GNOME 2.0 Desktop System Administration Guide

December 17, 2002

Revision History

Revision Number: GNOME 2.0 Desktop System Administration Guide V2.0

Date: December 2002

Author: Sun Microsystems

Email: gdocteam@sun.com

Publisher: GNOME Documentation Project

Contents

1 Using GConf 5

Introducing GConf 5

GConf Repository 6

GConf Configuration Sources 7

GConf Schemas 8

GConf Schema Definition Files 9

GConf Daemon 9

GConf Command Line Tool 10

To Set Preference Values 13

Setting General Preferences 13

To Set HTTP Proxy Preferences 14

To Set Print Manager Preferences 14

To Set the Number of Workspaces 14

To Set Keyboard Accessibility Preferences 15

To Set Keyboard Shortcut Preferences 15

To Set Global Panel Preferences 16

To Set Preferences for Individual Panels and Panel Objects 16

Schema Definition File for Individual Panels and Panel Objects 17

Sample Script for Setting Individual Panel Preferences 19

Setting Look-and-Feel Preferences 21

To Set Font Preferences 21

To Set Desktop Background Preferences 21

To Set Splash Image Preferences 22

To Restore Default Preference Values 22

2 Customizing Menus 23

Introduction to Menus 23

File Abstraction Layer 24

Vfolders 24

Desktop Entry Files 26

Directory Entry Files 27

Editing Menus 28

Adding Menus 28

To Add a Menu Using the File Manager 28

To Add a Menu Using Menu Files 29

To Add an Item to a Menu 29

To Edit the Properties of a Menu 30

To Edit a Menu Item 30

To Delete an Item from a Menu 30

To Configure Menus That Users Cannot Modify 31

Glossary 33

Index 35

CHAPTER **1**

Using GConf

The information in this chapter describes how to use GConf to manage user preferences.

- "Introducing GConf" on page 5
- "GConf Repository" on page 6
- "GConf Daemon" on page 9
- "GConf Command Line Tool" on page 10
- "To Set Preference Values" on page 13
- "Setting General Preferences" on page 13
- "Setting Look-and-Feel Preferences" on page 21
- "To Restore Default Preference Values" on page 22

Introducing GConf

GConf simplifies the administration of preferences for users in the GNOME desktop environment. 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 user interface does not contain all of the preference keys in the GConf repository. For example, the Sessions preference tool does not contain an option that corresponds to the `/apps/gnome-session/options/splash_screen_text` 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>avantgarde 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/`*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.
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 user interface. For example, `system_http_proxy.schemas` corresponds to the Network Proxy preference tool. Other schema definition files contain preference keys that are not present in the GNOME user interface. For example, the `/apps/gnome-session/options/splash_screen_text` key is not present in the user interface.

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

GConf Daemon

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

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

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

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

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

To terminate the GConf daemon, run the following command:

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

GConf Command Line Tool

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

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

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

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

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.
`--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.

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

Option	Function
`--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.
`--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.
`--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 value of all preference keys in all subdirectories in a directory that you specify.
`--recursive-unset`	Resets the values of all preference keys, in all subdirectories in a directory, from the user setting to the setting in the default configuration source.
`--set`	Sets the value of a preference key, and writes the value to the user configuration source. Use the `--type` option with the `--set` option to specify the data type of the value that you want to set. For example, the following command sets the value of the `/apps/gnome-terminal/profiles/Default/background_color` key in the user configuration source: `# ` **`gconftool-2 --set "/apps/gnome-terminal/profiles/Default/background_color" --type string "#000000"`** You can also use the `--direct` option and the `--config-source` option with the `--set` option to write a value to another configuration source.

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

Option	Function
`--set-schema`	Sets the value of an attribute in a schema key, and writes the value to the default configuration source. Use the following options with the `--set-schema` option to specify the attribute that you want to update: ■ `--type` ■ `--short-desc` ■ `--long-desc` ■ `--owner` For example, the following command sets the short description in the schema key for the `/apps/gnome-terminal/profiles/Default/background_color` key: `#` **`gconftool-2 --set-schema "/schemas/apps/gnome-terminal/profiles/Default/background_color" --short-desc "Default background color of terminal"`**
`--short-desc=`*description*	Use this option with the `--set-schema` option to specify a short description for a schema key.
`--shutdown`	Terminates the GConf daemon.
`--type=`*data_type*	Use this option to specify the data type when you set a value of a preference key. You can also use this option when you set the value of an attribute in a schema key. The following is a list of valid data types: ■ `bool` ■ `float` ■ `int` ■ `list` ■ `pair` ■ `string`
`--unset`	Resets the value of a preference key from the user setting to the setting in the default configuration source.
`--usage`	Displays a brief help message about the `gconftool-2` command, and the options that you can use with the `gconftool-2` command.

To Set Preference Values

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

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

```
# gconftool-2 --direct --config-source configuration_source --type
data_type --set preference_key value
```

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

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

The user cannot override this preference value.

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

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

The user can override this preference value.

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 Print Manager Preferences

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

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

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

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

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

To Set the Number of Workspaces

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

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

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

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

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

To Set Keyboard Accessibility Preferences

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

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

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

```
# gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.defaults --type bool --set
desktop/gnome/accessibility/keyboard/enable 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 two locations. The following table shows the locations to modify, and the part of the user interface to which the locations correspond:

GConf Location	User Interface Component
`/apps/panel/global`	Keyboard Shortcuts preference tool, **Desktop** section
`/apps/metacity/general`	Keyboard Shortcuts preference tool, **Window Management** section

For example, you might want users to use only the Alt + F3 keyboard shortcut to open the **Run Program** 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/panel/global/run_key '<Alt>F3'
```

You might want users to use Alt + F11 to maximize windows, by default. To set this default value, run the following command:

```
# gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.defaults --type string --set
/apps/metacity/window_keybindings/maximize '<Alt>F11'
```

You can also set other keyboard shortcut preferences. For information on the other keyboard shortcut preferences, see the `panel-global-config.schemas` and `metacity.schemas` schema definition files.

To Set Global Panel Preferences

To set global preferences for panels, you modify the values of the preference keys in the `/apps/panel/global` location. For example, if you do not want users to ever use panel animation, set a mandatory value as follows:

```
# gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type bool --set
/apps/panel/global/enable_animations 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/panel/global/enable_animations false
```

You can also set other global panel preferences. For information on the other global panel preferences, see the `panel-global-config.schemas` schema definition file.

To Set Preferences for Individual Panels and Panel Objects

The `panel-per-panel-config.schemas` schema definition file specifies the following details of the panels on the 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-per-panel-config.schemas` schema definition file. For more information on the `panel-per-panel-config.schemas` schema definition 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 many preferences is to use a script. For a sample script that sets preferences for a panel and objects on the panel, see "Sample Script for Setting Individual Panel Preferences" on page 19.

Schema Definition File for Individual Panels and Panel Objects

The schema definition file `panel-per-panel-config.schemas` contains sections that specify panels and panel contents for small, medium, and large screens. The section for small screens and the section for large screens are not functional at the time of publication of this guide. To learn about preference keys for panels or panel objects, read the section for medium screens only.

The medium section of the file `panel-per-panel-config.schemas` is structured as follows:

1. Preference keys that specify the general structure of panels, applets, and other panel objects on the desktop. The following keys specify the number of panels, applets, and other panel objects that appear on the desktop:

   - `/apps/panel/default_profiles/medium/general/panel_id_list`
   - `/apps/panel/default_profiles/medium/general/applet_id_list`
   - `/apps/panel/default_profiles/medium/general/object_id_list`

   The keys also assign identifiers to each panel, applet, and panel object. For example, the following excerpt from `panel-per-panel-config.schemas` specifies that two panels appear on the desktop:

   ```
   <key>/schemas/apps/panel/default_profiles/medium/
   general/panel_id_list</key>
   .
   .
   .
   <default>[00000001,00000002]</default>
   ```

   In `panel-per-panel-config.schemas`, the identifier 00000001 identifies the bottom edge panel, and the identifier 00000002 identifies the Menu Panel.

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

   /apps/panel/default_profiles/medium/panels/*panel_number*/*panel_property_key*

For example, the key `/apps/panel/default_profiles/medium/panels/00000001/screen_edge` specifies the edge of the display on which the bottom edge panel resides.

The keys that specify the properties of the bottom edge panel are listed first. The keys that specify the properties of the Menu Panel are listed at the end of the medium section in `panel-per-panel-config.schemas`.

3. Preference keys that list the panel objects, the panel object properties, and the panels in which the objects reside. For example, the following excerpt from `panel-per-panel-config.schemas` specifies object 00000001:

   ```
   <key>/schemas/apps/panel/default_profiles/medium/
   objects/00000001/object_type</key>
   .
   .
   .
   <default>launcher-object</default>
   .
   .
   .
   <key>/schemas/apps/panel/default_profiles/medium/
   objects/00000001/launcher_location</key>
   .
   .
   .
   <default>applications:///nautilus.desktop</default>
   .
   .
   .
   <key>/schemas/apps/panel/default_profiles/medium/
   objects/00000001/panel_id</key>
   .
   .
   .
   <default>00000002</default>
   ```

   The object is a launcher that starts Nautilus file manager. The launcher resides in the Menu Panel.

4. Preference keys that list the applets, the applet preferences, and the panels in which the applets reside. For example, the following excerpt from `panel-per-panel-config.schemas` specifies applet 00000001:

   ```
   <key>/schemas/apps/panel/default_profiles/medium/
   applets/00000001/object_type</key>
   .
   .
   .
   <default>bonobo-applet</default>
   ```

```
.
.
.
<key>/schemas/apps/panel/default_profiles/medium/
applets/00000001/panel_id</key>
.
.
.
<default>00000001</default>
.
.
.
<key>/schemas/apps/panel/default_profiles/medium/
applets/00000001/bonobo_iid</key>
.
.
.
<default>OAFIID:GNOME_TasklistApplet</default>
```

The applet is the Window List applet. The applet resides in the bottom edge panel.

Sample Script for Setting Individual Panel Preferences

This section contains a sample script that sets the preferences for a panel that has the following major characteristics:

- The panel is an edge panel, and resides at the bottom edge of the screen.
- The panel contains a file manager launcher and a Terminal launcher.
- The panel contains a Window List applet.

The sample script also sets other minor preferences. All of the preferences have default values, not mandatory values.

The sample script follows:

```
# Usage is: set_key <type> <key> <value> <extra options if any>

function set_key()
{
  TYPE="$1"
  shift
  KEY="$1"
  shift
  VALUE="$1"
  shift
  OTHER_ARGS="$*"
  gconftool-2 --direct --config-source xml:readwrite:/etc/gconf/gconf.xml.defaults
--owner=panel $OTHER_ARGS --type "$TYPE" --set "$KEY" "$VALUE"
}

# Define the number of panels, the number of launchers, and the number of applets
set_key list /apps/panel/default_profiles/medium/general
/panel_id_list [00000001] --list-type string
set_key list /apps/panel/default_profiles/medium/general
/object_id_list [00000001,00000002] --list-type string
set_key list /apps/panel/default_profiles/medium/general
/applet_id_list [00000001] --list-type string

# Define the characteristics of the panel
set_key string /apps/panel/default_profiles/medium/panels
/00000001/panel_type edge-panel
set_key string /apps/panel/default_profiles/medium/panels
/00000001/panel_size panel-size-x-small
set_key string /apps/panel/default_profiles/medium/panels
/00000001/screen_edge panel-edge-bottom
set_key bool /apps/panel/default_profiles/medium/panels
/00000001/hide_buttons_enabled false
set_key string /apps/panel/default_profiles/medium/panels
/00000001/panel_background_type no-background

# Define the first launcher
set_key string /apps/panel/default_profiles/medium/objects
/00000001/object_type launcher-object
set_key string /apps/panel/default_profiles/medium/objects
/00000001/launcher_location ’applications:///nautilus.desktop’
set_key string /apps/panel/default_profiles/medium/objects
/00000001/panel_id 00000001
set_key int /apps/panel/default_profiles/medium/objects
/00000001/position 20

# Define the second launcher
set_key string /apps/panel/default_profiles/medium/objects
/00000002/object_type launcher-object
set_key string /apps/panel/default_profiles/medium/objects
/00000002/launcher_location ’applications:///System/gnome-terminal.desktop’
set_key string /apps/panel/default_profiles/medium/objects
/00000002/panel_id 00000001
set_key int /apps/panel/default_profiles/medium/objects
/00000002/position 25

# Define the applet
set_key string /apps/panel/default_profiles/medium/applets
/00000001/object_type bonobo-applet
set_key string /apps/panel/default_profiles/medium/applets
/00000001/bonobo_iid OAFIID:GNOME_TasklistApplet
```

set_key string /apps/panel/default_profiles/medium/applets
/00000001/panel_id 00000001
set_key int /apps/panel/default_profiles/medium/applets
/00000001/position 25

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** section
/apps/nautilus/preferences/desktop_font	Font preference tool, **Desktop font** section

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 background 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 Desktop Background Preferences

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

1. Log in as the user whose default preference values you want to restore.
2. Run the following command:

   ```
   # gconftool-2 --recursive-unset
   ```

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

CHAPTER 2

Customizing Menus

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

- "Introduction to Menus" on page 23
- "File Abstraction Layer" on page 24
- "Vfolders" on page 24
- "Desktop Entry Files" on page 26
- "Directory Entry Files" on page 27
- "Editing Menus" on page 28
- "To Configure Menus That Users Cannot Modify" on page 31

Introduction to Menus

The way in which GNOME implements menus in the desktop 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:///
Desktop Preferences menu for all users	preferences-all-users:///

Vfolders

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>`	The name of the menu.
`<Desktop>`	The name of the directory entry file that specifies the name, comment, and icon for the menu.
`<Query>`	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 simple calculations
Exec=gnome-calculator
Icon=gnome-calc3.png
Terminal=false
Type=Application
X-GNOME-DocPath=gnome-calculator/gnome-calculator.xml
Categories=GNOME;Application;Utility;
```

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

TABLE 2–3 Desktop Entry Keys

Desktop Entry Key	Description
`Encoding`	Enter the encoding of the desktop entry file.
`Name`	Enter the name of the item. This name is displayed on the item in the menu.
`Comment`	Enter a short description of the item. The comment is displayed as a tooltip when you point to the item in the menu.
`Exec`	Enter a command to execute when you choose the item from the menu.
`Icon`	Enter a path to an icon that represents the item.
`Terminal`	Enter `true` if you want the command in the `Exec` key to run in a terminal window. Enter `true` in this key if the command does not create a window in which to run.

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

Desktop Entry Key	Description
`Type`	Enter the type of item. Enter 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.
`X-GNOME-DocPath`	Enter the help file to display when you choose Help on *application_name* from the menu item popup menu.
`Categories`	Enter the keywords that describe the item. Separate the keywords 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 that you enter to menus.

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 background objects also use desktop entry files. The desktop entry files for launchers and desktop background 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`	Enter the name of the menu. This name is displayed on the menu.
`Comment`	Enter a short description of the menu. The comment is displayed as a tooltip when you point to the menu.
`Icon`	Enter a path to an icon that represents the menu.
`Type`	The type of menu. The value of this key is always `Directory`.

Editing Menus

You use the following 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 2.0 Desktop 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/gnome/share/gnome/vfolders` directory. For more information on directory entry files, see "Directory Entry Files" on page 27.
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 `/usr/gnome/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" on page 24.

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 26.
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 2.0 Desktop 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 2.0 Desktop 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/vfolder` 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/vfolder` 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/vfolder` directory.
3. Set the permissions on the vfolder information file to read only.

Glossary

`.desktop` file See desktop entry file.

`.directory` file See directory entry file.

desktop entry file A data file that provides information about an item in a menu. The desktop entry file specifies details such as a name for the item, the command for the item to run, an icon to represent the item, 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.

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.

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.

Index

D

desktop entry files
- *See* menus, desktop entry files

.desktop files
- *See* menus, desktop entry files

directory entry files
- *See* menus, directory entry files

.directory files
- *See* menus, directory entry files

F

file abstraction layer, and menus, 24

G

GConf
- command line tool, 10
- configuration sources, 7
- daemon, 9
- introduction, 5
- repository, 6
- restoring default preference values, 22
- schema definition files
  - introduction, 9
  - panel and panel object, 17
- schema keys, 8
- schema objects, 8
- schemas, 8
- setting desktop background, 21
- setting fonts, 21

GConf (Continued)
- setting global panel preferences, 16
- setting HTTP proxy, 14
- setting individual panel preferences, 16
- setting keyboard accessibility, 15
- setting keyboard shortcuts, 15
- setting number of workspaces, 14
- setting panel object preferences, 16
- setting preference values, 13
- setting print manager, 14
- setting splash image, 22

gconfd
- *See* GConf daemon

M

menus
- adding items to, 29
- adding using file manager, 28
- adding using menu files, 29
- configuring menus that users cannot modify, 31
- deleting menu items, 30
- desktop entry files, 26
- directory entry files, 27
- editing, 28
- editing menu items, 30
- editing properties of, 30
- file abstraction layer, 24
- introduction, 23
- vfolder information files, 24
- vfolders, 24

V

vfolders
    See menus, vfolders