



Red Hat Enterprise Linux 7

桌面迁移和管理指南

RHEL 7 中的 GNOME 3 桌面迁移规划、部署、配置和管理

Red Hat Enterprise Linux 7 桌面迁移和管理指南

RHEL 7 中的 GNOME 3 桌面迁移规划、部署、配置和管理

Marie Doleželová

Red Hat Customer Content Services

mdolezel@redhat.com

Petr Kovář

Red Hat Customer Content Services

pkovar@redhat.com

Jana Heves

Red Hat Customer Content Services

法律通告

Copyright © 2018 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[®] 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 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.

摘要

桌面迁移 和管理指南介绍了 Red Hat Enterprise Linux 7 上的 GNOME 3 桌面迁移规划、部署、配置和管理。它面向对 Red Hat Enterprise Linux 系统有基本了解的系统管理员。

目录

| | |
|--|-----------|
| 第 1 章 GNOME 3 桌面介绍 | 4 |
| 1.1. GNOME 3 是什么？ | 4 |
| 1.2. 什么是 GNOME SHELL？ | 5 |
| 1.3. GNOME 经典是什么？ | 7 |
| 1.4. 有关可访问性的说明 | 9 |
| 部分 I. 迁移规划 | 11 |
| 第 2 章 LOGIND | 12 |
| 获取更多信息 | 14 |
| 第 3 章 GSETTINGS 和 DCONF | 15 |
| 获取更多信息 | 16 |
| 第 4 章 POLICYKIT | 17 |
| 4.1. 策略配置 | 17 |
| 4.2. 默认策略 | 18 |
| 4.3. 检查脚本的权限 | 18 |
| 4.4. 扩展 POLKIT 配置 | 18 |
| 第 5 章 GDM | 20 |
| 获取更多信息 | 20 |
| 第 6 章 GNOME SHELL 扩展 | 21 |
| 6.1. 替换时钟小程序 | 21 |
| 第 7 章 GNOME-SESSION | 22 |
| 获取更多信息 | 22 |
| 第 8 章 国际化 | 23 |
| 8.1. 输入法 | 23 |
| 8.2. 更改的文件位置 | 23 |
| 部分 II. 配置和管理 | 25 |
| 第 9 章 使用 GSETTINGS 和 DCONF 配置桌面 | 26 |
| 9.1. 术语解释：GSETTINGS、GSETTINGS 和 DCONF | 26 |
| 9.2. 用户和系统设置 | 26 |
| 9.3. 浏览桌面应用程序的 GSETTINGS 值 | 26 |
| 9.4. 什么是 DCONF 配置文件？ | 28 |
| 9.5. 配置自定义默认值 | 29 |
| 9.6. GSETTINGS 键属性 | 32 |
| 9.7. 通过 NFS 存储用户设置 | 32 |
| 第 10 章 默认外观 | 34 |
| 10.1. 创建 ANACONDA 品牌 | 34 |
| 10.2. 引导加载程序屏幕 | 34 |
| 10.3. PLYMOUTH | 36 |
| 10.4. 自定义登录屏幕 | 40 |
| 10.5. 自定义桌面背景 | 44 |
| 10.6. 配置字体 | 48 |
| 第 11 章 GNOME SHELL 扩展 | 57 |
| 11.1. GNOME SHELL 扩展是什么？ | 57 |
| 11.2. 启用机器范围的扩展 | 59 |

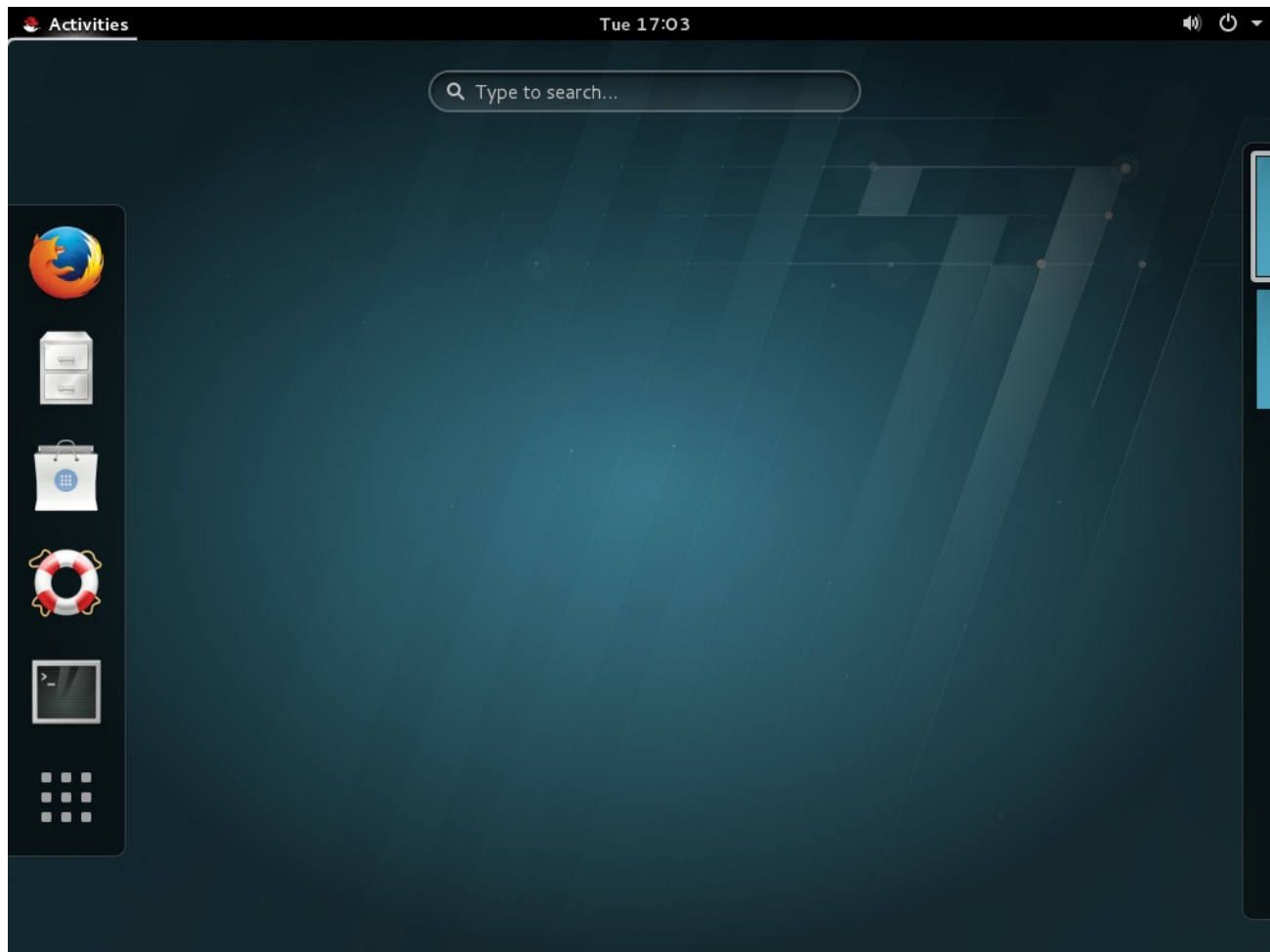
| | |
|-----------------------------------|------------|
| 11.3. 锁定启用的扩展 | 60 |
| 11.4. 设置强制扩展 | 61 |
| 第 12 章 集成应用程序 | 63 |
| 12.1. 定制菜单 | 63 |
| 12.2. 自定义默认喜欢的应用程序 | 67 |
| 12.3. 配置文件关联 | 69 |
| 第 13 章 自定义 GNOME 桌面功能 | 78 |
| 13.1. 允许和禁止在线帐户 | 78 |
| 13.2. 启用 CTRL+ALT+BACKSPACE 快捷键 | 79 |
| 13.3. 启用组合键 | 80 |
| 13.4. 禁用命令行访问 | 82 |
| 13.5. 锁定打印 | 83 |
| 13.6. 锁定保存在磁盘上的文件 | 84 |
| 13.7. 锁定重新分区 | 85 |
| 13.8. 锁定用户退出登录和用户切换 | 87 |
| 13.9. 单应用程序模式 | 88 |
| 13.10. 防止计算机在合上盖子时挂起 | 91 |
| 13.11. 在图形目标模式下按下 POWER 按钮时，会更改行为 | 93 |
| 第 14 章 会话管理 | 95 |
| 14.1. GDM 是什么？ | 95 |
| 14.2. 身份验证 | 96 |
| 14.3. 用户会话 | 99 |
| 第 15 章 虚拟文件系统和磁盘管理 | 110 |
| 15.1. GVFS | 110 |
| 15.2. GVFS 的后端 | 110 |
| 15.3. 挂载、卸载和弹出 | 112 |
| 15.4. 管理书签 | 113 |
| 15.5. 配置默认服务器列表 | 114 |
| 15.6. 向所有其他应用程序公开 GNOME 虚拟文件系统 | 115 |
| 15.7. GVFS 挂载的密码管理 | 115 |
| 15.8. GNOME 中的 GVFS 工具和 XDG-UTILS | 116 |
| 15.9. 使用 GVFS 元数据 | 118 |
| 15.10. 故障排除 | 119 |
| 第 16 章 硬件配置 | 123 |
| 16.1. 平板电脑 | 123 |
| 附录 A. KDE PLASMA 工作区 | 127 |
| 附录 B. 访问红帽文档 | 128 |
| B.1. 产品文档 | 128 |
| B.2. RED HAT ACCESS GUI | 128 |
| 附录 C. 致谢 | 130 |
| 附录 D. 修订历史记录 | 131 |

第 1 章 GNOME 3 桌面介绍

1.1. GNOME 3 是什么？

在 Red Hat Enterprise Linux 7 中，GNOME 3 是默认的桌面环境。这是 GNOME 桌面的下一个主要版本，它比之前随 Red Hat Enterprise Linux 5 和 6 一起提供的 GNOME 2 桌面相比，引入了一个新的用户界面和大量的功能改进。

图 1.1. GNOME 3 桌面(GNOME 经典)



GNOME 3 提供了一个鼓励工作效率的专注的工作环境。强大的搜索功能让您从一个地方可以访问所有的工作。例如，当您需集中精力处理手头的任务时，您可以关闭通知。

重要

为了正常工作，GNOME 要求您的系统支持 **3D 加速**。这包括裸机系统，以及 VMWare 等 hypervisor 解决方案。

如果在 VMWare 虚拟机(VM)上启动或性能不佳，请查看以下解决方案：[为什么 GUI 无法在 VMware 虚拟机上启动？](#)

如需更多信息，请参阅 [第 1.2.1 节“硬件加速和软件渲染”](#)。

GNOME 3 构建在多个功能强大的组件之上：

GNOME Shell

GNOME Shell 是一个现代的、直观的图形用户界面。它提供了高质量的用户体验，包括视觉效果和硬件加速支持。

GNOME 经典

GNOME 经典集新旧版本于一身；它保留了 GNOME 2 熟悉外观和感受，但增加了 GNOME Shell 的强大新特性和 3-D 功能。GNOME 经典是 Red Hat Enterprise Linux 7 中默认的 GNOME 会话和 GNOME Shell 模式。

GSettings

GSettings 是一个配置存储系统，替换了旧的 GNOME 版本中的 **GConf**。有关从 **GConf** 转换到 **GSettings** 的更多信息，请参阅 [第 3 章 GSettings 和 dconf](#)。要了解更多有关使用 **GSettings** 配置桌面的信息，请参阅 [第 9 章 使用 GSettings 和 dconf 配置桌面](#)。

GVFS

GVFS 提供完整的虚拟文件系统基础架构，一般处理 GNOME 桌面中的存储。通过 **GVFS**，GNOME 3 与在线文档存储服务、日历和联系列表良好集成，以便您的所有数据都可以从同一位置访问。阅读 [第 15 章 虚拟文件系统和磁盘管理](#) 中的 **GVFS** 的更多信息。

GTK+

GTK+ 是用于创建图形用户界面的多平台工具包，提供了一个高度丰富的功能丰富的 API。由于 **GTK+**，GNOME 3 能够更改应用程序的外观或提供平滑图形的外观。此外，**GTK+** 包含很多功能，如面向对象的编程支持(GObject)、国际字符集和文本布局的广泛支持(Pango)或一组可访问性接口(ATK)。

1.2. 什么是 GNOME SHELL ?

GNOME Shell 是 GNOME 桌面的用户界面，GNOME 3 的关键技术。它提供基本的用户界面功能，如切换窗口、启动应用程序或显示通知。

GNOME Shell 引入了创新的用户界面概念，来提供高质量的用户体验，包括在带有现代图形硬件的系统上的硬件加速。

GNOME Shell 用户界面的一些主要组件包括：

顶部栏。

屏幕顶部的水平栏提供了对 GNOME Shell 的一些基本功能的访问，如 **活动概览**、时钟和日历、系统状态图标以及屏幕左上角的系统菜单。

系统菜单。

系统菜单 位于右上角。您可以更新一些设置，查找有关 **Wi-Fi** 连接的信息，切换用户、退出登录，并从这个菜单关闭计算机。

活动概览。

活动概览 提供了窗口和应用程序视图，允许用户运行应用程序和窗口，并在它们之间进行切换。

顶部的搜索条目允许搜索桌面上可用的各种项目，包括应用程序、文档、文件和配置工具。

左侧的竖栏称为 **短划线**，它包含一个喜欢和运行的应用程序的列表。

工作区列表 显示在右侧，允许用户在多个工作区之间切换，或者将应用程序和窗口从一个工作区移到另一个工作区。

消息栏。

消息栏 是接近屏幕底部的水平栏，显示用户按下 **Super+M**。它提供对待处理的通知的访问。

特定于 **GNOME 经典** 的组件。

GNOME 经典 是 Red Hat Enterprise Linux 7 中默认的 **GNOME Shell** 模式。它改变了 **GNOME Shell** 行为的某些方面，以及 **GNOME Shell** 外观。它包括带有窗口列表的底部栏，以及顶栏中的 **Applications** 和 **Places** 菜单。有关 **GNOME 经典** 的详情，请参考 [第 1.3 节“GNOME 经典是什么？”](#)。

1.2.1. 硬件加速和软件渲染

GNOME Shell 具有视觉效果，利用由 **Clutter** 提供的硬件加速支持，这是一个基于 **OpenGL** 的图形库。

要使硬件加速能够正常工作，图形驱动程序必须支持 **GL 1.2** 和多文本扩展，或 **GL 1.3**。另外，驱动程序必须提供对 **GLES 1.1** 或 **GLES 2.0** 的支持。请记住，许多 **GPU** 型号和驱动程序都没有正确实现对 **GL** 或 **GLES** 的支持，因此带有这些 **GPU** 和驱动程序的系统上的硬件加速可能无法使用。

在没有满足 **GPU** 和驱动程序要求的系统上（包括虚拟机），软件渲染用于提供与支持的硬件加速所提供的相同的 **GNOME 3** 用户体验。软件渲染由 **LLVM pipe** 驱动程序 提供。

要确定系统是否使用软件渲染和 **LLVM pipe** 驱动程序，您可以运行 **glxinfo** 命令：

```
$ glxinfo | grep renderer
OpenGL renderer string: Gallium 0.4 on llvmpipe (LVVM 3.3, 128 bits)
```

请注意，因为软件渲染器不提供完全兼容的 **OpenGL** 实现，因此如果一些程序依赖于在应用程序之间具有一致的 **GLX** 状态视图的 **X** 服务器，则这些程序可能无法正常工作。考虑升级硬件，或者在具有完全支持硬件加速的 **GPU** 和驱动程序的系统上运行这些程序。

1.3. GNOME 经典是什么？

GNOME 经典是一种 GNOME Shell 特性和模式，适用于那些喜欢更传统的桌面体验的用户。虽然 GNOME 经典基于 GNOME 3 技术，但它提供了许多对用户界面的更改：

Applications 和 Places 菜单。

Applications 菜单显示在屏幕的左上角。它允许用户访问按类别组织的应用程序。用户也可以从该菜单中打开 **Activities Overview**。

Places 菜单显示在 **顶栏**上的 **Applications 菜单**旁边。它允许用户快速访问重要文件夹，如 **Downloads** 或 **Pictures**。

任务栏。

任务栏显示在屏幕底部，具有以下特性：

- 一个窗口列表，
- 通知图标显示在窗口列表的旁边，
- 当前工作区的简短标识，可用工作区的总数显示在通知图标的旁边。

四个可用的工作区：

在 GNOME 经典中，用户可使用的工作空间数默认设置为 4。

最小化和最大化按钮。

GNOME 经典中的窗口标题栏具有最小化和最大化按钮的功能，使用户能够快速最小化到窗口列表中，或者将它们最大化以占据桌面上的所有空间。

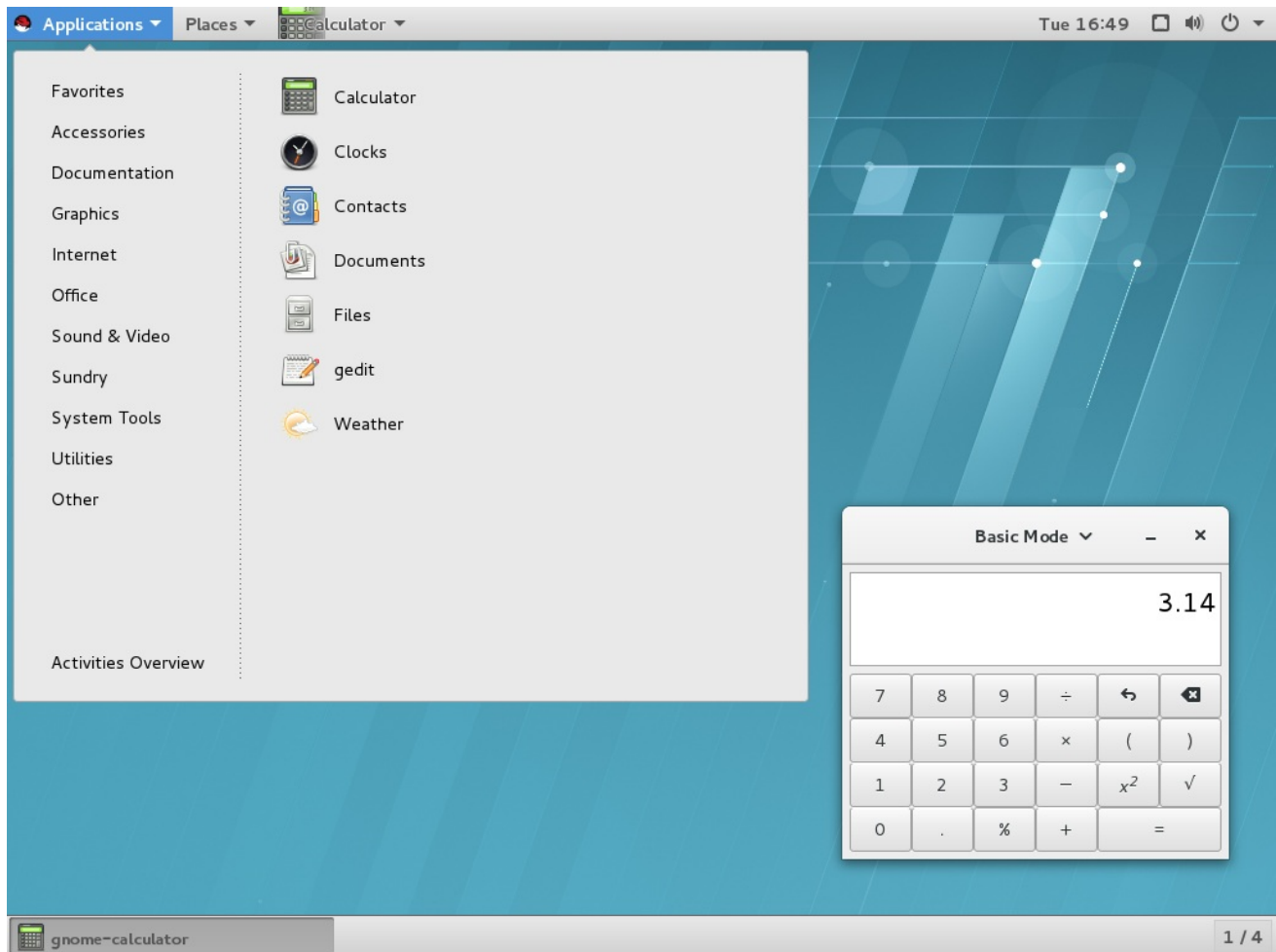
传统的 Super+Tab 窗口切换器。

在 GNOME 经典中，Super+Tab 窗口切换器中的窗口不会按应用程序分组。

系统菜单。

系统菜单 位于右上角。您可以更新一些设置，查找有关 Wi-Fi 连接的信息，切换用户、退出登录，并从这个菜单关闭计算机。

图 1.2. 带有计算器应用程序的 GNOME 经典和应用程序菜单的附件子菜单



1.3.1. GNOME 经典扩展

GNOME 经典作为一组 **GNOME Shell 扩展** 分发。GNOME 经典扩展作为 `gnome-classic-session` 软件包的依赖项被安装，它提供了运行 GNOME 经典会话所需的组件。由于 Red Hat Enterprise Linux 7 上默认启用了 GNOME 经典扩展，因此 GNOME 经典是默认的 Red Hat Enterprise Linux 7 桌面用户界面。

- **AlternateTab** (alternate-tab@gnome-shell-extensions.gcampax.github.com),
- **Applications Menu** (apps-menu@gnome-shell-extensions.gcampax.github.com),
- **启动新实例** (launch-new-instance@gnome-shell-extensions.gcampax.github.com)

- 位置状态指示器(1)
- 窗口列表(window-list@gnome-shell-extensions.gcampax.github.com)。

1.3.2. 从 GNOME 经典切换到 GNOME ， 并切换回来

用户可以通过退出登录并单击 **Sign In** 旁边的齿轮，从 GNOME 经典切换到 GNOME。齿轮会打开一个下拉菜单，其中包含 GNOME 经典。

要在用户会话中从 GNOME 经典切换到 GNOME，请运行以下命令：

```
$ gnome-shell --mode=user -r &
```

要在同一个用户会话中切回到 GNOME 经典，请运行以下命令：

```
$ gnome-shell --mode=classic -r &
```

1.3.3. 禁用 GNOME 经典作为默认会话

对于所有 Red Hat Enterprise Linux 7 上新创建的用户，GNOME 经典被设为默认会话。要覆盖特定用户的此设置，您需要在 `/var/lib/AccountsService/users/username` 文件中修改用户的帐户服务。有关如何进行此操作的详情，请查看 [第 14.3.2 节“配置用户默认会话”](#)。

获取更多信息

用户可以在 `gnome-user-docs` 软件包提供的 GNOME Help 中找到有关使用 GNOME 3、GNOME Shell 或 GNOME 经典的更多信息。要访问 GNOME 帮助，请按 Super 键进入 Activities Overview，输入 help，然后按 Enter 键。

1.4. 有关可访问性的说明

GNOME 桌面包括辅助技术，以支持有各种障碍和特殊需求的用户，并与常见的辅助设备交互。Universal access 菜单 为具有各种不灵活性的用户提供快速设置。图标放置在顶栏上，看起来像一个人被一个圆圈包围：



所有可访问性功能都记录在 **GNOME 帮助** 中，通过打开 "活动概览" 并键入 *help* 即可轻松访问。在 **GNOME 帮助** 菜单中选择 **Universal access**。



注意

要将 **GNOME** 作为视障人员访问，请按 **Super+Alt+S** 组合键，该组合在屏幕阅读器上切换。您可以在其 [帮助页面](#) 上找到更多有关使用 *Orca* 屏幕阅读器的详情。

部分 I. 迁移规划

迁移规划的重点是将默认的 Red Hat Enterprise Linux 桌面环境从与 Red Hat Enterprise Linux 5 和 6 一起提供的 GNOME 2 转移到 GNOME 3。本指南的一部分逐一简要介绍了某些组件所经历的更改，并描述了组件所具有的新特性。

本指南只涉及对 GNOME 桌面环境的更改。有关 Red Hat Enterprise Linux 7 其他部分的更改，请参考：

- ***Red Hat Enterprise Linux 7 系统管理员指南***，用于 GRUB 2 引导装载程序、软件包管理、systemd 或打印机配置等组件。
- ***Red Hat Enterprise Linux 7 迁移规划指南***概述了 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 之间在行为和兼容性方面的主要变化。迁移规划指南还介绍了红帽提供的工具，以帮助升级到 Red Hat Enterprise Linux 7。
- ***有关安装 Red Hat Enterprise Linux 7 和使用 Anaconda 安装程序的详细信息，Red Hat Enterprise Linux 7 安装指南***。

这些文档可在 http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/ 中找到。

第 2 章 LOGIND

logind（或者更具体地说是 **systemd-logind**）是管理用户登录的系统服务。此服务负责以下操作：

- 记录用户和会话、它们的进程及空闲状态，
- 为用户进程创建控制组，
- 为用户提供基于 **PolicyKit** 的访问，如系统关闭或休眠等操作。
- 为应用程序实现一个关闭/休眠阻止逻辑，
- 处理电源/休眠硬件键，
- 用户的多席位管理、会话切换管理和设备访问管理。
- 对虚拟终端（控制台）激活和用户运行时目录管理自动生成文本登录(gettys)。

logind 服务与 **systemd** 深度集成(Red Hat Enterprise Linux 7 中的新初始化系统)，并替换了 Red Hat Enterprise Linux 6 中的 **upstart** 初始化系统。这一变化带来了许多新的特性和功能。以下是对其中一些最重要的总结：

ConsoleKit

ConsoleKit 框架在 Red Hat Enterprise Linux 7 中已弃用。现在，**systemd** 提供了等效的功能。**ConsoleKit** 和 **logind** 都是用于跟踪当前运行的用户会话的服务。



注意

ConsoleKit 能够在系统中的活动会话更改时运行任意 **shell** 脚本（使用虚拟终端切换）。不再提供此功能。

`/var/log/ConsoleKit/history` 文件

在以前的版本中，ConsoleKit 将日志文件发送到 `/var/log/ConsoleKit/history`，当前 logind 不支持。该文件已被传统的 `wtmp` 和 `utmp` 文件替代，这些文件现在跟踪系统中的所有登录和注销。`/var/log/ConsoleKit/history` 提供与 `wtmp` 文件类似的信息，但采用不同的格式。由于功能重叠，logind 仅采用 `wtmp` 文件的角色。

seat.d 脚本

由于 ConsoleKit 不再在使用中，因此 `seat.d` 脚本不再补充 ConsoleKit 框架，且已被 `systemd-logind` 替代。

ck-list-sessions 命令

ConsoleKit 提供了 `ck-list-sessions` 命令，该命令返回了有关最新用户的扩展信息，不仅返回常规用户，还有使用 GDM 的 GUI 访问。现在，可以通过运行 `loginctl` 命令达到类似的结果：

```
$ loginctl list-sessions
```

多席位支持

logind 与 GDM 一起提供多席位功能，用户可以与其机器附加另一个 monitor、鼠标或键盘。这样做会出现另一个登录屏幕，用户可以登录，就像他们在使用另一台机器一样。

要列出系统上可用的席位，请运行以下命令：

```
$ loginctl list-seats
```

要显示系统上特定座位的状态，请运行以下命令：

```
$ loginctl seat-status seat
```

其中 `seat` 是席位的名称，如 `seat0`。

要给特定的座位分配特定的硬件，请运行以下命令：

```
# loginctl attach seat device
```

其中 `seat` 是席位的名称，如 `seat1`，`device` 是使用 `/sys` 设备路径指定的设备名称，如 `/sys/devices/pci0000:00/0000:00:02.0/drm/card0`。

要更改分配，请将硬件分配给不同的席位，或使用 `loginctl flush-devices` 命令。

获取更多信息

systemd-logind.service(8)- `logind` 的手册页提供了有关 `logind` 使用和功能的更多信息。它还涵盖了 `systemd-logind` 提供的 API (`logind D-Bus API 文档`)。

logind.conf(5)- `logind.conf` 手册页讨论了登录管理器配置文件。

loginctl(1)- `systemd` 登录管理器的手册页包括了更多有关多席位功能的信息。

第 3 章 GSETTINGS 和 DCONF

Red Hat Enterprise Linux 7 的主要变化是从 GConf（用于存储用户首选项）转变为 GSettings 高级配置系统和 dconf 后端的组合。

GConf

如前文所述，G Conf 配置系统已被两个系统替代：

- GSettings API，以及
- dconf 后端充当低级配置系统和以单一紧凑的二进制格式收集系统硬件和软件配置详情的程序。

`gsettings` 命令行工具和 `dconf` 工具都用于查看和更改用户设置。`gsettings` 工具直接在终端中执行，而 `dconf` 工具使用 `dconf-editor` GUI 编辑配置数据库。有关 `dconf-editor` 和 `gsettings` 工具的更多信息，请参阅 [第 9 章 使用 GSettings 和 dconf 配置桌面](#)。

gconftool

`gconftool-2` 工具已被 `gsettings` 和 `dconf` 替代。同样，`gconf-editor` 已被 `dconf-editor` 替代。

覆盖

Red Hat Enterprise Linux 7 中引入了 *keyfiles* 概念：`dconf` 工具允许系统管理员通过直接安装 *defaults overrides* 来覆盖默认设置。例如，通过使用放置在 `keyfile` 目录的 `keyfile` 中的 `dconf` 覆盖来执行所有用户的默认背景，如 `/etc/dconf/db/local.d/`。要了解更多有关默认值和覆盖的信息，请参阅 [第 9.5 节 “配置自定义默认值”](#)。

锁定设置

`dconf` 系统现在允许锁定单个设置或整个设置子路径，以防止用户自定义。有关如何锁定设置的详情，请参考 [第 9.5.1 节 “锁定特定设置”](#)。

NFS 和 dconf

对通过 NFS 共享的主目录使用 `dconf` 工具需要额外的配置。有关此主题的详情，请查看 [第 9.7 节 “通过 NFS 存储用户设置”](#)。

获取更多信息

有关使用 `GSettings` 和 `dconf` 来配置用户设置的更多信息，请参阅 [第 9 章 使用 `GSettings` 和 `dconf` 配置桌面](#)。

第 4 章 POLICYKIT

PolicyKit 工具是一个框架，它提供了一个由特权程序（也称为 *机制*）使用的授权 API，为非特权程序（也称为 *主题*）提供服务。以下是有关 PolicyKit 或其系统名称 polkit 的更改的详情。

4.1. 策略配置

至于有关的新特性，授权规则现在在 *JavaScript .rules* 文件中定义。这意味着相同的文件被用来定义规则和管理员状态。在以前的版本中，此信息存储在两个不同的文件类型 - *If pkla* 和 *If conf*，它使用键/值对来定义额外的本地授权。

这些新的 *.rules* 文件存储在两个位置；而用于本地自定义的 polkit 规则存储在 `/etc/polkit-1/rules.d/` 目录中，第三方软件包存储在 `/usr/share/polkit-1/rules.d/` 中。

现有的 *.conf* 和 *.pkla* 配置文件已被保留，并与 *.rules* 文件一起存在。polkit 已为 Red Hat Enterprise Linux 7 升级，并考虑了兼容性问题。

规则中的优先逻辑已改变。polkitd 现在从 `/etc/polkit-1/rules.d` 和 `/usr/share/polkit-1/rules.d` 目录中以字典顺序读取 *.rules* 文件。如果两个文件的名称相同，则 `/etc` 中的文件会在 `/usr` 中的文件之前进行处理。另外，现有规则由 `/etc/polkit-1/rules.d/49-polkit-pkla-compatible.rules` 文件应用。因此，它们可以被 `/usr` 或 `/etc` 中的 *.rules* 文件覆盖，其名称以字典顺序排列在 `49-polkit-pkla-compatible` 之前的名称。确保您的旧规则不被覆盖的最简单方法是在所有其它 *.rules* 文件的名称的开头使用大于 49 的数字。

下面是一个 *.rules* 文件的示例。它会创建一个规则，允许在系统设备上为 *storage* 组挂载文件系统。该规则存储在 `/etc/polkit-1/rules.d/10-enable-mount.rules` 文件中：

例 4.1. 允许在系统设备上挂载文件系统

```
polkit.addRule(function(action, subject) {
  if (action.id == "org.freedesktop.udisks2.filesystem-mount-system" &&
      subject.isInGroup("storage")) {
    return polkit.Result.YES;
  }
});
```

如需更多信息，请参阅：

- [polkit\(8\)](#)- 有关 JavaScript 规则和优先级规则的手册页。

- **pkla-admin-identities(8)**和 **pkla-check-authorization(8)**- 分别是 `.conf` 和 `.pkla` 文件格式文档的手册页。

4.2. 默认策略

默认配置现在允许 `wheel` 组的成员使用其自己的密码对管理操作进行身份验证，而不是要求提供 `root` 密码。默认策略在 `/etc/polkit-1/rules.d/50-default.rules` 中定义。

在 GNOME 设置的用户面板中，您可以将帐户配置为 **Administrator**。当您在 GNOME Initial Setup 期间首次创建用户时，默认情况下，您将创建一个 **Administrator** 帐户(`wheel` 组的成员)。



注意

虽然 `sudo` 用户被允许使用其他用户的安全权限运行程序，但 **Administrator** 是 `wheel` 组的成员，为用户提供额外的特殊系统权限来执行受限命令。

4.3. 检查脚本的权限

`pkcheck` 工具（用来检查进程是否获得操作授权）现在支持 `--process` 选项指定的参数的新格式。通过避免出现竞争条件，运行 `pkcheck` 更安全。新格式类似如下：

```
$ pkcheck --process pid,start-time,uid
```



重要

不要将裸机 `pid` 或 `pid,start-time` 格式用于 `--process` 选项。调用 `pkcheck` 的所有脚本都需要使用新格式 `pid,start-time,uid` 来避免竞争条件。

如需更多信息，请参阅 `pkcheck(1)` man page。

4.4. 扩展 POLKIT 配置

对替换后端授权实现的支持已删除。可以通过编写调用外部程序的 JavaScript `.rules` 文件来实现类似级别的灵活性。

支持替换 `PolkitBackendActionLookup` 实现（用来将数据提供给身份验证对话框的接口）也已从 Red Hat Enterprise Linux 7 中的 `polkit` 中删除。

有关 `polkit` 的更多信息，请参阅 `polkit(8)` 手册页。

第 5 章 GDM

GDM 是 GNOME 显示管理器，它提供图形登录环境。从 GNOME 2 转换到 GNOME 3 后，只有通过 systemd 配置 GDM，因为它不再支持其他 init 系统。

gdm 软件包

gdm 软件包已替换了 xorg-x11-xdm，后者为 X Window 系统提供了传统的显示登录管理器。如前所述，gdm 软件包提供了图形登录屏幕，在启动、注销和用户切换后不久显示。

GDM 和 logind

GDM 现在使用 logind 来定义和跟踪用户。如需更多信息，请参阅 [第 2 章 logind](#)。系统管理员也可以在 GDM 自定义配置文件中手动设置自动登录：`/etc/gdm/custom.conf`。

custom.conf

GDM 配置现在在 `/etc/gdm/custom.conf` 中找到。但是，为了向后兼容，如果找到 `/etc/gdm/gdm.conf`，它将被使用而不是 `custom.conf`。升级时，红帽建议删除旧的 `gdm.conf` 文件，并将任何自定义配置迁移到 `custom.conf`。

获取更多信息

有关 GDM 的详情，请参考 [第 14.1 节 “GDM 是什么？”](#)。

有关配置和管理用户会话的详情，请参考 [第 14.3 节 “用户会话”](#)。

有关自定义登录屏幕外观的详情，请参考 [第 10.4 节 “自定义登录屏幕”](#)。

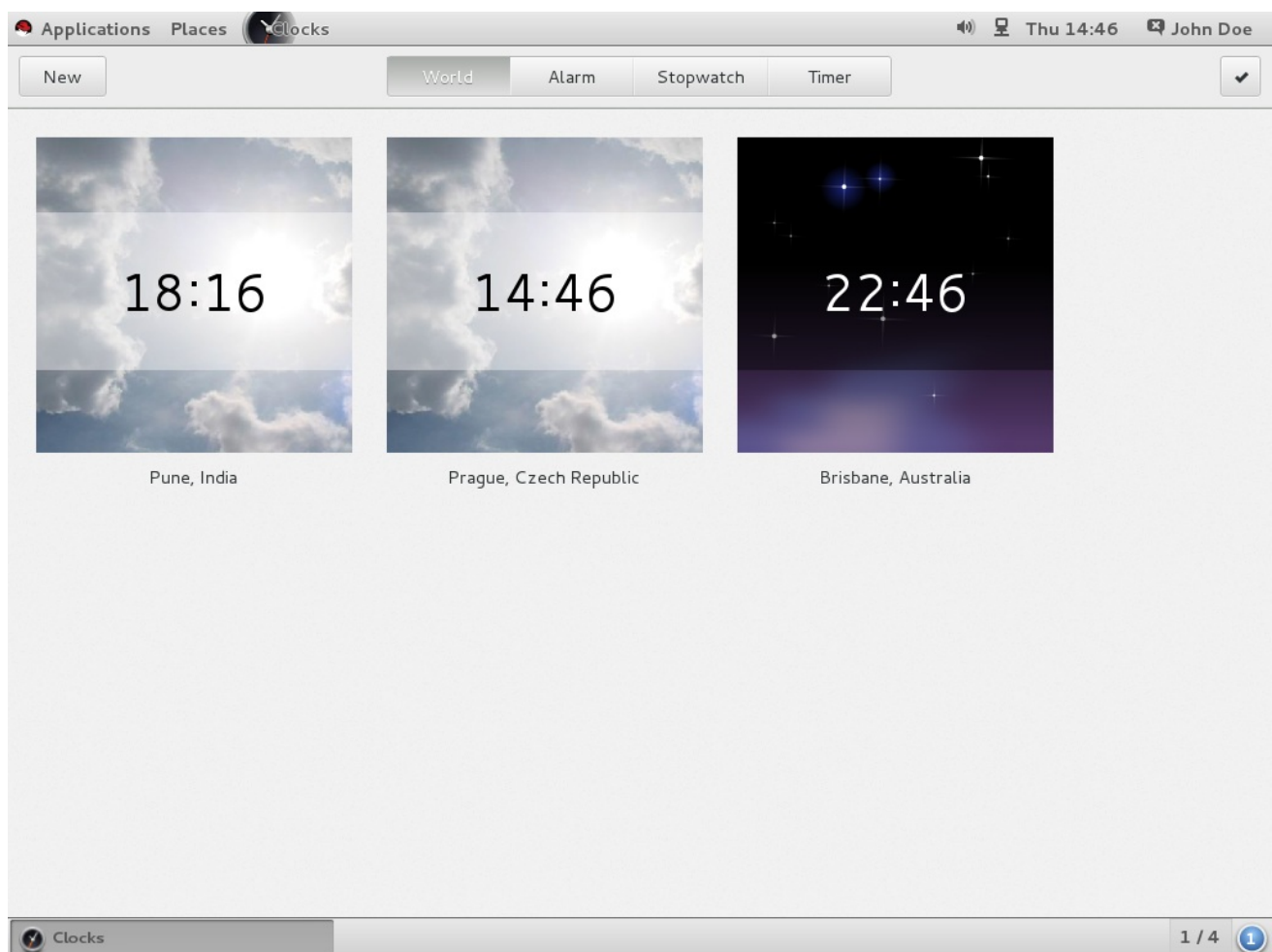
第 6 章 GNOME SHELL 扩展

Red Hat Enterprise Linux 7 中的 GNOME Shell 不支持小程序，后者用来自定义 Red Hat Enterprise Linux 5 和 6 中默认的 GNOME 2 界面。GNOME 3 使用 *GNOME Shell 扩展* 替换了小程序。扩展可以修改默认的 GNOME Shell 界面及其部分，如窗口管理和应用启动。

6.1. 替换时钟小程序

Red Hat Enterprise Linux 5 和 6 中的 GNOME 2 显示了 Clock 小程序，它提供对 GNOME 2 面板的日期、时间和日历的访问。在 Red Hat Enterprise Linux 7 中，小程序被 Clocks 应用程序替代，该应用程序由 `gnome-clocks` 软件包提供。用户可以通过单击 GNOME Shell 顶栏上的日历并选择 **Open Clocks** 来访问该应用程序。

图 6.1. 打开时钟



获取更多信息

有关 GNOME Shell 扩展是什么，以及如何配置和管理它们的详情，请参阅 [第 11.1 节 “GNOME Shell 扩展是什么？”](#)。

第 7 章 GNOME-SESSION

在 Red Hat Enterprise Linux 7 中也更新了 `gnome-session` 程序。它像以前一样启动 GNOME 桌面；尽管如此，它的一些组件也已发生了变化。

`gnome-session-properties`

`gnome-session-properties` 应用程序仍然是 `gnome-session` 软件包的一部分。但是，其功能已限制为管理单个用户的启动程序，并在退出登录时保存当前运行的应用程序。后一种功能已在 Red Hat Enterprise Linux 6 中保留下来。

命名的会话

Save now 按钮可以在特定时间保存会话并对其进行命名。保存的会话会在登录时恢复。当您在 `gnome-session-properties` 中点 **Automatically remember running applications when logging out** 时，保存的应用程序列表也会在登录时显示。

有了这个更新，也可以创建多个布局并重命名它们，或者为一个用户帐户选择多个用户会话。

获取更多信息

有关会话管理的详情，请参考 [第 14 章 会话管理](#)。

有关如何为所有用户管理启动（自动启动）应用程序的详情，请参考 [第 14.3.5 节 “为所有用户添加自动启动应用程序”](#)。

第 8 章 国际化

8.1. 输入法

Red Hat Enterprise Linux 7 中 GNOME 桌面的默认输入框架是 *IBus*（智能输入总线）。它与 GNOME 3 集成，并包含用于输入法选择的用户界面。

8.1.1. 配置和切换输入法

用户可以使用 GNOME 设置中的 Region & Language 面板来配置其输入法。有关使用输入法的更多信息，请参阅 GNOME 帮助。要访问它，请按 Super 键进入 Activities Overview，输入 help，然后按 Enter 键。

对于非 GNOME 会话，IBus 可以在 *ibus-setup* 工具中配置 XKB 布局和输入法，并使用快捷方式进行切换。

切换输入源的默认快捷方式是 Super+Space。在 Red Hat Enterprise Linux 6 中，快捷键是 Ctrl+Space。

8.1.2. IBus 的预测输入法

ibus-typing-booster 是 IBus 平台的预测输入法。它根据部分输入预测完整的词，允许更快速、更准确的文本输入。用户可以从建议的列表中选择所需的词。*ibus-typing-booster* 也可以使用 Hunspell 字典来为语言提供建议。

8.1.3. GNOME 桌面中的 iBus 替换了 im-chooser

由于 IBus 现在与 GNOME 桌面集成，除了使用非 IBus 输入法，*im-chooser* 已被弃用。

8.2. 更改的文件位置

在 Red Hat Enterprise Linux 7 中，对输入方法和字体配置文件以及目录的位置进行了以下更改：

- *.xinputrc* 文件已从用户的主目录移到 *~/.config/imsettings/* 目录。
- *.imsettings.log* 文件已从用户的主目录中移出，现在可以在 *~/.cache/imsettings/log* 中找

到。

- `~/.fonts.conf` 文件已弃用。我们鼓励用户将文件移到 `~/.config/fontconfig/` 目录。
- `~/.fonts.conf.d` 目录已被弃用。建议用户将目录移到 `~/.config/fontconfig/` 目录。
- `/etc/fonts/conf.avail/` 目录中所有禁用的 `fontconfig` 配置文件都已移到 `/usr/share/fontconfig/conf.avail/` 目录中。如果您有任何指向旧位置的本地符号链接，请记住更新它们。

部分 II. 配置和管理

*Red Hat Enterprise Linux 7 桌面迁移和管理指南*的第二部分描述并解释了管理 GNOME 桌面的不同方法。

第 9 章 使用 GSETTINGS 和 DCONF 配置桌面

9.1. 术语解释：GSETTINGS、GSETTINGS 和 DCONF

本节定义了几个容易混淆的术语。

dconf

dconf 是一个基于密钥的配置系统，用于管理用户设置。它是 Red Hat Enterprise Linux 7 中使用的 GSettings 的后端。**dconf** 管理一系列不同的设置，包括 GDM、应用程序和代理设置。

dconf

dconf 命令行工具用于从和写入 **dconf** 数据库的单个值或整个目录。

GSettings

GSettings 是应用程序设置的高级 API，即 **dconf** 的前端。

gsettings

gsettings 命令行工具用于查看和更改用户设置。

9.2. 用户和系统设置

dconf 允许系统管理员和用户配置进行多个级别的控制。

- 管理员可以定义适用于所有用户的默认设置。
- 用户可以使用自己的设置来覆盖默认值。
- 另外，管理员也可以锁定设置，以防止用户覆盖它们。如需更多信息，请参阅 [第 9.5.1 节“锁定特定设置”](#)。

9.3. 浏览桌面应用程序的 GSETTINGS 值

您可以使用两个工具来查看和编辑 **GSettings** 值：

- **dconf-editor GUI 工具。**
- **gsettings 命令行工具。**



注意

默认情况下，系统上可能没有安装 **dconf-editor**。要安装它，请运行以下命令：

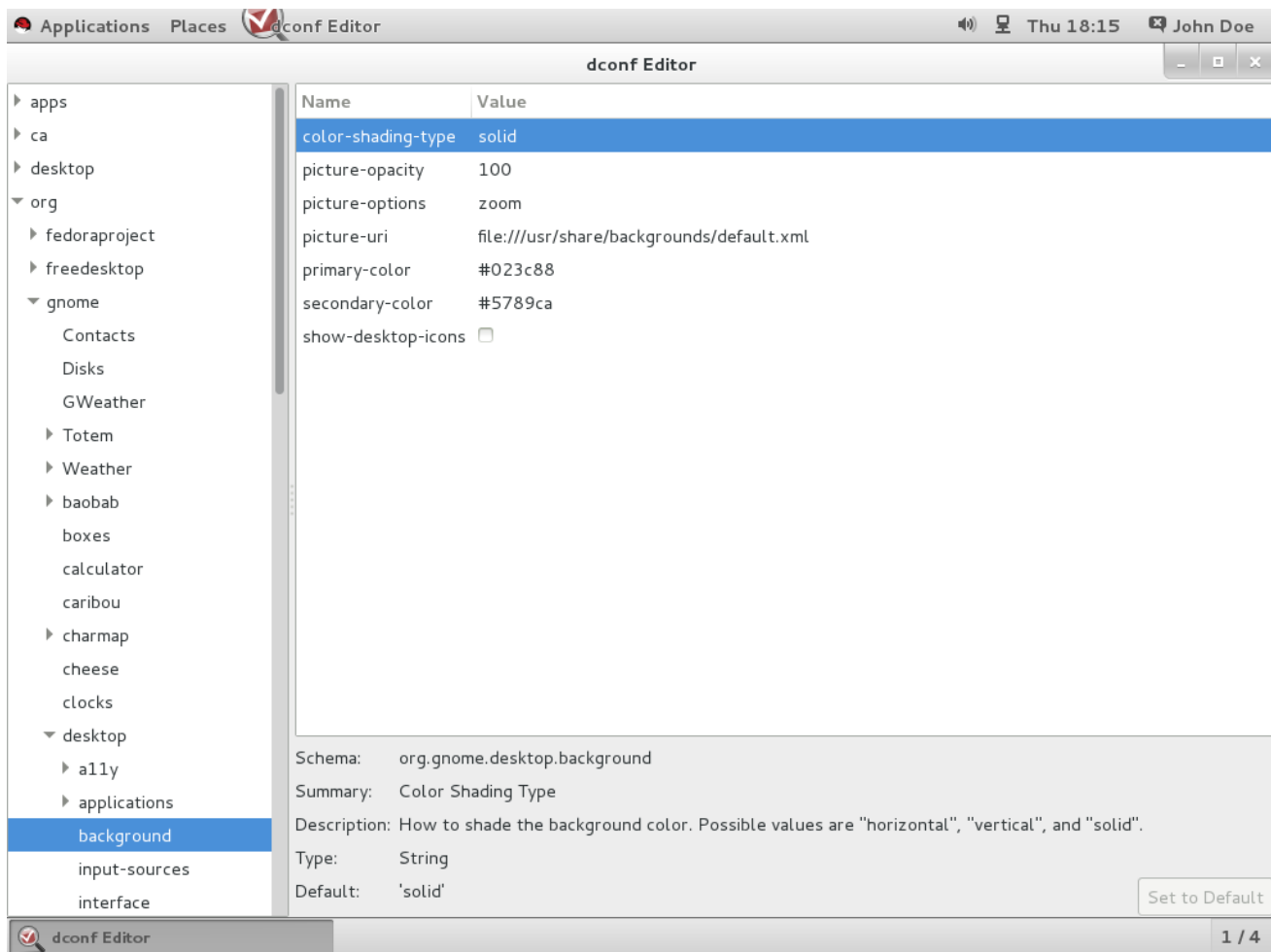
```
# yum install dconf-editor
```

dconf-editor 和 **gsettings** 工具可让您浏览和更改系统和应用程序首选项的选项。您甚至可以使用它们来更改图形用户界面中不存在的首选项。

dconf-editor 为浏览和编辑设置提供了一个 GUI。它在树视图中显示设置的层次结构，还显示有关每个设置的其他信息，包括描述、类型和默认值。**GSettings** 可用于显示和设置 **dconf** 值。它还包括命令和设置的 **Bash** 补全。**GSettings** 可用于 **shell** 脚本中自动配置。

请注意，**dconf-editor** 和 **gsettings** 工具都旨在浏览和修改当前用户的 **GSettings** 数据库。这意味着您应该始终以普通用户身份运行这些工具。

图 9.1. dconf-editor 显示 org.gnome.desktop.background GSettings 键



获取更多信息

有关 `dconf-editor` 工具的详情，请查看 `dconf-editor(1)` 手册页。

有关 `gsettings` 工具的更多信息，请参阅 `gsettings(1)` 手册页。

9.4. 什么是 DCONF 配置文件？

配置文件是 `dconf` 系统收集的系统和软件配置数据库的列表。`dconf` 配置文件允许您比较相同的系统来排除硬件或软件问题。

`dconf` 系统将其配置文件存储在文本文件中。`$DCONF_PROFILE` 环境变量可以指定 `/etc/dconf/profile/` 目录中文件的相对路径，或者指定绝对路径，比如在用户的主目录中。

`dconf` 配置文件中设置的键对将覆盖默认设置，除非您已设置的值有问题。

9.4.1. 选择一个 dconf 配置文件

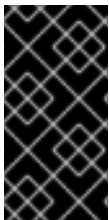
在启动时，dconf 会查询是否设置了变量 `$DCONF_PROFILE` 环境变量。如果设置，dconf 会尝试打开指定的配置文件，如果此步骤失败，则中止。

只要环境变量没有设置，dconf 就会尝试打开名为 `用户` 的配置文件。只要此步骤仍然失败，dconf 会回退到内部硬有线配置。

配置文件中的每一行都指定一个 dconf 数据库。第一行表示用于写入更改的数据库，而其余行则显示只读数据库。以下是存储在 `/etc/dconf/profile/user` 中的示例配置文件：

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

这个示例配置文件指定了三个数据库：`user` 是用户数据库的名称，通常在 `~/.config/dconf` 中找到，`local` 和 `site` 是系统数据库，位于 `/etc/dconf/db/` 中。



重要

会话的 dconf 配置文件是在登录时确定的，因此用户必须退出并登录，才能将新的 dconf 用户配置文件应用到其会话。

9.5. 配置自定义默认值

通过为 dconf 配置文件中的键提供默认设置，可以设置机器范围内的默认设置。这些默认值可以被用户覆盖。

要为键设置默认值，用户配置文件必须存在，键的值必须添加到 dconf 数据库中。

例 9.1. 设置默认背景

1.

如果不存在，请在 `/etc/dconf/profile/ user` 中创建用户配置文件：

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

其中 `local` 是 `dconf` 数据库的名称。

2.

在 `/etc/dconf/db/local.d/01-background` 中为本地数据库创建一个 `keyfile`，其中包含以下默认设置：

```
# 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'
```

在 `keyfile` 的默认设置中，使用以下 `GSettings` 键：

表 9.1. `org.gnome.desktop.background` 模式 `GSettings` 键

| 键名称 | 可能的值 | 描述 |
|------------------------------|--|--|
| <code>picture-options</code> | "none", "wallpaper", "centered", "scaled", "stretched", "zoom", "spanned" | 决定由 <code>wallpaper_filename</code> 设置的图像如何渲染。 |
| <code>picture-uri</code> | 带路径的文件名 | 用于后台镜像的 URI。请注意，后端只支持本地 (file://)URI。 |
| <code>primary-color</code> | 默认值：000000 | 绘制渐变或纯色时，左或上的颜色。 |
| <code>second-color</code> | 默认值：FFFFFF | 绘制渐变时右或底部的颜色，不用于纯色。 |

3.

根据您的偏好编辑 `keyfile`。如需更多信息，请参阅 [第 9.3 节“浏览桌面应用程序的 `GSettings` 值”](#)。

4.

更新系统数据库：

```
# dconf update
```



重要

创建或更改用户配置文件时，用户需要退出并再次登录，然后才能应用更改。

如果要避免创建用户配置文件，您可以使用 `dconf` 命令行工具从和向 `dconf` 数据库读取和写入单个值或整个目录。如需更多信息，请参阅 `dconf(1)` 手册页。

9.5.1. 锁定特定设置

`dconf` 中的锁定模式是防止用户更改特定设置的有用工具。

要锁定 GSettings 键，您需要在 `keyfile` 目录中创建一个 `locks` 子目录（例如，`/etc/dconf/db/local.d/locks/`）。此目录中的文件包含一个要锁定的键的列表，您可以向该目录添加任意数量的文件。



重要

如果您没有使用锁定强制系统设置，用户可轻松地使用他们的设置覆盖系统设置。用户所做的任何设置都优先于系统设置，除非有锁定强制系统设置。

以下示例演示了如何锁定默认壁纸的设置。对于您需要锁定的任何其他设置，请按照流程操作。

例 9.2. 锁定默认的壁纸

1. 按照第 10.5.1 节“自定义默认桌面背景”中的步骤设置默认的壁纸。
2. 创建一个名为 `/etc/dconf/db/local.d/locks/` 的新目录。
3. 在 `/etc/dconf/db/local.d/locks/00-default-wallpaper` 中创建一个新文件，其中包含以下内容，每行列出一个键：

```
# 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.

更新系统数据库：

```
# dconf update
```

9.6. GSETTINGS 键属性

您只能在 `dconf` 数据库中设置 `GSettings` 键一次。如果您在 `dconf` 数据库中的不同位置将相同的键设置为不同的值，则只有一个值将生效。换句话说，一个键会覆盖另一个键的设置。

在每个 `dconf` 系统数据库中，每个键只能有一个值。某些键的值是数组类型。对于这个值类型，可以将该值指定为用逗号分开的多个元素的列表。数组值的一个示例如下：

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

例 9.3. `org.gnome.desktop.input-sources.xkb-options` `GSettings` 键

设置 `org.gnome.desktop.input-sources.xkb-options` `GSettings` 键的工作方式如下。它只能被设置一次，如果您希望值中有两个元素，则需要要在同一个配置文件中指定它们。由于这个值是一个数组类型，因此它可以有多个元素：

```
[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']
```

9.7. 通过 NFS 存储用户设置

若要在 使用网络文件系统 (NFS) 主目录时 `dconf` 正常工作，必须使用 `dconf keyfile` 后端。

请注意，在使用 `dconf keyfile` 后端时，在系统中必须安装 `glib2-fam` 软件包。否则，在远程计算机上有关配置更改的通知将无法正常工作。

过程 9.1. 设置 `dconf Keyfile` 后端

1.

验证系统上是否安装了 `glib2-fam` 软件包。

- a. 系统必须订阅 **Optional** 频道。要了解系统如何订阅 **Optional** 频道，请阅读以下资源：[如何使用红帽订阅管理\(RHSM\)访问 Optional 和 Supplementary 频道以及 -devel 软件包？](#)

- b. 运行以下命令安装 **glib2-fam** 软件包：

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

2. 在每个客户端上创建或编辑 `/etc/dconf/profile/user` 文件。

3. 在此文件的最开始，添加以下行：

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

dconf keyfile 后端 仅在用户下次登录时生效。它将轮询 **keyfile** 来确定是否已进行了更新，因此设置可能不会立即更新。

第 10 章 默认外观

GNOME 3 桌面的外观和功能可由单个用户自定义，也可由系统管理员为所有用户自定义。系统管理员可以为所有用户提供自定义的默认桌面配置，甚至锁定这些功能以防止用户更改。

本章论述了如何自定义安装程序(Anaconda)、系统启动工具(GRUB、Plymouth)、登录屏幕、字体、键盘布局、屏幕屏蔽和桌面背景。

10.1. 创建 ANACONDA 品牌

如果您要在公司构建自己的自定义发行版，您可以更改 Red Hat Enterprise Linux 7 安装程序中使用的品牌图形和产品名称，Anaconda。

有关更改 Anaconda 图形和产品名称的更多信息，请参阅 Red Hat Enterprise Linux 7 的 [Anaconda 自定义指南](#)。

10.2. 引导加载程序屏幕

Red Hat Enterprise Linux 7 引导装载程序是 GRUB 2。您可以更改 GRUB 2 外观的几个部分。以下小节展示了如何更改发行版名称、菜单颜色和后台图像。

10.2.1. 发行版名称

默认情况下，GRUB 2 显示一个包含发行版名称的标题。您可以通过自定义 `/etc/default/grub` 文件中的 `GRUB_DISTRIBUTOR` 变量来更改标题。

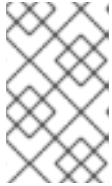
过程 10.1. 设置发行版名称

1. 以 root 用户身份，打开 `/etc/default/grub` 文件。
2. 使用 `GRUB_DISTRIBUTOR` 变量指定您自己的发行版名称。以下是 `/etc/default/grub` 文件的摘录。使用 `GRUB_DISTRIBUTOR` 变量更新第二行：

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=Our Corporate Distro V1.2
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
...
```

3. 以 `root` 身份运行以下命令，以确保更改生效：

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```



注意

每次进行更改后，您需要更新 `/etc/default/grub` 目录。

另外，您可以通过修改 `/etc/grub.d/40_custom` 纯文本文件或在 `/etc/grub.d/` 目录中添加另一个文件来更改 GRUB 2 屏幕中显示的颜色或字体。您可以从以下指令中选择：

- `set color_normal=foreground/background`
- `set color_highlight=foreground/background`
- `set menu_color_normal=foreground/background`
- `set menu_color_highlight=foreground/background`

有关这些变量的语义和可接受的值的更多信息，请参阅 `grub(8)` 手册页。

10.2.2. GRUB 2 背景

默认设置中没有配置 GRUB 背景。不过，可以向引导装载程序屏幕添加一张图像。

在为 GRUB 2 配置背景图像之前，必须为 `gfxterm` 图形终端安装 GRUB 2 Unicode 字体。默认情况下不提供任何字体，因此先决条件是将现有的 TTF 或 OTF 文件转换为 GRUB 2 使用的 PF2 格式：

通过运行 `grub2-mkfont` 命令，将现有的 TTF 或 OTF 文件转换为 PF2 格式。将输出文件命名为 `unicode.pf2`，以便使用 `grub2-mkconfig` 编写的默认配置。

例 10.1. 将 TTF 文件转换为 PF2 格式

本例演示了从 `LiberationSerif-Bold.ttf` 到 `.pf2` 格式的转换。新的 `.pf2` 格式文件名为 `unicode2`，因此不会与 `/grub2/fonts/` 目录中现有的 `unicode.pf2` 混淆。

```
grub2-mkfont --output=/boot/grub2/fonts/unicode2.pf2 --size=24
/usr/share/fonts/liberation/LiberationSerif-Bold.ttf
```

现在，您可以为 **GRUB 2** 配置背景图像。镜像文件可以位于 `boot/` 目录之外。

过程 10.2. 向引导加载程序屏幕添加图像

1. 以 `root` 用户身份打开 `/etc/default/grub` 文件。
2. 编辑文件中的以下配置变量：

```
GRUB_TERMINAL=gfxterm
GRUB_BACKGROUND=path_to_the_image.png
```

支持的格式有 **PNG**、**JPG**、**JPEG** 和 **TGA**。

3. 使用背景图像创建新的配置文件：

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 重启您的系统。

如有必要，图像将被扩展以适合屏幕。

10.3. PLYMOUTH

plymouth 是 Red Hat Enterprise Linux 7 的图形引导系统和日志记录器，它使用基于内核的模式设置(KMS)和直接渲染管理器(DRM)。Plymouth 还在引导过程中处理用户交互。

您可以通过从各种静态或者动画图形主体中选择一个来自定义引导屏幕的外观。可以根据现有的主题创建新主题。

10.3.1. 创建主题品牌

Plymouth 的每个主题都由一个主题数据文件和一个编译的 splash 插件模块 组成。数据文件有一个 `.plymouth` 扩展，并安装在 `/usr/share/plymouth/themes/` 目录中。

配置数据以键-值格式，在 `[Plymouth Theme]` 部分中指定。这个组的有效键为 `Name`、`Description` 和 `ModuleName`。虽然前两个键是自我解释器，但第三个指定了 Plymouth splash 插件模块的名称。不同的插件在引导时提供不同的动画，以及各种主题的底层实现：

例 10.2. `.plymouth` 文件样本

```
[Plymouth Theme]
Name=Charge
Description=A theme that features the shadowy hull of my logo charge up and finally burst
into full form.
ModuleName=two-step
```

过程 10.3. 更改 Plymouth 主题

1. 搜索现有的 Plymouth 主题，并选择最首选的一个。运行以下命令：

```
# yum search plymouth-theme
```

或运行 `plymouth-set-default-theme --list` 命令来查看已安装的主题。

您还可以在安装所有 `plymouth` 软件包时安装所有的主题。但是，您也会安装一些不必要的软件包。

```
# yum install plymouth\*
```

2. 使用 `plymouth-set-default-theme theme_name` 命令将新主题设置为默认。

例 10.3. 将 "spinfinity" 设为默认主题

您已经选择了 `spinfinity` 主题，因此您运行：

```
# plymouth-set-default-theme spinfinity
```

3.

编辑后重建 `initrd` 守护进程，否则您的主题不会在引导屏幕中显示。通过运行以下命令来做到这一点：

```
# dracut -f
```

10.3.2. 创建一个新的 Plymouth 主题

如果您不想从给定的主题列表中选择，则您可以创建您自己的主题。最简单的方法是复制一个现有主题，并对其进行修改。

过程 10.4. 从现有的主题创建您自己的主题

1.

复制 `plymouth/` 目录的整个内容。例如，作为模板目录，使用 Red Hat Enterprise Linux 7 的默认主题，`/usr/share/plymouth/themes/charge/charge.plymouth`，它使用两步 `splash` 插件（两步是两个阶段引导过程的常用引导负载功能，它以百计时到引导时间完成）：

```
[Plymouth Theme]
Name=Charge
Description=A theme that features the shadowy hull of my logo charge up and finally
burst into full form.
ModuleName=two-step

[two-step]
ImageDir=/usr/share/plymouth/themes/charge
HorizontalAlignment=.5
VerticalAlignment=.5
Transition=none
TransitionDuration=0.0
BackgroundStartColor=0x202020
BackgroundEndColor=0x202020
```

2.

使用新名称在 `/usr/share/plymouth/themes/newtheme/` 目录中保存 `charge.plymouth` 文

件，格式为：

```
newtheme.plymouth
```

3.

根据您的偏好更新 `/usr/share/plymouth/themes/newtheme / newtheme .plymouth` 文件中的设置，更改颜色、对齐或转换。

4.

运行以下命令，将 `newtheme` 设为默认：

```
# plymouth-set-default-theme newtheme
```

5.

运行以下命令，在更改主题后重建 `initrd` 守护进程：

```
# dracut -f
```

10.3.2.1. 使用品牌徽标

某些插件显示一个品牌徽标，来作为 `splash` 动画的一部分。如果您想在主题中添加自己的徽标，请按照以下简短流程操作。



重要

请记住，您的品牌徽标的图像格式必须是 `.png` 格式。

过程 10.5. 将您的徽标添加到主题中

1.

创建名为 `logo.png` 的图像文件，以及您的徽标。

2.

通过更新 `ImageDir` 键以指向您在第 1 步中创建的 `logo.png` 镜像文件，以编辑 `/usr/share/plymouth/themes/newtheme .plymouth` 文件：

```
ImageDir=/usr/share/plymouth/themes/newtheme
```

有关 **Plymouth** 的详情，请查看 **plymouth(8)** 手册页。

10.4. 自定义登录屏幕

GNOME 登录屏幕有几个可以被自定义的元素。这些更改只能由系统管理员执行，并会影响所有用户。这部分描述了如何自定义问候语文本、徽标、键盘布局 and 用户列表。

10.4.1. 添加一个问候语徽标

登录屏幕上的问候语徽标由 `org.gnome.login-screen.logo` **GSettings** 键控制。由于 **GDM** 使用自己的 `dconf` 配置文件，因此您可以通过更改该配置文件中的设置来添加问候语徽标。

有关 **GSettings** 和 `dconf` 的详情，请参考 [第 9 章 使用 GSettings 和 dconf 配置桌面](#)。

在为您的登录屏幕选择合适的徽标图片时，请考虑以下图片要求：

- 所有主要格式都支持：ANI, BPM, GIF, ICNS, ICO, JPEG, JPEG 2000, PCX, PNM, PBM, PGM, PPM, GTIFF, RAS, TGA, TIFF, XBM, WBMP, XPM, 以及 SVG。
- 图片的大小可以按比例扩展到 48 像素高。例如，如果您将徽标设为 1920x1080，则它将变为原始图的 85x48 缩略图。

过程 10.6. 向登录屏幕添加徽标

1. 创建或编辑 `/etc/dconf/profile/gdm` 中的 `gdm` 配置文件，其中包含以下行：

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

GDM 是 `dconf` 数据库的名称。

2. 在 `/etc/dconf/db/gdm.d/01-logo` 中为机器范围的设置创建一个 `gdm` 数据库：

```
[org/gnome/login-screen]
logo='/usr/share/pixmaps/logo/greeter-logo.png'
```

将 `/usr/share/pixmaps/logo/greeter-logo.png` 替换为您要用作问候语徽标的图像文件的路径。

3.

更新系统数据库：

```
# dconf update
```

下次登录时，屏幕将显示新的登录徽标。



注意

徽标没有更新会怎样？

确保您已以 `root` 用户身份运行 `dconf update` 命令来更新系统数据库。如果徽标没有更新，请尝试重启 GDM。如需更多信息，请参阅第 14.1.1 节“重启 GDM”。

10.4.2. 显示文本横幅

登录屏幕上的文本横幅由以下 GSettings 键控制的（有关 GSettings 的更多信息，请参阅第 9 章使用 GSettings 和 dconf 配置桌面）：

```
org.gnome.login-screen.banner-message-enable
```

启用显示横幅消息。

```
org.gnome.login-screen.banner-message-text
```

在登录窗口中显示文本横幅消息。

请注意，由于 GDM 使用自己的 dconf 配置文件，因此您可以通过更改该配置文件中的设置来配置文本横幅。

过程 10.7. 在登录屏幕上显示文本横幅

1. 创建或编辑 `/etc/dconf/profile/gdm` 中的 `gdm` 配置文件，其中包含以下行：

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

GDM 是 dconf 数据库的名称。

2. 在 `/etc/dconf/db/gdm.d/01-banner-message` 中为机器范围的设置创建一个 `gdm` 数据库：

```
[org/gnome/login-screen]
banner-message-enable=true
banner-message-text='Type the banner message here'
```



注意

横幅消息没有字符限制。GNOME Shell 会自动检测较长的文本，并进入两列模式。但是，横幅消息文本无法从外部文件读取。

3. 更新系统数据库：

```
# dconf update
```

当您从用户列表中选择了自己或开始在框中输入时，会显示横幅文本。下一次登录时，您会在输入密码时看到文本。

10.4.2.1. 如果横幅消息没有更新会怎样？

如果横幅消息没有显示，请确定您运行了 `dconf update` 命令。

如果横幅消息没有更新，请尝试重启 GDM。如需更多信息，请参阅 [第 14.1.1 节“重启 GDM”](#)。

10.4.3. 显示多键盘布局

您可以添加其它备选键盘布局，以使用户可以从登录屏幕上选择。

这对于通常使用与默认键盘布局不同的用户来说很有帮助，这些用户希望在登录屏幕时能够提供这些键盘布局。然而，选择仅在使用登录屏幕时适用。登录后，您自己的用户设置会接管。

过程 10.8. 更改系统键盘布局设置

1. 在 `/usr/share/X11/xkb/rules/base.lst` 文件的名为 `! layout` 部分找到所需语言布局的代码。
2. 使用 `localectl` 工具更改系统键盘布局设置，如下所示：

```
$ localectl set-x11-keymap layout
```

您可以用逗号分开的列表指定多个布局。例如，要将 `es` 设为默认布局，并将 `us` 设置为次要布局，请运行以下命令：

```
$ localectl set-x11-keymap es,us
```

3. 退出登录，看看是否可在登录屏幕的顶栏中找到已定义的布局。

请注意，您还可以使用 `localectl` 工具指定机器范围内的默认键盘模型、变体和选项。详情请查看 `localectl(1)` 手册页。

10.4.4. 禁用登录屏幕用户列表

您可以通过设置 `org.gnome.login-screen.disable-user-list` GSettings 键来禁用登录屏幕上显示的用户列表。

用户列表被禁用后，用户需要在提示下输入其用户名和密码才能登录。

过程 10.9. 设置 `org.gnome.login-screen.disable-user-list` 键

1. 创建或编辑 `/etc/dconf/profile/gdm` 中的 `gdm` 配置文件，其中包含以下行：

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

GDM 是 dconf 数据库的名称。

2. 在 `/etc/dconf/db/gdm.d/00-login-screen` 中为机器范围的设置创建一个 `gdm` 数据库：

```
[org/gnome/login-screen]
# Do not show the user list
disable-user-list=true
```

3. 通过更新 `dconf` 工具更新系统数据库：

```
# dconf update
```

10.5. 自定义桌面背景

使用 `dconf` 工具，您可以配置默认背景、添加额外的背景或添加多个背景。

如果系统的用户不允许更改这些设置的默认值，那么系统管理员需要使用锁目录锁定设置。否则，每个用户都可以自定义设置以适应他们自己的首选项。如需更多信息，请参阅 [第 9.5.1 节“锁定特定设置”](#)。

10.5.1. 自定义默认桌面背景

您可以通过在 `org.gnome.desktop.background` 模式中设置相关的 `GSettings` 键来配置默认的桌面背景及其外观。

有关 `GSettings` 的更多信息，请参阅 [第 9 章使用 GSettings 和 dconf 配置桌面](#)。

过程 10.10. 设置默认背景

1. 在 `/etc/dconf/db/local.d/00-background` 中为机器范围的设置创建一个 `local` 数据库：

```
# Specify the dconf path
[org/gnome/desktop/background]

# Specify the path to the desktop background image file
picture-uri='file:///usr/local/share/backgrounds/wallpaper.jpg'
# Specify one of the rendering options for the background image:
```



```
# 'none', 'wallpaper', 'centered', 'scaled', 'stretched', 'zoom', 'spanned'
picture-options='scaled'
# Specify the left or top color when drawing gradients or the solid color
primary-color='000000'
# Specify the right or bottom color when drawing gradients
secondary-color='FFFFFF'
```

2.

覆盖用户的设置，以防止用户在 `/etc/dconf/db/local.d/locks/background` 中更改它：

```
# 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
```

如需更多信息，请参阅 [第 9.5.1 节“锁定特定设置”](#)。

3.

更新系统数据库：

```
# dconf update
```

4.

在系统范围的设置生效前，用户必须退出并重新登录。

10.5.2. 添加额外的背景

您可以在系统上为用户提供额外的背景。

1.

使用 `org.gnome.desktop.background` 模式 创建一个指定额外背景的外观的 `filename.xml` 文件（对文件名没有要求）。以下是最常用的模式的列表：

表 10.1. `org.gnome.desktop.background` 模式 GSettings 键

| 键名称 | 可能的值 | 描述 |
|--------------------|---|--|
| picture-options | "none", "wallpaper", "centered", "scaled", "stretched", "zoom", "spanned" | 决定由 <code>wallpaper_filename</code> 设置的图像如何渲染。 |
| color-shading-type | "horizontal", "vertical" 和 "solid" | 如何渐变背景颜色。 |

| 键名称 | 可能的值 | 描述 |
|---------------|-------------|---------------------|
| primary-color | 默认值：#023c88 | 绘制渐变或纯色时，左或上的颜色。 |
| second-color | 默认值：#5789ca | 绘制渐变时右或底部的颜色，不用于纯色。 |

完整的选项范围可在 `dconf-editor` GUI 或 `gsettings` 命令行工具中找到。如需更多信息，请参阅第 9.3 节“浏览桌面应用程序的 GSettings 值”。

2.

将文件名.xml 文件存储在 `/usr/share/gnome-background-properties/` 目录中。

当用户在右上角点击其名称时，选择 **Settings**，然后在表的 **Personal** 部分中选择 **Background**，他们会看到可用的新背景。

查看示例，并看看 `org.gnome.desktop.background` GSettings 键实际上是如何实施的：

例 10.4. 额外的背景文件

```
<?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>
```

在一个配置文件中，您可以指定多个 `<wallpaper>` 元素，来添加更多的背景。

请参见以下示例，它显示了一个有两个 `<wallpaper>` 元素的 .xml 文件，添加了两个不同的背景：

例 10.5. 有两个壁纸元素的额外的背景文件

```
<?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>

```

10.5.3. 设置屏幕屏蔽

屏幕 Shield 是系统锁定时快速向下滑动的屏幕。它由 `org.gnome.desktop.screensaver.picture-uri` GSettings 键控制。由于 GDM 使用自己的 `dconf` 配置文件，因此您可以通过更改该配置文件中的设置来设置默认背景。

有关 GSettings 和 `dconf` 的详情，请参考第 9 章 [使用 GSettings 和 dconf 配置桌面](#)。

过程 10.11. 向屏幕屏蔽中添加徽标

1.

在 `/etc/dconf/db/gdm.d/01-screensaver` 中为机器范围的设置创建一个 `gdm` 数据库：

```

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

```

将 `/opt/corp/background.jpg` 替换为您要用作屏幕屏蔽的图像文件的路径。

支持的格式有 PNG、JPG、JPEG 和 TGA。如果需要，可以扩展图像，以适合屏幕。

2.

更新系统数据库：

■

```
# dconf update
```

3. 在系统范围的设置生效前，您必须退出登录。

下次锁定屏幕时，新屏幕屏蔽将在背景中显示。在前景中，将会显示时间、日期以及现在是星期几。

10.5.3.1. 如果屏幕屏蔽没有更新会怎样？

确保您已以 `root` 用户身份运行 `dconf update` 命令来更新系统数据库。

如果背景没有更新，请尝试重启 `GDM`。如需更多信息，请参阅 [第 14.1.1 节“重启 GDM”](#)。

10.6. 配置字体

Red Hat Enterprise Linux 7 使用 `fontconfig` 工具进行字体管理和自定义。`fontconfig` 简化了字体管理，并提供显示功能，如反别名。本节描述了以下字体管理任务：

- 添加新字体（为一个用户及为所有用户）
- 指定用来替换缺失字体的字体
- 配置字体别名
- 按语言定义字体首选项
- 自定义字体属性

要编译系统上可用的字体列表，`fontconfig` 会搜索 `/etc/fonts/fonts.conf` 配置文件中默认列出的目录。

要列出系统上安装的 `fontconfig` 已知的所有字体，您可以使用 `fc-list` 命令：

```
$ fc-list : file
```

有关 `fc-list` 的更多信息，请参阅 `fc-list(1)` man page。

有关 `fontconfig` 及其配置的详情，请查看 `fonts-conf(5)` 手册页。

10.6.1. 为所有用户添加额外的字体

您可以安装额外的字体，以提供给使用 `fontconfig` 进行字体处理的应用程序中的用户。

过程 10.12. 安装额外的字体

1. 以 `root` 用户身份，创建 `/usr/local/share/fonts/` 目录，并将字体复制到其中。

为每个要安装的字体集创建一个子目录，因为某些字体有多个用于粗体、斜体等的文件。

2. 运行以下命令，确保字体缓存已更新：

```
# fc-cache /usr/local/share/fonts/
```

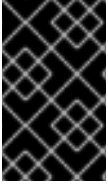
重要

`fontconfig` 将检测新的字体，并使其可用。与用户会话不同，在允许您使用新字体之前，一些应用程序可能需要重新启动。

10.6.2. 使用替代目录为所有用户添加额外的字体

或者，如果该目录列在 `/etc/fonts/fonts.conf` 文件中，您也可以将字体安装到另一个系统目录中，而不是 `/usr/local/share/fonts/`。

如果该目录没有列在 `/etc/fonts/fonts.conf` 文件中，则需要 `/etc/fonts/local.conf` 中创建自己的机器范围的配置文件。

**重要**

不要编辑 `/etc/fonts/fonts.conf` 文件，因为它会在 `fontconfig` 库被更新时被替换。

过程 10.13. 安装额外的字体

1. 以 `root` 用户身份，创建自己的机器范围的 `/etc/fonts/local.conf` 配置文件，包括带有字体的目录。

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
  <dir>/PATH/TO/YOUR/DIRECTORY_NAME</dir>
</fontconfig>
```

2. 确保目录的权限设置为 `755`。

```
# chmod -R 755 /PATH/TO/YOUR/DIRECTORY_NAME
```

3. 使用 `fc-cache` 命令更新字体缓存时，指定目录名称。

```
# fc-cache DIRECTORY_NAME
```

10.6.3. 为单个用户添加额外的字体

您可以安装一个额外的字体，该字体可供您系统上的特定用户使用 `fontconfig` 进行字体处理。

过程 10.14. 安装额外的字体

1. 将字体复制到 `~/.local/share/fonts/` 目录以安装它。
2. 运行以下命令，确保字体缓存已更新：

```
$ fc-cache ~/.local/share/fonts
```



重要

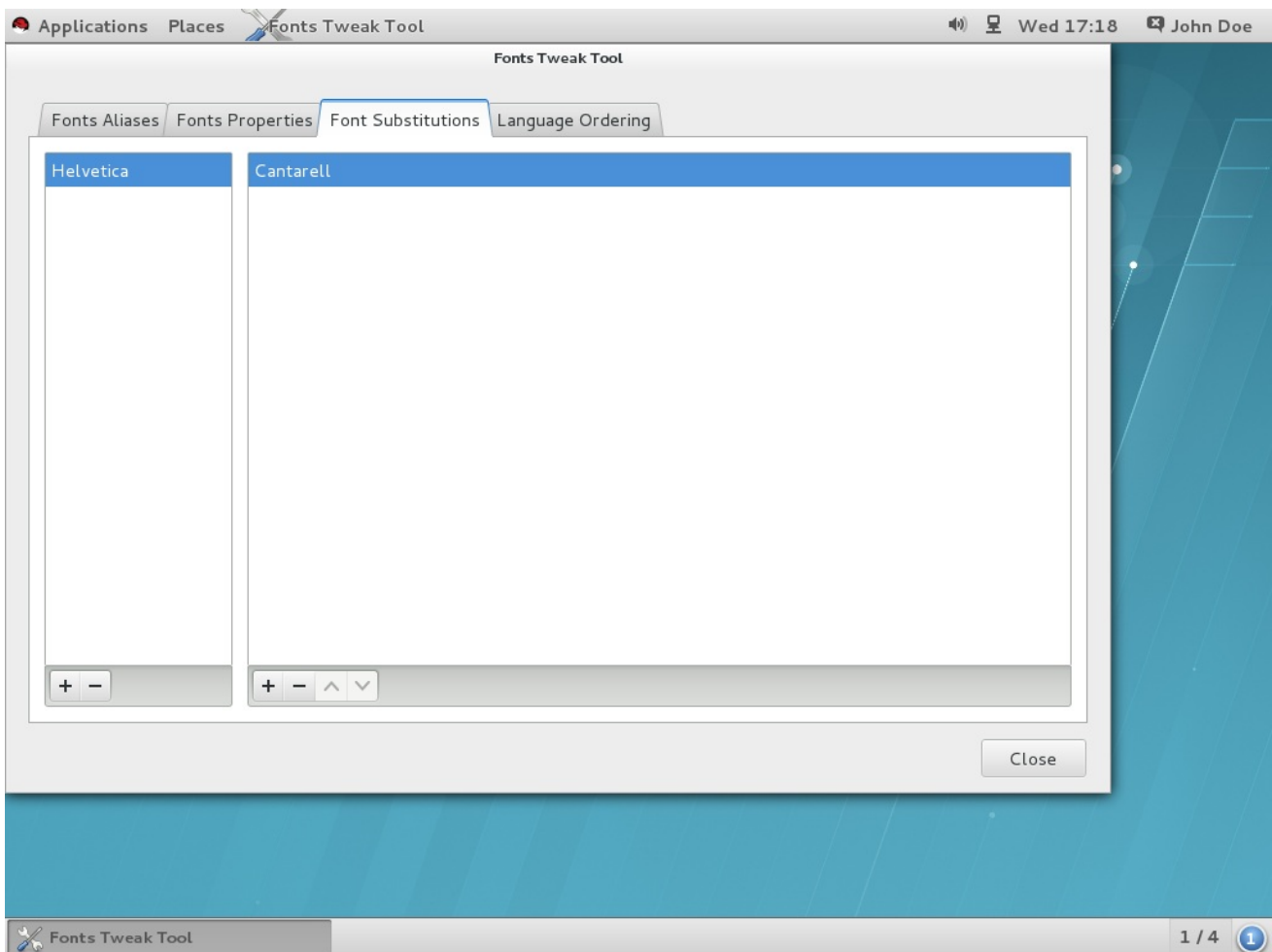
fontconfig 将检测新的字体，并使其可用。您可能需要重启正在运行的应用程序来查看更改是否有效。用户会话不需要重启。

10.6.4. 替换字体

当应用程序请求系统上不可用的字体时，**fontconfig** 会读取 `/etc/fonts/fonts.conf` 配置文件，以确定用于替换请求的字体的最相似的可用字体。如果单个字符在请求的字体中不存在，则它们也可以被替换。

要为特定的字体配置字体替换，您可以使用 **字体调整工具**。请注意，工具只能按用户配置来使用。

图 10.1. 使用字体调整工具替换字体



默认情况下，系统上可能没有安装 **字体调整工具**，要安装它，请运行以下命令：

```
# yum install fonts-tweak-tool
```

过程 10.15. 替换字体

1. 按 **Super** 键进入活动概览，输入 **Fonts Tweak Tool**，然后按 **Enter** 键启动字体调整工具。
2. 点 **Font Substitutions** 选项卡。
3. 点击左窗格左下角的 **+** 按钮，选择或输入您要替换的字体的名称，然后点击 **Add**。
4. 点击右侧窗格左下角的 **+** 按钮，选择要用于替换第一个字体的字体名称，然后点击 **Add**。
5. 点 **Close**。

现在，您已使用您喜欢的字体替换了旧字体，它现在可供您使用。

10.6.5. 配置字体别名

对于每个区域设置，字体调整工具允许单个用户配置不同的字体别名：

- **Sans Serif,**
- **Serif,**
- **Monospace,**
- **Cursive 和**
- **Fantasy.**

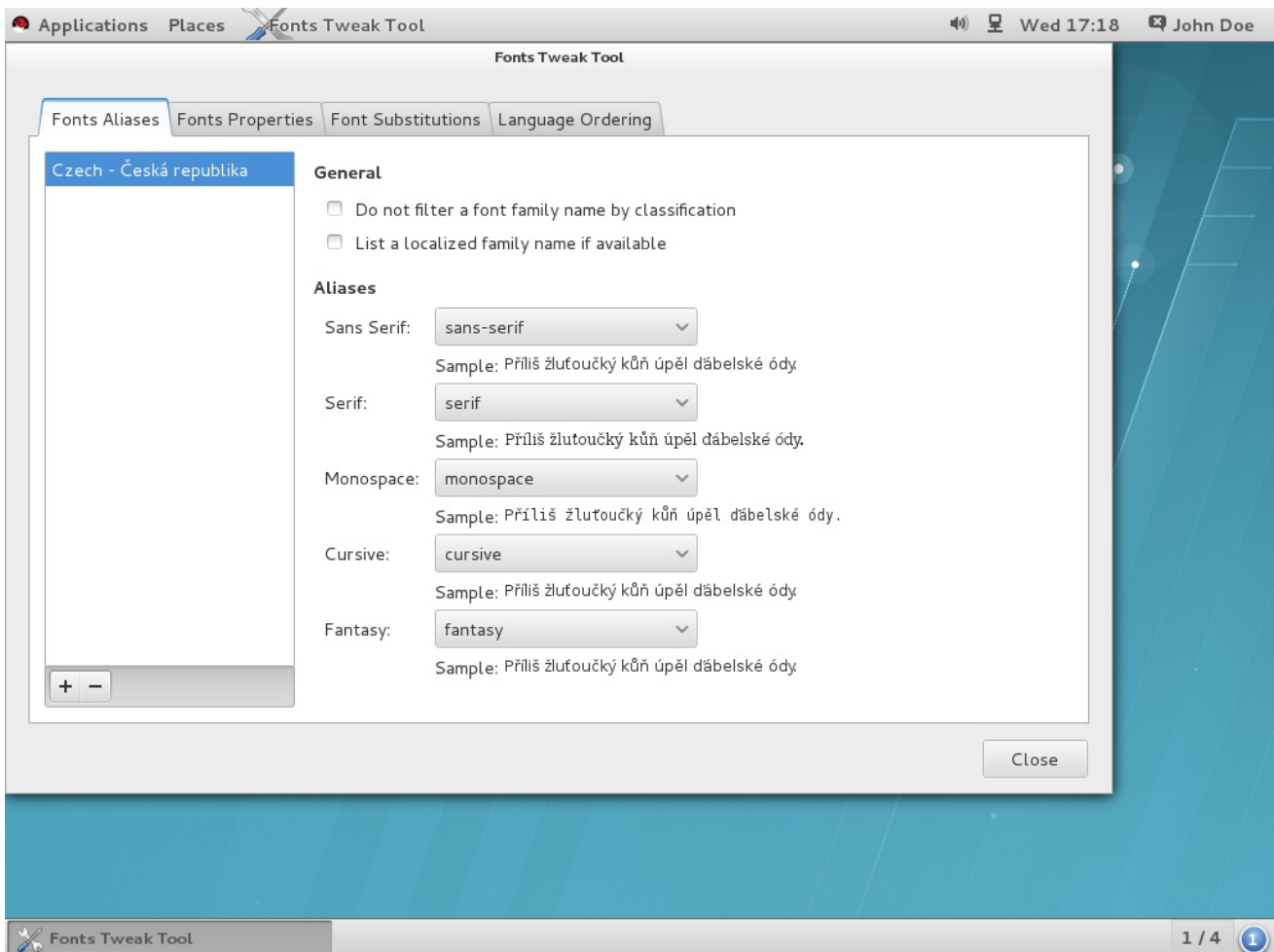
这些别名用来代表常见的字体类型，如 **serif** 和 **monospace** 类型。然后，应用程序和用户可以引用这些别名，而不必指定系统上安装的特定字体。

用户可以通过为这些别名的每一个选择一个自定义字体来覆盖系统默认的字體。

默认情况下，系统上可能没有安装字体调整工具，要安装它，请运行以下命令：

```
# yum install fonts-tweak-tool
```

图 10.2. 使用字体调整工具配置字体别名



过程 10.16. 配置字体别名

1. 按 **Super** 键进入活动概览，输入 **Fonts Tweak Tool**，然后按 **Enter** 键启动字体调整工具。
2. 点 **Font Aliases** 选项卡。
3. 点击左窗格左下角的 **+** 按钮，选择或键入您要为其配置字体别名的区域名称，然后点击 **Add**。

要配置默认的字体别名，而无论使用的区域是什么，请从区域列表中选择 **Default**。

4. 在右侧窗格中，找到您要覆盖系统默认值的字体别名，然后从下拉菜单中选择自定义字体。
5. 点 **Close**。

现在，您已覆盖了系统默认的别名，并选择了一个新的自定义字体。

10.6.6. 多语言顺序

字体调整工具 允许为用户界面配置了多个语言的用户更改语言在应用程序中显示的顺序。此功能对于使用拉丁和非拉丁字体，以及不想使用非拉丁字体来显示拉丁字符的用户特别有用。

例如，如果您将日语和英语配置为您的语言，并且希望避免使用日文非拉丁字体显示英文拉丁字符，您可以将英语配置为主要语言，并将日语配置为次要语言。然后，使用拉丁字体来显示英文字符，而非拉丁字体将仅用于显示日文字符。

默认情况下，系统上可能没有安装 字体调整工具，要安装它，请运行以下命令：

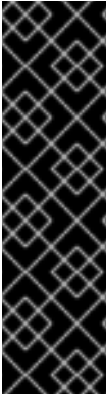
```
# yum install fonts-tweak-tool
```

过程 10.17. 配置多语言

1. 按 **Super** 键进入 活动概览，输入 **Fonts Tweak Tool**，然后按 **Enter** 键启动 字体调整工具。
2. 点 **Language Ordering** 选项卡。
3. 单击窗口左下角的 **+** 按钮，选择或输入您要配置为主语言的语言名称，然后单击 **Add**。
4. 要添加其他语言，请单击窗口左下角的 **+** 按钮，选择要配置为次要语言的语言名称，然后单击 **Add**。

重复此步骤来添加更多的语言。

5. 点 **Close**。



重要

对于用户界面，您现在已按照应用程序中显示的语言的顺序设置了您的首选项。

当配置了多种语言时，一些应用程序（如 `xterm` 和其他 `Xft` 应用程序）可能无法正确显示用户语言的所有字符。这是因为在这些应用程序中或在这些应用程序使用的渲染库中缺少对回退字体的支持。

10.6.7. 配置字体属性

字体调整工具 允许用户更改各种字体属性，允许按用户进行精细的字体配置。

默认情况下，系统上可能没有安装 字体调整工具，要安装它，请运行以下命令：

```
# yum install fonts-tweak-tool
```

过程 10.18. 更改字体属性

1. 按 **Super** 键进入 活动概览，输入 **Fonts Tweak Tool**，然后按 **Enter** 键启动 字体调整工具。
2. 点 **Fonts Properties** 选项卡。
3. 点击窗口左下角的 **+** 按钮，选择或输入您要更改属性的字体名称，然后点击 **Add**。

重复此步骤来添加更多字体。

4. 根据需要更改字体属性。

5.

点 **Close**。

根据添加的字体，用户可以在 **Fonts Properties** 选项卡中配置的一些字体属性包括：

如果可用，请使用嵌入的位图字体。

这对于喜欢位图字体而不是轮廓字体的用户很有用。要使用嵌入的位图字体，请添加合适的字体，然后单击 **Use embedded bitmap font if any**。

使用 **JIS X 2013:2004** 符号。

要使用 **JIS X 2013:2004** 标准中的日语，而不是来自 **JIS X 2013:2000** 或更早的版本，请添加支持 **JIS X 2013:2004** 的字体，然后单击 **Features** 列表中的 **jp04**。

第 11 章 GNOME SHELL 扩展

本章介绍 GNOME Shell 扩展的系统范围的配置。您将了解如何查看扩展、如何启用扩展、如何锁定启用的扩展的列表，或如何为系统的用户设置几个强制扩展。

在配置 GNOME Shell 扩展时，您将使用 `dconf`，设置以下两个 GSettings 键：

- `org.gnome.shell.enabled-extensions`
- `org.gnome.shell.development-tools`

有关 `dconf` 和 GSettings 的详情，请参考 [第 9 章 使用 GSettings 和 dconf 配置桌面](#)。

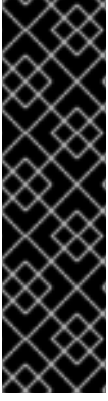
11.1. GNOME SHELL 扩展是什么？

GNOME Shell 扩展允许自定义默认的 GNOME Shell 界面及其部分，如窗口管理和应用程序启动。

每个 GNOME Shell 扩展都由一个唯一的标识符 `uuid` 来标识。`uuid` 也用于安装扩展的目录的名称。您可以在 `~/.local/share/gnome-shell/extensions/uuid`、`/usr/share/gnome-shell/extensions/uuid` 中按用户安装扩展。

`uuid` 标识符为全局唯一。选择它时，请记住 `uuid` 必须具有以下属性才能够防止某些攻击：

- 您的 `uuid` 不得包含 Unicode 字符。
- 您的 `uuid` 不得包含 `gnome.org` 结尾，它不能看起来与 GNOME 项目有关联。
- 您的 `uuid` 必须仅包含字母数字字符、句点(.)、at 符号(@)和下划线(_)。



重要

在 Red Hat Enterprise Linux 上部署第三方 GNOME Shell 扩展前，请确保阅读以下文档来了解红帽对第三方软件的支持策略：

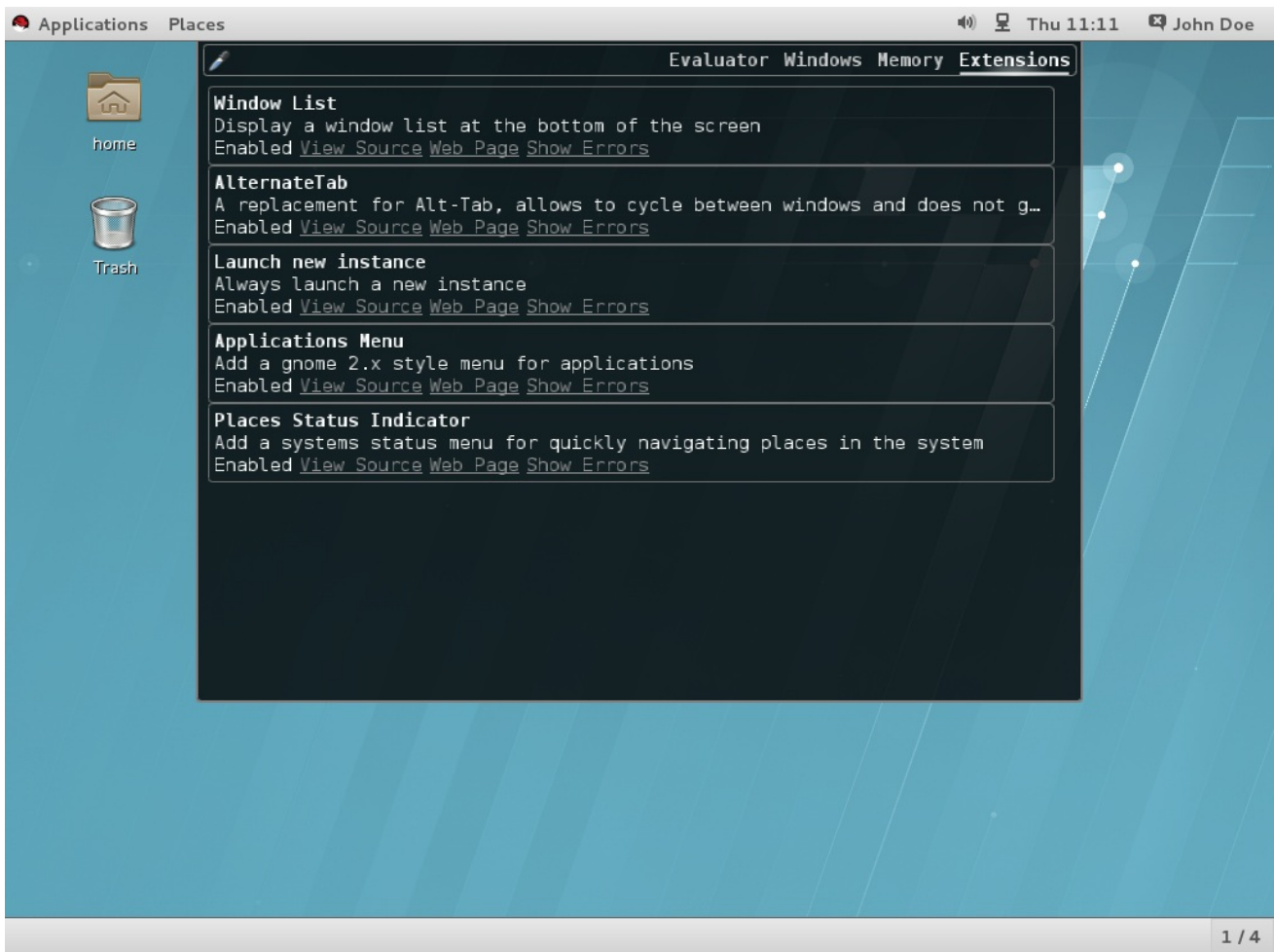
- [Red Hat 全球支持服务如何处理第三方软件、驱动程序和/或未经认证的硬件/hypervisor？](#)

要查看已安装的扩展，您可以使用 Looking Glass、GNOME Shell 的集成调试器和检查器工具。

过程 11.1. 查看已安装的扩展

1. 按 **Alt+F2**。
2. 输入 **lg**，然后按 **Enter** 键打开 Looking Glass。
3. 在 Looking Glass 的顶部栏中，单击 **Extensions** 以打开已安装的扩展列表。

图 11.1. 使用 Looking Glas 查看已安装的扩展



11.2. 启用机器范围的扩展

要使扩展可供系统上的所有用户使用，请在 `/usr/share/gnome-shell/extensions` 目录中安装它们。

为了设置默认启用的扩展，您需要设置 `org.gnome.shell.enabled-extensions` 键。但是，目前无法为已经登录的用户启用额外的扩展。这不适用于已安装并启用了他们自己的 GNOME 扩展的现有用户。

过程 11.2. 启用机器范围的扩展

1.

在 `/etc/dconf/db/local.d/00-extensions` 中为机器范围的设置创建一个 local 数据库文件：

```
[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']
```

`enabled-extensions` 键使用扩展的 `uuid` (`myextension1@myname.example.com` 和 `myextension2@myname.example.com`) 指定启用的扩展。

2.

更新系统数据库：

```
# dconf update
```

3.

在系统范围的设置生效前，用户必须退出并重新登录。

11.3. 锁定启用的扩展

在 GNOME Shell 中，您可以通过锁定 `org.gnome.shell.enabled-extensions` 和 `org.gnome.shell.development-tools` 键来防止用户启用或禁用扩展。

锁定 `org.gnome.shell.development-tools` 键可确保用户无法使用 GNOME Shell 的集成调试器和检查器工具(Looking Glass)来禁用任何强制扩展。

过程 11.3. 锁定启用的扩展

1.

在 `/etc/dconf/db/local.d/00-extensions` 中为机器范围的设置创建一个 local 数据库文件：

```
[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']
# Disable access to Looking Glass
development-tools=false
```

`enabled-extensions` 键使用扩展的 uuid (`myextension1@myname.example.com` 和 `myextension2@myname.example.com`) 指定启用的扩展。

`development-tools` 键被设置为 `false`，以禁用对 Looking Glass 的访问。

2.

覆盖用户的设置，并阻止用户在 `/etc/dconf/db/local.d/locks/extensions` 中更改它：

```
# Lock the list of mandatory extensions and access to Looking Glass
/org/gnome/shell/enabled-extensions
/org/gnome/shell/development-tools
```


3. **更新系统数据库：**

```
# dconf update
```

4. **在系统范围的设置生效前，用户必须退出并重新登录。**

锁定 `org.gnome.shell.enabled-extensions` 和 `org.gnome.shell.development-tools` 键后，在 `~/.local/share/gnome-shell/extensions` 或 `/usr/share/gnome-shell/extensions` 中安装的任何扩展都不会被 GNOME Shell 加载，从而防止用户使用它们。`org.gnome.shell.enabled-extensions`

11.4. 设置强制扩展

在 GNOME Shell 中，您可以提供用户必须使用的扩展集。要做到这一点，在 `/usr/share/gnome-shell/extensions` 目录中安装扩展，然后锁定 `org.gnome.shell.enabled-extensions` 和 `org.gnome.shell.development-tools` 键。

锁定 `org.gnome.shell.development-tools` 键可确保用户无法使用 GNOME Shell 的集成调试器和检查器工具(Looking Glass)来禁用任何强制扩展。

过程 11.4. 设置强制扩展

1. 在 `/etc/dconf/db/local.d/00-extensions-mandatory` 中为机器范围的设置创建一个 local 数据库文件：

```
[org/gnome/shell]
# List all mandatory extensions
enabled-extensions=['myextension1@myname.example.com',
'myextension2@myname.example.com']
# Disable access to Looking Glass
development-tools=false
```

`enabled-extensions` 键使用扩展的 uuid (`myextension1@myname.example.com` 和 `myextension2@myname.example.com`) 指定启用的扩展。

`development-tools` 键被设置为 `false`，以禁用对 Looking Glass 的访问。

2. 覆盖用户的设置，并阻止用户在 `/etc/dconf/db/local.d/locks/extensions-mandatory` 中更改它：

```
# Lock the list of mandatory extensions and access to Looking Glass  
/org/gnome/shell/enabled-extensions  
/org/gnome/shell/development-tools
```

3.

更新系统数据库：

```
# dconf update
```

4.

在系统范围的设置生效前，用户必须退出并重新登录。

第 12 章 集成应用程序

当将应用程序与 GNOME 桌面集成时，系统管理员通常会执行与自定义 Applications 菜单结构和 MIME 类型相关的任务，例如：

- 为应用程序添加或修改菜单项，或通过创建或修改子菜单来自定义 Applications 菜单结构。有关菜单自定义的更多信息，请参阅第 12.1 节“定制菜单”。
- 自定义活动概览中 GNOME Shell dash 上看到的默认喜欢的应用程序。有关如何进行此操作的更多信息，请参阅第 12.2 节“自定义默认喜欢的应用程序”。
- 为应用程序添加或修改 MIME 类型，并将应用程序与特定的 MIME 类型关联。有关配置 MIME 类型的更多信息，请参阅第 12.3 节“配置文件关联”。

12.1. 定制菜单

GNOME 菜单系统基于 freedesktop.org 桌面菜单规范，由三组主要的配置和数据文件组成：

桌面条目文件(.desktop)

.desktop 文件提供有关每个菜单项的数据，如其名称、要运行的命令及其图标。.desktop 条目文件还指定菜单层次结构中菜单项的位置，以及在活动概览中用于应用程序搜索的关键字。

系统 .desktop 文件位于 /usr/share/applications/ 目录中。特定于用户的 .desktop 文件位于 ~/.local/share/applications/ 目录中。

以下是名为 ~/.local/share/applications/myapplication1.desktop 的 .desktop 文件示例：

```
[Desktop Entry]
Type=Application
Name=My Application 1
Icon=myapplication1
Exec=myapplication1
Categories=Network;WebBrowser;
MimeType=application/x-newtype
```

上面的文件指定了应用的名称(My Application 1)、应用的图标(myapplication1)，以及运行应用程序的命令(myapplication1)。它还将应用程序放在指定类别(Network;WebBrowser;)中，并将应用

程序与 `application/x-newtype MIME` 类型相关联。

菜单定义文件(.menu)

`.menu` 文件是 XML 配置文件，用于指定菜单和菜单项的顺序、层次结构和合并。

机器范围的 `.menu` 文件位于 `/etc/xdg/menus/` 目录中。特定于用户的 `.menu` 文件位于 `~/.config/menus/` 目录中，可用于覆盖机器范围的 `.menu` 文件中指定的值。

特别是，`/etc/xdg/menus/applications.menu` 文件包含 `Applications` 菜单布局的定义。

目录条目文件(.directory)

`.directory` 文件提供有关每个菜单的数据，如其名称，并位于 `/usr/share/desktop-directories/` 中。

获取更多信息

有关桌面条目文件的更多信息，请参阅位于 `freedesktop.org` 网站的桌面条目规范：

- <http://freedesktop.org/wiki/Specifications/desktop-entry-spec>

有关描述 GNOME 菜单系统的实现的详细信息，请参阅位于 `freedesktop.org` 网站的桌面菜单规范：

- <http://standards.freedesktop.org/menu-spec/latest>

12.1.1. 为单个用户删除菜单项

给定用户的 `Applications` 菜单自定义默认存储在 `~/.config/menus/gnome-applications.menu` 定义文件中。可以通过设置 `$XDG_DATA_HOME` 环境变量来覆盖该文件的位置。

要覆盖 `Applications` 菜单的默认值，您首先需要创建一个 `gnome-applications.menu` 文件。请注意，从 `Applications` 菜单中删除项目，其子菜单也会将其从活动概览中的 `Applications` 视图中删除，从而防止用户在活动概览中搜索该项目。

过程 12.1. 示例：从附件子菜单中删除计算器菜单项

1. 查询 `/usr/share/applications/` 目录的内容，并确定与您要删除的菜单项对应的 `.desktop` 文件：

```
$ grep -r "Name=Calculator" /usr/share/applications/
/usr/share/applications/gcalctool.desktop:Name=Calculator
```

如上所示，计算器菜单项对应于 `/usr/share/applications/gcalctool.desktop` 文件。

2. 创建一个 `~/.config/menus/gnome-applications.menu` 文件：

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
"http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd">

<Menu>
  <Name>Applications</Name>
  <MergeFile type="parent">/etc/xdg/menus/gnome-applications.menu</MergeFile>

  <!-- Removes the Calculator from the Accessories submenu -->
  <Menu>
    <Name>Accessories</Name>
    <Exclude>
      <Filename>gcalctool.desktop</Filename>
    </Exclude>
  </Menu>
  <!-- END of Calculator removal content -->

</Menu>
```

如上所示，该文件包含一个 `<Menu>` 部分，用于指定子菜单的名称(附件)、`.desktop` 文件的名称(`gcalctool.desktop`)，并包含 `<Exclude>` 元素。

12.1.2. 删除所有用户的菜单项

默认情况下，所有用户的 `Applications` 菜单自定义都存储在 `/etc/xdg/menus/applications.menu` 定义文件中。可以通过设置 `$XDG_CONFIG_DIRS` 环境变量来覆盖该文件的位置。

要覆盖 `Applications` 菜单的默认值，您需要编辑 `.menu` 文件。请注意，从 `Applications` 菜单中删除项目，其子菜单也会将其从活动概览中的 `Applications` 视图中删除，从而防止用户在活动概览中搜索该项目。

过程 12.2. 示例：从附件子菜单中删除计算器菜单项

1. 查询 `/usr/share/applications/` 目录的内容，并确定与您要删除的菜单项对应的 `.desktop` 文件：

```
$ grep -r "Name=Calculator" /usr/share/applications/
/usr/share/applications/gcalctool.desktop:Name=Calculator
```

如上所示，计算器菜单项对应于 `/usr/share/applications/gcalctool.desktop` 文件。

2. 编辑 `/etc/xdg/menus/applications.menu` 文件，并使用 `<Menu>` 元素在该 `.menu` 文件末尾最后一个 `</Menu>` 标签前添加一个新的 `<Exclude>` 部分，如下所示：

```
<!-- Removes the Calculator from the Accessories submenu -->

<Menu>
  <Name>Accessories</Name>
  <Exclude>
    <Filename>gcalctool.desktop</Filename>
  </Exclude>
</Menu>

<!-- END of Calculator removal content -->

</Menu> <!-- End Applications -->
```

12.1.3. 删除单个用户的子菜单

给定用户的 **Applications** 菜单自定义默认存储在 `~/.config/menus/gnome-applications.menu` 定义文件中。可以通过设置 `$XDG_DATA_HOME` 环境变量来覆盖该文件的位置。

要覆盖 **Applications** 菜单的默认值，您首先需要创建一个 `gnome-applications.menu` 文件。请注意，从 **Applications** 菜单中删除子菜单也会从 **活动概览** 中的 **Applications** 视图中删除该子菜单中包含的所有菜单项，从而防止用户在 **活动概览** 中搜索这些项目。

例 12.1. 从应用程序菜单中删除系统工具子菜单

创建一个 `~/.config/menus/gnome-applications.menu` 文件：

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
"http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd">

<Menu>
  <Name>Applications</Name>
```

```

<MergeFile type="parent">/etc/xdg/menus/gnome-applications.menu</MergeFile>

<!-- Removes the System Tools submenu from the Applications menu-->

<Menu>
  <Name>System Tools</Name>
  <Deleted/>
</Menu>

<!-- END of System Tools removal content -->

</Menu>

```

如上所示，该文件包含一个指定子菜单名称（系统工具）的<Menu> 部分，并包含 <Deleted/> 标签。

12.1.4. 删除所有用户的子菜单

默认情况下，所有用户的 Applications 菜单自定义都存储在 /etc/xdg/menus/applications.menu 定义文件中。可以通过设置 \$XDG_CONFIG_DIRS 环境变量来覆盖该文件的位置。

要覆盖 Applications 菜单的默认值，您需要编辑 .menu 文件。请注意，从 Applications 菜单中删除子菜单也会从活动概览中的 Applications 视图中删除该子菜单中包含的所有菜单项，从而防止用户在活动概览中搜索这些项目。

例 12.2. 从应用程序菜单中删除系统工具子菜单

编辑 /etc/xdg/menus/applications.menu 文件，并使用 <Menu> 元素在该 .menu 文件末尾最后一个 </Menu> 标签前添加一个新的 <Deleted/> 部分，如下所示：

```

<!-- Removes the System Tools submenu from the Applications menu-->

<Menu>
  <Name>System Tools</Name>
  <Deleted/>
</Menu>

<!-- END of System Tools removal content -->

</Menu>

```

12.2. 自定义默认喜欢的应用程序

喜欢的应用程序是那些在 *活动概览* 中的 GNOME Shell dash 上看到的应用程序。您可以使用 `dconf` 为单个用户设置喜欢的应用程序，或者为所有用户设置相同的喜欢的应用程序。

12.2.1. 为单个用户设置不同的喜欢的应用程序

您可以通过修改 `~/.config/dconf/user` 中的用户数据库文件，为单个用户设置默认喜欢的应用程序。以下示例使用 `dconf` 将 `gedit`、`Terminal` 和 `Nautilus` 设置为用户的默认喜欢。示例代码允许用户稍后修改列表（如果他们想要这样做的话）。

例 12.3. `/etc/dconf/profile` 的内容：

```
# This line allows the user to change the default favorites later
user-db:user
```

例 12.4. `~/.config/dconf/user` 的内容：

```
# Set gedit, terminal and nautilus as default favorites
[org/gnome/shell]
favorite-apps = [gedit.desktop, 'gnome-terminal.desktop', 'nautilus.desktop']
```

注意

您也可以锁定上述设置以防止用户更改它们。如需更多信息，请参阅 [第 9.5.1 节“锁定特定设置”](#)。

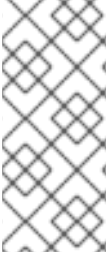
12.2.2. 为所有用户设置同样的喜欢的应用程序

要为所有用户具有相同的喜欢，您必须使用 `dconf keyfiles` 修改系统数据库文件。以下示例编辑 `dconf` 配置文件，然后创建一个 `keyfile` 来为机构的第一个 `floor` 中的所有员工设置默认喜欢的应用程序。

例 12.5. `/etc/dconf/profile` 的内容：

```
user-db:user

# This line defines a system database called first_floor
system-db:first_floor
```

注意

用户数据库文件的设置将优先于 `first_floor` 数据库文件中的设置，但 `first_floor` 数据库文件中引入的锁定将优先于用户中存在的设置。有关锁定的详情，请参考第 9.5.1 节“锁定特定设置”。

例 12.6. `/etc/dconf/db/first_floor.d/00_floor1_settings` 的内容：

```
# This sample sets gedit, terminal and nautilus as default favorites
# for all users in the first floor
[org/gnome/shell]
favorite-apps = ['gedit.desktop', 'gnome-terminal.desktop', 'nautilus.desktop']
```

运行 `dconf update` 命令将您的更改合并到系统数据库中。

在系统范围的设置生效前，用户必须退出并重新登录。

12.3. 配置文件关联

12.3.1. 什么是 MIME 类型？

在 GNOME 中，MIME（多用途互联网邮件扩展）类型用来识别文件的格式。GNOME 桌面使用 MIME 类型来：

- 确定默认情况下，哪个应用应打开特定的文件格式。
- 注册那些也可以打开特定文件格式的其他应用程序。
- 提供描述文件类型的字符串，例如，在 Files 应用程序的文件属性对话框中。
- 提供代表特定文件格式的图标，例如，在 Files 应用程序的文件属性对话框中。

MIME 类型名称采用给定格式：

media-type/subtype-identifier

例 12.7. MIME 类型格式

image/jpeg 是 MIME 类型的示例，其中 *image* 是介质类型，*jpeg* 是子类型标识符。

GNOME 遵循 [freedesktop.org Shared MIME Info](http://freedesktop.org/Shared-MIME-Info) 规范来决定：

- 机器范围和特定于用户的位置来保存所有 MIME 类型规范文件。
- 如何注册 MIME 类型，以便桌面环境知道可使用哪个应用程序来打开特定的文件格式。
- 用户如何更改哪些应用程序应打开哪些文件格式。

12.3.1.1. 什么是 MIME 数据库？

MIME 数据库是所有 MIME 类型规范文件的集合，GNOME 用来存储有关已知 MIME 类型的信息。

从系统管理员的角度来看，MIME 数据库最重要的部分是 `/usr/share/mime/packages/` 目录，其中存储了指定已知 MIME 类型信息的 MIME 类型相关文件。此类文件的一个示例是 `/usr/share/mime/packages/freedesktop.org.xml`，指定系统上默认可用的标准 MIME 类型的信息。该文件由 `shared-mime-info` 软件包提供。

获取更多信息

有关描述 MIME 类型系统的详细信息，请参阅 freedesktop.org 网站上的 `freedesktop.org` 共享 MIME 信息规范：

<http://www.freedesktop.org/wiki/Specifications/shared-mime-info-spec/>

12.3.2. 为所有用户添加自定义 MIME 类型

要为系统上的所有用户添加自定义 MIME 类型，并为该 MIME 类型注册默认应用程序，您需要在 `/usr/share/mime/packages/` 目录中创建一个新的 MIME 类型规格文件，并在 `/usr/share/applications/` 目录中创建一个 `.desktop` 文件。

过程 12.3. 为所有用户添加自定义 `application/x-newtype` MIME 类型

1. 创建 `/usr/share/mime/packages/application-x-newtype.xml` 文件：

```
<?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>
```

上面的示例 `application-x-newtype.xml` 文件定义了一个新的 MIME 类型 `application/x-newtype`，并将扩展名 `.xyz` 的文件名分配给该 MIME 类型。

2. 创建一个新的 `.desktop` 文件，例如 `myapplication1.desktop`，并将其放在 `/usr/share/applications/` 目录中：

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

上面的示例 `myapplication1.desktop` 文件将 `application/x-newtype` MIME 类型与名为 `My Application 1` 的应用程序相关联，后者由命令 `myapplication1` 运行。

根据 `myapplication1` 的启动方式，从 [Desktop Entry Specification](#) 中选择一个对应的字段代码。例如，对于能够打开多个文件的应用程序，请使用：

```
Exec=myapplication1 %F
```

3. 以 `root` 用户身份，更新 MIME 数据库以使更改生效：

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

4. 以 `root` 用户身份更新应用程序数据库：

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

5. 要验证您是否已成功将 `xyz` 文件与 `application/x-newtype` MIME 类型关联，请首先创建一个空文件，如 `test.xyz`：

```
$ touch test.xyz
```

然后运行 `gvfs-info` 命令：

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

6. 要验证 `myapplication1.desktop` 是否已正确设置为 `application/x-newtype` MIME 类型的默认注册的应用程序，请运行 `gvfs-mime --query` 命令：

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

12.3.3. 为单个用户添加自定义 MIME 类型

要为单个用户添加自定义 MIME 类型，并为该 MIME 类型注册默认应用程序，您需要在 `~/.local/share/mime/packages/` 目录中创建一个新的 MIME 类型规格文件，并在 `~/.local/share/applications/` 目录中创建一个 `.desktop` 文件。

过程 12.4. 为单个用户添加一个自定义 `application/x-newtype` MIME 类型

1. 创建 `~/.local/share/mime/packages/application-x-newtype.xml` 文件：

```
<?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>
```

上面的示例 `application-x-newtype.xml` 文件定义了一个新的 MIME 类型 `application/x-newtype`，并将扩展名 `.xyz` 的文件名分配给该 MIME 类型。

2. 创建一个新的 `.desktop` 文件，例如 `myapplication1.desktop`，并将其放在 `~/.local/share/applications/` 目录中：

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

上面的示例 `myapplication1.desktop` 文件将 `application/x-newtype` MIME 类型与名为 `My Application 1` 的应用程序相关联，后者由命令 `myapplication1` 运行。

根据 `myapplication1` 的启动方式，从 [Desktop Entry Specification](#) 中选择一个对应的字段代码。例如，对于能够打开多个文件的应用程序，请使用：

```
Exec=myapplication1 %F
```

3. 更新 MIME 数据库以使您的更改生效：

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

4. 更新应用程序数据库：

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

5. 要验证您是否已成功将 `xyz` 文件与 `application/x-newtype` MIME 类型关联，请首先创建一个空文件，如 `test.xyz`：

```
$ touch test.xyz
```

然后运行 `gvfs-info` 命令：

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

6. 要验证 `myapplication1.desktop` 是否已正确设置为 `application/x-newtype` MIME 类型的默认注册的应用程序，请运行 `gvfs-mime --query` 命令：

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

12.3.4. 覆盖所有用户的默认注册的应用程序

`/usr/share/applications/mimeapps.list` 和 `/usr/share/applications/[desktop environment name]-mimeapps.list` 文件是软件包安装的默认值，它指定默认注册哪个应用程序来打开特定的 MIME 类型。

要覆盖系统上所有用户的系统默认值，系统管理员需要创建 `/etc/xdg/mimeapps.list` 或 `/etc/xdg/[desktop environment name]-mimeapps.list` 文件，其中包含他们为其覆盖默认注册的应用程序的 MIME 类型列表。

应用配置的顺序如下：

- i. **`/usr/share/applications/`**
- ii. **`/etc/xdg/`**

在特定的地方，按以下顺序应用配置：

- i. **`mimeapps.list`**
- ii. **`[desktop environment name]-mimeapps.list`**

因此，系统管理员的配置优先于软件包配置。在每个配置中，特定于桌面的配置优先于不指定桌面环境的配置。

7.5 之前的 RED HAT ENTERPRISE LINUX 版本使用 DEFAULTS.LIST 文件，而不是 MIMEAPPS.LIST 文件。

过程 12.5. 覆盖所有用户的默认注册的应用程序

1.

查阅 `/usr/share/applications/mimeapps.list` 文件，以确定您要更改默认注册的应用程序的 MIME 类型。例如，以下 `mimeapps.list` 文件示例指定了 `text/html` 和 `application/xhtml+xml` MIME 类型的默认注册的应用程序：

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

默认应用程序(Firefox)通过指定对应的 `.desktop` 文件(`firefox.desktop`)来定义。其他应用程序的 `.desktop` 文件的默认位置为 `/usr/share/applications/`。

2.

创建 `/etc/xdg/mimeapps.list` 文件。在文件中，指定 MIME 类型及其相应的默认注册的应用程序：

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

这会将 `text/html` MIME 类型的默认注册的应用程序设为 `myapplication1.desktop`，将 `application/xhtml+xml` MIME 类型的默认注册的应用程序设为 `myapplication2.desktop`。

要使这些设置正常工作，请确保 `myapplication1.desktop` 和 `myapplication2.desktop` 文件都放在 `/usr/share/applications/` 目录中。

3.

您可以使用 `gvfs-mime query` 命令来验证默认注册的应用程序是否已正确设置：

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

12.3.5. 覆盖单个用户的默认注册的应用程序

`/usr/share/applications/mimeapps.list` 和 `/usr/share/applications/[desktop environment name]-mimeapps.list` 文件是软件包安装的默认值，它指定默认注册哪个应用程序来打开特定的 MIME 类型。

要覆盖单个用户的系统默认值，您需要创建 `~/.local/share/applications/mimeapps.list` 或 `~/.local/share/applications/[desktop environment id]-mimeapps.list` 文件，其中包含您要为其覆盖默认注册的应用程序的 MIME 类型列表。

应用配置的顺序如下：

- i. `/usr/share/applications/`
- ii. `/etc/xdg/`
- iii. `~/.local/share/application/`

在特定的地方，按以下顺序应用配置：

- i. `mimeapps.list`
- ii. `[desktop environment name]-mimeapps.list`

因此，用户的配置优先于系统管理员的配置，系统管理员的配置优先于软件包配置。在每个配置中，特定于桌面的配置优先于不指定桌面环境的配置。

7.5 之前的 RED HAT ENTERPRISE LINUX 版本使用 `DEFAULTS.LIST` 文件，而不是 `MIMEAPPS.LIST` 文件。

过程 12.6. 覆盖单个用户的默认注册的应用程序

1. 查阅 `/usr/share/applications/mimeapps.list` 文件，以确定您要更改默认注册的应用程序的 MIME 类型。例如，以下 `mimeapps.list` 文件示例指定了

text/html 和 application/xhtml+xml MIME 类型的默认注册的应用程序：

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

默认应用程序(Firefox)通过指定对应的 .desktop 文件(`firefox.desktop`)来定义。其他应用程序的 .desktop 文件的系统默认位置为 `/usr/share/applications/`。单个用户的 .desktop 文件可以保存在 `~/.local/share/applications/` 中。

2.

创建 `~/.local/share/applications/mimeapps.list` 文件。在文件中，指定 MIME 类型及其相应的默认注册的应用程序：

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

这会将 `text/html` MIME 类型的默认注册的应用程序设为 `myapplication1.desktop`，将 `application/xhtml+xml` MIME 类型的默认注册的应用程序设为 `myapplication2.desktop`。

要使这些设置正常工作，请确保 `myapplication1.desktop` 和 `myapplication2.desktop` 文件都放在 `/usr/share/applications/` 目录中。

3.

您可以使用 `gvfs-mime --query` 命令来验证默认注册的应用程序是否已正确设置：

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

第 13 章 自定义 GNOME 桌面功能

本章提到了三个主要的桌面功能。阅读后，您将了解如何默认为所有用户快速终止 X 服务器，如何启用 Compose 键 或如何禁用用户的命令行访问。

要确保您所做的更改生效，您需要更新 dconf 工具。用户退出登录并再次登录时会体验到不同。

13.1. 允许和禁止在线帐户

GNOME 在线帐户 (GOA) 用于设置个人网络帐户，然后自动与 GNOME 桌面和应用程序集成。用户可以使用 Online Accounts 应用程序添加其在线帐户，如 Google、Facebook、Flickr、ownCloud 等。

作为系统管理员，您可以

- 启用所有在线帐户；
- 有选择地启用几个在线帐户；
- 禁用所有在线帐户。

过程 13.1. 配置在线帐户

1. 如果系统上没有 `gnome-online-accounts` 软件包，请以 `root` 身份运行以下命令安装它：

```
# yum install gnome-online-accounts
```

2. 在 `/etc/dconf/db/local.d/goa` 中为本地数据库创建一个 keyfile，其中包含以下配置：

- 仅有选择地启用几个提供商：

```
[org/gnome/online-accounts]
whitelisted-providers= ['google', 'facebook']
```

- **禁用所有提供商：**

```
[org/gnome/online-accounts]
whitelisted-providers= []
```

- **允许所有可用的提供商：**

```
[org/gnome/online-accounts]
whitelisted-providers= ['all']
```

3. **锁定设置，以防止用户覆盖它们。**

- a. **如果不存在，请创建一个名为 `/etc/dconf/db/local.d/locks/` 的新目录。**

- b. **在 `/etc/dconf/db/local.d/locks/goa` 中创建一个新文件，其内容如下：**

```
# Prevent users from changing values for the following key:
/org/gnome/online-accounts
```

4. **更新系统数据库以使更改生效：**

```
# dconf update
```

5. **在系统范围的设置生效前，用户必须退出并重新登录。**

13.2. 启用 `CTRL+ALT+BACKSPACE` 快捷键

`Ctrl+Alt+Backspace` 快捷键组合用于终止 X 服务器。您可能想要终止 X 服务器，特别是当：

- **程序导致 X 服务器停止工作。**
- **您需要从登录的会话快速切换。**

- 您已启动了失败的程序。
- 由于各种原因，您无法在当前会话中操作。
- 您的屏幕冻结了。

要启用 **Ctrl+Alt+Backspace** 快捷方式，以默认为所有用户强制终止 X 服务器，您需要设置 `org.gnome.desktop.input-sources.xkb-options` GSettings 键。（有关 GSettings 键的更多信息，请参阅第 9.6 节“GSettings 键属性”。）

过程 13.2. 启用 Ctrl-Alt-Backspace 快捷键

1. 在 `/etc/dconf/db/local.d/00-input-sources` 中为机器范围的设置创建一个 local 数据库：

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

2. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/input-sources` 中更改它：

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

3. 更新系统数据库以使更改生效：

```
# dconf update
```

4. 在系统范围的设置生效前，用户必须退出并重新登录。

Ctrl+Alt+Backspace 组合键现已启用。所有用户都可以快速且轻松地终止 X 服务器，这样做会使他们回到登录提示符。

13.3. 启用组合键

Compose 键是一个功能，可让您输入键盘上找不到的特殊符号或字符。在 GNOME 桌面中，您可以

将键盘上的一个现有键定义为 **Compose 键**。**Compose 键** 与其他键（称为 **Compose 键序列**）结合使用，以输入您经常输入的特殊字符。

例 13.1. 使用组合键

如果按下并释放 **Compose 键**，则键入 **AE**（大写），您将会得到“unmarshal”。如果输入小写字母，则得到“æ”。

要启用 **Compose 键** 并将键盘上的特定键配置为 **Compose 键**，请设置 `org.gnome.desktop.input-sources.xkb-options` **GSettings** 键。这样，设置将会对系统上的所有用户默认启用。（有关 **GSettings** 键的更多信息，请参阅 [第 9.6 节“GSettings 键属性”](#)。）

过程 13.3. 将右 Alt 键设为组合键

1. 在 `/etc/dconf/db/local.d/00-input-sources` 中为机器范围的设置创建一个 **local** 数据库：

```
[org/gnome/desktop/input-sources]
# Set the Right Alt key as the Compose key and enable it
xkb-options=['compose:ralt']
```

如果要设置与 **Right Alt** 不同的键，请将 `ralt` 替换为 `xkeyboard-config(7)` man page 中指定的该键的名称，**section Compose 键位置**。

2. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/input-sources` 中更改它：

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

3. 更新系统数据库以使更改生效：

```
# dconf update
```

4. 在系统范围的设置生效前，用户必须退出并重新登录。

现在，您可以开始使用右 **Alt** 键作为 **Compose 键**。只需按下并释放 **Compose 键**，然后键入组合键以获得特定的符号。



注意

如果您作为系统管理员，在编辑完本地数据库后设置了锁定，则用户将无法添加其它 `xkb` 选项。锁定可防止用户设置 `Alt+Shift` 等组合键来切换键盘布局。

13.4. 禁用命令行访问

要对桌面用户禁用命令行访问，您需要在几个不同的上下文中进行配置更改。请记住，以下步骤不会删除桌面用户访问命令行的权限，而是删除桌面用户可能访问命令行的方法。

- 设置 `org.gnome.desktop.lockdown.disable-command-line` GSettings 键，这将阻止用户访问终端或指定要执行的命令行(`Alt+F2` 命令提示符)。
- 通过修改 X 服务器配置，禁用通过 `Ctrl+Alt+功能键` 快捷键切换到虚拟终端(VT)。
- 从 `Applications` 菜单和 `GNOME Shell` 中的 `Activities Overview` 中删除 `Terminal` 和任何其他提供访问终端的应用程序。这可以通过删除这些应用程序的菜单项来完成。有关如何删除菜单项的详情，请参考第 12.1.2 节“删除所有用户的菜单项”。

13.4.1. 设置 `org.gnome.desktop.lockdown.disable-command-line` 键

1. 在 `/etc/dconf/db/local.d/00-lockdown` 中为机器范围的设置创建一个 local 数据库：

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

2. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/lockdown` 中更改它：

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

3. 更新系统数据库：

```
# dconf update
```

4.

在系统范围的设置生效前，用户必须退出并重新登录。

13.4.2. 禁止虚拟终端切换

用户通常可以使用 **Ctrl+Alt+功能键** 快捷键（例如 **Ctrl+Alt+F2**）从 GNOME 桌面和 X 服务器切换到虚拟终端。您可以通过在 `/etc/X11/xorg.conf.d/` 目录中的 X 配置文件中的 **Serverflags** 部分添加 **DontVTSwitch** 选项来禁用对所有虚拟终端的访问。

过程 13.4. 禁止访问虚拟终端

1.

创建或编辑 `/etc/X11/xorg.conf.d/` 目录中的 X 配置文件：



注意

按照惯例，这些特定于主机的配置文件名称以两个数字和一个连字符开头，并且总是以 `.conf` 为扩展名。因此，以下文件名可以是 `/etc/X11/xorg.conf.d/10-xorg.conf`。

```
Section "Serverflags"
```

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

```
EndSection
```

2.

重启 X 服务器以使更改生效。

13.5. 锁定打印

您可以禁止向用户显示打印对话框。如果您为用户提供临时访问权限，或者您不希望用户打印到网络打印机上，这将非常有用。



重要

这个功能只能在支持它的应用程序中使用。并非所有 GNOME 和第三方应用都启用了此功能。这些更改对不支持此功能的应用程序没有影响。

您可以通过锁定 `org.gnome.desktop.lockdown.disable-printing` 键来防止应用程序打印。按照以下流程操作。

过程 13.5. 锁定 `org.gnome.desktop.lockdown.disable-printing` 键

1. 如果不存在，请创建用户配置文件(`/etc/dconf/profile/user`)：

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

2. 在 `/etc/dconf/db/local.d/00-lockdown` 中为机器范围的设置创建一个 local 数据库：

```
[org/gnome/desktop/lockdown]

# Prevent applications from printing
disable-printing=true
```

3. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/lockdown` 中更改它：

```
# List the keys used to configure lockdown
/org/gnome/desktop/lockdown/disable-printing
```

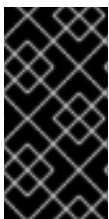
4. 运行以下命令来更新系统数据库

```
# dconf update
```

遵循了这些步骤后，支持此锁定键的应用程序将禁止打印。在此类应用程序中有 *Evolution*、*Evince*、*Eye of GNOME*、*Epiphany* 和 *Gedit*。

13.6. 锁定保存在磁盘上的文件

您可以禁用 **Save** 和 **Save As** 对话框。如果您为用户提供临时访问权限，或者您不希望用户将文件保存到计算机上，这将非常有用。

**重要**

这个功能只能在支持它的应用程序中使用。并非所有 GNOME 和第三方应用都启用了此功能。这些更改对不支持此功能的应用程序没有影响。

您可以通过锁定 `org.gnome.desktop.lockdown.disable-save-to-disk` 键来防止应用程序保存文件。按照以下流程操作：

过程 13.6. 锁定 `org.gnome.desktop.lockdown.disable-save-to-disk` 键

1. 在 `/etc/dconf/profile/user` 中创建 `user` 配置文件，除非它已存在：

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

2. 在 `/etc/dconf/db/local.d/00-lockdown` 文件中为机器范围的设置创建一个 `local` 数据库。

```
[org/gnome/desktop/lockdown]
# Prevent the user from saving files on disk
disable-save-to-disk=true
```

3. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/lockdown` 中更改它：

```
# Lock this key to disable saving files on disk
/org/gnome/desktop/lockdown/disable-save-to-disk
```

4. 运行以下命令来更新系统数据库

```
# dconf update
```

遵循了这些步骤后，支持此锁定键的应用程序，如 `QPC`、`Image Viewer`、`Evolution`、`Document Viewer` 或 `GNOME Shell` 将禁用其“Save As”对话框。

13.7. 锁定重新分区

`polkit` 可让您为单个操作设置权限。对于 `udisks2`，磁盘管理服务工具，配置位于 `/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy`。此文件包含一组操作和默认值，它们可被系统管理员覆盖。

**重要**

请记住，存储在 `/etc` 中的 `polkit` 配置会覆盖 `/usr/share/` 中软件包所提供的配置。

过程 13.7. 要禁止用户更改磁盘设置

1.

创建一个与 `/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy` 中内容相同的文件。

```
cp /usr/share/polkit-1/actions/org.freedesktop.udisks2.policy /etc/share/polkit-1/actions/org.freedesktop.udisks2.policy
```

不要更改 `/usr/share/polkit-1/actions/org.freedesktop.udisks2.policy` 文件，您的更改会被下一次软件包更新覆盖。

2.

删除您不需要的操作，并将以下行添加到 `/etc/polkit-1/actions/org.freedesktop.udisks2.policy` 文件中：

```
<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>
```

如果只想确保只有 `root` 用户才可以执行操作，则需要把 `no` 替换为 `auth_admin`。

3.

保存更改。

当用户尝试更改磁盘设置时，会返回以下信息：

```
Authentication is required to modify the disks settings
```

13.8. 锁定用户退出登录和用户切换

按照以下步骤阻止用户退出登录。

1. 创建 `/etc/dconf/profile/user` 配置文件，其中包含以下行：

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

`local` 是 `dconf` 数据库的名称。

2. 如果目录 `/etc/dconf/db/local.d/` 不存在，请创建该目录。
3. 创建密钥文件 `/etc/dconf/db/local.d/00-logout`，为本地数据库提供信息：

```
[org/gnome/desktop/lockdown]
# Prevent the user from user switching
disable-log-out=true
```

4. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/lockdown` 中更改它：

```
# Lock this key to disable user logout
/org/gnome/desktop/lockdown/disable-log-out
```

5. 更新系统数据库：

```
# dconf update
```

6. 在系统范围的设置生效前，用户必须退出并重新登录。

重要

用户可以通过切换到其他用户来逃避退出登录锁定，这可以阻止系统管理员的意图。这就是建议禁止“用户切换”以及阻止此类情况发生的原因。

过程 13.8. 防止用户切换到不同的用户帐户

1. 创建 `/etc/dconf/profile/user` 配置文件，其中包含以下行：

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

`local` 是 `dconf` 数据库的名称。

2. 如果目录 `/etc/dconf/db/local.d/` 不存在，请创建该目录。

3. 创建密钥文件 `/etc/dconf/db/local.d/00-user-switching`，为本地数据库提供信息：

```
[org/gnome/desktop/lockdown]
# Prevent the user from user switching
disable-user-switching=true
```

4. 覆盖用户设置，并防止用户在 `/etc/dconf/db/local.d/locks/lockdown` 中更改它：

```
# Lock this key to disable user switching
/org/gnome/desktop/lockdown/disable-user-switching
```

5. 更新系统数据库：

```
# dconf update
```

6. 在系统范围的设置生效前，用户必须退出并重新登录。

13.9. 单应用程序模式

单应用程序模式是修改的 `shell`，可将 `shell` 重新配置为交互式 `kiosk`。管理员锁定一些行为，来使标准桌面对用户有更严格的限制，从而使他们专注于所选的功能。

在多个领域（从通信到娱乐或教育）为各种功能设置单应用程序模式，并将其用作自助服务机、事件管理器、注册点等。

过程 13.9. 设置单应用程序模式

1.

使用以下内容创建以下文件：

- `/usr/bin/redhat-kiosk`

```
#!/bin/sh

if [ ! -e ~/.local/bin/redhat-kiosk ]; then
    mkdir -p ~/.local/bin ~/.config
    cat > ~/.local/bin/redhat-kiosk << EOF
#!/bin/sh
# This script is located in ~/.local/bin.
# It's provided as an example script to show how
# the kiosk session works. At the moment, the script
# just starts a text editor open to itself, but it
# should get customized to instead start a full screen
# application designed for the kiosk deployment.
# The "while true" bit just makes sure the application gets
# restarted if it dies for whatever reason.

while true; do
    gedit ~/.local/bin/redhat-kiosk
done

EOF

    chmod +x ~/.local/bin/redhat-kiosk
    touch ~/.config/gnome-initial-setup-done
fi

exec ~/.local/bin/redhat-kiosk "$@"
```

**重要**

`/usr/bin/redhat-kiosk` 文件必须是可执行的。

将 `gedit ~/.local/bin/redhat-kiosk` 代码替换为您要在 kiosk 会话中执行的命令。这个示例启动了一个为 kiosk 部署而设计的全屏应用程序，名为 `http://mine-kios-web-app`：

```
[...]
while true; do
    firefox --kiosk http://mine-kios-web-app
done
[...]
```

•

/usr/share/applications/com.redhat.Kiosk.Script.desktop

```
[Desktop Entry]
Name=Kiosk
Type=Application
Exec=redhat-kiosk
```

•

/usr/share/applications/com.redhat.Kiosk.WindowManager.desktop

```
[Desktop Entry]
Type=Application
Name=Mutter
Comment=Window manager
Exec=/usr/bin/mutter
Categories=GNOME;GTK;Core;
OnlyShowIn=GNOME;
NoDisplay=true
X-GNOME-Autostart-Phase=DisplayServer
X-GNOME-Provides=windowmanager;
X-GNOME-Autostart-Notify=true
X-GNOME-AutoRestart=false
X-GNOME-HiddenUnderSystemd=true
```

•

/usr/share/gnome-session/sessions/redhat-kiosk.session

```
[GNOME Session]
Name=Kiosk
RequiredComponents=com.redhat.Kiosk.WindowManager;com.redhat.Kiosk.Script;
```

•

/usr/share/xsessions/com.redhat.Kiosk.desktop

```
[Desktop Entry]
Name=Kiosk
Comment=Kiosk mode
Exec=/usr/bin/gnome-session --session=redhat-kiosk
DesktopNames=Red-Hat-Kiosk;GNOME;
```

2.

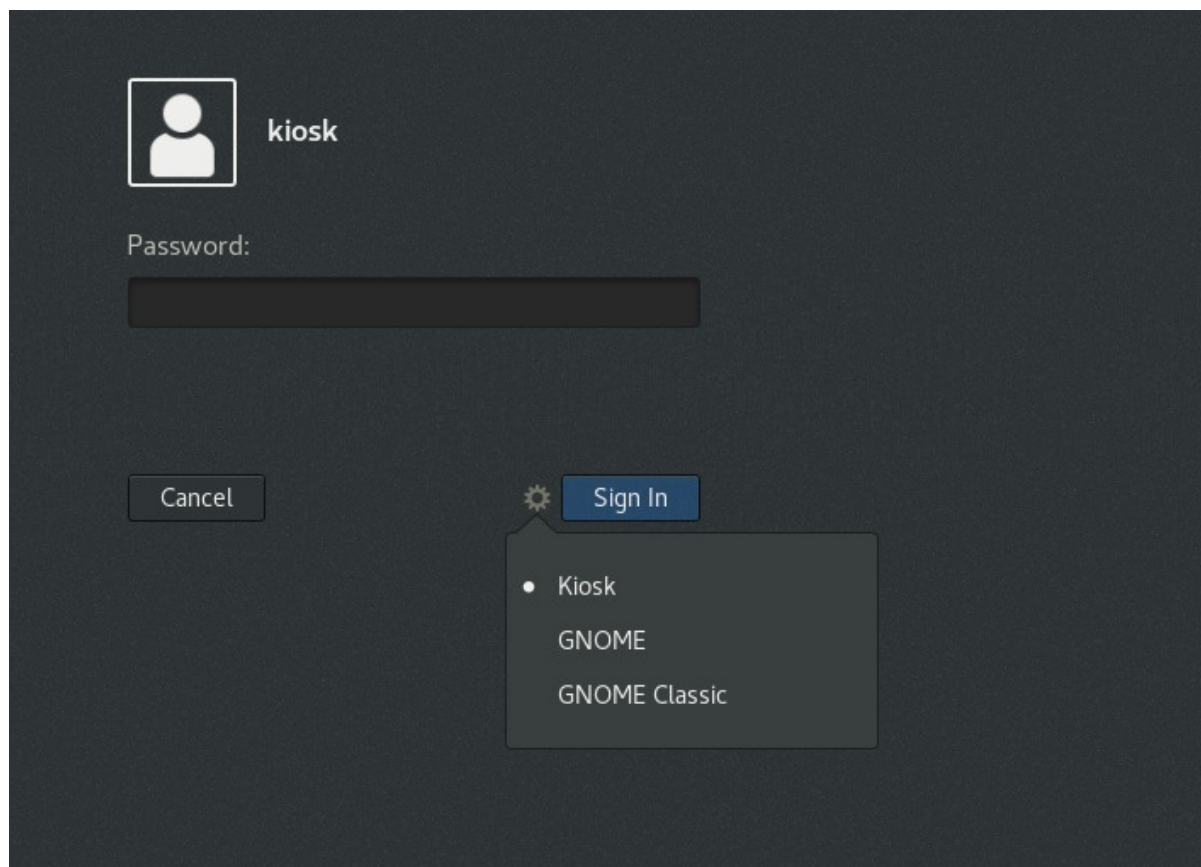
重启 GDM 服务：

```
systemctl restart gdm.service
```

3.

为 kiosk 会话创建一个单独的用户，并选择 Kiosk 作为 kiosk 会话的用户的会话类型。

图 13.1. 选择 kiosk 会话



通过启动 Kiosk 会话，用户将启动一个专为 kiosk 部署而设计的全屏应用程序，。

13.10. 防止计算机在合上盖子时挂起

合上笔记本电脑的盖子时，您的计算机会挂起以省电。您可以通过更改该行为的设置来防止计算机在合上盖子时挂起。



警告

一些笔记本电脑在盖子合上时仍在运行会导致过热，尤其是放在像背包这样密闭的地方。因此，请考虑更改默认设置（挂起）是否是您的最佳选项。

过程 13.10. 配置盖子开关

1. 打开 `/etc/systemd/logind.conf` 文件进行编辑。
2. 在文件中找到 `HandleLidSwitch=suspend` 行。如果开头是用 `#` 字符引用的，请去掉引用。

如果文件中不存在该行，请添加它。

3. 将默认 `suspend` 参数替换为

- `lock` 用于锁定屏幕；
- `ignore` 对于什么事情都不发生；
- `poweroff` 以关闭计算机。

例如：

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

4. 保存更改并关闭编辑器。
5. 运行以下命令，以便您的更改保留到下一次系统重启：

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



警告

请记住，重启服务会强制中断任何已登录的桌面用户当前运行的 GNOME 会话。这会导致用户丢失未保存的数据。

有关 `/etc/systemd/logind.conf` 文件的详情，请查看 `logind.conf (5)` 手册页。

13.11. 在图形目标模式下按下 POWER 按钮时，会更改行为

当机器引导至图形登录屏幕或用户会话时，按 `power` 按钮默认使机器挂起。当用户物理按下 `power` 按钮或从远程控制台按下虚拟电源按钮时，会出现这种情况。要在按下 `power` 按钮时实现不同的行为，请使用 `dconf` 设置此按钮的功能。

例如，如果您希望系统在按下 `power` 按钮后关机，请使用以下步骤：

过程 13.11. 在按下 `power` 按钮后，使用 `dconf` 将系统配置为关闭

1. 在 `/etc/dconf/db/local.d/01-power` 文件中为系统范围的设置创建一个 `local` 数据库：

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

2. 覆盖用户的设置，并阻止用户在 `/etc/dconf/db/local.d/locks/01-power` 文件中更改它：

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

3. 更新系统数据库：

```
# dconf update
```

4. 在系统范围的设置生效之前，退出并重新登录。

在按下 `power` 按钮后，此配置将启动系统关闭。要将系统配置为采取不同的操作，您可以设置特定按钮的行为。

特定按钮的选项

`nothing`

什么都不做

suspend

挂起系统

hibernate

休眠系统

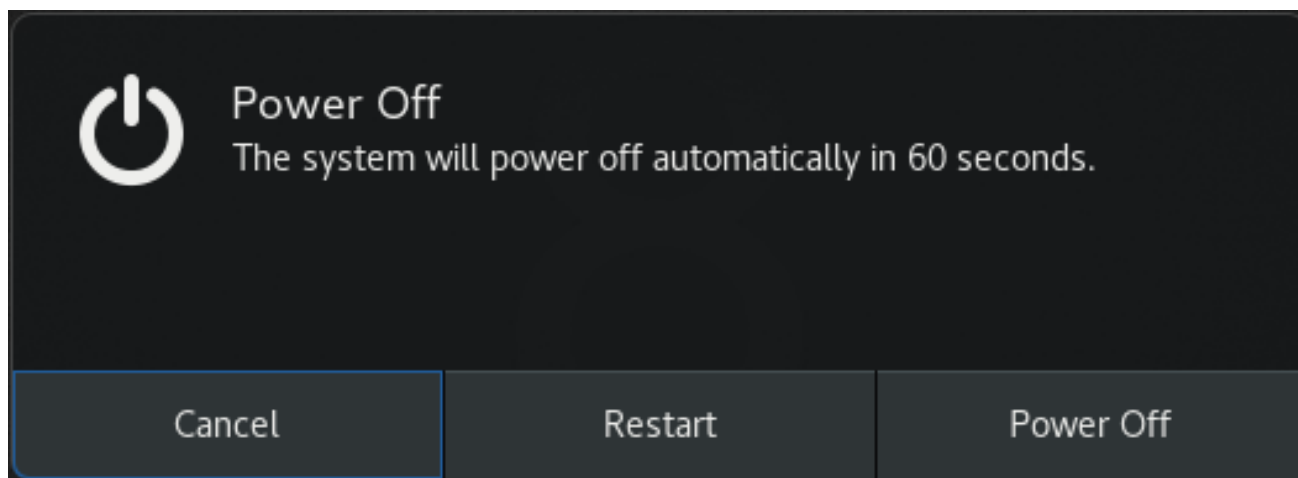
interactive

显示一个弹出查询，询问用户要做什么

使用交互模式时，在按下 **power** 按钮后，系统会在 **60** 秒后自动关闭。

但是，您可以从弹出查询中选择不同的行为，如下图所示。

图 13.2. 交互模式的弹出查询



第 14 章 会话管理

14.1. GDM 是什么？

GNOME 显示管理器 (GDM)是一个在后台运行的图形登录程序，它运行和管理本地和远程登录的 X 服务器。

GDM 是 XDM (X 显示管理器)的替代品。但是 GDM 并不是来自 XDM，不包含任何原始的 XDM 代码。另外，在 GDM 中不支持图形配置工具，因此需要编辑 `/etc/gdm/custom.conf` 配置文件来更改 GDM 设置。

14.1.1. 重启 GDM

当您更改系统配置时，如设置登录屏幕横幅消息、登录屏幕徽标或登录屏幕背景时，您需要重启 GDM 以使更改生效。



警告

请记住，重启服务会强制中断任何已登录的桌面用户当前运行的 GNOME 会话。这会导致用户丢失未保存的数据。

要重启 GDM 服务，请运行以下命令：

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

有关管理 Red Hat Enterprise Linux 7 上服务的更多信息，请参阅 [系统管理员指南](#)。

14.1.2. 显示 GDM 配置的结果

要显示 GDM 配置的结果，请运行以下命令：

```
$ DCONF_PROFILE=gdm gsettings list-recursive org.gnome.login-screen
```

14.2. 身份验证

14.2.1. 使用企业凭证登录到 GNOME

如果您的网络有活动目录或身份管理域，并且您有域帐户，则您可以使用您的域凭证登录到 GNOME。

如果已为域帐户成功配置了计算机，则用户可以使用他们的帐户登录 GNOME。在登录提示下，键入域用户名，后跟 @ 符号，然后是您的域名。例如，如果您的域名是 `example.com`，用户名是 `User`，则输入：

```
User@example.com
```

在已为域帐户配置了计算机的情况下，您应该会看到描述登录格式的帮助提示。

14.2.1.1. 在欢迎屏幕中选择使用企业凭证

如果您还没有为企业凭证配置机器，您可以在作为 GNOME Initial Setup 程序一部分的欢迎屏幕中完成此操作。

过程 14.1. 配置企业凭证

1. 在登录欢迎屏幕中，选择 **Use Enterprise Login**。
2. 如果还没有预先填充，请在 **Domain** 字段中输入您的域名。
3. 在相关字段中输入您的域帐户用户和密码。
4. 点 **Next**。

根据域的配置方式，提示可能会显示输入域管理员的名称和密码，以便继续。

14.2.1.2. 更改为使用企业凭据登录到 GNOME

如果您已完成初始设置，并且希望启动一个域帐户来登录到 GNOME，那么您可以在 GNOME 设置

中的 **Users** 面板上完成此操作。

过程 14.2. 配置企业凭证

1. 点顶部栏中的名称，然后从菜单中选择 **Settings**。
2. 从项目列表中，选择 **Users**。
3. 点 **Unlock** 按钮，并输入计算机管理员的密码。
4. 单击窗口左下角的 **+** 按钮。
5. 选择 **Enterprise Login** 窗格。
6. 输入您的 **Enterprise** 帐户的域、用户和密码，然后单击 **Add**。

根据您的配置域的方式，提示可能会显示输入域管理员的名称和密码，以便继续。

14.2.1.3. 故障排查和高级设置

`realm` 命令及其各种子命令可用于对企业登录功能进行故障排除。例如，要查看是否已为企业登录配置了机器，请运行以下命令：

```
$ realm list
```

建议网络管理员事先将工作站加入相关域。这可以通过 `kickstart realm join` 命令完成，或者通过脚本以自动的方式运行 `realm join`。

获取更多信息

[Red Hat Enterprise Linux 7 Windows 集成指南](#) - Red Hat Enterprise Linux 7 的 Windows 集成指南 提供了有关使用 `realmd` 连接到 Active Directory 域的更多详细信息。

14.2.2. 启用智能卡验证

启用智能卡验证需要两个连续的步骤：

1. **配置 GDM 以允许提示智能卡**
2. **配置操作系统以允许使用智能卡登录**

1.配置 GDM 以允许提示智能卡

您可以使用两种方式配置 GDM，以允许提示智能卡验证：

dconf 编辑器 GUI

过程 14.3. 使用 dconf 编辑器 GUI 启用智能卡验证

1. **取消选中 `org.gnome.login-screen enable-password-authentication` dconf 键的框。**
2. **选中 `org.gnome.login-screen enable-smartcard-authentication` dconf 键的框。**

dconf-tool

过程 14.4. 使用 dconf-tool 启用智能卡验证

1. **在 `/etc/dconf/db/gdm.d` 目录中创建一个 keyfile。**
2. **将以下内容添加到此 keyfile 中：**

```
[org/gnome/login-screen]
enable-password-authentication='false'
enable-smartcard-authentication='true'
```

3. **更新系统 dconf 数据库：**

```
# dconf update
```

2. 配置操作系统以允许使用智能卡登录

为智能卡验证配置了 GDM 后，使用 `system-config-authentication` 工具将系统配置为允许用户使用智能卡，使其可用于 GDM 作为图形环境的有效验证方法。该工具由 `authconfig-gtk` 软件包提供。

要了解更多有关将系统配置为允许智能卡验证的信息，并了解更多有关 `system-config-authentication` 工具的信息，请参阅 [Red Hat Enterprise Linux 7 系统级身份验证指南](#)。

14.2.3. 启用指纹身份验证

要允许用户使用其注册的指纹登录，请使用 `system-config-authentication` 工具启用指纹身份验证。该工具由 `authconfig-gtk` 软件包提供。

要了解更多有关指纹身份验证和 `system-config-authentication` 工具的信息，请参阅 [Red Hat Enterprise Linux 7 系统级身份验证指南](#)。

14.3. 用户会话

14.3.1. 用户会话中典型的进程是什么？

在普通的 GNOME 会话中，称为 **守护进程** 的程序作为后台进程在系统上运行。您应该能够找到以下默认运行的守护进程：

`dbus-daemon`

`dbus-daemon` 提供了一个消息总线守护进程，程序可以使用它来交换消息。`dbus-daemon` 使用 D-Bus 库实现，它在两个应用程序之间提供一对一通信。

有关进一步的信息，请参阅 `dbus-daemon(1) man page`。

`gnome-keyring-daemon`

不同程序和 Web 站点的用户名和密码等凭证使用 `gnome-keyring-daemon` 安全地存储。此信息被写入到名为 `keyring` 文件的加密文件中，保存在用户的主目录下。

有关进一步的信息，请参阅 `gnome-keyring-daemon(1) man page`。

`gnome-session`

gnome-session 程序负责使用显示管理器（如 GDM）运行 GNOME 桌面环境。用户的默认会话是系统管理员在系统安装时设置的。**gnome-session** 通常加载系统上成功运行的最后一个会话。

有关进一步的信息，请参阅 **gnome-session(1) man page**。

gnome-settings-daemon

gnome-settings-daemon 处理 GNOME 会话的设置，以及会话内运行的所有程序。

有关进一步的信息，请参阅 **gnome-settings-daemon(1) man page**。

gnome-shell

gnome-shell 为 GNOME 提供核心用户界面功能，如启动程序、浏览目录、查看文件等。

有关进一步的信息，请参阅 **gnome-shell(1)手册页**。

pulseaudio

pulseaudio 是 Red Hat Enterprise Linux 的一个声音服务器，可让程序使用 Pulseaudio 守护进程输出音频。

有关进一步的信息，请参阅 **pulseaudio(1) man page**。

根据用户的设置，您可能还会看到以下一些内容：

- **dconf-service**
- **ibus**
- **at-spi2-dbus-launcher**

- **at-spi2-registryd**
- **gnome-shell-calendar-server**
- **goa-daemon**
- **gsd-printer**
- **各种 Evolution 工厂进程**
- **各种 GVFS 进程**

14.3.2. 配置用户默认会话

默认会话是从名为 **AccountsService** 的程序检索的。**AccountsService** 将此信息存储在 **/var/lib/AccountsService/users/** 目录中。



注意

在 **GNOME 2** 中，用户主目录中的 **.dmrc** 文件用于创建默认会话。不再使用这个 **.dmrc** 文件。

过程 14.5. 为用户指定默认会话

1. 确保您通过运行以下命令安装了 **gnome-session-xsession** 软件包：

```
# yum install gnome-session-xsession
```

2. 导航到 **/usr/share/xsessions** 目录，您可以在其中找到每个可用会话的 **.desktop** 文件。查阅 **.desktop** 文件的内容以确定您要使用的会话。
3. 要为用户指定默认会话，请在 **/var/lib/AccountsService/users/username** 文件中更新用户的帐户服务：

```
[User]
Language=
XSession=gnome
```

在这个示例中，使用 `/usr/share/xsessions/gnome.desktop` 文件将 GNOME 设置为默认会话。请注意，Red Hat Enterprise Linux 7 中的系统默认是 GNOME 经典(`/usr/share/xsessions/gnome-classic.desktop` 文件)。

为用户指定默认会话后，该会话将在下次登录时使用，除非用户从登录屏幕选择了不同的会话。

14.3.3. 创建自定义会话

要使用自定义配置创建自己的会话，请按照以下步骤操作：

1.

在 `/etc/X11/sessions/new-session.desktop` 中创建一个 `.desktop` 文件。确保文件指定了以下条目：

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=Custom Session
Comment=This is our custom session
Exec=gnome-session --session=new-session
```

`Exec` 条目指定要执行的命令（可能带有参数）。您可以使用 `gnome-session --session=new-session` 命令运行自定义会话。

有关您可以在 `gnome-session` 中使用的参数的更多信息，请参阅 `gnome-session (1)` 手册页。

2.

在 `/usr/share/gnome-session/sessions/new-session.session` 中创建一个自定义会话文件，您可以在其中指定会话的名称和所需组件：

```
[GNOME Session]
Name=Custom Session
RequiredComponents=gnome-shell-classic;gnome-settings-daemon;
```

请注意，您在 `RequiredComponents` 中指定的任何项目都需要在 `/usr/share/applications/` 中具有对应的 `.desktop` 文件。

配置自定义会话文件后，GDM 登录屏幕的会话列表中提供新会话。

14.3.4. 查看用户会话日志

如果要查找用户会话中问题的更多信息，您可以查看 `systemd` 日志。因为 Red Hat Enterprise Linux 7 是一个基于 `systemd` 的系统，所以用户会话日志数据以二进制格式直接存储在 `systemd` 日志中。



注意

在 Red Hat Enterprise Linux 6 中，用户会话日志数据存储在 `~/.xsession-errors` 文件中，该文件已不再使用。

过程 14.6. 查看用户会话日志

1. 运行以下命令，确定您的用户 ID (uid)：

```
$ id --user  
1000
```

2. 查看上面确定的用户 ID 的日志：

```
$ journalctl _UID=1000
```

获取更多信息

`journalctl(1)` man page 提供了有关 `systemd` 日志使用情况的更多信息。

有关在 Red Hat Enterprise Linux 7 中使用 `systemd` 日志的详情，请查看 [Red Hat Enterprise Linux 7 系统级身份验证指南](#)。

14.3.5. 为所有用户添加自动启动应用程序

要在用户登录时自动启动应用程序，您需要在 `/etc/xdg/autostart/` 目录中为该应用程序创建一个 `.desktop` 文件。

要为单个用户管理自动启动（启动）应用程序，请使用 `gnome-session-properties` 应用程序。

过程 14.7. 为所有用户添加自动启动（启动）应用程序

1. 在 `/etc/xdg/autostart/` 目录中创建一个 `.desktop` 文件：

```
[Desktop Entry]
Type=Application
Name=Files
Exec=nautilus -n
OnlyShowIn=GNOME;
AutostartCondition=GSettings org.gnome.desktop.background show-desktop-icons
```

2. 使用应用程序的名称替换 `Files`。
3. 使用您要用来运行应用程序的命令替换 `nautilus -n`。
4. 您可以使用 `AutostartCondition` 键来检查 `GSettings` 键的值。

如果键的值为 `true`，则会话管理器会自动运行应用程序。如果键值在运行的会话中发生了更改，会话管理器将启动或停止应用程序，具体取决于键之前的值是多少。

14.3.6. 配置自动登录

具有 `Administrator` 帐户类型的用户可以从 `GNOME Settings` 中的 `Users` 面板中启用 `Automatic Login`。系统管理员也可以在 `GDM` 自定义配置文件中手动设置自动登录，如下所示：

例 14.1. 为用户 `john` 配置自动登录

编辑 `/etc/gdm/custom.conf` 文件，并确保文件中的 `[daemon]` 部分指定了以下内容：

```
[daemon]
AutomaticLoginEnable=True
AutomaticLogin=john
```

使用您要自动登录的用户替换 `john`。

14.3.7. 配置自动退出登录

闲置一段时间的用户会话可以自动终止。您可以通过设置相应的 **GSettings** 键，根据机器是否通过电池运行还是通过主电源运行来设置不同的行为，然后锁定它。



警告

请记住，如果闲置会话被自动终止，用户可能会丢失未保存的数据。

过程 14.8. 为由电源供电的机器设置自动退出登录

1. 在 `/etc/dconf/db/local.d/00-autologout` 中为机器范围的设置创建一个 **local** 数据库：

```
[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. 覆盖用户的设置，并阻止用户在 `/etc/dconf/db/local.d/locks/autologout` 中更改它：

```
# 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. 更新系统数据库：

```
# dconf update
```

4. 在系统范围的设置生效前，用户必须退出并重新登录。

下面的 **GSettings** 键值得关注的：

-

`org.gnome.settings-daemon.plugins.power.sleep-inactive-ac-timeout`

如果计算机是通过 AC 电源运行的，则它在休眠前需要处于非活动状态的秒数。

-

`org.gnome.settings-daemon.plugins.power.sleep-inactive-ac-type`

如果计算机通过 AC 电源运行，则超时时会发生什么。

-

`org.gnome.settings-daemon.plugins.power.sleep-inactive-battery-timeout`

如果计算机通过电源运行，它在休眠前需要处于非活动状态的秒数。

-

`org.gnome.settings-daemon.plugins.power.sleep-inactive-battery-type`

如果计算机通过电池运行，超时时会发生什么。

您可以在键上运行 `gsettings range` 命令，以获取您可以使用的值的列表。例如：

```
$ gsettings range org.gnome.settings-daemon.plugins.power sleep-inactive-ac-type
enum
'blank'
'suspend'
'shutdown'
'hibernate'
'interactive'
'nothing'
'logout'
```

14.3.8. 设置屏幕亮度和空闲时间

通过设置以下 `GSettings` 键，您可以设置亮度级的下降，并设置亮度级别和闲置时间。

例 14.2. 设置亮度级的下降

要在设备闲置一段时间后设置亮度级别的下降，请在 `/etc/dconf/db/local.d/00-power` 中为机器范

围的设置创建一个 local 数据库，如下例所示：

```
[org/gnome/settings-daemon/plugins/power]
idle-dim=true
```

例 14.3. 设置亮度级

要更改亮度级别，请在 `/etc/dconf/db/local.d/00-power` 中为机器范围的设置创建一个 local 数据库，如下例所示，并将 30 替换为您要使用的整数值：

```
[org/gnome/settings-daemon/plugins/power]
idle-brightness=30
```

例 14.4. 设置空闲时间

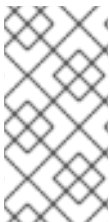
要设置屏幕显示的空闲时间，且显示默认屏保，请在 `/etc/dconf/db/local.d/00-session` 中为机器范围的设置创建一个 local 数据库，如下例所示，并将 900 替换为您要使用的整数值：

```
[org/gnome/desktop/session]
idle-delay=uint32 900
```

您必须包含 `uint32` 以及整数值，如上所示。

以 root 用户身份运行 `dconf update` 命令，将您的更改合并到系统数据库中。

在系统范围的设置生效前，用户必须退出并重新登录。



注意

您也可以锁定上述设置以防止用户更改它们。有关锁的详情，请参考第 9.5.1 节“锁定特定设置”。

14.3.9. 当用户空闲时锁定屏幕

如果要启用屏保，并在用户闲置时自动锁定屏幕，您需要创建一个 `dconf` 配置文件，设置 `GSettings` 键对，然后将其锁定以防止用户编辑它。

过程 14.9. 启用屏保和锁定屏幕

1.

在 `/etc/dconf/db/local.d/00-screensaver` 中为系统范围的设置创建一个 `local` 数据库：

```
[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
```

您必须包含 `uint32` 以及整数键值，如上所示。

2.

覆盖用户的设置，并阻止用户在 `/etc/dconf/db/local.d/locks/screensaver` 文件中更改它：

```
# Lock desktop screensaver settings
/org/gnome/desktop/session/idle-delay
/org/gnome/desktop/screensaver/lock-enabled
/org/gnome/desktop/screensaver/lock-delay
```

3.

更新系统数据库：

```
# dconf update
```

4.

在系统范围的设置生效前，用户必须退出并重新登录。

14.3.10. 录屏

GNOME Shell 具有内置屏幕广播记录器，允许用户在会话期间记录桌面或应用程序活动，并以 `webm` 格式将录制作为高分辨率视频文件分发。

过程 14.10. 进行录屏

1.

要启动录制，请按 **Ctrl+Alt+Shift+R**。

当记录器捕获屏幕活动时，它会在屏幕的右上角显示一个红色圆圈。

2. 要停止录制，请按 **Ctrl+Alt+Shift+R**。屏幕右下角的红色圆就消失了。
3. 导航到 `~/Videos` 文件夹，您可以在其中找到记录的视频，其文件名以 **Screencast** 开头，并包括录制的日期和时间。

请注意，内置的记录器总是捕获整个屏幕，包括多监视器设置中的所有监视器。

第 15 章 虚拟文件系统和磁盘管理

15.1. GVFS

GVFS (GNOME 虚拟文件系统)是构建 **GNOME** 桌面的库所提供的虚拟文件系统接口的扩展。**GVFS** 提供完整的虚拟文件系统基础架构，并处理 **GNOME** 桌面中的存储。

GVFS 根据 **URI (统一资源标识符)** 标准使用地址进行完全识别，语法上与 **Web** 浏览器中使用的 **URL** 地址类似。这些 **schema://user@server/path** 形式的地址是确定服务类型的关键信息。

15.2. GVFS 的后端

GVFS 中有多个后端，它们提供对特定类型的资源的访问。以下是可用 **GVFS** 后端及其规格的列表：

表 15.1. 可用的后端

| 后端 | 描述 |
|-----------|--|
| afc | 与 MTP (媒体传输协议)类似，公开苹果 iDevice (通过 USB 连接) 上的文件。 |
| afp | 苹果文件协议(AFP)客户端，用于访问 Mac 操作系统 X 和原始 Mac 操作系统的文件服务。 |
| archive | 以只读方式处理各种存档文件 (ZIP、TAR)。 |
| burn | 刻录应用程序使用的虚拟后端，用作新 CD/DVD/BD 介质内容的临时存储。 |
| cdda | 通过单独的波形音频文件格式(WAV)文件公开音频 CD。 |
| computer | 虚拟后端整合活动挂载和物理卷。行为类似于路标。之前，Nautilus 用于其 Computer 视图。 |
| dav, davs | WebDAV 客户端，包括安全变体。仅可以在挂载期间进行身份验证，不支持以后基于每个文件夹重新进行身份验证。 |
| dns-sd | DNS 服务发现 - 在网络浏览期间使用的 Avahi 客户端，为发现的服务形成持久的 URI。 |
| ftp | 全功能的 FTP (文件传输协议) 客户端，暂不支持 FTPS。默认支持被动传输。 |
| gphoto2 | 图片传输协议(PTP)客户端，用于访问通过 USB 或 FireWire 连接的相机。 |
| http | 处理所有 HTTP 请求；对于在客户端应用程序中从 web 轻松下载文件非常有用。 |

| 后端 | 描述 |
|----------|--|
| locatest | 简单的测试后端代理 file:/// URI。；带有错误注入支持。 |
| mtp | 媒体传输协议后端用于访问媒体播放器和智能手机内存。 |
| network | 用于浏览网络，显示附近的 Avahi 和 Samba 服务器。 |
| obexftp | 一个蓝牙客户端。 |
| recent | GtkFileChooser 中使用的后端，用于列出 GNOME 应用程序最近使用的文件。 |
| sftp | 全功能的 SFTP（SSH 文件传输协议）客户端。 |
| smb | 访问 Samba 和 Windows 共享。 |
| trash | 允许恢复已删除的文件的垃圾后端。 |



注意

某些后端被单独打包，默认情况下不安装。要安装其他后端，请使用 `yum` 软件包管理器。

要使用后端的 *服务*，必须形成一个 **URI 字符串**。这个字符串是在 **GVFS** 中使用的基本标识符，它包含了唯一标识所需的所有信息，如服务类型（后端 ID）、绝对路径和用户名（如果需要）。您可以在 **Nautilus** 地址栏中看到此信息，**GTK+** 打开或保存文件对话框。

以下示例是 **URI 字符串** 的非常基本的形式，指向在 `ftp.myserver.net` 域中运行的 **FTP**(文件传输协议) 服务器的根目录(`/`)：

例 15.1. 指向根目录的 URI 字符串

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

以下示例使用身份验证指向指定路径中的文本文件：

例 15.2. 指向文本文件的 URI 字符串

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

15.3. 挂载、卸载和弹出

在虚拟文件系统中，特定的资源被设为自动挂载，但最常见的方法是手动触发挂载。

过程 15.1. 手动挂载

1. 在 Nautilus 中打开一个文件（即 Files 应用程序）。
2. 在屏幕顶部的位置栏中，输入一个格式良好的 URI 字符串。如果没有显示位置栏，请按 **Ctrl+L**。

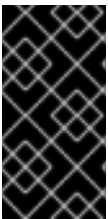
或者，Nautilus 提供 **连接到服务器** 对话框，您可以通过 **转至 文件 → 连接到服务器**。

3. 当系统要求登录凭据时，在相关条目框中输入您的名字和密码。
4. 挂载过程完成后，您可以使用文件。

当您需要卸载资源时，请遵循以下简单的流程。

过程 15.2. 卸载

1. 点击所选挂载上的弹出图标。
2. 等待，直到挂载消失或有关安全移除的通知出现。



重要

出于性能的原因，数据可能会被缓存或者在后台缓慢写入。要在设备或远程资源上安全地传输数据，切勿拔下或断开设备或远程资源的连接。

挂载在应用程序之间共享，并在运行的桌面会话中全局跟踪，这意味着即使您退出了触发挂载的应用程序，它仍可用于任何其他应用程序。同样，多个应用程序可以同时访问挂载，除非其被后端限制（有些协议因为设计问题而只允许一个通道）。



重要

GVFS 挂载（以及物理卷）仅限于其所有者，不允许其他用户利用您的隐私。

15.4. 管理书签

您可以通过为其建立书签来保存对位置的引用。

过程 15.3. 为位置添加书签：

1. 选择您要为其建立书签的文件夹或文件。
2. 按 **Ctrl+D**。

第一次激活书签时，GVFS 子系统会查找现有的挂载，并在不存在时生成新的挂载。这样，您可以在打开或保存对话框中进行身份验证。

书签在 GTK+ 和 GNOME 桌面中良好集成：显示标准 GTK+ 打开或保存对话框的每个应用程序（技术称为 `GtkFileChooser`）都会在对话框左侧面板中列出书签。另外，`Nautilus` 及其克隆会在边栏中显示书签，或者更通用地在 `Files` 菜单中显示书签。



注意

如果您尚未为页面添加书签，则不会显示 **Bookmarks** 标签。

除了 **Bookmarks** 外，所有其他可用的 GVFS 卷和挂载都列在 `GtkFileChooser` 边栏中。有时，一个书签和一个 GVFS 卷合并为一个项目，以防止重复和混淆。然后，书签可以像 GVFS 挂载一样具有弹出图标。

书签位于 `~/.config/gtk-3.0/bookmarks` 文件中。在以下示例中，书签的位置为 `~/Music`、`~/Pictures`、`~/Videos`、`~/Downloads` 和 `~/bin`，因此 `~/.config/gtk-3.0/bookmarks` 文件的内容如下：

例 15.3. `~/.config/gtk-3.0/bookmarks` 文件

```
file:///home/username/Music
```

```
file:///home/username/Pictures
file:///home/username/Videos
file:///home/username/Downloads
file:///home/username/bin
```

使用您要使用的用户名替换 **username**。

过程 15.4. 要编辑书签：

1. 打开顶部栏中的 **Files** 菜单。
2. 单击 **Bookmark** 以打开书签编辑器。

15.5. 配置默认服务器列表

从系统管理员的角度来看，无法一次为所有用户设置一组书签。不过，系统管理员可以允许其用户访问文件共享。

Nautilus 以 **XBEL** 格式在 `~/.config/nautilus/servers` 文件中存储文件共享服务器的列表。可以将文件共享服务器的列表添加到该文件，以便用户可以轻松地访问文件共享。



备注

XBEL (XML 书签交换语言) 是一个可让您共享 **URI** (统一资源标识符) 的 **XML** 标准。在 **GNOME** 中，**XBEL** 用于共享应用程序 (如 **Nautilus**) 中的桌面书签。

在以下示例中，**Nautilus** 使用 `~/.config/nautilus/servers` 文件中的 **URI** `ftp://ftp.gnome.org/` 创建名为 **GNOME FTP** 的书签。

例 15.4. `~/.config/nautilus/servers` 文件

```
<?xml version="1.0" encoding="UTF-8"?>
<xbel version="1.0"
  xmlns:bookmark="http://www.freedesktop.org/standards/desktop-bookmarks"
  xmlns:mime="http://www.freedesktop.org/standards/shared-mime-info">
  <bookmark href="<input>ftp://ftp.gnome.org/</input>">
    <title><input>GNOME FTP</input></title>
  </bookmark>
</xbel>
```

15.6. 向所有其他应用程序公开 GNOME 虚拟文件系统

除了使用 GIO 库构建的应用程序可以访问 GVFS 挂载外，GVFS 还提供公开活跃 GVFS 挂载的 FUSE 守护进程。这意味着，任何应用程序都可以使用标准 POSIX API 访问活跃的 GVFS 挂载，就像它们是常规的文件系统一样。

然而，有一些应用程序，它们中的其他库依赖项和新的 VFS 子系统的细节可能不合适或过于复杂。因此，为了提高兼容性，GVFS 提供了一个 FUSE (用户空间中的文件系统) 守护进程，它通过挂载来公开活跃的挂载，以进行标准 POSIX (可移植操作系统接口) 访问。此守护进程会以透明的方式转换传入的请求，来为应用程序模拟本地文件系统。



重要

来自不同设计的翻译与 100% 功能不兼容，您可能会遇到某些应用程序和 GVFS 后端的某些组合。

FUSE 守护进程使用 GVFS 主守护进程自动启动，并将其挂载放在 `/run/user/UID/gvfs` 或 `~/.gvfs` 文件中作为回退。手动浏览显示每个 GVFS 挂载都有单独的目录。当您使用非原生应用程序从 GVFS 位置打开文档时，转换的路径将作为参数传递。请注意，原生 GIO 应用程序会自动将此路径转换为原生 URI。

15.7. GVFS 挂载的密码管理

典型的 GVFS 挂载在其激活时请求凭证，除非资源允许匿名身份验证，或者根本不需要任何凭证。如标准 GTK+ 对话框中显示的，用户可以选择密码是否应保存或不保存。

过程 15.5. 示例：经过身份验证的挂载进程

1. 按 **Ctrl+L** 打开 Files 并激活地址栏。
2. 输入需要身份验证的服务的格式良好的 URI 字符串 (例如 `sftp://localhost/`) 。
3. 此时会显示凭证对话框，要求输入用户名、密码和密码存储选项。

4.

填写凭证并确认：

如果选择了持久性存储，密码会保存在用户 **keyring** 中。**GNOME Keyring** 是 **secret** 存储的中心位置。它是加密的，默认情况下，它会使用登录时提供的密码自动解锁桌面会话启动。如果它被其他密码保护，则在第一次使用时设置密码。

为了管理存储的密码和 **GNOME Keyring** 本身，会提供 **Seahorse** 应用程序。它允许删除单个记录或更改密码。有关 **Seahorse** 的更多信息，请参阅直接嵌入在桌面中的 **Seahorse** 帮助手册。

15.8. GNOME 中的 GVFS 工具和 XDG-UTILS

GVFS 附带一些可能对脚本或测试很有用的命令。提供了一组对应的 **POSIX** 命令：

- **gvfs-cat**
- **gvfs-less**
- **gvfs-mkdir**
- **gvfs-mount**
- **gvfs-rename**
- **gvfs-set-attribute**
- **gvfs-copy**
- **gvfs-ls**
- **gvfs-move**

- **gvfs-rm**
- **gvfs-trash**
- **gvfs-info**
- **gvfs-save**
- **gvfs-tree**

提供了一些额外的命令来更好地控制 GVFS 细节：

- **gvfs-monitor-dir**
- **gvfs-monitor-file**
- **gvfs-mime**
- **gvfs-open**

所有这些命令都是原生的 GIO 客户端，不需要运行回退 FUSE 守护进程。它们的目的是不是代替 POSIX 命令，实际上，支持的切换范围极小。在其基本形式中，URI 字符串（而不是本地路径）被用作参数。

这一切都允许 GNOME 在 xdg-tools（一个freedesktop.org 的互操作性项目）中得到很好的支持。例如，当检测到 GNOME 会话运行时，常用的 xdg-open 实际上会调用 gvfs-open，从正确的位置读取文件类型关联。

以下是 GVFS 命令用法的几个示例：

- 要列出本地文件系统上 /tmp 中的所有文件，请执行：

```
$ gvfs-ls file:///tmp
```

- 以下命令列出远程机器上文本文件的内容：

```
$ gvfs-cat ssh://joe@ftp.myserver.net/home/joe/todo.txt
```

- 要将引用的文本文件复制到本地 /tmp 目录中，请运行：

```
$ gvfs-copy ssh://joe@ftp.myserver.net/home/joe/todo.txt /tmp/
```



注意

为方便起见，**bash** 完成作为软件包的一部分提供。

15.9. 使用 GVFS 元数据

GVFS 将其元数据存储实施为绑定到特定文件的一组简单键/值对。因此，用户或应用程序有一个工具可用来保存为运行时信息设计的小型数据，如图标位置、最近播放的位置、文档中的位置、标志、注释等。

每当移动文件或目录时，会相应地移动元数据，使其保持与对应文件的连接。**GVFS** 私有地存储所有元数据，因此仅在机器上可用。但是，**GVFS** 挂载和可移动介质也被跟踪。



注意

可移动介质现在挂载到 `/run/media/` 而不是 `/media` 目录中。

要查看和处理元数据，您可以使用：

- **gvfs-info** 命令；

- **gvfs-set-attribute** 命令;
- 或者任何其他与属性一起工作的原生 GIO 方式。

在以下示例中，设置了自定义的元数据属性。请注意移动或重命名后特定的 **gvfs-info** 调用和数据持久性之间的区别（请注意 **gvfs-info** 命令输出）：

例 15.5. 设置自定义的元数据属性

```
$ touch /tmp/myfile
$ gvfs-info -a 'metadata::*' /tmp/myfile
attributes:
$ gvfs-set-attribute -t string /tmp/myfile 'metadata::mynote' 'Please remember to delete this file!'
$ gvfs-info -a 'metadata::*' /tmp/myfile
attributes:
  metadata::mynote: Please remember to delete this file!
$ gvfs-move /tmp/myfile /tmp/newfile
$ gvfs-info -a 'metadata::*' /tmp/newfile
attributes:
  metadata::mynote: Please remember to delete this file!
```

15.10. 故障排除

15.10.1. 活动状态的 VFS 挂载不可见

如果您的活动状态的 VFS 挂载不可见，这意味着您的应用程序不是原生的 GIO 客户端。原生的 GIO 客户端通常是使用 GNOME 库 (glib、gio) 的所有 GNOME 应用程序。有一个服务 **gvfs-fuse**，作为非 GIO 客户端的回退提供。

要查找活动但不可见的 VFS 挂载的原因，请检查 **gvfs-fuse** 进程是否在运行。因为 **gvfs-fuse** 会自动运行，因此不建议自己启动它，请尝试退出登录，并作为第一个选项登录。

另外，您可以在终端中手动启动 VFS 兼容性挂载：

1. 通过运行 **id** 命令(**gvfsd-fuse** 守护进程需要一个路径)，查找 **/run/user/UID /gvfs/** 路径

的UID（系统用户 ID）。

或者，当 `/run/user/UID/gvfs/` 路径不可用时，`gvfsd-fuse` 会使用主目录中的 `.gvfs` 路径。

2. 运行 `/usr/libexec/gvfsd-fuse -f /run/user/UID/gvfs` 命令启动 `gvfsd-fuse` 守护进程。
3. 现在，VFS 挂载可用，您可以手动浏览应用程序中的路径。

15.10.2. 连接的 USB 磁盘不可见

在某些情况下，当您连接一个闪存驱动器时，GNOME 桌面可能不会显示它。如果驱动器不可见，这意味着：

- 您不能在 `Disks` 应用程序中看到该设备。
- 您已运行 `udisksctl dump` 命令，它会列出 `udisks` 守护进程的当前状态，并显示有关所有对象的信息，但您的闪存驱动器不在其中。
- 您已运行 `dmesg` 命令。在日志的末尾，有与 USB 设备检测相关的消息，以及检测到的分区的列表，但您的闪存驱动器不在其中。

如果您的闪存驱动器不可见，您可以尝试在 `Disks` 中设置 `Show in user interface` 标志：

1. 按 `Super` 键打开 `Disks`，以进入 `Activities Overview`，输入 `Disks`，然后按 `Enter` 键。
2. 在 `Volumes` 操作菜单中，点 `Edit Mount Options..`
3. 点 `Show in user interface`。
4. 单击 `OK` 进行确认。

如果闪存驱动器仍然不可见，您可以尝试删除该驱动器，并尝试再次连接它。

有关存储的更多信息，请参阅 [存储管理指南](#)。

15.10.3. Nautilus 显示未知或不需要的分区

检查设备是否在 `/etc/fstab` 文件中列出，因为默认情况下设备不会在用户界面中显示。`/etc/fstab` 文件通常列出了要在操作系统中使用的磁盘分区，并指示它们的挂载方式。某些挂载选项可能会允许或阻止在用户界面中显示卷。

隐藏卷的解决方案之一是在 `Disks` 应用程序中的 `Mount Options` 窗口中取消选择 `Show in user interface`：

1. 按 `Super` 键打开 `Disks`，以进入 `Activities Overview`，输入 `Disks`，然后按 `Enter` 键。
2. 在 `Volumes` 操作菜单中，点 `Edit Mount Options...`
3. 取消选中 `Show in user interface`，然后单击 `OK` 进行确认。

15.10.4. 到远程文件系统的连接不可用

在有些情况下，客户端意外地、不情愿地与虚拟文件系统（或远程磁盘）挂载断开了连接，之后未自动重新连接，并返回错误消息。几个原因会触发这些情况：

- 连接中断（例如，您的笔记本电脑与 `Wi-Fi` 的连接断开了）。
- 用户在一段时间内不活跃，被服务器断开连接（空闲超时）。
- 计算机从休眠模式恢复。

解决方案是卸载并再次挂载文件系统，这样可重新连接资源。



注意

如果连接被频繁禁用，请检查 GNOME Settings 中 Network 面板中的设置。

15.10.5. 如果磁盘忙，该怎么办？

如果您收到有关磁盘非常忙碌的通知，请确定正在访问该磁盘的程序。然后，您可以定期结束您正在运行的程序。或者，您可以使用 System Monitor 来强制终止程序。

在何处以及如何查看系统进程？

- 运行 `lsdf` 命令以获取与进程一起打开文件的列表。如果 `lsdf` 不可用，请运行 `ps ax` 命令，该命令还提供正在运行的进程的列表。
- 或者，您可以使用 System Monitor 应用程序在 GUI 中显示正在运行的进程。
- 运行以下命令确保您已安装了 `iostat`：

```
# yum install iostat
```

然后以 `root` 用户身份运行 `iostat` 来查看系统进程。

当您确定程序后，请按如下所示终止或杀死程序：

- 在命令行中执行 `kill` 命令。
- 在 System Monitor 中，右键单击带有程序进程名称的行，然后单击 `End Process` 或 `Kill Process` 下拉菜单项。

第 16 章 硬件配置

16.1. 平板电脑

16.1.1. 为新平板添加支持

libwacom 是一个平板电脑信息客户端库，用于存储有关 Wacom 模型的数据。GNOME 中的 **gnome-settings-daemon** 组件和 Wacom Tablet 设置面板都使用这个库。

要将对新平板电脑的支持添加到 **libwacom** 中，必须创建一个新的平板电脑定义文件。平板电脑定义文件包含在 **libwacom-data** 软件包中。如果安装了此软件包，则平板电脑定义文件会在 **/usr/share/libwacom/** 目录中本地可用。

要正确使用屏幕映射，对平板电脑的支持必须包含在 **libwacom** 数据库和 **udev** 规则文件中。



重要

libwacom 不支持设备的一个常见指示是它通常在 GNOME 会话中工作，但设备没有正确映射到屏幕上。

过程 16.1. 如何添加平板电脑描述

1. 使用 **libwacom-list-local-devices** 工具列出 **libwacom** 识别的所有本地设备。

如果您的设备没有被列出，但可作为内核中的事件设备（请参阅 **/proc/bus/input/devices**）和 X 会话（请参阅 **xinput list**），则 **libwacom** 的数据库中缺少该设备。

2. 创建一个新的平板电脑定义文件。使用以下的 **data/wacom.example**，并编辑相应的行。



注意

新的 **.tablet** 文件可能已经提供，因此先在 <https://sourceforge.net/p/linuxwacom/libwacom/ci/master/tree/> 检查上游存储库。如果在列表中找到了您的平板电脑型号，只需将该文件复制到本地机器就可。

```
# 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*

3.

添加和安装以 .tablet 为后缀的新文件：


```
cp the-new-file.tablet /usr/share/libwacom/
```

安装后，平板电脑是 `libwacom` 的数据库的一部分。然后，可通过 `libwacom-list-local-devices` 提供平板电脑。

4.

创建包含以下内容的新文件 `/etc/udev/rules/99-libwacom-override.rules`，以便您的设置不会被覆盖：

```
ACTION!="add|change", GOTO="libwacom_end"
KERNEL!="event[0-9]*", GOTO="libwacom_end"

[new tablet match entries go here]

LABEL="libwacom_end"
```

5.

重启您的系统。

16.1.2. Wacom 平板电脑配置存储在哪里？

您的 Wacom 平板电脑的配置保存在 `GSettings` 的 `/org/gnome/settings-daemon/peripherals/wacom/machine-id-device-id` 键中，其中 `machine-id` 是一个 D-Bus 机器 ID，而 `device-id` 是一个平板电脑设备 ID。平板电脑的配置模式是 `org.gnome.settings-daemon.peripherals.wacom`。

同样，手写笔配置存储在 `/org/gnome/settings-daemon/peripherals/wacom/device-id/tool-id` 键中，其中 `tool-id` 是用于专业范畴的手写笔的标识符。对于不支持 `tool-id` 的消费者范畴，改为使用通用标识符。手写笔的配置模式是 `org.gnome.settings-daemon.peripherals.wacom.stylus`，橡皮擦的配置模式是 `org.gnome.settings-daemon.peripherals.wacom.eraser`。

要获得特定机器上使用的平板电脑配置路径的完整列表，您可以使用 `gsd-list-wacom` 工具，该工具由 `gnome-settings-daemon-devel` 软件包提供。

要验证系统中是否安装了 `gnome-settings-daemon-devel` 软件包，请确定系统已订阅 `Optional` 频道，并运行以下命令：

```
# yum install gnome-settings-daemon-devel
```

要了解如何将系统订阅到 `Optional` 频道，请阅读以下资源：

-

<https://access.redhat.com/solutions/392003>

验证安装了该软件包后，请运行以下命令：

```
$ /usr/libexec/gsd-list-wacom
```

请注意，在配置路径中使用 `machine-id`、`device-id` 和 `tool-id` 允许按机器与独立的平板电脑配置共享主目录。

16.1.3. 在计算机之间共享主目录时，Wacom 设置仅适用于一台机器

这是因为您的 Wacom 平板电脑的 D-Bus 机器 ID(`machine-id`)包含在 `/org/gnome/settings-daemon/peripherals/wacom/machine-id-device-id` GSettings 键的配置路径中，该键存储了您的平板电脑设置。

附录 A. KDE PLASMA 工作区

作为默认的 GNOME 桌面环境的替代方案，Red Hat Enterprise Linux 7 提供了 KDE Plasma 工作区（以前称为 K 桌面环境）的版本 4，以匹配不同的工作风格和首选项。

在安装过程中将 KDE Plasma 工作区 设置为默认桌面，或将您当前的桌面环境改为 KDE Plasma 工作区，请参阅 [Red Hat Enterprise Linux 7 安装指南](#)。有关 KDE Plasma 工作区的更多信息，请参阅其上游网站，如 <https://www.kde.org/> 和 <https://docs.kde.org/>。

附录 B. 访问红帽文档

B.1. 产品文档

位于 <https://access.redhat.com/site/documentation/> 处的红帽产品文档充当中央信息资源。目前已被翻译成 22 种语言，对于每种产品，它都以 HTML、PDF 和 EPUB 格式提供了不同种类的书籍，从发布和技术说明到安装、用户和参考指南。

以下是一个与本书直接或间接相关的简要的文档列表：

- **Red Hat Enterprise Linux 7 系统管理员指南**，可从 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html-single/System_Administrators_Guide/index.html 获得，包含有关各种系统组件的详细信息，例如 GRUB 2 引导装载程序、软件包管理、systemd 或打印机配置。
- **Red Hat Enterprise Linux 7 安装指南**，可从 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/index.html 获得，包含有关安装 Red Hat Enterprise Linux 7 和使用 Anaconda 安装程序的详细信息。
- **Red Hat Enterprise Linux 7 迁移规划指南**，可从 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html-single/Migration_Planning_Guide/index.html 获得，包含了 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 之间在行为和兼容性方面的重要变化的概述。迁移规划指南还介绍了红帽提供的工具，以帮助升级到 Red Hat Enterprise Linux 7。
- **Red Hat Enterprise Linux 7 网络指南**，可从 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html-single/Networking_Guide/index.html 中获得，包含了有关配置和管理 Red Hat Enterprise Linux 7 网络的信息。
- **Red Hat Enterprise Linux 7 虚拟化部署和管理指南**，可从 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/ 中获得，包含了安装、配置和管理 Red Hat Enterprise Linux 虚拟化的信息。

B.2. RED HAT ACCESS GUI

另一个强烈推荐的信息源是一个桌面应用程序 Red Hat Access GUI，它可让您使用红帽知识库、资源

和功能查找帮助、回答和使用诊断服务。如果您在 [红帽客户门户网站](#) 中有一个有效的帐户，您可以通过关键字访问其它信息，并轻松浏览知识库的提示。如果您选择安装了 GNOME 桌面，则 Red Hat Access GUI 已安装。

有关这个工具的好处、安装和用法的更多信息，请参阅 [Red Hat Access GUI](#)。

附录 C. 致谢

这个文本的某些部分首先出现在 [GNOME 桌面系统管理指南](#) 中。版权所有 © 2014 GNOME Project, Micha, Micha, Jeremy Bicha, Ekaterina Gerasimova, minnie_eg, Aruna Sankaranarayan, Sindhu S, Shobha Tyagi, Shaun McCance, David King 等。根据 [Creative Commons Attribution-ShareAlike 3.0 Unported License](#) 授权。

Red Hat Enterprise Linux 7 桌面迁移和管理指南 的编辑非常感谢 GNOME 社区成员对 GNOME 桌面系统管理指南 做出的宝贵贡献。

附录 D. 修订历史记录

| | | |
|--|------------------------|-------------------------|
| 修订 0.1-95 7.7 GA 发布版本。 | Mon Aug 05 2018 | Marie Doleželová |
| 修订 0.1-87 7.4 GA 发布版本。 | Wed Jul 26 2017 | Jana Heves |
| 修订 0.1-86 7.3 GA 发布版本。 | Mon Oct 17 2016 | Marie Doleželová |
| 修订 0.1-85 7.3 GA 发布版本。 | Mon Oct 17 2016 | Marie Doleželová |
| 修订 0.1-84 本书的 Red Hat Enterprise Linux 7.3 Beta 版本。 | Wed 17 Aug 2016 | Marie Doleželová |
| 修订 0.1-83 反映了 GNOME rebase 到版本 3.14 的重新发布。 | Fri 01 Apr 2016 | Jana Heves |
| 修订 0.1-82 本书的 Red Hat Enterprise Linux 7.2 发行版。 | Wed 11 Nov 2015 | Jana Heves |
| 修订 0.1-80 添加了对 Red Hat Enterprise Linux 7 的 Anaconda 自定义指南的链接。 | Wed 24 Jun 2015 | Petr Kovář |
| 修订 0.1-78 添加了 <i>允许和禁止在线账户</i> 的部分。 | Tue 07 Apr 2015 | Petr Kovář |
| 修订 0.1-76 本书的 Red Hat Enterprise Linux 7.1 发行版。 | Tue 17 Feb 2015 | Petr Kovář |
| 修订 0.1-74 本书的 Red Hat Enterprise Linux 7.1 Beta 发行版。 | Thu 04 Dec 2014 | Petr Kovář |
| 修订 0.1-69 本书的 Red Hat Enterprise Linux 7.0 发行版。 | Mon 02 Jun 2014 | Petr Kovář |
| 修订 0.1-35 本书的 Red Hat Enterprise Linux 7.0 Beta 发行版。 | Wed 11 Dec 2013 | Petr Kovář |