



GNOME 2.0 Desktop for the Solaris Operating Environment System Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 806-6878-13
July 2003

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



030826@6671



Contents

Preface	7
1 Using GConf	11
Introducing GConf	11
GConf Repository	12
GConf Configuration Sources	13
GConf Schemas	14
GConf Schema Definition Files	15
GConf Daemon	15
GConf Command Line Tool	16
To Set Preference Values	19
Setting General Preferences	19
To Set HTTP Proxy Preferences	20
To Set Print Manager Preferences	20
To Set the Number of Workspaces	20
To Set Keyboard Accessibility Preferences	21
To Set Keyboard Shortcut Preferences	21
To Set Global Panel Preferences	22
To Set Preferences for Individual Panels and Panel Objects	22
Setting Look-and-Feel Preferences	27
To Set Font Preferences	27
To Set Desktop Background Preferences	27
To Set Splash Image Preferences	28
To Restore Default Preference Values	28

2	Customizing Menus	29
	Introduction to Menus	29
	File Abstraction Layer	30
	Vfolders and Menus	30
	Desktop Entry Files	32
	Directory Entry Files	33
	Editing Menus	34
	Adding Menus	34
	To Add an Item to a Menu	35
	To Edit the Properties of a Menu	36
	To Edit a Menu Item	36
	To Delete an Item from a Menu	36
	To Configure Menus That Users Cannot Modify	37
3	Installing Themes	39
	Introduction to Themes	39
	To Install a Desktop Theme	40
	To Install a Window Frame Theme	40
	Installing Icons for Themes	41
	To Create a Custom Desktop Theme	42
4	MIME Types	43
	Introduction to MIME Types	43
	Detecting the MIME Type for a File	44
	File Content Sniffers	44
	Pattern Masks	45
	MIME Type Registry	45
	Registering Applications for MIME Types	48
	Adding an Application to the Desktop Environment	50
5	Setting Screensavers	51
	Introduction to Screensavers	51
	Setting Screensaver Preferences	52
	Modifying Screensaver Displays	53

6	Session Management	55
	Introduction to Sessions	55
	Setting Session Defaults	56
	Login Scripts	56
7	Improving Performance	57
	Introduction to Improving Performance	57
	Reducing CPU Usage	57
	To Use Themes Which Require Less CPU Resources	58
	To Reduce CPU Usage by Turning Off Display of Icons in Menus	59
	To Reduce CPU Usage by Turning On Wireframe Mode	59
	To Reduce CPU Usage by Turning Off the Splash Screen	59
	To Reduce CPU Usage by Turning Off Panel Animation	60
	Improving File Manager Performance	60
	To Use the mediaLib Libraries	63
	Reducing X Window System Network Traffic	63
	To Use Themes Which Create Less Network Traffic	63
	To Reduce Network Traffic by Turning Off Display of Icons in Menus	64
	To Reduce Network Traffic by Turning On Wireframe Mode	64
	Reducing Color Usage and Improving Display Quality	64
	To Use Themes That Use the Websafe Color Palette	65
	To Reduce Color Usage by Turning Off Display of Icons in Menus	65
	To Reduce Color Usage by Turning Off the Splash Screen	65
	To Reduce Color Usage by Using a Solid Color for the Desktop Background	66
A	Hidden Directories	67
	Glossary	71
	Index	75

Preface

GNOME 2.0 Desktop for the Solaris Operating Environment System Administration Guide provides information on how to administer a system running the GNOME 2.0 Desktop on the Solaris™ 8 operating environment and the Solaris 9 operating environment.

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 2.0 Desktop. For information on how to use the GNOME 2.0 Desktop, see *GNOME 2.0 Desktop for the Solaris Operating Environment 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 set preferences for the screensaver. This chapter also provides information on how to modify the displays that are available for the screensaver.
- Chapter 6 introduces session management, and describes how to set session defaults. This chapter also contains information on sessions and login scripts.
- Chapter 7 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.

Related Documents

The following manuals are related to this guide:

- *GNOME 2.0 Desktop for the Solaris Operating Environment Accessibility Guide*
- *GNOME 2.0 Desktop for the Solaris Operating Environment Installation Guide*
- *GNOME 2.0 Desktop for the Solaris Operating Environment Release Notes*
- *GNOME 2.0 Desktop for the Solaris Operating Environment Troubleshooting Guide*
- *GNOME 2.0 Desktop for the Solaris Operating Environment User Guide*

Accessing Sun Documentation Online

The docs.sun.comSM 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>.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see “Buy printed documentation” at <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 <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	<code>machine_name%</code> su Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type rm <i>filename</i> .
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> 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.

- “Introducing GConf” on page 11
- “GConf Repository” on page 12
- “GConf Daemon” on page 15
- “GConf Command Line Tool” on page 16
- “To Set Preference Values” on page 19
- “Setting General Preferences” on page 19
- “Setting Look-and-Feel Preferences” on page 27
- “To Restore Default Preference Values” on page 28

Introducing GConf

GConf simplifies the administration of preferences for users in the 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 Desktop 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 <code>.gconf</code> directory in the home directory of the user. When the user sets a preference, the new preference information is added to this location.
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: <ul style="list-style-type: none">■ 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	<code>/desktop/gnome/interface/font_name</code>
Schema key	<code>/schemas/desktop/gnome/interface/font_name</code>
Schema object	<pre><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></pre>

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

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

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

Typically, schemas are stored in the default configuration source.

GConf Schema Definition Files

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

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

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

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

GConf Daemon

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

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

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

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

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

To terminate the GConf daemon, run the following command:

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

GConf Command Line Tool

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

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

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

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

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
<code>--all-dirs</code>	Lists all subdirectories in a directory that you specify.
<code>--all-entries</code>	Displays the values of all keys in a directory that you specify.
<code>--config-source=<i>configuration-source</i></code>	Use this option with the <code>--direct</code> 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
<code>--direct</code>	Use this option with the <code>--config-source</code> option to access a configuration source directly. When you use this option, GConf bypasses the server. Ensure that the GConf daemon, <code>gconfd-2</code> , is not running before you use this option.
<code>--get</code>	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.
<code>--help</code>	Displays a help message about the <code>gconftool-2</code> command, and the options that you can use with the <code>gconftool-2</code> command.
<code>--long-desc=description</code>	Use this option with the <code>--set-schema</code> option to specify a long description for a schema key.
<code>--makefile-install-rule</code>	Installs schema definition files to applications.
<code>--owner=owner</code>	Use this option with the <code>--set-schema</code> option to specify an owner for a schema key.
<code>--recursive-list</code>	Displays the value of all preference keys in all subdirectories in a directory that you specify.
<code>--recursive-unset</code>	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.
<code>--set</code>	Sets the value of a preference key, and writes the value to the user configuration source. Use the <code>--type</code> option with the <code>--set</code> option to specify the data type of the value that you want to set. For example, the following command sets the value of the <code>/apps/gnome-terminal/profiles/Default/background_color</code> key in the user configuration source: <pre># gconftool-2 --set "/apps/gnome-terminal/profiles/Default/background_color" --type string "#000000"</pre> You can also use the <code>--direct</code> option and the <code>--config-source</code> option with the <code>--set</code> option to write a value to another configuration source.

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

Option	Function
<code>--set-schema</code>	<p>Sets the value of an attribute in a schema key, and writes the value to the default configuration source.</p> <p>Use the following options with the <code>--set-schema</code> option to specify the attribute that you want to update:</p> <ul style="list-style-type: none">■ <code>--type</code>■ <code>--short-desc</code>■ <code>--long-desc</code>■ <code>--owner</code> <p>For example, the following command sets the short description in the schema key for the <code>/apps/gnome-terminal/profiles/Default/background_color</code> key:</p> <pre># gconftool-2 --set-schema "/schemas/apps/gnome- terminal/profiles/Default/background_color" --short-desc "Default background color of terminal"</pre>
<code>--short-desc=description</code>	Use this option with the <code>--set-schema</code> option to specify a short description for a schema key.
<code>--shutdown</code>	Terminates the GConf daemon.
<code>--type=data-type</code>	<p>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:</p> <ul style="list-style-type: none">■ <code>bool</code>■ <code>float</code>■ <code>int</code>■ <code>list</code>■ <code>pair</code>■ <code>string</code>
<code>--unset</code>	Resets the value of a preference key from the user setting to the setting in the default configuration source.
<code>--usage</code>	Displays a brief help message about the <code>gconftool-2</code> command, and the options that you can use with the <code>gconftool-2</code> 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
<code>/apps/panel/global</code>	Keyboard Shortcuts preference tool, Desktop section
<code>/apps/metacity/general</code>	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 in the desktop environment:

- 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 25.

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 in the desktop environment. The following keys specify the number of panels, applets, and other panel objects that appear in the desktop environment:
 - `/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 in the desktop environment:

```
<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 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, 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.

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 29
- “File Abstraction Layer” on page 30
- “Vfolders and Menus” on page 30
- “Desktop Entry Files” on page 32
- “Directory Entry Files” on page 33
- “Editing Menus” on page 34
- “To Configure Menus That Users Cannot Modify” on page 37

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	<code>applications-all-users:///</code>
Desktop Preferences menu for all users	<code>preferences-all-users:///</code>

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>	<p>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.</p> <p>The query in the excerpt searches for desktop entry files that contain the keywords <code>Application</code> and <code>Utility</code> in the <code>Categories</code> key. Desktop entry files that match are displayed in the Applications menu.</p> <p>This element is optional.</p>
<DontShowIfEmpty/>	<p>If this element is present, the menu is not displayed if the menu does not contain any items.</p> <p>This element is optional.</p>

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	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 <code>Exec</code> key runs in a terminal window. If the value is <code>true</code> the command runs in a terminal window. If the command does not create a window in which to run, the value of this key must be <code>true</code> .

TABLE 2-3 Desktop Entry Keys (Continued)

Desktop Entry Key	Description
Type	Specifies the type of item. This value is one of the following: <ul style="list-style-type: none">■ 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	Specifies the help file to display when you choose Help on <i>application-name</i> from the menu item popup menu.
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.

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 <code>Directory</code> .

Editing Menus

You use the following desktop environment 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 *GNOME 2.0 Desktop for the Solaris Operating Environment 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 33.
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 and Menus” on page 30.

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 32.
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 *GNOME 2.0 Desktop for the Solaris Operating Environment 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 *GNOME 2.0 Desktop for the Solaris Operating Environment 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.

Installing Themes

This chapter describes the types of theme that are available in the GNOME Desktop, how to install themes, and how to create a custom theme.

- “Introduction to Themes” on page 39
- “To Install a Desktop Theme” on page 40
- “To Install a Window Frame Theme” on page 40
- “Installing Icons for Themes” on page 41
- “To Create a Custom Desktop Theme” on page 42

Introduction to Themes

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

The following types of theme affect different parts of the desktop environment:

Desktop themes	Desktop themes determine the visual appearance of windows, panels, and applets. Desktop themes also determine 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 desktop themes that are available are designed for special accessibility needs. Users can choose a desktop theme from the Widget Theme tabbed section in the Theme preference tool.
Window frame themes	Window frame themes determine the appearance of the frames around windows only. Users can choose a window frame theme from the Window Frames tabbed section in

the Theme preference tool.

To Install a Desktop Theme

Desktop themes reside in the `/usr/share/themes` directory. The typical structure of a desktop theme in the file system is as follows:

Theme file `/usr/share/themes/theme-name/gtk-2.0/gtkrc`

Image files `/usr/share/themes/theme-name/pixmaps/*.*`

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

Users can install their own desktop themes. If a user installs a desktop theme, the theme is stored in the `$HOME/.themes` directory.

To Install a Window Frame Theme

Window frame themes reside in the `/usr/share/themes/theme-name/metacity-1` directory. The typical structure of a window frame theme in the file system is as follows:

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

Image files `/usr/share/themes/theme-name/metacity-1/*.*`

Typically, a new window frame theme is supplied as a `.tar.gz` file. To install the new window frame theme, unzip the `.tar.gz` file, then untar the `.tar` file into the `/usr/share/themes` directory.

Users can install their own window frame themes. If a user installs a window frame theme, the theme is stored in the `$HOME/.themes` directory.

Installing Icons for Themes

The desktop environment 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. The themes contain several versions of each icon so that the icon can be displayed 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 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 Desktop Theme” on page 40 or “To Install a Window Frame Theme” on page 40. For example, to add icons to the HighContrast theme, add the icons to the `/usr/share/themes/HighContrast/pixmaps` directory. Add references to the icons to the relevant theme files. For example, to add icons to the HighContrast theme, add references to the icons to the `/usr/share/themes/HighContrast/gtk-2.0/gtkrc` file.

The instructions above apply only to the following types of icon:

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

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

To Create a Custom Desktop Theme

If the desktop themes are not suitable for the needs of your users, you can create a custom desktop theme. To create a custom desktop theme, perform the following steps:

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

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 desktop environment.

- “Introduction to MIME Types” on page 43
- “Detecting the MIME Type for a File” on page 44
- “Registering Applications for MIME Types” on page 48
- “Adding an Application to the Desktop Environment” on page 50

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

TABLE 4-1 Fields in a File Content Sniffer (Continued)

Field	Description
pattern_mask	Specifies a <i>pattern mask</i> , 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/gnome/share/mime-info`. The MIME type registry contains the following files:

File	Filename Extension
MIME information file	<code>.mime</code>
MIME keys file	<code>.keys</code>

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
<code>can_be_executable</code>	Specifies whether files of this MIME type can be executed.
<code>description</code>	Describes the MIME type. This description can be displayed in the file manager and other applications.
<code>icon_filename</code>	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 <code>application</code> 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 Types and Programs 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/gnome/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-ppixmap,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 <code>short_list_application_ids_for_novice_user_level</code> 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 <code>true</code> , the application registration entry must also contain a <code>supported_uri_schemes</code> 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 <code>true</code> 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 Desktop Environment

To add an application to the desktop environment, 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/gnome/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 44.
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 46.
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 47.
6. If the application uses a new MIME type, add an icon for the MIME type to `/usr/gnome/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 48.

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 51
- “Setting Screensaver Preferences” on page 52
- “Modifying Screensaver Displays” on page 53

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.

Note – The XScreenSaver application does not allow you to lock your screen if you are logged in as `root`. If you want to be able to lock your screen, log in as a normal user. To obtain system administrator privileges, open a terminal window, then use the `su` command to log in as `root`.

Setting Screensaver Preferences

Default screensaver preferences are stored in a file that is called `XScreenSaver`. This file is stored in different locations, depending on the platform, as follows:

Platform	File Location
RedHat Linux	<code>/usr/X11R6/lib/X11/app-defaults/XScreenSaver</code>
Solaris Operating Environment	<code>/usr/openwin/lib/app-defaults/XScreenSaver</code>

To modify screensaver application preferences, users can use the Screensaver preference tool. To start the Screensaver preference tool, choose Applications → Desktop Preferences → Screensaver from the Menu Panel. 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 the *GNOME 2.0 Desktop for the Solaris Operating Environment User Guide*.

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

Users can also run the command `/usr/openwin/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 `/usr/openwin/lib/xscreensaver/hacks` directory. Add the command for the screensaver display to the `XScreenSaver` file or the `$HOME/.xscreensaver` file. Include any arguments that are required to run the screensaver display on the whole screen, rather than in a window. For example, you might want to include the `-root` option to display the screensaver display on the whole screen.

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

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

To disable screensaver displays for all users on a Sun Ray™ thin client, rename the `/usr/openwin/lib/xscreensaver/hacks` directory to any name other than `hacks`.

OpenGL screensaver displays are provided with the GNOME Desktop. By default, all OpenGL screensaver displays are disabled, because many users do not have the OpenGL library installed. Users who have the OpenGL library installed can use the **XScreenSaver** dialog to enable OpenGL screensaver displays.

Some screensaver displays show the contents of the desktop environment of the user. To avoid security violations, screensaver displays that show the contents of the desktop environment are not included in the Sun Microsystems distribution of the GNOME Desktop.

Note – The Pluggable Authentication Modules (PAM) service name for the XScreenSaver application is `dtsession`. This name is used to be compatible with previous applications.

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 55
- “Setting Session Defaults” on page 56
- “Login Scripts” on page 56

Introduction to Sessions

A session occurs between the time that a user logs in to the desktop environment 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 desktop environment.
- The Metacity window manager.

Setting Session Defaults

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

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

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

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

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

Login Scripts

The GNOME Desktop does not support system-wide login scripts or logout scripts. The session manager only allows users to configure user sessions. Users can configure non-session-managed startup applications with the Sessions preference tool.

Tip – You can work around the lack of support for system-wide login scripts. To create a system-wide login script, create the script in the `/usr/dt/config/Xsession.d` directory. Name the script as follows:

number-greater-than-1000.name

For example, name the script `1050.login`. For more information on the `/usr/dt/config/Xsession.d` file, see *The XSession File* in the `dtlogin(1)` man page.

Improving Performance

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

- “Introduction to Improving Performance” on page 57
- “Reducing CPU Usage” on page 57
- “Reducing X Window System Network Traffic” on page 63
- “Reducing Color Usage and Improving Display Quality” on page 64

Introduction to Improving Performance

This chapter lists several preferences whose settings you can change to improve the performance of the desktop environment. 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 desktop environment.

To Use Themes Which Require Less CPU Resources

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

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

- Atlanta
- Esco

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

- AgingGorilla
- Bright
- Metabox

Note – Metabox does not work well with inverse desktop themes such as HighContrastInverse. Use Atlanta with inverse desktop themes.

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

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

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

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

Alternatively, users can choose Applications → Desktop Preferences → Theme, then select the appropriate theme.

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

```
# metacity-theme-viewer theme-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 choose Applications → Desktop Preferences → Menus & Toolbars, then deselect the **Show icons in menus** option.

To Reduce CPU Usage by Turning On Wireframe Mode

The Metacity window manager has a wireframe mode for when you move and resize windows. When wireframe mode is turned on, only the outline of windows is displayed when you move and resize windows. The contents of the window do not need to be updated during the move or resize. The contents of the window are displayed when the move or resize is complete.

To turn on wireframe mode, run the following command:

```
# gconftool-2 --type bool --set  
/apps/metacity/sun_extensions/wireframe_move_resize true
```

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 choose Applications → Desktop Preferences → Advanced → Sessions, then 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 choose Applications → Desktop Preferences → Advanced → Panel, then deselect the **Animation** option.

Improving File Manager Performance

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

To Modify Performance Preferences

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

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

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

Preference	Description
preview_sound	<p>Specifies when to preview the content of sound files. To never preview the content of sound files, run the following command:</p> <pre># gconftool-2 --type string --set /apps/nautilus/preferences/preview_sound never</pre> <p>Alternatively, users can perform the following steps:</p> <ol style="list-style-type: none"> 1. Choose Edit → Preferences from a file manager window, then choose Performance. 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
```

Alternatively, users can choose Edit → Preferences from a file manager window, then choose **Windows**, and then deselect the **Display side pane** option. The next time that users open a file manager window, the window does not display a side pane.

To turn off the toolbar, run the following command:

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

Alternatively, users can choose Edit → Preferences from a file manager window, then choose **Windows**, and then deselect the **Display toolbar** option. The next time that users open a file manager window, the window does not display a toolbar.

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
```

Alternatively, users can choose Edit → Preferences from a file manager window, then choose **Windows**, and then deselect the **Display location bar** option. The next time that users open a file manager window, the window does not display a location bar.

To Turn Off the Desktop

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

- Use the Desktop menu.
- Use the file manager to change the pattern or color of the desktop background.
- Use the **Home** object and the **Trash** object. The **Home** object and the **Trash** object 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
```

Alternatively, users can choose Edit → Preferences from a file manager window. Users can then choose **Desktop & Trash**, and then deselect the **Use Nautilus to draw the desktop** option.

To Use the mediaLib Libraries

The mediaLib™ libraries are a collection of functions that accelerate common multimedia operations. The desktop environment uses the mediaLib libraries if the libraries are installed on the system. The performance of the desktop environment, especially Nautilus, improves significantly when the mediaLib libraries are installed. The desktop environment requires mediaLib version 2.0 or higher.

For information on how to obtain and install the mediaLib libraries, see the mediaLib website at <http://www.sun.com/processors/vis/mlib.html>.

Reducing X Window System Network Traffic

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

To Use Themes 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 that uses solid colors. That is, use one of the following window frame themes:

- Atlanta
- Esco

For information on how to change the theme, see “To Use Themes Which Require Less CPU Resources” on page 58.

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 example, if a Sun Ray server runs the panels and displays the panels on a Sun Ray client.

For information on how to turn off this feature, see “To Reduce CPU Usage by Turning Off Display of Icons in Menus” on page 59.

To Reduce Network Traffic by Turning On Wireframe Mode

If the application that displays the window is running on a remote host, wireframe mode reduces X Window System network traffic.

For more information on wireframe mode, see “To Reduce CPU Usage by Turning On Wireframe Mode” on page 59.

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 desktop environment 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 desktop environment 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 desktop environment for systems that support 8-bit color.

To Use Themes That Use the Websafe Color Palette

Some window frame themes 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 themes 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 the theme, see “To Use Themes Which Require Less CPU Resources” on page 58.

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 59.

To Reduce Color Usage by Turning Off the Splash Screen

You can turn off the splash screen to make more colors available for the desktop environment 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 59.

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

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

To set a solid color for the desktop 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 choose Applications → Desktop Preferences → Background, then choose a solid color for the desktop background.

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: <ul style="list-style-type: none">■ Configuration information.■ Lock information for objects that are referenced by an <i>Interoperable Object Reference (IOR)</i>.■ 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
<code>.gnome-desktop</code>	<p>The Nautilus file manager contains a preference that enables users to use the file manager to manage the desktop. If the user selects this option, this directory contains the objects that are on the desktop of the user. This directory contains the following:</p> <ul style="list-style-type: none">■ Objects on the desktop, for example, the Home object, the Trash object, and other launchers. The objects appear in the directory as desktop entry files. For example, the <code>starthere.desktop</code> file contains a link to the Start Here location.■ Removable media volumes that are mounted. <p>The file manager also contains a preference that enables users to use the home directory as the desktop directory, instead of <code>.gnome-desktop</code>. If a user selects this option, the contents of the home directory are displayed as desktop objects.</p>
<code>.gnome2</code>	<p>Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains the following:</p> <ul style="list-style-type: none">■ Keyboard shortcut information.■ Window location information.■ Desktop entry files for panel launchers. <p>This directory also contains user-specific menu data. If a user modifies menus, the details are stored here.</p>
<code>.gnome2-private</code>	<p>Ignore this directory. This directory has no function at the time of publication of this guide.</p>
<code>.metacity</code>	<p>Contains session data for the Metacity window manager.</p>
<code>.nautilus</code>	<p>Contains file manager data that is specific to the user. For example, this directory can contain the following:</p> <ul style="list-style-type: none">■ Metadata for the directories with which the user works.■ Nautilus themes that the user adds.■ Nautilus emblems that the user adds.■ Nautilus desktop images.

TABLE A-1 Hidden Directories in User Home Directories (Continued)

Directory	Description
<code>.themes</code>	Contains desktop themes and window frame themes that the user adds. The user can add themes from the Theme preference tool.
<code>.thumbnails</code>	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.
<code>.xscreensaver</code>	Contains screensaver configuration data and screensaver preference data.

Glossary

.desktop file	See desktop entry file.
.directory file	See directory entry file.
application registry	The application registry is a location that contains text files which register applications. The location of the application registry for the GNOME Desktop is <code>/usr/share/gnome/application-registry</code> .
desktop entry file	A data file that provides information about an item in a menu. The desktop entry file specifies the details for the item such as a name, a command to run, an icon, and so on. Desktop entry files have a <code>.desktop</code> file extension.
directory entry file	A data file that provides information about a menu. The directory entry file specifies details such as a name for the menu, a tooltip for the menu, and an icon to represent the menu. Directory entry files have a <code>.directory</code> file extension.
file content sniffer	A file content sniffer specifies a pattern to search for in a file. A file content sniffer associates the pattern with a MIME type. If a match for the pattern is found, the MIME type associated with the pattern is the MIME type of the file.
GConf configuration source	A storage location in the GConf repository. For example, <code>xml:readonly:/etc/gconf/gconf.xml.defaults</code> .
GConf preference key	An element in the GConf repository that corresponds to an application preference. For example, the <code>/apps/gnome-session/options/show_splash_screen</code> key corresponds to the Show splash screen on login option in the Sessions preference tool.
GConf path file	A file that lists the GConf configuration sources, and the order in which to search the sources.
GConf schema	A collective term for a schema key and a schema object.

GConf schema definition file	A GConf schema definition file lists the keys in a particular application, and defines the characteristics of the keys. GConf schemas are generated from schema definition files. Schema definition files have a <code>.schemas</code> extension.
GConf schema key	A key that stores a schema object for a preference key. For example, <code>/schemas/desktop/gnome/interface/font_name</code> is a schema key for the <code>/desktop/gnome/interface/font_name</code> preference key.
GConf schema object	An element in a configuration source that contains information about a preference key. The schema object contains information such as a default value for the preference key, and documentation on the preference key.
Interoperable Object Reference	An Interoperable Object Reference (IOR) is a string reference to a CORBA object. An IOR encodes a hostname and port to which messages can be sent to control the object. The IOR also contains an object key to identify the object.
MIME	Multipurpose Internet Mail Extension.
MIME information file	A MIME information file is a text file that associates MIME types with filename extensions and filename patterns. MIME information files have a <code>.mime</code> file extension.
MIME keys file	A MIME keys file provides information about a MIME type that is used in the user interface. For example, the MIME keys file specifies an icon to represent files of that MIME type. MIME keys files have a <code>.keys</code> file extension.
MIME type	A MIME type identifies the format of a file. The MIME type enables applications to read the file. For example, an email application can use the MIME type to detect what type of file is in a file attached to an email.
MIME type registry	The MIME type registry is a location that contains text files which register MIME types for the desktop environment. The location of the MIME type registry for the GNOME Desktop is <code>/usr/gnome/share/mime-info</code> .
pattern mask	A pattern mask is a series of hexadecimal characters in a file content sniffer. The pattern mask identifies bits in the pattern to ignore when searching for a pattern in a file.
screensaver	A screensaver is an application that replaces the image on a screen when the screen is not in use. The screensaver application for the GNOME Desktop is 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.

Uniform Resource Identifier	A Uniform Resource Identifier (URI) is a string that identifies a particular location in a file system or on the Web. For example, the address of a web page is a URI.
vfolder	A virtual representation of items that reside in a physical location or physical locations on your system. For example, a vfolder might represent the contents of several directories. In terms of menus, a vfolder is a representation in a menu of items that might be physically located in several directories.
vfolder information file	An XML file that describes a vfolder. Vfolder information files specify the structure of your menus.
websafe color palette	The websafe color palette is a general-purpose palette of 216 colors. The websafe color palette is designed to optimize the use of color on systems that support 8-bit color. The websafe color palette is also called the <i>Netscape color palette</i> and the <i>Netscape color cube</i> .

Index

Numbers and Symbols

* directories, *See* hidden directories

A

applications
 adding, 50
 registry, 48

C

color usage, reducing, 64
CPU usage, reducing, 57

D

desktop
 font preferences, setting with GConf, 27
 turning off, 63
desktop background
 preferences, setting with GConf, 27
 using solid color, 66
desktop entry files, 32
.desktop files, *See* desktop entry files
desktop themes, *See* themes
detecting MIME types, 44
directory entry files, 33
.directory files, *See* directory entry files
display quality, improving, 64

F

file abstraction layer, and menus, 30
file content sniffers, 44
file manager
 adding menus with, 34
 desktop, turning off, 63
 location bar, turning off, 62
 modifying performance preferences, 60
 side pane, turning off, 62
 toolbar, turning off, 62
fonts, setting with GConf, 27

G

GConf
 command line tool, 16
 configuration sources, 13
 daemon, 15
 desktop background, setting, 27
 fonts, setting, 27
 global panel preferences, setting, 22
 HTTP proxy, setting, 20
 introduction, 11
 keyboard accessibility, setting, 21
 keyboard shortcuts, setting, 21
 number of workspaces, setting, 20
 panel object preferences, setting, 22
 panel preferences, setting individual, 22
 preference values, setting, 19
 print manager, setting, 20
 repository, 12
 restoring default preference values, 28

GConf (Continued)
 schema definition files
 introduction, 15
 panel and panel object, 23
 schema keys, 14
 schema objects, 14
 schemas, 14
 splash image, setting, 28
gconfd, *See* GConf daemon
gconftool-2, *See* GConf command line tool

H
hidden directories, 67
HTTP proxy, setting with GConf, 20

I
icons, in menus, *See* menu icons

K
keyboard
 setting accessibility preferences with
 GConf, 21
 setting shortcut preferences with GConf, 21

M
mediaLib libraries, and CPU usage, 63
menu icons
 and color usage, 65
 and CPU usage, 59
 and network traffic, 64
 turning off, 59
menus
 adding items to, 35
 adding using file manager, 34
 adding using menu files, 35
 configuring menus that users cannot
 modify, 37
 deleting menu items, 36
 editing, 34
 editing menu items, 36

menus (Continued)
 editing properties of, 36
 file abstraction layer, 30
 introduction, 29
 vfolder information files, 30
 vfolders, 30
MIME types
 adding applications, 50
 detecting, 44
 file content sniffers, 44
 introduction, 43
 MIME information files, 46
 MIME keys files, 47
 MIME type registry, 45
 pattern mask, 45
 registering applications for, 48

N
Nautilus, *See* file manager
network traffic, reducing, 63

P
panels
 animation, turning off, 60
 preferences, individual, setting with
 GConf, 22
 setting global preferences with GConf, 22
pattern mask, 45
performance, improving
 introduction, 57
 reducing CPU usage, 57
preference values
 restoring to default with GConf, 28
 setting with GConf, 19
print manager, setting with GConf, 20

R
reducing
 color usage, 64
 CPU usage, 57
 X Window System network traffic, 63

S

schemas

- description, 14
- schema definition files
 - introduction, 15
 - panel and panel object, 23
- schema keys, 14
- schema objects, 14

screensavers

- introduction, 51
- modifying screensaver displays, 53
- setting preferences, 52

sessions

- introduction, 55
- login scripts, 56
- setting defaults, 56

splash screen

- and color usage, 65
- and CPU usage, 59
- image, setting with GConf, 28
- turning off, 59

T

themes

- and color usage, 65
- and CPU usage, 58
- and network traffic, 63
- changing, 58
- creating custom desktop, 42
- desktop theme, installing, 40
- installing icons, 41
- introduction, 39
- window frame theme, installing, 40

V

vfolder information files, 30

vfolders, 30

W

window frame themes, *See* themes

wireframe mode

- and CPU usage, 59

wireframe mode (Continued)

and network traffic, 64

turning on, 59

workspaces, setting number with GConf, 20

X

X Window System network traffic, reducing, 63

XScreenSaver, *See* screensavers

