**
  - **spanned**
- 3 Specify the left or top color when drawing gradients or the solid color.
- 4 Specify the right or bottom color when drawing gradients.

2. Optional: If you want to prevent a user from changing the default background, override the user's setting in the **/etc/dconf/db/local.d/locks/background** file:

```
# List the keys used to configure the desktop background
/org/gnome/desktop/background/picture-uri
/org/gnome/desktop/background/picture-options
/org/gnome/desktop/background/primary-color
/org/gnome/desktop/background/secondary-color
```

3. Update the system databases:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

## Additional resources

- For more information about **GSettings**, see [Section 6.1, “Introduction to configuring GNOME”](#).

## 8.1.2. Adding extra backgrounds

You can make extra backgrounds available to users on your system.

### Procedure

1. Create the **/usr/share/gnome-background-properties/extra-backgrounds.xml** file.
2. In the new file, specify the extra background files and their appearance in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wallpapers SYSTEM "gnome-wp-list.dtd">
<wallpapers>
  <wallpaper deleted="false">
    <name>Background name</name>
    <filename>full-path-to-the-image</filename>
    <options>display-option</options>
    <shade_type>background-shade</shade_type>
    <pcolor>primary-color</pcolor>
    <scolor>secondary-color</scolor>
  </wallpaper>
</wallpapers>
```

3. The new backgrounds are now available to all users in the **Background** section of the **Settings** application.

## 8.1.3. Frequently used background schema keys

The following configuration controls the behavior of desktop backgrounds in the GSettings system.

**Table 8.1. Frequently used GSettings and XML background keys**

Key name	XML name	Possible values	Description
----------	----------	-----------------	-------------

Key name	XML name	Possible values	Description
<b>picture-options</b>	<b>options</b>	<ul style="list-style-type: none"> <li>• none</li> <li>• wallpaper</li> <li>• centered</li> <li>• scaled</li> <li>• stretched</li> <li>• zoom</li> <li>• spanned</li> </ul>	Determines how the image set by <b>wallpaper_filename</b> is rendered.
<b>color-shading-type</b>	<b>shade_type</b>	<ul style="list-style-type: none"> <li>• horizontal</li> <li>• vertical</li> <li>• solid</li> </ul>	Determines the shade of the background color.
<b>primary-color</b>	<b>pcolor</b>	default: <b>#023c88</b>	Left or Top color when drawing gradients, or the solid color.
<b>secondary-color</b>	<b>scolor</b>	default: <b>#5789ca</b>	Right or Bottom color when drawing gradients, not used for solid color.

#### Example 8.1. An extra backgrounds file with one<wallpaper> element

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wallpapers SYSTEM "gnome-wp-list.dtd">
<wallpapers>
  <wallpaper deleted="false">
    <name>Company Background</name>
    <name xml:lang="de">Firmenhintergrund</name>
    <filename>/usr/local/share/backgrounds/company-wallpaper.jpg</filename>
    <options>zoom</options>
    <shade_type>solid</shade_type>
    <pcolor>#ffffff</pcolor>
    <scolor>#000000</scolor>
  </wallpaper>
</wallpapers>

```

#### Example 8.2. An extra backgrounds file with two<wallpaper> elements

In one configuration file, you can specify multiple **<wallpaper>** elements to add more backgrounds as shown in the following example with two **<wallpaper>** elements, adding two different backgrounds.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wallpapers SYSTEM "gnome-wp-list.dtd">
<wallpapers>
  <wallpaper deleted="false">
    <name>Company Background</name>
    <name xml:lang="de">Firmenhintergrund</name>
    <filename>/usr/local/share/backgrounds/company-wallpaper.jpg</filename>
    <options>zoom</options>
    <shade_type>solid</shade_type>
    <pcolor>#ffffff</pcolor>
    <scolor>#000000</scolor>
  </wallpaper>
  <wallpaper deleted="false">
    <name>Company Background 2</name>
    <name xml:lang="de">Firmenhintergrund 2</name>
    <filename>/usr/local/share/backgrounds/company-wallpaper-2.jpg</filename>
    <options>zoom</options>
    <shade_type>solid</shade_type>
    <pcolor>#ff0000</pcolor>
    <scolor>#00ffff</scolor>
  </wallpaper>
</wallpapers>
```

### 8.1.4. Setting the screen shield

Screen shield is the screen that slides down when you lock the system. You can set a default screen shield picture for all users on the system.

#### Procedure

1. Create the **/etc/dconf/db/gdm.d/01-screensaver** file:

```
[org/gnome/desktop/screensaver]
picture-uri='file:///opt/corp/background.jpg'
```

Replace **/opt/corp/background.jpg** with the absolute path to the image file that you want to use as the default screen shield. Supported formats are PNG, JPG, JPEG, and TGA. Note that the screen shield scales your image if necessary to fit the screen.

2. Update the system databases:

```
# dconf update
```

3. Users must log out and back in again before the system-wide settings take effect.

#### Troubleshooting

- If the screen shield does not update:
  - a. Verify that you have updated the system databases:

```
# dconf update
```



b. Try restarting GDM:

```
# systemctl restart gdm.service
```



### WARNING

Restarting the **gdm** service terminates all currently running GNOME sessions of all desktop users who are logged in. This might result in users losing unsaved data.

### Additional resources

- For more information about **GSettings** and **dconf**, see [Section 6.1, “Introduction to configuring GNOME”](#).

## 8.2. CUSTOMIZING THE BRANDING OF THE LOGIN SCREEN

You can change the logo that appears on the GNOME login screen (GDM) using a **dconf** profile.

### Prerequisites

- Prepare an image file in any of the supported formats: ANI, BPM, GIF, GTIFF, ICNS, ICO, JPEG, JPEG 2000, PCX, PNM, PBM, PGM, PPM, RAS, SVG, TGA, TIFF, WBMP, XBM, or XPM.
- The image must be around 48 pixels in height. If it is significantly larger, it will exceed the logo area.
- Store the image file in a location that the **gdm** user can access. For example, select a universally readable system directory such as **/opt/** or **/usr/local/**.

### Procedure

1. Create the **/etc/dconf/db/gdm.d/01-override-logo** configuration file with the following content:

```
[org/gnome/login-screen]
logo='/path/to/logo.png'
```

Replace **/path/to/logo.png** with the full path to the image file that you want to use as the login screen logo.

2. Update the system databases:

```
# dconf update
```

### Verification

1. Log out or otherwise switch to the login screen.
2. Check if it displays the logo that you selected.

3. If the logo does not update, restart GDM:

```
# systemctl restart gdm
```



#### WARNING

Restarting the **gdm** service terminates all currently running GNOME sessions of all desktop users who are logged in. This might result in users losing unsaved data.

## 8.3. USING GNOME SHELL EXTENSIONS TO CUSTOMIZE DESKTOP ENVIRONMENT

GNOME Shell extensions allow for the customization of the default GNOME Shell interface and its parts, such as window management and application launching.



#### IMPORTANT

Before deploying third-party GNOME Shell extensions on Red Hat Enterprise Linux, make sure to read the following document to learn about the Red Hat support policy for third-party software: [How does Red Hat Global Support Services handle third-party software, drivers, and/or uncertified hardware/hypervisors?](#)

### 8.3.1. Overview of GNOME Shell extensions

The following is an overview of GNOME Shell extensions available on RHEL 8, including the name of the package providing a particular extension, and the description of what each extension does.

**Table 8.2. Overview of available GNOME Shell extensions**

Package name	Extension name	Description
gnome-shell-extension-apps-menu	apps-menu	<b>Applications</b> menu for GNOME Shell
gnome-shell-extension-top-icons	Top Icons	Show legacy icons on top
gnome-shell-extension-user-theme	user-theme	Support for custom themes in GNOME Shell
gnome-shell-extension-drive-menu	drive-menu	Drive status menu for GNOME Shell
gnome-shell-extension-window-list	window-list	Display a window list at the bottom of the screen in GNOME Shell

Package name	Extension name	Description
gnome-shell-extension-dash-to-dock	Dash to Dock	Dock for the Gnome Shell by micxgx.gmail.com
gnome-shell-extension-desktop-icons	Desktop Icons	Desktop icons support for the GNOME Classic experience
gnome-shell-extension-no-hot-corner	nohotcorner	Disable the hot corner in GNOME Shell
gnome-shell-extension-systemMonitor	systemMonitor	<b>System Monitor</b> for GNOME Shell
gnome-shell-extension-updates-dialog	Updates Dialog	Show a modal dialog when there are software updates
gnome-shell-extension-window-grouper	window-grouper	Keep windows that belong to the same process on the same workspace
gnome-shell-extension-panel-favorites	panel-favorites	Favorite launchers in GNOME Shell's top bar
gnome-shell-extension-windowsNavigator	windowNavigator	Support for keyboard selection of windows and workspaces in GNOME shell
gnome-shell-extension-auto-move-windows	Autom Move Windows	Assign specific workspaces to applications in GNOME Shell
gnome-shell-extension-launch-new-instance	launch-new-instance	Always launch a new application instance for GNOME Shell
gnome-shell-extension-workspace-indicator	workspace-indicator	Workspace indicator for GNOME Shell
gnome-shell-extension-disable-screenshield	Disable Screen Shield	Disable GNOME Shell screen shield if lock is disabled
gnome-shell-extension-native-window-placement	native-window-placement	Native window placement for GNOME Shell
gnome-shell-extension-screenshot-window-sizer	screenshot-window-sizer	Screenshot window sizer for GNOME Shell
gnome-shell-extension-horizontal-workspaces	horizontal-workspaces	Desktop icons support for the GNOME Classic experience

Package name	Extension name	Description
gnome-shell-extension-places-menu	places-menu	<b>Places</b> status menu for GNOME Shell
gnome-classic-session	–	GNOME Classic mode session

### 8.3.2. Enabling machine-wide extensions

#### Prerequisites

To make extensions available to all users on the system, install them in the **/usr/share/gnome-shell/extensions** directory.

#### Procedure

1. Create a local database file for machine-wide settings in **/etc/dconf/db/local.d/00-extensions**:

```
[org/gnome/shell]
# List all extensions that you want to have enabled for all users
enabled-extensions=['myextension1@myname.example.com',
'myextension2@myname.example.com']
```

The **enabled-extensions** key specifies the enabled extensions using the extensions' UUID (**myextension1@myname.example.com** and **myextension2@myname.example.com**).

2. Update the system databases:

```
# dconf update
```

3. Users must log out and back in again before the system-wide settings take effect.



#### NOTE

There is currently no way to enable additional extensions for users who have already logged in. This does not apply for existing users who have installed and enabled their own GNOME extensions.

### 8.3.3. Locking down enabled extensions

You can prevent the user from enabling or disabling extensions by locking down the **org.gnome.shell.enabled-extensions** key.

#### Procedure

1. Create a local database file for machine-wide settings in **/etc/dconf/db/local.d/00-extensions**:

```
[org/gnome/shell]
# List all extensions that you want to have enabled for all users
enabled-extensions=['myextension1@myname.example.com',
'myextension2@myname.example.com']
```

The **enabled-extensions** key specifies the enabled extensions using the extensions' UUID (**myextension1@myname.example.com** and **myextension2@myname.example.com**).

2. Override the user's setting and prevent the user from changing it in **/etc/dconf/db/local.d/locks/extensions**:

```
# Lock the list of mandatory extensions
/org/gnome/shell/enabled-extensions
```

3. Update the system databases:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

After locking down the **org.gnome.shell.enabled-extensions**, any extensions installed in **~/.local/share/gnome-shell/extensions** or **/usr/share/gnome-shell/extensions** that are not listed in the **org.gnome.shell.enabled-extensions** key will not be loaded by GNOME Shell, thus preventing the user from using them.

### 8.3.4. Setting up mandatory extensions

In GNOME Shell, you can provide a set of extensions that the user has to use.

#### Prerequisites

The extensions must be installed under the **/usr/share/gnome-shell/extensions** directory.

#### Procedure

1. Create a local database file for machine-wide settings in **/etc/dconf/db/local.d/00-extensions-mandatory**:

```
[org/gnome/shell]
# List all mandatory extensions
enabled-extensions=['myextension1@myname.example.com',
'myextension2@myname.example.com']
```

The **enabled-extensions** key specifies the enabled extensions using the extensions' UUID (**myextension1@myname.example.com** and **myextension2@myname.example.com**).

2. Override the user's setting and prevent the user from changing it in **/etc/dconf/db/local.d/locks/extensions-mandatory**:

```
# Lock the list of mandatory extensions
/org/gnome/shell/enabled-extensions
```

3. Update the system databases:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

## CHAPTER 9. CUSTOMIZING GNOME DESKTOP FEATURES

### 9.1. CHANGING THE LANGUAGE USING DESKTOP GUI

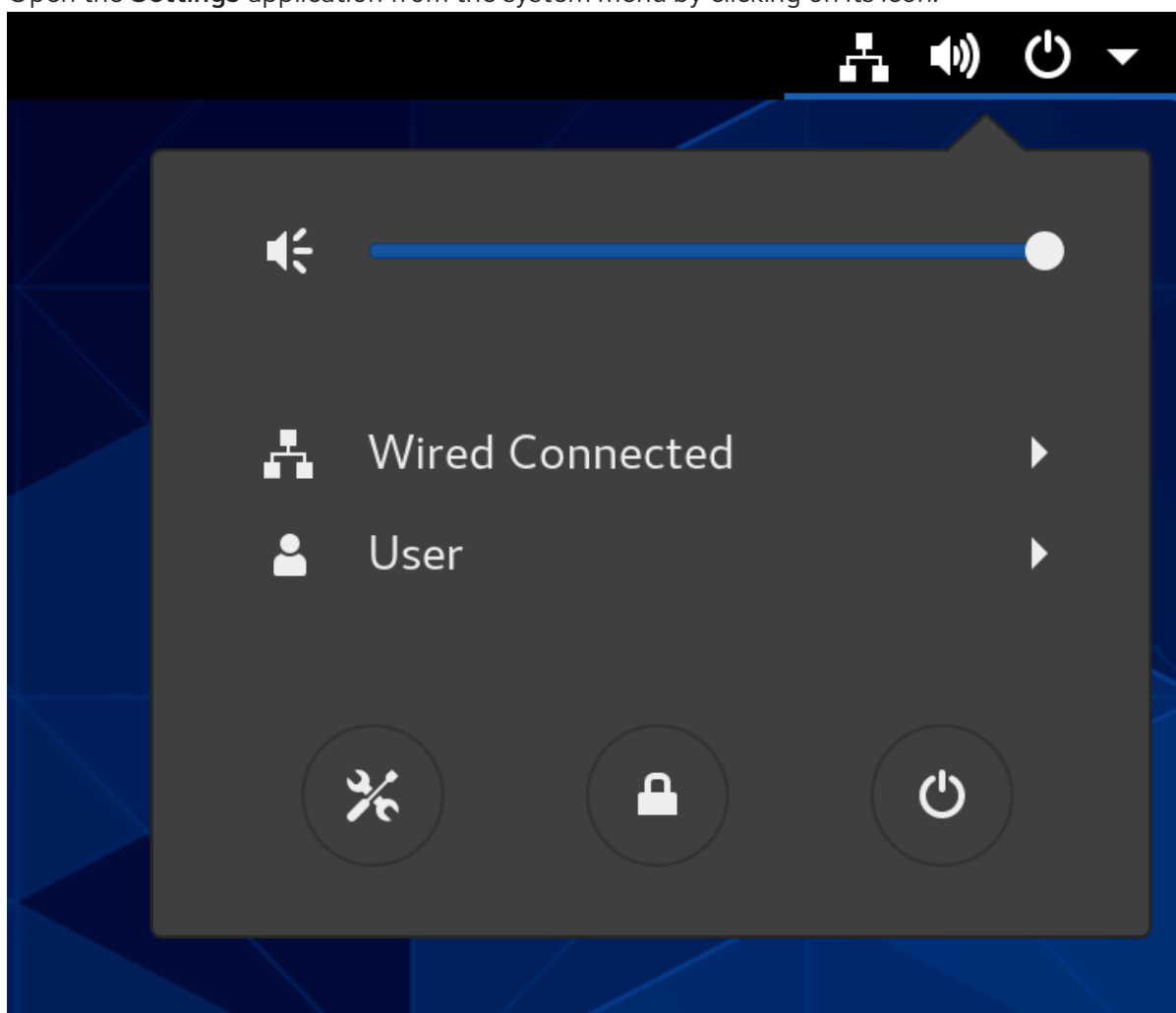
You can change the system language using the desktop GUI.

#### Prerequisites

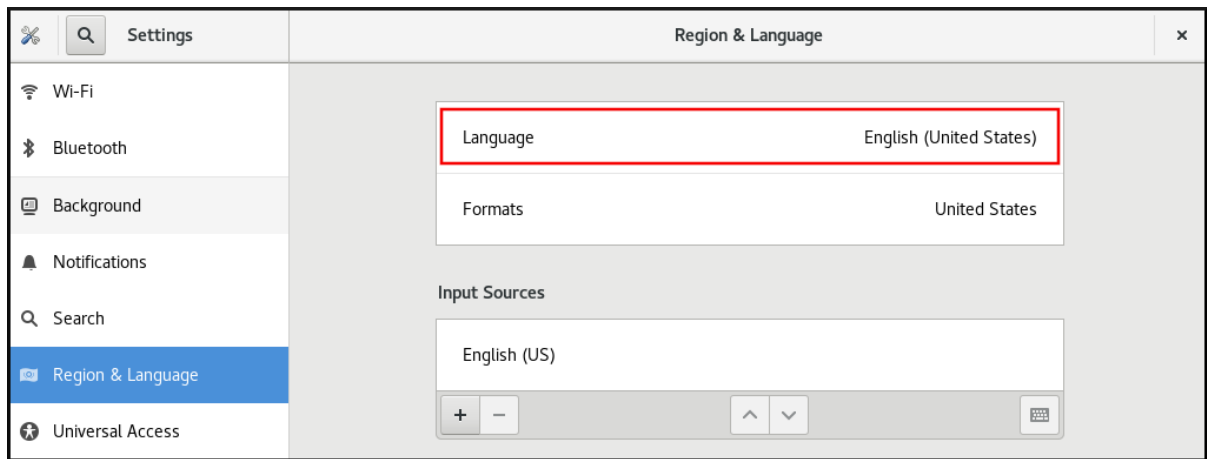
- Required language packages are installed on your system

#### Procedure

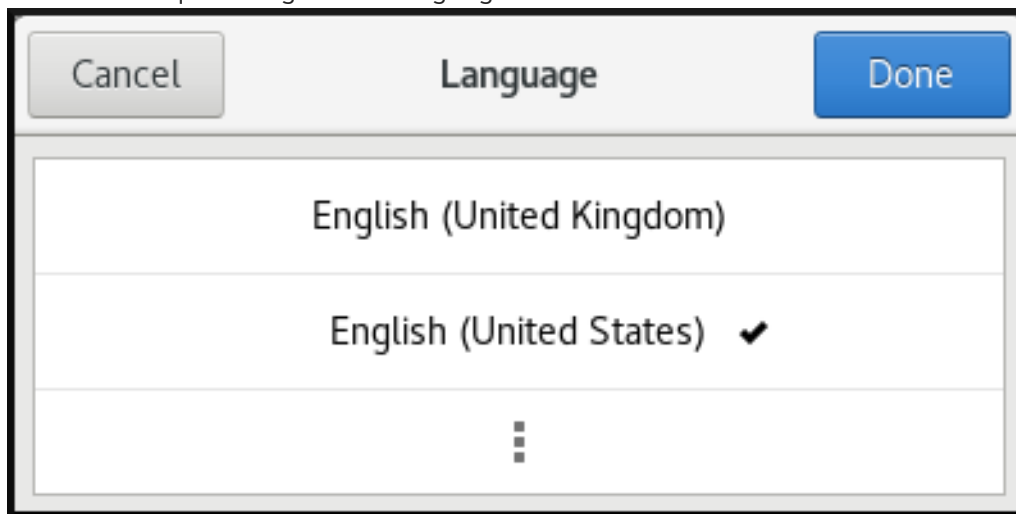
1. Open the **Settings** application from the system menu by clicking on its icon.



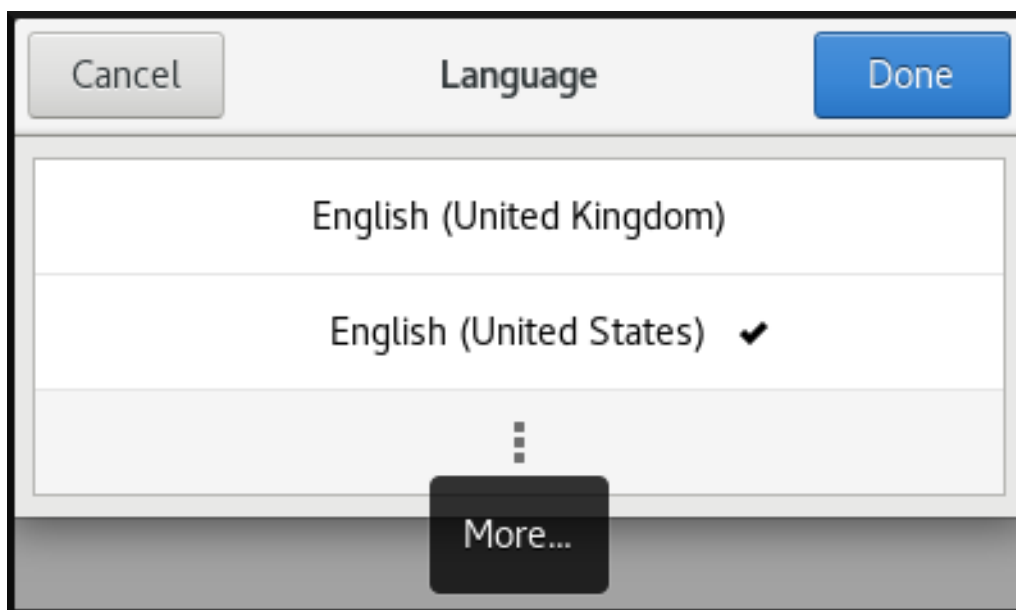
2. In **Settings**, choose **Region & Language** from the left vertical bar.
3. Click the **Language** menu.



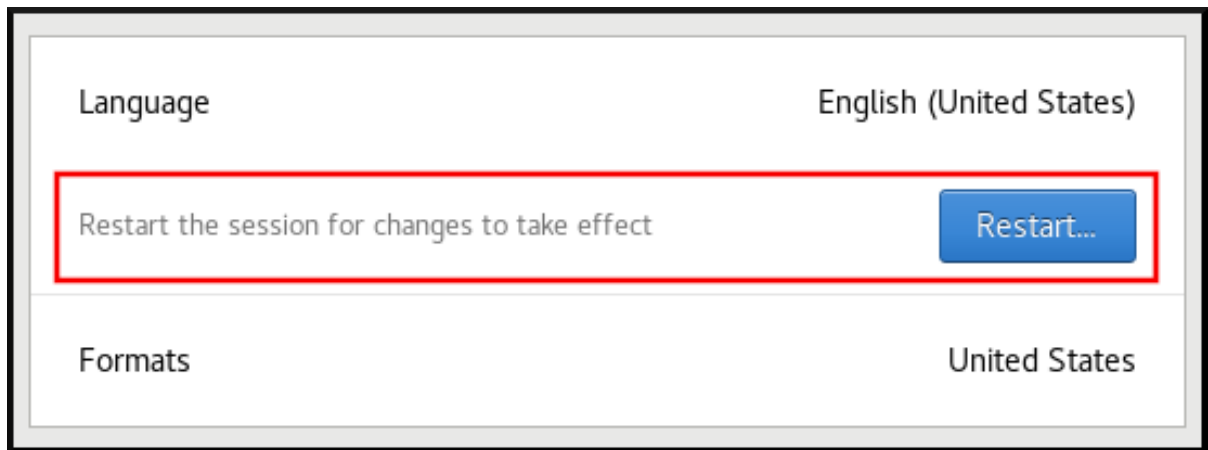
4. Select the required region and language from the menu.



If your region and language are not listed, scroll down, and click **More** to select from available regions and languages.



5. Click **Done**.
6. Click **Restart** for changes to take effect.



## NOTE

Some applications do not support certain languages. The text of an application that cannot be translated into the selected language remains in US English.

## Additional resources

- [Launching applications in GNOME](#)

## 9.2. ENABLING THE CTRL+ALT+BACKSPACE SHORTCUT

The **Ctrl+Alt+Backspace** shortcut key combination is used for terminating the **X.Org** display server.

You might want to terminate **X.Org** especially when:

- A program caused **X.Org** to stop working.
- You need to switch from your logged-in session quickly.
- You have launched a program that failed.
- You cannot operate in the current session.
- Your screen freezes.

To enable the **Ctrl+Alt+Backspace** shortcut to forcibly terminate **X.Org** by default for all users, you need to set the **org.gnome.desktop.input-sources.xkb-options** GSettings key.

## Procedure

1. Create a local database for machine-wide settings in **/etc/dconf/db/local.d/00-input-sources**:

```
[org/gnome/desktop/input-sources]
# Enable Ctrl-Alt-Backspace for all users
xkb-options=["terminate:ctrl_alt_bksp"]
```

2. Override the user's setting, and prevent the user from changing it in **/etc/dconf/db/local.d/locks/input-sources**:

```
# Lock the list of enabled XKB options
/org/gnome/desktop/input-sources/xkb-options
```



3. Update the system databases for the changes to take effect:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

If the **Ctrl+Alt+Backspace** key combination is enabled, all users can terminate **X.Org**, which brings them back to the login prompt.

## 9.3. DISABLING COMMAND-LINE ACCESS

To disable command-line access for a desktop user, you need to make configuration changes in a number of different contexts:

- [Section 9.3.1, “Setting the org.gnome.desktop.lockdown.disable-command-line Key”](#)
- [Section 9.3.2, “Disabling virtual terminal switching on X.Org”](#)
- Remove **Terminal** and any other application that provides access to the terminal from the **Applications menu** and **Activities Overview** in GNOME Shell. This is done by removing menu items for those applications.



### NOTE

The following steps do not remove the desktop user’s permissions to access a command line, but rather remove the ways that the desktop user could access command line.

### 9.3.1. Setting the org.gnome.desktop.lockdown.disable-command-line Key

This approach prevents the user from:

- Accessing the terminal
- Specifying a command line to be executed by using the **Alt+F2** command prompt

#### Procedure

1. Create a local database for machine-wide settings in **/etc/dconf/db/local.d/00-lockdown**:

```
[org/gnome/desktop/lockdown]
# Disable command-line access
disable-command-line=true
```

2. Override the user’s setting and prevent the user from changing it in **/etc/dconf/db/local.d/locks/lockdown**:

```
# Lock the disabled command-line access
/org/gnome/desktop/lockdown
```

3. Update the system databases:

```
# dconf update
```

- Users must log out and back in again before the system-wide settings take effect.

### 9.3.2. Disabling virtual terminal switching on X.Org

With the **X.Org** display server, users can normally use the **Ctrl+Alt+function** key shortcuts to switch from the GNOME Desktop and **X.Org** to a virtual terminal. You can disable access to all virtual terminals by modifying the **X.Org** configuration. The **X.Org** configuration should be modified by adding the **DontVTSwitch** option to the **Serverflags** section of an X configuration file in the **/etc/X11/xorg.conf.d/** directory, as shown by the following procedure.



#### IMPORTANT

You cannot apply the procedure if **GNOME Shell on Wayland** is used as the display server.

#### Procedure

- Create or edit an X configuration file in the **/etc/X11/xorg.conf.d/** directory:



#### NOTE

By convention, these host-specific configuration file names start with two digits and a hyphen and always have the **.conf** extension. Thus, the following file name can be **/etc/X11/xorg.conf.d/10-xorg.conf**.

```
Section "Serverflags"
```

```
Option "DontVTSwitch" "yes"
```

```
EndSection
```

- Restart the **X.Org** display server for the changes to take effect.

## 9.4. PREVENTING THE COMPUTER FROM SUSPENDING WHEN CLOSING THE LID

When closing the lid of your laptop, the computer by default suspends in order to save power. You can prevent the computer from suspending when closing the lid by changing the setting for that behavior.



#### WARNING

Some laptops can overheat if they are left running with the lid closed, especially if they are in a confined place. Therefore, consider whether changing the default setting from suspend to an other option is beneficial in your case.

#### Procedure

- Open the **/etc/systemd/logind.conf** file for editing.

- Find the **HandleLidSwitch=suspend** line in the file.  
If it is quoted out with the **#** character at the start, unquote it by removing **#**.

If the line is not present in the file, add it.

- Replace the default **suspend** parameter with:

- **lock** for the screen to lock
- **ignore** for nothing to happen
- **poweroff** for the computer to switch off

For example:

```
[Login]
HandleLidSwitch=lock
```

- Save your changes, and close the editor.
- Run the following command so that your changes preserve the next restart of the system:

```
# systemctl restart systemd-logind.service
```



#### WARNING

Restarting the service forcibly interrupts any currently running GNOME session of any desktop user who is logged in. This can result in users losing unsaved data.

For more information about the `/etc/systemd/logind.conf` file, see the **logind.conf** man page.

## 9.5. CHANGING BEHAVIOR WHEN PRESSING THE POWER BUTTON IN GRAPHICAL TARGET MODE

When the machine is booted to a graphical login screen or user session, hitting the **power** button makes the machine suspend by default. This happens both in cases when the user presses the **power** button physically or when pressing a virtual **power** button from a remote console. To achieve a different behavior when pressing the **power** button, set the function of this button with **dconf**.

For example, if you want the system to shutdown after pressing the **power** button, use the following procedure:

### Procedure

- Create a local database for system-wide settings in the `/etc/dconf/db/local.d/01-power` file:

```
[org/gnome/settings-daemon/plugins/power]
power-button-action='interactive'
```

2. Override the user's setting, and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/01-power** file:

```
/org/gnome/settings-daemon/plugins/power/power-button-action
```

3. Update the system databases:

```
# dconf update
```

4. Log out and back in again before the system-wide settings take effect.

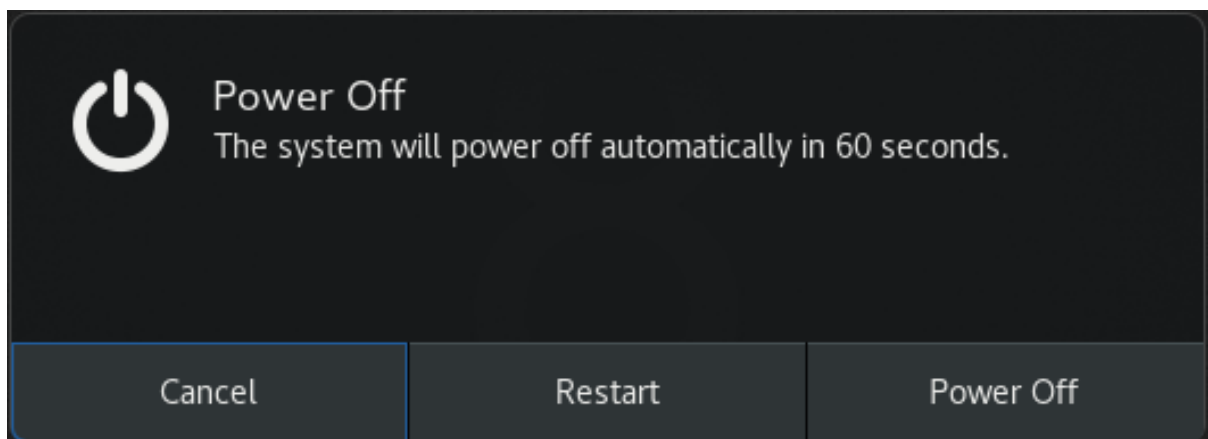
This configuration initiates a system shutdown after pressing the **power** button. To configure the system differently, you can set the behavior of particular buttons.

Options for particular buttons:

- nothing  
does nothing
- suspend  
suspends the system
- hibernate  
hibernates the system
- interactive  
shows a pop-up query asking the user what to do

With interactive mode, the system powers off automatically after 60 seconds when hitting the power button. However, you can choose a different behavior from the pop-up query as shown in the figure below.

#### Pop-up query for interactive mode



## CHAPTER 10. DISPLAYING THE SYSTEM SECURITY CLASSIFICATION

As an administrator of deployments where the user must be aware of the security classification of the system, you can set up a notification of the security classification. This can be either a permanent banner or a temporary notification, and it can appear on login screen, in the GNOME session, and on the lock screen.

### 10.1. ENABLING SYSTEM SECURITY CLASSIFICATION BANNERS

You can create a permanent classification banner to state the overall security classification level of the system. This is useful for deployments where the user must always be aware of the security classification level of the system that they are logged into.

The permanent classification banner can appear within the running session, the lock screen, and login screen, and customize its background color, its font, and its position within the screen.

This procedure creates a red banner with a white text placed on both the top and bottom of the login screen.

#### Procedure

1. Install the **gnome-shell-extension-classification-banner** package:

```
# yum install gnome-shell-extension-classification-banner
```



#### NOTE

The package is only available in RHEL 8.6 and later.

2. Create the **99-class-banner** file at either of the following locations:
  - To configure a notification at the login screen, create **/etc/dconf/db/gdm.d/99-class-banner**.
  - To configure a notification in the user session, create **/etc/dconf/db/local.d/99-class-banner**.
3. Enter the following configuration in the created file:

```
[org/gnome/shell]
enabled-extensions=['classification-banner@gnome-shell-extensions.gcampax.github.com']

[org/gnome/shell/extensions/classification-banner]
background-color='rgba(200,16,46,0.75)'
message='TOP SECRET'
top-banner=true
bottom-banner=true
system-info=true
color='rgb(255,255,255)'
```

**WARNING**

This configuration overrides similar configuration files that also enable an extension, such as [Notifying of the system security classification](#).

To enable multiple extensions, specify all of them in the **enabled-extensions** list. For example:

```
enabled-extensions=['heads-up-display@gnome-shell-
extensions.gcampax.github.com', 'classification-banner@gnome-shell-
extensions.gcampax.github.com']
```

4. Update the **dconf** database:

```
# dconf update
```

5. Reboot the system.

**Troubleshooting**

- If the classification banners are not displayed for an existing user, log in as the user and enable the **Classification banner** extension using the **Tweaks** application.

**10.2. NOTIFYING OF THE SYSTEM SECURITY CLASSIFICATION**

You can set up a notification that contains a predefined message in an overlay banner. This is useful for deployments where the user is required to read the security classification of the system before logging in.

Depending on your configuration, the notification can appear at the login screen, after logging in, on the lock screen, or after a longer time with no user activity. You can always dismiss the notification when it appears.

**Procedure**

1. Install the **gnome-shell-extension-heads-up-display** package:

```
# yum install gnome-shell-extension-heads-up-display
```

2. Create the **99-hud-message** file at either of the following locations:
  - To configure a notification at the login screen, create **/etc/dconf/db/gdm.d/99-hud-message**.
  - To configure a notification in the user session, create **/etc/dconf/db/local.d/99-hud-message**.
3. Enter the following configuration in the created file:

```
[org/gnome/shell]
enabled-extensions=['heads-up-display@gnome-shell-extensions.gcampax.github.com']

[org/gnome/shell/extensions/heads-up-display]
message-heading="Security classification title"
message-body="Security classification description"
# The following options control when the notification appears:
show-when-locked=true
show-when-unlocking=true
show-when-unlocked=true
```

Replace the following values with text that describes the security classification of your system:

#### ***Security classification title***

A short heading that identifies the security classification.

#### ***Security classification description***

A longer message that provides additional details, such as references to various guidelines.



#### **WARNING**

This configuration overrides similar configuration files that also enable an extension, such as [Enabling system security classification banners](#).

To enable multiple extensions, specify all of them in the **enabled-extensions** list. For example:

```
enabled-extensions=['heads-up-display@gnome-shell-
extensions.gcampax.github.com', 'classification-banner@gnome-shell-
extensions.gcampax.github.com']
```

4. Update the **dconf** database:

```
# dconf update
```

5. Reboot the system.

### **Troubleshooting**

- If the notifications are not displayed for an existing user, log in as the user and enable the **Heads-up display message** extension using the **Tweaks** application.

## CHAPTER 11. ENABLING CHINESE, JAPANESE, OR KOREAN TEXT INPUT

If you write with Chinese, Japanese, or Korean characters, you can configure RHEL to input text in your language.

### 11.1. INPUT METHODS

Certain scripts, such as Chinese, Japanese, or Korean, require keyboard input to go through an Input Method Engine (IME) to enter native text.

An input method is a set of conversion rules between the text input and the selected script. An IME is a software that performs the input conversion specified by the input method.

To input text in these scripts, you must set up an IME. If you installed the system in your native language and selected your language at the **GNOME Initial Setup** screen, the input method for your language is enabled by default.

### 11.2. AVAILABLE INPUT METHOD ENGINES

The following input method engines (IMEs) are available on RHEL from the listed packages:

**Table 11.1. Available input method engines**

Languages	Scripts	IME name	Package
Chinese	Simplified Chinese	Intelligent Pinyin	<b>ibus-libpinyin</b>
Chinese	Traditional Chinese	New Zhuyin	<b>ibus-libzhuyin</b>
Japanese	Kanji, Hiragana, Katakana	Kana Kanji	<b>ibus-kkc</b>
Korean	Hangul	Hangul	<b>ibus-hangul</b>
Other	Various	M17N	<b>ibus-m17n</b>

### 11.3. INSTALLING INPUT METHOD ENGINES

This procedure installs input method engines (IMEs) that you can use to input Chinese, Japanese, and Korean text.

#### Procedure

- Install all available input method packages:

```
# yum install @input-methods
```

### 11.4. SWITCHING THE INPUT METHOD IN GNOME



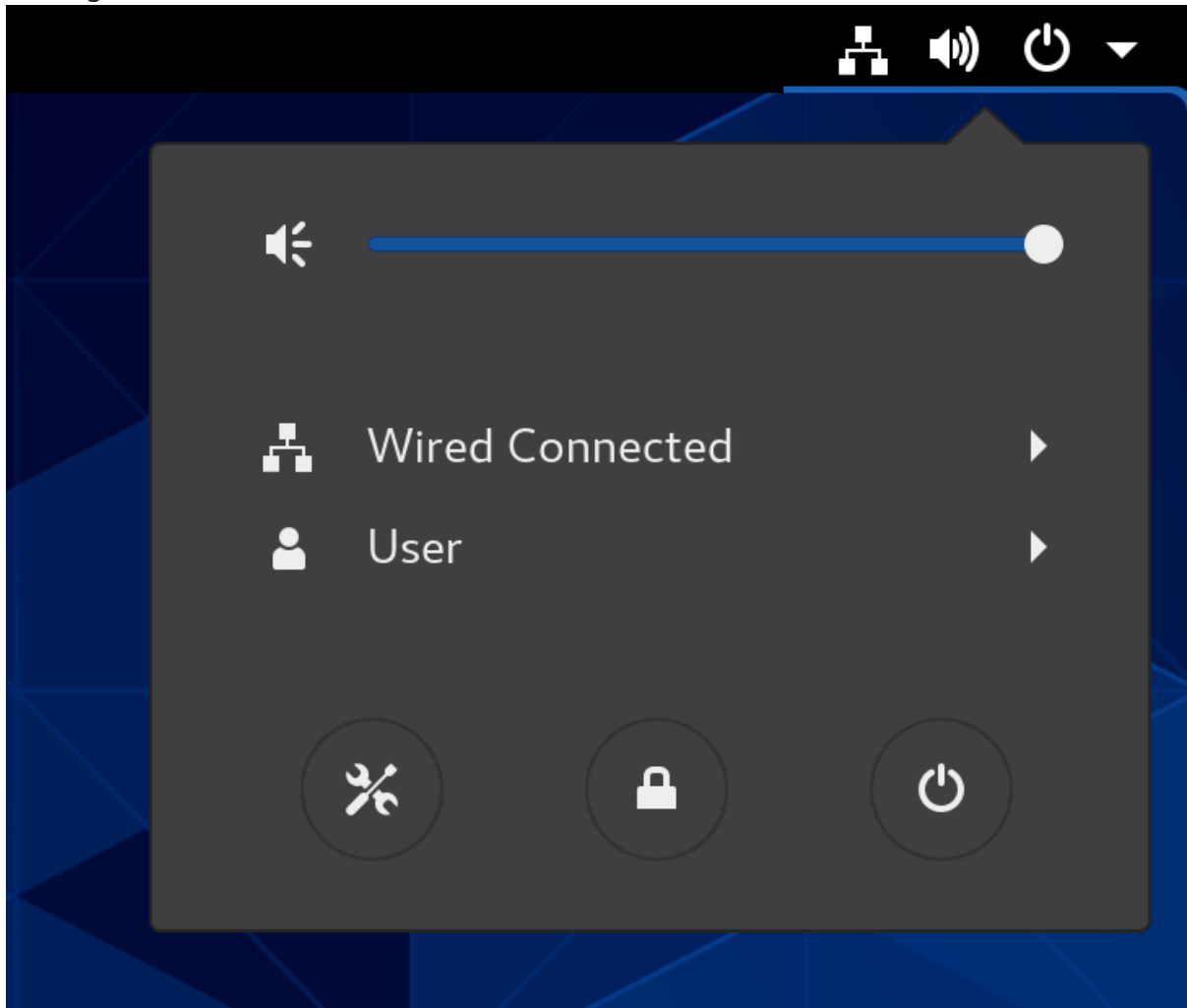
This procedure sets up the input method for your script, such as for Chinese, Japanese, or Korean scripts.

### Prerequisites

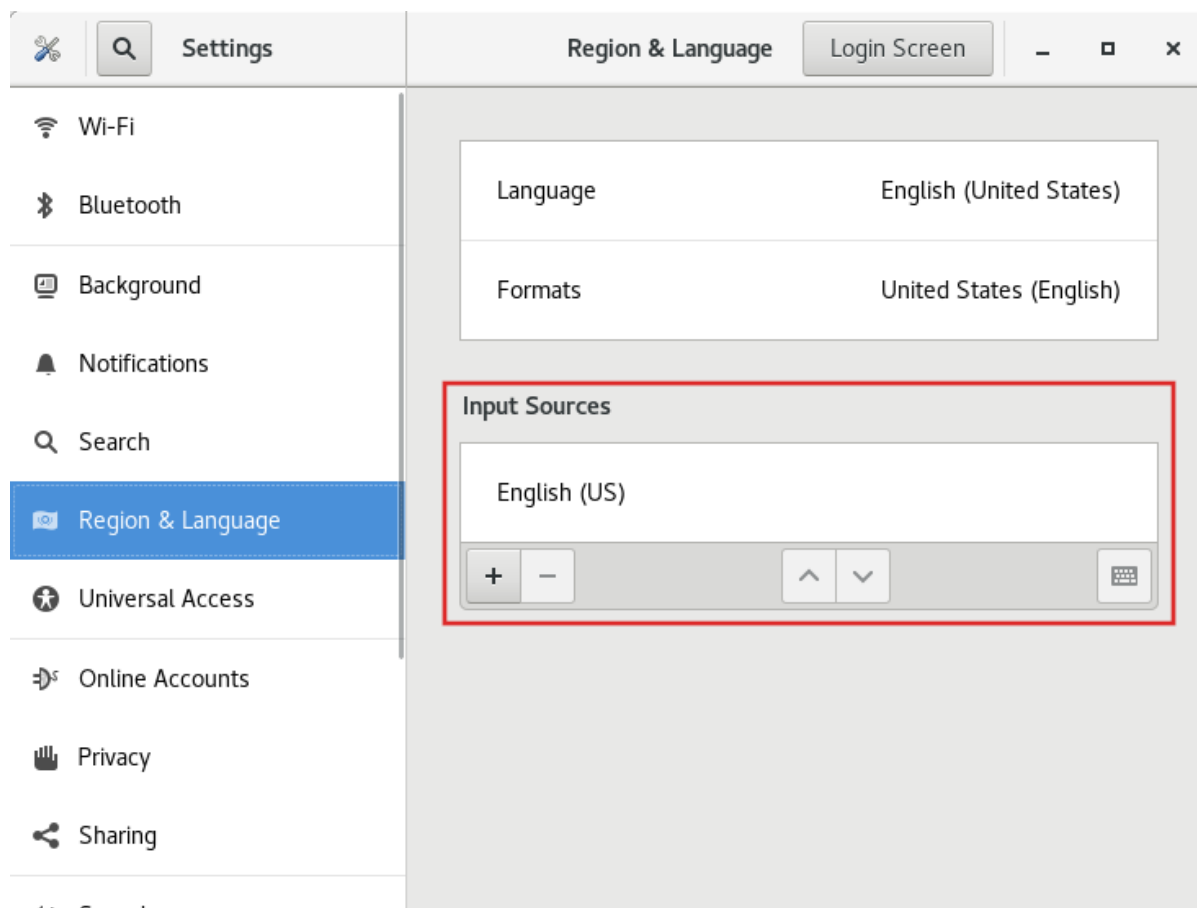
- The input method packages are installed.

### Procedure

1. Go to the **system menu**, which is accessible from the top-right screen corner, and click **Settings**.

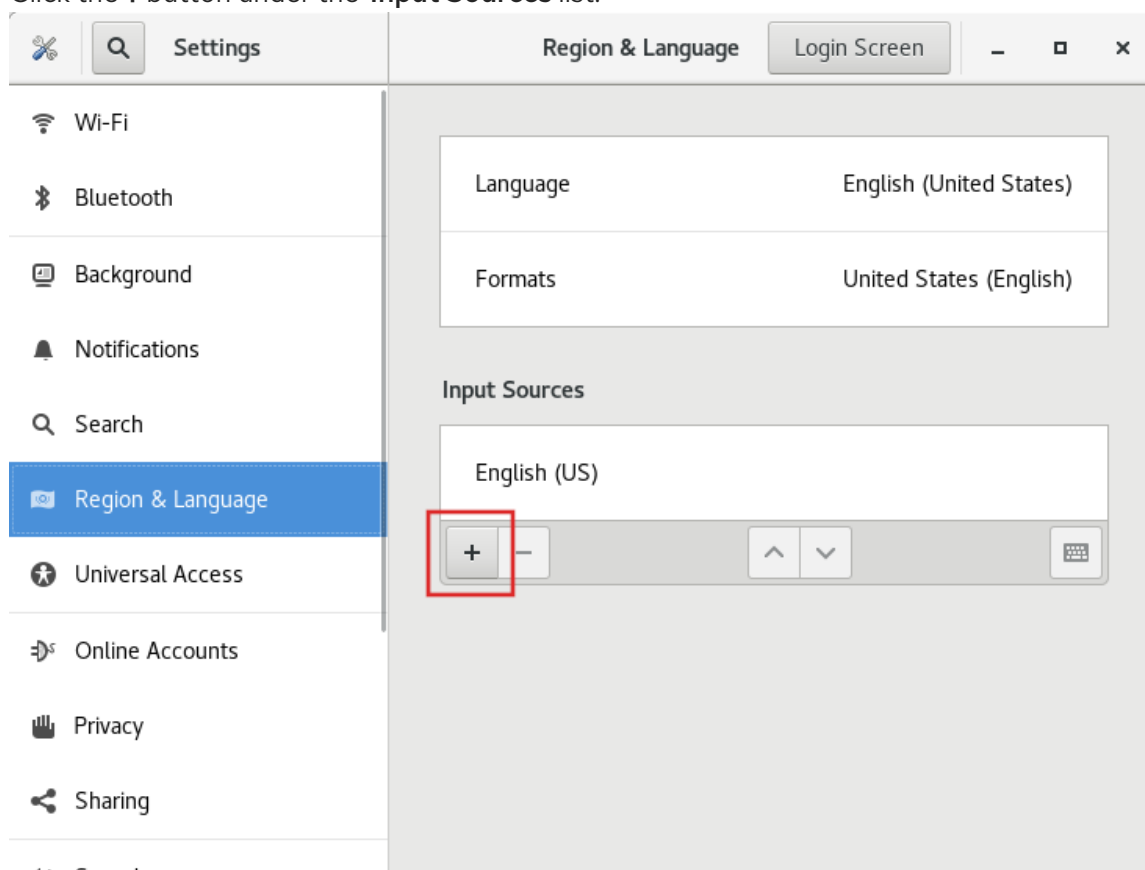


2. Select the **Region & Language** section.
3. In the **Input Sources** list, review the currently enabled input methods.

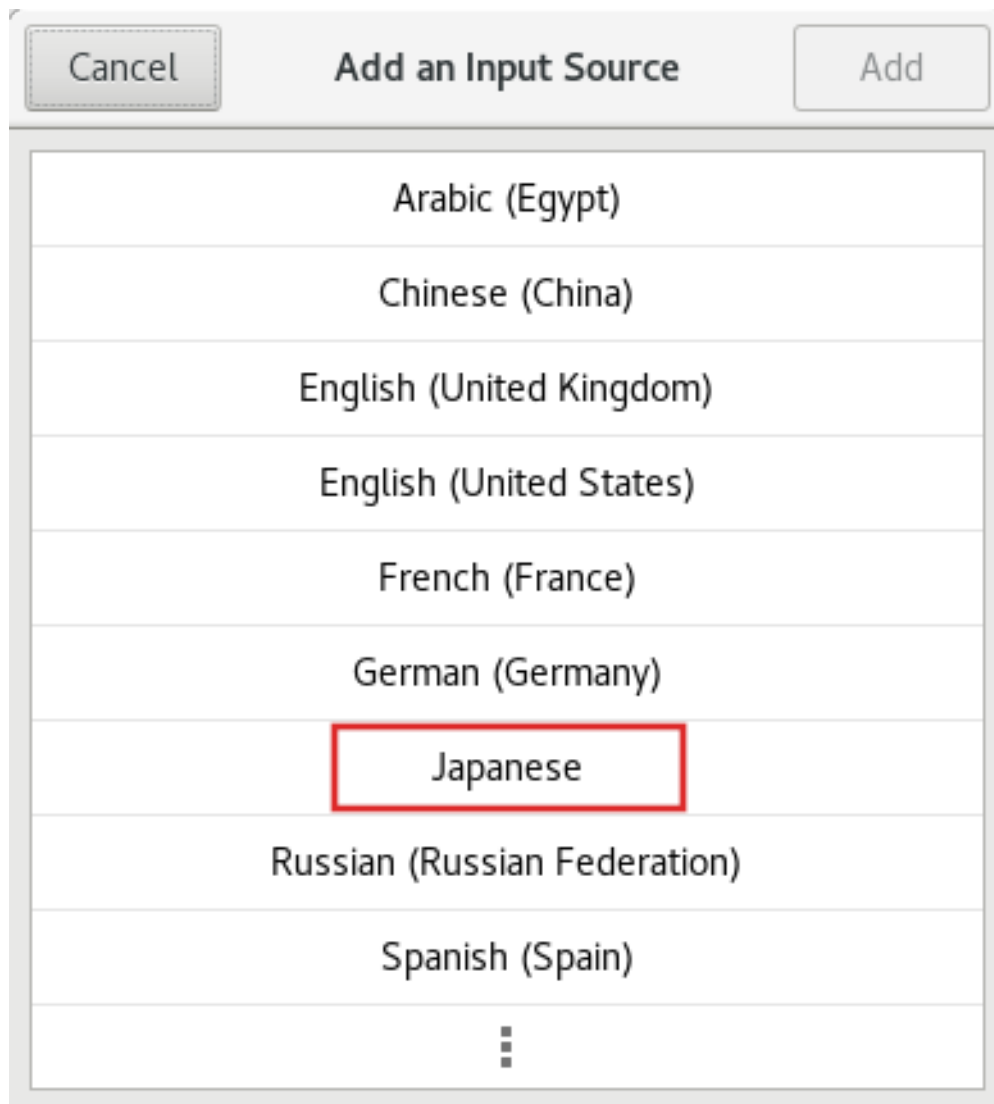


4. If your input method is missing:

- a. Click the **+** button under the **Input Sources** list.

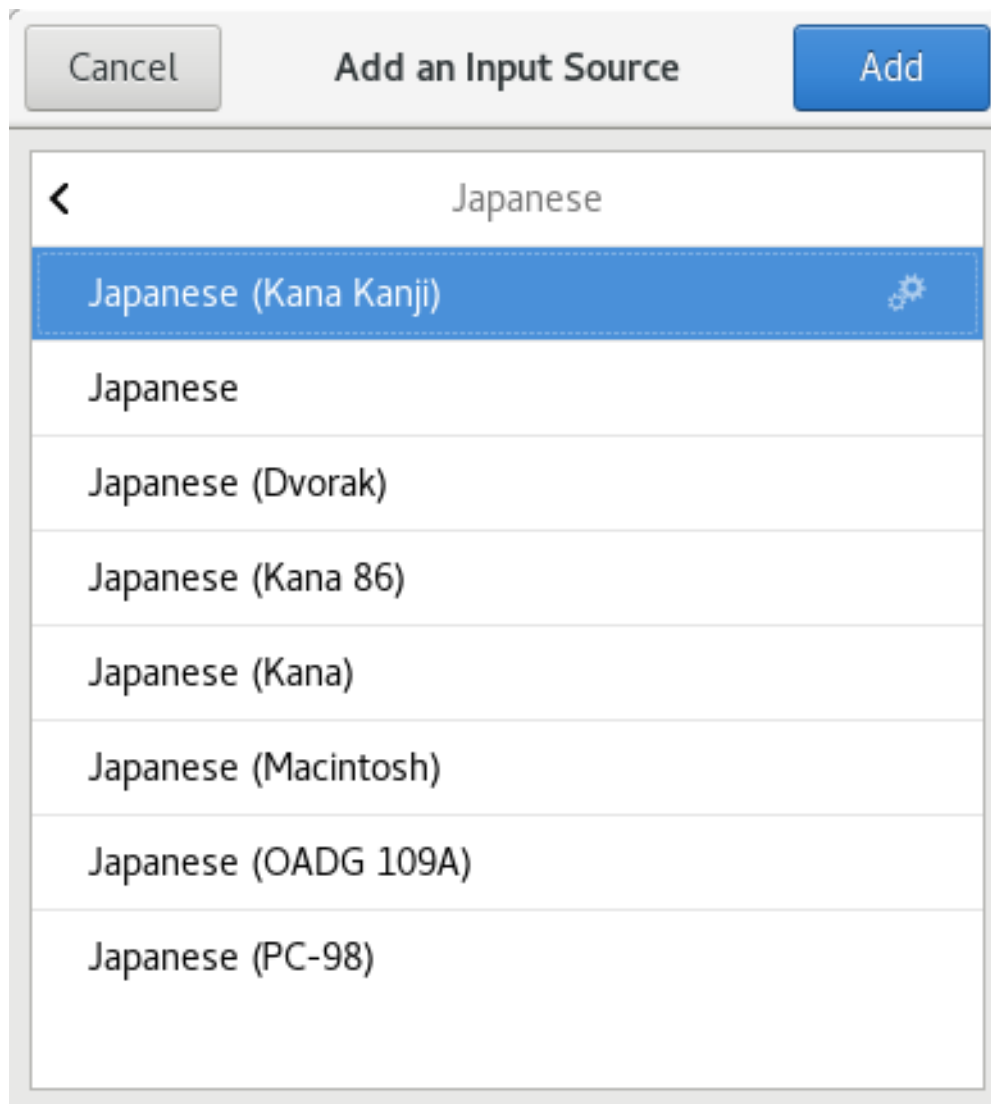


- b. Select your language.

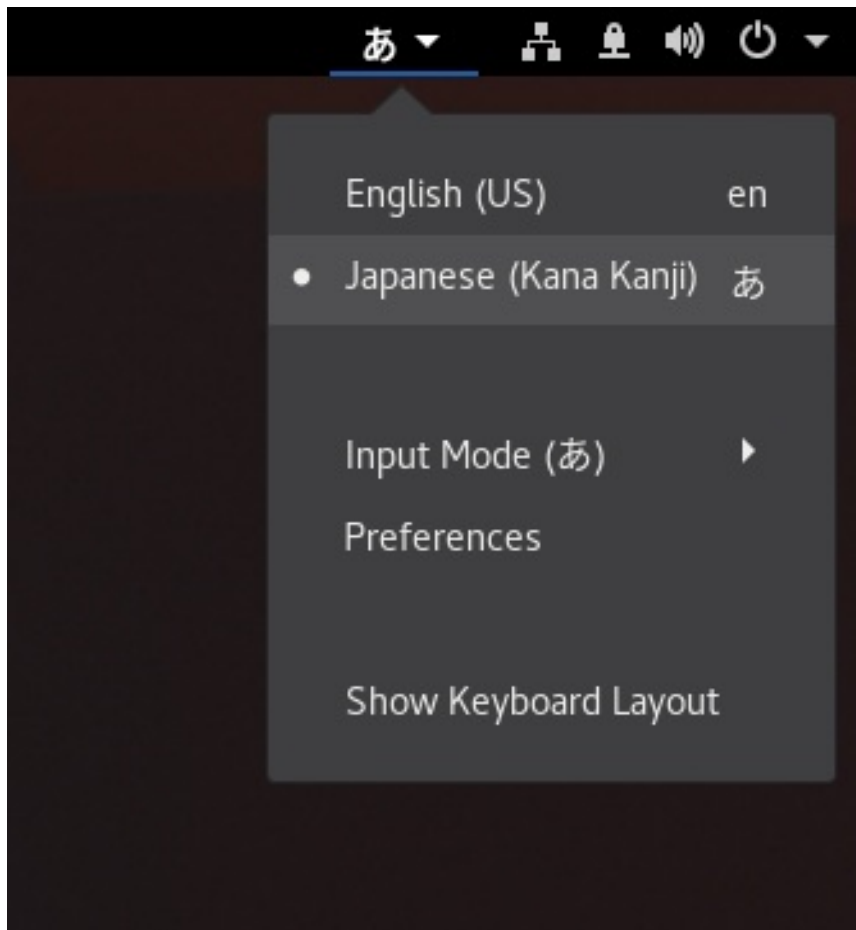
**NOTE**

If you cannot find your language in the menu, click the three dots icon (**More...**) at the end of the menu.

- c. Select the input method that you want to use. A cog wheel icon marks all input methods to distinguish them from simple keyboard layouts.



- d. Confirm your selection by clicking **Add**.
5. Switch the active input method using one of the following ways:
  - Click the input method indicator on the right side of the top panel and select your input method.



- Switch between the enabled input methods using the **Super+Space** keyboard shortcut.

### Verification

1. Open a text editor.
2. Type text in your language.
3. Verify that the text appears in your native script.

## CHAPTER 12. USING SPECIAL CHARACTERS IN GNOME

In GNOME, you can use the Compose Key to type special characters from different languages and symbol sets, including those not available on your keyboard. You can enter and view special characters from different languages and symbol sets, making it easy to work with diverse character sets in GNOME.

To input these special characters, you can define one of the existing keys on your keyboard as a Compose Key. Once enabled, the Compose Key allows you to type special characters and symbols by pressing multiple keys in a specific sequence.

### 12.1. ENABLING THE COMPOSE KEY FOR AN INDIVIDUAL USER WITH THE TWEAKS APPLICATION

To enable the Compose key for an individual user by Tweaks application follow these steps.

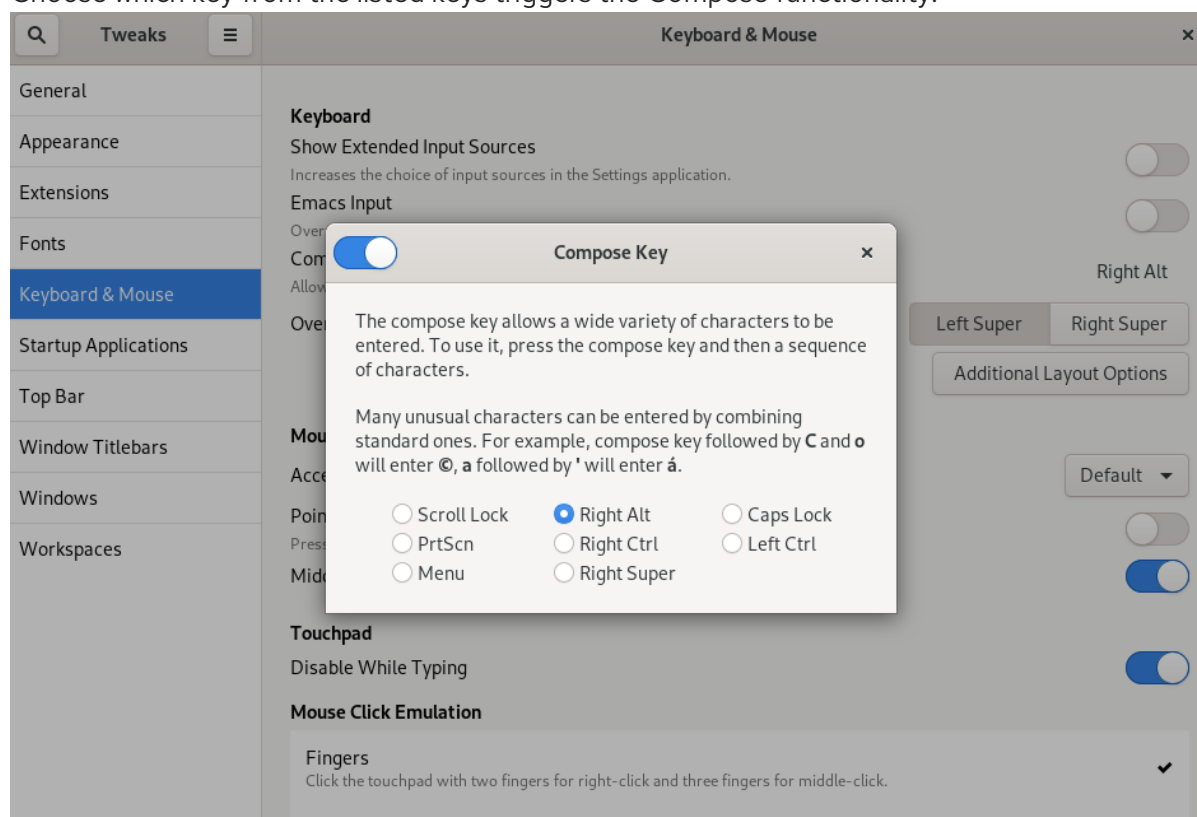
#### Prerequisites

- The **Tweaks** application is installed on your system.

```
# yum install gnome-tweaks
```

#### Procedure

1. Open the **Tweaks** application.
2. Select **Keyboard & Mouse** in the side bar.
3. Enable **Compose Key**.
4. Choose which key from the listed keys triggers the Compose functionality.



## 12.2. ENABLING THE COMPOSE KEY FOR ANOTHER USER

You can enable the Compose Key for another user with the **gsettings** utility.

### Prerequisites

- Administrative access.

### Procedure

1. Allow all clients to connect to the X server:

```
# xhost +
```

2. Run the following command to set the Compose Key:

```
# su - <username> -c "gsettings set org.gnome.desktop.input-sources xkb-options \"
['compose:<compose_key>']\""
```

Replace **<username>** with the username of the user for whom you want to enable the Compose Key. Replace **<compose\_key>** with the key you want to use as the Compose Key. You can use the **ra**lt option to designate the right **Alt** key as the Compose Key.

To see other Compose Key options that you can use to set up a Compose Key on your keyboard, use:

```
$ grep compose /usr/share/X11/xkb/rules/evdev.lst
```

3. Resets the access control:

```
# xhost -
```

### Verification

- To check the Compose Key settings for another user, use:

```
# su - <username> -c "gsettings get org.gnome.desktop.input-sources xkb-options"
```

Replace **<username>** with the username of the user for whom you want to check the Compose Key setting.

## 12.3. ENABLING THE COMPOSE KEY FOR ALL USERS

To enable the Compose key for all users, follow these steps.

### Procedure

1. Open the **/etc/dconf/db/local.d/00-input-sources** file as the **root** user.
2. Enter the following content in the file:

```
[org/gnome/desktop/input-sources]
# Enable the Compose key
xkb-options=['compose:selected-key']
```

Replace **selected-key** with the key that triggers Compose. For example, use **ralt** to set the right **Alt** key.

- To override the user's Compose settings and prevent the user from changing them, create the **/etc/dconf/db/local.d/locks/input-sources** file and enter the following:

```
# Lock the list of enabled XKB options
/org/gnome/desktop/input-sources/xkb-options
```

- Update the system databases for the changes to take effect:

```
# dconf update
```

- Users must log out and back in again before the system-wide settings take effect.

### Additional resources

- See the **xkeyboard-config(7)** man page for the names of other keys.

## 12.4. COMPOSE KEY SEQUENCES FOR SPECIAL CHARACTERS

The table showcases Compose Key sequences used to input special characters with diacritics or accents in GNOME. Each row displays a Compose Key sequence alongside its corresponding result

Table 12.1. Compose Key sequences for special characters

Compose Key Sequence	Result
<b>Compose+'+letter</b>	Letter with acute accent (é, á, ñ)
<b>Compose+`+letter</b>	Letter with grave accent (è, ù, ò)
<b>Compose++letter</b>	Letter with umlaut or diaeresis (ë, ö, ü)
<b>Compose+--+letter</b>	Letter with macron (ā, ē, ō)
<b>Compose+/-+letter</b>	Letter with stroke or diacritic (ø, †, ø)
<b>Compose+=+letter</b>	Letter with double acute accent (ő, ű, ä)
<b>Compose+.,+letter</b>	Letter with dot above (à, ç, ď)
<b>Compose+;,+letter</b>	Letter with cedilla (ç, ș, ț)



Compose Key Sequence	Result
<b>Compose+<sup>^</sup>+letter</b>	Letter with circumflex accent (â, ê, î)
<b>Compose+<sup>~</sup>+letter</b>	Letter with tilde accent (ã, ñ, õ)

## CHAPTER 13. TYPING EMOJI CHARACTERS

You can type emoji characters using several different methods in GNOME, depending on the type of the application.

### 13.1. TYPING EMOJI CHARACTERS IN GTK APPLICATIONS

This procedure inserts an emoji character in an application that uses the GTK graphical toolkit, such as in native GNOME applications.

#### Prerequisites

- Make sure that the application is built on the GTK toolkit.

#### Procedure

1. Open a GTK application.
2. Make sure that a text field is active.
3. Press **Ctrl+.**.  
The emoji selection menu opens.
4. Browse the emoji characters or type a keyword that identifies the emoji character that you want to insert, such as **smile**.  
For the full list of keywords associated with emoji characters, see the *Other Keywords* column on the [Emoji List](#) page.
5. Click the selected character, or navigate to it using the cursor keys and press **Enter**.

#### Verification

- Check that the intended emoji character now appears at your cursor.

### 13.2. TYPING EMOJI CHARACTERS IN ANY APPLICATIONS

This procedure inserts an emoji character in any application, regardless of the graphical toolkit that the application uses.

#### Procedure

1. Open an application.
2. Make sure that a text field is active.
3. Press **Ctrl+Shift+e**.  
The underscored letter **e** appears at your cursor.
4. Type a keyword that identifies the emoji character that you want to insert, such as **smile**.  
For the full list of keywords associated with emoji characters, see the *Other Keywords* column on the [Emoji List](#) page.
5. Repeatedly press **Space** to browse the emoji characters that match your keyword.

6. Confirm the selected emoji character by pressing **Enter**.

#### Verification

- Check that the intended emoji character now appears at your cursor.

## CHAPTER 14. MANAGING STORAGE VOLUMES IN GNOME

This section describes how you can manage storage volumes in GNOME with a virtual file system. GNOME Virtual File System (GVFS) is an extension of the virtual file system interface provided by the libraries the GNOME desktop is built on.

### 14.1. THE GVFS SYSTEM

GVFS provides complete virtual file system infrastructure and handles storage in the GNOME desktop. It uses addresses for full identification based on the Uniform Resource Identifier (URI) standard, syntactically similar to URL addresses in web browsers. These addresses in the form of ***schema://user@server/path*** are the key information determining the kind of service.

GVFS helps to mount the resources. These mounts are shared between multiple applications. Resources are tracked globally within the running desktop session, which means that even if you quit an application that triggered the mount, the mount continues to be available for any other application. Multiple applications can access the mount at the same time unless it is limited by a back end. Some protocols by design permit only a single channel.

GVFS mounts removable media in the ***/run/media/*** directory.

### 14.2. THE FORMAT OF THE GVFS URI STRING

You must form a URI string to use back end services. This string is a basic identifier used in GVFS, which carries all necessary information needed for unique identification, such as type of service, back end ID, absolute path, or user name if required. You can see this information in the **Files** address bar and GTK+ open or save file dialog.

The following example is a very basic form of the URI string, which points to a root directory (/) of the File Transfer Protocol (FTP) server running at the **ftp.myserver.net** domain:

#### Example 14.1. A URI string pointing to the root FTP directory

```
ftp://ftp.myserver.net/
```

#### Example 14.2. A URI string pointing to a text file on FTP

```
ssh://joe@ftp.myserver.net/home/joe/todo.txt
```

### 14.3. MOUNTING A STORAGE VOLUME IN GNOME

You can manually mount a local storage volume or a network share in the **Files** application.

#### Procedure

1. Open the **Files** application.
2. Click **Other Locations** in the side bar.  
The window lists all connected storage volumes and all network shares that are publicly available on your local area network.

If you can see the volume or network share in this list, mount it by clicking the item.

If you want to connect to a different network share, use the following steps.

3. Enter the GVFS URI string to the network share in the **Connect to Server** field.
4. Press **Connect**.
5. If the dialog asks you for login credentials, enter your name and password into the relevant fields.
6. When the mounting process finishes, you can browse the files on the volume or network share.

## 14.4. UNMOUNTING A STORAGE VOLUME IN GNOME

You can unmount a storage volume, a network share, or another resource in the **Files** application.

### Procedure

1. Open the **Files** application.
2. In the side bar, click the **Unmount** (⏏) icon next to the chosen mount.
3. Wait until the mount disappears from the side bar or a notification about the safe removal appears.

## 14.5. ACCESS TO GVFS MOUNTS IN THE FILE SYSTEM

Learn more about FUSE, the main daemon for the GVFS virtual file system.

Applications built with the GIO library can access GVFS mounts. In addition, GVFS provides a FUSE daemon which exposes active GVFS mounts. Any application can access active GVFS mounts using the standard POSIX APIs as though mounts were regular file systems.

In certain applications, additional library dependency and new virtual file system (VFS) subsystem specifics might be unsuitable or too complicated. For such reasons and to boost compatibility, GVFS provides a File System in Userspace (FUSE) daemon, which exposes active mounts through its mount for standard Portable Operating System Interface (POSIX) access. This daemon transparently translates incoming requests to imitate a local file system for applications.



### IMPORTANT

You might experience difficulties with certain combinations of applications and GVFS back ends.

The FUSE daemon starts automatically with the main **gvfs** daemon and mounts volumes either in the **/run/user/UID/gvfs/** or **~/.gvfs/** directories as a fallback.

Manual browsing shows individual directories for each GVFS mount. The system passes the transformed path as an argument when you are opening documents from GVFS locations with non-native applications. Note that native GIO applications automatically translate this path back to a native URI.

## 14.6. AVAILABLE GIO COMMANDS

GIO provides several commands that might be useful for scripting or testing.

Here is a set of POSIX commands counterparts as follows:

Command	Description
<b>gio cat</b>	Displays the content of a file.
<b>gio mkdir</b>	Creates a new directory.
<b>gio rename</b>	Renames a file.
<b>gio mount</b>	Provides access to various aspects of the <b>gio</b> mounting functionality.
<b>gio set</b>	Sets a file attribute on a file.
<b>gio copy</b>	Makes a copy of a file.
<b>gio list</b>	Lists directory contents.
<b>gio move</b>	Moves a file from one location to another.
<b>gio remove</b>	Removes a file.
<b>gio trash</b>	Sends files or directories to the <b>Trashcan</b> . This can be a different folder depending on where the file is located, and not all file systems support this concept. In the common case that the file lives inside a user's home directory, the trash folder is <b>\$XDG_DATA_HOME/Trash</b> .
<b>gio info</b>	Displays information of the given locations.
<b>gio save</b>	Reads from standard input and saves the data to the given location.
<b>gio tree</b>	Lists the contents of the given locations recursively, in a tree-like format. If no location is given, it defaults to the current directory.

Following additional commands provide more control of GIO specifics:

Command	Description
<b>gio monitor</b>	Monitors files or directories for changes, such as creation, deletion, content and attribute changes, and mount and unmount operations affecting the monitored locations.
<b>gio mime</b>	Lists the registered and recommended applications for the mimetype if no handler is given, else, it is set as the default handler for the mimetype.

Command	Description
<b>gio open</b>	Opens files with the default application that is registered to handle files of this type.

**NOTE**

For user convenience, **bash** completion is provided as a part of the package.

All these commands are native GIO clients, there is no need for the fallback FUSE daemon to be running. Their purpose is not to be drop-in replacements for POSIX commands, in fact, a very little range of switches is supported. In their basic form, these commands take an URI string as an argument instead of a local path.

**Additional resources**

- The **gio(1)** man page.

## 14.7. SAMPLE GIO COMMANDS

The following section provides a few examples of the GIO commands usage.

**Example 14.3. List all files in the local/tmp directory**

```
$ gio list file:///tmp
```

**Example 14.4. List the content of a text file from a remote system**

```
$ gio cat ssh://joe@ftp.myserver.net/home/joe/todo.txt
```

**Example 14.5. Copy the previous text file to a local/tmp directory**

```
$ gio copy ssh://joe@ftp.myserver.net/home/joe/todo.txt /tmp/
```

**Additional resources**

- The **gio** man page.

## 14.8. OVERVIEW OF GVFS METADATA

The GVFS metadata storage is implemented as a set of key-and-value pairs that bind information to a particular file. Thus, there is a tool for a user or application to save small data designed for runtime information such as icon position, last-played location, position in a document, emblems, notes, and so on.

Whenever you move a file or directory, GVFS moves the metadata accordingly so that metadata stays connected to the respective file. The GVFS stores all metadata privately, so metadata is available only on the machine. However, GVFS tracks mounts and removable media as well.

**NOTE**

GVFS mounts removable media in the **/run/media/** directory.

To view and manipulate with metadata, you can use:

- the **gio info** command,
- the **gio set** command, or
- any other native GIO way of working with attributes.

**Additional resources**

- The **gio** man page.

## 14.9. SETTING CUSTOM GIO METADATA ATTRIBUTE

This procedure describes how to set a custom metadata attribute.

Notice the differences between particular **gio info** calls and data persistence after a move or rename. Note the **gio info** command output.

**Procedure**

1. Create an empty file:

```
$ touch /tmp/myfile
```

2. View the metadata of this file:

```
$ gio info -a 'metadata:*' /tmp/myfile
uri: file:///tmp/myfile
attributes:
```

3. Set a string to this file:

```
$ gio set -t string /tmp/myfile 'metadata::mynote' 'Please remember to delete this file!'
```

4. View the metadata:

```
$ gio info -a 'metadata:*' /tmp/myfile
uri: file:///tmp/myfile
attributes:
  metadata::mynote: Please remember to delete this file!
```

5. Move this file to a new location:

```
$ gio move /tmp/myfile /tmp/newfile
```



6. View the metadata:

```
$ gio info -a 'metadata::*' /tmp/newfile
uri: file:///tmp/newfile
attributes:
  metadata::mynote: Please remember to delete this file!
```

The metadata persists when you move the file using the GIO API.

### Additional resources

- The **gio** man page.

## 14.10. PASSWORD MANAGEMENT OF GVFS MOUNTS

Learn more about the GVFS mount authentication.

A typical GVFS mount authenticates on its activation unless the resource allows anonymous authentication or does not require any authentication at all.

In a standard GTK+ dialog, you can choose whether to store the password.

When you select the persistent storage, the password is stored in the user keyring. **GNOME Keyring** is a central place for secrets storage. The password is encrypted and automatically unlocked on desktop session start using the password provided on login. For protecting it by a different password, you can set the password at the first use.

The **Passwords and Keys** application helps to manage the stored password and **GNOME Keyring**. It allows removing individual records or changing passwords.

## 14.11. GVFS BACK ENDS

Back ends in GVFS provide access to a specific type of resource. This section provides a list of available GVFS back ends and their specifications.



### NOTE

Some back ends are packaged separately and not installed by default. For installing additional back ends, use the **yum** package manager.

Table 14.1. Available back ends

Back end	Description
<b>afc</b>	Similar to Media Transfer Protocol (MTP), exposes files on your Apple iDevice connected through USB.
<b>afp</b>	An Apple Filing Protocol (AFP) client to access file services of macOS and the original Mac operation system.
<b>archive</b>	Handles various archiving files (ZIP, TAR) in read-only way.

Back end	Description
<b>admin</b>	Provides administrator access to the local file system.
<b>burn</b>	A virtual back end that burning applications use as a temporary storage for new CD, DVD, or BD medium content.
<b>cdda</b>	Exposes Audio CD through separate Waveform Audio File Format (WAV) files.
<b>computer</b>	A virtual back end consolidating active mounts and physical volumes. Acts similarly to a signpost. Previously used by <b>Files</b> for its <b>Computer</b> view.
<b>dav, davs</b>	A WebDAV client, including secure variant. Authentication is possible only during mount. The back end does not support later re-authentication on per-folder basis.
<b>dns-sd</b>	DNS Service Discovery: An Avahi client, used during network browsing, forms persistent URLs to discovered services.
<b>ftp</b>	A fully featured File Transfer Protocol (FTP) client. Supports passive transfers by default. Also, handles secure mode over <b>ftps</b> (explicit mode) and <b>ftpis</b> (implicit mode) schemes.
<b>gphoto2</b>	A Picture Transfer Protocol (PTP) client to access your camera attached by USB or FireWire.
<b>google</b>	Provides access to Google Drive. The Google Drive account needs to be configured in the <b>Online Accounts</b> settings.
<b>http</b>	Handles all HTTP requests. Useful for easy downloading files from web in client applications.
<b>locatest</b>	A simple testing back end that proxies the <b>file://</b> URI. The back end supports error injection.
<b>mtp</b>	A Media Transfer Protocol (MTP) back end for accessing media player and smart phone memory.
<b>network</b>	Allows you to browse Window Network and show shares discovered over Avahi.
<b>recent</b>	A back end used in the file chooser dialog to list recent files used by GNOME applications.
<b>sftp</b>	A fully-featured SSH File Transfer Protocol (SFTP) client.
<b>smb</b>	Accesses Samba and Windows shares.
<b>trash</b>	A trash back end that allows to restore deleted files.

## CHAPTER 15. TROUBLESHOOTING VOLUME MANAGEMENT IN GNOME

Following are some common errors of volume management in GNOME and ways to resolve them.

### 15.1. TROUBLESHOOTING ACCESS TO GVFS LOCATIONS FROM NON-GIO CLIENTS

If you have problems accessing GVFS locations from your application, it might mean that it is not native GIO client. Native GIO clients are typically all GNOME applications using GNOME libraries (**glib**, **gio**). The **gvfs-fuse** service is provided as a fallback for non-GIO clients.

#### Prerequisite

- The **gvfs-fuse** package is installed.

```
$ yum install gvfs-fuse
```

#### Procedure

1. Ensure that **gvfs-fuse** is running.

```
$ ps ax | grep gvfsd-fuse
```

If **gvfs-fuse** is not running, log out and log back in. Red Hat does not recommend starting **gvfs-fuse** manually.

2. Find the system user ID (UID) for the **/run/user/UID/gvfs/** path.  
The **gvfsd-fuse** daemon requires a path where it can expose its services. When the **/run/user/UID/gvfs/** path is unavailable, **gvfsd-fuse** uses the **~/.gvfs** path.

```
$ id -u
```

3. If **gvfsd-fuse** is still not running, start the **gvfsd-fuse** daemon:

```
$ /usr/libexec/gvfsd-fuse -f /run/user/_UID_/gvfs
```

Now, the FUSE mount is available, and you can manually browse for the path in your application.

4. Find the GVFS mounts under the **/run/user/UID/gvfs/** or **~/.gvfs** locations.

### 15.2. TROUBLESHOOTING AN INVISIBLE CONNECTED USB DISK

Under certain circumstances, when you connect a flash drive, the GNOME Desktop might not display it. If your flash drive is not visible in **Files**, but you can see it in the **Disks** application, you can attempt to set the **Show in user interface** option in **Disks**.

#### Procedure

1. Open the **Disks** application.
2. Select the disk in the side bar.

3. Below **Volumes**, Click **Additional partition options (⚙️) → Edit Mount Options....**
4. Click **Show in user interface**
5. Confirm by clicking **OK**.
6. If the flash drive is still not visible, you can try to physically remove the drive and try connecting it again.

## 15.3. TROUBLESHOOTING UNKNOWN OR UNWANTED PARTITIONS LISTED IN FILES

Sometimes, you might see unknown or unwanted partitions when you plug a disk in. For example, when you plug in a flash disk, it is automatically mounted and its volumes are shown in the **Files** side bar. Some devices have a special partition with backups or help files, which you might not want to see each time you plug in the device.

### Procedure

1. Open the **Disks** application.
2. Select the disk in the side bar.
3. Below **Volumes**, Click **Additional partition options (⚙️) → Edit Mount Options....**
4. Uncheck **Show in user interface**
5. Confirm by clicking **OK**.

## 15.4. TROUBLESHOOTING IF A CONNECTION TO THE REMOTE GVFS FILE SYSTEM IS UNAVAILABLE

There are number of situations in which the client is unexpectedly and unwillingly disconnected from a virtual file system or a remote disk mount and is not reconnected automatically.

You might see the error messages in such situations. Several causes trigger such situations:

- The connection is interrupted. For example, your laptop is disconnected from the Wi-Fi.
- The user is inactive for some time and is disconnected by the server (idle timeout).
- The computer is resumed from sleep mode.

### Procedure

1. Unmount file system.
2. Mount it again.
3. If the connection is getting disabled more often, check the settings in the **Network** panel in the GNOME **Settings**.

## 15.5. TROUBLESHOOTING A BUSY DISK IN GNOME

If you receive a notification about your disk being busy, determine the programs that are accessing the disk. Then, you can end the programs that are running. You can also use the **System Monitor** application to kill the programs forcefully.

### Prerequisites

- The **iostat** utility is installed:

```
# yum install iostat
```

### Procedure

1. Examine the list of open files.
  - Run the **lsof** command to get the list of open files.
  - If **lsof** is not available, run the **ps ax** command.
  - You can use **System Monitor** to display the running processes in a GUI.
2. When you have determined the programs, terminate them using any of the following methods:
  - On the command line, execute the **kill** command.
  - In **System Monitor**, right-click the line with the program process name, and click **End** or **Kill** from the context menu.

### Additional resources

- The **kill** man page.

## CHAPTER 16. MANAGING BOOKMARKS IN GNOME

In GNOME, you can edit the bookmarks that are displayed in applications and dialogs that manage files.

### 16.1. BOOKMARKS IN GNOME

Bookmarks integrate into GTK+ and in the GNOME desktop. Every application that presents a standard GTK+ **Open** and **Save** dialog (**GtkFileChooser**) lists bookmarks in the left panel of the dialog. Also, **Files** and its clones present bookmarks in a sidebar.

Besides bookmarks, **GtkFileChooser** lists all other available GVFS volumes and mounts in the sidebar.

While activating a bookmark for the first time, the GVFS subsystem looks for existing mounts and tries to mount the location if the mount does not exist yet. This way, you can authenticate even within the **Open** or **Save** dialog.

Bookmarks are located in the `~/.config/gtk-3.0/bookmarks` file. In the following example, the bookmarked locations are `~/Music`, `~/Pictures`, `~/Videos`, `~/Downloads`, and `~/bin`, so the content of the `~/.config/gtk-3.0/bookmarks` file looks as follows:

```
file:///home/username/Music
file:///home/username/Pictures
file:///home/username/Videos
file:///home/username/Downloads
file:///home/username/bin
```



#### NOTE

Replace username with the user name you are logged in.

You can use this file to edit bookmarks based on your requirements.

### 16.2. ADDING A BOOKMARK IN FILES

You can save a reference to a location by bookmarking it.

#### Procedure

1. Select the folder or file you want to bookmark.
2. Do one of the following:
  - Press **Ctrl+D** or
  - Drag and drop the file or folder to the sidebar.

### 16.3. ADDING A BOOKMARK FOR ALL USERS

As a system administrator, you can set a bookmark for several users at once so that file shares are easily accessible to all the users.

#### Procedure

1. In the home directory of each existing user, edit the **~user/.config/gtk-3.0/bookmarks** file.
2. In the file, add a Uniform Resource Identifiers (URI) line that identifies the bookmark.  
For example, the following lines add bookmarks to the **/usr/share/doc/** directory and to the GNOME FTP network share:

```
file:///usr/share/doc/  
ftp://ftp.gnome.org/
```

3. Optional: To also add the bookmarks for every newly created user on the system:
  - a. Create the **/etc/skel/.config/gtk-3.0/bookmarks** file.
  - b. Enter the bookmark URI lines in the file.

## CHAPTER 17. CUSTOMIZING DEFAULT FAVORITE APPLICATIONS

You can customize frequently used applications as your favorite applications. You can see these favorite applications on the GNOME Shell *dash* in the Activities overview. You can use **dconf** to set the favorite applications for an individual user or for all users.

### 17.1. SETTING DIFFERENT FAVORITE APPLICATIONS FOR INDIVIDUAL USERS

You can set the default favorite applications for individual users.

#### Procedure

1. Open the **Activities** overview by clicking **Activities** at the top left of the screen.
2. Add applications into your favorite list using any of the following methods:
  - Click the grid button to find the application you want, right-click the application icon, and select **Add to Favorites**.
  - Click-and-drag the icon into the dash.
3. View all the applications that exists in the favorite list:

```
$ dconf read /org/gnome/shell/favorite-apps
```



#### NOTE

If you want to lock down the above settings to prevent users from changing them, see [Chapter 19, Locking down selected tasks](#).

### 17.2. SETTING THE SAME FAVORITE APPLICATIONS FOR ALL USERS

You can modify system database files using **dconf** keyfiles to set the same favorites for all users. The following steps edit the **dconf** profile and then create a keyfile to set default favorite applications for all users in the local configuration database.

#### Procedure

1. Create the key file **/etc/dconf/db/local.d/00-favorite-apps** to provide information for the local database. **/etc/dconf/db/local.d/00-favorite-apps** contents:

```
# Snippet sets gedit, terminal and nautilus as default favorites for all users
[org/gnome/shell]
favorite-apps = ['gedit.desktop', 'gnome-terminal.desktop', 'nautilus.desktop']
```

2. To prevent users from overriding these settings, create the file **/etc/dconf/db/local.d/locks/favorite-apps** with the following content:

```
# Lock default favorite applications
/org/gnome/shell/favorite-apps
```



3. Run the **dconf update** command to incorporate your changes into the system databases.
4. Logout and login again for system-wide changes to take effect.

## CHAPTER 18. AUTHENTICATING THE USER IN THE DESKTOP ENVIRONMENT

You can perform the following operations:

- Configure enterprise login options in GNOME,
- Enable smart card authentication, and
- Enable fingerprint authentication.

### 18.1. USING ENTERPRISE CREDENTIALS TO AUTHENTICATE IN GNOME

You can use your enterprise domain credentials to access your system. This section explains how to log in using enterprise credentials in GNOME, configure enterprise credentials at the GNOME welcome screen, and add an authenticated user with enterprise credentials in GNOME.

#### 18.1.1. Logging in with Enterprise Credentials in GNOME

You can use your domain credentials to login to GNOME if your network has an Active Directory or Identity Management domain available, and you have a domain account.

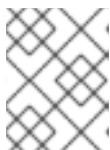
##### Prerequisites

- System is configured to use enterprise domain accounts  
For more information, see [Joining the RHEL 8 system to the IdM domain using the web console](#)

##### Procedure

- While logging in, enter the domain user name followed by an @ sign, and then your domain name. For example, if your domain name is **example.com** and the user name is **User**, enter:

User@example.com



##### NOTE

If the machine is already configured for domain accounts, you should see a helpful hint describing the login format.

#### 18.1.2. Configuring enterprise credentials at the GNOME welcome screen

Perform the following steps to configure workstation for enterprise credentials using the welcome screen that belongs to the **GNOME Initial Setup** program.

The initial setup runs only when you create a new user and log into that account for the first time.

##### Procedure

1. At the login welcome screen, choose **Use Enterprise Login**.
2. Enter your domain name into the **Domain** field.

3. Enter your domain account user name and password.
4. Click **Next**.
5. Depending on the domain configuration, a pop up prompts for the domain administrator's credentials.

### 18.1.3. Adding an authenticated user with enterprise credentials in GNOME

This procedure helps to create a new user through the **GNOME Settings** application. The user is authenticated using enterprise credentials.

#### Prerequisites

- Configured enterprise credentials at the GNOME welcome screen. For more information, see [Configuring enterprise credentials at the GNOME welcome screen](#).

#### Procedure

1. Open the **Settings** window clicking icons in the top right corner of the screen.
2. From the list of items, select **Details > Users**.
3. Click **Unlock** and enter the administrator's password.
4. Click **Add user...**
5. Click **Enterprise Login**.
6. Fill out the **Domain**, **Username**, and **Password** fields for your enterprise account.
7. Click **Add**.
8. Depending on the domain configuration, a pop up prompts for the domain administrator's credentials.

### 18.1.4. Troubleshooting enterprise login in GNOME

You can use the realm utility and its various sub-commands to troubleshoot the enterprise login configuration.

#### Procedure

- To see whether the machine is configured for enterprise logins, run the following command:

```
$ realm list
```



#### NOTE

Network administrators can configure and pre-join workstations to the relevant domains using the kickstart **realm join** command, or running **realm join** in an automated fashion from a script.

#### Additional resources

- The **realm** man page.

## 18.2. ENABLING SMART CARD AUTHENTICATION

You can enable workstations to authenticate using smart cards. In order to do so, you must configure GDM to allow prompting for smart cards and configure operating system to log in using a smart card.

You can use two ways to configure the GDM to allow prompting for smart card authentication with GUI or using the command line.

### 18.2.1. Configuring smart card authentication in GDM using the GUI

You can enable smart card authentication using **dconf** editor GUI. The **dconf** Editor application helps to update the configuration-related values on a dconf database.

#### Prerequisites

- Install the dconf-editor package:

```
# yum install dconf-editor
```

#### Procedure

1. Open the dconf-Editor application and navigate to **/org/gnome/login-screen**.
2. Turn on the **enable-password-authentication** option.
3. Turn on the **enable-smartcard-authentication** option.

#### Additional resources

- The **dconf-editor** man page.
- The **dconf** man page.

### 18.2.2. Configuring smart card authentication in GDM using the command line

You can use the **dconf** command-line utility to enable the GDM login screen to recognize smart card authentication.

#### Procedure

1. Create a keyfile for the GDM database in **/etc/dconf/db/gdm.d/login-screen**, which contains the following content:

```
[org/gnome/login-screen]
enable-password-authentication='false'
enable-smartcard-authentication='true'
```

2. Update the system **dconf** databases:

```
# dconf update
```

### Additional resources

- The **dconf** man page.

### 18.2.3. Enabling the smart card authentication method in the system

For smart card authentication you can use the **system-config-authentication** tool to configure the system to allow you to use smart cards. Thus, you can avail GDM as a valid authentication method for the graphical environment. The tool is provided by the **authconfig-gtk** package.

### Prerequisites

- Install **authconfig-gtk** package
- Configure GDM for smart card authentication

### Additional resources

- For details about configuring system to allow smart card authentication and the **system-config-authentication** tool, see [Configuring smart cards using authselect](#)

## 18.3. FINGERPRINT AUTHENTICATION

You can use the **system-config-authentication** tool to enable fingerprint authentication to allow users to login using their enrolled fingerprints. The tool is provided by the **authconfig-gtk** package.

### Additional resources

- For more information about fingerprint authentication and the **system-config-authentication** tool, see the [Configuring user authentication using authselect](#).

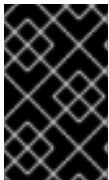
## CHAPTER 19. LOCKING DOWN SELECTED TASKS

This section describes how to lock down for users the following tasks:

- Printing
- File saving on disk
- Repartitioning
- User logout and user switching

### 19.1. LOCKING DOWN PRINTING

You can disable the print dialog from being shown to users. This can be useful if you are giving temporary access to a user or you do not want the user to print to network printers.



#### IMPORTANT

The feature only works in applications which support it. Not all GNOME and third party applications have this feature enabled. The changes do not have effect on applications which do not support this feature.

To prevent applications from printing, lock down the **org.gnome.desktop.lockdown.disable-printing** key:

#### Procedure

1. Create the user profile in **/etc/dconf/profile/user** unless it already exists:

```
user-db:user
system-db:local
```

2. Create a local database for machine-wide settings in the **etc/dconf/db/local.d/00-lockdown** file:

```
[org/gnome/desktop/lockdown]

# Prevent applications from printing
disable-printing=true
```

3. Override the user's setting and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/lockdown** file:

```
# List the keys used to configure lockdown
/org/gnome/desktop/lockdown/disable-printing
```

4. Update the system databases:

```
# dconf update
```

Having followed these steps, applications supporting this lockdown key, such as **Evolution**, **Evince**, or **Gedit**, will disable printing.

## 19.2. LOCKING FILE SAVING ON DISK

You can disable the **Save** and **Save As** dialogs. This can be useful if you are giving temporary access to a user or you do not want the user to save files to the computer.



### IMPORTANT

The feature only works in applications which support it. Not all GNOME and third party applications have this feature enabled. The changes will have no effect on applications which do not support this feature.

To prevent applications from file saving, lock down the **org.gnome.desktop.lockdown.disable-save-to-disk** key:

### Procedure

1. Create the user profile in **/etc/dconf/profile/user** unless it already exists:

```
user-db:user
system-db:local
```

2. Create a local database for machine-wide settings in the **/etc/dconf/db/local.d/00-lockdown** file:

```
[org/gnome/desktop/lockdown]

# Prevent the user from saving files on disk
disable-save-to-disk=true
```

3. Override the user's setting and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/lockdown** file:

```
# Lock this key to disable saving files on disk
/org/gnome/desktop/lockdown/disable-save-to-disk
```

4. Update the system databases:

```
# dconf update
```

Having followed these steps, applications supporting this lockdown key, for example **Videos**, **Image Viewer**, **Evolution**, **Document Viewer**, or **GNOME Shell**, will disable their **Save As** dialogs.

## 19.3. LOCKING REPARTITIONING

**polkit** enables you to set permissions for individual operations. For **udisks2**, the utility for disk management services, the configuration is located at **/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy**. This file contains a set of actions and default values, which can be overridden by system administrator.



## IMPORTANT

**polkit** configuration stored in **/etc** overrides the configuration shipped by packages in **/usr/share/**.

### Procedure

1. Create a file with the same content as in **/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy**:

```
cp /usr/share/polkit-1/actions/org.freedesktop.udisks2.policy /etc/share/polkit-1/actions/org.freedesktop.udisks2.policy
```

Do not change the **/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy** file, your changes will be overwritten by the next package update.

2. Delete the action you do not need, and add the following lines to the **/etc/polkit-1/actions/org.freedesktop.udisks2.policy** file:

```
<action id="org.freedesktop.udisks2.modify-device">
  <message>Authentication is required to modify the disks settings</message>
  <defaults>
    <allow_any>no</allow_any>
    <allow_inactive>no</allow_inactive>
    <allow_active>yes</allow_active>
  </defaults>
</action>
```

Replace **no** by **auth\_admin** if you want to ensure only the **root** user is able to perform the action.

3. Save the changes.

When the user tries to change the disks settings, the following message is returned:

Authentication is required to modify the disks settings.

## 19.4. LOCKING DOWN USER LOGOUT AND USER SWITCHING

To prevent the user from logging out, use the following procedure.

### Procedure

1. Create the **/etc/dconf/profile/user** profile, which contains the following lines:

```
user-db:user
system-db:local
```

where **local** is the name of a dconf database

2. Create the **/etc/dconf/db/local.d/** directory if it does not already exist.
3. Create the **/etc/dconf/db/local.d/00-logout** key file to provide information for the local database:



```
[org/gnome/desktop/lockdown]
# Prevent the user from user switching
disable-log-out=true
```

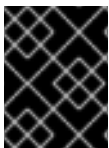
4. Override the user's setting, and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/lockdown** file:

```
# Lock this key to disable user logout
/org/gnome/desktop/lockdown/disable-log-out
```

5. Update the system databases:

```
# dconf update
```

6. Users must log out and back in again before the system-wide settings take effect.



### IMPORTANT

Users can evade the logout lockdown by switching to a different user. To prevent such scenario, lock down user switching as well.

To lock down user switching, use the following procedure:

### Procedure

1. Create the **/etc/dconf/profile/user** profile, which contains the following lines:

```
user-db:user
system-db:local
```

where **local** is the name of a dconf database

2. Create the **/etc/dconf/db/local.d/** directory if it does not already exist.
3. Create the **/etc/dconf/db/local.d/00-user-switching** key file to provide information for the local database:

```
# Prevent the user from user switching
[org/gnome/desktop/lockdown]
disable-user-switching=true

[org/gnome/desktop/screensaver]
user-switch-enabled=false
```

4. Override the user's setting, and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/lockdown** file:

```
# Lock this key to disable user switching
/org/gnome/desktop/lockdown/disable-user-switching
/org/gnome/desktop/screensaver/user-switch-enabled
```

5. Update the system databases:

```
| # dconf update
```

6. Users must log out and back in again before the system-wide settings take effect.

## CHAPTER 20. RESTRICTING THE SESSION TO A SINGLE APPLICATION

You can start the GNOME session in single-application mode, also known as kiosk mode. In this session, GNOME displays only a full-screen window of the application that you have selected.

### 20.1. SINGLE-APPLICATION MODE

Single-application mode is a modified GNOME session that reconfigures the Mutter window manager into an interactive kiosk. This session locks down certain behavior to make the standard desktop more restrictive. The user can interact only with a single application selected by the administrator.

You can set up single-application mode for several use cases, such as:

- In the communication, entertainment, or education fields
- As a self-serve machine
- As an event manager
- As a registration point

The **gnome-session-kiosk-session** package provides the single-application mode configuration and sessions in RHEL 8.

### 20.2. ENABLING SINGLE-APPLICATION MODE

This procedure installs and enables single-application mode, which restricts the GNOME session to a single application.

#### Procedure

1. Install the **gnome-session-kiosk-session** package:

```
# yum install gnome-session-kiosk-session
```

2. As the user that will open the single-application session, create the **/home/user/.local/bin/redhat-kiosk** file:

```
[user]$ mkdir -p ~/.local/bin
```

```
[user]$ touch ~/.local/bin/redhat-kiosk
```

3. Edit the **/home/user/.local/bin/redhat-kiosk** file and enter the executable name of the application that you want to launch in single-application mode.  
For example, to launch the **Firefox** browser in single-application mode, enter the following content:

```
#!/bin/sh

while true; do
    firefox --kiosk https://example.org
done
```

The **while true** loop ensures that the application restarts if it terminates for any reason.

4. Make the file executable:

```
[user]$ chmod +x ~/.local/bin/redhat-kiosk
```

5. If you created the file or its containing directories as a different user than the single-application user, such as **root**, ensure that the file has the correct permissions:

```
# chown -R user:group ~user/.local
```

6. At the GNOME login screen, select the **Kiosk** session from the gear button menu and log in as the single-application user.

## CHAPTER 21. MANAGING USER SESSIONS

### 21.1. WHAT GDM IS

The GNOME Display Manager (GDM) is a graphical login program running in the background that runs and manages the **X.Org** display servers for both local and remote logins.

GDM is a replacement for XDM, the X Display Manager. However, GDM is not derived from XDM and does not contain any original XDM code. In addition, there is no support for a graphical configuration tool in GDM, so editing the **/etc/gdm/custom.conf** configuration file is necessary to change the GDM settings.

### 21.2. RESTARTING GDM

When you make changes to the system configuration such as setting up the login screen banner message, login screen logo, or login screen background, restart GDM for your changes to take effect.



#### WARNING

Restarting the **gdm** service terminates all currently running GNOME sessions of all desktop users who are logged in. This might result in users losing unsaved data.

#### Procedure

- To restart the GDM service, run the following command:

```
# systemctl restart gdm.service
```

#### Procedure

- To display results of the GDM configuration, run the following command:

```
$ DCONF_PROFILE=gdm gsettings list-recursively org.gnome.login-screen
```

### 21.3. ADDING AN AUTOSTART APPLICATION FOR ALL USERS

You can set application to start automatically when any user logs into the GNOME environment.

#### Procedure

1. Create a **.desktop** file in the **/etc/xdg/autostart/** directory, such as **/etc/xdg/autostart/nautilus.desktop**. Enter the following content in the file:

```
[Desktop Entry]
Type=Application
Name=Files
```

```
Exec=/usr/bin/nautilus -n
OnlyShowIn=GNOME;
X-GNOME-Autostart-enabled=true
```

- Replace **Files** with the name of the application.
  - Replace **/usr/bin/nautilus -n** with the command that starts the application. Use the full file path.
2. Optionally, configure the application to start only when a selected GSettings key is enabled. GNOME then runs the application automatically if the key's value is true. If the key's value changes in the running session, GNOME starts or stops the application to match the new value. Add the following line in the **.desktop** file:

```
AutostartCondition=GSettings org.gnome.desktop.background show-desktop-icons
```

- Replace **org.gnome.desktop.background show-desktop-icons** with the GSettings schema and key that the automatic start depends on.

### Additional resources

- You can also configure an autostart application for a specific user. Use the **Tweaks** application, which is available from the **gnome-tweaks** package.

## 21.4. CONFIGURING AUTOMATIC LOGIN

As an administrator, you can enable automatic login from the **Users** panel in **GNOME Settings**, or you can set up automatic login manually in the GDM custom configuration file, as follows.

Run the following procedure to set up automatic login for a user **john**.

### Procedure

- Edit the **/etc/gdm/custom.conf** file, and make sure that the **[daemon]** section in the file specifies the following:

```
[daemon]
AutomaticLoginEnable=True
AutomaticLogin=john
```

Replace **john** with the user that you want to be automatically logged in.

## 21.5. CONFIGURING AUTOMATIC LOGOUT

User sessions that have been idle for a specific period of time can be ended automatically. You can set different behavior based on whether the machine is running from a battery or from mains power by setting the corresponding GSettings key, then locking it.

**WARNING**

Users can potentially lose unsaved data if an idle session is automatically ended.

To set automatic logout for a mains powered machine:

**Procedure**

1. Create a local database for machine-wide settings in the **/etc/dconf/db/local.d/00-autologout** file:

```
[org/gnome/settings-daemon/plugins/power]
# Set the timeout to 900 seconds when on mains power
sleep-inactive-ac-timeout=900
# Set action after timeout to be logout when on mains power
sleep-inactive-ac-type='logout'
```

2. Override the user's setting, and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/autologout** file:

```
# Lock automatic logout settings
/org/gnome/settings-daemon/plugins/power/sleep-inactive-ac-timeout
/org/gnome/settings-daemon/plugins/power/sleep-inactive-ac-type
```

3. Update the system databases:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

The following GSettings keys are of interest:

- `org.gnome.settings-daemon.plugins.power.sleep-inactive-ac-timeout`  
The number of seconds that the computer needs to be inactive before it goes to sleep if it is running from AC power.
- `org.gnome.settings-daemon.plugins.power.sleep-inactive-ac-type`  
What should happen when the timeout has passed if the computer is running from AC power.
- `org.gnome.settings-daemon.plugins.power.sleep-inactive-battery-timeout`  
The number of seconds that the computer needs to be inactive before it goes to sleep if it is running from power.
- `org.gnome.settings-daemon.plugins.power.sleep-inactive-battery-type`  
What should happen when the timeout has passed if the computer is running from battery power.

If you want to list available values for a key, use the following procedure:

**Procedure**

- Run the **gsettings range** command on the required key. For example:

```
$ gsettings range org.gnome.settings-daemon.plugins.power sleep-inactive-ac-type
enum
'blank'
'suspend'
'shutdown'
'hibernate'
'interactive'
'nothing'
'logout'
```

## 21.6. SETTING A DEFAULT DESKTOP SESSION FOR ALL USERS

You can configure a default desktop session that is preselected for all users that have not logged in yet.

If a user logs in using a different session than the default, their selection persists to their next login.

### Procedure

1. Copy the configuration file template:

```
# cp /usr/share/accountsservice/user-templates/standard \
/etc/accountsservice/user-templates/standard
```

2. Edit the new **/etc/accountsservice/user-templates/standard** file. On the **Session=gnome** line, replace **gnome** with the session that you want to set as the default.
3. *Optional:* To configure an exception to the default session for a certain user, follow these steps:
  - a. Copy the template file to **/var/lib/AccountsService/users/user-name**:

```
# cp /usr/share/accountsservice/user-templates/standard \
/var/lib/AccountsService/users/user-name
```

- b. In the new file, replace variables such as **\${USER}** and **\${ID}** with the user values.
- c. Edit the **Session** value.

## 21.7. SETTING SCREEN BRIGHTNESS AND IDLE TIME

By creating a local database, you can, for example:

- Configure the drop in the brightness level
- Set brightness level
- Set idle time

### Configuring the drop in the brightness level

To set the drop in the brightness level when the device has been idle for some time:

### Procedure



1. Create a local database for machine-wide settings in the **/etc/dconf/db/local.d/00-power** file including these lines:

```
[org/gnome/settings-daemon/plugins/power]
idle-dim=true
```

2. Update the system databases:

```
# dconf update
```

3. Users must log out and back in again before the system-wide settings take effect.

## Setting brightness level

To set brightness level:

### Procedure

1. Create a local database for machine-wide settings in the **/etc/dconf/db/local.d/00-power** file, as in the following example:

```
[org/gnome/settings-daemon/plugins/power]
idle-brightness=30
```

Replace **30** with the integer value you want to use.

2. Update the system databases:

```
# dconf update
```

3. Users must log out and back in again before the system-wide settings take effect.

## Setting idle time

To set idle time after which the screen is blanked and the default screensaver is displayed:

### Procedure

1. Create a local database for machine-wide settings in **/etc/dconf/db/local.d/00-session**, as in the following example:

```
[org/gnome/desktop/session]
idle-delay=uint32 900
```

Replace **900** with the integer value you want to use.

You must include the **uint32** along with the integer value as shown.

2. Update the system databases:

```
# dconf update
```

3. Users must log out and back in again before the system-wide settings take effect.

## 21.8. LOCKING THE SCREEN WHEN THE USER IS IDLE

To enable the screensaver and make the screen lock automatically when the user is idle, follow this procedure:

### Procedure

1. Create a local database for system-wide settings in the **etc/dconf/db/local.d/00-screensaver** file:

```
[org/gnome/desktop/session]
# Set the lock time out to 180 seconds before the session is considered idle
idle-delay=uint32 180
[org/gnome/desktop/screensaver]
# Set this to true to lock the screen when the screensaver activates
lock-enabled=true
# Set the lock timeout to 180 seconds after the screensaver has been activated
lock-delay=uint32 180
```

You must include the **uint32** along with the integer key values as shown.

2. Override the user's setting, and prevent the user from changing it in the **/etc/dconf/db/local.d/locks/screensaver** file:

```
# Lock desktop screensaver settings
/org/gnome/desktop/session/idle-delay
/org/gnome/desktop/screensaver/lock-enabled
/org/gnome/desktop/screensaver/lock-delay
```

3. Update the system databases:

```
# dconf update
```

4. Users must log out and back in again before the system-wide settings take effect.

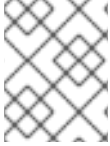
## 21.9. SCREENCAST RECORDING

GNOME Shell features a built-in screencast recorder. The recorder allows users to record desktop or application activity during their session and distribute the recordings as high-resolution video files in the **webm** format.

To make a screencast:

### Procedure

1. To start the recording, press the **Ctrl+Alt+Shift+R** shortcut.  
When the recorder is capturing the screen activity, it displays a red circle in the top-right corner of the screen.
2. To stop the recording, press the **Ctrl+Alt+Shift+R** shortcut.  
The red circle in the top-right corner of the screen disappears.
3. Navigate to the **~/Videos** directory where you can find the recorded video with a file name that starts with *Screencast* and includes the date and time of the recording.

**NOTE**

The built-in recorder always captures the entire screen, including all monitors in multi-monitor setups.

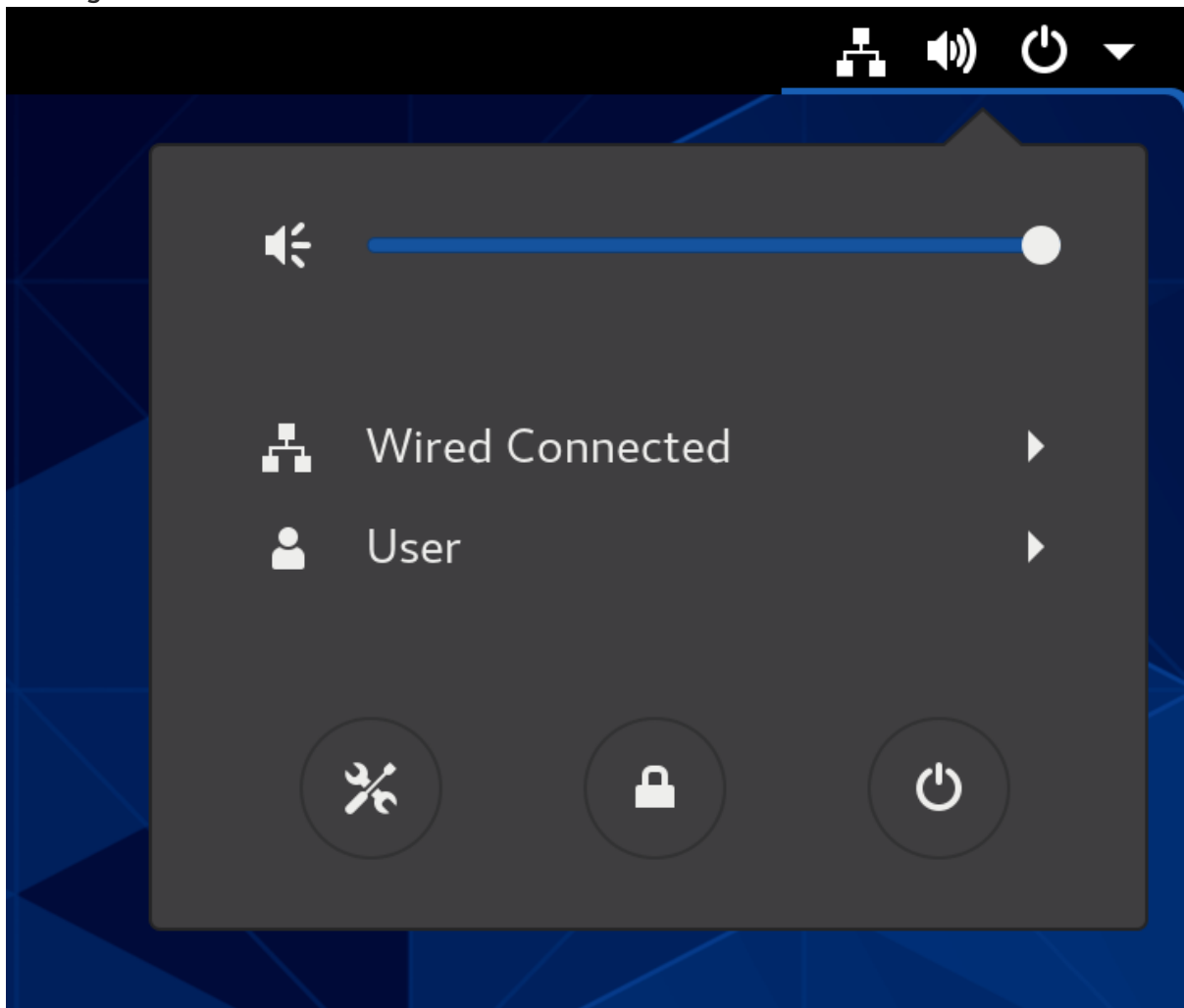
## CHAPTER 22. CONFIGURING THE SCREEN ROTATION

### 22.1. CONFIGURING THE SCREEN ROTATION FOR A SINGLE USER

This procedure sets screen rotation for the current user.

#### Procedure

1. Go to the **system menu**, which is accessible from the top-right screen corner, and click the **Settings** icon.



2. In the **Settings** → **Devices** section, choose **Displays**.
3. Configure the rotation using the **Orientation** field.
4. Confirm your choice by clicking **Apply**.
5. If you are satisfied with the new setup preview, click on **Keep changes**.

The setting persists to your next login.

#### Additional resources

- For information about rotating the screen for all users on a system, see [Configuring the screen rotation for all users](#).

## 22.2. CONFIGURING THE SCREEN ROTATION FOR ALL USERS

This procedure sets a default screen rotation for **all** users on a system and is suitable for mass deployment of homogenized display configuration.

### Procedure

1. Prepare the preferable setup for a single user as in [Configuring the screen rotation for a single user](#).
2. Copy the **transform** section of the `~/.config/monitors.xml` configuration file, which configures the screen rotation. An example portrait orientation:

```
<?xml version="1.0" encoding="UTF-8"?>

<transform>
  <rotation>left</rotation>
  <flipped>no</flipped>
</transform>
```

3. Paste the content in the `/etc/xdg/monitors.xml` file that stores system-wide configuration.
4. Save the changes.

The new setup takes effect for all the users the next time they log in in the system.

### Additional resources

- [Configuring the screen rotation for a single user](#)

## CHAPTER 23. ENABLING ACCESSIBILITY FOR VISUALLY IMPAIRED USERS

As a system administrator, you can configure the desktop environment to support users with a visual impairment.

To enable accessibility, perform the following procedures.

### 23.1. COMPONENTS THAT PROVIDE ACCESSIBILITY FEATURES

On the RHEL 8 desktop, the **Orca** screen reader ensures accessibility for users with a visual impairment. **Orca** is included in the default RHEL installation.

**Orca** reads information from the screen and communicates it to you using the following components:

#### Speech Dispatcher

**Orca** uses **Speech Dispatcher** to communicate with the speech synthesizer. **Speech Dispatcher** supports various speech synthesis backends, ensures that messages from other applications do not to interrupt the messages from **Orca**, and provides other functionality.

#### Speech synthesizer

Provides a speech output. The default speech synthesizer is **eSpeak-NG**.

#### Braille display

Provides a tactile output. The **BRLTTY** service enables this functionality.

#### Additional resources

- [Orca help page](#)

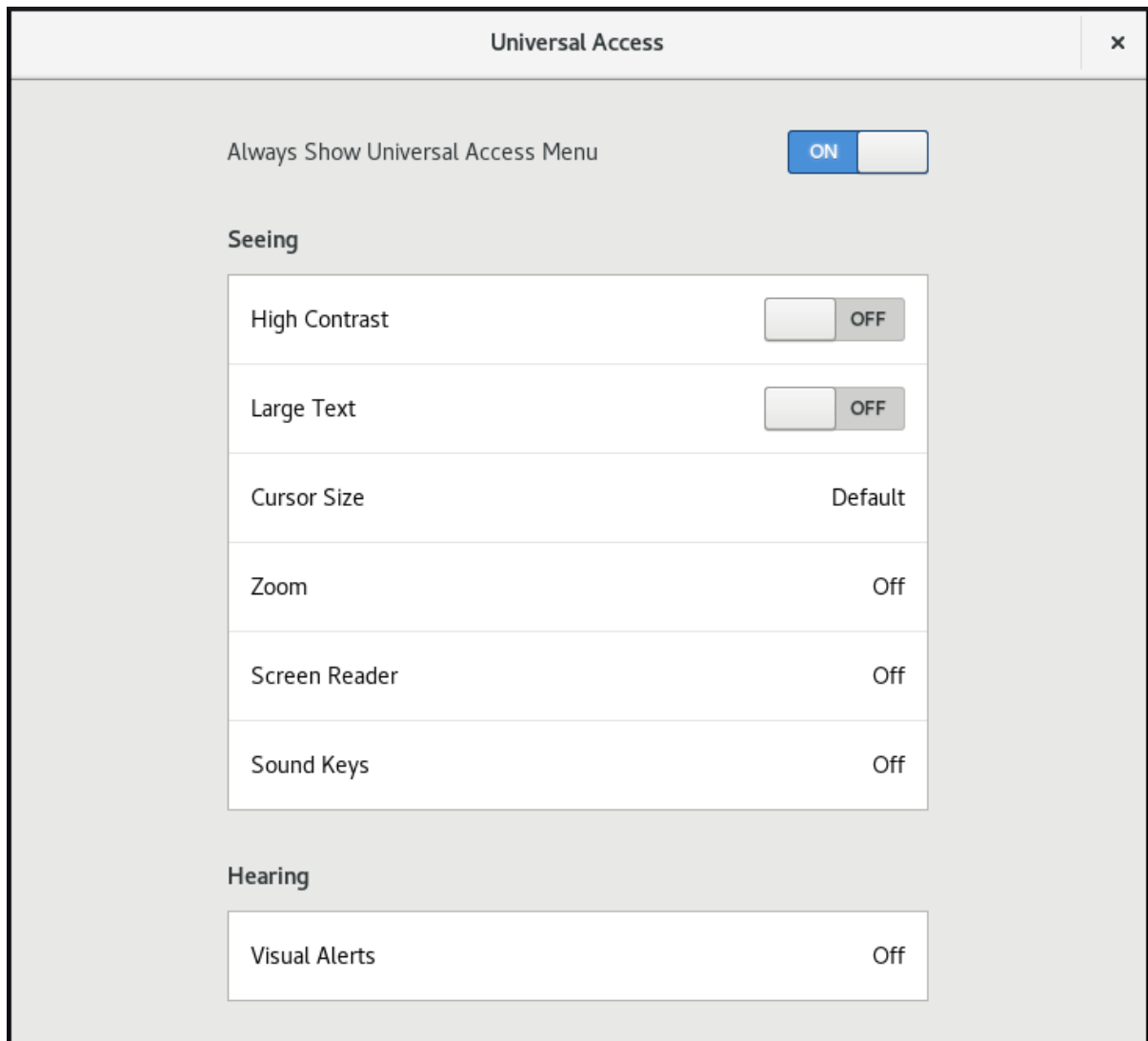
### 23.2. ENABLING THE UNIVERSAL ACCESS MENU

You can enable the **Universal Access Menu** icon in the top panel, which provides a menu with several accessibility options.

#### Procedure

1. Open the **Settings** application.
2. Select **Universal Access**.
3. Enable the **Always Show Universal Access Menu** item.

#### Enabling the Universal Access menu in Settings



### Verification

- Check that the **Universal Access Menu** icon is displayed on the top bar even when all options from this menu are switched off.



## 23.3. ENABLING THE SCREEN READER

You can enable the **Orca** screen reader in your desktop environment. The screen reader then reads the text displayed on the screen to improve accessibility.

### Procedure

- Enable the screen reader using either of the following ways:
  - Press the **Super+Alt+S** keyboard shortcut.
  - If the top panel shows the **Universal Access** menu, select **Screen Reader** in the menu.

### Verification

1. Open an application with text content.

2. Check that the screen reader reads the text in the application.

## 23.4. ENABLING A BRAILLE DISPLAY DEVICE

The Braille display is a device that uses the **brlTTY** service to provide tactile output for visually impaired users.

In order for the Braille display to work correctly, perform the following procedures.

### 23.4.1. Supported types of Braille display device

The following types of Braille display devices are supported on RHEL 8.

**Table 23.1. Braille display device types and the corresponding syntax**

Braille device type	Syntax of the type	Note
Serial device	<b>serial:path</b>	Relative paths are at <b>/dev</b> .
USB device	<b>[serial-number]</b>	The brackets ( <b>[]</b> ) here indicate optionality.
Bluetooth device	<b>bluetooth:address</b>	

### 23.4.2. Enabling the brlTTY service

To enable the Braille display, enable the **brlTTY** service to start automatically on boot. By default, **brlTTY** is disabled.

#### Prerequisites

- Ensure that the **brlTTY** package is installed:

```
# yum install brlTTY
```

- Optionally, you can install speech synthesis support for **brlTTY**:

```
# yum install brlTTY-espeak-ng
```

#### Procedure

- Enable the **brlTTY** service to start on boot:

```
# systemctl enable --now brlTTY
```

#### Verification steps

1. Reboot the system.
2. Check that the **brlTTY** service is running:



```
# systemctl status brlty
● brlty.service - Braille display driver for Linux/Unix
   Loaded: loaded (/usr/lib/systemd/system/brlty.service; enabled; vendor pres>
   Active: active (running) since Tue 2019-09-10 14:13:02 CEST; 39s ago
   Process: 905 ExecStart=/usr/bin/brlty (code=exited, status=0/SUCCESS)
   Main PID: 914 (brlty)
     Tasks: 3 (limit: 11360)
    Memory: 4.6M
   CGroup: /system.slice/brlty.service
           └─914 /usr/bin/brlty
```

### 23.4.3. Authorizing users of a Braille display device

To use a Braille display device, you must set the users who are authorized to use the Braille display device.

#### Procedure

1. In the **/etc/brlty.conf** file, ensure that **keyfile** is set to **/etc/brlapi.key**:

```
api-parameters Auth=keyfile:/etc/brlapi.key
```

This is the default value. Your organization might have overridden it.

2. Authorize the selected users by adding them to the **brlapi** group:

```
# usermod --append -G brlapi user-name
```

#### Additional resources

- [Editing user groups using the command line](#)

### 23.4.4. Setting the driver for a Braille display device

The **brlty** service automatically chooses a driver for your Braille display device. If the automatic detection fails or takes too long, you can set the driver manually.

#### Prerequisites

- The automatic driver detection has failed or takes too long.

#### Procedure

1. Open the **/etc/brlty.conf** configuration file.
2. Find the **braille-driver** directive, which specifies the driver for your Braille display device.
3. Specify the identification code of the required driver in the **braille-driver** directive. Choose the identification code of required driver from the list provided in **/etc/brlty.conf**. For example, to use the XWindow driver:

```
# XWindow
braille-driver xw
```

To set multiple drivers, list them separated by commas. Automatic detection then chooses from the listed drivers.

### 23.4.5. Connecting a Braille display device

The **brltty** service automatically connects to your Braille display device. If the automatic detection fails, you can set the connection method manually.

#### Prerequisites

- The Braille display device is physically connected to your system.
- The automatic connection has failed.

#### Procedure

1. If the device is connected by a serial-to-USB adapter, find the actual device name in the kernel messages on the device plug:

```
# journalctl --dmesg | fgrep ttyUSB
```

2. Open the **/etc/brltty.conf** configuration file.
3. Find the **braille-device** directive.
4. In the **braille-device** directive, specify the connection.  
You can also set multiple devices, separated by commas, and each of them will be probed in turn.

For example:

#### Example 23.1. Settings for the first serial device

```
braille-device serial:ttyS0
```

#### Example 23.2. Settings for the first USB device matching Braille driver

```
braille-device usb:
```

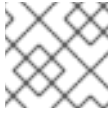
#### Example 23.3. Settings for a specific USB device by serial number

```
braille-device usb:nnnnn
```

#### Example 23.4. Settings for a serial-to-USB adapter

Use the device name that you found earlier in the kernel messages:

```
braille-device serial:ttyUSB0
```

**NOTE**

Setting **braille-device** to **usb:** does not work for a serial-to-USB adapter.

**Example 23.5. Settings for a specific Bluetooth device by address**

```
braille-device bluetooth:xx:xx:xx:xx:xx:xx
```

**23.4.6. Setting the text table**

The **brlty** service automatically selects a text table based on your system language. If your system language does not match the language of a document that you want to read, you can set the text table manually.

**Procedure**

1. Edit the **/etc/brlty.conf** file.
2. Identify the code of your selected text table.  
You can find all available text tables in the **/etc/brlty/Text/** directory. The code is the file name of the text table without its file suffix.
3. Specify the code of the selected text table in the **text-table** directive.  
For example, to use the text table for American English:

```
text-table en_US # English (United States)
```

**23.4.7. Setting the contraction table**

You can select which table is used to encode the abbreviations with a Braille display device. Relative paths to particular contraction tables are stored within the **/etc/brlty/Contraction/** directory.

**WARNING**

If no table is specified, the **brlty** service does not use a contraction table.

**Procedure**

- Choose a contraction table from the list in the **/etc/brlty.conf** file.  
For example, to use the contraction table for American English, grade 2:

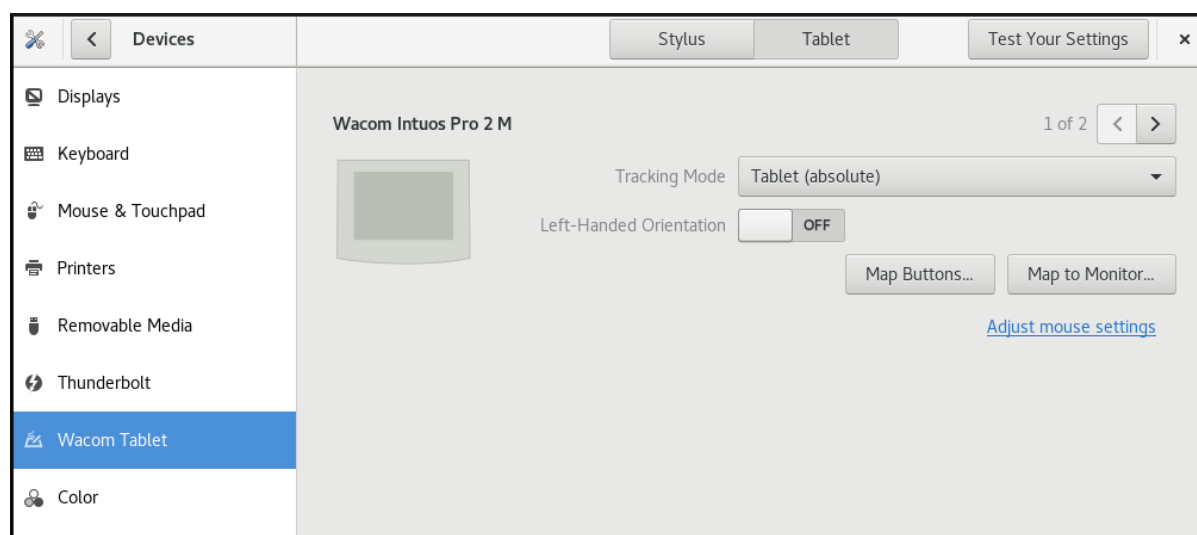
```
contraction-table en-us-g2 # English (US, grade 2)
```

## CHAPTER 24. TABLETS

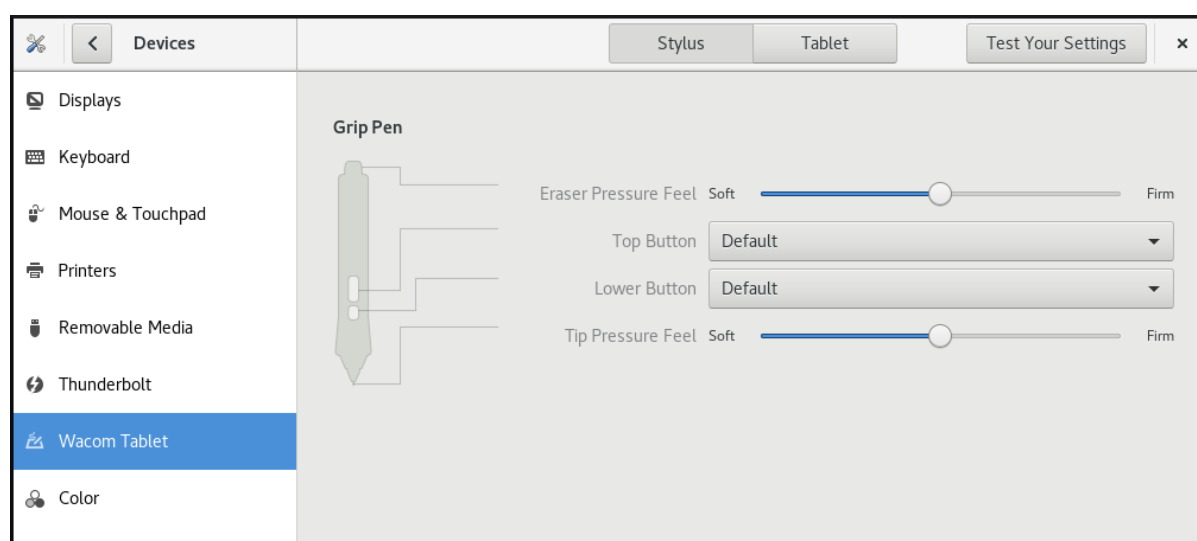
To manage Wacom tablets connected to your system, use the following tools:

- The **gnome-settings-daemon** service
- The **Wacom Tablet** settings panel in the GNOME environment

### The Wacom Tablet settings panel for a tablet



### The Wacom Tablet settings panel for a grip pen



Both these tools, as well as the **libinput** stack, use the **libwacom** tablet client library, which stores the data about Wacom tablets.

If you want to add support for a new tablet into the **libwacom** library, you must ensure that a definition file for this new tablet exists.

## 24.1. PREPARING A TABLET DEFINITION FILE

The **libwacom** tablet client library needs a definition file for the tablet that you want to add.

To ensure that the tablet definition file exists, follow this procedure.

#### Prerequisites

## Prerequisites

- List all local devices recognized by **libwacom**:

```
$ libwacom-list-local-devices
```

Make sure that your device is recognized in the output.

If your device is not listed, the device is missing from the **libwacom** database. However, the device might still be visible as an event device in the kernel under **/proc/bus/input/devices**, and if you use the **X.Org** display server, in the **X11** session on the **xinput** list.

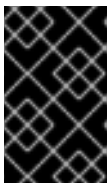
## Procedure

1. Install the package that provides tablet definition files:

```
# yum install libwacom-data
```

The package installs tablet definitions in the **/usr/share/libwacom/** directory.

2. Check whether the definition file is available in the **/usr/share/libwacom/** directory.  
To use the screen mapping correctly, support for your tablet must be included in the **libwacom** database and in the **udev** rules file.



### IMPORTANT

A common indicator that a device is not supported by **libwacom** is that it works normally in a GNOME session, but the device is not correctly mapped to the screen.

3. If the definition file for your device is not available in **/usr/share/libwacom/**, you have these options:
  - The required definition file may already be available in the [linuxwacom/libwacom](#) upstream repository. You can try to find the definition file there. If you find your tablet model in the list, copy the file to the local machine.
  - You can create a new tablet definition file. Use the **data/wacom.example** file below, and edit particular lines based on the characteristics of your device.

#### Example 24.1. Example model file description for a tablet

```
[Device]

# The product is the product name announced by the kernel
Product=Intuos 4 WL 6x9

# Vendor name of this tablet
Vendor=Wacom

# DeviceMatch includes the bus (usb, serial), the vendor ID and the actual
# product ID
DeviceMatch=usb:056a:00bc

# Class of the tablet. Valid classes include Intuos3, Intuos4, Graphire, Bamboo, Cintiq
Class=Intuos4
```

```

# Exact model of the tablet, not including the size.
Model=Intuos 4 Wireless

# Width in inches, as advertised by the manufacturer
Width=9

# Height in inches, as advertised by the manufacturer
Height=6

# Optional features that this tablet supports
# Some features are dependent on the actual tool used, e.g. not all styli
# have an eraser and some styli have additional custom axes (e.g. the
# airbrush pen). These features describe those available on the tablet.
#
# Features not set in a file default to false/0

[Features]
# This tablet supports styli (and erasers, if present on the actual stylus)
Stylus=true

# This tablet supports touch.
Touch=false

# This tablet has a touch ring (Intuos4 and Cintiq 24HD)
Ring=true
# This tablet has a second touch ring (Cintiq 24HD)
Ring2=false

# This tablet has a vertical/horizontal scroll strip
VStrip=false
HStrip=false

# Number of buttons on the tablet
Buttons=9

# This tablet is built-in (most serial tablets, Cintiqs)
BuiltIn=false

```

## 24.2. ADDING SUPPORT FOR A NEW TABLET

To add support for a new tablet into the **libwacom** tablet information client library, follow this procedure.

### Prerequisites

- The definition file for the tablet that you want to add exists.  
For more information about ensuring that the definition file exists, see [Section 24.1, “Preparing a tablet definition file”](#).

### Procedure

1. Add and install the definition file with the **.tablet** suffix:

■

```
# cp the-new-file.tablet /usr/share/libwacom/
```

Once installed, the tablet is part of the **libwacom** database. The tablet is then available through **libwacom-list-local-devices**.

2. Create a new **/etc/udev/rules/99-libwacom-override.rules** file with the following content so that your settings are not overwritten:

```
ACTION!="add|change", GOTO="libwacom_end"
KERNEL!="event[0-9]*", GOTO="libwacom_end"

[new tablet match entries go here]

LABEL="libwacom_end"
```

3. Reboot your system.

## 24.3. WHERE IS THE WACOM TABLET CONFIGURATION STORED

Configuration for your Wacom tablet is stored in GSettings in the **/org/gnome/settings-daemon/peripherals/wacom/machine-id-device-id** key, where **machine-id** is a D-Bus machine ID, and **device-id** is a tablet device ID.

The configuration schema for the tablet is **org.gnome.settings-daemon.peripherals.wacom**.

Stylus configuration is stored in the **/org/gnome/settings-daemon/peripherals/wacom/device-id/tool-id** key, where **tool-id** is the identifier for the stylus used for professional ranges. For the consumer ranges with no support for **tool-id**, a generic identifier is used instead.

The configuration schema for the stylus is **org.gnome.settings-daemon.peripherals.wacom.stylus**. The configuration schema for the eraser is **org.gnome.settings-daemon.peripherals.wacom.eraser**.

## 24.4. LISTING AVAILABLE WACOM TABLET CONFIGURATION PATHS

To get the full list of tablet configuration paths used on a particular machine, use the **gsd-list-wacom** tool, which is provided by the **gnome-settings-daemon** package.

### Prerequisites

- Make sure that the **gnome-settings-daemon** package is installed on your system.

```
# yum install gnome-settings-daemon
```

### Procedure

- To get the full list of tablet configuration paths used on your machine, run the following command:

```
$ /usr/libexec/gsd-list-wacom
```



## IMPORTANT

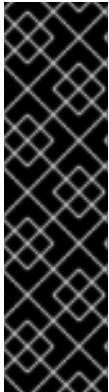
Using **machine-id**, **device-id**, and **tool-id** in configuration paths allows for shared home directories with independent tablet configuration per machine. However, when sharing home directories between Machines, the Wacom settings apply only to one machine.

This is because the **machine-id** for your Wacom tablet is included in the configuration path of the **/org/gnome/settings-daemon/peripherals/wacom/machine-id-device-id GSettings** key, which stores your tablet settings.



## CHAPTER 25. INSTALLING APPLICATIONS USING FLATPAK

You can install certain applications using the Flatpak package manager. The following sections describe how to search for, install, launch, and update Flatpak applications on the command line and in the graphical interface.



### IMPORTANT

Red Hat provides Flatpak applications only as a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see <https://access.redhat.com/support/offerings/techpreview>.

The Flatpak package manager itself is fully supported.

### 25.1. THE FLATPAK TECHNOLOGY

Flatpak provides a sandbox environment for application building, deployment, distribution, and installation.

Applications that you launch using Flatpak have minimum access to the host system, which protects the system installation against third-party applications. Flatpak provides application stability regardless of the versions of libraries installed on the host system.

Flatpak applications are distributed from repositories called remotes. Red Hat provides a remote with RHEL applications. Additionally, third-party remotes are available as well. Red Hat does not support applications from third-party remotes.

### 25.2. SETTING UP FLATPAK

This procedure installs the Flatpak package manager.

#### Procedure

- Install the **flatpak** package:

```
# yum install flatpak
```

### 25.3. ENABLING THE RED HAT FLATPAK REMOTE

This procedure configures the Red Hat Container Catalog as a Flatpak remote on your system.

#### Prerequisites

- You have an account on the Red Hat Customer Portal.



## NOTE

For large-scale deployments where the users do not have Customer Portal accounts, Red Hat recommends using registry service accounts. For details, see [Registry Service Accounts](#).

## Procedure

1. Enable the **rhel** Flatpak remote:

```
$ flatpak remote-add \
  --if-not-exists \
  rhel \
  https://flatpaks.redhat.io/rhel.flatpakrepo
```

2. Log into the Red Hat Container Catalog:

```
$ podman login registry.redhat.io

Username: your-user-name
Password: your-password
```

Provide the credentials to your Red Hat Customer Portal account or your registry service account tokens.

By default, Podman saves the credentials only until you log out.

3. Optional: Save your credentials permanently. Use one of the following options:

- Save the credentials for the current user:

```
$ cp $XDG_RUNTIME_DIR/containers/auth.json \
  $HOME/.config/flatpak/oci-auth.json
```

- Save the credentials system-wide:

```
# cp $XDG_RUNTIME_DIR/containers/auth.json \
  /etc/flatpak/oci-auth.json
```

For best practices, Red Hat recommends that you log into the Red Hat Container Catalog using registry account tokens when installing credentials system-wide.

## Verification

- List the enabled Flatpak remotes:

```
$ flatpak remotes

Name  Options
rhel  system,oci,no-gpg-verify
```

## 25.4. SEARCHING FOR FLATPAK APPLICATIONS

This procedure searches for an application in the enabled Flatpak remotes on the command line. The search uses the application name and description.

### Prerequisites

- Flatpak is installed.
- The Red Hat Flatpak repository is enabled.

### Procedure

- Search for an application by name:

```
$ flatpak search application-name
```

For example, to search for the **LibreOffice** application, use:

```
$ flatpak search LibreOffice
```

The search results include the ID of the application:

Application ID	Version	Branch	Remotes	Description
org.libreoffice.LibreOffice		stable	rhel	The LibreOffice productivity suite

## 25.5. INSTALLING FLATPAK APPLICATIONS

This procedure installs a selected application from the enabled Flatpak remotes on the command line.

### Prerequisites

- Flatpak is installed.
- The Red Hat Flatpak remote is enabled.

### Procedure

- Install an application from the **rhel** remote:

```
$ flatpak install rhel application-id
```

Replace *application-id* with the ID of the application. For example:

```
$ flatpak install rhel org.libreoffice.LibreOffice
```

## 25.6. LAUNCHING FLATPAK APPLICATIONS

This procedure launches an installed Flatpak application from the command line.

### Prerequisites

- Flatpak is installed.

- The selected Flatpak application is installed.

### Procedure

- Launch the application:

```
$ flatpak run application-id
```

Replace *application-id* with the ID of the application. For example:

```
$ flatpak run org.libreoffice.LibreOffice
```

## 25.7. UPDATING FLATPAK APPLICATIONS

This procedure updates one or more installed Flatpak applications to the most recent version in the corresponding Flatpak remote.

### Prerequisites

- Flatpak is installed.
- A Flatpak remote is enabled.

### Procedure

- Update one or more Flatpak applications:
  - To update a specific Flatpak application, specify the application ID:

```
$ flatpak update application-id
```

- To update all Flatpak applications, specify no application ID:

```
$ flatpak update
```

## 25.8. INSTALLING FLATPAK APPLICATIONS IN THE GRAPHICAL INTERFACE

This procedure searches for Flatpak applications using the **Software** application.

### Prerequisites

- Flatpak is installed.
- The Red Hat Flatpak remote is enabled.

### Procedure

1. Open the **Software** application.
2. Make sure that the **Explore** tab is active.

3. Click the search button in the upper-left corner of the window.
4. In the input box, type the name of the application that you want to install, such as **LibreOffice**.
5. Select the correct application in the search results.  
If the application is listed several times, select the version where the **Source** field in the **Details** section reports **flatpaks.redhat.io**.
6. Click the **Install** button.
7. If **Software** asks you to log in, enter your Customer Portal credentials or your registry service account tokens.
8. Wait for the installation process to complete.
9. Optional: Click the **Launch** button to launch the application.

## 25.9. UPDATING FLATPAK APPLICATIONS IN THE GRAPHICAL INTERFACE

This procedure updates one or more installed Flatpak applications using the **Software** application.

### Prerequisites

- Flatpak is installed.
- A Flatpak remote is enabled.

### Procedure

1. Open the **Software** application.
2. Select the **Updates** tab.
3. In the **Application Updates** section, you can find all available updates to Flatpak applications.
4. Update one or more applications:
  - To apply all available updates, click the **Update All** button.
  - To update only a specific application, click the **Update** button next to the application item.
5. Optional: Enable automatic application updates.
  - a. Click the menu button in the upper-right corner of the window.
  - b. Select **Update Preferences**.
  - c. Enable **Automatic Updates**.  
Flatpak applications now update automatically.