



# **GNOME 2.14 Desktop System Administration Guide**

**GNOME**

# GNOME 2.14 Desktop System Administration Guide

by Brent Smith and Sun GNOME Documentation Team

Copyright © 2006 Brent Smith

Copyright © 2005 Brent Smith

Copyright © 2004 Sun Microsystems

Copyright © 2003 Sun Microsystems

## Abstract

The GNOME System Administration Guide provides information to administrators on how to configure and manage different aspects of the desktop environment for users.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License (GFDL), Version 1.1 or any later version published by the Free Software Foundation with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. You can find a copy of the GFDL at this link [[ghelp:fdl](http://ghelp:fdl)] or in the file COPYING-DOCS distributed with this manual.

This manual is part of a collection of GNOME manuals distributed under the GFDL. If you want to distribute this manual separately from the collection, you can do so by adding a copy of the license to the manual, as described in section 6 of the license.

Many of the names used by companies to distinguish their products and services are claimed as trademarks. Where those names appear in any GNOME documentation, and the members of the GNOME Documentation Project are made aware of those trademarks, then the names are in capital letters or initial capital letters.

DOCUMENT AND MODIFIED VERSIONS OF THE DOCUMENT ARE PROVIDED UNDER THE TERMS OF THE GNU FREE DOCUMENTATION LICENSE WITH THE FURTHER UNDERSTANDING THAT:

1. DOCUMENT IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS FREE OF DEFECTS MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY, ACCURACY, AND PERFORMANCE OF THE DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS WITH YOU. SHOULD ANY DOCUMENT OR MODIFIED VERSION PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL WRITER, AUTHOR OR ANY CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER; AND
2. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER IN TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE AUTHOR, INITIAL WRITER, ANY CONTRIBUTOR, OR ANY DISTRIBUTOR OF THE DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER DAMAGES OR LOSSES ARISING OUT OF OR RELATING TO USE OF THE DOCUMENT AND MODIFIED VERSIONS OF THE DOCUMENT, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES.

## Feedback

To report a bug or make a suggestion regarding the GNOME Desktop or this manual, follow the directions in the GNOME Feedback Page [[ghelp:gnome-feedback](http://ghelp:gnome-feedback)].

GNOME



## Table of Contents

1. Using GConf .....	2
Introduction to GConf .....	2
GConf Repository .....	2
GConf Configuration Sources .....	3
GConf Schemas .....	4
GConf Schema Definition Files .....	5
GConf Daemon .....	5
GConf Command Line Tool .....	6
To Set Preference Values .....	8
Setting General Preferences .....	8
To Set HTTP Proxy Preferences .....	8
To Set Print Manager Preferences .....	9
To Set the Number of Workspaces .....	9
To Set Keyboard Accessibility Preferences .....	9
To Set Keyboard Shortcut Preferences .....	10
Setting Panel and Panel Object Preferences .....	10
Setting Look-and-Feel Preferences .....	13
To Set Font Preferences .....	13
To Set Background Preferences .....	13
To Set Splash Image Preferences .....	13
To Restore Default Preference Values .....	14
2. Customizing Menus .....	16
Introduction to Menus .....	16
Menu Definition Files .....	16
Desktop Entry Files .....	18
Directory Entry Files .....	20
Editing System Menus .....	20
Adding Menus .....	20
Adding an Item to a Menu .....	21
Editing the Properties of a Menu .....	21
Editing a Menu Item .....	21
Deleting an Item from a Menu .....	21
Editing User Menus and Menu Merging .....	22
Merging the System Menu .....	22
Merging Arbitrary Menus .....	23
3. Installing Themes .....	25
Introduction to Themes .....	25
Theme Index File .....	25
To Install a New Controls Option .....	26
To Install a New Window Frame Option .....	26
To Install a New Icons Option .....	27
Installing Icons for Themes .....	27
To Create a Custom Controls Option .....	28
4. Configuring Fonts .....	30
Introduction to Configuring Fonts .....	30
Font Substitution .....	30
To Add a Font for All Users .....	30
To Add a Font for an Individual User .....	30
5. MIME Types .....	33
Introduction to MIME Types .....	33
The MIME Database .....	33
Location .....	34
Contents .....	34

Refreshing the MIME Database .....	35
The source XML files .....	36
Modifying MIME types .....	37
Adding or Modifying MIME types .....	38
Verifying Changes .....	38
application/x-newtype Example .....	38
Registering Applications for MIME Types .....	39
Adding an Application to the GNOME Desktop .....	40
6. Setting Screensavers .....	42
Introduction to Screensavers .....	42
Setting Screensaver Preferences .....	42
Modifying Screensaver Displays .....	42
7. Session Management .....	45
Introduction to Sessions .....	45
Setting Session Defaults .....	45
8. Help System .....	47
Introduction to the Help System .....	47
OMF Files .....	47
ScrollKeeper Cataloging System .....	47
9. Improving Performance .....	50
Introduction to Improving Performance .....	50
Reducing CPU Usage .....	50
To Use Theme Options Which Require Less CPU Resources .....	50
To Reduce CPU Usage by Turning Off Display of Icons in Menus .....	51
To Reduce CPU Usage by Turning Off the Splash Screen .....	51
To Reduce CPU Usage by Turning Off Panel Animation .....	51
Improving File Manager Performance .....	51
Reducing X Window System Network Traffic .....	53
To Use Theme Options Which Create Less Network Traffic .....	53
To Reduce Network Traffic by Turning Off Display of Icons in Menus .....	54
Reducing Color Usage and Improving Display Quality .....	54
To Use Theme Options That Use the Websafe Color Palette .....	54
To Reduce Color Usage by Turning Off Display of Icons in Menus .....	55
To Reduce Color Usage by Turning Off the Splash Screen .....	55
To Reduce Color Usage by Using a Solid Color for the Background .....	55
10. Disabling GNOME Desktop Features .....	57
Introduction to Disabling GNOME Desktop Features .....	57
To Disable Lock Screen and Log Out .....	57
To Disable Command Line Operations .....	57
To Disable Panel Configuration .....	58
A. Hidden Directories .....	59
B. Security Note on Configuring SMB Printers .....	61
Glossary .....	62
Index .....	64

## List of Tables

2.1. Menu Definition File Elements .....	17
2.2. Desktop Entry Keys .....	19
2.3. Directory Entry Keys .....	20
3.1. Keys in Theme Index Files .....	25
5.1. Child elements of <mime-info> .....	36
A.1. Hidden Directories in User Home Directories .....	59

## List of Examples

2.1. Example of a <code>.menu</code> file .....	17
2.2. Deleting an Item from a Menu .....	22
2.3. Merging the System Menu .....	23
5.1. Example: <code>/usr/share/mime/text/plain.xml</code> file .....	34
5.2. Example of a source XML file: <code>diff.xml</code> .....	37
5.3. <code>Overrides.xml</code> file .....	39
5.4. A User's <code>defaults.list</code> file .....	39

# Chapter 1

---

## Using GConf



The information in this chapter describes how to use GConf to manage user preferences. This chapter also describes how to use Configuration Editor.

## Introduction to GConf

GConf simplifies the administration of preferences for GNOME Desktop users. GConf enables system administrators to do the following:

- Set mandatory values for particular preferences for all users. In this way, system administrators can control whether users can update particular preferences.
- Set default values for particular preferences for all users.
- Use suggested values for preferences that are specified in definition files for the preferences.
- Read documentation on each preference.

GConf also notifies applications when a preference value changes, locally or across a network. In this way, when you change a preference, all applications that use the preference are immediately updated.

GConf has the following components:

- A repository of user preferences.
- A daemon, **gconfd-2**.
- A command line tool, **gconftool-2**.

## GConf Repository

Each preference in the GConf repository is expressed as a key-value pair. A *GConf preference key* is an element in the repository that corresponds to an application preference. For example, the `/apps/gnome-session/options/show_splash_screen` preference key corresponds to the Show splash screen on login option in the Sessions preference tool. The GNOME Desktop user interface does not contain all of the preference keys in the GConf repository. For example, the Panel preference tool does not contain an option that corresponds to the `/apps/panel/global/tooltips_enabled` key.

The repository is structured like a simple hierarchical file system. The repository contains the following:

- Directories that correspond to applications that use the GConf repository. For example, the file system contains the directory `/apps/metacity`.
- Subdirectories that correspond to categories of preferences. For example, the file system contains the directory `/apps/metacity/general`.
- Special files that list the preference keys in the directory, and contain information about the keys. For example, a file that contains information about the keys that relate to the HTTP proxy preferences is in the directory `/system/http_proxy`.
- A `/schemas` directory that contains files that describe all of the preference keys.

Preference keys typically have simple values such as strings, integers, or lists of strings and integers. The format of the preference key in the repository depends on the backend module that is used to read the repository. The following is an example of the `/desktop/gnome/interface/font_name` preference key when an Extensible Markup Language (XML) backend module is used to read the repository:

```
<entry name="font_name" mtime="1038323555" muser="user123" type="string">
<stringvalue>Sans 10</stringvalue></entry>
```



When this guide refers to a preference key, the path to the key is added to the name of the key. For example, the `font_name` preference key in the `/desktop/gnome/interface` subdirectory is referred to as `/desktop/gnome/interface/font_name`.

## GConf Configuration Sources

The GConf repository contains a series of storage locations that are called *configuration sources*. The configuration sources are listed in the *GConf path file*. The location of the GConf path file is `/etc/gconf/gconf-version-number/path`. Each user has a path file. The path file specifies the following information for each configuration source:

- The backend module to use to read the repository.
- The permissions on the repository.
- The location of the repository.

The GConf path file also contains `include` instructions. By default, the contents of the GConf path file are as follows:

```
xml:readonly:/etc/gconf/gconf.xml.mandatory
include /etc/gconf/2/local-mandatory.path
include "${HOME}/.gconf.path"
include /etc/gconf/2/local-defaults.path
xml:readwrite:${HOME}/.gconf
xml:readonly:/etc/gconf/gconf.xml.defaults
```

When GConf searches for a preference value, GConf reads the configuration sources in the order specified in the path file. The following table describes the configuration sources in the path file:

Configuration Source	Description
Mandatory	The permissions on this configuration source are set to read only. Users cannot overwrite the values in this source, so the preferences in the source are mandatory.
User	This configuration source is stored in the <code>.gconf</code> directory in the home directory of the user. When the user sets a preference, the new preference information is added to this location.  You can use the Configuration Editor to modify the user configuration source.
Default	This configuration source contains the default preference settings.

The sequence of the configuration sources in the path file ensures that mandatory preference settings override user preference settings. The sequence also ensures that user preference settings override default preference settings. That is, GConf applies preferences in the following order of priority:

1. Mandatory preferences
2. User-specified preferences
3. Default preferences

The `include` instructions in the GConf path file enable system administrators to specify other configuration sources.

Included Configuration Source	Description
<code>/etc/gconf/2/local-mandatory.path</code>	Use this configuration source to store mandatory preference values for a particular system.
<code>\$(HOME)/.gconf.path</code>	The user specifies the location of the configuration source in the home directory, in a file that is called <code>.gconf.path</code> .
<code>/etc/gconf/2/local-defaults.path</code>	Use this configuration source to store default preference values for a particular system.

## GConf Schemas

A *GConf schema* is a collective term for a *GConf schema key* and a *GConf schema object*. The following table describes schema keys and schema objects and the relationship of these items to preference keys:

Item	Description
Preference key	An element in the GConf repository that corresponds to an application preference.
Schema key	A key that stores a schema object for a preference key.
Schema object	An element in a configuration source that contains information for a preference key, such as the following: <ul style="list-style-type: none"> <li>• The name of the application that uses the preference key.</li> <li>• The type of value required for the preference key, for example integer, boolean, and so on.</li> <li>• A default value for the preference key.</li> <li>• Brief documentation on the preference key.</li> </ul>

The following table gives examples of a preference key, a schema key, and a schema object:

Item	Example
Preference key	<code>/desktop/gnome/interface/font_name</code>
Schema key	<code>/schemas/desktop/gnome/interface/font_name</code>
Schema object	<pre>&lt;schema&gt;   &lt;applyto&gt;/desktop/gnome/interface/font_name&lt;/applyto&gt;   &lt;key&gt;/schemas/desktop/gnome/interface/font_name&lt;/key&gt;   &lt;owner&gt;gnome&lt;/owner&gt;   &lt;type&gt;string&lt;/type&gt;   &lt;default&gt;Sans 10&lt;/default&gt;   &lt;locale name="C"&gt;     &lt;short&gt;Default font&lt;/short&gt;     &lt;long&gt;Name of the default font used by gtk+.&lt;/long&gt;   &lt;/locale&gt; &lt;/schema&gt;</pre>

You can associate a schema key with a preference key. For example, the following `/desktop/gnome/interface/font_name` key includes a schema key:

```
<entry name="font_name" mtime="1034873859"
schema="/schemas/desktop/gnome/interface/font_name" />
```

When you associate a schema key with a preference key, the preference uses the suggested value that is specified in the schema object of the schema key. The suggested value is contained in the `<default>` element in the schema object. By default, all the preference keys in the default configuration source are associated with schema keys.

Typically, schemas are stored in the default configuration source.

## GConf Schema Definition Files

Schemas are generated from *schema definition files*. A schema definition file defines the characteristics of all of the keys in a particular application. Schema definition files have a `.schemas` extension.

The schema definition files are included in the `/etc/gconf/schemas` directory. You can use the schema definition files to create a new configuration source.

Some schema definition files correspond closely to a part of the GNOME Desktop user interface. For example, `system_http_proxy.schemas` corresponds to the Network Proxy preference tool. Other schema definition files contain preference keys that are not present in the GNOME Desktop user interface. For example, the `/apps/panel/global/tooltips_enabled` key is not present in the user interface.

Some parts of the GNOME Desktop user interface contain preferences that represent preference keys from more than one schema definition file. For example, the Keyboard Shortcuts preference tool contains preferences that represent keys from the `panel-global-config.schemas` and `metacity.schemas` files.

## GConf Daemon

The GConf daemon is called **gconfd-2**. The GConf daemon notifies applications when a preference value changes. For example, you might select to show only icons in toolbars in the Menus & Toolbars preference tool. When you select this option in the preference tool, the toolbars on all open applications are updated instantly. The GConf daemon can operate locally, or across a network.

An instance of the GConf daemon is started for each user. The GConf daemon does not have to deal with complex problems such as authentication and data security. When the GConf daemon starts, the daemon loads the GConf path file. The GConf daemon manages all access between applications and the configuration sources.

When an application requests the value of a preference key, the daemon searches the configuration sources as follows:

1. Search for the value of the preference key in each configuration source, in the order specified in the path file. If the value is found, return the value.
2. If a value is not found, search for the schema key that corresponds to the preference key in each configuration source, in the order specified in the path file.
3. If the schema key is found, check the value of the schema key.
4. If the value of the schema key is a schema object, return the suggested value in the `<default>` element of the schema object.

The GConf daemon also caches preference key values. All applications use this cache, so applications only need to access the configuration sources once.

To terminate the GConf daemon, run the following command:

```
gconftool-2 --shutdown
```

## GConf Command Line Tool

GConf includes a command line tool, **gconftool-2**. You can use the **gconftool-2** command to perform the following tasks:

- Set the values of keys.
- Display the values of keys.
- Install schemas from schema definition files when you install an application.

For example, use the following command to display the values of all keys in the `/desktop/gnome` directory and subdirectories.

```
gconftool-2 --recursive-list /desktop/gnome
```

Below are listed some of the options that you can use with the **gconftool-2** command. For detailed command line options, please see `man:gconftool-2(1)`.

### **gconftool-2** Command Options

`--all-dirs`

Lists all subdirectories in a directory that you specify.

`--all-entries`

Displays the values of all keys in a directory that you specify.

`--config-source=configuration-source`

Use this option with the `--direct` option to specify a configuration source to use. If you do not specify a configuration source with this option, the command runs on all configuration sources in the path file.

`--direct`

Use this option with the `--config-source` option to access a configuration source directly. When you use this option, GConf bypasses the server. Ensure that the GConf daemon, **gconfd-2**, is not running before you use this option.

`--dump`

Generates a list that contains all preference keys in a GConf repository directory that you specify. The list contains XML descriptions of all the keys. The list is contained in a `<gconfentryfile>` element.

For example, you can redirect the output from this option to generate a file that lists all keys that are related to your panel configuration. You can use the `--load` option with this file.

`--get`

Displays the value of a preference key that you specify. Also displays the values of the elements in the schema object for a schema key that you specify.

`--help`

Displays a help message about the **gconftool-2** command, and the options that you can use with the **gconftool-2** command.

`--load=filename`

Use this option to sets the values of preference keys in the current directory in a configuration source to the values in the file that you specify. The file that you specify must contain XML descriptions of the keys, in a `<gconfentryfile>` element.

`--long-desc=description`

Use this option with the `--set-schema` option to specify a long description for a schema key.

`--makefile-install-rule`

Installs schema definition files to applications.

`--owner=owner`

Use this option with the `--set-schema` option to specify an owner for a schema key.

`--recursive-list`

Displays the values of all preference keys in all subdirectories in a directory that you specify.

`--recursive-unset`

Resets the values of all preference keys, in all subdirectories in a directory, from the user setting to the setting in the default configuration source.

`--set`

Sets the value of a preference key, and writes the value to the user configuration source. Use the `--type` option with the `--set` option to specify the data type of the value that you want to set. For example, the following command sets the value of the `/apps/gnome-terminal/profiles/Default/background_color` key in the user configuration source:

```
gconftool-2 \
  --set "/apps/gnome-terminal/profiles/Default/background_color" \
  --type string "#000000"
```

You can also use the `--direct` option and the `--config-source` option with the `--set` option to write a value to another configuration source.

`--set-schema`

Sets the value of an attribute in a schema key, and writes the value to the default configuration source.

Use the following options with the `--set-schema` option to specify the attribute that you want to update:

- `--type`
- `--short-desc`
- `--long-desc`
- `--owner`

For example, the following command sets the short description in the schema key for the `/apps/gnome-terminal/profiles/Default/background_color` key:

```
gconftool-2 \
  --set-schema "/schemas/apps/gnome-terminal/profiles/Default/background_color" \
  --short-desc "Default background color of terminal"
```

`--short-desc=description`

Use this option with the `--set-schema` option to specify a short description for a schema key.

`--shutdown`

Terminates the GConf daemon.

`--type=data-type`

Use this option to specify the data type when you set a value of a preference key. You can also use this option when you set the value of an attribute in a schema key. The following is a list of valid data types:

- `bool`
- `float`
- `int`
- `list`
- `pair`
- `string`

`--unset`

Resets the value of a preference key from the user setting to the setting in the default configuration source.

`--usage`

Displays a brief help message about the **gconftool-2** command, and the options that you can use with the **gconftool-2** command.

## To Set Preference Values

You can set a mandatory value or a default value for a preference key. Before you change mandatory preference values or default preference values for users, you must ensure that the GConf daemon is not running for any user. Ensure that all users are logged out before you change preference values for users.

To set a mandatory value or a default value for a preference key, use the **gconftool-2** command, as follows:

```
gconftool-2 --direct \
--config-source configuration-source \
--type data-type \
--set preference-key value
```

For example, to set `wwwproxy.xyz.com` as the mandatory HTTP proxy host, run the following command:

```
gconftool-2 --direct \
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
--type string \
--set /system/http_proxy/host wwwproxy.xyz.com
```

The user cannot override this preference value.

You can also use the **gconftool-2** command to set default values. For example, to set the default number of workspaces to five, run the following command:

```
gconftool-2 --direct \
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \
--type int \
--set /apps/metacity/general/num_workspaces 5
```

The user can override this preference value.



Before you change mandatory preference values or default preference values for users, you must ensure that all users are logged out.

## Setting General Preferences

The following sections describe how to assign mandatory or default values to general preferences.

### To Set HTTP Proxy Preferences

To set HTTP proxy preferences, you modify the values of the preference keys in the `/system/http_proxy/` location. For example, to set a mandatory value for the HTTP proxy host, run the following command:

```
gconftool-2 --direct \
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
--type string \
--set /system/http_proxy/host proxy-name
```

To set a default value for the HTTP proxy host, run the following command:

```
gconftool-2 --direct \
```

```
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \  
--type string \  
--set /system/http_proxy/host proxy-name
```

You can also set other HTTP proxy-related preferences. For information on the other HTTP proxy preferences, see the `system_http_proxy.schemas` schema definition file.

## To Set Print Manager Preferences

To set print manager preferences, you modify the values of the preference keys in the `/apps/gnome-print-manager` location. For example, if you do not want users to view the print jobs of other users, set a mandatory value as follows:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \  
--type bool \  
--set /apps/gnome-print-manager/show_all_jobs false
```

To set a default value for this preference, run the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \  
--type bool \  
--set /apps/gnome-print-manager/show_all_jobs false
```

You can also set other print manager preferences. For information on the other print manager preferences, see the `gnome-print-manager.schemas` schema definition file.

## To Set the Number of Workspaces

To set a mandatory number of workspaces, use the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \  
--type int \  
--set /apps/metacity/general/num_workspaces integer
```

To set a default number of workspaces, use the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \  
--type int \  
--set /apps/metacity/general/num_workspaces integer
```

You can also set other window manager preferences. For information on the other window manager preferences, see the `metacity.schemas` schema definition file.

## To Set Keyboard Accessibility Preferences

To set keyboard accessibility preferences, you modify the values of the preference keys in the `/desktop/gnome/accessibility/keyboard` location. For example, if you want to set a mandatory value so that keyboard accessibility features are enabled, run the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \  
--type bool \  
--set /desktop/gnome/accessibility/keyboard/enable true
```

To set a default value for this preference, run the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \  
--type bool \  
--set /desktop/gnome/accessibility/keyboard/enable true
```



```
--set /desktop/gnome/accessibility/keyboard/enable false
```

You can also set other keyboard accessibility preferences. For information on the other keyboard accessibility preferences, see the `desktop_gnome_accessibility_keyboard.schemas` schema definition file.

## To Set Keyboard Shortcut Preferences

To set keyboard shortcut preferences, you modify the values of preference keys in `/apps/metacity/global_keybindings` location. For example, you might want users to use only the **Alt-F3** keyboard shortcut to open the Run Application dialog. To set this mandatory value, run the following command:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /apps/metacity/global_keybindings '<Alt>F3'
```

You can also set other keyboard shortcut preferences. For information on the other keyboard shortcut preferences, see the `metacity.schemas` schema definition file.

## Setting Panel and Panel Object Preferences

The `panel-default-setup.entries` file specifies the following details of the panels in the GNOME Desktop:

- Number of panels.
- Types of the panels.
- Properties of the panels.
- Contents of the panels.

The configuration of individual panels and of panel objects is a complex task. To configure individual panels and panel objects, you must first understand the structure of the `panel-default-setup.entries` file. For more information on the `panel-default-setup.entries` file, see the next section.

To set preferences for individual panels and panel objects, you must set the values of many preferences in a configuration source. The easiest way to set the values of panel preferences is to use the **gconftool-2** command with the `--dump` and `--load` options. For more information on how to set preferences for panels and objects on panels, see the section called “To Set Preferences for Individual Panels and Panel Objects”.

### *Specifying Individual Panels and Panel Objects*

The file `panel-default-setup.entries` contains sections that specify panels and panel contents. The `panel-default-setup.entries` file specifies values for schema keys. The `panel-default-setup.entries` file resides in the `/etc/gconf/schemas` directory.

The `panel-default-setup.entries` file is structured as follows:

1. Keys that specify the general structure of panels, applets, and other panel objects in the GNOME Desktop. The following keys specify the number of panels, panel objects, and applets that appear in the GNOME Desktop:
  - `/apps/panel/default_setup/general/toplevel_id_list`
  - `/apps/panel/default_setup/general/object_id_list`
  - `/apps/panel/default_setup/general/applet_id_list`

The keys also assign identifiers to each panel, panel object, and applet. For example, the following sample from `panel-default-setup.entries` specifies that one panel appears in the GNOME Desktop:

```
<entry>
```

```

<key>toplevel_id_list</key>
<schema_key>/schemas/apps/panel/general/toplevel_id_list</schema_key>
<value>
  <list type="string">
    <value>
      <string>bottom_panel</string>
    </value>
  </list>
</value>
</entry>

```

In the `panel-default-setup.entries` file, the identifier `bottom_panel` identifies the bottom edge panel.

2. Keys that specify the properties of the panels. The panel property keys are structured as follows:

```
/apps/panel/default_setup/toplevels/panel-name/panel-property-key
```

For example, the key `/apps/panel/default_setup/toplevels/bottom_panel/size` specifies the size of the bottom panel.

3. Keys that specify the panel objects, the panel object properties, and the panels in which the objects reside. For example, the following sample from `panel-default-setup.entries` specifies a Main Menu object at the left side of the bottom panel:

```

<entrylist base="/apps/panel/default_setup/objects/main_menu">
  <entry>
    <key>object_type</key>
    <schema_key>/schemas/apps/panel/objects/object_type</schema_key>
    <value>
      <string>menu-object</string>
    </value>
  </entry>
  <entry>
    <key>toplevel_id</key>
    <schema_key>/schemas/apps/panel/objects/toplevel_id</schema_key>
    <value>
      <string>bottom_panel</string>
    </value>
  </entry>
  <entry>
    <key>position</key>
    <schema_key>/schemas/apps/panel/objects/position</schema_key>
    <value>
      <int>0</int>
    </value>
  </entry>
  <!-- Possibly more entry elements -->
</entrylist>

```

4. Keys that specify the applets, the applet preferences, and the panels in which the applets reside. For example, the following sample from `panel-default-setup.entries` specifies the Window List applet, in the bottom panel:

```

<entrylist base="/apps/panel/default_setup/applets/window_list">
  <entry>
    <key>object_type</key>
    <schema_key>/schemas/apps/panel/objects/object_type</schema_key>
    <value>
      <string>bonobo-applet</string>
    </value>
  </entry>
  <entry>
    <key>toplevel_id</key>
    <schema_key>/schemas/apps/panel/objects/toplevel_id</schema_key>
    <value>
      <string>bottom_panel</string>
    </value>
  </entry>
  <entry>
    <key>position</key>
    <schema_key>/schemas/apps/panel/objects/position</schema_key>
    <value>
      <int>2</int>
    </value>
  </entry>
</entrylist>

```

```

</entry>
<!-- Possibly more entry elements -->
<entry>
  <key>bonobo_iid</key>
  <schema_key>/schemas/apps/panel/objects/bonobo_iid_type</schema_key>
  <value>
    <string>OAFIID:GNOME_WindowListApplet</string>
  </value>
</entry>
</entrylist>

```

The OAFIID is a unique identifier for an applet. To find the OAFIID for a particular applet, see the `.server` file for the applet in the `/usr/lib/bonobo/servers` directory. For example, the following excerpt from `GNOME_Wncklet_Factory.server` shows the OAFIID for the Window List applet:

```

<oaf_server iid="OAFIID:GNOME_WindowListApplet"
type="factory" location="OAFIID:GNOME_Wncklet_Factory">

```

## To Set Preferences for Individual Panels and Panel Objects

To set the preferences for a panel and the objects on a panel perform the following steps:

1. Log in to a GNOME session, and configure the panels as required.
2. Use the `--dump` option with the **gconftool-2** command line tool to generate a file that contains an XML description of your panel configuration. The `--dump` option generates a list that contains all preference keys in a GConf repository directory that you specify.

For example, the following command creates an XML description of the default panel configuration in a file called `my-panel-setup.entries`:

```
gconftool-2 --dump /apps/panel > my-panel-setup.entries
```

3. Open the `my-panel-setup.entries` file in a text editor, and modify the file as required.

For example, you might want to change the location of the desktop entry files. The following is an excerpt from a file generated with the `--dump` option:

```

<entry>
  <key>objects/object_16/launcher_location</key>
  <schema_key>/schemas/apps/panel/objects/launcher_location</schema_key>
  <value>
    <string>hadjaha-00adce02f7.desktop</string>
  </value>
</entry>

```

In the sample above, you might want to change the reference to `hadjaha-00adce02f7.desktop` to another desktop entry file that is available globally.

When you generate a panel configuration with the `--dump` option, the positions of the panel objects are absolute positions. You might want to change the positions of panel objects from absolute positions to relative positions. The object at the extreme left of a panel has a `position` value of 0. The next object has a `position` value of 1, and so on. If you want object positions to be relative to the right side of the panel, set the value of the `right_stick` key to `true`.

4. Use the `--load` option with the **gconftool-2** command line tool to set the values of the default configuration source to the values in the `my-panel-setup.entries` file. For example, the following command sets the values of the keys in the default configuration source to the values of the corresponding keys in `my-panel-setup.entries`:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \
  --load my-panel-setup.entries
```

# Setting Look-and-Feel Preferences

The following sections describe how to assign mandatory or default values to look-and-feel preferences.

## To Set Font Preferences

To set font preferences, you modify the values of two preference keys. The following table shows the keys to modify, and the part of the user interface to which the keys correspond:

GConf Location	User Interface Component
/desktop/gnome/interface/font_name	Font preference tool, Application font option
/apps/nautilus/preferences/desktop_font	Font preference tool, Desktop font option

For example, to set `Sans 12` as the mandatory application font, run the following command:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /desktop/gnome/interface/font_name "Sans 12"
```

To set `palatino 12` as the default desktop object font, run the following command:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \
  --type string \
  --set /apps/nautilus/preferences/desktop_font "palatino 12"
```

## To Set Background Preferences

To set preferences for the desktop background, you modify the values of the preference keys in the `/desktop/gnome/background` location. For example, to set a mandatory image for the background, run the following command:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /desktop/gnome/background/picture_filename filename.png
```

To set a default value for this preference, run the following command:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \
  --type string \
  --set /desktop/gnome/background/picture_filename filename.png
```

You can also set other background preferences. For information on the other background preferences, see the `desktop_gnome_background.schemas` schema definition file.

## To Set Splash Image Preferences

To set splash image preferences, you modify the value of the preference keys in the `/apps/gnome-session/options/` location. For example, if you do not want users ever to see a splash image, set a mandatory value as follows:

```
gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
```

```
--type bool \  
--set /apps/gnome-session/options/show_splash_screen false
```

To set a default value for this preference, run the following command:

```
gconftool-2 --direct \  
--config-source xml:readwrite:/etc/gconf/gconf.xml.defaults \  
--type bool \  
--set /apps/gnome-session/options/show_splash_screen false
```

You can also set other splash image preferences. For information on the other splash image preferences, see the `gnome-session.schemas` schema definition file.

## To Restore Default Preference Values

To restore the default preference values for a user, run the following command:

```
gconftool-2 --direct \  
--config-source user-configuration-source \  
--recursive-unset
```

Replace *user-configuration-source* with the configuration source in the `.gconf` directory in the home directory of the user.

This command resets the values of all preference keys, in all subdirectories, from the user setting to the setting in the default configuration source.

# **Chapter 2**

---

## **Customizing Menus**

The information in this chapter describes how the GNOME Desktop implements menus and how you can customize menus.

## Introduction to Menus

The GNOME Desktop implements menus according to the XDG menu specification [<http://www.freedesktop.org/wiki/Standards/menu-spec>]. By supporting this specification, GNOME allows you to:

- Customize the menu hierarchy easily. You can edit a small number of files to customize the menu hierarchy. You do not need to modify your applications or move files.
- Install applications easily. You do not need to provide information about the menu hierarchy to applications when you install the applications.
- Configure menus so that users cannot modify the menus.

Menus in the GNOME Desktop use the following components:

- Menu definition files
- Desktop entry files
- Directory entry files

## Menu Definition Files

Menu files define the hierarchy of menus that are used in the GNOME menu bar. By modifying these files, you can customize menus for all users, or for a single user depending on the location of the `applications.menu` file that you modify.

Menu files must reside at `$XDG_CONFIG_DIRS/menus/applications.menu`. If `$XDG_CONFIG_DIRS`<sup>1</sup> is not set, then the default path `/etc/xdg/` is used. This also implies that a user specific version may be located at `$XDG_CONFIG_HOME/menus/applications.menu` which is searched first. If `$XDG_CONFIG_HOME` is not set, then the default path `~/.config/` is used. Directories which appear first in `$XDG_CONFIG_DIRS` are given precedence when there are several `applications.menu` files. The first file found is used and subsequent files are ignored.

If you are confused about the order in which paths are searched, here is a simple list for resolving the location of `applications.menu`:

1. Search each directory in `$XDG_CONFIG_HOME` in order to find `/menus/applications.menu`. If `$XDG_CONFIG_HOME` is not set, default to `~/.config/`
2. Search each directory in `$XDG_CONFIG_DIRS` in order to find `/menus/applications.menu`. If `$XDG_CONFIG_DIRS` is not set, default to `/etc/xdg/`
3. Use the first `applications.menu` file found.

You can see an example of a `.menu` file in Example 2.1, “Example of a `.menu` file”. In this example, the top level menu is named `Applications`, which is specified using the `<Name>` element. The `Applications` menu contains a single submenu, but several submenus are allowed. Each submenu may also have an `<Include>` element. The purpose of the `<Include>` element is to perform a filter on the set of available desktop entries using matching rules.

For example, the `<Category>` element is a basic matching rule that selects a *desktop entry* only if the `Categor-`

---

<sup>1</sup>`$XDG_CONFIG_DIRS` is the environment variable defined in the XDG base directory specification [<http://www.freedesktop.org/Standards/basedir-spec>].

ies key contains the content of the `<Category>` element. In the example, the `Accessories` menu will include a *desktop entry* only if it contains “Utility” but not “System” in the `Categories` key. For more information on the `Categories` key, see the section called “Desktop Entry Files”.

### Example 2.1. Example of a `.menu` file

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

  <!-- Read standard .directory and .desktop file locations -->
  <DefaultAppDirs/>
  <DefaultDirectoryDirs/>

  <!-- Accessories submenu -->
  <Menu>
    <Name>Accessories</Name>
    <Directory>Accessories.directory</Directory>
    <Include>
      <And>
        <Category>Utility</Category>
        <Not>
          <Category>System</Category>
        </Not>
      </And>
    </Include>
  </Menu> <!-- End Accessories -->

  <!-- possibly more submenus -->
</Menu> <!-- End Applications -->
```

Table 2.1, “Menu Definition File Elements” describes some of the elements in `.menu` files. For a more detailed description, please see the XDG menu specification [<http://www.freedesktop.org/wiki/Standards/menu-spec>].

### Table 2.1. Menu Definition File Elements

Element	Description
<code>&lt;Menu&gt;</code>	The root element which may contain nested <code>&lt;Menu&gt;</code> elements that define sub-menus. How these elements are nested determines the menu structure.
<code>&lt;Name&gt;</code>	Specifies the name of the menu. Every <code>&lt;Menu&gt;</code> element must contain a <code>&lt;Name&gt;</code> element.
<code>&lt;Directory&gt;</code>	Specifies the name of the directory entry file that specifies the name, comment, and icon for the menu. If this element is not specified, then the <code>&lt;Name&gt;</code> element is to be used to display the menu name.  By default, <code>.directory</code> files are searched for in the location <code>\$XDG_DATA_DIRS/desktop-directories/</code> as set forth in the <i>XDG menu specification</i> .
<code>&lt;DefaultAppDirs&gt;</code>	This is an instruction which indicates that all the available desktop entries from <code>\$XDG_DATA_DIRS/applications/</code> should be scanned. If this instruction is not included, then these locations are not scanned for desktop entries.
<code>&lt;DefaultDirectoryDirs&gt;</code>	This is an instruction which indicates that all the available directory entries from <code>\$XDG_DATA_DIRS/desktop-directories/</code> should be scanned. If the instruction is not included, then these locations are not scanned for directory



Element	Description
	entries.
<Include>	Contains a list of matching rules by which the contents of a menu are generated. May include the <Filename>, <Category>, <And>, <Or>, <Not>, or <All> matching rules. If more than one rule is present, the rules are logically ORed so that <i>desktop entries</i> that match any rule are included.
<Exclude>	The opposite of <Include> since any <i>desktop entries</i> that are matched in this element are excluded from the previous set of included elements. For this reason, this element must appear after the <Include> element.
<Filename>	A matching rule that selects a <i>desktop entry</i> when the Desktop File-Id matches the contents of the <Filename> element.
<Category>	A matching rule that selects a <i>desktop entry</i> when the Categories key matches the contents of the <Category> element.
<And>	A matching rule that selects a <i>desktop entry</i> when it is selected by <i>all</i> the nested matching rules in the <And> element.
<Or>	A matching rule that selects a <i>desktop entry</i> when it is selected by <i>any</i> of the nested matching rules in the <Or> element.
<Not>	A matching rule that does not select a <i>desktop entry</i> when it is selected by <i>any</i> of the nested matching rules in the <Not> element.
<All>	A matching rule which selects all <i>desktop entries</i> .

## Desktop Entry Files

A *desktop entry file* is a data file that provides information about an item in a menu. The desktop entry file specifies the details for the item such as a name, a command to run, an icon, and so on. It also contains keywords which determine the location of the item in the menu hierarchy.

Desktop entry files must reside in the `$XDG_DATA_DIRS/applications` directory and must have a `.desktop` file extension. If `$XDG_DATA_DIRS`<sup>2</sup> is not set, then the default path is `/usr/share/` is used. This also implies that user specific desktop entries may be located at `$XDG_DATA_HOME/applications/` which is searched first. If `$XDG_DATA_HOME` is not set, then the default path `~/.local/share` is used. Desktop entries are collected from all directories in the `$XDG_DATA_DIRS` environment variable. Directories which appear first in `$XDG_CONFIG_DIRS` are given precedence when there are several `.desktop` files with the same name.

The following is a sample desktop entry file:

```
[Desktop Entry]
Encoding=UTF-8
Name=Calculator
Name[ar]=### #####
...
Comment=Perform calculations
Comment[ar]=##### #####
...
Exec=gcalctool
Icon=gnome-calculator
Terminal=false
Type=Application
StartupNotify=true
Categories=GTK;GNOME;Application;Utility;
```

Table 2.2, “Desktop Entry Keys” describes the most important keys in desktop entry files. To get more information about desktop entry files, see the XDG Desktop Entry Specification [<http://www.freedesktop.org/Standards/desktop-entry-spec>].

<sup>2</sup>`$XDG_DATA_DIRS` is the environment variable defined in the XDG base directory specification [<http://www.freedesktop.org/Standards/basedir-spec>].

**Table 2.2. Desktop Entry Keys**

Desktop Entry Key	Description
Encoding	Specifies the encoding of the desktop entry file.
Name	Specifies the name of the item. This name is displayed on the item in the menu.
Comment	Specifies a short description of the item. The comment is displayed as a tooltip when you point to the item in the menu.
Exec	Specifies a command to execute when you choose the item from the menu.
Icon	Specifies the filename of an icon that represents the item. Does not specify the path to the filename, or the file extension.
Terminal	Specifies whether the command in the <code>Exec</code> key runs in a terminal window. If the value is <code>true</code> the command runs in a terminal window.  If the command does not create a window in which to run, the value of this key must be <code>true</code> .
Type	Specifies the type of item. This value is one of the following: <ul style="list-style-type: none"> <li>• <code>Application</code>: An item that starts an application.</li> <li>• <code>Link</code>: An item that links to a file, folder, or a remote resource (such as a FTP site, a web page, a Windows share...).</li> <li>• <code>FSDevice</code>: An item that is a file system device.</li> <li>• <code>Directory</code>: An item that is a Directory.</li> </ul>
Categories	Specifies the keywords that describe the item. The keywords are separated with semicolons (;). To see a list of the standard category keywords, see the desktop menu specification at the following URL:  <div style="background-color: #f0f0f0; padding: 5px; text-align: center;"> <a href="http://www.freedesktop.org/Standards/menu-spec">http://www.freedesktop.org/Standards/menu-spec</a> </div> Menu Definition Files map desktop entries to menus by using matching rules against the <code>Categories</code> key.
MimeType	Specifies the MIME types

For more information on the keys in desktop entry files, see the desktop entry specification at the following URL:

<http://www.freedesktop.org/Standards/desktop-entry-spec>



Panel launchers and desktop objects also use desktop entry files. The desktop entry files for launchers and desktop objects provide the same information as for items in a menu. For example, the desktop entry files provide the command to run when a user chooses the launcher or object.

## Directory Entry Files

A *directory entry file* is a data file that provides information about a menu. The directory entry file specifies the details for the menu such as a name, a tooltip, and an icon. Directory entry files have a `.directory` file extension.

Directory entry files must reside at `$XDG_DATA_DIRS/desktop-directories/`. If `$XDG_DATA_DIRS` is not set, then the default path is `/usr/share/` is used. This also implies that user specific directory entries may be located at `$XDG_DATA_HOME/desktop-directories/` which is searched first. If `$XDG_DATA_HOME` is not set, then the default path `~/.local/share/` is used. Directory entries are collected from all directories in the `$XDG_DATA_DIRS` environment variable. Directories which appear first in `$XDG_DATA_DIRS` are given precedence when there are several `.directory` files with the same name.

The following is a sample directory entry file:

```
[Desktop Entry]
Name=Accessories
Comment=Accessories menu
Icon=gnome-util.png
Type=Directory
```

Table 2.3, “Directory Entry Keys” describes the most important keys in directory entry files.

**Table 2.3. Directory Entry Keys**

Directory Entry Key	Description
Name	Specifies the name of the menu. This name is displayed on the menu.
Comment	Specifies a short description of the menu. The comment is displayed as a tooltip when you point to the menu.
Icon	Specifies the filename of an icon that represents the menu. Does not specify the path to the filename, or the file extension.
Type	Specifies the type of menu. The value of this key is always <code>Directory</code> .

## Editing System Menus

You can edit menu configuration files and menu data files manually to customize menus.

### Adding Menus

To add a menu for all users, perform the following steps:

1. Create a directory entry file for the item that you want to add. Place the directory entry file in the `$XDG_DATA_DIRS/desktop-directories/` directory. For more information on directory entry files, see the section called “Directory Entry Files”.
2. Locate the `$XDG_CONFIG_DIRS/menus/applications.menu` file.
3. In the `.menu` file, add a `<Menu>` element for the new menu. For more information on `.menu` files, see the section called “Menu Definition Files”.
4. Create a `<Name>` element below `<Menu>`. The content of the element should contain the name for the menu.

5. Create a `<Directory>` element below `<Menu>`. The content of the element should contain the name of the directory entry file.
6. See the section called “Adding an Item to a Menu” for how to add an item to the menu.

The next time that users log in, the menu should appear in the menu bar.



### Missing Menu?

If you did not specify any matching rules in the `<Include>` element, or if the rule did not match any desktop entries, then you may not see the menu in the menu bar.

## Adding an Item to a Menu

To add an item to a menu for all users, perform the following steps:

1. Create a desktop entry file for the item that you want to add. For more information on desktop entry files, see the section called “Desktop Entry Files”.
2. Place the desktop entry file in the folder `$XDG_DATA_DIRS/applications/`
3. Locate the `$XDG_CONFIG_DIRS/menus/applications.menu` file.
4. Verify that a `<Menu>` element contains an `<Include>` element with a matching rule that selects the desktop entry file made in step 1.

The next time that users log in, the menu item is in the assigned location.

## Editing the Properties of a Menu

To edit the properties of a menu for all users, perform the following steps:

1. Locate the `$XDG_CONFIG_DIRS/menus/applications.menu` file.
2. Find the `<Menu>` entry in this file that corresponds to the menu you want to modify. Note the filename of the directory entry in the `<Directory>` element.
3. Locate the directory entry for this menu. Modify the contents to change the properties of the menu. For more information on `.directory` files, see the section called “Directory Entry Files”.

## Editing a Menu Item

To edit a menu item, perform the following steps:

1. Locate the desktop entry in `$XDG_DATA_DIRS/applications/` corresponding to the menu item.
2. Edit the desktop entry to change the properties of the menu item. For more information on desktop entry files, see the section called “Desktop Entry Files”.

## Deleting an Item from a Menu

To delete an item from a menu for all users:

1. Locate the `$XDG_CONFIG_DIRS/menus/applications.menu` file.
2. Find the `<Menu>` element in this file that contains the desktop entry you want to delete.
3. Insert an `<Exclude>` element after the closing tag for the `<Include>` element. Make sure this is in the `<Menu>` element determined in step 2.
4. Insert the `<Filename>` matching rule as a subelement of `<Exclude>` to specifically exclude a desktop

entry.

The next time that users log in, the menu item is not displayed in the menu. Example 2.2, “Deleting an Item from a Menu” shows how this done in the `applications.menu` file. The desktop entry for `dasher.desktop` is explicitly excluded from showing up in the accessibility menu.

### Example 2.2. Deleting an Item from a Menu

```
<!-- ... -->
<Menu>
  <Name>Accessibility</Name>
  <Directory>Accessibility.directory</Directory>
  <Include>
    <And>
      <Category>Accessibility</Category>
      <Not><Category>Settings</Category></Not>
    </And>
  </Include>
  <Exclude>
    <Filename>dasher.desktop</Filename>
  </Exclude>
</Menu>
<!-- ... -->
```

## Editing User Menus and Menu Merging

You can use the following GNOME Desktop applications to edit menus for users:

- GNOME Menu Editor

A simple menu editor is available for users to edit their menus. For more information, see [Working With Menus \[ghelp:user-guide?goseditmainmenu-1\]](#) in the GNOME User Guide [ghelp:user-guide]. Alternatively, you can manually create and edit a user menu file.

To manually create a custom menu for a user, the `$XDG_CONFIG_HOME/menus/applications.menu` must exist. In the case that `$XDG_CONFIG_HOME` is not set, the default `~/.config/` is used. Since this is the first location that is searched for the `applications.menu` file, it takes precedence over all other menu files.

User menus can contain all the elements described in the section called “Menu Definition Files”. For a complete list of the elements allowed, see the XDG menu specification [<http://www.freedesktop.org/wiki/Standards/menu-spec>].

Since user menu files take precedence over the system menu file, it will completely replace the system menu unless it explicitly *merges* the system menu. Information on menu merging is available in the following subsections.

### Merging the System Menu

Often, a user only wants to add or delete menu items in addition to the standard system menu. To support single changes like these, it is recommended that you use the `<MergeFile>` element with the attribute `type="parent"` within the user's `applications.menu` file.

The `<MergeFile>` element allows a menu to be merged with the contents of the user's menu file. When you specify the attribute `type="parent"`, then the contents of the `<MergeFile>` element are ignored and the next `applications.menu` file in `$XDG_CONFIG_DIRS/menus/` is used for merging.



## Older Specifications

Older specifications did not include the `type` attribute and simply required the location of the menu file to be merged as the content of the `<MergeFile>` element. As a result, you may still see a location specified in the contents of `<MergeFile>`, even when `type="parent"`.

The merging is performed as follows:

- The children of the root `<Menu>` element in the merged menu file<sup>3</sup> are substituted for the `<MergeFile>` element in the base menu file.
- All child `<Menu>` elements with the same name are consolidated into a single `<Menu>` element. This is done by appending all child elements of each `<Menu>` element with the same name into the *last* occurrence of the menu element.

Example 2.3, “Merging the System Menu” shows an example of a user menu file explicitly merging the system menu file.

### Example 2.3. Merging the System Menu

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
"http://www.freedesktop.org/standards/menu-spec/menu-1.0.dtd">
<Menu>
  <Name>Applications</Name>
  <MergeFile type="parent">/etc/xdg/menus/applications.menu</MergeFile>
  <Menu>
    <Name>Accessibility</Name>
    <Exclude>
      <Filename>dasher.desktop</Filename>
    </Exclude>
  </Menu>
</Menu>
```

## Merging Arbitrary Menus

Arbitrary menu files can be merged in much the same way as system menus. The difference is that the `type` attribute must be set to `path` or must be excluded from the `<MergeFile>` element in order to do this type of merge.

The merge is performed in the same way except that the location of the *merged menu file* is specified in the contents of the `<MergeFile>` element.

<sup>3</sup>Merged menu file refers to the next `applications.menu` in `$XDG_CONFIG_DIRS/menus/`

# Chapter 3

---

## Installing Themes

This chapter describes themes, the theme settings that are available in the GNOME Desktop, how to install options for theme settings, and how to create a custom option.

## Introduction to Themes

A theme is a group of coordinated settings that specifies the visual appearance of a part of the GNOME Desktop. Users can choose themes to change the appearance of the GNOME Desktop.

A theme contains settings that affect different parts of the GNOME Desktop, as follows:

### Controls

The controls setting for a theme determines the visual appearance of windows, panels, and applets. The controls setting also determines the visual appearance of the GNOME-compliant interface items that appear on windows, panels, and applets, such as menus, icons, and buttons. Some of the controls setting options that are available are designed for special accessibility needs. Users can choose an option for the controls setting from the Controls tabbed section in the Theme preference tool.

### Window frame

The window frame setting for a theme determines the appearance of the frames around windows only. Users can choose an option for the window frame setting from the Window Border tabbed section in the Theme preference tool.

### Icon

The icon setting for a theme determines the appearance of the icons on panels and the desktop background. Users can choose an option for the icon setting from the Icons tabbed section in the Theme preference tool.

## Theme Index File

Each theme has an index file which defines the characteristics of the theme. The name of the index file is `/usr/share/themes/theme-name/index.theme`.

The following is a sample theme index file:

```
[Desktop Entry]
Type=X-GNOME-Metatheme
Name=High Contrast Large
Name[es]=Alto contraste grande
Comment=Large black-on-white text and icons
Comment[es]=Textos e iconos grandes en negro sobre blanco
Encoding=UTF-8

[X-GNOME-Metatheme]
GtkTheme=HighContrastLargePrint
IconTheme=HighContrast
MetacityTheme=Atlanta
ApplicationFont=sans 18
```

The following table describes the keys in theme index files:

**Table 3.1. Keys in Theme Index Files**

Index File Key	Description
Type	Specifies that this theme determines the appearance of several theme options, such as controls, window frames, and icons.



<b>Index File Key</b>	<b>Description</b>
Name	The name of the theme. This is the name of the theme that is displayed in the Theme preference tool.
Comment	A brief description of the theme. This is the text that is displayed under the name of the theme in the Theme preference tool.
GtkTheme	Corresponds to the controls setting in the Theme preference tool. Specifies which controls setting option to apply to windows, panels, and applets.
IconTheme	Corresponds to the icons setting in the Theme preference tool. Specifies which icons setting option to apply to panels and the desktop background.
MetacityTheme	Corresponds to the window frame setting in the Theme preference tool. Specifies which window frame setting option to apply to windows.
ApplicationFont	Corresponds to the application font setting in the Font preference tool.

## To Install a New Controls Option

You can add a new option for the controls setting in the Theme preference tool. Controls options reside in the `/usr/share/themes` directory. The typical structure of a controls option in the file system is as follows:

Option file

```
/usr/share/themes/option-name/gtk-2.0/gtkrc
```

Image files

```
/usr/share/themes/option-name/pixmaps/*.*
```

Typically, a new option for the controls setting is supplied as a `.tar.gz` file. To install the new controls option, unzip the `.tar.gz` file, then untar the `.tar` file into the `/usr/share/themes` directory.

Users can install their own options for the controls setting. If a user installs an option for the controls setting, the option is stored in the `$HOME/.themes` directory.

## To Install a New Window Frame Option

You can add a new option for the window frame setting in the Theme preference tool. Window frame options reside in the `/usr/share/themes/option-name/metacity-1` directory. The typical structure of a window frame option in the file system is as follows:

Option file

```
/usr/share/themes/option-name/metacity-1/metacity-theme-1.xml
```

Image files

```
/usr/share/themes/option-name/metacity-1/*.*
```

Typically, a new option for the window frame setting is supplied as a `.tar.gz` file. To install the new window

frame option, unzip the `.tar.gz` file, then untar the `.tar` file into the `/usr/share/themes` directory.

Users can install their own options for the window frame setting. If a user installs an option for the window frame setting, the option is stored in the `$HOME/.themes` directory.

## To Install a New Icons Option

You can add a new option for the icons setting in the Theme preference tool. Icons options reside in the `/usr/share/icons/option-name` directory. The typical structure of an icons option in the file system is as follows:

Option file

```
/usr/share/icons/option-name
```

Image files

```
/usr/share/icons/option-name/icons/*.*
```

Typically, a new option for the icons setting is supplied as a `.tar.gz` file. To install the new icons option, unzip the `.tar.gz` file, then untar the `.tar` file into the `/usr/share/icons` directory.

Users can install their own options for the icons setting. If a user installs an option for the icons setting, the option is stored in the `$HOME/.icons/option-name` directory.

## Installing Icons for Themes

The GNOME Desktop provides several themes that are designed for users with special visual needs. For example, some of the themes are designed for users with low vision. Several versions of icons might be required so that the icon can be displayed properly in each theme.

You might need to install a new icon for an application. When you install a new icon, you must create several versions of the icon, so that the icon displays correctly in the themes. You must create several versions of the following types of icon:

- Icons that are used within applications in the GNOME Desktop.
- Icons that are used internally by GTK+ applications, or GTK+ stock icons.

When you install a new icon, you must create several versions of the icon, so that the icon displays correctly in the themes. You can create the icons in several formats, for example Portable Network Graphic (PNG) format. The suggested size of icons for the desktop environment is 48 pixels by 48 pixels. At this size, most themes can rescale the icons.

When you install a new icon, create the following 48 pixel by 48 pixel versions of the icon:

- Regular icon
- Low contrast icon
- High contrast icon
- Inverse high contrast icon

If possible, create 16 pixel by 16 pixel versions of the each of the icons above also, for themes that do not require large print.

Install the icons to the image files location that is specified for the theme in the section called “To Install a New Controls Option” or the section called “To Install a New Window Frame Option”. For example, to add icons to the HighContrastLargePrint theme, add the icons to the /

`usr/share/themes/HighContrastLargePrint/pixmaps` directory. Add references to the icons to the relevant theme files. For example, to add icons to the `HighContrastLargePrint` theme, add references to the icons to the `/usr/share/themes/HighContrastLargePrint/gtk-2.0/gtkrc` file. Modify the `gtkrc` file for the theme to associate the icon with a GTK stock icon identifier.

For more information on how to create icons for application launchers and for panels, see <http://www.freedesktop.org/Standards/icon-theme-spec>.

## To Create a Custom Controls Option

If the options for the controls setting are not suitable for the needs of your users, you can create a custom controls option. To create a custom controls option, perform the following steps:

1. Create a directory structure for the option in the `/usr/share/themes` directory. Use the same directory structure that other options use. For example, to create an option that is called `SmallPrint`, create the following directories:
  - `/usr/share/themes/SmallPrint`
  - `/usr/share/themes/SmallPrint/gtk-2.0`
2. Locate the `gtkrc` file that is closest to meeting the needs of your users. Copy the file to the `gtk-2.0` directory of your new option.
3. Open the `gtkrc` file in a text editor, and modify the attributes of the interface elements as required.
4. If the new option includes images, install the images for the new option in the `pixmaps` directory of your new option. If the new option uses images from another option, you do not need to create copies of the images for the new option. Instead, ensure that the reference to the images in the  `pixmap_path`  entry in the `gtkrc` file is correct.

Users can now choose the new option for the controls setting.

For more information on `gtkrc` files, see <http://developer.gnome.org/doc/API/2.0/gtk/index.html>.

# **Chapter 4**

---

## **Configuring Fonts**

This chapter describes how to configure fonts in the GNOME Desktop.

## Introduction to Configuring Fonts

GNOME Desktop uses the `fontconfig` font configuration and customization library. The `fontconfig` library can use all kinds of fonts, including PostScript Type 1 fonts and TrueType fonts.

The `fontconfig` library provides a list of all the fonts available on a GNOME Desktop system. To compile this list, `fontconfig` searches the directories listed in the `/etc/fonts/fonts.conf` file. To view all the fonts available on a system, access the `fonts:///` location in the file manager on the system.

For more information about the `fontconfig` library, see the following URL:

```
http://freedesktop.org/software/fontconfig
```

## Font Substitution

The `fontconfig` library performs font substitution when either entire fonts, or individual characters, are not present. If the system needs to display a font that is not available, `fontconfig` attempts to display another, similar font. For example, if a web page requests to display the Verdana font, and that font is not installed on the system, `fontconfig` displays a similar font, such as Helvetica. The list of similar fonts is defined in the `/etc/fonts/fonts.conf` file.

If the system needs to display a character that is not present in the selected font, `fontconfig` attempts to display the character in another, similar font. For example, you might select Bitstream Vera Sans as the font for the Text Editor application. The Bitstream Vera font family does not include Cyrillic characters. If you open a document which contains a Cyrillic character, Text Editor uses a similar font that includes Cyrillic characters to display the character.

The `fontconfig` library also defines aliases for fonts, for example, `serif`, `sans-serif`, and `monospace`. When you select one of the aliases for a font, the system uses the first font that is defined for that alias in the `/etc/fonts/fonts.conf`.

## To Add a Font for All Users

To add a font for all users, perform the following steps:

1. Copy the font file to one of the directories in the `/etc/fonts/fonts.conf` file. Typically, fonts are stored in the `/usr/share/fonts/` directory.
2. The `fontconfig` library updates the list of fonts automatically. If the list of fonts is not updated, run the following command:

```
# fc-cache directory-name
```

## To Add a Font for an Individual User

To add a font for an individual users, perform the following steps:

1. Copy the font file to the `$HOME/.fonts` directory of the user. If you drag the font file to the `fonts:///` location in the file manager, the font file is copied to the `$HOME/.fonts` directory.

2. The `fontconfig` library updates the list of fonts automatically. If the list of fonts is not updated, run the following command:

```
# fc-cache directory-name
```

# Chapter 5

---

## MIME Types

This chapter describes how applications detect MIME types, how to register MIME types, and how to add applications to the GNOME Desktop.

The purpose of this chapter is to make it easy for an administrator to understand how to configure different parts of the MIME database and give an general overview of the MIME system. Therefore we will *not* attempt to go into details where it is not necessary. For the gory details, it is recommended that you refer to the XDG shared mime info specification [<http://www.freedesktop.org/Standards/shared-mime-info-spec>].

## Introduction to MIME Types

A *Multipurpose Internet Mail Extension* [MIME] type identifies the format of a file. Applications such as Internet browsers and email applications use the MIME type of a file to decide which actions to perform on it. For example, an email application can use the MIME type to detect the format of an attachment and choose an appropriate viewer for the file, such as opening a text document with `gedit`.

As another example, the Nautilus file manager needs to know the MIME type of a file to perform the following tasks:

- Open the file in an appropriate application.
- Display a string that describes the type of file.
- Display an appropriate icon to represent the file.
- Display a list of other applications that can open the file.

MIME types were originally proposed as a standard for identifying the message body of an e-mail message. Now, many systems use MIME types to identify the format of arbitrary files on the file system. MIME types are composed of a top-level *media type* followed by a *subtype identifier*, separated by a forward slash character, “/”. One example of a MIME type is `image/jpeg`. The media type in this example is “image” and the subtype identifier is “jpeg”. The top-level media type is meant to be a general categorization about the content of the file, while the subtype identifier is meant to specifically identify the format of the file.

There are eight media types currently blessed by the IANA [Internet Assigned Naming Authority]. These eight media types are application, audio, image, message, model, multipart, text, video. Many subtypes exist for each media type; for more information, see MIME Media Types [<http://www.iana.org/assignments/media-types/>] at the IANA [<http://www.iana.org/>] web site.

Implementation of MIME types in GNOME follows the XDG shared mime info specification [<http://www.freedesktop.org/Standards/shared-mime-info-spec>]. This specification provides the following advantages:

- Standard locations for all MIME related files.
- A standard way for applications to register information about a new MIME type.
- A standard way to retrieve the MIME type for a file.
- A standard way to retrieve information about a MIME type.

The rest of the chapter describes the files and directories that make up the MIME database, details about source XML files, how to create or modify MIME types, how to register applications as handlers for certain MIME types and finally, how to add an application to the GNOME desktop.

## The MIME Database

The MIME database is a collection of files that make up:



- The set of known MIME types
- The method for determining the MIME type of a file
- Meta information regarding a MIME type, such as a human readable description to use when displaying files of this type.

## Location

As an administrator, the most important and basic step to understanding the MIME system is learning the locations where these files are stored. Since the XDG shared mime info specification [<http://www.freedesktop.org/Standards/shared-mime-info-spec>] was drafted by the X Desktop Group, it also makes use of the XDG base directory specification [<http://www.freedesktop.org/wiki/Standards/basedir-spec>]. It is highly recommended that you familiarize yourself with this specification, as it is also important for other system administration tasks such as editing menus. A brief summary of the directory locations as pertaining to the MIME specification is given below.

The MIME database is created from the set of files located in the `$XDG_DATA_HOME/mime` and `$XDG_DATA_DIRS/mime` directories. If these environment variables are unset, then they default to the values `~/.local/share` and `/usr/local/share:/usr/share` respectively. As can be seen from the default value for `$XDG_DATA_DIRS`, each environment variable is actually a colon separated list of directories. The user's database at `$XDG_DATA_HOME/mime` has precedence over the system database at `$XDG_DATA_DIRS/mime` when conflicting definitions are encountered. Similar to the *XDG shared mime specification*, we will refer to this set of directories as `<MIME>` in the rest of this document.

For example, assuming default paths for the environment variables, “Load `<MIME>/text/plain.xml`” means to load the following files:

- `~/.local/share/mime/text/plain.xml`
- `/usr/local/share/mime/text/plain.xml`
- `/usr/share/mime/text/plain.xml`

## Contents

The following is a list of directories and files that are found inside the MIME database along with brief descriptions:

`<MIME>/packages/`

This directory contains any number of XML files, each of which describe a collection of MIME types. By default, the `freedesktop.org.xml` file is installed in the `/usr/share/mime/packages` directory. This file contains all the default MIME types that are widely used and recognized.

Applications which provide information about new MIME types are to install a *single* new XML file here. Depending on the prefix where the application is installed, it will create the file in the `/mime/package` subdirectory of one of the directories in `$XDG_DATA_HOME:$XDG_DATA_DIRS`. For example, an application installed to `/usr/bin` should install a new source XML file to the `/usr/share/mime/packages` directory. For more information about the XML files in the `packages` directory, please see the section called “The source XML files”.

`<MIME>/MEDIA/SUBTYPE.xml`

These directories and files are automatically generated from the collection of source XML files in the `<MIME>/packages/` subdirectory by the `update-mime-database` application. For example, for each `mime-type` element in the `/usr/share/mime/packages/freedesktop.org.xml` file, a directory is created at `/usr/share/mime/` with the *media type* of the MIME type. An XML file is created in that directory with the *subtype identifier* of that MIME type as well. The contents of the created XML file include comments (and translations for them), subclasses designations and aliases.

### Example 5.1. Example: `/usr/share/mime/text/plain.xml` file

```
<?xml version='1.0' encoding='utf-8'?>
<mime-type xmlns="http://www.freedesktop.org/standards/shared-mime-info" type="text/plain">
<!--Created automatically by update-mime-database. DO NOT EDIT!-->
  <comment>plain text document</comment>
  <!-- possibly more translations -->
  <comment xml:lang="es">documento de texto sencillo</comment>
  <comment xml:lang="eu">testu soileko dokumentua</comment>
  <comment xml:lang="fi">perustekstiasiakirja</comment>
  <comment xml:lang="fr">document plein texte</comment>
  <!-- possibly more translations -->
</mime-type>
```

This file is generated by the update-mime-database application, using the default source XML file freedesktop.org.xml.

#### <MIME>/globs

Contains one line with a MIME type and a glob pattern, separated by a colon. Files which match the glob pattern are resolved to the MIME type specified before the colon. There are special rules about how filenames are matched by the glob pattern; for more details see the *XDG shared mime specification*.

This file is also generated by the update-mime-database application, using the default source XML file freedesktop.org.xml.

#### <MIME>/magic

A binary file which contains information on how to resolve MIME types by “sniffing” the content of the file. This is generally a set of one or more rules such as “check for the string %PDF- at byte offset 0 in the file; if found, assign it the MIME type application/pdf”.

This file is also generated by the update-mime-database application.

#### <MIME>/XMLnamespaces

Contains a mapping of XML namespaces to MIME types. Each line contains three fields, the namespace, the localName and the MIME type. Each field is separated by a space. If the localName is empty, then there are two spaces between the namespace and the MIME type.

This file is also generated by the update-mime-database application.

#### <MIME>/aliases

Contains a list of aliases for each MIME type. An alias is simply a MIME type that is sometimes known as another type. For each line in this file there are two fields: the first field is the alias name, and the second field is the MIME type. The fields are separated by a space.

This file is also generated by the update-mime-database application.

#### <MIME>/subclasses

Contains a list of subclassed MIME types and their “parent” MIME type. From the *XDG shared mime specification*:

A type is a subclass of another type if any instance of the first type is also an instance of the second. For example, all image/svg files are also text/xml, text/plain and application/octet-stream files. Subclassing is about the format, rather than the category of the data (for example, there is no 'generic spreadsheet' class that all spreadsheets inherit from).

The format of this file is similar to the `aliases` file. Each line contains two fields, where the first field is the subclassed MIME type and the second field is the parent MIME type. Each field is separated by a space.

This file is also generated by the update-mime-database application.

## Refreshing the MIME Database

Understanding how to refresh the MIME database is important for administrators who wish to add new MIME types to the system, or otherwise modify information about a MIME type. The application update-mime-database is inten-

ded for this purpose.

For example, if an application installs information about a new MIME type to `/usr/share/mime/packages/diff.xml`, then `update-mime-database` must be called with the parameter `/usr/share/mime`.

```
# update-mime-database /usr/share/mime
***
* Updating MIME database in /usr/share/mime...
***
```

The MIME database is refreshed by scanning all the source XML files in the directory `<MIME>/packages`.

## The source XML files

Located in the `<MIME>/packages` directory, these XML files provide all the information regarding MIME types that is installed into the database by the `update-mime-database` application. There are a few rules about the XML file itself:

- It must specify the namespace as “`http://www.freedesktop.org/standards/shared-mime-info`”
- The root element must be `mime-info`
- Zero or more `mime-type` elements can be specified as children of the `mime-info` element. The `type` attribute is used to specify the MIME type that is being defined.

By default, the `freedesktop.org.xml` file is installed to the `packages` directory in one of the `<MIME>` paths (usually `/usr/share/mime/packages`).

Table 5.1, “Child elements of `<mime-info>`” gives a brief description for each of the elements that can occur as children to the `mime-type` element.

**Table 5.1. Child elements of `<mime-info>`**

Element (and attributes)	Description
<code>&lt;glob pattern="*.xyz"&gt;</code>	This element specifies a glob pattern against filenames. If the filename matches, then it is assigned the MIME type of the parent <code>mime-type</code> element. The <code>pattern</code> attribute is mandatory.
<code>&lt;magic priority="50"&gt;</code>	This element contains a list of <code>match</code> elements as its children. The <code>priority</code> attribute is optional, and specifies a priority between 0 and 100, with 100 being the highest matching priority. Each child <code>match</code> element has three required attributes: <code>type</code> , <code>offset</code> , <code>value</code> and a fourth optional attribute, <code>mask</code> . For details on these attributes, see the XDG shared mime info specification [ <a href="http://www.freedesktop.org/Standards/shared-mime-info-spec">http://www.freedesktop.org/Standards/shared-mime-info-spec</a> ].
<code>&lt;alias type="media/subtype"&gt;</code>	This element defines an alias for the parent <code>mime-type</code> element. An alias is simply a MIME type that is sometimes known as another type. For example, <code>application/x-pdf</code> is an alias for the MIME type <code>application/pdf</code> .
<code>&lt;sub-class-of type="media/subtype"&gt;</code>	This element defines the parent <code>mime-type</code> element as a subclass of the MIME type specified in the <code>type</code> attribute. For example, <code>image/svg</code> is a sub class of the MIME type <code>text/xml</code> , <code>text/plain</code> , and <code>application/octet-stream</code> .

Element (and attributes)	Description
<code>&lt;comment xml:lang="locale"&gt;</code>	This element provides a human readable description for the MIME type. There can be zero or more occurrences of this element as long as each one contains a unique value for the <code>xml:lang</code> attribute.
<code>&lt;root-XML namespaceURI="namespace" localName=" " &gt;</code>	If a file is determined to an XML file, then this element helps to further classify it through the use of the <code>namespaceURI</code> and <code>localName</code> attributes, both of which are required. The attribute <code>namespaceURI</code> is the namespace of the document, and <code>localName</code> is the name of the root element for the document. If <code>localName</code> is present but its value is empty, then the root element may have any name, but the namespace must still match.

The easiest way to understand these files is to take a look at an example. Borrowing from the *XDG shared mime specification*, Example 5.2, “Example of a source XML file: `diff.xml`” displays the contents of a source XML file called `diff.xml`. This example defines the MIME type `text/x-diff`. There are multiple comment elements which give a human readable name to the MIME type in a number of different languages. This MIME type has both rules for matching through glob patterns *and* through the use of content “sniffing” (better known as magic rules). Any file with the extension `.diff` or `.patch` will resolve to this MIME type. Additionally any file whose contents start with the strings specified in the value attributes of the `match` element, will resolve to the `text/x-diff` MIME type.

The order in which glob patterns and magic rules apply is beyond the scope of this document. For details on this, see the XDG shared mime info specification [<http://www.freedesktop.org/Standards/shared-mime-info-spec>].

### Example 5.2. Example of a source XML file: `diff.xml`

```
<?xml version='1.0'?>
<mime-info xmlns='http://www.freedesktop.org/standards/shared-mime-info'>
  <mime-type type="text/x-diff">
    <comment>Differences between files</comment>
    <comment xml:lang="af">verskille tussen lêers</comment>
    <!-- more translated comment elements -->
    <magic priority="50">
      <match type="string" offset="0" value="diff\t"/>
      <match type="string" offset="0" value="***\t"/>
      <match type="string" offset="0" value="Common subdirectories: "/>
    </magic>
    <glob pattern="*.diff"/>
    <glob pattern="*.patch"/>
  </mime-type>
</mime-info>
```

## Modifying MIME types

You should never directly modify the source XML files that are installed to the `<MIME>/packages` directory by applications. Instead, modify the `Overrides.xml` file. This file has precedence over all other source XML files installed into the same `packages` directory. If you are an application author, then this rule does not apply. You should create a new source XML file and place it in the proper `<MIME>/packages` directory (your Makefile will take care of this, of course).

You can modify the MIME database for all users on the system or for a particular user depending on the location of the file you change. To modify the database for all users, make changes to the file `Overrides.xml` in the `$XDG_DATA_DIRS/mime/packages` directory. To modify the database for a single user, make changes to the file `Overrides.xml` in the `$XDG_DATA_HOME/mime/packages` directory.

After changes are made, you must always run the `update-mime-database` application, with the directory location of

the MIME database as the first parameter.

## Adding or Modifying MIME types

To add one or more MIME types for all users:

1. Create or modify an existing `Overrides.xml` source XML file, containing the definitions for the MIME types. For more information, see the section called “The source XML files”.
2. Place the `Overrides.xml` file in the `/usr/share/mime/packages` directory.
3. Update the MIME database by running `update-mime-database` using the system account.

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

To add one or more MIME types for a single user, follow the same steps, except place your `Overrides.xml` file in the `~/.local/share/mime/packages` directory. Additionally, call `update-mime-database` with `~/.local/share/mime/packages` as the first parameter.

## Verifying Changes

After you have made a change to the MIME database and refreshed its contents, you can verify that the change has taken effect using the `gnomevfs-info` application. This application prints the MIME type and other useful information about a file.

Running `gnomevfs-info` on a SVG file gives you the output shown below. You'll notice the default application for this MIME type is `eog.desktop`; We will discuss default applications in the section called “Registering Applications for MIME Types”.

```
$ gnomevfs-info mime-diagram.svg
Name           : mime-diagram.svg
Type           : Regular
MIME type      : image/svg+xml
Default app    : eog.desktop
Size           : 14869
Blocks        : 32
I/O block size : 4096
Local          : YES
SUID           : NO
SGID           : NO
Sticky        : NO
Permissions   : 600644
Link count     : 1
UID            : 1000
GID            : 100
Access time    : Wed Feb 22 18:24:47 2006
Modification time : Wed Feb 22 18:24:42 2006
Change time    : Wed Feb 22 18:24:42 2006
Device #      : 775
Inode #       : 297252
Readable      : YES
Writable      : YES
Executable    : NO
$
```

the section called “`application/x-newtype Example`” walks through the steps of creating a new MIME type and then verifying the changes using `gnomevfs-info`.

## `application/x-newtype Example`

To create (or override) a MIME type and verify the changes:

1. Make a new, empty file in your home directory called `testing.xyz`.
2. Use `gnomevfs-info` on the file to find out the MIME type. The MIME type for this file should be detected as

text/plain because there are not any glob patterns or magic rules that match it<sup>1</sup>.

3. Create (or modify) the `Overrides.xml` file as described in the section called “Adding or Modifying MIME types” with the contents given in Example 5.3, “`Overrides.xml` file”.
4. Refresh the database using `update-mime-database`.
5. Use `gnomevfs-info` to verify that your change has taken effect. You should see the MIME type for the `testing.xyz` file resolved as `application/x-newtype`.

```
$ gnomevfs-info testing.xyz | grep MIME
MIME type      : application/x-newtype
$
```

### Example 5.3. `Overrides.xml` file

```
<?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>
```

## Registering Applications for MIME Types

Registering applications to handle MIME types is fairly straightforward. Applications are registered by creating a `MimeType` key in their `.desktop` entry file and listing each MIME type separated by a semicolon. The `MimeType` key should only be used in `.desktop` files whose `Type` key has the value “Application”. For more information on `.desktop` files, see the section called “Desktop Entry Files”.

After creating or modifying a `.desktop` entry file, you must update the application database using the `update-desktop-database` application (very similar to `update-mime-database`, except it does not take a parameter). This will create a `mimeinfo.cache` file in the `applications` subdirectory for each directory in `$XDG_DATA_HOME:$XDG_DATA_DIRS`. The cache file is necessary so that all the `.desktop` files do not need to be scanned for just the `MimeType` key, as this causes unnecessary disk I/O.

Default applications to use for specific MIME types should be specified in a file called `defaults.list`. This file is located in the `applications` subdirectory for each directory in `$XDG_DATA_HOME` and `$XDG_DATA_DIRS`. The format for this file consists of the MIME type, the `=` symbol and the Desktop File ID (which is the filename for the desktop entry file). Example 5.4, “A User's `defaults.list` file” is a short example of a `defaults.list` file in a user's `~/ .local/share/applications` directory.

### Example 5.4. A User's `defaults.list` file

```
[Default Applications]
application/pdf=evince.desktop
text/html=epiphany.desktop
text/plain=gedit.desktop
image/jpeg=eog.desktop
image/png=eog.desktop
text/xml=gedit.desktop
```

<sup>1</sup>When no glob patterns or magic rules match a file, then it is resolved to the MIME type `text/plain` if it contains textual data or `application/octet-stream` for binary data. If the file is empty, then it defaults to `text/plain`.



## XDG Desktop Entry Specification

How to register MIME types for applications is part of the XDG desktop entry specification [<http://www.freedesktop.org/wiki/Standards/desktop-entry-spec>], rather than the XDG shared mime info specification [<http://www.freedesktop.org/Standards/shared-mime-info-spec>].

# Adding an Application to the GNOME Desktop

To add an application to the GNOME Desktop, perform the following steps:

1. Add a menu item for the application. For more information on how to add an item to a menu, see the section called “Editing System Menus”.
2. Add an icon for the application to `/usr/share/icons/theme-name/icon-size/apps`. For more information on icons and themes, see the section called “Installing Icons for Themes”.
3. If the application uses a new MIME type, add a source XML file to the MIME database. For more information, see the section called “Modifying MIME types”.
4. If the application uses a new MIME type, add an icon for the MIME type to `/usr/share/icons/theme-name/icon-size/mimetypes`. For more information on icons and themes, see Chapter 3, *Installing Themes*.
5. To associate the application with a MIME type, include a `MimeType` key in your `.desktop` file. For more information, see the section called “Registering Applications for MIME Types”.

# Chapter 6

---

## Setting Screensavers



This chapter describes how to set preferences for the screensaver. This chapter also provides information on how to modify the displays that are available for the screensaver.

## Introduction to Screensavers

A *screensaver* is an application that replaces the image on a screen when the screen is not in use. The screensaver application for the GNOME Desktop is XScreenSaver. The following sections describe how to set preferences for the XScreenSaver application, and how to modify the displays that are available for the screensaver.

## Setting Screensaver Preferences

Default screensaver preferences are stored in a file that is called XScreenSaver. This file is stored in `/usr/X11R6/lib/X11/app-defaults/XScreenSaver`.

To modify screensaver application preferences, users can use the Screensaver preference tool. When a user modifies the screensaver preferences, the preferences are stored in the home directory of the user, in the `$HOME/.xscreensaver` file. For information on screensaver preferences, see *GNOME 2.6 Desktop User Guide*.

Users can also run the command `/usr/bin/xscreensaver-demo` to open the XScreenSaver dialog.

To set default screensaver preferences for all users, modify the XScreenSaver file. You can also use the XScreenSaver dialog to create a `$HOME/.xscreensaver` file, then copy the file to the location of the XScreenSaver file.

To restore the default settings for a user, delete the `$HOME/.xscreensaver` file from the home directory of the user. If no `$HOME/.xscreensaver` file is present, the default preferences in the XScreenSaver file are used.



The default display behavior of XScreenSaver is to display a blank screen. The blank screen might confuse users. You might want to change this default display behavior.

To activate changes to the screensaver preferences, use the following command to reload screensaver preferences:

```
# xscreensaver-command -restart
```



The `xset` application does not work with XScreenSaver at the time of publication of this guide. To modify screensaver preferences, modify the XScreenSaver file or the `$HOME/.xscreensaver` file.

## Modifying Screensaver Displays

The screensaver application allows users to choose one or more *screensaver displays*. A screensaver display is an application that displays images on the screen of the user when the screen is not in use. The screensaver displays are listed in the XScreenSaver file and in the `$HOME/.xscreensaver` file.

To add a new screensaver display, copy the executable file for the display to the directory where the displays are located. Add the command for the screensaver display to the XScreenSaver file or the `$HOME/.xscreensaver` file. Include any arguments that are required to run the screensaver display on the whole screen, rather than in a window. For example, you might want to include the `-root` option to display the screensaver display on the whole screen.

To disable a screensaver display, add a minus sign (-) at the start of the command for the screensaver display in the preferences file. The following excerpt from a `$HOME/.xscreensaver` file shows a disabled `Qix (solid)` screensaver display:

```
- "Qix (solid)" qix -root -solid -segments 100
```

# **Chapter 7**

---

## **Session Management**

This chapter introduces session management, and describes how to set session defaults. The chapter also contains information on sessions and login scripts.

## Introduction to Sessions

A session occurs between the time that a user logs in to the GNOME Desktop and the time that the user logs out. The session manager starts after the Login Manager authenticates the user. The session manager enables the user to manage the session. For example, a user can save the state of a session and return to that session the next time that the user logs in.

At a minimum, the following applications run in a session:

- The session manager, **gnome-session**.
- The GConf X settings daemon, **gnome-settings-daemon**.
- The gnome-panel application, which runs the panels in the GNOME Desktop.
- The Metacity window manager.

## Setting Session Defaults

The following table lists the files that contain default session information:

<b>File</b>	<b>Description</b>
<code>/usr/share/gnome/default.session</code>	Default session file. Default session details are stored in this file.
<code>\$HOME/.gnome2/session</code>	User session file. When a user modifies the session, the details are stored in this file.

To set default session details for all users, modify the default session file.

To restore the default session settings for a user, delete the session file from the home directory of the user. If no user session file is present, the default settings in `/usr/share/gnome/default.session` are used.

To save the current session as the default session, users can run the **gnome-session-save** command.

# Chapter 8

---

## Help System

This chapter describes the Help system in the GNOME Desktop.

## Introduction to the Help System

The GNOME Desktop displays Help in the Yelp Help browser. The source documents for the Help are XML files. The XML files are written to the DocBook XML Version 4.1.2 document type definition (DTD). The DocBook XML files are converted to HTML, and the HTML is displayed in the Help browser. For more information on DocBook XML, see the following URL:

```
http://www.oasis-open.org/docbook/xml [http://www.oasis-open.org/docbook/xml/]
```

The Help system uses *Open Source Metadata Framework* (OMF) files, and a cataloging system. The next sections describe OMF files and the cataloging system.

## OMF Files

The XML file for each manual has an associated OMF file. The OMF file contains information about the manual that is used by the Help browser. OMF files have a `.omf` extension.

When you install an application, the installation process installs an OMF file to a directory of OMF files. If a Help document has an associated OMF file in an OMF file directory, the Help document is displayed in the Help browser.

OMF files contain information about Help documents such as the following:

- The location of the XML file for the Help document
- The title of the Help document
- A subject category to which the Help document belongs

## ScrollKeeper Cataloging System

ScrollKeeper is a cataloging system for documentation. The Help browser uses ScrollKeeper to catalog the manuals on the system. ScrollKeeper manages the information in the OMF files for the Help browser.

ScrollKeeper uses the information in the OMF files to enable the Help browser to find Help documents on a system. When you install an application, the installation process installs an OMF file to a directory of OMF files.

ScrollKeeper builds a table of contents for each Help document that has an associated OMF file. ScrollKeeper builds the table of contents from the XML file specified in the OMF file. The table of contents for the Help document is displayed in the left pane of the Help browser window, and enables users to navigate the Help document in the left pane. The title of the document in the OMF file is also displayed in the left pane of the Help browser window.

ScrollKeeper contains a hierarchy of subject categories to which a Help document can belong. The Help documents are organized into this hierarchy of subject categories in the left pane of the Help browser window. The subject category of the document in the OMF file determines the location of the document in the hierarchy. The hierarchy of subject categories is defined in the following file:

```
/usr/share/scrollkeeper/Templates/C/scrollkeeper_cl.xml
```

The subject category of a document must correspond to a subject category in the following file:

```
/usr/share/scrollkeeper/Templates/C/scrollkeeper_cl.xml
```



If the Help document XML file is moved to a new location, then you might need to update the location in the OMF file.

# **Chapter 9**

---

## **Improving Performance**



This chapter describes how to improve the performance of the GNOME Desktop.

## Introduction to Improving Performance

This chapter lists several preferences whose settings you can change to improve the performance of the GNOME Desktop. You can use the **gconftool-2** command to set values for preferences for users. The example commands in this chapter show how to set values in the user configuration source.

You can also use the `--direct` and `--config-source` options to set mandatory values or default values for preferences. You can also use the **gconftool-2** command in a script to set the values of many preferences. For more information on the **gconftool-2** command and the options that are available with the command, see Chapter 1, *Using GConf*.

## Reducing CPU Usage

This section describes preferences that you can set to reduce CPU usage by the GNOME Desktop.

### To Use Theme Options Which Require Less CPU Resources

Some window frame theme options load image files to draw the window frame. Other options use simpler techniques to draw the window frame.

The Crux window frame option loads image files, and can be slow on systems with limited CPU resources. To reduce CPU usage, use one of the following window frame options:

- Atlanta
- Esco

The following window frame options also use less CPU resources than Crux:

- AgingGorilla
- Bright
- Metabox



Metabox does not work well with inverse controls options such as `HighContrastInverse`. Use Atlanta with inverse controls options.

To change the window frame theme option, run the following command:

```
# gconftool-2 --type string --set /apps/metacity/general/theme option-name
```

For example, to use Atlanta, run the following command:

```
# gconftool-2 --type string --set /apps/metacity/general/theme Atlanta
```

Alternatively, users can use the Theme preference tool to select the appropriate option.



You can use the Metacity Theme Viewer to measure the performance of a window frame option, and to preview the option. To start Metacity Theme Viewer, use the following command:

```
# metacity-theme-viewer option-name
```

For example, to measure the performance of Atlanta and preview Atlanta, run the following command:

```
# metacity-theme-viewer Atlanta
```

## To Reduce CPU Usage by Turning Off Display of Icons in Menus

Some items in menus display an icon beside the item. To turn off this feature, run the following command:

```
# gconftool-2 --type bool --set /desktop/gnome/interface/menus_have_icons false
```

Alternatively, users can use the Menus & Toolbars preference tool to deselect the Show icons in menu option.

## To Reduce CPU Usage by Turning Off the Splash Screen

By default, when users log in to the desktop environment, a splash screen is displayed. Icons are displayed on the splash screen while the user logs in. You can turn off the splash screen to reduce CPU usage during login.

To turn off the splash screen, run the following command:

```
# gconftool-2 --type bool --set /apps/gnome-session/options/show_splash_screen false
```

Alternatively, users can use the Sessions preference tool to deselect the Show splash screen on login option.

## To Reduce CPU Usage by Turning Off Panel Animation

When users show or hide panels, the panels can show or hide in an animated style. To turn off panel animation, run the following command:

```
# gconftool-2 --type bool --set /apps/panel/global/enable_animations false
```

Alternatively, users can use the Panel preference tool to deselect the Drawer and panel animation option.

## Improving File Manager Performance

The Nautilus file manager includes some features that you can modify to improve performance.

### *To Modify Performance Preferences*

The file manager includes performance-related preferences. Each performance preference can take one of three values. The values are described in the following table:

Value	Description
always	Performs the action for both local files, and files on other file systems.
local_only	Performs the action for local files only. When you set a performance preference to <code>local_only</code> , the CPU usage is reduced.
never	Never performs the action. When you set a performance preference to <code>never</code> , the CPU usage and the network traffic are reduced.

Value	Description
-------	-------------

The following table describes the performance preferences for the file manager. For the fastest performance, set the value of the preferences to never.

Preference	Description
<code>show_icon_text</code>	<p>Specifies when to preview the content of text files in the icon that represents the file. To never preview the content of text files, run the following command:</p> <pre># gconftool-2 --type string --set / apps/nautilus/preferences/ show_icon_text never</pre> <p>Alternatively, users can perform the following steps:</p> <ol style="list-style-type: none"> <li>1. Choose Edit→Preferences from a file manager window, then choose Preview.</li> <li>2. Select an option for the Show text in icons preference.</li> </ol>
<code>show_directory_item_counts</code>	<p>Specifies when to show the number of items in folders. To never show the number of items in folders, run the following command:</p> <pre># gconftool-2 --type string --set / apps/nautilus/preferences/ show_directory_item_counts never</pre> <p>Alternatively, users can perform the following steps:</p> <ol style="list-style-type: none"> <li>1. Choose Edit→Preferences from a file manager window, then choose Preview.</li> <li>2. Select an option for the Count number of items preference.</li> </ol>
<code>show_image_thumbnails</code>	<p>Specifies when to show thumbnails of image files. To never show thumbnails, run the following command:</p> <pre># gconftool-2 --type string --set / apps/nautilus/preferences/ show_image_thumbnails never</pre> <p>Alternatively, users can perform the following steps:</p> <ol style="list-style-type: none"> <li>1. Choose Edit→Preferences from a file manager window, then choose Preview.</li> <li>2. Select an option for the Show thumbnails preference.</li> </ol>
<code>preview_sound</code>	<p>Specifies when to preview the content of sound files. To never preview the content of sound files, run the following command:</p> <pre># gconftool-2 --type string --set / apps/nautilus/preferences/preview_sound</pre>

Preference	Description
	<b>never</b>
	Alternatively, users can perform the following steps:
	1. Choose Edit→Preferences from a file manager window, then choose Preview.
	2. Select an option for the Preview sound files preference.

### *To Turn Off the Side Pane, Toolbar, and Location Bar*

The file manager includes preferences that enable you to turn off the side pane and the toolbar. To improve file manager performance, turn off the side pane and the toolbar.

To turn off the side pane, run the following command:

```
# gconftool-2 --type bool --set /apps/nautilus/preferences/start_with_sidebar false
```

To turn off the toolbar, run the following command:

```
# gconftool-2 --type bool --set /apps/nautilus/preferences/start_with_toolbar false
```

You can also turn off the location bar. Users can use the **Ctrl-L** keyboard shortcut to display a location bar when required.

To turn off the location bar, run the following command:

```
# gconftool-2 --type bool --set /apps/nautilus/preferences/start_with_location_bar false
```

### *To Turn Off the Desktop*

The file manager contains a preference that enables users to use Nautilus to manage the desktop. You can disable the desktop to improve performance. However, if you disable the desktop, you cannot do the following:

- Use the Desktop menu.
- Use the file manager to change the pattern or color of the desktop background.
- Use the desktop objects, such as Trash. The desktop objects are not displayed on the desktop.

To disable the desktop, run the following command:

```
# gconftool-2 --type bool --set /apps/nautilus/preferences/show_desktop false
```

## Reducing X Window System Network Traffic

This section describes preferences that you can set to reduce X Window System network traffic on the GNOME Desktop.

### To Use Theme Options Which Create Less Network Traffic

Remote display protocols do not transfer every pixel in a block of pixels if all pixels in the block are the same color.

To reduce X Window System network traffic, use a window frame theme option that uses solid colors. That is, use one of the following window frame options:

- Atlanta
- Esco

For information on how to change theme options, see the section called “To Use Theme Options Which Require Less CPU Resources”.

## To Reduce Network Traffic by Turning Off Display of Icons in Menus

Some items in menus display an icon beside the item. If the icon is located on another file system, this feature can increase X Window System network traffic. This feature can also increase X Window System network traffic if the panels are displayed on a remote host.

For information on how to turn off this feature, see the section called “To Reduce CPU Usage by Turning Off Display of Icons in Menus”.

# Reducing Color Usage and Improving Display Quality

Many modern computer systems support 24-bit color, that is, 16,777,216 colors. However, many users still use systems that support 8-bit color, that is, 256 colors.

The GNOME Desktop uses the *websafe color palette*. This palette is a general-purpose palette of 216 colors, which is designed to optimize the use of color on systems that support 8-bit color. However, some visual components of the GNOME Desktop are designed for systems that support 24-bit color.

The following display problems might occur on systems that support 8-bit color:

- Windows, icons, and background images might appear grainy. Many themes, background images, and icons use colors that are not in the websafe color palette. The colors that are not in the palette are replaced with the nearest equivalent or a dithered approximation. This use of replacement colors causes the grainy appearance.
- Applications that do not use the websafe color palette have less colors available. Color errors might occur. Some colors might not appear in the user interface of the application. Some applications might crash if the application cannot allocate colors.
- Color flashing might occur when users switch between applications that use the websafe color palette, and applications that do not use this palette. The applications that do not use the websafe color palette might use a custom colormap. When the custom colormap is used, other visual components might lose colors, then become unviewable.

The following sections describe how to optimize the appearance of the GNOME Desktop for systems that support 8-bit color.

## To Use Theme Options That Use the Websafe Color Palette

Some window frame theme options use colors that are in the websafe color palette. Bright and Esco use colors from the websafe color palette. Bright and Esco do not have the grainy appearance of other window frame options on 8-bit color displays. Use Bright or Esco for the best color display on 8-bit visual modes.

For information on how to change theme options, see the section called “To Use Theme Options Which Require Less CPU Resources”.

## To Reduce Color Usage by Turning Off Display of Icons in Menus

Some items in menus display an icon beside the item. If the icon contains colors that are not in the websafe color palette, this feature can increase the number of colors used.

For information on how to turn off this feature, see the section called “To Reduce CPU Usage by Turning Off Display of Icons in Menus”.

## To Reduce Color Usage by Turning Off the Splash Screen

You can turn off the splash screen to make more colors available for the GNOME Desktop and for applications.

For information on how to turn off the splash screen, see the section called “To Reduce CPU Usage by Turning Off the Splash Screen”.

## To Reduce Color Usage by Using a Solid Color for the Background

Use a solid color for the desktop background. Use of a solid color reduces the number of colors used by the GNOME Desktop.

To set a solid color for the background, run the following commands:

```
# gconftool-2 --type string --set /desktop/gnome/background/picture_options  
none  
  
# gconftool-2 --type string --set /desktop/gnome/background/color_shading_type  
solid  
  
# gconftool-2 --type string --set /desktop/gnome/background/primary_color  
\#hexadecimal-color
```

Alternatively, users can use the Background preference tool to choose a solid color for the background.

# Chapter 10

---

## Disabling GNOME Desktop Features

This chapter describes how to disable particular features of the GNOME Desktop.

## Introduction to Disabling GNOME Desktop Features

The GNOME Desktop includes features that you can use to restrict access to certain functions in the GNOME Desktop. The disable features are useful in various situations where you want to restrict the actions that users can perform on a computer. For example, you might want to prevent command line operations on a computer that is for public use at a trade show. The disable features are also known as *lockdown* features.

You set GConf keys to disable features. For information about how to set GConf keys, see Chapter 1, *Using GConf*. You can also use the Configuration Editor application to set GConf keys in a user configuration source. For more information about the Configuration Editor application, see the *GConf Editor Manual*.

## To Disable Lock Screen and Log Out

To disable the lock screen and log out functions, set the `/apps/panel/global/disable_lock_screen` key and the `/apps/panel/global/disable_log_out` key to `true`.

When you disable the lock screen and log out functions, the following items are removed from the panels:

- Lock Screen and Log Out *user* menu items from the Main Menu.
- Lock and Log Out menu items from the Add to Panel→Actions menu. To open this menu, right-click on a vacant space on a panel, then choose Add to Panel→Actions.
- Lock Screen and Log Out *user* menu items from the Actions menu in the Menu Bar applet.

Also, any Lock Screen buttons and Log Out buttons on panels are disabled.

## To Disable Command Line Operations

To disable operations from a command line, set the `/desktop/gnome/lockdown/disable_command_line` key to `true`.

When you disable command line operations, the following changes occur in the user interface:

- The Run Application menu item is removed from the following menus:
  - Main Menu
  - Actions submenu in the Add to Panel menu
  - Actions menu in the Menu Bar applet
- Any Run buttons on panels are disabled.

To disable command line operations, you must also remove menu items that start terminal applications. For example, you might want to remove menu items that contain the following commands from the menus:

- GNOME Terminal command, that is `/usr/bin/gnome-terminal`
- `/usr/bin/xterm`
- `/usr/bin/setterm`



The items are removed from the following menus:

- Main Menu
- Add to Panel→Launcher from menu

To disable command line operations, you must also disable the Command Line applet. To disable the Command Line applet, add the applet to the `/apps/panel/global/disabled_applets` key. When you disable the Command Line applet, the Command Line applet is removed from the Main Menu and the Add to Panel→Utility menu.

## To Disable Panel Configuration

To disable panel configuration, set the `/apps/panel/global/locked_down` key to `true`.

When you disable panel configuration, the following changes occur in the user interface:

- The following items are removed from the panel popup menu, and from the drawer popup menu:
  - Add to Panel
  - Delete This Panel
  - Properties
  - New Panel
- The launcher popup menu is disabled.
- The following items are removed from the applet popup menu:
  - Remove From Panel
  - Lock
  - Move
- The Main Menu popup menu is disabled.
- The launcher drag feature is disabled, so that users cannot drag launchers to, or from, panels.
- The panel drag feature is disabled, so that users cannot drag panels to new locations.

# Appendix A. Hidden Directories

This appendix describes the hidden directories that the GNOME Desktop adds to the home directories of users.

A hidden directory is a directory that has a name that begins with a period (.). Table A.1, “Hidden Directories in User Home Directories” describes the hidden directories that the GNOME Desktop adds to the home directories of users.

**Table A.1. Hidden Directories in User Home Directories**

Directory	Description
<code>.esd_auth</code>	Contains the authentication cookie for the GNOME sound daemon. The GNOME sound daemon is the Enlightened Sound Daemon (ESD).
<code>.gconf</code>	Contains the GConf configuration source for the user. When the user sets a preference, the new preference information is added to this location.
<code>.gconfd</code>	Contains the following GConf daemon details: <ul style="list-style-type: none"> <li>• Configuration information.</li> <li>• Lock information for objects that are referenced by an <i>Interoperable Object Reference (IOR)</i>.</li> <li>• State information for objects that are referenced by an IOR.</li> </ul>
<code>.gnome</code>	Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains MIME type information and session information for the user.
<code>.gnome-desktop</code>	The Nautilus file manager contains a preference that enables users to use the file manager to manage the desktop. If this option is selected, this directory contains the objects that are on the desktop of the user. This directory contains the following: <ul style="list-style-type: none"> <li>• Objects on the desktop, for example, the Home object, the Trash object, and other launchers. The objects appear in the directory as desktop entry files. For example, the <code>starthere.desktop</code> file contains a link to the Start Here location.</li> <li>• Removable media volumes that are mounted.</li> </ul> <p>The file manager also contains a preference that enables users to use the home directory as the desktop directory, instead of <code>.gnome-desktop</code>. If a user selects this option, the contents of the home directory are displayed as desktop objects.</p>

<b>Directory</b>	<b>Description</b>
<code>.gnome2</code>	<p>Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains the following:</p> <ul style="list-style-type: none"><li>• Keyboard shortcut information.</li><li>• Window location information.</li><li>• Desktop entry files for panel launchers.</li></ul> <p>This directory also contains user-specific menu data. If a user modifies menus, the details are stored here.</p>
<code>.gnome2-private</code>	<p>Ignore this directory. This directory has no function at the time of publication of this guide.</p>
<code>.metacity</code>	<p>Contains session data for the Metacity window manager.</p>
<code>.nautilus</code>	<p>Contains file manager data that is specific to the user. For example, this directory can contain the following:</p> <ul style="list-style-type: none"><li>• Metadata for the directories with which the user works.</li><li>• Nautilus emblems that the user adds.</li><li>• Nautilus desktop images.</li></ul>
<code>.themes</code>	<p>Contains controls theme options, window frame theme options, and icons theme options that the user adds. The user can add themes from the Theme preference tool.</p>
<code>.thumbnails</code>	<p>Contains image thumbnails for the user. The image thumbnails are used in the file manager. The file manager contains a preference that the user can select to stop generation of thumbnail images.</p>
<code>.xscreensaver</code>	<p>Contains screensaver configuration data and screensaver preference data.</p>

# Appendix B. Security Note on Configuring SMB Printers

This appendix provides a security note related to the configuration of printers on Windows network shares.

Windows network shares are also referred to as Samba or *SMB* shares. When you configure a printer on an SMB share, you must enter a username and password for the print queue.

The username and password for the print queue is stored as unencrypted text in the file `/etc/cups/printers.conf`. This file has read-only permissions for users with root privileges, so any user with root privileges can read the username and password for the print queue.

To reduce the impact of possible security violations, ensure that the username and password required to access the print queue is used only for the print queue. This ensures that any possible security violation is restricted to unauthorized use of the print queue.

# Glossary

<code>.desktop</code> file	See desktop entry file.
<code>.directory</code> file	See directory entry file.
<code>.omf</code> file	See OMF file.
application registry	The application registry is a location that contains text files which register applications. The location of the application registry for the GNOME Desktop is <code>/usr/share/gnome/application-registry</code> .
desktop entry file	A data file that provides information about an item in a menu. The desktop entry file specifies the details for the item such as a name, a command to run, an icon, and so on. Desktop entry files have a <code>.desktop</code> file extension.
directory entry file	A data file that provides information about a menu. The directory entry file specifies details such as a name for the menu, a tooltip for the menu, and an icon to represent the menu. Directory entry files have a <code>.directory</code> file extension.
file content sniffer	A file content sniffer specifies a pattern to search for in a file. A file content sniffer associates the pattern with a MIME type. If a match for the pattern is found, the MIME type associated with the pattern is the MIME type of the file.
GConf configuration source	A storage location in the GConf repository. For example, <code>xml:readonly:/etc/gconf/gconf.xml.defaults</code> .
GConf preference key	An element in the GConf repository that corresponds to an application preference. For example, the <code>/apps/gnome-session/options/show_splash_screen</code> key corresponds to the Show splash screen on login option in the Sessions preference tool.
GConf path file	A file that lists the GConf configuration sources, and the order in which to search the sources.
GConf schema	A collective term for a schema key and a schema object.
GConf schema definition file	A GConf schema definition file lists the keys in a particular application, and defines the characteristics of the keys. GConf schemas are generated from schema definition files. Schema definition files have a <code>.schemas</code> file extension.
GConf schema key	A key that stores a schema object for a preference key. For example, <code>/schemas/desktop/gnome/interface/font_name</code> is a schema key for the <code>/desktop/gnome/interface/font_name</code> preference key.
GConf schema object	An element in a configuration source that contains information about a preference key. The schema object contains information such as a default value for the preference key, and documentation on the preference key.
Interoperable Object Reference	An Interoperable Object Reference (IOR) is a string reference to a CORBA object. An IOR encodes a hostname and port to which messages can be sent to control the object. The IOR also contains an object key to identify the object.
MIME	Multipurpose Internet Mail Extension.
MIME information file	A MIME information file is a text file that associates MIME types with file ex-

	tensions and filename patterns. MIME information files have a <code>.mime</code> file extension.
MIME keys file	A MIME keys file provides information about a MIME type that is used in the user interface. For example, the MIME keys file specifies an icon to represent files of that MIME type. MIME keys files have a <code>.keys</code> file extension.
MIME type	A MIME type identifies the format of a file. The MIME type enables applications to read the file. For example, an email application can use the <code>image/png</code> MIME type to detect that a Portable Networks Graphic (PNG) file is attached to an email.
MIME type registry	The MIME type registry is a location that contains text files which register MIME types for the desktop environment. The location of the MIME type registry for the GNOME Desktop is <code>/usr/share/mime-info</code> .
OMF file	Open Source Metadata Framework file. This is a file that is associated with the XML file for a manual. The OMF file contains information about the manual that is used by the Help browser. OMF files have a <code>.omf</code> file extension.
PAM	Pluggable Authentication Modules.
pattern mask	A pattern mask is a series of hexadecimal characters in a file content sniffer. The pattern mask identifies bits in the pattern to ignore when searching for a pattern in a file.
screensaver	A screensaver is an application that replaces the image on a screen when the screen is not in use. The screensaver application for the GNOME Desktop is <code>XScreenSaver</code> .
screensaver display	A screensaver display is an application that displays images on the screen of the user when the screen is not in use.
UDP	User Datagram Protocol.
Uniform Resource Identifier	A Uniform Resource Identifier (URI) is a string that identifies a particular location in a file system or on the web. For example, the address of a web page is a URI.
vfolder	A virtual representation of items that reside in a physical location or physical locations on your system. For example, a vfolder might represent the contents of several directories. In terms of menus, a vfolder is a representation in a menu of items that might be physically located in several directories.
vfolder information file	An XML file that describes a vfolder. Vfolder information files specify the structure of your menus.
websafe color palette	The websafe color palette is a general-purpose palette of 216 colors. The websafe color palette is designed to optimize the use of color on systems that support 8-bit color. The websafe color palette is also called the <i>Netscape color palette</i> and the <i>Netscape color cube</i> .

# Index

## Symbols

- \* directories (see hidden directories)
- .desktop files (see desktop entry files)
- .directory files (see directory entry files)
- .menu files (see menu definition files)
- .omf files (see OMF files)

## A

- applications
  - adding, 40
  - registry, 39

## B

- background
  - preferences, setting with GConf, 13
  - using solid color, 55

## C

- color usage
  - reducing, 54
- configuring fonts (see fonts)
- CPU usage
  - reducing, 50

## D

- desktop
  - font preferences, setting with GConf, 13
  - turning off, 53
- desktop entry files, 18
- directory entry files, 20
- disabling features
  - command line, 57
  - introduction, 57
  - lock screen, 57
  - log out, 57
  - panel configuration, 58
- display quality, improving, 54

## F

- file manager
  - desktop, turning off, 53
  - location bar, turning off, 53
  - modifying performance preferences, 51
  - side pane, turning off, 53
  - toolbar, turning off, 53
- fontconfig (see fonts)
- fonts
  - adding for all users, 30
  - adding for individual users, 30
  - introduction, 30
  - setting with GConf, 13

substitution, 30

## G

- GConf
  - background, setting, 13
  - command line tool, 6
  - configuration sources, 3
  - daemon, 5
  - fonts, setting, 13
  - HTTP proxy, setting, 8
  - introduction, 2
  - keyboard accessibility, setting, 9
  - keyboard shortcuts, setting, 10
  - number of workspaces, setting, 9
  - panel object preferences, setting, 10
  - panel preferences, setting individual , 10
  - preference values, setting, 8
  - print manager, setting, 9
  - repository, 2
  - restoring default preference values, 14
  - schema definition files
    - introduction, 5
    - panel and panel object, 10
  - schema keys, 4
  - schema objects, 4
  - schemas, 4
  - splash image, setting, 13
- gconfd (see GConf daemon)
- gconftool-2 (see GConf command line tool)
- GTK+ themes (see themes, controls options)

## H

- Help
  - introduction, 47
  - OMF files, 47
  - ScrollKeeper cataloging system, 47
- hidden directories, 59
- HTTP proxy
  - setting with GConf, 8

## I

- icons, in menus (see menu icons)
- icon themes (see themes, icons options)

## K

- keyboard
  - setting accessibility preferences with GConf, 9
  - setting shortcut preferences with GConf, 10

## L

- lockdown (see disabling features)

## M

- menu definition files, 16
- menu icons

- and color usage, 55
- and CPU usage, 51
- and network traffic, 54
- turning off, 51
- menus
  - adding, 20
  - adding items to, 21
  - deleting menu items, 21
  - editing, 20
  - editing menu items, 21
  - editing properties of, 21
  - editing user menus, 22
  - introduction, 16
  - merging arbitrary menus, 23
  - merging the system menu, 23
- Metacity themes (see themes, window frame options)
- MIME types
  - adding applications, 40
  - database, 33
  - introduction, 33
  - modifying, 37
  - registering applications for, 39
  - source XML files, 36
  - verifying changes, 37

## N

- Nautilus (see file manager)
- network traffic, reducing, 53

## O

- OMF files, 47
- online Help (see Help)

## P

- panels
  - animation, turning off, 51
  - preferences, individual, setting with GConf, 10
- performance, improving
  - introduction, 50
  - reducing CPU usage, 50
- preference values
  - restoring to default with GConf, 14
  - setting with GConf, 8
- print manager
  - setting with GConf, 9

## R

- reducing
  - color usage, 54
  - CPU usage, 50
  - X Window System network traffic, 53

## S

- Samba (see SMB)
- schemas
  - description, 4

- schema definition files
  - introduction, 5
  - panel and panel object, 10
  - schema keys, 4
  - schema objects, 4
- screensavers
  - introduction, 42
  - modifying screensaver displays, 42
  - setting preferences, 42
- ScrollKeeper cataloging system, 47
- sessions
  - introduction, 45
  - setting defaults, 45
- SMB
  - configuring printers, 61
- splash screen
  - and color usage, 55
  - and CPU usage, 51
  - image, setting with GConf, 13
  - turning off, 51

## T

- themes
  - and color usage, 54
  - and CPU usage, 50
  - and network traffic, 53
  - controls options
    - creating custom, 28
    - installing, 26
    - introduction, 25
  - icons options
    - installing, 27
    - introduction, 25
  - installing icons, 27
  - introduction, 25
  - window frame options
    - and color usage, 54
    - and CPU usage, 50
    - and network traffic, 53
    - changing, 50
    - installing, 26
    - introduction, 25

## W

- Windows network shares (see SMB)
- workspaces
  - setting number with GConf, 9

## X

- XScreenSaver (see screensavers)
- X Window System network traffic, reducing, 53