GNOME 2.2 Desktop on Linux System Administration Guide

Adobe PostScript™

031020@6671

Contents

Preface

GNOME 2.2 Desktop on Linux System Administration Guide provides information on how to administer a system running the GNOME 2.2 Desktop on the Linux operating system.

Who Should Use This Book

This book is for system administrators who are responsible for administering one or more systems that are running the GNOME 2.2 Desktop. For information on how to use the GNOME 2.2 Desktop, see *GNOME 2.2 Desktop on Linux User Guide*.

Before You Read This Book

Before you read this book, you should ensure that you have some familiarity with the following topics:

- UNIX® system administration
- Structure of Extensible Markup Language (XML) files
- How to use XML files in the context of system administration

How This Book Is Organized

This book is organized as follows:

- Chapter 1 describes how to use GConf to manage user preferences.
- Chapter 2 describes the implementation of menus and how to customize menus.
- Chapter 3 describes the types of theme that are available in the GNOME Desktop, how to install themes, and how to create a custom theme.
- Chapter 4 describes how applications detect MIME types, how to register MIME types, and how to add applications to the GNOME Desktop.
- Chapter 5 describes how to configure GDM.
- Chapter 6 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.
- Chapter 7 introduces session management, and describes how to set session defaults. This chapter also contains information on sessions and login scripts.
- Chapter 8 describes the Help system in the GNOME Desktop.
- Chapter 9 describes how to improve the performance of the GNOME Desktop.
- Appendix A describes the hidden directories that the GNOME Desktop adds to the home directories of users.
- Glossary is a list of words and phrases found in this book and their definitions.

Accessing Sun Documentation Online

The docs.sun.com^SM Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

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.

Introducing 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>

Note – 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. Each line in the path file specifies the following information:

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

By default, the contents of the GConf path file are as follows:

xml:readonly:/etc/gconf/gconf.xml.mandatory
include "$(HOME)/.gconf.path"
xml:readwrite:$(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 .gconf 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` instruction in the GConf path file enables users to use another configuration source. To use another configuration source, the user must specify the location of the configuration source. The user specifies the location of the configuration source in the home directory, in a file that is called `.gconf.path`.

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: ■ The name of the application that uses the preference key. ■ The type of value required for the preference key, for example integer, boolean, and so on. ■ A default value for the preference key. ■ Brief documentation on the preference key.

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

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

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 Internet 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 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

Table 1–1 lists some of the options that you can use with the `gconftool-2` command.

TABLE 1–1 gconftool-2 Command Options

Option	Function
--all-dirs	Lists all subdirectories in a directory that you specify.

TABLE 1–1 gconftool-2 Command Options   *(Continued)*

Option	Function
`--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.

TABLE 1–1 gconftool-2 Command Options   *(Continued)*

Option	Function
`--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.

TABLE 1–1 gconftool-2 Command Options   *(Continued)*

Option	Function
--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.

Caution – 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 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 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 "To Set Preferences for Individual Panels and Panel Objects" on page 25.

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>
  .
  .
  .
  </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>
.
.
.
  <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/profiles/default >**
   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.

Using Configuration Editor

Configuration Editor enables you to do the following:

- Modify the values of the keys in your user configuration source.
- View documentation on the keys in your user configuration source.

To open Configuration Editor, choose Applications → System Tools → Configuration Editor. Figure 1–1 shows the Configuration Editor window.
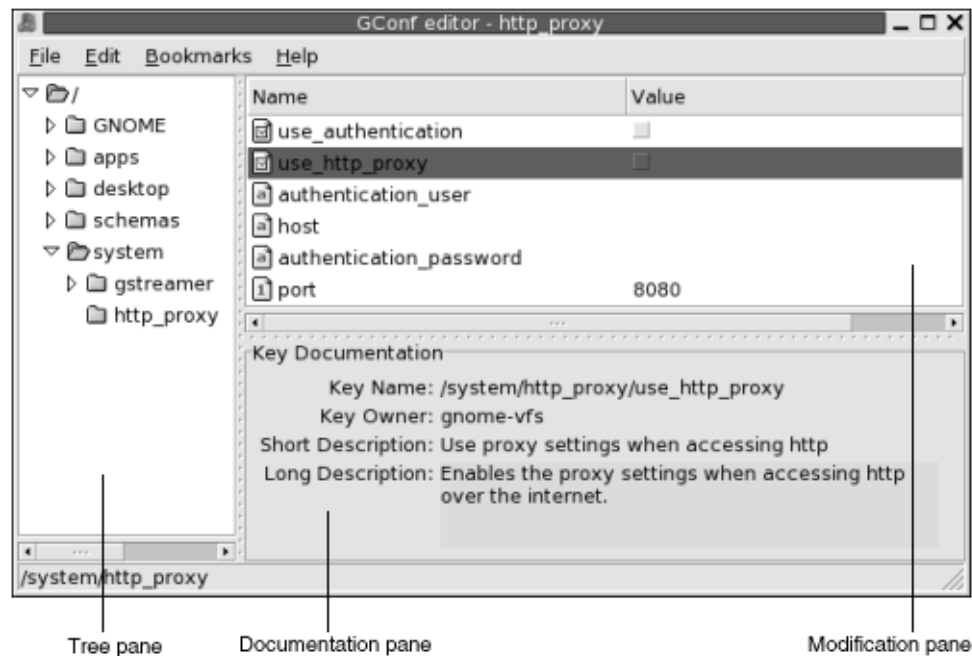
FIGURE 1–1 Configuration Editor Window

The Configuration Editor window contains the following panes:

Tree pane
: Enables you to navigate the directories and subdirectories in the GConf repository. Use this pane to display the keys that you want to modify in the modification pane. The tree pane is on the left side of the window.

Modification pane
: Displays the keys in the selected GConf repository directory. Use this pane to select keys that you want to modify, and to modify the values of keys. The modification pane is in the upper part of the right side of the window.

The icons beside the keys in the modification pane indicate what type of value you can enter for the key. For example, the check mark icon beside the `/system/http_proxy/use_http_proxy` key indicates that you can enter a boolean value for the key.

The icons also indicate if you cannot edit the value of a key. For example, a key icon beside a schema key indicates that you cannot modify the value of the key.

Documentation pane   Displays documentation for the currently selected key. Use this pane to read more information on the GConf preference keys.

You can copy the names of keys so that you can paste the name of the key into another application. You can also add bookmarks to keys.

The following sections describe how to use the Configuration Editor.

To Modify the Value of a Key

To modify the value of a key, perform the following steps:

1. Use the tree pane to display the key that you want to modify in the modification pane.
2. Select the key to modify.
3. To change the value of an integer key or a string key, click in the **Value** column of the key. Type the new value for the key.

    To change the value of a boolean key, click in the **Value** column of the key.

To Copy a Key Name

To copy a key name, select the key whose name you want to copy in the modification pane. Choose Edit → Copy key name. If required, you can paste the name of the key into another application.

Using Bookmarks With Keys

To access a key in your bookmarks, choose the key from the Bookmarks menu. You can add keys to your Bookmarks menu.

To Add a Bookmark

To add a bookmark, select the key that you want to bookmark in the modification pane. Choose Bookmarks → Add bookmark.

To Delete a Bookmark

To delete a bookmark, choose Bookmarks → Edit bookmarks. An **Edit bookmarks** dialog is displayed. Select the bookmark in the dialog, then click **Delete**.

Customizing Menus

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

- "Introduction to Menus" on page 31
- "File Abstraction Layer" on page 32
- "Vfolders and Menus" on page 32
- "Desktop Entry Files" on page 34
- "Directory Entry Files" on page 35
- "Editing Menus" on page 36
- "To Configure Menus That Users Cannot Modify" on page 39

Introduction to Menus

The way in which the GNOME Desktop implements menus enables you to do the following:

- Customize the menu hierarchy easily. The menu hierarchy is not based on the file system hierarchy. 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:

- File abstraction layer
- Vfolders
- Desktop entry files
- Directory entry files

File Abstraction Layer

The `gnome-vfs` file abstraction layer provides a simplified and generalized way for applications to interact with files. The file abstraction layer also provides *Uniform Resource Identifier* (URI) locations that map to particular menu configuration files. To add a menu or a menu item for all users, you must add the menu or menu item to one of the URI locations. Table 2–1 lists the menus to which you can add items, and the URI locations that correspond to the menus.

TABLE 2–1 Menus and URI Locations

Menu	URI Locations
Applications menu for all users	`applications-all-users:///`
Preferences menu for all users	`preferences-all-users:///`

Vfolders and Menus

In general terms, a *vfolder* is 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. A vfolder is an abstraction from one or more physical locations. In terms of menus in the GNOME Desktop, a vfolder is a representation in a menu of items that might be physically located in several directories.

A *vfolder information file* is an XML file that describes a vfolder. Vfolder information files specify the structure of your menus. Vfolder information files specify the names of your menus, and the order in which applications appear in your menus. Vfolder information files have a `.vfolder-info` file extension.

The following is an excerpt from a vfolder information file:

```
<?xml version="1.0"?>
<VFolderInfo>
.
.
.
 <Folder>
   <Name>Applications</Name>
   <Desktop>Applications.directory</Desktop>
   <Folder>
```

```
      <Name>Accessories</Name>
      <DontShowIfEmpty/>
      <Desktop>Accessories.directory</Desktop>
      <Query>
       <And>
         <Keyword>Application</Keyword>
         <Keyword>Utility</Keyword>
       </And>
      </Query>
     </Folder>
.
.
.
 </Folder>
</VFolderInfo>
```

Table 2–2 describes some of the elements in vfolder information files.

TABLE 2–2 Vfolder Information File Elements

Element	Description
`<Folder>`	Contains the elements that define the name, content, and structure of the menu.
`<Name>`	Specifies the name of the menu.
`<Desktop>`	Specifies the name of the directory entry file that specifies the name, comment, and icon for the menu.
`<Query>`	Specifies a query to run on desktop entry files. If a desktop entry file matches the requirements in the query, the menu item is displayed in the menu.
	The query in the excerpt searches for desktop entry files that contain the keywords `Application` and `Utility` in the `Categories` key. Desktop entry files that match are displayed in the Applications menu.
	This element is optional.
`<DontShowIfEmpty/>`	If this element is present, the menu is not displayed if the menu does not contain any items.
	This element is optional.

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. The desktop entry file also contains keywords which determine the location of the item in the menu hierarchy. Desktop entry files have a `.desktop` file extension.

The following is a sample desktop entry file:

```
[Desktop Entry]
Encoding=UTF-8
Name=Calculator
Comment=Perform calculations
Exec=gcalctool
Icon=gcalctool.png
Terminal=false
Type=Application
Categories=GNOME;Application;Utility;
X-GNOME-DocPath=gcalctool/gcalctool.xml
```

Table 2–3 describes the most important keys in desktop entry files.

TABLE 2–3 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 filename extension.
Terminal	Specifies whether the command in the `Exec` key runs in a terminal window. If the value is `true` 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 `true`.

TABLE 2–3 Desktop Entry Keys   *(Continued)*

Desktop Entry Key	Description
Type	Specifies the type of item. This value is one of the following: ■ `Application`: Enter this option for an item that starts an application. ■ `Link`: Enter this option for an item that links to a file, folder, or FTP site.
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 following URL: http://www.freedesktop.org/standards/VFolderDesktops.txt The vfolder information files map the keywords to menus.
X-GNOME-DocPath	Specifies the help file to display when you choose Help on *application-name* from the menu item popup menu.

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

http://www.freedesktop.org/standards/desktop-entry-spec/desktop-entry-spec.html

Note – 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.

The following is a sample directory entry file:

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

Table 2–4 describes the most important keys in directory entry files.

TABLE 2–4 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 filename extension.
Type	Specifies the type of menu. The value of this key is always Directory.

Editing Menus

You use the following GNOME Desktop components to edit menus:

- Nautilus file manager
- Menus on panels

When you use the file manager to add menus or menu items for all users, you must add the menu or menu item to a URI location. Table 2–1 lists the menus to which you can add items, and the URI locations that correspond to the menus.

When you use panels to customize menus for all users, you use the menu item popup menu. For more information, see *Working With Menus* in the *GNOME 2.2 Desktop on Linux User Guide*.

You can also use menu configuration files and menu data files to customize menus.

Adding Menus

You can add menus for all users in the following ways:

- Use the file manager.
- Modify the menu configuration files and menu data files.

To Add a Menu Using the File Manager

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

1. In a file manager window, access the location where you want to add the menu. For example, to add a menu to the Applications menu, type `applications-all-users:///` in the **Location** field, then press Return.

2. Choose File → New Folder. An untitled folder is added to the view pane. The name of the folder is selected.

3. Type a name for the folder, then press Return. The vfolder information file for the location that you accessed in step 1 is automatically updated with the details of the new menu. The name of the folder is displayed as the name of the menu.

Note – You might need to reload the display of the file manager window before you can type the name for the folder.

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

To Add a Menu Using Menu Files

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. Create the directory entry file in the `/usr/share/gnome/vfolders` directory. For more information on directory entry files, see "Directory Entry Files" on page 35.

2. Locate the vfolder information file for the location where you want to add the menu. For example, to add a menu to the Applications menu, locate the file `/etc/gnome-vfs-2.0/vfolders/applications-all-users.vfolder-info`.

3. In the vfolder information file, add a `<Folder>` element for the new menu. For more information on vfolder information files, see "Vfolders and Menus" on page 32.

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

To Add 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 "Desktop Entry Files" on page 34.

2. Open a file manager window. Choose File → New Window to open a second file manager window.

3. In one window, access the location where you want to add the menu item. For example, to add a menu item to the Preferences menu, type `preferences-all-users:///` in the **Location** field, then press Return.

4. In the other window, select the desktop entry file that you created for the menu item. Drag the desktop entry file to the location where you want to add the menu item.

   Alternatively, you can copy the desktop entry file, then paste the file into the location where you want to add the menu item.

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

To Edit the Properties of a Menu

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

1. From a panel, open the menu that you want to edit. Right-click on any item in the menu.

2. Choose Entire menu → Properties. A **Launcher Properties** dialog is displayed.

3. Modify the properties of the menu in the **Launcher Properties** dialog. For more information on the elements in the **Launcher Properties** dialog, see *Working With Panels* in the *GNOME 2.2 Desktop on Linux User Guide*.

4. Click OK.

To Edit a Menu Item

To edit a menu item, perform the following steps:

1. From a panel, open the menu that contains the item that you want to edit. Right-click on the item that you want to edit.

2. Choose Properties. A **Launcher Properties** dialog is displayed.

3. Modify the properties of the menu item in the **Launcher Properties** dialog. For more information on the elements in the **Launcher Properties** dialog, see *Working With Panels* in the *GNOME 2.2 Desktop on Linux User Guide*.

4. Click OK.

To Delete an Item from a Menu

To delete an item from a menu, from a panel, open the menu that contains the item that you want to delete. Right-click on the item that you want to delete. Choose Remove this item.

The next time that users log in, the menu item is not displayed in the menu.

To Configure Menus That Users Cannot Modify

Users cannot modify a menu if the following conditions are true:

- A vfolder information file that corresponds to the menu is present in the `/etc/gnome-vfs-2.0/vfolders` directory.
- The vfolder information file has the same name as the URI location that corresponds to the menu.
- The user permissions for the vfolder information file are set to read only.

To configure a menu so that users cannot modify the menu, perform the following steps:

1. Create a vfolder information file for the menu that you want to configure in the `/etc/gnome-vfs-2.0/vfolders` directory.

2. Give the vfolder information file the name of the URI location that corresponds to the menu that you want to configure. For example, to configure the Applications menu, create a vfolder information called `applications.vfolder-info` in the `/etc/gnome-vfs-2.0/vfolders` directory.

3. Set the permissions on the vfolder information file to read only.

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" on page 41
- "To Install a New Controls Option" on page 43
- "To Install a New Window Frame Option" on page 43
- "Installing Icons for Themes" on page 44
- "To Create a Custom Controls Option" on page 45

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.
`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.

TABLE 3–1 Keys in Theme Index Files   *(Continued)*

Index File Key	Description
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 × 48 pixel versions of the icon:

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

If possible, create 16 × 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 "To Install a New Controls Option" on page 43 or "To Install a New Window Frame Option" on page 43. 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.

MIME Types

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

- "Introduction to MIME Types" on page 47
- "Detecting the MIME Type for a File" on page 48
- "Registering Applications for MIME Types" on page 52
- "Adding an Application to the GNOME Desktop" on page 54

Introduction to MIME Types

A *Multipurpose Internet Mail Extension* (MIME) type identifies the format of a file. The MIME type enables applications to read the file. Applications such as Internet browsers and email applications use the MIME type to handle files of different types. For example, an email application can use the MIME type to detect what type of file is in a file attached to an email.

The Nautilus file manager uses MIME types to identify the type of a file. The 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.

If you add a new application, you must ensure that other applications can recognize the files associated with the application. You must perform several tasks to enable other applications to detect the MIME type of the application files.

This section describes how applications detect the MIME types of files, and how applications are associated with MIME types. This chapter also describes the procedure that you must follow to add a new application.

Detecting the MIME Type for a File

Applications can detect the MIME type of a file as follows:

1. The application uses *file content sniffers* to search for a particular pattern in the file. A file content sniffer associates a specific pattern in a file with a MIME type. If the application finds a match for the pattern, the MIME type associated with the pattern is the MIME type of the file.

2. If file content sniffers do not identify the MIME type, then the application can check the filename. The application checks the filename against the *MIME type registry*. The MIME type registry associates particular filename extensions and filename patterns, with particular MIME types. If a match for the filename is found, the MIME type associated with the extension or pattern is the MIME type of the file.

The following sections provide further information on file content sniffers and the MIME type registry.

File Content Sniffers

File content sniffers are specified in the file `/etc/gnome-vfs-mime-magic`. The following is an example of a file content sniffer:

0   string   \x89PNG   image/png

The syntax for file content sniffers is as follows:

offset_start[:offset_end] pattern_type pattern [&pattern_mask] type

Table 4–1 describes the fields in a file content sniffer.

TABLE 4–1 Fields in a File Content Sniffer

Field	Description
`offset_start`	Specifies the number of characters to ignore in the file before searching for a text pattern.
`pattern_type`	Specifies the type of pattern to search for. The `string` pattern type is the only pattern type that is supported at the time of publication of this guide.
`pattern`	Specifies the pattern to search for.

TABLE 4–1 Fields in a File Content Sniffer	*(Continued)*
Field	**Description**
`pattern_mask`	Specifies a *pattern mask*, in hexadecimal format. For more information on pattern masks, see the next section.
	This field is optional. This field is not present in the example.
`type`	Specifies the MIME type to associate with files that match this entry.

Pattern Masks

A pattern mask identifies bits in the pattern to ignore when searching for a pattern in a file. The following is an example of a file content sniffer with a pattern mask:

0   string   BMxxxx\000\000 &0xffff00000000ffff   image/bmp

The pattern and mask in the example are as follows:

Pattern	B	M	x	x	x	x	\000	\000
Mask	ff	ff	00	00	00	00	ff	ff

The pattern and mask specify a file with the following characteristics:

1. The file begins with BM.
2. BM is followed by four bytes with any values.
3. The four bytes are followed by \000\000.

The file content sniffer specifies that the MIME type of files that match the pattern and mask is image/bmp.

MIME Type Registry

The MIME type registry is located in `/usr/share/mime-info`. The MIME type registry contains the following files:

File	Filename Extension
MIME information file	`.mime`
MIME keys file	`.keys`

The following sections describe MIME information files and MIME keys files.

MIME Information Files

MIME information files associate MIME types with one or both of the following:

■ Filename extensions
■ Filename patterns

When an application searches for the MIME type of a file, the application checks the filename against the MIME information files. If a match for the filename is found, the MIME type associated with the extension or pattern is the MIME type of the file.

In MIME information files, the filename pattern to search for is written as a regular expression.

The format of MIME type entries in MIME information files is as follows:

MIME-type
      ext[,priority]: *list-of-extensions*
      regex[,priority]: *list-of-regular-expressions*

You can specify a priority value for the filename extension and the regular expression. You can use the priority value to differentiate composite filenames. For example, you can assign a priority of 1 to the .gz extension, and assign a higher priority of 2 to the .tar.gz extension. In this case, the file abc.tar.gz takes the MIME type for .tar.gz.

Note – You must indent the ext field and the regex field with a tab character (\t).

The following MIME type entries are samples from the gnome-vfs.mime MIME information file:

application/x-compressed-tar
      regex,2: tar\.gz$
      ext: tgz
audio/x-real-audio
      ext: rm ra ram
image/jpeg
      ext: jpe jpeg jpg
image/png
      ext: png
text/html
      ext: html htm HTML
text/plain
      ext: asc txt TXT
text/x-readme
      regex: README.*

Note – The file manager reads the MIME information files alphabetically. The alphabetical order determines the order in which MIME types are assigned to filename extensions or regular expressions. For example, if the same file extension is assigned to different MIME types in the files `abc.mime` and `def.mime`, the MIME type in `abc.mime` is used.

MIME Keys Files

MIME keys file provide information about a MIME type that is used in the user interface. For example, the MIME keys file provides a description of a MIME type, and specifies an icon to represent files of that MIME type.

The following is a sample from a MIME keys file:

```
text/html
    description=HTML page
    icon_filename=gnome-text-html
    default_action_type=application
    short_list_application_ids_for_novice_user_level=mozilla,netscape,galeon
    category=Documents/World Wide Web
```

Note – You must indent the keys in a MIME keys file with a tab character (`\t`).

Table 4–2 describes the most important keys in MIME keys files. Typically, the `description` key and the `category` key are localized.

TABLE 4–2 Keys in MIME Keys Files

Key	Description
can_be_executable	Specifies whether files of this MIME type can be executed.
description	Describes the MIME type. This description can be displayed in the file manager and other applications.
icon_filename	Specifies the filename of an icon to represent the MIME type. Does not specify the path to the filename, or the filename extension.
	This icon can be displayed in the file manager and other applications.

TABLE 4–2 Keys in MIME Keys Files   *(Continued)*

Key	Description
`default_action_type`	Specifies the category of action to take when a file of this MIME type is opened by the user. Enter `application` for this MIME type for most applications.
short_list_application_ids _for_novice_user_level	Specifies the application to use when a file of this MIME type is opened by a user. Specify one or more applications, in order of priority. The applications must also be registered in the application registry.
`category`	Specifies a category for the MIME type. The value of this key determines the location of the MIME type in the File Associations preference tool.

Registering Applications for MIME Types

The *application registry* contains text files that register applications. The application registration files contain a series of key-value pairs that specify details for applications. For example, the application registration files contain the following information:

- The command to use to start the application.
- MIME types to associate with the application.

An application registration file can contain one or more application registrations. Application registration files have a `.applications` extension.

The location of the application registry is `/usr/share/application-registry`. This directory contains a default application registration file that is called `gnome-vfs.applications`.

To register an application, add a registration file for the application to the application registry.

The following is an example of an application registration:

eog
    command=eog
    name=Eye of Gnome
    can_open_multiple_files=true
    expects_uris=false

```
requires_terminal=false
mime_types=image/bmp,image/gif,image/jpeg,image/png,image/tiff,
```
image/x-xpixmap,image/x-bmp,image/x-png,image/x-portable-anymap,
image/x-portable-bitmap,image/x-portable-graymap,
image/x-portable-pixmap

Table 4–3 describes the keys in application registration files.

TABLE 4–3 Keys for an Application Registration

Key	Description
Application identifier	Specifies a unique identifier for the application. This identifier must be the same as the identifier in the short_list_application_ids_for_novice_user_level key in the MIME keys file for the application.
command	Specifies the command to use to start the application, and any options to use with the command.
name	Specifies a name for the application. The name is used in the user interface. For example, the name is used in the Open With submenu in the file manager.
can_open_multiple_files	Specifies whether the application can open several files at the same time.
expects_uris	Specifies whether the application can process URIs. If the value of this key is true, the application registration entry must also contain a supported_uri_schemes key.
supported_uri_schemes	Specifies the URI schemes that the application can process.
requires_terminal	Specifies whether to run the application in a terminal window. Enter true for this field for an application that does not create a window in which to run.
mime_types	Specifies the MIME types that the application can use.

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 Chapter 2.

2. Add an icon for the application to `/usr/share/icons/`*theme-name*`/icon-size`/`apps`. For more information on icons and themes, see Chapter 3.

3. If the application uses a new MIME type, add a file content sniffer for the new MIME type. For more information on file content sniffers, see "File Content Sniffers" on page 48.

4. If the application uses a new MIME type, add a MIME information file for the application to the MIME type registry. For more information on MIME information files, see "MIME Information Files" on page 50.

5. Add a MIME keys file for the application to the MIME type registry. For more information on MIME keys files, see "MIME Keys Files" on page 51.

6. 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.

7. To associate the application with a MIME type, add an application registration file to the application registry. For more information on the application registry, see "Registering Applications for MIME Types" on page 52.

Configuring GDM

This chapter describes how to configure GDM in the GNOME Desktop, and how to log in to a new session.

Introduction to GDM

To start a GNOME Desktop session, users must log in to authenticate their identity. The login screen provides a gateway to the GNOME Desktop for the user. The GNOME Display Manager (GDM) application controls the login screen. You can configure GDM to customize how users log in to the system.

GDM also enables you to log in to a new session. You do not need to log out from the current session before you log in to the new session.

Configuring GDM

To configure the GDM login screen, use the Login Screen Setup application. To open Login Screen Setup, choose Applications → System Tools → Login Screen Setup.

Note – You must have system administrator or `root` access privileges to configure the GDM login screen.

You can configure GDM in the following functional areas:

- **General**
- **Standard greeter**
- **Graphical greeter**
- **Security**
- **XDMCP**

General Settings

Use the **General** tabbed section to set general preferences for the GDM login screen.

Table 5–1 lists the general GDM settings that you can configure.

TABLE 5–1 General GDM Settings

Element	Description
Local	Select the type of interface to use for the login screen when users log in from a local system. Select one of the following options from the drop-down list: **Graphical greeeter**: Select this option to use a graphical login screen when users log in from a local system. **Standard greeeter**: Select this option to use the standard GNOME login screen when users log in from a local system.
Remote	Select the type of interface to use for the login screen when users log in from a remote system. Select one of the following options from the drop-down list: **Graphical greeeter**: Select this option to use a graphical login screen when users log in from a remote system. **Standard greeeter**: Select this option to use the standard GNOME login screen when users log in from a remote system.
Always use 24 hour clock format	Select this option to display the time on the login screen in the 00.00 to 24.00 format.

TABLE 5–1 General GDM Settings *(Continued)*

Element	Description
Welcome string	Type a welcome message to display on the standard GNOME login screen when users log in from a local system.
	Note – If you want to display the system name in the welcome message, type %n in this text box.
Remote welcome string	Type a welcome message to display on the standard GNOME login screen when users log in from a remote system.
Login a user automatically on first bootup	Select this option to log in a user automatically when the user boots up the system for the first time.
Automatic login username	Use the drop-down combination box to enter a username to use when the system logs in a user automatically.
Login a user automatically after a specified number of seconds	Select this option to log in a user automatically after an interval that you specify.
Timed login username	Use the drop-down combination box to enter a username to use when the system logs in a user automatically after an interval.
Seconds before login	Use the spin box to specify how long to wait to log in the user automatically.

Standard GNOME Login Screen Settings

Use the **Standard greeter** tabbed section to set preferences for the standard GNOME login screen.

Table 5–2 lists the standard GNOME login screen settings that you can configure.

TABLE 5–2 Standard GNOME Login Screen Settings

Element	Description
Logo	Choose an image to display as a logo on the standard GNOME login screen. To choose an image, enter the filename of the image in the drop-down combination box. Alternatively, click Browse to display a dialog from which you can choose an image.

TABLE 5–2 Standard GNOME Login Screen Settings   *(Continued)*

Element	Description
Show choosable user images (face browser)	Select this option to display images of users on the standard GNOME login screen. If this option is selected, users can select an image instead of type a username.
No background	Select this option if you do not want to display an image or color in the background of the standard GNOME login screen.
Image	Select this option to display an image in the background of the standard GNOME login screen. To choose an image, enter the filename of the image in the drop-down combination box at the right side of the dialog. Alternatively, click on the Browse button at the right side of the dialog to display a dialog from which you can choose an image.
Color	Select this option to display a color in the background of the standard GNOME login screen. Use the **Background color** button to specify the color.
Scale background image to fit	If you select the **Image** option, select this option to scale the background image to fit the background of the standard GNOME login screen. The width-to-height ratio of the image is retained.
Only color on remote displays	Select this option to display a color in the background of the standard GNOME login screen when users log in from a remote system.
Background color	If you select the **Color** option or the **Only color on remote displays** option for the background, use this button to specify the color. Click on the color selector button to display the color selector dialog. Choose the color that you require from the color selector dialog.

Graphical Login Screen Settings

Use the **Graphical greeter** tabbed section to set preferences for the graphical login screen.

Table 5–3 lists the graphical login screen settings that you can configure.

TABLE 5–3 Graphical Login Screen Settings

Element	Description
Theme list	Select a theme to use for the graphical login screen. You can preview the theme at the right side of the dialog.
Install new theme	You can add a theme to the list of available themes. The new theme must be an archive file that is tarred and zipped. That is, the new theme must be a `.tar.gz` file. To install a new theme, perform the following steps: 1. Click on the Install new theme button. 2. Use the dialog to select the archive file. When you have selected the file, click OK.
Delete theme	To delete a theme, select the theme, the click on the Delete theme button.

Security Settings

Use the **Security** tabbed section to set security preferences for GDM.

Table 5–4 lists the security settings that you can configure.

TABLE 5–4 Security Settings

Element	Description
Allow root to login with GDM	Select this option to enable users with system administrator or `root` access privileges to use GDM to log in from a local system. **Note –** Systems that support *Pluggable Authentication Modules (PAM)* ignore this option. The PAM libraries determine whether the user is on the local system.
Allow root to login remotely with GDM	Select this option to enable users with system administrator or `root` access privileges to use GDM to log in from a remote system.
Allow remote timed logins	Select this option to enable GDM to log in a user automatically after an interval, from a remote system.
Show actions menu	Select this option to enable users to use the Actions menu on the login screen.

TABLE 5–4 Security Settings　　*(Continued)*

Element	Description
Allow configuration from the login screen	Select this option to enable users to use the Configure the login manager item from the Actions menu on the login screen.
Allow running XDMCP chooser from the login screen	Select this option to enable users to use the Run XDMCP chooser item from the Actions menu on the login screen. The Run XDMCP chooser item displays a list of hosts that can offer display management services. Users can choose a host to manage a session from this list.
Always disallow TCP connections to X server (disables all remote connections)	Select this option if you do not want users to be able to connect to the X Window System server from remote systems.
Retry delay (seconds)	Use the spin box to specify how long to wait after a failed login attempt, to reactivate the **Username** field on the login screen. **Note –** Systems that support *Pluggable Authentication Modules (PAM)* ignore this option.

XDMCP Settings

Use the **XDMCP** tabbed section to set preferences for X Display Manager Control Protocol (XDMCP).

Table 5–5 lists the XDMCP settings that you can configure.

TABLE 5–5 XDMCP Settings

Element	Description
Enable XDMCP	Select this option to enable a remote X Windows System display to request an X Windows System session from the system.
Honour indirect requests	Select this option to enable remote X Windows System displays that do not have a display manager to request XDMCP display management services from this system.
Listen on UDP port	Use the spin box to specify the port number on which to listen for *User Datagram Protocol (UDP)* requests.

TABLE 5–5 XDMCP Settings   *(Continued)*

Element	Description
Maximum pending requests	Use the spin box to specify the maximum number of queued requests for sessions from the system.
	Note – Use this option to help avoid denial of service attacks. This option specifies the number of displays that can *request* a session at one time. This option does not specify the total number of remote sessions which GDM allows.
Max pending indirect requests	GDM creates a queue of requests for sessions from the system. Use the spin box to specify the maximum number of queued requests for sessions from displays that do not have a display manager.
Maximum remote sessions	Use the spin box to specify the total number of remote sessions which GDM allows.
Maximum wait time	Use the spin box to specify how long to wait before GDM removes a request from the queued requests.
Maximum indirect wait time	Use the spin box to specify how long to wait before GDM removes displays that do not have a display manager, from the queue of displays that have requested sessions.
Displays per host	Use the spin box to specify the total number of sessions which GDM allows from a host.
Ping interval (minutes)	GDM pings sessions to check that the session is still active. Use the spin box to specify the interval between pings from GDM.

Logging in to a New Session

To log in to a new session, choose Applications → System Tools → New login. A login screen is displayed. Enter your username and password to log in.

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" on page 63
- "Setting Screensaver Preferences" on page 63
- "Modifying Screensaver Displays" on page 64

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 2.2 Desktop on Linux 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.

Note – 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
```

Note – 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.

In Java Desktop System, the displays are located in the `/usr/lib/xscreensaver` directory.

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

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" on page 67
- "Setting Session Defaults" on page 68

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:

File	Description
/usr/share/gnome/default.session	Default session file. Default session details are stored in this file.
$HOME/.gnome2/session	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.

Help System

This chapter describes the Help system in the GNOME Desktop.

- "Introduction" on page 69
- "OMF Files" on page 69
- "ScrollKeeper Cataloging System" on page 70

Introduction

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

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
```

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

Improving Performance

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

- "Introduction to Improving Performance" on page 71
- "Reducing CPU Usage" on page 71
- "Reducing X Window System Network Traffic" on page 76
- "Reducing Color Usage and Improving Display Quality" on page 77
- "Improving Performance of Help" on page 79

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.

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

Note – 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.

Tip – 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 menus** 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 local_only, the CPU usage is reduced.
never	Never performs the action. When you set a performance preference to never, the CPU usage and the network traffic are reduced.

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
show_icon_text	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: `# gconftool-2 --type string --set /apps/nautilus/preferences/ show_icon_text never` 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 **Show text in icons** preference.

Preference	Description
show_directory_item_counts	Specifies when to show the number of items in folders. To never show the number of items in folders, run the following command: `# gconftool-2 --type string --set /apps/nautilus/preferences/ show_directory_item_counts never` 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 **Count number of items** preference.
show_image_thumbnails	Specifies when to show thumbnails of image files. To never show thumbnails, run the following command: `# gconftool-2 --type string --set /apps/nautilus/preferences/ show_image_thumbnails never` 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 **Show thumbnails** preference.
preview_sound	Specifies when to preview the content of sound files. To never preview the content of sound files, run the following command: `# gconftool-2 --type string --set /apps/nautilus/preferences/preview_sound never` 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 "To Use Theme Options Which Require Less CPU Resources" on page 72.

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 "To Reduce CPU Usage by Turning Off Display of Icons in Menus" on page 73.

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 "To Use Theme Options Which Require Less CPU Resources" on page 72.

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 "To Reduce CPU Usage by Turning Off Display of Icons in Menus" on page 73.

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 "To Reduce CPU Usage by Turning Off the Splash Screen" on page 73.

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.

Improving Performance of Help

The source documents for the Help are XML files. The XML files are converted to HTML, and the Yelp Help browser displays the HTML. The HTML can be *pregenerated*, that is, the XML can be converted to HTML before the Help is used.

Alternatively, the conversion can be performed automatically when a user calls a Help file. If the conversion is performed when the user calls the Help file, the conversion has a significant impact on the speed with which Help documents are displayed.

To avoid performance problems with the Help, pregenerate the HTML so that the Help browser does not need to do the conversion. Use the `yelp-pregenerate` command to pregenerate a Help document. The syntax of the `yelp-pregenerate` command is as follows:

yelp-pregenerate [-a | [-f *filename*]] [*locale*]

The following table describes the options and arguments for the `yelp-pregenerate` command:

Option or Argument	Description
-a	Pregenerates all XML files that are registered in the ScrollKeeper database. This option is the default option.
-f	Pregenerates only the XML files that you specify in the command.
filename	The name of the XML file or XML files that you want to pregenerate.
locale	The locale for which you want to pregenerate the XML files. The default locale is C, or English.

For example, to pregenerate the English version of all XML files in the file system, run the following command:

yelp-pregenerate -a

You do not need to specify the locale in this example, as the default locale is C. To pregenerate the Spanish version of the gedit manual, run the following command:

yelp-pregenerate -f gedit.xml es

When a user opens a Help file, the Help browser does the following:

1. The Help browser searches for the appropriate HTML file. If the Help browser finds the HTML file, the Help browser compares the modification dates of the HTML file and the corresponding XML file. If the HTML file is more recent than the XML file, the HTML file is displayed.
2. If the HTML file is older than the XML file, the XML file is converted to HTML.
3. If the Help browser does not find a HTML file for a Help document, the XML file is converted to HTML.

In other words, the Help browser displays pregenerated HTML if the pregenerated HTML file is present. If the pregenerated HTML file is not present, the Help browser converts the XML file to HTML.

Note – You might want to pregenerate the Help for any new applications that you want to add to the GNOME Desktop. If there are severe space restrictions in your environment, you might choose not to pregenerate the Help.

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 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
.esd_auth	Contains the authentication cookie for the GNOME sound daemon. The GNOME sound daemon is the Enlightened Sound Daemon (ESD).
.gconf	Contains the GConf configuration source for the user. When the user sets a preference, the new preference information is added to this location.
.gconfd	Contains the following GConf daemon details: ■ Configuration information. ■ Lock information for objects that are referenced by an *Interoperable Object Reference (IOR).* ■ State information for objects that are referenced by an IOR.
.gnome	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.

TABLE A–1 Hidden Directories in User Home Directories   *(Continued)*

Directory	Description
`.gnome-desktop`	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: ■ 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 `starthere.desktop` file contains a link to the **Start Here** location. ■ Removable media volumes that are mounted. The file manager also contains a preference that enables users to use the home directory as the desktop directory, instead of `.gnome-desktop`. If a user selects this option, the contents of the home directory are displayed as desktop objects.
`.gnome2`	Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains the following: ■ Keyboard shortcut information. ■ Window location information. ■ Desktop entry files for panel launchers. This directory also contains user-specific menu data. If a user modifies menus, the details are stored here.
`.gnome2-private`	Ignore this directory. This directory has no function at the time of publication of this guide.
`.metacity`	Contains session data for the Metacity window manager.
`.nautilus`	Contains file manager data that is specific to the user. For example, this directory can contain the following: ■ Metadata for the directories with which the user works. ■ Nautilus emblems that the user adds. ■ Nautilus desktop images.

TABLE A–1 Hidden Directories in User Home Directories   *(Continued)*

Directory	Description
.themes	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.
.thumbnails	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.
.xscreensaver	Contains screensaver configuration data and screensaver preference data.

Glossary

`.desktop` file	See desktop entry file.
`.directory` file	See directory entry file.
`.omf` 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 `/usr/share/gnome/application-registry`.
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 `.desktop` 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 `.directory` 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, `xml:readonly:/etc/gconf/gconf.xml.defaults`.
GConf preference key	An element in the GConf repository that corresponds to an application preference. For example, the `/apps/gnome-session/options/show_splash_screen` 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 `.schemas` extension.
GConf schema key	A key that stores a schema object for a preference key. For example, `/schemas/desktop/gnome/interface/font_name` is a schema key for the `/desktop/gnome/interface/font_name` 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 filename extensions and filename patterns. MIME information files have a `.mime` 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 `.keys` 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 `image/png` 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 `/usr/share/mime-info`.
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 `.omf` 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.

pregenerate	To convert source XML files for online Help to HTML before the online Help is used. You can pregenerate Help to improve the performance of the Help browser. Use the `yelp-pregenerate` command to pregenerate Help.
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 XScreenSaver.
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 *Netscape color palette* and the *Netscape color cube*.

Index

V
vfolder information files, 32
vfolders, 32

W
workspaces, setting number with GConf, 20

X
X Window System network traffic, reducing, 76
XDMCP, configuring for GDM, 60
XScreenSaver, *See* screensavers