



Red Hat Enterprise Linux 8.0 Beta

Managing RHEL systems from your desktop

A guide to managing Red Hat Enterprise Linux 8 from your desktop

Red Hat Enterprise Linux 8.0 Beta Managing RHEL systems from your desktop

A guide to managing Red Hat Enterprise Linux 8 from your desktop

Legal Notice

Copyright © 2019 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, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document describes how to manage Red Hat Enterprise Linux 8 from the desktop. The title focuses on new features in GNOME and on configuring and managing GNOME at low level.

Table of Contents

THIS IS A BETA VERSION!	3
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	4
CHAPTER 1. GNOME UPDATES	5
1.1. WAYLAND	5
1.1.1. Key Differences Between Wayland and X.Org	5
1.1.1.1. Libinput	5
1.1.1.1.1. Gestures	5
1.1.2. Current Wayland Limitations	5
1.1.2.1. Nvidia drivers	5
1.1.2.2. Remote desktop	6
1.1.2.3. Additional Limitations	6
1.1.3. Switching to X.Org	6
1.2. DESKTOP ICONS	7
CHAPTER 2. CONFIGURING GNOME AT LOW LEVEL	8
2.1. INTRODUCTION TO CONFIGURING GNOME	8
2.2. MANAGING USER AND SYSTEM GNOME SETTINGS	8
2.3. DISPLAYING GSETTINGS VALUES FOR DESKTOP APPLICATIONS	8
2.3.1. Using dconf-editor and gsettings utility	8
2.3.2. Additional information	9
2.4. USING DCONF PROFILES	9
2.4.1. Introduction to dconf profiles	9
2.4.2. Selecting a dconf profile	10
2.5. CONFIGURING CUSTOM DEFAULT VALUES	11
2.5.1. Locking down specific settings	12
2.6. STORING USER SETTINGS OVER NFS	12
2.7. SETTING GSETTINGS KEYS PROPERTIES	13
2.8. WORKING WITH GSETTINGS KEYS ON COMMAND LINE	13
2.8.1. Setting key value	14
2.8.2. Monitoring key changes	14
2.8.3. Checking whether key is writable	15
2.8.4. Checking key valid values	15
2.8.5. Checking description of valid key values	15
2.8.6. Querying key value	16
2.8.7. Resetting key value	16
2.8.8. Resetting schema	16
2.8.9. Listing installed non-relocable schemas	17
2.8.10. Listing schema keys	17
2.8.11. Listing schema children	18
2.8.12. Listing schema's keys and values	18
2.9. ACKNOWLEDGEMENTS	19

THIS IS A BETA VERSION!

Thank you for your interest in Red Hat Enterprise Linux 8.0 Beta. Be aware that:

- Beta code should not be used with production data or on production systems.
- Beta does not include a guarantee of support.
- Feedback and bug reports are welcome. Discussions with your account representative, partner contact, and Technical Account Manager (TAM) are also welcome.
- Upgrades to or from a Beta are not supported or recommended.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
 1. Go to the [Bugzilla](#) website.
 2. As the Component, use **Documentation**.
 3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
 4. Click **Submit Bug**.

CHAPTER 1. GNOME UPDATES

1.1. WAYLAND

The Gnome Display Manager (GDM) uses Wayland as the default display server in Red Hat Enterprise Linux 8, replacing the X.Org server that was used with Red Hat Enterprise Linux 7. However, the X.Org server is still available.

Wayland is a protocol for a compositor to talk to its clients as well as a C library implementation of that protocol. The compositor can be a standalone display server running on Linux kernel modesetting and libinput input devices, an X application, or a Wayland client itself. The clients can be traditional applications, X servers (rootless or fullscreen), or other display servers. For more details, refer to <http://wayland.freedesktop.org>.

New installations of Red Hat Enterprise Linux 8 automatically use Wayland as the display server. When upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8, the current display server setting is inherited. If you previously used X.Org, it is automatically set as the default server in Red Hat Enterprise Linux 8.

There are also a few environments where X.Org is preferred over Wayland, such as:

- Cirrus graphics are used in a VM environment
- Matrox graphics are used

1.1.1. Key Differences Between Wayland and X.Org

1.1.1.1. Libinput

Red Hat Enterprise Linux 7 used the evdev driver as the default driver for input devices. By default, Red Hat Enterprise Linux 8 uses the libinput driver, which also replaces the Synaptic driver. For backward compatibility for certain devices, you can switch to X.Org if necessary as described in [Section 1.1.3, “Switching to X.Org”](#). X.Org still uses evdev as a fallback for devices that are not compatible with libinput. This fallback is automatic.



NOTE

X.Org in Red Hat Enterprise Linux 8 also uses libinput as the default driver, and only switches to evdev in cases of incompatible input devices.

In addition, the wacom driver remains the primary driver for Wacom devices.

1.1.1.1.1. Gestures

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.

1.1.2. Current Wayland Limitations

1.1.2.1. Nvidia drivers

Proprietary Nvidia binary drivers are not supported on Wayland. To avoid any complications while using the Nvidia GPU, switch to X.Org as described in [Section 1.1.3, “Switching to X.Org”](#).

**NOTE**

Nouveau is still supported and is the default driver for Nvidia graphics.

1.1.2.2. Remote desktop

Screen sharing and screencasting are currently not available on Wayland because the VNC protocol, which supported this functionality, is deprecated in Red Hat Enterprise Linux 8. This functionality is still available on X.Org. However, the screen sharing user service pipewire needs to be installed (pipewiere rpm) and running. For more details, refer to [PipeWire project](#).

Due to these limitations you may need to switch to the X.Org display server as described in [Section 1.1.3, “Switching to X.Org”](#).

1.1.2.3. Additional Limitations

The following additional Wayland limitations should be noted:

- Multiple GPU support is not available.
- X.Org screen manipulation utilities are not available.
- The xrandr utility is not supported since Wayland handles layout, rotations, and resolutions differently.
- The GNOME Shell cannot be restarted using the ALT+F2/r method.
- Due to stability issues, using X11 instead of Wayland is recommended in virtual environments.

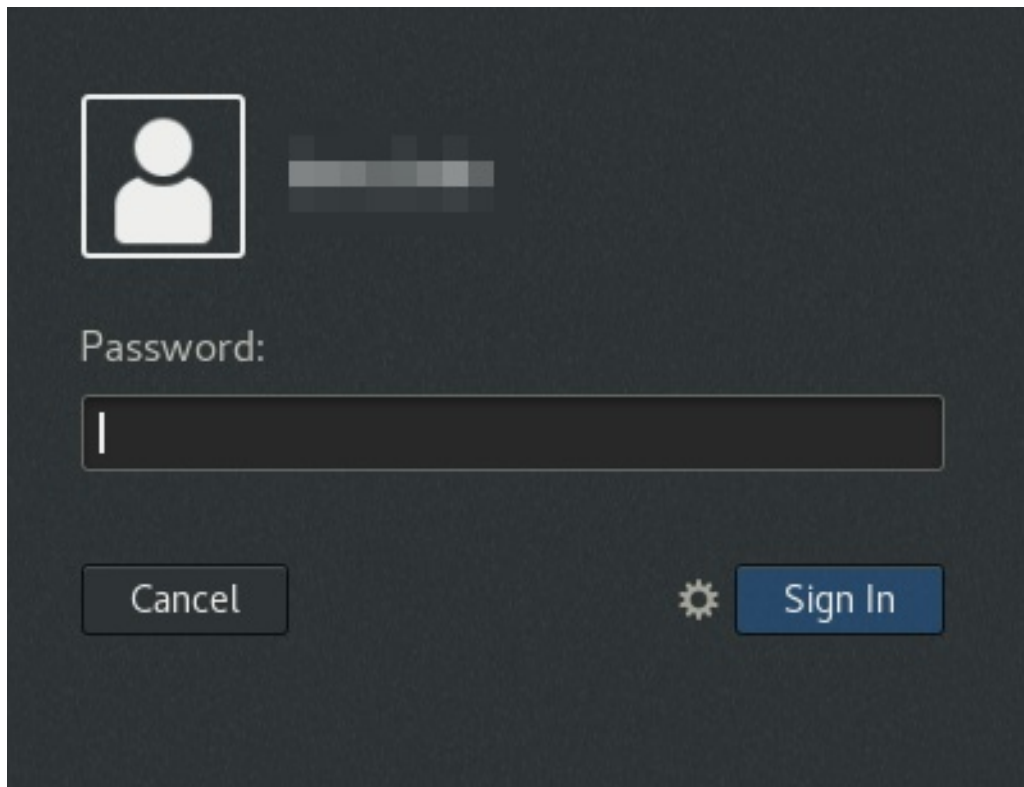
1.1.3. Switching to X.Org

Due to current Wayland limitations described in [Section 1.1.2, “Current Wayland Limitations”](#), you may want to switch to the X.Org display server.

1. From the login screen (GDM), click the cogwheel 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 **Standard (X11 display server) on Xorg** or **Classic (X11 display server)**.

Optionally, if you typically bypass the login screen, you can select the X.Org server by adding the following entry to the `/etc/gdm/custom.conf` file:

```
WaylandEnable=false
```

1.2. DESKTOP ICONS

Desktop icon functionality has been removed with Nautilus version 3.28, since Nautilus is no longer maintaining the desktop. By default, no desktop icons are displayed. If you want to use this functionality, install and enable the desktop icons gnome-shell extension provided by the `gnome-shell-extension-desktop-icons` rpm.

CHAPTER 2. CONFIGURING GNOME AT LOW LEVEL

2.1. INTRODUCTION TO CONFIGURING GNOME

To be able to configure the GNOME Desktop Environment, you need to understand these basic terms:

- `dconf`
- `GSettings`
- `gsettings`

dconf has two different meanings.

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

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

GSettings is a high-level API for application settings which serves as the front end for `dconf`.

gsettings is a command-line tool which is used to view and change user settings.

2.2. MANAGING USER AND SYSTEM GNOME SETTINGS

dconf allows system administrators and users several levels of control over GNOME configuration:

- 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.

2.3. DISPLAYING GSETTINGS VALUES FOR DESKTOP APPLICATIONS

2.3.1. Using `dconf-editor` and `gsettings` utility

Viewing and editing of the **GSettings** values can be achieved with one of the following tools:

- `dconf-editor` GUI tool
- `gsettings` command-line utility



NOTE

The **dconf-editor** is not installed on the system by default. To install it, run the following command as the **root** user:

```
~]# yum install dconf-editor
```

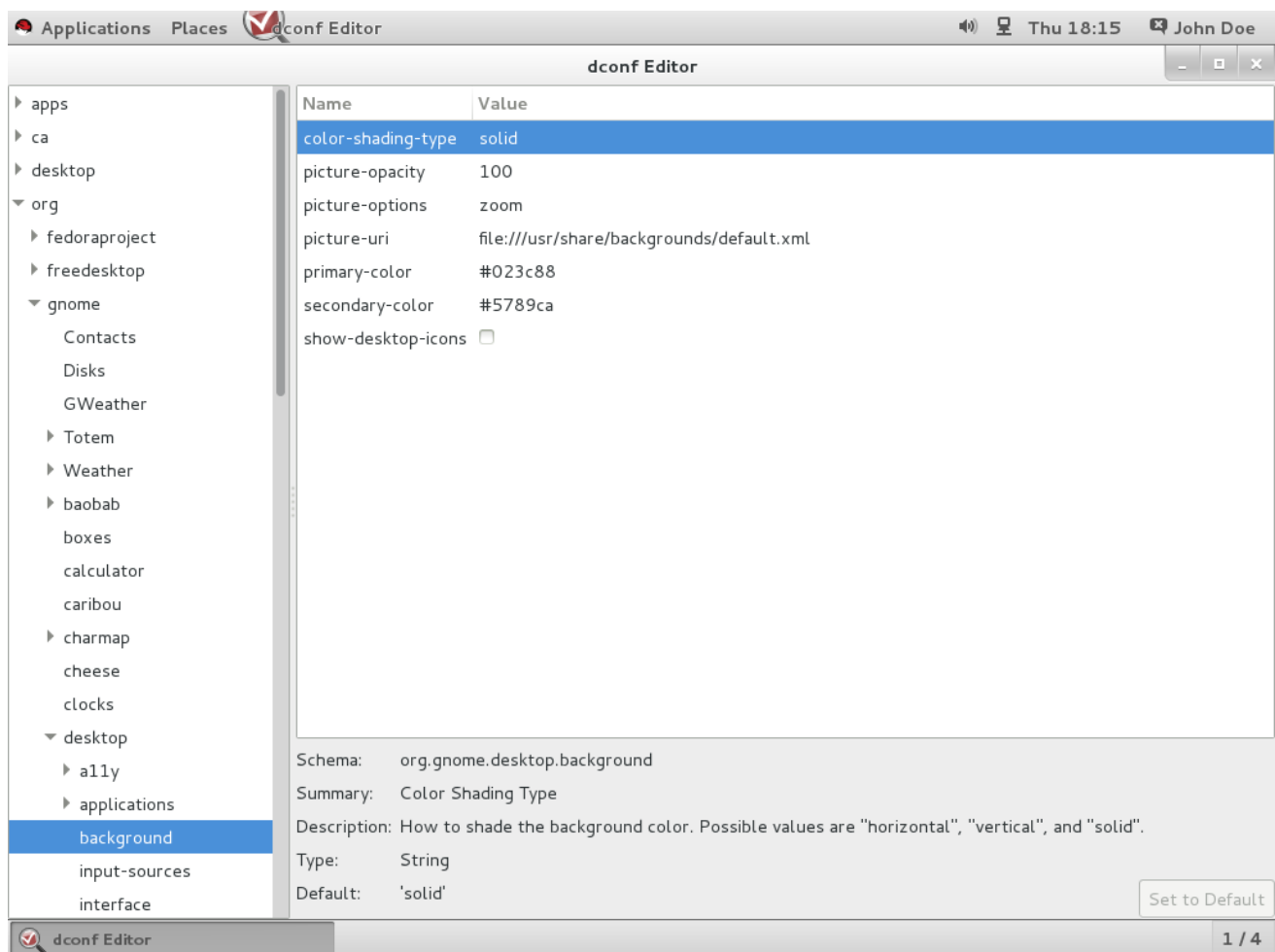
The **dconf-editor** and **gsettings utility** has the following in common:

- allow browsing and changing options for system and application preferences
- allow to change preferences
- can be run by regular users, because both tools are intended to browse and modify the current user's GSettings database

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

Figure 2.1. dconf-editor showing org.gnome.desktop.background GSettings keys



2.3.2. Additional information

- For more information on the **dconf-editor** tool, see the **dconf-editor(1)** man page and the [dconf-editor Project documentation](#).
- For more information on the **gsettings** utility, see the **gsettings(1)** man page.

2.4. USING DCONF PROFILES

2.4.1. Introduction to dconf profiles

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

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

The dconf system stores its profiles in the text files which 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 which are set in a dconf profile override the default settings.

2.4.2. Selecting a dconf profile

On startup, **dconf** consults the `$DCONF_PROFILE` environment 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** is the name of the user database which can be found in `~/ .config/dconf`, and **local** and **site** are 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.



WARNING

As a user or application developer, do not manipulate dconf directly. To manipulate dconf, always use the **dconf-editor** or the **gsettings utility**. The only exception to use dconf directly is when setting system-wide default configurations, because the aforementioned tools do not allow to manipulate such configurations.

2.5. CONFIGURING CUSTOM DEFAULT VALUES

Machine-wide default settings can be set by providing a default for a key in a dconf profile. These defaults can be overridden by the user.

Setting a default for a key has two prerequisites:

- the user profile exists
- the value for the key was added to a dconf database

For example, to set the default background:

1. Create the user profile in `/etc/dconf/profile/user`:

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

where **local** is the name of a dconf database.

2. Create a keyfile for the local database in `/etc/dconf/db/local.d/01-background`, which contains the following default settings:

```
~]# dconf path

[org/gnome/desktop/background]

# GSettings key names and their corresponding values
picture-uri='file:///usr/local/share/backgrounds/wallpaper.jpg'
picture-options='scaled'
primary-color='000000'
secondary-color='FFFFFF'
```

In the default setting of the keyfile, the following GSettings keys are used:

Table 2.1. org.gnome.desktop.background schemas GSettings Keys

Key Name	Possible Values	Description
picture-options	"none", "wallpaper", "centered", "scaled", "stretched", "zoom", "spanned"	Determines how the image set by wallpaper_filename is rendered.
picture-uri	filename with the path	BURI to use for the background image. Note that the backend only supports local file:// URIs.
primary-color	default: 000000	Left or Top color when drawing gradients, or the solid color.
secondary-color	default: FFFFFFFF	Right or Bottom color when drawing gradients, not used for solid color.

1. Edit the keyfile according to your preferences.
For more information, see [Section 2.3, “Displaying GSettings values for desktop applications”](#).
2. Update the system databases:

```
~]# dconf update
```

When the user profile is created or changed, the user needs to log out and log in again before the changes will be applied.

If you want to avoid creating a user profile, you can use the **dconf command-line 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.

2.5.1. Locking down specific settings

By using the lockdown mode in dconf, you can prevent users from changing specific settings.

To lock down a GSettings key:

1. Create a locks subdirectory in the keyfile directory such as **/etc/dconf/db/local.d/locks/**.
2. Add any number of files with keys that you want to lock into this directory.

Without enforcing the system settings using a lockdown, any settings that users make take precedence over the system settings. User can thus override the system settings with their own.

For example, to lock settings for the default wallpaper:

1. Set a default wallpaper.
2. Create a new **/etc/dconf/db/local.d/locks/** directory.
3. Create a new file in **/etc/dconf/db/local.d/locks/00-default-wallpaper** with the following contents, 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
```

2.6. STORING USER SETTINGS OVER NFS

For **dconf** to work correctly when using Network File System (NFS) home directories, the **dconf keyfile back** end must be used.

Note that **dconf keyfile back end** only works properly if the **glib2-fam** package is installed. Without this package, notifications on configuration changes made on remote machines are not displayed properly.

With Red Hat Enterprise Linux 8, **glib2-fam** is available in the BaseOs repository.

To set the **dconf keyfile back end**:

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

To verify whether the package is installed on the system:

```
~]# yum list installed
```

If **glib2-fam** is not in the list of installed packages, install it by running:

```
~]# yum install glib2-fam
```

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

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

The **dconf keyfile back end** takes effect the next time that the user logs in. It polls the keyfile to determine whether updates have been made, so settings may not be updated immediately.

2.7. SETTING GSETTINGS KEYS PROPERTIES

This section describes how to set GSettings keys properties for a single-logged user.

Each GSettings key can have only one value in a dconf database. Setting the same key to a different value at a different place of the dconf database overrides the previous value.

Values of some keys are of array type. For array type, you can specify the value of the key as a list of multiple elements separated by a comma.

To set a GSettings key of array type, follow this syntax:

```
key=['option1', 'option2']
```

The following example shows setting of the **org.gnome.desktop.input-sources.xkb-options** GSettings key whose value is of array type:

Example settings of the **org.gnome.desktop.input-sources.xkb-options** GSettings Key

```
[org/gnome/desktop/input-sources]
# Enable Ctrl-Alt-Backspace for all users
# Set the Right Alt key as the Compose key and enable it
xkb-options=['terminate:ctrl_alt_bksp', 'compose:ralt']
```

2.8. WORKING WITH GSETTINGS KEYS ON COMMAND LINE

This section focuses on using of the **gsettings** command to configure, manipulate and manage the GSettings keys. The most frequent use cases that can be resolved by using the **gsettings** command are shown.

2.8.1. Setting key value

To set a value of a key:

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

Note that the value is specified as a serialised GVariant.

Example 2.1. Adding selected applications into the favorite applications key

To add selected applications among your favorite applications:

```
$ gsettings set org.gnome.shell favorite-apps "['firefox.desktop',  
'evolution.desktop', 'rhythmbox.desktop', 'shotwell.desktop',  
'org.gnome.Nautilus.desktop', 'org.gnome.Software.desktop',  
'yelp.desktop', 'org.gnome.Terminal.desktop',  
'org.gnome.clocks.desktop']"
```

If the operation succeeds, no return code is shown. As a result, all listed applications are added to favorite applications. The change is valid immediately.

2.8.2. Monitoring key changes

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

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

Note that if the KEY argument is not specified, all keys in the schema are monitored. Monitoring continues until the process is terminated.

Example 2.2. Monitoring changes of the favorite applications key

To monitor the changes of the **favorite applications** key, open two terminals and run:

In the first terminal:

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

In the second terminal:

```
$ gsettings set org.gnome.shell favorite-apps "['firefox.desktop',  
'evolution.desktop', 'rhythmbox.desktop', 'shotwell.desktop',  
'org.gnome.Nautilus.desktop', 'org.gnome.Software.desktop',  
'yelp.desktop', 'org.gnome.Terminal.desktop']"
```

As a result, a notification whether and how **favorite applications** changed is displayed in the first terminal:

```
favorite-apps: ['firefox.desktop', 'evolution.desktop',
'rhythmbox.desktop', 'shotwell.desktop', 'org.gnome.Nautilus.desktop',
'org.gnome.Software.desktop', 'yelp.desktop',
'org.gnome.Terminal.desktop']
```

2.8.3. Checking whether key is writable

To check whether a key is writable:

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

Example 2.3. Checking whether the favorite applications key is writable

To check whether the **favorite applications** key is writable:

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

As a result, the return code shows **True**.

2.8.4. Checking key valid values

To check the range of valid values for a key:

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

Example 2.4. Checking the range of valid values for the remember-mount-password key

To check valid values for the **remember-mount-password** key:

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

As a result, the return code displays type of the key value, which is **type b** in this particular case. For more information, see [GNOME developer](#).

2.8.5. Checking description of valid key values

To check the description of valid values for a key:

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

Example 2.5. Checking the description of valid values for the picture-uri key

To check the description of valid values for the **picture-uri** key:

```
$ gsettings describe org.gnome.desktop.screensaver picture-uri
```

As a result, the following output is displayed:

```
-
```

URI to use for the background image. Note that the backend only supports local `file://` URIs.

2.8.6. Querying key value

To get the value of a key:

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

Note that the value is displayed as a serialised GVariant.

Example 2.6. Querying value of the remember-mount-password key

To get value of the `remember-mount-password` key:

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

As a result, the return code displays `false`.

2.8.7. Resetting key value

To reset the value of a key:

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

If resetting succeeds, no return code is displayed. Default values are in stored `dconf` and `gsettings-desktop-schemas` files.

Example 2.7. 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.

Users can set the value of `lock-delay` as needed.

For example, to set the `lock-delay` key for screensaver to 200:

```
$ gsettings set org.gnome.desktop.screensaver lock-delay 200
```

To reset the `lock-delay` key for screensaver to its default value:

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

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

2.8.8. Resetting schema

To reset a schema:

```
gsettings reset-recursive SCHEMA [:PATH]
```

Example 2.8. Resetting the `org.gnome.desktop.screensaver` schema to its defaults

To reset the `org.gnome.desktop.screensaver` schema to its defaults:

```
$ gsettings reset-recursive 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.

2.8.9. Listing installed non-relocable schemas

To list installed schemas that are non-relocable:

```
gsettings list-schemas [--print-paths]
```

If the `--print-paths` argument is specified, the path where each schema is mapped is printed as well.

Example 2.9. Listing installed non-relocable schemas

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

```
$ gsettings list-schemas
```

As a result, a full list of schemas is returned. The following list is truncated.

```
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
```

2.8.10. Listing schema keys

To list the keys that are in the selected schema:

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

Example 2.10. Listing keys in the `org.gnome.shell` schema

To list keys in the `org.gnome.shell` schema:

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

As a result, a list of keys is returned. The following list is truncated.

```
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
```

2.8.11. Listing schema children

To list children of a selected schema:

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

Note that the list is empty if there are no children.

Example 2.11. Listing children of the `org.gnome.shell` schema

To list children of the `org.gnome.shell` schema:

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

As a result, the following output is returned:

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

2.8.12. Listing schema's keys and values

To list keys and values of a selected schema recursively:

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

Note that if the schema whose keys you want to list is not specified, all keys within all schemas are listed.

Example 2.12. Listing keys and values recursively

To list keys and values in all schemas recursively:

```
$ gsettings list-recursively
```

As a result, all key and values in all schemas on system are listed, as shown below. Note that the following list is truncated.

```
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-accelers false
org.gnome.Vinagre history-size 15
org.gnome.Vinagre shared-flag true
```

2.9. 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.